



MobiLink™

Erste Orientierung

Version 16.0

Februar 2013

Version 16.0
Februar 2013

© 2013 SAP AG oder ein SAP-Konzernunternehmen. Alle Rechte vorbehalten.

Sie können diese Dokumentation (ganz oder teilweise) unter folgenden Bedingungen benutzen, reproduzieren und verteilen: 1) Sie müssen diese und alle anderen Urheberrechtsvermerke auf allen Kopien oder Auszügen der Dokumentation wiedergeben. 2) Sie dürfen die Dokumentation nicht verändern. 3) Sie dürfen nichts tun, aus dem abgeleitet werden könnte, dass Sie oder jemand anderer als SAP Verfasser oder Quelle der Dokumentation ist. Die hier enthaltenen Informationen können jederzeit ohne vorherigen Hinweis geändert werden.

Einige Softwareprodukte, die von der SAP AG oder einem ihrer Vertriebspartner vermarktet werden, enthalten Softwarekomponenten anderer Softwareanbieter. Die nationalen Produktspezifikationen können unterschiedlich sein.

Diese Dokumentationen werden von der SAP AG und ihren Tochtergesellschaften ("SAP Group") lediglich zu Informationszwecken bereitgestellt, ohne dass eine Gewährleistung oder eine Garantie irgendeiner Art gegeben wird. Die SAP Group übernimmt keine Verantwortung im Hinblick auf Fehler oder Auslassungen in den Dokumentationen. Die einzigen Garantien für Produkte und Dienstleistungen der SAP Group sind diejenigen, die in den mit den Produkten und Dienstleistungen eventuell gelieferten ausdrücklichen Garantieerklärungen enthalten sind. Keine der hier enthaltenen Informationen kann als Gewährung einer weitergehenden Garantie betrachtet werden.

SAP und weitere erwähnte SAP-Produkte und -Dienstleistungen sowie die entsprechenden Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und anderen Ländern. Weitere Hinweise finden Sie unter <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark>.

Inhalt

Über diese Dokumentation	v
MobiLink-Technologie	1
MobiLink-Synchronisation	1
MobiLink-Plug-In für Sybase Central	22
MobiLink-Beispiel CustDB	60
MobiLink-Beispielanwendung "Contact"	75
Praktische Einführungen in MobiLink	89
Praktische Einführung: Einführung in MobiLink	89
Praktische Einführung: MobiLink mit einer konsolidierten SQL Anywhere-Datenbank verwenden	110
Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden	123
Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden	142
Praktische Einführung: Der Einsatz von Java oder .NET für die benutzerdefinierte Authentifizierung	163
Praktische Einführung: Verwenden der direkten Zeilenbehandlung	172
Praktische Einführung: Synchronisation mit Microsoft Excel	199
Praktische Einführung: Synchronisation mit XML	219
Praktische Einführung: Zentrale Administration von entfernten Datenbanken	241
Praktische Einführung: Änderung eines Schemas mit der Skriptversionsklausel	268
Praktische Einführung: Änderung eines Schemas mit der erweiterten Option ScriptVersion	277
Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm	284
Index	301

Über diese Dokumentation

In diesem Handbuch wird MobiLink, das sitzungsbasierte Synchronisationssystem für relationale Datenbanken beschrieben. Die MobiLink-Technologie ermöglicht die Zwei-Wege-Replikation und ist besonders für den Einsatz in Systemen mit zahlreichen mobilen Endgeräten geeignet.

MobiLink-Technologie

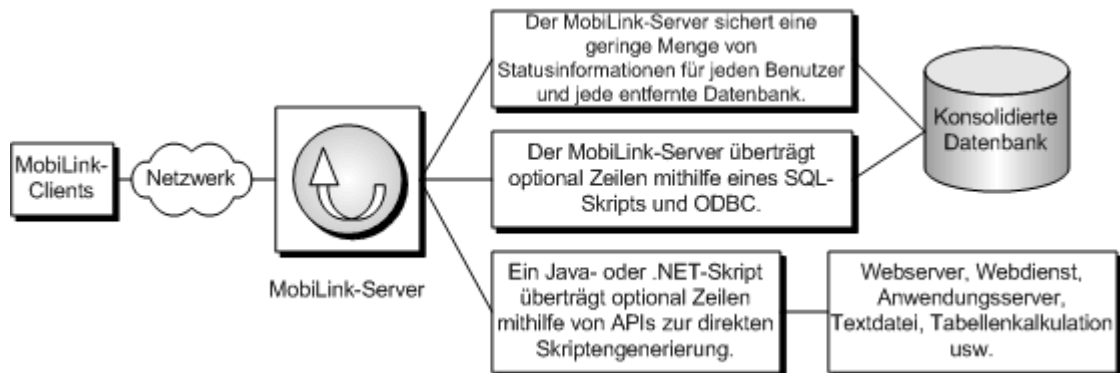
Dieser Abschnitt bietet eine Einführung in die MobiLink-Synchronisationstechnologie und beschreibt, wie sie für die Replikation von Daten zwischen zwei und mehr Datenbanken eingesetzt werden kann.

MobiLink-Synchronisation

MobiLink ist eine sitzungsbasierte Synchronisationstechnologie für die Synchronisation von entfernten UltraLite- und SQL Anywhere-Datenbanken mit einer konsolidierten Datenbank.

Bestandteile der MobiLink-Anwendung

In einer typischen MobiLink-Synchronisation führen zahlreiche Clients eine Synchronisation mit zentralen Datenquellen über den MobiLink-Server durch.



- **MobiLink-Clients** Der Client kann auf einem mobilen Gerät, einem Server, einem PC oder einem Smartphone installiert werden. Es werden zwei Typen von Clients unterstützt, nämlich UltraLite- und SQL Anywhere-Datenbanken. In einer MobiLink-Installation können einer der beiden Typen von Clients oder beide zusammen benutzt werden. Siehe „[MobiLink-Clients](#)“ [[MobiLink - Clientadministration](#)].
- **Netzwerk** Für die Verbindung zwischen dem MobiLink-Server und dem MobiLink-Client können mehrere Protokolle benutzt werden. Siehe:
 - MobiLink-Server: „[mlsrv16-Option -x](#)“ [[MobiLink - Serveradministration](#)]
 - UltraLite- und SQL Anywhere-Clients: „[Netzwerkprotokolloptionen des MobiLink-Clients](#)“ [[MobiLink - Clientadministration](#)]
- **MobiLink-Server** Dieser Server verwaltet den Synchronisationsprozess und bildet die Schnittstelle zwischen allen MobiLink-Clients und dem Server der konsolidierten Datenbank. Siehe „[MobiLink-Server](#)“ [[MobiLink - Serveradministration](#)].
- **Konsolidierte Datenbank** In dieser Datenbank befindet sich normalerweise die zentrale Kopie der Anwendungsdaten im Synchronisationssystem. Sie enthält für gewöhnlich Systemtabellen und

Prozeduren, die für eine MobiLink-Synchronisation erforderlich sind, sowie Statusinformationen, die für eine Synchronisation benötigt werden. Siehe „[MobiLink - konsolidierte Datenbanken](#)“ [[MobiLink - Serveradministration](#)].

- **Statusinformationen** Der MobiLink-Server speichert für gewöhnlich Synchronisationsinformationen in Systemtabellen in der konsolidierten Datenbank. Dies geschieht über eine ODBC-Verbindung.

Sie können auch wählen, Statusinformationen in einer anderen Datenbank zu speichern. Siehe „[MobiLink-Systemdatenbank](#)“ [[MobiLink - Serveradministration](#)].

- **Zeilenbehandlung über SQL** Wenn Sie dem MobiLink-Server SQL-Skripten zur Verfügung stellen, werden diese Skripten für die Übermittlung der Zeilen in und aus der konsolidierten Datenbank über eine ODBC-Verbindung verwendet. Siehe „[Optionen zum Erstellen serverseitiger Synchronisationslogik](#)“ auf Seite 12.
- **Direkte Zeilenbehandlung** Zusätzlich zur konsolidierten Datenbank können Sie mithilfe der direkten Zeilenbehandlung in MobiLink auch eine Synchronisation mit anderen Datenquellen durchführen. Siehe „[Direkte Zeilenbehandlung](#)“ [[MobiLink - Serveradministration](#)].
- **Synchronisationsskripten** Sie schreiben für jede Tabelle in der entfernten Datenbank Synchronisationsskripten und speichern diese in den MobiLink-Systemtabellen in der konsolidierten Datenbank. Diese Skripten legen fest, was mit den heraufgeladenen Daten geschieht und welche Daten heruntergeladen werden sollen. Es gibt zwei Arten von Skripten: Tabellenskripten und Verbindungsskripten. Siehe:
 - „[Überblick über MobiLink-Ereignisse](#)“ [[MobiLink - Serveradministration](#)]
 - „[Schreiben von Synchronisationsskripten](#)“ [[MobiLink - Serveradministration](#)]
 - „[Synchronisationsereignisse](#)“ [[MobiLink - Serveradministration](#)]
 - „[Optionen zum Erstellen serverseitiger Synchronisationslogik](#)“ auf Seite 12

MobiLink-Funktionen

Die MobiLink-Synchronisation ist anpassbar und flexibel. Sie bietet unter anderem folgende Hauptfunktionen:

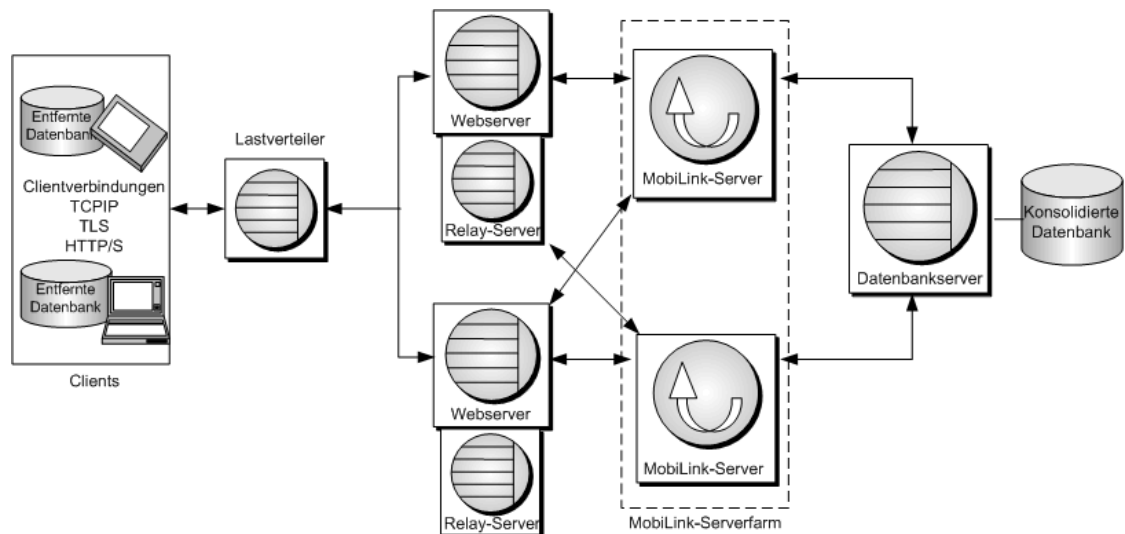
Funktionen

- **Einfacher Einstieg** Mit dem **Assistenten zum Erstellen eines Synchronisationsmodells** können Sie Synchronisationsanwendungen schnell erstellen. Der Assistent übernimmt viele schwierige Implementierungsdetails für komplexe Synchronisationssysteme. Sybase Central bietet die Möglichkeit, ein Synchronisationsmodell offline zu betrachten, stellt eine einfache Oberfläche zum Durchführen von Änderungen zur Verfügung und bietet eine Deployment-Option, mit der Sie das Deployment des Modells in Ihrer konsolidierten Datenbank durchführen können.
- **Monitoring und Berichterstellung** MobiLink bietet drei Verfahren für das Monitoring der Synchronisationen: MobiLink-Profiler, SQL Anywhere-Monitor für MobiLink und statistische Skripten.

- **Performance-Optimierung** Sie haben die Möglichkeit, mehrere Verfahren zur Optimierung der MobiLink-Performance einzusetzen. Sie können z.B. den Konfliktgrad, die Größe des Upload-Caches, die Anzahl der Datenbankverbindungen, die Ausführlichkeit der Protokollierung oder die BLOB-Cachegröße einstellen.
- **Skalierbarkeit** MobiLink ist eine äußerst skalierbare und robuste Synchronisationsplattform. Ein MobiLink-Server kann Tausende von gleichzeitigen Synchronisationen verwalten und mehrere MobiLink-Server können gleichzeitig ausgeführt werden, wobei eine Lastverteilung stattfindet. Der MobiLink-Server verwendet mehrere Threads und ein Verbindungspooling zur konsolidierten Datenbank.
- **Sicherheit** MobiLink bietet umfassende Sicherheitsoptionen, einschließlich einer Benutzerauthentifizierung, die in Ihr bestehendes Authentifizierungssystem integriert werden kann, Verschlüsselungsoptionen und Transportschichtsicherheit, die über den Austausch von Sicherheitszertifikaten funktioniert. MobiLink verfügt außerdem über FIPS-zertifizierte Sicherheitsoptionen.
- **Relay Server und Sybase Relay Server-Hostingdienst** Der Relay Server ermöglicht die sichere und ausgewogene Kommunikation zwischen mobilen Geräten und Backend-Servern, die über einen Webserver Daten austauschen. Siehe „Einführung in den Relay Server“ [\[Relay Server\]](#).

Der Sybase Relay Server-Hostingdienst ist eine Farm von Relay Servern, die bei Sybase gehostet werden und Ihnen bei der Entwicklung und Auswertung mobiler Anwendungen helfen sollen, die MobiLink-Datensynchronisation verwenden, insbesondere wenn Daten über öffentliche Drahtlosnetze übertragen werden. Siehe „Sybase Hosted Relay Service“ [\[Relay Server\]](#).

Die folgenden Abbildung zeigt, wie sich Relay Server in eine MobiLink-Umgebung einfügen.



Architektur

- **Datenkoordination** MobiLink bietet die Möglichkeit, bestimmte Datenbereiche für die Synchronisation auszuwählen. Mit der MobiLink-Synchronisation können Sie außerdem Konflikte

zwischen den in den unterschiedlichen Datenbanken vorgenommenen Änderungen lösen. Der Synchronisationsprozess wird von der Synchronisationslogik gesteuert, die als SQL-, Java- oder .NET-Anwendung geschrieben werden kann. Die einzelnen Teile dieser Logik werden jeweils als **Skript** bezeichnet. Mit Skripten können Sie zum Beispiel festlegen, wie übertragene Daten in die konsolidierte Datenbank übernommen und welche Daten heruntergeladen werden, sowie unterschiedliche Schemata und Namen in der konsolidierten und der entfernten Datenbank verwalten. Ereignisbasierte Skripten bieten hohe Flexibilität bei der Konzeption des Synchronisationsprozesses, z.B. im Hinblick auf Konfliktlösung, Fehlerberichterstellung und Benutzerauthentifizierung.

- **Zwei-Wege-Synchronisation** Änderungen in einer Datenbank können an jedem beliebigen Standort vorgenommen werden.
- **Reine Upload- oder Download-Synchronisation** Die Synchronisation erfolgt standardmäßig in zwei Richtungen, also mit Upload und Download. Sie können aber auch eine Synchronisation durch reinen Upload oder Download wählen.
- **Dateibasierter Download** Downloads können als Dateien verteilt werden, wodurch eine Offline-Verteilung von Synchronisationsänderungen möglich ist. Mit dieser Funktion wird auch sichergestellt, dass die korrekten Daten verwendet werden.
- **Serverinitiierte Synchronisation** Sie können die MobiLink-Synchronisation von der konsolidierten Datenbank aus starten. Das bedeutet, dass Sie Datenaktualisierungen in entfernte Datenbanken bzw. Daten aus entfernten Datenbanken in die konsolidierte Datenbank übertragen können. Siehe [MobiLink - Serverinitiierte Synchronisation](#).

Sie können serverinitiierte entfernte Aufgaben (SIRT für "server-initiated remote task") als Alternative zur serverinitiierten Synchronisation verwenden. Weitere Hinweise finden Sie unter „Zentrale Administration von entfernten Datenbanken“ [[MobiLink - Serveradministration](#)] und „Serverinitiierte entfernte Aufgaben (SIRT)“ [[MobiLink - Serveradministration](#)].

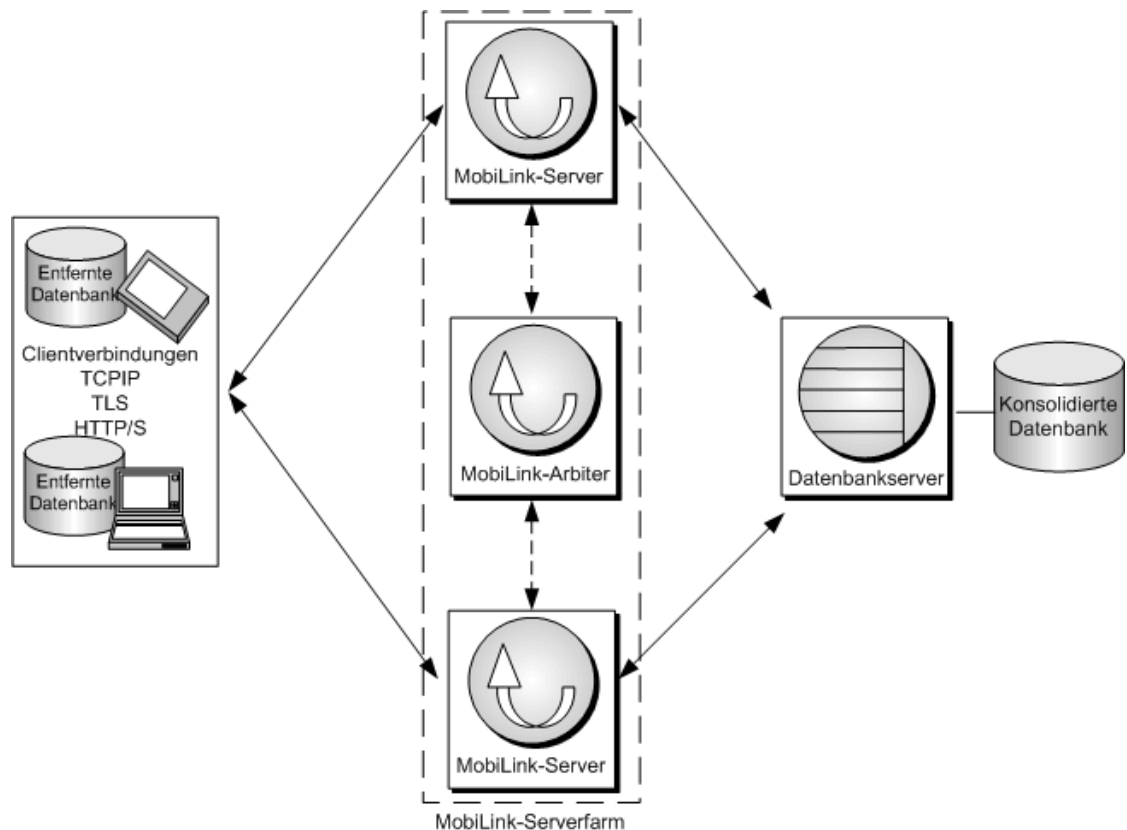
- **Wahl des Netzwerkprotokolls** Die Synchronisation kann über TCP/IP, HTTP oder HTTPS durchgeführt werden. Windows Mobile-Geräte können mithilfe von Microsoft ActiveSync synchronisiert werden.
- **Sitzungsbasiert** Alle Änderungen können in beiden Richtungen jeweils in nur einer Transaktion übertragen werden. Am Ende jeder einzelnen Synchronisationssitzung stimmen die entfernte Datenbank und die konsolidierte Datenbank überein. (Um die Reihenfolge von Transaktionen beizubehalten, können Sie auch wählen, jede Transaktion in die entfernte Datenbank als separate Transaktion hochzuladen.)

Eine Transaktion wird entweder ganz synchronisiert oder gar nicht. So wird die Transaktionsintegrität für die einzelnen Datenbanken gewährleistet.

- **Datenkonsistenz** MobiLink arbeitet gemäß der Strategie der losen Konsistenz. Alle Änderungen werden irgendwann auf konsistente Weise mit jedem Standort synchronisiert, aber einzelne Standorte können zu einem bestimmten Zeitpunkt unterschiedliche Datenbestände aufweisen.
- **Breite Palette von Hardware- und Softwareplattformen** Als konsolidierte Datenbanken für MobiLink stehen viele Datenbank-Managementsysteme zur Verfügung. Mit der MobiLink-Server-API können Sie auch eine Synchronisation mit einer beliebigen Datenquelle definieren. Entfernte

Datenbanken können SQL Anywhere- oder UltraLite-Datenbanken sein. Der MobiLink-Server kann unter Windows, Unix und Mac OS X ausgeführt werden. SQL Anywhere kann unter Windows, Windows Mobile oder Unix, Linux und Mac OS X ausgeführt werden. UltraLite kann unter Windows Mobile oder BlackBerry ausgeführt werden. Siehe „Unterstützte Plattformen“ [[SQL Anywhere 16 - Einführung](#)].

- **MobiLink-Arbiter** Ein MobiLink-Arbiter stellt sicher, dass nur ein einziger MobiLink-Server in einer Serverfarm als Primärserver läuft. Dies verhindert redundante Benachrichtigungen in einer serverinitiierten Synchronisationsumgebung. Die folgende Abbildung zeigt den MobiLink-Arbiter in einer Serverfarmumgebung.



Schnellstart mit MobiLink

MobiLink dient der Synchronisation von Daten zwischen zahlreichen entfernten Anwendungen, die über eine oder mehrere zentrale Datenquellen miteinander verbunden sind. In einer grundlegenden MobiLink-Anwendung sind die entfernten Clients SQL Anywhere- oder UltraLite-Datenbanken und die zentrale Datenquelle ist eine der unterstützten ODBC-kompatiblen relationalen Datenbanken. Diese Architektur kann durch die MobiLink-Server-API erweitert werden. Es gibt serverseitig praktisch keine Einschränkungen im Hinblick auf die Synchronisation.

In allen MobiLink-Anwendungen ist der MobiLink-Server der Schlüssel für die Synchronisationsvorgänge. Die Synchronisation beginnt für gewöhnlich, wenn ein entfernter MobiLink-Standort eine Verbindung mit einem MobiLink-Server herstellt. Während der Synchronisation überträgt der MobiLink-Client am entfernten Standort Datenbankänderungen, die seit der vorherigen Synchronisation in der entfernten Datenbank vorgenommen wurden. Nach Erhalt dieser Daten aktualisiert der MobiLink-Server die konsolidierte Datenbank und überträgt anschließend alle Änderungen der konsolidierten Datenbank auf die entfernte Datenbank.

Eine schnelle Möglichkeit, die Entwicklung einer MobiLink-Anwendung zu starten, bietet der **Assistent zum Erstellen eines Synchronisationsmodells**. Der Assistent führt die meisten der im Folgenden beschriebenen Schritte für Sie aus. Siehe „[Synchronisationsmodelle](#)“ auf Seite 30.

Auch bei der Verwendung eines MobiLink-Modells müssen Sie jedoch den Prozess und die Komponenten der MobiLink-Synchronisation kennen.

Überblick über eine MobiLink-Anwendung

1. Einrichten einer konsolidierten Datenbank

- Führen Sie Setupskripten in der Datenbank aus, um von der MobiLink-Synchronisation benötigte Systemobjekte hinzuzufügen. Alternativ dazu können Sie auch eine separate Systemdatenbank zur Speicherung dieser Objekte erstellen.

Siehe „[MobiLink - konsolidierte Datenbanken](#)“ [[MobiLink - Serveradministration](#)].

2. Richten Sie entfernte Datenbanken ein

- Die entfernten Datenbanken können SQL Anywhere, UltraLite oder eine Kombination der beiden sein.
- Erstellen Sie MobiLink-Benutzer in den entfernten Datenbanken. Siehe „[MobiLink-Benutzer](#)“ [[MobiLink - Clientadministration](#)].
- Erstellen Sie Publikationen und Subskriptionen, um den Upload in eine entfernte SQL Anywhere-Datenbank festzulegen. Siehe „[Publikationen](#)“ [[MobiLink - Clientadministration](#)].
Erstellen Sie Publikationen, um den Upload in eine entfernte UltraLite-Datenbank festzulegen. Siehe „[Publizieren von Daten in UltraLite](#)“ [[UltraLite - Datenbankverwaltung](#)].

3. Erstellen Sie Server-Synchronisationslogik, um festzulegen, wie der Upload durchgeführt wird

Siehe „[Schreiben von Synchronisationsskripten](#)“ [[MobiLink - Serveradministration](#)].

4. Richten Sie die Synchronisation auf Zeitstempelbasis ein, um Daten herunterzuladen, die seit dem letzten Download geändert wurden.

Siehe „[Implementieren zeitstempelbasierter Downloads](#)“ [[MobiLink - Serveradministration](#)].

5. Starten des MobiLink-Servers

Siehe „[MobiLink-Server](#)“ [[MobiLink - Serveradministration](#)].

6. Starten Sie die Synchronisation auf dem Client

- Weitere Hinweise zu entfernten SQL Anywhere-Datenbanken finden Sie unter „[Initiieren einer Synchronisation](#)“ [[MobiLink - Clientadministration](#)]

- Hinweise zu entfernten UltraLite-Datenbanken finden Sie unter „UltraLite-Clientsynchronisationsplanung“ [*UltraLite - Datenbankverwaltung*].

Einführende Informationen

- „MobiLink-Synchronisation“ auf Seite 1
- „Synchronisationsmethoden“ [*MobiLink - Serveradministration*]

Praktische Einführungen

- „MobiLink-Beispiel CustDB“ auf Seite 60
- „MobiLink-Beispielanwendung "Contact"" auf Seite 75
- „Praktische Einführung: Einführung in MobiLink“ auf Seite 89
- „Praktische Einführung: MobiLink mit einer konsolidierten SQL Anywhere-Datenbank verwenden“ auf Seite 110
- „Praktische Einführung: Erstellen der Beispielanwendung CustDB“ [*UltraLite - Datenbankverwaltung*]
- „Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden“ auf Seite 123
- „Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden“ auf Seite 142
- „Praktische Einführung: Der Einsatz von Java oder .NET für die benutzerdefinierte Authentifizierung“ auf Seite 163
- „Praktische Einführung: Verwenden der direkten Zeilenbehandlung“ auf Seite 172
- „Praktische Einführung: Synchronisation mit Microsoft Excel“ auf Seite 199
- „Praktische Einführung: Synchronisation mit XML“ auf Seite 219
- „Praktische Einführung: Zentrale Administration von entfernten Datenbanken“ auf Seite 241
- „Praktische Einführung: Änderung eines Schemas mit der Skriptversionsklausel“ auf Seite 268
- „Praktische Einführung: Änderung eines Schemas mit der erweiterten Option ScriptVersion“ auf Seite 277
- „Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm“ auf Seite 284

Weitere Ressourcen für die ersten Schritte

- MobiLink stellt Beispiele bereit, die Sie sich ansehen und ausführen können, um die Funktionen von MobiLink kennenzulernen. MobiLink-Beispiele werden mit dem Produkt im Verzeichnis %SQLANYSAMPI6%\MobiLink installiert.
- Beispiele für den Austausch von MobiLink-Code finden Sie unter <http://www.sybase.com/detail?id=1058600#319>. Sie benötigen ein Login für Sybase.com, um diese Seite anzuzeigen.

MobiLink-Anwendungsdesign

Es gibt zwei grundlegende Architekturen für Datenbankanwendungen:

- **Online-Anwendungen** Benutzer aktualisieren Daten, indem Sie sich direkt mit der zentralen Datenbank verbinden. Wenn eine Verbindung nicht verfügbar ist, kann der Benutzer nicht arbeiten.

- **Gelegentlich verbundene Smart Client-Anwendungen** Jeder Benutzer hat eine lokale Datenbank. Ihre Datenbankanwendung ist immer verfügbar, unabhängig von der Systemanbindung, und wird mit anderen Datenbanken im System synchronisiert.

MobiLink ist für die Erstellung gelegentlich verbundener Smart Client-Anwendungen ausgelegt. Smart Client-Anwendungen können die Benutzerfreundlichkeit, Effizienz und Skalierbarkeit einer Anwendung erheblich steigern, doch sie stellen auch neue Anforderungen an Anwendungsentwickler. Dieser Abschnitt beschreibt einige der wichtigsten Anforderungen von Smart Client-Anwendungen für Entwickler. Außerdem wird erläutert, wie Sie Lösungen in einer MobiLink-Synchronisationsumgebung implementieren können.

Nur die erforderlichen Daten synchronisieren

In den meisten Anwendungen wäre es äußerst problematisch, wenn die gesamte konsolidierte Datenbank heruntergeladen werden müsste, wenn bestimmte Daten auf dem entfernten Gerät aktualisiert werden sollen. Der Zeitaufwand und die Netzauslastung wären enorm und würden das gesamte System praktisch zum Stillstand bringen. Es gibt verschiedene Verfahren, um sicherzustellen, dass der Upload und der Download nur die von den verschiedenen Benutzern benötigten Elemente umfassen.

Zunächst sollte eine entfernte Datenbank nur eine Teilmenge der Tabellen und Spalten in der konsolidierten Datenbank umfassen. Ein Vertriebsmitarbeiter in der Region A benötigt beispielsweise andere Tabellen und Spalten als ein Vertriebsmitarbeiter in der Region B oder ein Vorgesetzter.

Von den Tabellen und Spalten, die auf einem entfernten Gerät zur Verfügung gestellt werden, sollten nur diejenigen für die Synchronisation vorgesehen werden, die tatsächlich synchronisiert werden müssen. Sie können in einer MobiLink-Anwendung Tabellen und Spalten unabhängig von ihren Namen zuordnen, sofern ihre Datentypen übereinstimmen. Daten werden standardmäßig hoch- und heruntergeladen, doch Sie können in MobiLink auch festlegen, dass bestimmte Spalten nur für den Upload oder für den Download vorgesehen sind.

Die Synchronisation sollte nur die Zeilen in eine entfernte Datenbank herunterladen, die für den Benutzer relevant sind. Sie können den Download nach der entfernten Datenbank, nach dem Benutzer oder nach anderen Kriterien aufteilen. Ein Vertriebsmitarbeiter in der Region A benötigt beispielsweise nur Daten über die Region A.

Sie sollten nur die geänderten Daten aktualisieren. In einer MobiLink-Anwendung basiert der Upload auf dem Transaktionslog. Daher werden Daten standardmäßig nur dann hochgeladen, wenn sie in der entfernten Datenbank geändert wurden. Um dasselbe Verfahren für den Download festzulegen, wählen Sie die Zeitstempel-basierte Synchronisation, sodass das System die Uhrzeit des Daten-Downloads aufzeichnet. Daten werden dann nur heruntergeladen, wenn sie nach diesem Zeitpunkt geändert wurden.

Sie können auch ein System mit einer prioritären Synchronisation implementieren. Zeitsensitive Daten werden in diesem Fall häufig aktualisiert, während weniger zeitkritische Daten nachts oder beim Einstecken des Geräts in eine Dockingstation aktualisiert werden. Sie können die Synchronisation mit Priorität implementieren, indem Sie verschiedene Publikationen erstellen, deren Ausführung zu unterschiedlichen Zeiten geplant ist.

Ihre Benutzer profitieren möglicherweise auch von einem Push-Synchronisationssystem, in dem Daten bei Bedarf an entfernte Geräte übergeben werden. Wenn ein Speditionsunternehmen beispielsweise von einer Verkehrsunterbrechung erfährt, kann es eine Aktualisierung für die LKW-Fahrer herunterladen, die

in dem betreffenden Gebiet unterwegs sind. Dies wird in MobiLink als serverinitiierte Synchronisation bezeichnet.

Upload-Konflikte behandeln

Angenommen Sie haben ein Lager. Jeder Mitarbeiter hat ein mobiles Gerät, über das er den Lagerbestand aktualisiert, wenn er Kisten hinzufügt oder entnimmt. Es wird bei 100 Kisten begonnen, sodass die entfernte Datenbank jedes Mitarbeiters den Wert 100 enthält, ebenso wie die konsolidierte Datenbank. David entnimmt 20 Kisten. Er aktualisiert seine Datenbank und führt eine Synchronisation durch. Jetzt enthält seine Datenbank ebenso wie die konsolidierte Datenbank den Wert 80. Nun entnimmt Susan zehn Kisten. Wenn Susan jedoch ihre Datenbank aktualisiert und synchronisiert, geht ihre Anwendung davon aus, dass die konsolidierte Datenbank den Wert von 100 Kisten enthält und nicht 80. Dadurch entsteht ein Upload-Konflikt.

In dieser Lageranwendung besteht die Lösung darin, eine Konfliktlösungslogik zu erstellen, die besagt, dass der richtige Wert der von David geänderte Wert abzüglich des von Susan eingegebenen Werts ist:

$$80 - (100 - 90) = 70$$

Diese Konfliktlösungslogik funktioniert für die Lagerverwaltung, sie ist jedoch nicht für alle Geschäftsanwendungen geeignet. Sie können in MobiLink folgende Fälle mit Konfliktlösungslogik abdecken:

- **Lagermodell** Es wird nur die Zeile für die richtige Anzahl von Einheiten aktualisiert.
- **Datum** Die letzte Aktualisierung gewinnt (basierend darauf, wann der Wert in der Datenbank geändert wurde und nicht wann der Wert synchronisiert wurde).
- **Person** Zum Beispiel, der Vorgesetzte gewinnt immer, oder der Eigentümer des Datensatzes gewinnt immer.
- **Benutzerdefiniert** Praktisch jede andere Geschäftslogik, die Sie implementieren müssen.

In einigen Fällen können Sie das System so planen, dass keine Upload-Konflikte auftreten. Wenn Daten auf den entfernten Datenbanken aufgeteilt werden, damit sie nicht überlappen, können Konflikte vermieden werden. Wenn es jedoch zu Konflikten kommen kann, sollten Sie eine anwendungsgesteuerte Lösung für die Erkennung und Lösung der Konflikte erstellen.

Eindeutige Primärschlüssel

Für den Upload von Daten, zur Erkennung von Konflikten und zur Synchronisation von gelöschten Zeilen in der konsolidierten Datenbank muss jede synchronisierte Tabelle im Datenbanksystem eindeutige Primärschlüssel haben. Jede Zeile muss einen Primärschlüssel haben, der nicht nur in der Datenbank selbst, sondern im gesamten Datenbanksystem eindeutig ist. Primärschlüssel dürfen nicht aktualisiert werden.

MobiLink bietet mehrere Möglichkeiten, eindeutige Primärschlüssel zu garantieren. Eine von ihnen ist, den Datentyp des Primärschlüssels auf eine GUID festzulegen. Ein GUID (Globally Unique Identifier) ist ein global eindeutiger Bezeichner und eine 16 Byte große hexadezimale Zahl. MobiLink stellt die Funktion NEWID zur Verfügung, die bewirkt, dass für eine neue Zeile automatisch ein GUID erstellt wird.

Eine weitere Lösung ist ein zusammengesetzter Schlüssel. In MobiLink hat jede entfernte Datenbank einen eindeutigen Wert, der als entfernte ID bezeichnet wird. Ihre Primärschlüssel können zum Beispiel aus der entfernten ID und einem normalen Primärschlüssel, z.B. einer Ordinalzahl, gebildet werden.

SQL Anywhere stellt auch eine Lösung mit globalen Autoinkrement-Werten zur Verfügung. Sie deklarieren hierbei eine Spalte als GLOBAL AUTOINCREMENT. Beim Hinzufügen einer Zeile wird der Primärschlüssel automatisch erstellt, indem der letzte Wert erhöht wird. Diese Lösung ist am besten geeignet, wenn die konsolidierte Datenbank eine SQL Anywhere-Datenbank ist.

Schließlich können Sie auch einen Pool von Primärschlüsseln erstellen, die auf entfernte Datenbanken verteilt werden.

Wie Sie das geeignete Primärschlüsselsystem auswählen, hängt wie zahlreiche andere Entscheidungen bei der Entwicklung einer Synchronisationslösung davon ab, welche Stufe der Kontrolle über die konsolidierte und die entfernte Datenbank Sie benötigen. Häufig müssen die entfernten Datenbanken in der Lage sein, ohne jegliche Administration zu arbeiten. Sie stellen möglicherweise auch fest, dass es schwierig ist, das Schema in der konsolidierten Datenbank zu ändern. Die Wahl des RDBMS-Systems für die konsolidierte Datenbank schränkt möglicherweise auch Ihre Möglichkeiten ein, da nicht alle RDBMS-Systeme alle Funktionen unterstützen.

Löschungen handhaben

Ein weiterer wichtiger Punkt in einem Synchronisationssystem ist die Behandlung von Zeilen, die aus der konsolidierten Datenbank gelöscht werden. Angenommen Sie löschen eine Zeile aus der konsolidierten Datenbank. Wenn ein anderer Mitarbeiter seine entfernte Datenbank das nächste Mal synchronisiert, wird die Löschung heruntergeladen und die Zeile damit aus seiner Datenbank gelöscht. Was geschieht jedoch mit der Löschung in der konsolidierten Datenbank? Die Zeile kann nicht gelöscht werden, da die Löschung auch noch in die entfernte Datenbank eines weiteren Benutzers heruntergeladen werden muss.

Es gibt zwei Möglichkeiten, Download-Löschungen zu handhaben. Sie können jeder Tabelle eine Statusspalte hinzufügen, die anzeigt, ob die Zeile gelöscht wurde. In diesem Fall wird die Zeile nie gelöscht, sondern nur für die Löschung markiert. Sie können die für die Löschung markierten Zeilen in regelmäßigen Abständen bereinigen, wenn Sie sicher sind, dass alle entfernten Datenbanken aktualisiert wurden. Alternativ dazu können Sie auch für jede Tabelle eine Schattentabelle erstellen. In der Schattentabelle werden die Primärschlüsselwerte von gelöschten Zeilen gespeichert. Wenn eine Zeile gelöscht wurde, füllt ein Trigger die Schattentabelle, und die Werte in der Schattentabelle legen fest, was in der entfernten Datenbank gelöscht werden soll.

Transaktionen

In einem synchronisierten Datenbanksystem sollten nur festgeschriebene Datenbanktransaktionen synchronisiert werden. Außerdem sollten alle festgeschriebenen Transaktionen, die zu synchronisierende Daten einbeziehen, synchronisiert werden. Andernfalls sollte ein Fehler generiert werden. Dies ist das Standardverhalten in MobiLink.

Sie müssen auch die Isolationsstufe der Verbindung zur konsolidierten Datenbank berücksichtigen. Sie müssen eine Isolationsstufe verwenden, die eine optimale Performance ebenso wie die Datenkonsistenz sicherstellt. Die Isolationsstufe 0 (READ UNCOMMITTED) eignet sich gewöhnlich nicht zur Synchronisation, da sie Dateninkonsistenzen verursachen kann.

Standardmäßig verwendet MobiLink die Isolationsstufe `SQL_TXN_READ_COMMITTED` für Uploads und, wenn möglich, die Snapshot-Isolation für Downloads (andernfalls wird `SQL_TXN_READ_COMMITTED` verwendet). Die Snapshot-Isolation vermeidet das Problem, dass Downloads blockiert werden, bis Transaktionen in der konsolidierten Datenbank geschlossen werden. Doch nicht alle RDBMS-Systeme unterstützen die Snapshot-Isolation.

Sommerzeit

Die jährliche Umstellung auf die Sommer- bzw. Winterzeit kann während der Stunde, in der die Zeit umgestellt wird, ein Problem für synchronisierte Datenbanken darstellen. Im Herbst wird die Zeit von 03:00 Uhr auf 02:00 Uhr zurückgestellt. Wenn Sie versuchen, zwischen 02:00 Uhr und 03:00 Uhr zu synchronisieren, ist der Zeitstempel der Synchronisation nicht eindeutig. Es kann zum Beispiel unklar sein, ob es sich um das erste 02:15 Uhr oder das zweite 02:15 Uhr handelt.

Um dieses Problem zu lösen, können Sie den Server während der Zeitumstellung eine Stunde lang herunterfahren oder Sie können den konsolidierten Datenbankserver auf die koordinierte universelle Zeit (Coordinated Universal Time, UTC) umstellen.

Siehe auch

- „Synchronisationsmethoden“ [[MobiLink - Serveradministration](#)]

Optionen für die MobiLink-Anwendungsentwicklung

MobiLink bietet eine Reihe von Möglichkeiten zur Entwicklung einer Anwendung. Sie können diese Methoden allein oder kombiniert verwenden.

- **Assistent zum Erstellen eines Synchronisationsmodells** Der Assistent leitet Sie durch die Entwicklung Ihrer Anwendung. Sie müssen mit einer zentralen Datenbank starten, die das Schema enthält, und können entfernte Datenbanken und die für die Synchronisation erforderlichen Skripten erstellen. Der Assistent kann auch Schattentabellen in Ihrer konsolidierten Datenbank erstellen, um beispielsweise Download-Löschungen zu verarbeiten. Wenn der Assistent abgeschlossen wurde, können Sie das Modell weiter anpassen. Der **Assistent für das Deployment eines Synchronisationsmodells** erstellt Datenbanken und Tabellen, aktualisiert die MobiLink-Systemtabellen und erstellt Skripten, die MobiLink-Dienstprogramme ausführen.

Nachdem Sie das Deployment eines MobiLink-Modells vorgenommen haben und weitere Anpassungen vornehmen möchten, können Sie dafür eine der nachstehenden Methoden verwenden.

- **Sybase Central** Mit dem MobiLink 16-Plug-In für Sybase Central können Sie alle Elemente Ihrer MobiLink-Anwendung aktualisieren.
- **Systemprozeduren** Wenn Sie eine zentrale Datenbank als konsolidierte Datenbank einrichten, werden Systemobjekte erstellt, die von der MobiLink-Synchronisation verwendet werden. Diese umfassen MobiLink-Systemtabellen, in denen der Serverbereich der MobiLink-Anwendung zum großen Teil gespeichert ist. Sie umfassen außerdem Systemprozeduren und Dienstprogramme, mit denen Sie beispielsweise MobiLink-Skripten in die MobiLink-Systemtabellen einfügen oder entfernte Benutzer registrieren können.

- **Direkte Bearbeitung von MobiLink-Systemtabellen** Fortgeschrittene Benutzer möchten Daten möglicherweise direkt den MobiLink-Systemtabellen hinzufügen, sie löschen und aktualisieren. Hierzu muss der Benutzer weitgehende Kenntnisse über die Funktionsweise von MobiLink besitzen.

Siehe auch

- „MobiLink-Plug-In für Sybase Central“ auf Seite 22
- „Synchronisationsmodellaufgaben“ auf Seite 33
- „Systemprozeduren des MobiLink-Servers“ [*MobiLink - Serveradministration*]
- „MobiLink-Dienstprogramme“ [*MobiLink - Serveradministration*]
- „Systemtabellen des MobiLink-Servers“ [*MobiLink - Serveradministration*]

Optionen zum Erstellen serverseitiger Synchronisationslogik

MobiLink-Synchronisationsskripten können in SQL, in Java (mit der MobiLink-Server-API für Java) oder in .NET (mit der MobiLink-Server-API für .NET) geschrieben werden.

Die SQL-Synchronisationslogik ist gewöhnlich am besten, wenn eine Synchronisation mit einer unterstützten konsolidierten Datenbank durchgeführt wird.

Wählen Sie Java und .NET, wenn Sie eine Synchronisation mit einer Datenbank durchführen, bei der es sich nicht um eine unterstützte konsolidierte Datenbank handelt. Verwenden Sie Java und .NET auch dann, wenn Ihr Design durch die SQL-Sprache bzw. die Möglichkeiten des Datenbank-Managementsystems eingeschränkt ist oder wenn Sie Portierbarkeit über verschiedene RDBMS-Typen wünschen.

Die Java- und .NET-Synchronisationslogik kann auf dieselbe Weise wie SQL-Logikfunktionen eingesetzt werden. Der MobiLink-Server kann beim Eintritt von MobiLink-Ereignissen Java- oder .NET-Methoden genauso wie SQL-Skripten aufrufen. Wenn Sie mit Java oder .NET arbeiten, können Sie die Ereignisse verwenden, um zusätzliche Prozessverarbeitungen vorzunehmen. Wenn Sie jedoch Skripten für Ereignisse verarbeiten, die Upload- oder Downloadzeilen direkt verarbeiten, muss Ihre Implementierung eine SQL-Zeichenfolge zurückgeben. Mit Ausnahme der beiden Ereignisse für die direkte Zeilenbehandlung kann auf Uploads und Downloads über die Java- oder .NET-Synchronisationslogik nicht direkt zugegriffen werden: MobiLink führt die Zeichenfolge aus, die von Java oder .NET als SQL zurückgegeben wurde.

Die direkte Zeilenbehandlung, die die Ereignisse `handle_UploadData` und `handle_DownloadData` zur Synchronisation einer Datenquelle verwendet, bearbeitet die Upload- und Download-Zeilen **direkt**.

Im Folgenden sind einige Szenarios beschrieben, in denen Java- oder .NET-Code sinnvoll eingesetzt werden kann.

- **Direkte Zeilenbehandlung** Mithilfe der Java- oder .NET-Synchronisationslogik können Sie MobiLink verwenden, um auf Daten aus Datenquellen zuzugreifen, die keine konsolidierten Datenbanken sind, beispielsweise Anwendungsserver, Webserver und Dateien.

- **Authentifizierung** Eine Benutzerauthentifizierungsprozedur kann in Java oder .NET geschrieben werden, sodass die MobiLink-Authentifizierung den Sicherheitsvorschriften Ihres Unternehmens entspricht.
- **Gespeicherte Prozeduren** Wenn Ihr RDBMS-System keine benutzerdefinierten gespeicherten Prozeduren verwenden kann, können Sie in Java oder .NET eine Methode erstellen.
- **Externe Aufrufe** Wenn Ihr Programm während eines Synchronisationsereignisses eine Verbindung zu einem externen Server herstellen soll, können Sie die Java- oder .NET-Synchronisationslogik einsetzen, damit Synchronisationsereignisse bestimmte Aktionen auslösen. Java- und .NET-Synchronisationslogik kann für mehrere Verbindungen gemeinsam genutzt werden.
- **Variable** Falls Ihre Datenbank nicht in der Lage ist, mit Variablen umzugehen, können Sie eine Variable in Java oder .NET einrichten, die während der gesamten Verbindung bzw. Synchronisation bestehen bleibt. (Alternativ können Sie mit SQL-Skripten benutzerdefinierte benannte Parameter verwenden, die mit allen konsolidierten Datenbanktypen kompatibel sind. Siehe „[Benutzerdefinierte benannte Parameter](#)“ [[MobiLink - Serveradministration](#)].)

MobiLink-Server-APIs

Auf die Java- und .NET-Synchronisationslogik können Sie über die MobiLink-Server-APIs zugreifen. Die MobiLink-Server-APIs sind Gruppen von Klassen und Schnittstellen für die MobiLink-Synchronisation.

Die MobiLink-Server-API für Java bietet Folgendes:

- Zugriff auf die vorhandene ODBC-Verbindung mit der konsolidierten Datenbank als JDBC-Verbindung.
- Zugriff auf andere Datenquellen über Schnittstellen wie JDBC, Webdienste und JNI.
- Die Fähigkeit, neue JDBC-Verbindungen mit der konsolidierten Datenbank einzurichten, um Datenbankänderungen außerhalb der aktuellen Synchronisationsverbindung vorzunehmen. Auf diese Weise können Sie beispielsweise eine Fehlerprotokollierung oder ein Auditing vornehmen, auch wenn die Synchronisationsverbindung ein Rollback durchführt.
- Im Bezug auf die Synchronisation mit der konsolidierten Datenbank die Fähigkeit, Java-Code vor dem Ausführen durch den MobiLink-Server schreiben und debuggen zu können. SQL-Entwicklungsumgebungen sind für viele Datenbank-Managementsysteme im Vergleich zu denen, die für Java-Anwendungen verfügbar sind, recht primitiv.
- SQL-Zeilenbehandlung und direkte Zeilenbehandlung
- Die volle Nutzung der Java-Sprache und ihrer großen Bandbreite bestehender Codes und Bibliotheken.

Siehe „[Referenz der MobiLink-Server-API für Java](#)“ [[MobiLink - Serveradministration](#)].

Die MobiLink-Server-API für .NET bietet Folgendes:

- Zugriff auf die vorhandene ODBC-Verbindung mit der konsolidierten Datenbank mithilfe von iAnywhere-Klassen, die ODBC von .NET aus aufrufen.

- Zugriff auf andere Datenquellen über Schnittstellen wie ADO.NET, Webdienste und OLE DB.
- In Bezug auf die Synchronisation mit der konsolidierten Datenbank die Fähigkeit, .NET-Code vor dem Ausführen durch den MobiLink-Server schreiben und debuggen zu können. SQL-Entwicklungsumgebungen sind für viele Datenbank-Managementsysteme im Vergleich zu denen, die für .NET-Anwendungen verfügbar sind, recht primitiv.
- SQL-Zeilenbehandlung und direkte Zeilenbehandlung
- Code, der innerhalb der .NET-CLR-Sprache (Common Language Runtime) abläuft und Zugriff auf alle .NET-Bibliotheken einschließlich SQL-Zeilenbehandlung und direkte Zeilenbehandlung ermöglicht

Siehe „Referenz zur MobiLink-Server-API für .NET“ [[MobiLink - Serveradministration](#)].

Siehe auch

- „Schreiben von Synchronisationsskripten“ [[MobiLink - Serveradministration](#)]
- „Synchronisationsmethoden“ [[MobiLink - Serveradministration](#)]
- „Schreiben eines Synchronisationsskripts in Java“ [[MobiLink - Serveradministration](#)]
- „Synchronisationsskripten in .NET“ [[MobiLink - Serveradministration](#)]
- „Direkte Zeilenbehandlung“ [[MobiLink - Serveradministration](#)]

Der Synchronisationsprozess

Bei der **Synchronisation** handelt es sich um den Prozess des bidirektionalen Datenaustausches zwischen MobiLink-Clients und einer zentralen Datenquelle. Während dieses Vorgangs muss der Client eine Verbindung mit dem MobiLink-Server herstellen und aufrechterhalten. Gelingt dies, stimmen die entfernte und die konsolidierte Datenbank weiterhin überein.

Der Synchronisationsprozess wird normalerweise vom Client initiiert. Zunächst wird eine Verbindung zum MobiLink-Server hergestellt.

Upload und Download

Zum Upload von Zeilen wird auf der Clientseite von MobiLink ein **Upload** vorbereitet und gesendet, in dem eine Liste der Zeilen enthalten ist, die seit der letzten Synchronisation in der entfernten Datenbank geändert, eingefügt oder gelöscht wurden. Ebenso wird zum Download von Zeilen auf dem MobiLink-Server ein **Download** mit den Datensätzen vorbereitet und gesendet, die auf der Serverseite geändert, eingefügt bzw. gelöscht wurden.

- **Upload** Standardmäßig protokolliert der MobiLink-Client automatisch die Zeilen, die seit der letzten erfolgreichen Synchronisation in der entfernten Datenbank eingefügt, aktualisiert oder gelöscht wurden. Wenn die Verbindung hergestellt ist, überträgt der MobiLink-Client eine Liste all dieser Änderungen an den MobiLink-Server.

Der Upload-Datenstrom besteht aus einer Reihe neuer und alter Zeilenwerte für in der entfernten Datenbank geänderte Zeilen. (Aktualisierungen enthalten neue und alte Zeilenwerte, Löschungen alte Werte und Einfügungen neue Werte.) Wenn eine Zeile aktualisiert oder gelöscht wurde, sind die alten Werte jene, die unmittelbar nach der letzten erfolgreichen Synchronisation vorhanden waren. Wenn

eine Zeile eingefügt oder aktualisiert wurde, sind die neuen Werte die aktuellen Zeilenwerte. Es werden keine Zwischenwerte gesendet, auch wenn die Zeile mehrmals geändert wurde, bevor sie den aktuellen Status erreicht hat.

Der MobiLink-Server empfängt den Upload und führt die definierten Upload-Skripten aus. Standardmäßig übernimmt er alle Änderungen in nur einer Transaktion. Anschließend schreibt der MobiLink-Server die Transaktion fest.

- **Download** Der MobiLink-Server erstellt unter Verwendung der von Ihnen erstellten Synchronisationslogik eine Liste der auf dem MobiLink-Client einzufügenden, zu aktualisierenden oder zu löschenden Zeilen. Er überträgt diese Zeilen an den MobiLink-Client. Zum Zusammenstellen dieser Liste öffnet der MobiLink-Server eine neue Transaktion in der konsolidierten Datenbank.

Der MobiLink-Client empfängt den Download. Er betrachtet die Ankunft dieses Downloads als Bestätigung, dass die konsolidierte Datenbank alle heraufgeladenen Änderungen erfolgreich übernommen hat. Als Nächstes wird dafür gesorgt, dass diese Änderungen nicht nochmals an die konsolidierte Datenbank gesendet werden.

Anschließend verarbeitet der MobiLink-Client den Download und löscht alte Zeilen, fügt neue Zeilen ein und aktualisiert geänderte Zeilen. Alle diese Änderungen werden in der entfernten Datenbank in einer einzigen Transaktion durchgeführt. Danach schreibt er die Transaktion fest.

Im Verlauf einer MobiLink-Synchronisation werden einige ganz bestimmte Informationen ausgetauscht. Der Client baut den gesamten Upload auf und überträgt ihn. Als Antwort baut der MobiLink-Server den gesamten Download auf und überträgt ihn. Es ist wichtig, die Ausführlichkeitsstufe des Protokolls einzuschränken, wenn die Verbindung langsam ist und eine längere Verarbeitungszeit erfordert, wie z.B. bei der Verwendung von Telefonleitungen oder öffentlichen Mobilfunknetzwerken.

Hinweis

MobiLink verwendet die ODBC-Isolationsstufe `SQL_TXN_READ_COMMITTED` als Standardisolationsstufe für die konsolidierte Datenbank. Wenn das als konsolidierte Datenbank verwendete RDBMS die Snapshot-Isolation unterstützt und das Erstellen von Snapshots für die Datenbank aktiviert ist, verwendet MobiLink standardmäßig die Snapshot-Isolation für Downloads. Siehe „MobiLink-Isolationsstufen“ [[MobiLink - Serveradministration](#)].

Siehe auch

- „Überblick über MobiLink-Ereignisse“ [[MobiLink - Serveradministration](#)]
- „Ereignisse beim Upload“ [[MobiLink - Serveradministration](#)]
- „Ereignisse beim Download“ [[MobiLink - Serveradministration](#)]

MobiLink-Ereignisse

Wenn der MobiLink-Client eine Synchronisation initiiert, laufen mehrere Synchronisationsereignisse ab. Beim Eintreten eines Synchronisationsereignisses sucht MobiLink ein Skript, das mit dem Ereignis übereinstimmt. Das Skript enthält Anweisungen für die auszuführenden Aufgaben. Wenn Sie ein Skript für das Ereignis erstellt und in eine MobiLink-Systemtabelle eingefügt haben, wird dieses aufgerufen.

MobiLink-Skripten

Sobald ein Ereignis eintritt, führt der MobiLink-Server das zugeordnete Skript aus, falls Sie ein solches erstellt haben. Wenn kein Skript vorhanden ist, tritt das nächste Ereignis in der Folge ein.

Hinweis

Wenn Sie den **Assistenten zum Erstellen eines Synchronisationsmodells** verwenden, um Ihre MobiLink-Anwendung zu erstellen, werden alle notwendigen MobiLink-Skripten für Sie erstellt. Sie können die Standardskripten jedoch anpassen und auch neue Skripten erstellen.

Im Folgenden finden Sie einige typische Uploadskripten für Tabellen. Das erste Ereignis, upload_insert, löst die Ausführung des Skripts upload_insert aus, mit dem alle Änderungen in den Spalten emp_id und emp_name in die Tabelle emp eingefügt werden. Die Skripten upload_delete und upload_update führen Funktionen zum Löschen und Aktualisieren in der Tabelle emp aus.

Ereignis	Inhalt des Beispielskripts
upload_insert	<code>INSERT INTO emp (emp_id,emp_name) VALUES ({ml r.emp_id}, {ml r.emp_name})</code>
upload_delete	<code>DELETE FROM emp WHERE emp_id = {ml r.emp_id}</code>
upload_update	<code>UPDATE emp SET emp_name = {ml r.emp_name} WHERE emp_id = {ml r.emp_id}</code>

Das Download-Skript benutzt einen Cursor. Das folgende Beispiel ist ein download_cursor-Skript:

```
SELECT order_id, cust_id  
FROM ULOrder  
WHERE last_modified >= {ml s.last_table_download}  
AND emp_name = {ml r.emp_id}
```

Weitere Hinweise zu Ereignissen und Skripten finden Sie unter:

- „Schreiben von Synchronisationsskripten“ [[MobiLink - Serveradministration](#)]
- „Synchronisationsereignisse“ [[MobiLink - Serveradministration](#)]

Skripten können in SQL, Java oder .NET geschrieben werden

Sie können Skripten mithilfe des nativen SQL-Dialekts Ihrer konsolidierten Datenbank schreiben oder die Java- oder .NET-Synchronisationslogik verwenden. Mit der Java- oder .NET-Synchronisationslogik kann der Benutzer Code schreiben, der vom MobiLink-Server aufgerufen wird, um eine Verbindung mit einer Datenbank herzustellen, Variablen zu verarbeiten, heraufgeladene Zeilenvorgänge direkt zu bearbeiten oder Zeilenvorgänge zum Download hinzuzufügen. Es gibt eine MobiLink-Server-API für Java und eine MobiLink-Server-API für .NET, die über Klassen und Methoden verfügen, die die Anforderungen der Synchronisation erfüllen.

Siehe „Optionen zum Erstellen serverseitiger Synchronisationslogik“ auf Seite 12.

Weitere Hinweise zur RDBMS-abhängigen Skripterstellung finden Sie unter „[MobiLink - konsolidierte Datenbanken](#)“ [[MobiLink - Serveradministration](#)].

Skripten speichern

SQL-Skripten werden in MobiLink-Systemtabellen in der konsolidierten Datenbank gespeichert. Bei Skripten, die mit MobiLink-Server-APIs geschrieben wurden, wird der vollständige qualifizierte Methodenname als Skript gespeichert. Sie können einer konsolidierten Datenbank Skripten auf verschiedene Arten hinzufügen:

- Wenn Sie den **Assistenten zum Erstellen eines Synchronisationsmodells** verwenden, werden Skripten in den MobiLink-Systemtabellen gespeichert, wenn Sie das Deployment für Ihr Projekt durchführen.
- Sie können Skripten manuell zu den Systemtabellen hinzufügen, indem Sie gespeicherte Prozeduren verwenden, die installiert werden, wenn Sie eine konsolidierte Datenbank einrichten.
- Sie können Skripten mithilfe von Sybase Central manuell zu den Systemtabellen hinzufügen.

Siehe „[Hinzufügungen und Löschungen von Skripten](#)“ [*MobiLink - Serveradministration*].

Transaktionen im Synchronisationsprozess

Der MobiLink-Server übernimmt die Änderungen, die von den einzelnen MobiLink-Clients in einer Transaktion in die konsolidierte Datenbank hochgeladen wurden. Die Änderungen werden festgeschrieben, nachdem neue Zeilen eingefügt, alte Zeilen gelöscht, Aktualisierungen durchgeführt und Konflikte gelöst wurden.

Vorsicht

Ihre SQL-Synchronisationsskripten bzw. die Prozeduren oder Skripten, die von Ihren SQL-Synchronisationsskripten aufgerufen werden, dürfen keine impliziten oder expliziten COMMIT- oder ROLLBACK-Anweisungen enthalten. COMMIT- und ROLLBACK-Anweisungen innerhalb eines SQL-Skripts ändern den transaktionsbasierten Ablauf der Synchronisationsschritte. Falls Sie diese Anweisungen benutzen, kann MobiLink die Integrität der Daten bei einer Systemstörung nicht garantieren.

Heruntergeladene Daten protokollieren

MobiLink verwendet den letzten Download-Zeitstempel, der in der entfernten Datenbank gespeichert wurde, um die Erstellung von Downloads zu vereinfachen.

Die vorrangige Rolle der Downloadtransaktion besteht darin, Zeilen in der konsolidierten Datenbank auszuwählen. Wenn der Download fehlschlägt, lädt die entfernte Datenbank denselben letzten Download-Zeitstempel erneut hoch, und es gehen keine Daten verloren.

Transaktionen starten und beenden

Der MobiLink-Client verarbeitet die Downloaddaten in einer Transaktion. Zeilen werden eingefügt, aktualisiert und gelöscht, um die entfernte Datenbank mit der konsolidierten Datenbank in Übereinstimmung zu bringen.

Der MobiLink-Server verwendet zwei weitere Transaktionen. Eine zu Beginn der Synchronisation und die andere an deren Ende. Mit diesen Transaktionen können Sie Daten zu den einzelnen

Synchronisationen und deren Dauer aufzeichnen. Auf diese Weise können Sie Statistiken über versuchte und erfolgreiche Synchronisationen und über die Dauer der Synchronisationen protokollieren. Da Daten an verschiedenen Stellen im Prozess festgeschrieben werden, können Sie mit diesen Transaktionen auch Daten festschreiben, die bei der Analyse von Synchronisationsfehlschlägen hilfreich sind.

Siehe auch

- „Zeitpunkt des letzten Downloads in Skripten“ [[MobiLink - Serveradministration](#)]
- „Überblick über MobiLink-Ereignisse“ [[MobiLink - Serveradministration](#)]
- „Ereignisse beim Upload“ [[MobiLink - Serveradministration](#)]
- „Ereignisse beim Download“ [[MobiLink - Serveradministration](#)]

Verfahren bei Synchronisationsfehlern

Die MobiLink-Synchronisation ist fehlertolerant. Wenn z.B. eine Kommunikationsverbindung während einer Synchronisation abbricht, bleiben sowohl die entfernte Datenbank als auch die konsolidierte Datenbank in einem konsistenten Zustand.

Auf dem Client wird ein Fehlschlag durch einen Rückgabecode angegeben.

Je nach Ursache werden Synchronisationsfehler unterschiedlich bearbeitet. In den folgenden Fällen wird auf unterschiedliche Weise verfahren:

- **Fehlschlag beim Upload** Wenn der Fehlschlag auftritt, während der Upload erstellt oder angewendet wird, behält die entfernte Datenbank genau den Zustand bei, den sie zu Beginn der Synchronisation hatte. Auf dem Server werden alle bereits ausgeführten Teile des Upload wieder zurückgesetzt.
- **Fehlschlag zwischen Upload und Download** Wenn der Fehlschlag eintritt, nachdem der Upload abgeschlossen wurde und bevor der MobiLink-Client den Download erhalten hat, kann der Client nicht sicher sein, ob die übertragenen Änderungen erfolgreich in die konsolidierte Datenbank übernommen wurden. Der Upload kann vollständig ausgeführt und festgeschrieben worden sein. Der Fehlschlag könnte aber auch eingetreten sein, bevor der Server den gesamten Upload übernommen hat. Der MobiLink-Server setzt unvollständige Transaktionen in der konsolidierten Datenbank automatisch zurück.

Der MobiLink-Client bewahrt eine Aufzeichnung aller Upload-Änderungen auf. Bei der nächsten Synchronisation des MobiLink-Clients fragt er daher vor dem Erstellen des neuen Upload zunächst den Status des letzten Uploads ab. Wurde der letzte Upload nicht festgeschrieben, enthält der neue Upload auch alle Änderungen des letzten Uploads.

- **Fehlschlag beim Download** Wenn auf dem entfernten Gerät während des Downloads ein Fehler auftritt, werden alle bereits ausgeführten Teile des Downloads zurückgesetzt, und die entfernte Datenbank behält den Zustand bei, den sie vor dem Download hatte.

Wenn Sie eine nicht blockierende Downloadbestätigung verwenden, wurde die Downloadtransaktion bereits festgeschrieben, doch weder das Skript `nonblocking_download_ack` noch das Skript `publication_nonblocking_download_ack` wird aufgerufen.

Wenn Sie keine Downloadbestätigung verwenden, gibt es bei einem Downloadfehler keine serverseitigen Auswirkungen.

Neu startbare Download-Funktion

MobiLink verfügt über Funktionen, die bei der Wiederherstellung nach einem Download-Fehler Unterstützung bieten und eine Übertragungswiederholung des gesamten Downloads verhindern. Für diese Funktion gibt es unterschiedliche Implementierungen für entfernte SQL Anywhere- und UltraLite-Datenbanken. Siehe „[mlsrv16-Option -ds](#)“ [[MobiLink - Serveradministration](#)].

Wenn ein Fehler auftritt, gehen keine Daten verloren. Der MobiLink-Server und der MobiLink-Client regeln das für Sie. Weder Sie noch der Benutzer brauchen sich um die Aufrechterhaltung eines konsistenten Datenbestands in der Anwendung zu kümmern.

Verarbeitung des Uploads

Wenn der MobiLink-Server einen Upload von einem MobiLink-Client erhält, wird der gesamte Upload im Speicher gehalten, bis die Synchronisation abgeschlossen ist. Diese Vorgehensweise hat folgende Gründe:

- **Zeilen für den Download filtern** Die häufigste Methode zur Bestimmung der Zeilen für den Download besteht darin, die Zeilen herunterzuladen, die seit dem letzten Download verändert wurden. Beim Synchronisieren hat der Upload Vorrang vor dem Download. Bei den Zeilen, die während des Uploads eingefügt oder geändert wurden, handelt es sich um die Zeilen, die seit dem vorherigen Download geändert wurden.

Es wäre schwierig, ein `download_cursor`-Skript zu schreiben, das die im Rahmen des Uploads übertragenen Zeilen aus dem Download ausschließt. Aus diesem Grund filtert der MobiLink-Server automatisch diese Zeilen aus dem Download aus.

- **Einfügungen und Aktualisierungen verarbeiten** Standardmäßig werden Upload-Tabellen in der konsolidierten Datenbank in einer Reihenfolge angewendet, die Verletzungen der referenziellen Integrität vermeidet. Die Tabellen im Upload werden anhand ihrer Fremdschlüsselbeziehungen sortiert. Wenn z.B. in Tabelle A und in Tabelle C jeweils Fremdschlüssel enthalten sind, die eine Primärschlüsselspalte in Tabelle B referenzieren, werden Einfügungen und Aktualisierungen für Tabelle B zuerst heraufgeladen.
- **Löschungen nach Einfügungen und Aktualisierungen verarbeiten** Löschungen werden in der konsolidierten Datenbank ausgeführt, nachdem alle Einfügungen und Aktualisierungen vorgenommen wurden. Wenn Löschungen ausgeführt werden, werden Tabellen in der umgekehrten Reihenfolge des Uploads verarbeitet. Wenn eine Zeile, die gelöscht wird, eine Zeile in einer anderen Tabelle referenziert, die gelöscht wird, wird durch diese Reihenfolge sichergestellt, dass die referenzierende Zeile gelöscht wird, bevor die referenzierte Zeile gelöscht wird.
- **Deadlock** Wird ein Upload in die konsolidierte Datenbank ausgeführt, kann es aufgrund von Parallelität mit anderen Transaktionen zu Deadlock-Situationen kommen. Bei diesen Transaktionen kann es sich um Upload-Transaktionen anderer Datenbankverbindungen des MobiLink-Servers bzw. um Transaktionen anderer Anwendungen handeln, die auf die konsolidierte Datenbank zugreifen.

Wenn bei einer Upload-Transaktion ein Deadlock eintritt, dann wird diese zurückgesetzt, und der MobiLink-Server beginnt automatisch, den Upload erneut von Beginn an auszuführen.

Hinweis

Performancetipp Sie müssen Ihre Synchronisationsskripten so schreiben, dass Konfliktsituationen so weit wie möglich vermieden werden. Konfliktsituationen haben erheblichen Einfluss auf die Performance, wenn mehrere Benutzer gleichzeitig synchronisieren.

Referenzielle Integrität und Synchronisation

Mit Ausnahme von UltraLite Java Edition erzwingen alle MobiLink-Clients referenzielle Integrität, wenn sie den Download in die entfernte Datenbank integrieren.

Der MobiLink-Client löscht standardmäßig automatisch alle Zeilen, die die referenzielle Integrität verletzen, damit die Downloadtransaktion nicht fehlschlägt.

Diese Funktion bietet folgende Vorteile:

- Schutz vor Fehlern in den Synchronisationsskripten. Aufgrund der Flexibilität der Skripten ist es möglich, dass ungewollt ein Download von Zeilen erfolgt, die die Integrität der Datenbank verletzen würden. Der MobiLink-Client bewahrt automatisch die referenzielle Integrität, ohne dass ein Eingriff von außen erforderlich wird.
- Sie können diesen Mechanismus zur referenziellen Integrität einsetzen, um Daten aus einer entfernten Datenbank effizient zu löschen. Wenn Sie einen Löschvorgang für einen übergeordneten Datensatz senden, entfernt der MobiLink-Client automatisch alle untergeordneten Datensätze. Damit wird das von MobiLink an die entfernte Datenbank zu übertragende Datenaufkommen erheblich reduziert.

MobiLink-Clients geben wie folgt Benachrichtigungen aus, wenn sie explizit Zeilen löschen müssen, um die referenzielle Integrität zu erhalten:

- Für SQL Server Anywhere-Clients schreibt dbmlsync einen Eintrag in das Log. Außerdem können Sie dbmlsync-Ereignis-Hooks verwenden. Siehe:
 - „sp_hook_dbmlsync_download_ri_violation“ [*MobiLink - Clientadministration*]
 - „sp_hook_dbmlsync_download_log_ri_violation“ [*MobiLink - Clientadministration*]
- Für UltraLite-Clients wird die Warnung `SQL_ROW_DELETED_TO_MAINTAIN_REFERENTIAL_INTEGRITY` ausgegeben. Diese Warnung verwendet den Tabellennamen als Parameter. Um die referenzielle Integrität aufrechtzuerhalten, wird die Warnung für jede gelöschte Zeile ausgegeben. Ihre Anwendung kann die Warnung ignorieren, wenn mit der Synchronisation fortgefahren werden soll. Falls Sie Warnmeldungen explizit behandeln wollen, können Sie sie mit der Fehler-Callback-Funktion abfangen und beispielsweise die Anzahl der gelöschten Zeilen ermitteln.

Wenn die Synchronisation bei Ausgabe der Warnung abgebrochen werden soll, müssen Sie eine Synchronisationsbeobachtungsfunktion implementieren und dann der Beobachtungsfunktion (z.B.

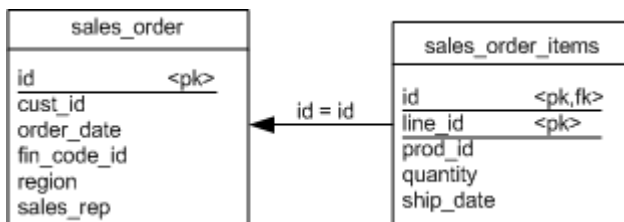
durch eine globale Variable) von der Fehler-Callback-Funktion aus ein Signal senden. In diesem Fall schlägt die Synchronisation beim nächsten Aufruf der Beobachtungsfunktion fehl.

Referenzielle Integrität wird am Ende der Transaktion geprüft

Der MobiLink-Client integriert die Änderungen aus dem Download in einer einzigen Transaktion. Um eine größere Flexibilität zu erzielen, findet die Prüfung der referenziellen Integrität am Ende dieser Transaktion statt. Da die Prüfung mit einer gewissen Verzögerung stattfindet, befindet sich die Datenbank zeitweise unter Umständen in einem Zustand, in dem die referenzielle Integrität nicht gewährleistet ist. Der Grund hierfür ist, dass Zeilen, die die referenzielle Integrität verletzen, vor dem Festschreiben des Downloads automatisch gelöscht werden.

Beispiel

Angenommen, eine UltraLite-Verkaufsanwendung enthält die folgenden beiden Tabellen: Eine Tabelle enthält Aufträge. Eine andere Tabelle enthält Artikel, die in den einzelnen Aufträgen verkauft wurden. Sie haben folgende Beziehung:



Wenn Sie `download_delete_cursor` für die Tabelle `sales_order` verwenden, um einen Auftrag zu löschen, dann löscht der standardmäßige Mechanismus der referenziellen Integrität automatisch alle Zeilen in `sales_order_items`, die auf die gelöschten Bestellungen verweisen, um die referenzielle Integrität aufrechtzuerhalten.

Dieser Mechanismus bietet folgende Vorteile:

- Sie benötigen kein `sales_order_items`-Skript, da die betreffenden Zeilen dieser Tabelle automatisch gelöscht werden.
- Die Effizienz der Synchronisation wird verbessert. Sie brauchen den Download der aus der Tabelle `sales_order_items` zu löschenden Zeilen nicht vorzunehmen. Wenn die einzelnen Bestellungen mehrere Artikel enthalten, wird dadurch die Performance gesteigert, da der Download nun kleiner ist. Diese Technik ist dann besonders hilfreich, wenn relativ langsame Verbindungen verwendet werden.

Das Standardverhalten ändern

Bei SQL Anywhere-Clients können Sie den Client-Ereignis-Hook `sp_hook_dbmsync_download_ri_violation` verwenden, um auf die Verletzung der referenziellen Integrität zu reagieren. Dbmsync schreibt außerdem einen Eintrag in sein Log.

Siehe:

- „`sp_hook_dbmsync_download_log_ri_violation`“ [*MobiLink - Clientadministration*]
- „`sp_hook_dbmsync_download_ri_violation`“ [*MobiLink - Clientadministration*]

Sicherheit

Es gibt mehrere Aspekte bei der Sicherung von Daten in einem großen verteilten System, wie einer MobiLink-Installation:

- **Daten in der konsolidierten Datenbank schützen** Daten in der konsolidierten Datenbank können mit dem Datenbanksystem zur Benutzerauthentifizierung und mit anderen Sicherheitskomponenten geschützt werden.

Weitere Hinweise finden Sie in Ihrer Datenbankdokumentation. Wenn Sie eine konsolidierte SQL Anywhere-Datenbank verwenden, finden Sie weitere Hinweise unter „[Datensicherheit](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

- **Daten in der entfernten Datenbank schützen** Wenn Sie mit entfernten SQL Anywhere-Datenbanken arbeiten, können die Daten durch SQL Anywhere-Sicherheitskomponenten geschützt werden. Diese Funktionen sind standardmäßig so ausgelegt, dass ein nicht autorisierter Zugriff über Client/Server-Verbindungen verhindert wird. Sie bieten jedoch keinen sicheren Schutz vor einem ernsthaften Versuch, Daten direkt aus der Datenbankdatei zu extrahieren.

Dateien auf dem Client sind durch die Sicherheitskomponenten auf dem Client-Betriebssystem geschützt.

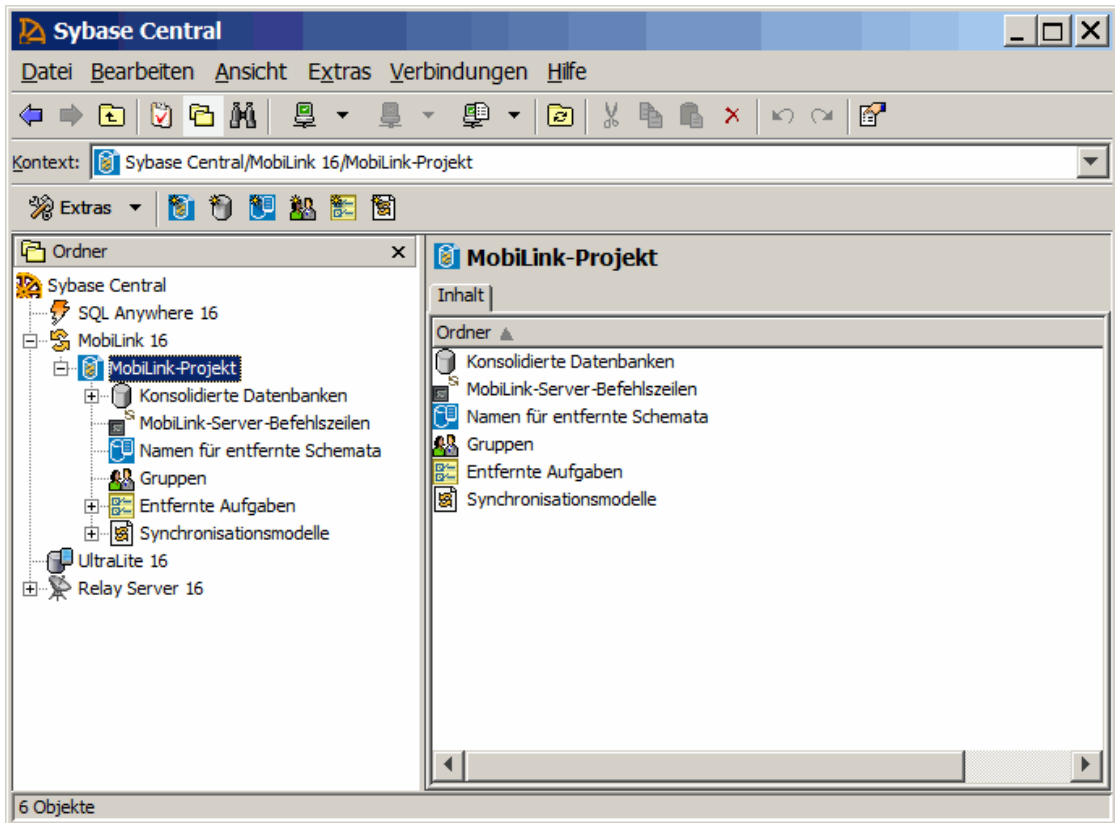
Wenn Sie eine entfernte SQL Anywhere-Datenbank verwenden, finden Sie weitere Hinweise unter „[Datensicherheit](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

Wenn Sie eine UltraLite-Datenbank verwenden, finden Sie weitere Hinweise unter „[Datenbanksicherheit](#)“ [[UltraLite - Datenbankverwaltung](#)].

- **Daten bei der Synchronisation schützen** Die Kommunikation zwischen MobiLink-Clients und MobiLink-Servern kann durch die Sicherheitskomponenten der MobiLink-Transportschicht geschützt werden. Siehe „[Transportschichtsicherheit](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].
- **Synchronisationssystem vor nicht autorisierten Benutzern schützen** Die MobiLink-Synchronisation kann durch ein Benutzerauthentifizierungssystem mit Kennwörtern geschützt werden. Dieses Verfahren verhindert, dass nicht autorisierte Benutzer Daten synchronisieren. Siehe „[MobiLink-Benutzer](#)“ [[MobiLink - Clientadministration](#)].

MobiLink-Plug-In für Sybase Central

Das MobiLink-Plug-In für Sybase Central wurde in Version 12 neu gestaltet. In vorherigen Versionen hatte das Plug-In zwei Modi: Modellmodus und Admin-Modus. Die MobiLink-Funktionen waren zwischen diesen beiden Modi aufgeteilt, daher mussten Sie darauf achten, in welchem Modus Sie gerade arbeiten. In Version 12 und später sind diese Modi nicht mehr vorhanden, wie weiter unten gezeigt wird.



Die Funktionen der obersten Ebene im **Ordner**-Fensterausschnitt des MobiLink-Plug-Ins sind:

- Mit MobiLink-Projekten arbeiten. Siehe „[Erstellen eines MobiLink-Projekts](#)“ auf Seite 24.
- Mit konsolidierten Datenbanken arbeiten. Siehe „[Konsolidierte Datenbank hinzufügen](#)“ auf Seite 25.
- Mit MobiLink-Server-Befehlszeilen arbeiten Siehe „[MobiLink-Server-Befehlszeilen erstellen](#)“ auf Seite 28.
- Mit entfernten Schemanamen arbeiten. Siehe „[MobiLink-Agenten in Sybase Central](#)“ [*MobiLink - Serveradministration*].
- Mit Gruppen arbeiten. Siehe „[MobiLink-Agenten in Sybase Central](#)“ [*MobiLink - Serveradministration*].
- Mit entfernten Aufgaben arbeiten. Siehe „[Entfernte Aufgaben](#)“ [*MobiLink - Serveradministration*].
- Mit Synchronisationsmodellen arbeiten. Siehe „[Synchronisationsmodelle](#)“ auf Seite 30.

Name des entfernten Schemas, Gruppe und entfernte Aufgaben sind alle Teil der zentralen Verwaltung von Funktionen auf entfernten Datenbanken. Siehe „[Zentrale Administration von entfernten Datenbanken](#)“ [*MobiLink - Serveradministration*].

Um mit MobiLink in Sybase Central zu arbeiten, müssen Sie zuerst ein MobiLink-Projekt definieren.

Ein **MobiLink-Projekt** ist eine Struktur zur Organisation der Synchronisationsmodelle, konsolidierten Datenbanken und entfernten Aufgaben, die mit einer mobilen Anwendung in Zusammenhang stehen.

Ein MobiLink-Projekt ist eine benannte Sammlung folgender Elemente:

- Eine Liste von Synchronisationsmodellen
- Eine Liste der angelegten, aber nicht bereitgestellten entfernten Aufgaben
- Eine Liste der Verbindungen mit konsolidierten Datenbanken
- Eine Liste mit benutzerdefinierten Gruppen von MobiLink-Benutzern

Erstellen eines MobiLink-Projekts

Erstellen Sie ein MobiLink-Projekt, bevor Sie mit MobiLink in Sybase Central arbeiten.

Voraussetzungen

Es gibt keine Voraussetzungen für das Ausführen dieser Aufgabe.

Kontext und Bemerkungen

Ein Beispiel für ein MobiLink-Projekt finden Sie unter `%SQLANYSAMPI6%\MobiLink\CustDB\project.mlp`.

Aufgabe

1. Klicken Sie in Sybase Central auf **Extras » MobiLink 16 » Neues Projekt**.
2. Geben Sie in das Feld **Wie lautet der Name des neuen Projekts?** einen Namen für das Projekt ein.
3. Geben Sie im Feld **Wo soll das neue Projekt gespeichert werden?** den Speicherort des Projektordners ein oder klicken Sie auf **Durchsuchen**, um den Ordner für Ihre Projektdatei auszuwählen.
4. Klicken Sie auf **Weiter**.
5. Führen Sie auf der Seite **Konsolidierte Datenbank angeben** folgende Aufgaben durch:
 - a. Geben Sie im Feld **Anzeigename der Datenbank** den Anzeigenamen ein, den Sie für die konsolidierte Datenbank verwenden möchten. Dies ist der Name, der in der Liste der konsolidierten Datenbanken in Ihrem Projekt enthalten ist.
 - b. Geben Sie im Feld **Verbindungszeichenfolge** die Datenbankverbindungsparameter ein, die verwendet werden sollen, um eine Verbindung mit der konsolidierten Datenbank herzustellen, oder klicken Sie auf **Bearbeiten**, um ein Fenster für die Verbindung mit einer ODBC-Datenquelle zu öffnen.

- c. Wählen Sie **Kennwort speichern**, um das Kennwort zu speichern, das Sie für die Verbindung zur Datenbank benutzt haben.
Das Aktivieren dieser Option führt dazu, dass das Kennwort in verschleierter Form in der Projektdatei gespeichert wird.
 - d. Klicken Sie auf **Weiter**.
6. Auf der Seite **Neues entferntes Datenbankschema** wählen Sie die konsolidierten Tabellen und Spalten, die in der entfernten Datenbank erscheinen sollen. Klicken Sie auf **Weiter**.
 7. Wählen Sie **Dem Projekt einen Namen für ein entferntes Schema hinzufügen**, um eine Gruppe von entfernten Datenbanken zu identifizieren, die dasselbe Schema haben oder fügen Sie später eins hinzu. Namen für entfernte Schemata sind nur sinnvoll, wenn Sie entfernte Aufgaben erstellen.

Wenn Sie einen entfernten Schemanamen hinzufügen, geben Sie Folgendes ein:

- **Wie lautet der Name des neuen entfernten Schemas** Geben Sie den Namen ein, den Sie verwenden möchten, um die Gruppe der entfernten Datenbanken zu identifizieren, die dasselbe Schema verwenden. Es empfiehlt sich, eine Versionsnummer hinzuzufügen.

Klicken Sie auf **Weiter**.

8. Auf der Seite **Geben Sie einen Typ für die entfernte Datenbank ein** wählen Sie **SQL Anywhere** oder **UltraLite** als Typ der entfernten Datenbank, die Sie verwenden wollen. Sie können diese Einstellung später auf der Seite **Eigenschaften** für das Projekt ändern.
9. Klicken Sie auf **Fertig stellen**, um das neue Projekt zu speichern.
10. Wenn Sie aufgefordert werden, die MobiLink-Systemkonfiguration zu installieren, klicken Sie auf **Ja** und dann auf **OK**.

Ergebnisse

Das MobiLink-Projekt wird erstellt, eine Verbindung mit der angegebenen Datenbank wird hergestellt und ein Synchronisationsmodell wird erstellt.

Nächste Schritte

Sie können das Synchronisationsmodell nun bearbeiten oder sein Deployment durchführen oder mit anderen Objekten im MobiLink-Projekt arbeiten, wie zum Beispiel entfernten Aufgaben.

Siehe auch

- „Konzepte zur zentralen Administration“ [[MobiLink - Serveradministration](#)]

Konsolidierte Datenbank hinzufügen

Fügen Sie einem MobiLink-Projekt in Sybase Central eine oder mehrere konsolidierte Datenbanken hinzu.

Voraussetzungen

Ein MobiLink-Projekt muss definiert sein.

Kontext und Bemerkungen

Einer entfernten Aufgabe muss mindestens eine konsolidierte Datenbank zugewiesen sein, bevor ihr Deployment vorgenommen werden kann.

Aufgabe

1. Wählen Sie das MobiLink-Projekt aus.
2. Doppelklicken Sie auf den Projektnamen und klicken Sie auf **Datei » Neu » Konsolidierte Datenbank**.
3. Geben Sie die erforderlichen Datenbankverbindungsparameter ein und klicken Sie auf **Weiter**.
4. Geben Sie im Feld **Anzeigename** den Namen ein, den Sie für diese Datenbank in Ihrem Projekt verwenden möchten. Der Standardanzeigename ist der ODBC-Datenquellenname. Um eine Beschreibung der Datenbank einzugeben, geben Sie sie im Feld **Beschreibung** ein.
5. Wählen Sie **Kennwort speichern**, um das Kennwort zu speichern, das Sie für die Verbindung zur Datenbank benutzt haben.

Das Aktivieren dieser Option führt dazu, dass das Kennwort in verschleierter Form in der Projektdatei gespeichert wird.

6. Klicken Sie auf **Fertig stellen**, um die konsolidierte Datenbank dem Projekt hinzuzufügen.
7. Wenn Sie aufgefordert werden, die MobiLink-Systemkonfiguration zu installieren, klicken Sie auf **Ja** und dann auf **OK**.

Ergebnisse

Die konsolidierte Datenbank wird dem MobiLink-Projekt hinzugefügt.

Siehe auch

- „Konzepte zur zentralen Administration“ [[MobiLink - Serveradministration](#)]

MobiLink-Systemkonfiguration

Sie müssen Objekte wie Tabellen, Spalten und Trigger hinzufügen, die für die Synchronisation erforderlich sind, bevor Sie eine Datenbank als konsolidierte Datenbank für MobiLink verwenden können. Sie fügen die Objekte hinzu, indem Sie in der Datenbank ein Setupskript ausführen. Für jedes unterstützte RDBMS-System ist ein eigenes Setupskript vorhanden. Alle diese Skripten befinden sich im Ordner `%SQLANY16%\MobiLink\setup`. Sie können die Aktionen des Skripts genau überprüfen, indem Sie es in einem Texteditor öffnen.

Wenn Sie Ihrem MobiLink-Projekt eine konsolidierte Datenbank hinzufügen, wird die MobiLink-Systemkonfiguration geprüft. Wenn sie fehlt, werden Sie aufgefordert, die MobiLink-Systemkonfiguration zu installieren. Sie werden vielleicht auch aufgefordert, ein Upgrade vorzunehmen, wenn die Prüfung eine ältere Version gefunden hat. Die Aufforderung zur Installation der MobiLink-Systemkonfiguration erfolgt auch, wenn Sie ein Deployment eines Synchronisationsmodells vornehmen, wenn Sie dies noch nicht getan haben.

Siehe auch

- „MobiLink-Systemkonfiguration überprüfen“ auf Seite 27
- „Einrichten einer konsolidierten Datenbank“ [*MobiLink - Serveradministration*]
- „Deployment des Synchronisationsmodells“ auf Seite 50

MobiLink-Systemkonfiguration überprüfen

Prüfen Sie, ob die für die Synchronisation erforderlichen Objekte vorhanden sind.

Voraussetzungen

Ein MobiLink-Projekt muss definiert sein und das Projekt muss mindestens eine konsolidierte Datenbank enthalten.

Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.
2. Doppelklicken Sie auf **Konsolidierte Datenbanken**, wählen Sie die konsolidierte Datenbank, die Sie prüfen möchten, und klicken Sie auf **MobiLink-Systemkonfiguration prüfen**.
3. Wenn die Konfiguration noch nicht installiert oder nicht aktualisiert ist, klicken Sie auf **Ja**, um es zu installieren oder zu aktualisieren, und klicken Sie dann auf **OK**.

Ergebnisse

Die Systemkonfiguration wird je nach Auswahl installiert oder aktualisiert.

MobiLink-Systemkonfiguration löschen

Löschen Sie die Objekte, die für die Synchronisation erforderlich sind und die durch die MobiLink-Systemkonfiguration installiert wurden.

Voraussetzungen

Es muss ein MobiLink-Projekt mit einer konsolidierten Datenbank mit installierter MobiLink-Systemeinrichtung definiert sein.

Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.

2. Doppelklicken Sie auf **Konsolidierte Datenbanken**, wählen Sie die konsolidierte Datenbank, die Sie erstellen möchten, und klicken Sie auf **MobiLink-Systemkonfiguration entfernen**.
3. Sie werden gewarnt, dass alle Synchronisationsmodelle aus der Datenbank entfernt werden, der Synchronisationszustand verlorengeht und die Datenbank aus dem Projekt entfernt wird. Klicken Sie auf **Ja**.

Ergebnisse

Die MobiLink-Systemkonfiguration wird aus der Datenbank entfernt und die Datenbank wird aus dem Projekt entfernt.

MobiLink-Server-Befehlszeilen erstellen

Verwenden Sie das Fenster **Eigenschaften der MobiLink-Server-Befehlszeile**, um die MobiLink-Server-Befehlszeilen für MobiLink-Projekte zu speichern.

Voraussetzungen

Ein MobiLink-Projekt muss definiert sein.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen.
2. Doppelklicken Sie auf **MobiLink-Server-Befehlszeilen**
3. Klicken Sie auf **Datei » Neu » MobiLink-Server-Befehlszeile**.
4. Geben Sie einen aussagekräftigen Namen für die Befehlszeile ein.
5. Wählen Sie eine konsolidierte Datenbank aus der Dropdown-Liste **Konsolidierte Datenbank**.
6. Klicken Sie auf **Hinzufügen**, um Netzwerkoptionen anzugeben, die festlegen, wie der MobiLink-Server auf Synchronisationsanforderungen wartet. Wenn keine Optionen angegeben werden, wartet der MobiLink-Server auf Port 2439 auf TCP/IP-Verbindungen.
7. Wählen Sie die Ausführlichkeitsstufe aus der Dropdown-Liste **Ausführlichkeitsstufe**. Wenn Sie die Option **Benutzerdefiniert** wählen, wird ein Fenster geöffnet, in dem Sie aus den verfügbaren Ausführlichkeitsoptionen wählen können.
8. Wählen Sie falls gewünscht weitere Optionen auf der Seite **Erweitert**. Wenn eine erforderliche Option nicht in der Liste enthalten ist, können Sie sie im Feld **Weitere Optionen** eingeben.
9. Überprüfen Sie die angezeigte Befehlszeile im Feld **Befehlszeile**. Bearbeiten Sie die oben genannten Optionen falls nötig oder klicken Sie auf **OK**, um die Befehlszeile zu speichern.

Ergebnisse

Die Befehlszeile wird mit den angegebenen Optionen gespeichert.

LDAP-Server hinzufügen

Fügen Sie einen vertrauenswürdigen LDAP-Server hinzu, den der MobiLink-Server zur Authentifizierung von Benutzern verwendet.

Voraussetzungen

Es muss ein MobiLink-Projekt mit mindestens einer konsolidierten Datenbank definiert sein.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen und auf die konsolidierte Datenbank, mit der Sie arbeiten möchten.
2. Doppelklicken Sie auf **LDAP-Server**.
3. Klicken Sie auf **Datei » Neu » LDAP-Server**.
4. Befolgen Sie die Anweisungen im **Assistenten zum Erstellen eines LDAP-Servers**.
5. Klicken Sie auf **Fertig stellen**.

Ergebnisse

Der LDAP-Server wird erstellt und auf der Registerkarte **LDAP-Server** angezeigt.

Nächste Schritte

Sie können Benutzerauthentifizierungsrichtlinien einrichten, um festzulegen, wie MobiLink-Benutzer mithilfe von LDAP-Servern authentifiziert werden. Siehe [„Benutzerauthentifizierungsrichtlinie hinzufügen“](#) auf Seite 29.

Benutzerauthentifizierungsrichtlinie hinzufügen

Erstellen Sie eine Benutzerauthentifizierungsrichtlinie, um einen MobiLink-Benutzer für einen LDAP-Server zu authentifizieren.

Voraussetzungen

In der konsolidierten Datenbank muss mindestens ein LDAP-Server definiert sein.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen und auf die konsolidierte Datenbank, mit der Sie arbeiten möchten.
2. Doppelklicken Sie auf **Benutzerauthentifizierungsrichtlinien**.
3. Klicken Sie auf **Datei » Neu » Benutzerauthentifizierungsrichtlinie**.

4. Befolgen Sie die Anweisungen im **Assistenten zum Erstellen einer Benutzerauthentifizierungsrichtlinie**.
5. Klicken Sie auf **Fertig stellen**.

Ergebnisse

Die Benutzerauthentifizierungsrichtlinie wird erstellt und auf der Registerkarte **Benutzerauthentifizierungsrichtlinien** angezeigt.

Nächste Schritte

Sie können neue MobiLink-Benutzer erstellen oder bestehende MobiLink-Benutzer zur Verwendung der Authentifizierungsrichtlinie bearbeiten.

Synchronisationsmodelle

Ein **Synchronisationsmodell** erleichtert Ihnen die Erstellung von MobiLink-Anwendungen. Ein Synchronisationsmodell ist eine Datei, die mithilfe des **Assistenten zum Erstellen eines Projekts** oder des **Assistenten zum Erstellen eines Synchronisationsmodells** in Sybase Central erstellt wird.

Nachdem ein Synchronisationsmodell erstellt wurde, können Sie mit der Anpassung des Synchronisationsmodells fortfahren. Änderungen an den konsolidierten oder entfernten Datenbanken werden erst vorgenommen, wenn Sie das Deployment durchführen. Das Modell wird in einer Modelldatei mit der Erweiterung *.mlsm* gespeichert und eine Referenz auf diese Datei wird in Ihrer MobiLink-Projektdatei gespeichert.

Wenn Ihr Modell vollständig ist, können Sie sein Deployment mithilfe des **Assistenten für das Deployment eines Synchronisationsmodells** durchführen. Der **Assistent für das Deployment eines Synchronisationsmodells** erstellt Skriptdateien, um MobiLink-Server und -Client mit den von Ihnen gewählten Deployment-Optionen auszuführen. Sie können wählen, ob Sie an den vorhandenen Datenbanken Änderungen vornehmen, wenn Sie das Deployment durchführen, oder ob Sie vom Assistenten SQL-Skriptdateien erstellen lassen, die Sie später ausführen. Dateien, die für die entfernte Datenbank erstellt werden, können in entfernten Aufgaben eingesetzt werden.

Nach dem Deployment können Sie die Anpassung des Synchronisationsmodells oder der Datenbank fortsetzen und das Deployment erneut vornehmen. Wenn erforderlich, können Sie das per Deployment bereitgestellte Synchronisationssystem aus Sybase Central mithilfe der in der MobiLink-Dokumentation beschriebenen Verfahren ändern.

Einrichten einer MobiLink-Anwendung mit dem Assistenten zum Erstellen eines Synchronisationsmodells

Verwenden Sie den **Assistenten zum Erstellen eines Synchronisationsmodells** in Sybase Central zum Einrichten von Synchronisationslogik für eine MobiLink-Anwendung.

Voraussetzungen

Ein MobiLink-Projekt muss definiert sein.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projekt-Namen und doppelklicken Sie dann auf **Synchronisationsmodelle**.
2. Klicken Sie auf **Datei » Neu » Synchronisationsmodell**, um den **Assistenten zum Erstellen eines Synchronisationsmodells** zu starten.
3. Auf der Seite **Willkommen** wählen Sie einen Namen für Ihr Synchronisationsmodell. Ihr Modell wird als *.mlsm*-Datei im Projektverzeichnis gespeichert. Klicken Sie auf **Weiter**.
4. Bestätigen Sie auf der Seite **Primärschlüssel-Anforderungen**, dass Ihr System die Primärschlüssel-Anforderungen erfüllt. Aktivieren Sie dann die drei Kontrollkästchen und klicken Sie auf **Weiter**. Weitere Hinweise zu Primärschlüsseln finden Sie unter „[Eindeutige Primärschlüssel](#)“ [*MobiLink - Serveradministration*].
5. Wählen Sie auf der Seite **Schema der konsolidierten Datenbank** die konsolidierte Datenbank, die für das Beziehen Ihres konsolidierten Datenbankschemas benutzt werden soll, und klicken Sie dann auf **Weiter**.

Wenn Sie eine Oracle-Datenbank auswählen, werden Sie unter Umständen aufgefordert, eine Teilmenge von Eigentümern auszuwählen, da das Laden eines Schemas für alle Eigentümer lange Zeit in Anspruch nehmen kann.

6. Sie können das Schema der entfernten Datenbank basierend auf dem Schema der konsolidierten Datenbank oder einer vorhandenen entfernten Datenbank erstellen. Die vorhandene entfernte Datenbank kann eine SQL Anywhere- oder UltraLite-Datenbank sein. Hilfe bei Ihrer Entscheidung finden Sie unter „[Entfernte Schemata](#)“ auf Seite 32.
7. Befolgen Sie die restlichen Anweisungen des **Assistenten zum Erstellen eines Synchronisationsmodells**. Wenn möglich werden Standardempfehlungen basierend auf Erfahrungen verwendet.
8. Klicken Sie auf **Fertig stellen**.

Ergebnisse

Wenn Sie auf **Fertig stellen** klicken, wird das Synchronisationsmodell im rechten Fensterausschnitt von Sybase Central angezeigt, in dem Sie das Modell anzeigen oder bearbeiten bzw. das Deployment des Modells durchführen können. Siehe „[Synchronisationsmodellaufgaben](#)“ auf Seite 33.

Nächste Schritte

Nachdem Sie ein Synchronisationsmodell erstellt haben, gehen Sie wie folgt vor:

- Testen Sie das Synchronisationsmodell mithilfe des Fensters **Test**.

- Bearbeiten Sie die Zuordnungen, Ereignisse und Authentifizierungsinformationen. Siehe „[Tabellen- und Spaltenzuordnungen](#)“ auf Seite 34.
- Verwenden Sie für das Deployment des Synchronisationsmodells den **Assistenten für das Deployment eines Synchronisationsmodells**. Siehe „[Deployment des Synchronisationsmodells](#)“ auf Seite 50.

Synchronisationsmodell entfernen

Verwenden Sie das MobiLink-Plug-In, um Synchronisationsmodelle aus einer konsolidierten Datenbank zu entfernen, wenn sie mit dem MobiLink-Plug-In hinzugefügt wurden.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen, das mithilfe des MobiLink-Plug-Ins hinzugefügt wurde.

Kontext und Bemerkungen

Diese Funktion ist für DB2-Datenbanken nicht verfügbar.

Aufgabe

1. Doppelklicken Sie auf das Projekt, in dem das zu entfernende Synchronisationsmodell enthalten ist.
2. Doppelklicken Sie auf **Synchronisationsmodelle**, klicken Sie mit der rechten Maustaste auf das Synchronisationsmodell, das Sie entfernen wollen, und wählen Sie **Aus der konsolidierten Datenbank entfernen**.
3. Wenn das Synchronisationsmodell in mehr als einer konsolidierten Datenbank vorhanden ist, wählen Sie die konsolidierte Datenbank, aus der das Synchronisationsmodell entfernt werden soll.
4. Klicken Sie auf **OK**.

Ergebnisse

Das Synchronisationsmodell wird aus der ausgewählten konsolidierten Datenbank entfernt.

Entfernte Schemata

Ein Synchronisationsmodell enthält ein Schema für eine entfernte Datenbank. Dieses Schema können Sie einer vorhandenen entfernten Datenbank oder einer konsolidierten Datenbank entnehmen.

Verwenden Sie eine entfernte Datenbank in folgenden Fällen:

- Wenn bereits eine entfernte Datenbank vorhanden ist, insbesondere wenn ihr Schema keine Teilmenge des konsolidierten Datenbankschemas ist.
- Wenn die konsolidierten und entfernten Spalten verschiedene Typen haben müssen.

- Wenn die entfernten Tabellen unterschiedliche Eigentümer als die Tabellen in der konsolidierten Datenbank haben müssen. Bei entfernten Schemata einer neuen SQL Anywhere-Datenbank, die aus der konsolidierten Datenbank erstellt wurden, entspricht der Eigentümer der entfernten Tabellen dem Eigentümer der entsprechenden Tabellen in der konsolidierten Datenbank. Wenn Sie einen anderen Eigentümer wünschen, verwenden Sie eine bestehende entfernte SQL Anywhere-Datenbank und richten den Tabelleneigentümer entsprechend ein.

Hinweis

Sie können ein bestehendes Datenbankschema manuell ändern und dann den **Assistenten zur Schemaaktualisierung** ausführen, um das Synchronisationsmodell in Ihrem MobiLink-Projekt zu aktualisieren. Siehe „[Aktualisierung der Schemata](#)“ auf Seite 49.

Wenn Sie das Deployment Ihres Modells durchführen, haben Sie für Ihre entfernte Datenbank drei Optionen, unabhängig davon, wie Sie das entfernte Schema im Modell erstellt haben. Für das Deployment der entfernten Datenbank stehen Ihnen folgende Optionen zur Verfügung:

- **Neue entfernte Datenbank erstellen** Ein Deployment kann eine neue entfernte Datenbank mithilfe des entfernten Schemas aus dem Synchronisationsmodell erstellen. Die Datenbank wird mit Standardoptionen erstellt.
- **Vorhandene entfernte Datenbank ohne Benutzertabellen aktualisieren** Sie können wählen, die zu synchronisierenden Tabellen während des Deployments zu erstellen oder neu zu erstellen. Wählen Sie diese Option, wenn Sie nicht standardmäßige Optionen zum Erstellen einer Datenbank verwenden möchten, z.B. eine bestimmte Kollation.

Für SQL Anywhere-Datenbanken gibt es Optionen, die nicht festgelegt werden können, nachdem die Datenbank erstellt wurde. Siehe [Bemerkungen zum Dienstprogramm Initialisierung \(dbinit\) \[SQL Anywhere Server - Datenbankadministration\]](#).

Eigenschaften von UltraLite-Datenbanken können nicht geändert werden, nachdem die Datenbank erstellt wurde. Siehe „[UltraLiteDatenbank-Erstellungsparameter](#)“ [[UltraLite - Datenbankverwaltung](#)].

- **Eine vorhandene entfernte Datenbank aktualisieren, deren Schema mit dem Schema im Modell übereinstimmt** Diese Option wird benutzt, um eine vorhandene entfernte Datenbank zu synchronisieren. Wenn Sie das Deployment direkt in eine vorhandene entfernte Datenbank durchführen, können Sie festlegen, dass keine der zu synchronisierenden Tabellen erstellt oder neu erstellt wird. Vorhandene Tabellen und deren Inhalte werden nicht geändert.

Die Eigentümer von Tabellen in entfernten SQL Anywhere-Datenbanken sind mit den Eigentümern der ursprünglichen Datenbank identisch. UltraLite-Datenbanktabellen haben keinen Eigentümer.

Siehe auch

- „[Deployment des Synchronisationsmodells](#)“ auf Seite 50

Synchronisationsmodellaufgaben

Sie können mit Ihrem Synchronisationsmodell mehrere Aufgaben ausführen, nachdem Sie es mit dem **Assistenten zum Erstellen eines Synchronisationsmodells** erstellt haben. Änderungen werden nur in

der Synchronisationsmodelldatei gespeichert. Sie werden in ihrer konsolidierten oder entfernten Datenbank erst gespeichert, wenn ein Deployment des Synchronisationsmodells vorgenommen wird.

Sie können Synchronisationsmodelle außerhalb von Sybase Central ändern, aber diese Änderungen nicht in das Modell zurückführen. Sie können beispielsweise MobiLink-Skripten mithilfe von Systemprozeduren hinzufügen oder ändern. Siehe „[Systemprozeduren des MobiLink-Servers](#)“ [[MobiLink - Serveradministration](#)].

Tabellen- und Spaltenzuordnungen

Tabellenzuordnungen geben an, welche Tabellen synchronisiert werden, wie Tabellen synchronisiert werden und wie die synchronisierten Daten zwischen der konsolidierten und der entfernten Datenbank zugeordnet werden.

Synchronisation durch reinen Upload, reinen Download oder keine Synchronisation für Tabellen und Spalten MobiLink führt standardmäßig eine vollständige bidirektionale Synchronisation durch. Sie können jede Tabelle in einen reinen Upload oder einen reinen Download ändern. Sie können auch entscheiden, eine Tabelle nicht zu synchronisieren, wodurch ihre Tabellenzuordnung entfernt wird.

In einem Synchronisationsmodell können Tabellen nur als reiner Download festgelegt werden. Sie können keine reinen Downloadpublikationen erstellen. Siehe „[Reine Download-Publikationen](#)“ [[MobiLink - Clientadministration](#)].

Richtung für Tabellenzuordnungen ändern

Ändern Sie die Richtung für Tabellenzuordnungen, um anzuzeigen, wie die Tabelle synchronisiert werden soll oder dass sie nicht synchronisiert werden soll.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen.

Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.
2. Doppelklicken Sie auf **Synchronisationsmodelle** und doppelklicken Sie auf den Namen Ihres Synchronisationsmodells.
3. Klicken Sie auf die Registerkarte **Zuordnungen**.
4. Wählen Sie im Fenster **Tabellenzuordnungen** eine konsolidierte Tabelle aus.
5. Wählen Sie in der Dropdown-Liste **Richtung** eine der folgenden Optionen:
 - **Nicht synchronisiert** Die Auswahl dieser Option entspricht dem Löschen der Tabellenzuordnung.
 - **Bidirektional** Datenbankvorgänge werden von der entfernten Datenbank und der konsolidierten Datenbank synchronisiert und umgekehrt.

- **Nur in entfernte Datenbank herunterladen** Änderungen werden nur von der konsolidierten Datenbank in die entfernte Datenbank synchronisiert.
- **Nur in konsolidierte Datenbank hochladen** Änderungen werden nur von der entfernten Datenbank in die konsolidierte Datenbank synchronisiert.

Ergebnisse

Die Tabellenzuordnungen werden aktualisiert.

Tabellenzuordnung entfernen

Durch das Entfernen einer Tabellenzuordnung wird die Tabelle nicht mehr synchronisiert.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen.
2. Doppelklicken Sie auf **Synchronisationsmodelle** und doppelklicken Sie auf den Namen Ihres Synchronisationsmodells.
3. Klicken Sie auf die Registerkarte **Zuordnungen**.
4. Wählen Sie im Fenster **Tabellenzuordnungen** eine Tabellenzuordnung aus.
5. Klicken Sie in der Dropdown-Liste **Richtung** auf die Option **Nicht synchronisiert**.

Ergebnisse

Die Zuordnung wird gelöscht, wenn Sie das Synchronisationsmodell das nächste Mal speichern.

Tabellenzuordnungen ändern

Prüfen Sie Tabellen- und Spaltenzuordnungen und passen Sie sie an. Wenn Ihr Modell auf einer vorhandenen entfernten Datenbank basiert, wird angenommen, dass es sich bei den Tabellen- und Spaltenzuordnungen um die bestmöglichen Zuordnungen handelt.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen.
2. Doppelklicken Sie auf **Synchronisationsmodelle** und doppelklicken Sie auf den Namen Ihres Synchronisationsmodells.

3. Klicken Sie auf die Registerkarte **Zuordnungen**.
4. Im Fensterausschnitt **Tabellenzuordnungen** klicken Sie auf die entfernte Tabelle für die Zuordnung, die Sie ändern wollen.
5. Klicken Sie auf die Schaltfläche mit den drei Punkten neben dem Namen der entfernten Tabelle und wählen Sie eine andere Tabelle aus der Liste der nicht synchronisierten entfernten Tabellen.
 - Sie können nur entfernte Tabelle auswählen, die nicht bereits konsolidierten Tabellen zugeordnet sind.
 - Zum Hinzufügen von Tabellen zu Ihrem entfernten Schema siehe „Aktualisierung der Schemata“ auf Seite 49.

Ergebnisse

Die Tabellenzuordnungen werden aktualisiert.

Eine nicht zugeordnete Tabelle einer Zuordnung hinzufügen

Fügen Sie Tabellen in der konsolidierten Datenbank hinzu, die dem Synchronisationsmodell noch nicht zur Synchronisation zugeordnet wurde.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen.
2. Doppelklicken Sie auf **Synchronisationsmodelle** und doppelklicken Sie auf den Namen Ihres Synchronisationsmodells.
3. Klicken Sie auf die Registerkarte **Zuordnungen**.
4. Klicken Sie auf **Datei » Neu » Tabellenzuordnungen**, um das Fenster **Neue Tabellenzuordnungen erstellen** zu öffnen und dort die hinzuzufügenden konsolidierten Tabellen auszuwählen.
5. Markieren Sie eine oder mehrere der konsolidierten Tabellen, um sie zum entfernten Schema hinzuzufügen. Die neuen entfernten Tabellen haben denselben Namen und dieselben Spalten wie die entsprechenden konsolidierten Tabellen. Die Zuordnung zwischen der entfernten und der konsolidierten Tabelle wird automatisch erzeugt.

Die folgenden Optionen stehen zur Verfügung:

- **Die ausgewählten Tabellen dem entfernten Schema hinzufügen, wenn sie noch nicht existieren** Wenn Sie die Änderungen am entfernten Schema nicht durchführen wollen, deaktivieren Sie diese Option.
- **Wählen Sie die Spalten für die entfernten Tabellennamen.** Wählen Sie diese Option, um die Tabellen, aber nicht alle Spalten zum entfernten Schema hinzuzufügen. Durch Aktivieren dieser Option können Sie die Spalten für jede ausgewählte Tabelle wählen.

- **Tabellen ausblenden, die wie Synchronisationsmodell-Schattentabellen benannt sind** Standardmäßig werden konsolidierte Datenbanktabellen mit Namen wie "Synchronisationsmodell-Schattentabelle" nicht angezeigt, weil solche Schattentabellen nicht synchronisiert werden sollen.

Ergebnisse

Die neuen Tabellen werden der Zuordnung für das Synchronisationsmodell hinzugefügt.

Eine Spaltenzuordnung für eine Tabellenzuordnung ändern

Prüfen Sie die Tabellen- und Spaltenzuordnungen und passen Sie sie falls erforderlich an. Wenn Ihr Modell auf einer vorhandenen entfernten Datenbank basiert, wird angenommen, dass es sich bei den Tabellen- und Spaltenzuordnungen um die bestmöglichen Zuordnungen handelt.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen.

Kontext und Bemerkungen

Sie können eine Spalte in einer synchronisierten konsolidierten Tabelle einer entfernten Tabellenspalte bzw. einem während der Synchronisation ermittelten Wert zuordnen oder die Spalte aus der Synchronisation ausschließen. Wenn eine Zuordnung zu einem Wert erfolgt, können Sie den MobiLink-Benutzernamen, die entfernte Datenbank-ID oder einen SQL-Ausdruck verwenden (wobei benannte MobiLink-Parameter eingeschlossen sind). Wenn Sie eine Primärschlüsselspalte einem Wert zuordnen und die Tabellenzuordnung bidirektional ist, müssen Sie doppelte Primärschlüssel verhindern, wenn ein Download in eine entfernte Datenbank erfolgt.

Aufgabe

1. Wählen Sie auf der Registerkarte **Tabellenzuordnungen** die Tabellenzuordnung aus.
2. Öffnen Sie im Fensterausschnitt **Details** die Registerkarte **Spaltenzuordnungen**.
3. Rechtsklicken Sie auf die Spaltenzuordnung, die Sie ändern möchten, und wählen Sie eine der folgenden Optionen aus dem Kontextmenü:
 - **Keine**
 - **MobiLink-Benutzername**
 - **Entfernte ID**
 - **Benutzerdefiniert**
 - Eine nicht zugeordnete entfernte Spalte

Um die konsolidierte Spalte mit einer entfernten Spalte zu synchronisieren, wählen Sie die nicht zugeordnete entfernte Spalte aus der unteren Gruppe des Menüs aus. Nur nicht zugeordnete Spalten werden aufgelistet.

Um die konsolidierte Spalte aus der Synchronisation auszuschließen, klicken Sie auf **Keine**. Das Richtungssymbol zeigt, dass die konsolidierte Spalte nicht synchronisiert ist.

Wenn Sie die konsolidierte Spalte einem Wert zuordnen möchten, können Sie mithilfe der Option **MobiLink-Benutzernamen**, **Entfernte ID** oder **Benutzerdefiniert** einen SQL-Ausdruck eingeben, der ausgewertet wird, wenn während der Synchronisation die Synchronisationsskripten `upload_insert`, `upload_update` und `upload_delete` der entfernten Tabelle ausgeführt werden. Das Richtungssymbol zeigt, dass der Wert nur hochgeladen wird. Die konsolidierte Spalte wird nicht in die entfernte Datenbank heruntergeladen.

Ergebnisse

Die Spaltenzuordnungen werden aktualisiert.

Downloadtyp ändern

Ändern Sie den Downloadtyp auf der Registerkarte **Downloadtyp** im Fensterausschnitt **Details**. Der Downloadtyp einer Tabellenzuordnung kann Zeitstempel, Snapshot oder benutzerdefiniert sein.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen.

Kontext und Bemerkungen

Die Downloadtypen lauten wie folgt:

- **Zeitstempelbasierter Download** Wählen Sie diese Option, um den zeitstempelbasierten Download als Standard zu verwenden. Es werden nur Zeilen heruntergeladen, die seit der letzten Synchronisation geändert wurden. Siehe „Implementieren zeitstempelbasierter Downloads“ [[MobiLink - Serveradministration](#)].
- **Snapshot-Download** Wählen Sie diese Option, um den Snapshot-Download als Standard zu verwenden. Alle Zeilen werden bei jeder Synchronisation heruntergeladen, auch wenn sie seit der letzten Synchronisation nicht geändert wurden. Siehe:
 - „Snapshot-Synchronisation“ [[MobiLink - Serveradministration](#)]
 - Einsatz der Snapshot-Synchronisation [[MobiLink - Serveradministration](#)]
- **Benutzerdefinierte Downloadlogik** Wählen Sie diese Option, um eigene `download_cursor`- und `download_delete_cursor`-Skripten zu erstellen, statt sie generieren zu lassen. Siehe:
 - „Schreiben von Synchronisationsskripten“ [[MobiLink - Serveradministration](#)]
 - „download_cursor-Skripten“ [[MobiLink - Serveradministration](#)]
 - „download_delete_cursor-Skripten“ [[MobiLink - Serveradministration](#)]

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen.
2. Doppelklicken Sie auf **Synchronisationsmodelle** und doppelklicken Sie auf den Namen Ihres Synchronisationsmodells.

3. Öffnen Sie die Registerkarte **Zuordnungen**.
4. Wählen Sie im Fenster **Tabellenzuordnungen** eine Tabellenzuordnung aus.
5. Wählen Sie im Fensterausschnitt **Details** die Registerkarte **Downloadtyp**.
6. Wählen Sie in der Dropdown-Liste **Downloadtyp** die Option **Zeitstempel**, **Snapshot** oder **Benutzerdefiniert**.
7. Wenn Sie **Benutzerdefiniert** wählen, klicken Sie im oberen Fensterausschnitt auf die Registerkarte **Ereignisse** und geben die Namen Ihrer download_cursor- und download_delete_cursor-Skripten ein.
8. Wenn Sie **Zeitstempel** wählen, führen Sie die folgenden Aufgaben aus:
 - a. Geben Sie einen Spaltennamen im Feld **Name der Zeitstempelspalte** ein.
 - b. Wählen Sie einen der folgenden Abschnitte:
 - i. **Zeitstempelspalte in konsolidierter Tabelle verwenden**
 - **Spaltenstandardwert anstatt eines Triggers verwenden**
 - ii. **Schattentabelle für Zeitstempelspalte verwenden**

Ergebnisse

Der Downloadtyp wird geändert.

Aufzeichnungsweise von Löschungen ändern

Verwenden Sie die Registerkarte **Löschungen herunterladen**, um zu steuern, ob Löschungen in der konsolidierten Datenbank heruntergeladen werden, und wie die Informationen über die Löschungen in der konsolidierten Datenbank gespeichert werden.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen und der Downloadtyp für die Tabellenzuordnung muss auf **Zeitstempel** gesetzt sein.

Kontext und Bemerkungen

Wenn Sie den Snapshot-Download verwenden, werden alle Zeilen in der entfernten Datenbank gelöscht, bevor der Snapshot heruntergeladen wird, sodass es nicht erforderlich ist, die Vorgänge zu protokollieren. Wenn Sie den zeitstempelbasierten Download verwenden, können Sie festlegen, wie Löschungen in der konsolidierten Datenbank für die Übertragung in die entfernte Datenbank aufgezeichnet werden.

Um Zeilen aus der entfernten Datenbank zu löschen, wenn sie aus der konsolidierten Datenbank gelöscht werden, müssen Sie eine Aufzeichnung über jede gelöschte Zeile behalten. Sie können die Informationen mit Schattentabellen aufzeichnen oder logische Löschungen verwenden.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen.
2. Doppelklicken Sie auf **Synchronisationsmodelle** und wählen Sie dann den Namen Ihres Synchronisationsmodells aus.
3. Klicken Sie auf die Registerkarte **Zuordnungen**.
4. Wählen Sie im Fenster **Tabellenzuordnungen** eine Tabellenzuordnung aus.
5. Öffnen Sie im Fensterausschnitt **Details** die Registerkarte **Löschungen herunterladen**.
6. Aktivieren Sie das Kontrollkästchen **Löschungen herunterladen**, um Löschungen aus der konsolidierten Datenbank herunterzuladen. Deaktivieren Sie das Kontrollkästchen, um Löschungen aus der konsolidierten Datenbank nicht herunterzuladen.

Um Löschungen aufzuzeichnen, können Sie eine Schattentabelle oder logische Löschungen verwenden.

Ergebnisse

Die Methode zur Behandlung von Löschungen wird aktualisiert.

Logische Löschungen

Bei der Unterstützung des MobiLink-Synchronisationsmodells für logische Löschungen wird davon ausgegangen, dass eine logische Löschspalte sich nur in der konsolidierten Datenbank und nicht in der entfernten Datenbank befindet. Wenn ein konsolidiertes Schema in ein neues entferntes Schema kopiert wird, lassen Sie alle Spalten aus, die mit den logischen Löschspalten in den Synchronisationseinstellungen des Modells übereinstimmen.

Spalten, die zum Standardnamen für logische Löschspalten passen, werden automatisch nicht in neue entfernte Schemata kopiert.

Wenn Sie logische Löschungen in der entfernten Datenbank verwenden möchten, wählen Sie, dass Löschungen nicht heruntergeladen werden, und wenn nötig, aktualisieren Sie das entfernte Schema, um die logische Löschspalte einzubeziehen.

Hinweis

Sie müssen die Spaltenzuordnung für die logische Löschspalte im entfernten Schema auf die logische Löschspalte im konsolidierten Schema festlegen.

Siehe auch

- „Löschungen“ [[MobiLink - Serveradministration](#)]
- „download_delete_cursor-Skripten“ [[MobiLink - Serveradministration](#)]
- „download_cursor (Tabellenergebnis)“ [[MobiLink - Serveradministration](#)]

Teilmengen herunterladen

Jede entfernte MobiLink-Datenbank kann eine Teilmenge der Daten aus der konsolidierten Datenbank synchronisieren. Sie können die Download-Teilmenge für jede Tabelle anpassen.

Folgende Optionen stehen zum Anpassen der Download-Teilmenge zur Verfügung:

- **Benutzer** Wählen Sie diese Option, um Daten nach dem MobiLink-Benutzernamen aufzuteilen. Hierbei werden unterschiedliche Daten für unterschiedliche registrierte MobiLink-Benutzer heruntergeladen.

Um diese Option zu verwenden, muss jede Zeile eine Spalte enthalten, die einen MobiLink-Benutzernamen enthält. Sie wählen die MobiLink-Benutzernamen aus, wenn Sie das Deployment durchführen. Auf diese Weise können Sie Namen auswählen, die vorhandenen Werten in der konsolidierten Datenbank entsprechen. (Die für die MobiLink-Benutzernamen verwendete Spalte muss von einem Typ sein, der die für den Benutzernamen verwendeten Werte speichern kann.) Wenn sich die MobiLink-Benutzernamen nicht in der Tabelle befinden, für die Sie eine Teilmenge erstellen, müssen Sie eine Verknüpfung zu der Tabelle herstellen, in der sie sich befinden.

- **Entfernte ID** Wählen Sie diese Option, um Daten nach der entfernten ID zu partitionieren. Hierbei werden unterschiedliche Daten in unterschiedliche entfernte Datenbanken heruntergeladen.

Um diese Option zu verwenden, muss jede Zeile eine Spalte mit einer entfernten ID enthalten. Entfernte IDs werden standardmäßig als GUIDs erstellt. Sie können jedoch die entfernten IDs so festlegen, dass sie den vorhandenen Werten in der konsolidierten Datenbank entsprechen. (Die für die entfernten IDs verwendete Spalte muss von einem Typ sein, der die für die entfernten IDs verwendeten Werte speichern kann.) Wenn sich die entfernten IDs nicht in der Tabelle befinden, für die Sie eine Teilmenge erstellen, müssen Sie eine Verknüpfung zu der Tabelle herstellen, in der sie sich befinden.

Hinweis

Es ist in der Regel besser, nach Benutzern oder Authentifizierungsparametern zu partitionieren als nach entfernter ID, da sich die entfernte ID ändern kann, wenn der entfernte Computer zurückgesetzt oder ausgetauscht wird.

- **Benutzerdefiniert** Wählen Sie diese Option, um einen SQL-Ausdruck zu verwenden, der bestimmt, welche Zeilen heruntergeladen werden. Jede Synchronisation lädt nur Zeilen herunter, für die der SQL-Ausdruck TRUE ist. Dieser SQL-Ausdruck wird der WHERE-Klausel des generierten download_cursor-Skripts hinzugefügt. Sie können benannte MobiLink-Parameter im Ausdruck verwenden. Sie können auch andere Tabellen referenzieren. Wenn Sie andere Tabellen referenzieren, müssen Sie die anderen Tabellen im Feld über dem Ausdruck auflisten und die Join-Bedingung in Ihren Ausdruck aufnehmen.

Siehe auch

- „MobiLink-Benutzer“ [[MobiLink - Clientadministration](#)]
- „Entfernte IDs“ [[MobiLink - Clientadministration](#)]
- „Partitionierte Zeilen in entfernten Datenbanken“ [[MobiLink - Serveradministration](#)]
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [[MobiLink - Clientadministration](#)]
- „Zeitpunkt des letzten Downloads in Skripten“ [[MobiLink - Serveradministration](#)]
- „download_cursor-Skripten“ [[MobiLink - Serveradministration](#)]

Download-Teilmenge ändern

Passen Sie die Download-Teilmenge für jede Tabelle an. Jede entfernte MobiLink-Datenbank kann eine Teilmenge der Daten in der konsolidierten Datenbank synchronisieren.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen und die Tabellenzuordnung darf nicht vom Downloadtyp **Benutzerdefiniert** sein.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen.
2. Doppelklicken Sie auf **Synchronisationsmodelle** und wählen Sie den Namen Ihres Synchronisationsmodells aus.
3. Klicken Sie auf die Registerkarte **Zuordnungen**.
4. Wählen Sie im Fenster **Tabellenzuordnungen** eine entfernte Tabelle aus.
5. Öffnen Sie im Fensterausschnitt **Details** die Registerkarte **Download-Teilmenge**.
6. Wählen Sie eine der folgenden Download-Teilmenge aus der Dropdown-Liste **Download-Teilmenge: Keine, Benutzer, Entfernt** oder **Benutzerdefiniert**.
7. Wenn Sie **Benutzer** oder **Entfernt** wählen, geben Sie an, wo sich die Spalte mit dem Benutzernamen und der entfernten ID befindet.

Wenn sich die Spalte in der konsolidierten Tabelle befindet, die synchronisiert wird, wählen Sie **Spalte in der konsolidierten Tabelle verwenden**. Wählen Sie dann die Spalte mit dem Benutzernamen oder der entfernten ID aus der Dropdown-Liste **Spaltenname**.

Wenn sich die Spalte in einer anderen Tabelle befindet, wählen Sie **Spalte in einer gemeinsamen Beziehungstabelle verwenden**. Wählen Sie die Tabelle, die die Spalte enthält, aus der Dropdown-Liste **Zu verknüpfende Tabelle**. Wählen Sie die Spalte, die den Benutzernamen oder die entfernte ID enthält, in der Dropdown-Liste **Übereinstimmende Spalte** aus. Verwenden Sie die Join-Bedingung, um die Verknüpfung der synchronisierenden Tabelle mit der Schattentabelle zu definieren.

8. Wenn Sie **Benutzerdefiniert** gewählt haben, gibt es zwei Textfelder, in die Sie Informationen zum Erstellen eines download_cursor-Skripts eingeben können. Sie müssen kein vollständiges

download_cursor-Skript schreiben. Geben Sie lediglich zusätzliche Informationen ein, um den Join und andere Einschränkungen in der Download-Teilmenge zu identifizieren.

- Geben Sie in das erste Textfeld (**Tabellen, die der FROM-Klausel des Download-Cursors hinzugefügt werden sollen**) den Tabellennamen ein, wenn das download_cursor-Skript einen Join mit anderen Tabellen erfordert. Wenn der Join mehrere Tabellen erfordert, trennen Sie die Tabellen durch ein Komma.
- Geben Sie in das zweite Textfeld (**In der WHERE-Klausel des Download-Cursors zu verwendender SQL-Ausdruck**), einen SQL-Ausdruck ein, der der generierten WHERE-Klausel hinzugefügt werden soll und der die Download-Teil mengenbedingung und die Join-Bedingung festlegt. Sie können benannte MobiLink-Parameter einschließlich Authentifizierungsparametern in dem Ausdruck verwenden. Standardmäßig werden derselbe Ausdruck und dieselben verknüpften Tabellen für die Download-Löschen-Teilmenge verwendet. Wenn Sie eine Schattentabelle zum Protokollieren von Löschungen verwenden und denselben Ausdruck verwenden möchten, vermeiden Sie die Verwendung des Namens der Basistabelle im Ausdruck. Wenn dies nicht möglich ist, verwenden Sie eine benutzerdefinierte Download-Löschen-Teilmenge.

Ergebnisse

Die Download-Teilmenge wird geändert.

Beispiel

Benutzerbeispiel

Beispielsweise kann die Tabelle ULOrder in CustDB von Benutzern gemeinsam verwendet werden. Standardmäßig werden Bestellungen dem Mitarbeiter zugeordnet, der sie erstellt hat. Es gibt jedoch Fälle, bei denen ein Mitarbeiter Aufträge anzeigen muss, die von einem anderen Mitarbeiter erstellt wurden. Beispiel: Ein Abteilungsleiter benötigt Zugriff auf alle Aufträge, die von den Mitarbeitern der Abteilung erstellt wurden. Die CustDB-Datenbank stellt zu diesem Zweck die ULEmpCust-Tabelle zur Verfügung. Über diese Tabelle können Sie Kunden Mitarbeitern zuordnen. Diese Tabellen laden alle Aufträge für die Mitarbeiter-Kundenbeziehung herunter.

Zeigen Sie download_cursor für ULOrder ohne Download-Teilmenge an. Wählen Sie auf der Registerkarte **Zuordnungen** die Tabelle ULEmpCust aus. Wählen Sie **Zeitstempel** für **Downloadtyp** und **Keine** für **Download-Teilmenge**. Klicken Sie mit der rechten Maustaste in die Tabelle, und wählen Sie **Gehe zu Ereignissen**. Das download_cursor-Skript für die Tabelle sieht folgendermaßen aus:

```
SELECT "DBA"."ULOrder"."order_id",
       "DBA"."ULOrder"."cust_id",
       "DBA"."ULOrder"."prod_id",
       "DBA"."ULOrder"."emp_id",
       "DBA"."ULOrder"."disc",
       "DBA"."ULOrder"."quant",
       "DBA"."ULOrder"."notes",
       "DBA"."ULOrder"."status"
FROM "DBA"."ULOrder"
WHERE "DBA"."ULOrder"."last_modified" >= {ml s.last_table_download}
```

Wechseln Sie erneut zur Registerkarte **Zuordnungen**. Wählen Sie auf der Registerkarte **Download-Teilmenge** des Fensterausschnitts **Details** in der Dropdown-Liste **Download-Teilmenge** für ULOrder die

Option **Benutzer**. Wählen Sie **Eine Spalte in einer verknüpften Beziehungstabelle verwenden**. Wählen Sie ULEmpCust als die zu verknüpfende Tabelle. Wählen Sie für die Übereinstimmungsspalte emp_id. Wählen Sie für die Join-Bedingung DBA.ULOrder.cust_id = DBA.ULEmpCust.cust_id.

Klicken Sie mit der rechten Maustaste in die Tabelle im oberen Fensterbereich und wählen Sie **Gehe zu Ereignissen** aus. Das download_cursor-Skript für die Tabelle sieht nun folgendermaßen aus (die neuen Zeilen werden in Fettdruck gezeigt):

```
SELECT "DBA"."ULOrder"."order_id",
       "DBA"."ULOrder"."cust_id",
       "DBA"."ULOrder"."prod_id",
       "DBA"."ULOrder"."emp_id",
       "DBA"."ULOrder"."disc",
       "DBA"."ULOrder"."quant",
       "DBA"."ULOrder"."notes",
       "DBA"."ULOrder"."status"
FROM "DBA"."ULOrder", "DBA"."ULEmpCust"
WHERE "DBA"."ULOrder"."last_modified" >= {ml s.last_table_download}
AND "DBA"."ULOrder"."cust_id" = "DBA"."ULEmpCust"."cust_id"
AND "DBA"."ULEmpCust"."cust_id" = {ml s.username}
```

Beispiel für benutzerdefinierte Einstellungen

Angenommen, Sie möchten den Download einer Tabelle namens **Customer** nach MobiLink-Benutzern in Teilmengen aufteilen und nur Zeilen mit active=1 herunterladen. Die MobiLink-Benutzernamen sind in der auf Teilmengen aufgeteilten Tabelle nicht enthalten, daher müssen Sie einen Join zu einer Tabelle namens SalesRep erstellen, die die MobiLink-Benutzernamen enthält.

Wählen Sie auf der Registerkarte **Zuordnungen** die Zuordnung für die Tabelle **Customer**. Öffnen Sie die Registerkarte **Downloadtyp** im Fensterausschnitt **Details** Setzen Sie den Downloadtyp auf **Zeitstempel**. Öffnen Sie die Registerkarte **Download-Teilmenge** im Fensterausschnitt **Details**. Wählen Sie **Benutzerdefiniert** in der Dropdown-Liste **Download-Teilmenge**. Geben Sie in das erste Textfeld (**Tabellen, die der FROM-Klausel des Download-Cursors hinzugefügt werden sollen**) Folgendes ein:

```
SalesRep
```

Geben Sie in das zweite Textfeld (**In der WHERE-Klausel des Download-Cursors zu verwendender SQL-Ausdruck**) Folgendes ein:

```
SalesRep.ml_username = {ml s.username}
AND Customer.active = 1
AND Customer.cust_id = SalesRep.cust_id
```

Beide Tabellen haben eine cust_id-Spalte, sodass bei Referenzen auf diese Spalten der Tabellename im Ausdruck vorangestellt werden muss. Wenn Sie eine Schattentabelle zum Protokollieren von Löschungen verwenden, die heruntergeladen werden sollen, müssen Sie **Keine** oder **Benutzerdefiniert** in der Spalte **Download-Löschen-Teilmenge** für die Customer-Tabellenzuordnung auswählen, weil der Name der Schattentabelle "Customer_del" und nicht "Customer" lautet.

Klicken Sie mit der rechten Maustaste in die Tabelle im oberen Fensterbereich und wählen Sie **Gehe zu Ereignissen** aus. Das download_cursor-Skript für die Tabelle sieht nun folgendermaßen aus:

```
SELECT "DBA"."Customer"."cust_id",
       "DBA"."Customer"."cust_name"
FROM "DBA"."Customer", SalesRep
```

```
WHERE "DBA"."Customer"."last_modified" >= {ml s.last_table_download}
AND SalesRep.ml_username = {ml s.username}
AND Customer.active = 1
AND Customer.cust_id = SalesRep.cust_id
```

Die letzte Zeile der WHERE-Klausel erstellt einen Schlüssel-Join zwischen den Tabellen Customer und SalesRep.

Download-Löschen-Teilmenge ändern

Wenn Sie eine Download-Teilmenge zum Synchronisieren der Daten der konsolidierten Datenbank verwenden, ist die Download-Löschen-Teilmenge automatisch auf **Identisch** eingestellt und daher mit der Download-Teilmenge identisch. Sie können dies in **Keine** oder **Benutzerdefiniert** ändern.

Wenn die Download-Teilmenge **Benutzerdefiniert** ist und Sie eine Schattentabelle zum Verfolgen von Löschvorgängen verwenden, müssen alle Spaltenwerte, die von der Download-Teilmenzenlogik verwendet werden, in die Schattentabelle kopiert werden, wenn eine Zeile gelöscht wird. Primärschlüsselspaltenwerte werden automatisch in die Schattentabelle kopiert. Wenn Sie Zugriff auf beliebige andere Spaltenwerte benötigen, geben Sie diese auf der Registerkarte **Download-Löschen-Teilmenge** an, indem Sie alle erforderlichen Spalten in der Liste **Schattentabellenspalten löschen** markieren.

Die Definition einer benutzerdefinierten Download-Teilmenge ist identisch mit dem Einrichten einer benutzerdefinierten Download-Löschen-Teilmenge mit Ausnahme der Angabe der zusätzlichen Spalten, die der Schattentabelle hinzuzufügen sind.

Siehe auch

- [„Download-Teilmenge ändern“ auf Seite 42](#)

Konflikterkennung und Konfliktlösung

Wenn eine Zeile sowohl in der entfernten als auch in der konsolidierten Datenbank aktualisiert wird, tritt bei der nächsten Synchronisation der Datenbanken ein Konflikt auf.

Sie haben folgende Möglichkeiten zum Erkennen von Konflikten:

- **Zeilenbasierte Konflikterkennung** Ein Konflikt wird erkannt, wenn die Zeile seit der letzten Synchronisation sowohl von der entfernten als auch von der konsolidierten Datenbank aktualisiert wurde.

Diese Option definiert ein upload_fetch-Skript und ein upload_update-Skript. Siehe [„Konflikterkennung mit upload_fetch- oder upload_fetch_column_conflict-Skripten“ \[MobiLink - Serveradministration\]](#).

- **Spaltenbasierte Konflikterkennung** Ein Konflikt wird erkannt, wenn dieselbe Spalte für die Zeile sowohl in der entfernten als auch in der konsolidierten Datenbank aktualisiert wurde.

Diese Option definiert ein upload_fetch_column_conflict-Skript. Siehe [„Konflikterkennung mit upload_fetch- oder upload_fetch_column_conflict-Skripten“ \[MobiLink - Serveradministration\]](#).

Wenn eine Tabelle BLOBs enthält und Sie die spaltenbasierte Konflikterkennung gewählt haben, wird die zeilenbasierte Konflikterkennung verwendet.

Sie haben folgende Möglichkeiten zum Lösen von Konflikten:

- **Konsolidiert** Erste gewinnt: Hochgeladene Aktualisierungen werden bei einem Konflikt abgelehnt.
- **Entfernt** Letzte gewinnt: Heraufgeladene Aktualisierungen werden immer angewendet.
- **Zeitstempel** Die neueste Aktualisierung gewinnt. Um diese Option zu verwenden, müssen Sie eine TIMESTAMP-Spalte für die Tabelle erstellen und verwalten. Die TIMESTAMP-Spalte zeichnet den Zeitpunkt der letzten Änderung dieser Zeile auf. Die Spalte muss in der konsolidierten und in der entfernten Datenbank vorhanden sein und darf nicht dieselben Spalten umfassen, die für zeitstempelbasierte Downloads verwendet werden. Die entfernte und die konsolidierte Datenbank müssen die gleiche Zeitzone verwenden (vorzugsweise UTC). Darüber hinaus müssen die Uhren beider Datenbanken synchronisiert werden.
- **Benutzerdefiniert** Sie schreiben Ihre eigenen resolve_conflict-Skripten. Sie führen dies auf der Registerkarte **Ereignisse** durch. Siehe „[Konfliktlösung mit resolve_conflict-Skripten](#)“ [*MobiLink - Serveradministration*].

Siehe auch

- „[Konflikterkennung und -lösung ändern](#)“ auf Seite 46
- „[Konfliktverarbeitung](#)“ [*MobiLink - Serveradministration*]

Konflikterkennung und -lösung ändern

Verwenden Sie Sybase Central, um die Einstellungen für die Konflikterkennung und -lösung anzupassen.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen.
2. Doppelklicken Sie auf den Namen Ihres Synchronisationsmodells.
3. Klicken Sie auf die Registerkarte **Zuordnungen**.
4. Wählen Sie im Fenster **Tabellenzuordnungen** eine Tabellenzuordnung aus.
5. Öffnen Sie im Fensterausschnitt **Details** die Registerkarte **Konfliktverarbeitung**.
6. Klicken Sie in der Dropdown-Liste **Konflikterkennung** auf **Zeilenbasiert** oder **Spaltenbasiert**.
7. Wählen Sie **Konsolidiert**, **Entfernt**, **Zeitstempel** oder **Benutzerdefiniert** in der Dropdown-Liste **Konfliktlösung**.

8. Wenn Sie **Zeitstempel** für die Konfliktlösung gewählt haben, geben Sie den Namen einer zu verwendenden **TIMESTAMP-Spalte** ein.
9. Wenn Sie **Benutzerdefiniert** für die Konfliktlösung gewählt haben, öffnen Sie die Registerkarte **Ereignisse**, und schreiben Sie ein resolve_conflict-Skript für die Tabelle.

Ergebnisse

Die Einstellungen für Konflikterkennung und -lösung werden aktualisiert.

Skriptänderung in einem Synchronisationsmodell

Sie ändern Skripten in einem Synchronisationsmodell auf der Registerkarte **Ereignisse**.

Auf der Registerkarte **Ereignisse** können Sie folgende Aufgaben ausführen:

- Skripten, die basierend auf den aktuellen Tabellenzuordnungen generiert wurden, anzeigen und bearbeiten
- Erstellen eines neuen Skripts

Im oberen Bereich der Registerkarte **Ereignisse** wird die Gruppe angezeigt, zu der das ausgewählte Skript gehört. Alle Skripten für eine einzelne Tabelle werden zu einer Gruppe zusammengefasst. Im oberen Bereich der Registerkarte **Ereignisse** wird der Name des ausgewählten Skripts angezeigt. Außerdem wird angezeigt, ob es vom MobiLink-Plug-In generiert wurde, ob es benutzerdefiniert ist oder ob ein bereits generiertes Skript überschrieben wurde. Darüber hinaus sehen Sie hier, ob die Synchronisationslogik in SQL, .NET oder Java geschrieben ist.

Das Skript steht vollständig unter Ihrer Kontrolle, wenn Sie ein Skript zum Überschreiben eines generierten Skripts hinzufügen. Es wird nicht automatisch geändert, wenn Sie eine zugehörige Einstellung ändern. Wenn Sie beispielsweise ein download_delete_cursor-Skript für ein Modell ändern und dann die **Lösch**-Spalte im Feld **Tabellenzuordnungen** unter **Zuordnungen** deaktivieren, wird diese Änderung nicht auf das benutzerdefinierte download_delete_cursor-Skript angewendet.

Mithilfe der Optionen im Menü **Datei** können Sie generierte Skripten, die Sie überschrieben haben, wiederherstellen, Skripten, die zuvor ignoriert werden sollten, wiederherstellen oder von Ihnen neu hinzugefügte Skripten entfernen. Wählen Sie die Skripten aus, die Sie wiederherstellen möchten, und klicken Sie auf das Menü **Datei**, um die verfügbaren Optionen anzuzeigen.

Skript für eine bestimmte Tabelle suchen

Verwenden Sie Sybase Central, um die Skripten für eine bestimmte Tabelle zu finden.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen.

Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.
2. Doppelklicken Sie auf den Namen Ihres Synchronisationsmodells.
3. Öffnen Sie die Registerkarte **Ereignisse**.
4. Wählen Sie in der Dropdown-Liste **Gruppe** die Tabelle aus, deren Ereignisse Sie sehen möchten.
5. Wählen Sie im Feld **Ereignis** den Namen des Skripts, das Sie finden wollen. Vorhandene Skripten werden in Fettdruck markiert.

Ergebnisse

Der Cursor bewegt sich auf das ausgewählte Skript.

Nächste Schritte

Führen Sie die gewünschten Änderungen im ausgewählten Skript durch.

Auf einem externen POP3-, IMAP- oder LDAP-Server authentifizieren

Aktivieren Sie die Authentifizierung der Synchronisation auf einem externen Server für ein Synchronisationsmodell mithilfe der Registerkarte **Authentifizierung**.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen.
2. Doppelklicken Sie auf **Synchronisationsmodelle** und doppelklicken Sie auf den Namen Ihres Synchronisationsmodells.
3. Öffnen Sie die Registerkarte **Authentifizierung** im rechten Fensterausschnitt.
4. Wählen Sie **Benutzerdefinierte Authentifizierung für dieses Synchronisationsmodell zulassen**.
5. Wählen Sie den Servertyp, für den Sie die Authentifizierung durchführen möchten.
6. Geben Sie die entsprechenden Host-, Port- und URL-Informationen in die entsprechenden Felder ein.

Weitere Hinweise zu diesen Feldern finden Sie unter „[Eigenschaften für externe Authentifizierer](#)“ [[MobiLink - Clientadministration](#)].

Ergebnisse

Der Standardwert ist "Aktiviert".

Aktualisierung der Schemata

Verwenden Sie den **Assistenten zur Schemaaktualisierung**, um die konsolidierten und entfernten Datenbankschemata in Ihrem Synchronisationsmodell zu aktualisieren.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen.

Kontext und Bemerkungen

Verwenden Sie den **Assistenten zur Schemaaktualisierung**, nachdem Sie das Deployment für Ihr Modell und folgende Aufgaben durchgeführt haben:

- Sie haben eine Änderung an der entfernten Datenbank vorgenommen, die in das Modell eingefügt werden muss.
- Sie haben eine Änderung an der konsolidierten Datenbank vorgenommen, die in das Modell eingefügt werden muss.

Sie müssen beispielsweise das Aktualisierungsschema ausführen, bevor Sie das erneute Deployment für ein Modell durchführen, das einen zeitstempelbasierten Download für eine oder mehrere Tabellen erstellt hat. Das Schema der konsolidierten Datenbank wurde beim vorherigen Deployment geändert, indem eine Zeitstempelspalte oder Schattentabelle hinzugefügt wurde. Aus diesem Grund müssen Sie das Schema aktualisieren.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen.
2. Doppelklicken Sie auf **Synchronisationsmodelle**.
3. Doppelklicken Sie auf den Namen Ihres Synchronisationsmodells und klicken Sie auf **Datei » Schema aktualisieren**.
4. Wählen Sie eine der folgenden Optionen:
 - **Das Schema der konsolidierten Datenbank** Das konsolidierte Schema im Modell wird aktualisiert. Das entfernte Schema im Modell wird nicht geändert.
 - **Das Schema der entfernten Datenbank** Das entfernte Schema im Modell wird aktualisiert. Das konsolidierte Schema im Modell wird nicht geändert.
 - **Das Schema der konsolidierten und der entfernten Datenbank** Das Schema der konsolidierten und das Schema der entfernten Datenbank werden im Modell aktualisiert, um den Schemata der vorhandenen Datenbanken zu entsprechen.
5. Befolgen Sie die Anweisungen des **Assistenten zur Schemaaktualisierung**.

Wenn Sie auf **Fertig stellen** klicken, werden Änderungen außerhalb des Modells erst nach dem Deployment des Synchronisationsmodells durchgeführt. Bis zu diesem Zeitpunkt wird die konsolidierte Datenbank nicht geändert und die entfernte Datenbank nicht erstellt oder geändert.

6. Ordnen Sie die neuen entfernten Tabellen auf der Registerkarte **Zuordnungen** zu.

Ergebnisse

Das Schema wird aktualisiert.

Siehe auch

- „Tabellen- und Spaltenzuordnungen“ auf Seite 34

Deployment des Synchronisationsmodells

Mit dem **Assistenten zum Erstellen eines Synchronisationsmodells** können Sie das Deployment von MobiLink-Modellen vornehmen.

Das Deployment kann für folgende Elemente erfolgen:

- Änderungen an der konsolidierten Datenbank
- Entfernte SQL Anywhere- oder UltraLite-Datenbanken. Sie können wählen, ob Sie eine Datenbank erstellen, Tabellen zu einer vorhandenen leeren Datenbank hinzufügen oder eine vorhandene Datenbank verwenden möchten, die bereits über die entfernten Tabellen verfügt.
- Batchdateien für das Deployment des Modells (am Anfang der generierten Batchdateien befinden sich Variablendeklarationen, die Sie vor der Ausführung der Batchdateien bearbeiten können).
- Batchdateien zum Starten des MobiLink-Servers und des MobiLink-Clients

Im MobiLink 16-Plug-In werden Schemata verglichen, wenn Sie ein Deployment eines Synchronisationsmodells vornehmen. Es werden nur Datenbankobjekte (wie z.B. Tabellen, Indizes etc.) geändert, die sich von den Objekten im Deployment unterscheiden. Sie werden gewarnt, wenn das Deployment neuer Objekte das bestehende Synchronisationssystem brechen würde, und Sie erhalten die Möglichkeit zum Abbrechen des Vorgangs. Diese Funktionalität ist für konsolidierte Datenbanken des Typs IBM DB2 LUW nicht verfügbar.

Bei einem erneuten Deployment von Aktualisierungen derselben Skriptversion werden automatisch Schemata entfernt, die von der neuen Version nicht mehr benötigt werden. Beispiel: Wenn eine last_modified-Spalte einer Tabelle zur Unterstützung von Downloads hinzugefügt wird, Sie aber die Synchronisation auf reinen Upload ändern, wird die last_modified-Spalte während des Deployments gelöscht.

Sie können auch mithilfe der ml_model_drop-Systemprozedur ein Synchronisationsmodell und sein Schema aus der konsolidierten Datenbank löschen. Diese Systemprozedur löscht die Synchronisationsskripten und alle Schemata, die beim Deployment des Synchronisationsmodells erstellt wurden, einschließlich Schattentabellen, Tracking-Spalten, Triggern und Indizes. Schemata, die gemeinsam mit einer anderen script_version genutzt werden, werden nicht gelöscht. Es können nur Schemata gelöscht werden, die mit dem MobiLink 16-Plug-In installiert wurden. Diese Funktionalität ist für konsolidierte Datenbanken des Typs IBM DB2 LUW nicht verfügbar. Siehe „ml_model_drop-Systemprozedur“ [*MobiLink - Serveradministration*].

Deployment in der konsolidierten Datenbank

Der **Assistent für das Deployment eines Synchronisationsmodells** bietet zwei Optionen für das Deployment in der konsolidierten Datenbank:

- Wenden Sie das Synchronisationsmodell direkt in der konsolidierten Datenbank an, indem Sie die MobiLink-Systemtabellen füllen und alle erforderlichen Schattentabellen, Spalten, Trigger und gespeicherten Prozeduren erstellen.
- Erstellen Sie eine in UTF-8 kodierte SQL-Datei, die alle Änderungen enthält und führen Sie die SQL-Datei in einer konsolidierten Datenbank aus. Sie können diese Datei jederzeit untersuchen, ändern und ausführen. Dies führt zu demselben Ergebnis wie die direkte Übernahme der Änderungen.

Synchronisationsmodell vor dem Deployment testen

Verwenden Sie das Fenster **Test**, um ein Synchronisationsmodell zu testen, bevor Sie das Deployment durchführen.

Voraussetzungen

In Ihrem MobiLink-Projekt muss ein Synchronisationsmodell definiert sein.

Kontext und Bemerkungen

Wenn Sie die Testfunktion verwenden, werden Änderungen an der konsolidierten Datenbank vorgenommen.

Aufgabe

1. Doppelklicken Sie auf den MobiLink-Projektnamen.
2. Doppelklicken Sie auf **Synchronisationsmodelle** und doppelklicken Sie auf den Namen Ihres Synchronisationsmodells.
3. Klicken Sie im Fensterausschnitt **Deployment** auf **Test**. Sie werden gewarnt, dass der Test des Synchronisationsmodells Änderungen in der konsolidierten Datenbank bewirkt und die Daten in Tabellen ändert. Klicken Sie auf **OK**.
4. Klicken Sie auf **Synchronisieren**.

Ergebnisse

Das Synchronisationsmodell wird getestet und Sie können sehen, ob die Synchronisation erfolgreich war.

Nächste Schritte

Sie können die folgenden Aufgaben durchführen, wenn Ihr Test abgeschlossen ist:

- Die Informationen über die Registerkarte **Daten** überprüfen, um die Daten in der konsolidierten und der entfernten Datenbank zu vergleichen.

- Die Informationen über die Registerkarte **Client-Log** auf Informationen über Fehler prüfen, die während des Tests aufgetreten sind. Diese ist nur für SQL Anywhere-Clients verfügbar.
- Die Informationen über die Registerkarte **MobiLink-Log** auf Informationen über Fehler prüfen, die während des Tests aufgetreten sind.
- Ändern Sie die entfernte und/oder die konsolidierte Datenbank vor der Synchronisation erneut. Es gibt mehrere Optionen zum Ändern der Datenbanken:
 - Verwenden Sie die Registerkarte **Daten**, um Daten direkt in den zu synchronisierenden Tabellen zu ändern.
 - Verwenden Sie das Menü **Aktionen**, um das Dienstprogramm "dbisql" für eine Datenbank zu öffnen.

Deployment einer konsolidierten Datenbank aus einer SQL-Datei

Verwenden Sie den **Assistenten für das Deployment eines Synchronisationsmodells**, um das Deployment einer konsolidierten Datenbank durchzuführen.

Voraussetzungen

Wenn beim Deployment Schattentabellen entstehen, müssen Sie als Eigentümer oder Administrator der Basistabellen, für die die Schattentabellen erstellt werden, eine Verbindung mit der konsolidierten Datenbank herstellen.

Aufgabe

- Wenn Sie den **Assistenten für das Deployment eines Synchronisationsmodells** ausgeführt haben und (auf der Seite **Deployment-Ziel der konsolidierten Datenbank**) festgelegt haben, dass Sie eine Datei zur späteren Ausführung erstellen möchten, führen Sie die Batchdatei aus, die sich im Unterordner *project-name_deploy* Ihres Modells befindet. Diese Datei erstellt alle Objekte, die in der konsolidierten Datenbank erstellt werden sollen, einschließlich Synchronisationsskripten, Schattentabellen und Trigger. Mit ihr können MobiLink-Benutzer auch in der konsolidierten Datenbank registriert werden.

Um diese Datei auszuführen, müssen Sie zum Verzeichnis *project-name_deploy* gehen und die Datei *cons_setup.bat* oder *cons_setup.sh* ausführen. Sie müssen Verbindungsinformationen angeben. Zum Beispiel:

```
cons_setup.bat "DSN=my_odbc_datasource;UID=myuserid;PWD=mypassword"
```

Bei einigen Treibern enthält die ODBC-Datenquelle die Benutzer-ID und das Kennwort, sodass sie nicht angegeben werden müssen.

Ergebnisse

Das Deployment der konsolidierten Datenbank wird durchgeführt.

Deployment der entfernten Datenbank

Sie können entscheiden, ob Sie eine bestehende entfernte Datenbank verwenden oder mit dem Assistenten eine entfernte Datenbank erstellen möchten. Der Assistent kann entfernte Datenbanken direkt erstellen oder sie können eine mit UTF-8 kodierte SQL-Datei und eine Batchdatei erstellen lassen, die Sie ausführen können, um entfernte Datenbanken zu erstellen oder zu aktualisieren.

Der Assistent erstellt eine entfernte Datenbank (SQL Anywhere oder UltraLite) mit Standardoptionen zum Erstellen von Datenbanken und dem Datenbank Eigentümer, den Sie im Modell festgelegt haben. Sie können eine entfernte Datenbank auch außerhalb des **Assistenten für das Deployment eines Synchronisationsmodells** mit Ihren eigenen benutzerdefinierten Einstellungen erstellen und mit dem Assistenten die erforderlichen entfernten Tabellen hinzufügen. Sie können auch das Deployment in eine bestehende entfernte Datenbank durchführen, die bereits über die entfernten Tabellen verfügt.

Deployment einer entfernten Datenbank aus einer SQL-Datei durchführen

Verwenden Sie für das Deployment einer entfernten Datenbank den **Assistenten für das Deployment eines Synchronisationsmodells**.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

- Wenn Sie den **Assistenten für das Deployment eines Synchronisationsmodells** ausgeführt und (auf der Seite **Auswählen, wie Datenbanken für die Synchronisation vorbereitet werden**) das Erstellen einer Datei zur späteren Ausführung festgelegt haben, müssen Sie die Batchdatei, die Sie zusammen mit der SQL-Datei erstellt haben, im Verzeichnis *project-name_deploy* ausführen. Diese Datei erstellt alle Objekte, die in der entfernten Datenbank erstellt werden sollen, einschließlich Tabellen, Publikationen, Subskriptionen und MobiLink-Benutzer.

Um diese Datei auszuführen, müssen Sie zum Verzeichnis *project-name_deploy* gehen und die Datei *remote_setup.bat* oder *remote_setup.bat.sh* ausführen. Zum Beispiel:

```
remote_setup.bat
```

Wenn Sie eine vorhandene entfernte Datenbank verwenden, werden Sie aufgefordert, ein Kennwort einzugeben.

Ergebnisse

Die entfernte Datenbank wird per Deployment bereitgestellt.

Deployment einer Batchdatei zum Ausführen von Synchronisationstools

Der Assistent kann die folgenden Batchdateien erstellen:

- Eine Batchdatei zum Ausführen des MobiLink-Servers mit Optionen, die Sie selbst definieren.
- Bei entfernten SQL Anywhere-Datenbanken eine Batchdatei zum Ausführen von dbmlsync mit Optionen, die Sie selbst definieren.
- Bei entfernten UltraLite-Datenbanken eine Batchdatei zum Ausführen von ulsync mit Optionen, die Sie selbst definieren. Ulsync wird zum Testen der Synchronisation verwendet und bietet Unterstützung, wenn Sie nicht über eine funktionsfähige UltraLite-Anwendung verfügen.

Erneutes Deployment eines Synchronisationsmodells

Sie können ein Synchronisationsmodell ändern, nachdem Sie sein Deployment durchgeführt haben, indem Sie Änderungen am Synchronisationsmodell vornehmen und dann ein erneutes Deployment vornehmen. Sie können Ihr Modell auch mithilfe von Systemprozeduren oder mit anderen Methoden ändern. Allerdings können Sie die Änderungen nicht in das Synchronisationsmodell zurückführen, wenn Sie ein bereitgestelltes Modell außerhalb von Sybase Central ändern. Änderungen außerhalb von Sybase Central werden überschrieben, wenn Sie das Deployment des Modells erneut durchführen.

Ein Deployment führt häufig zu Schemaänderungen, sodass eine Aktualisierung des Schemas erforderlich sein kann, auch wenn Sie keine weiteren Änderungen vorgenommen haben. Wenn Sie beispielsweise das Deployment eines Modells durchführen, das eine TIMESTAMP-Spalte zu jeder synchronisierten Tabelle in der konsolidierten Datenbank hinzufügt (was dem Standardverhalten bei neu erstellten Modellen entspricht), müssen Sie vor dem erneuten Deployment das konsolidierte Schema im Modell aktualisieren. Ebenso müssen Sie das konsolidierte Schema im Modell aktualisieren und neue entfernte Tabellen erstellen, wenn Sie eine Tabelle zur konsolidierten Datenbank hinzufügen und das Deployment daraufhin erneut durchführen möchten.

Siehe „Aktualisierung der Schemata“ auf Seite 49.

Per Deployment bereitgestellte Synchronisationsmodelle

Die Synchronisationsmodelldateien befinden sich im MobiLink-Projektverzeichnis. Synchronisationsdateien haben die Dateierweiterung *.mlsm*. Wenn Sie ein Deployment eines Synchronisationsmodells vornehmen, wird ein Verzeichnis angelegt, das mit Ihrem Modellnamen beginnt und mit *_deploy* endet.

Abhängig von den gewählten Deployment-Optionen verfügen Sie dann über folgende Dateien:

Datei	Beschreibung
<i>cons_setup.bat</i>	Eine Batchdatei, die die SQL-Datei zum Einrichten der konsolidierten Datenbank ausführt
<i>cons_setup.sql</i>	Eine SQL-Datei zum Einrichten der konsolidierten Datenbank

Datei	Beschreibung
<i>mlsrv.bat</i>	Eine Batchdatei zum Ausführen des MobiLink-Servers
<i>remote_setup.bat</i>	Eine Batchdatei, die die SQL-Datei zum Einrichten der entfernten Datenbank ausführt
<i>remote_setup.sql</i>	Eine SQL-Datei zum Einrichten der konsolidierten Datenbank
<i>summary.txt</i>	Eine Datei, die die Details des Synchronisationsmodells zusammenfasst
<i>sync.bat</i>	Wenn Sie eine entfernte SQL Anywhere-Datenbank per Deployment bereitgestellt haben, wird diese Batchdatei zum Synchronisieren von SQL Anywhere-Datenbanken mit dbmlsync verwendet.
<i>modelname_remote.db</i>	Wenn Sie eine neue entfernte SQL Anywhere-Datenbank erstellen, ist dies die Datenbankdatei.
<i>modelname_remote.udb</i>	Wenn Sie eine neue entfernte UltraLite-Datenbank einrichten, ist dies die Datenbankdatei.
<i>modelname_ulsync.bat</i>	Wenn Sie das Deployment einer UltraLite-Datenbank durchgeführt haben, ist dies die Batchdatei zum Testen der Synchronisation mit einer entfernten UltraLite-Datenbank mithilfe des Dienstprogramms ulsync.

Batchdateien ausführen

Führen Sie die Batchdateien, die vom **Assistenten für das Deployment eines Synchronisationsmodells** erstellt wurden, in der Befehlszeile aus, um Ihr Synchronisationsmodell zu synchronisieren.

Voraussetzungen

Sie müssen über ein Synchronisationsmodell verfügen.

Kontext und Bemerkungen

Wenn Sie noch keine MobiLink-Setupskripten in der konsolidierten Datenbank ausgeführt haben, führen Sie sie vor dem Deployment aus. Siehe „[Einrichten einer konsolidierten Datenbank](#)“ [*MobiLink - Serveradministration*].

Um viele der Batchdateien auszuführen, müssen Sie die Verbindungsinformationen angeben. Möglicherweise müssen Sie auch ODBC-Datenquellen erstellen, bevor Sie diese Batchdateien ausführen können. Siehe „[ODBC-Datenquellen](#)“ [*SQL Anywhere Server - Datenbankadministration*].

Aufgabe

1. Wenn Sie den **Assistenten für das Deployment eines Synchronisationsmodells** ausgeführt und (auf der Seite **Auswählen, wie Datenbanken für die Synchronisation vorbereitet werden**) das Erstellen einer Datei zur späteren Ausführung festgelegt haben, müssen Sie die Batchdatei im Unterordner *project-name_deploy* Ihres Modells ausführen. Diese Datei erstellt alle Objekte, die in der konsolidierten Datenbank erstellt werden sollen, einschließlich Synchronisationsskripten, Schattentabellen und Trigger. Mit ihr können MobiLink-Benutzer auch in der konsolidierten Datenbank registriert werden.

Um diese Datei auszuführen, gehen Sie zum Verzeichnis *project-name_deploy* und führen die Datei *cons_setup.bat* oder *cons_setup.sh* aus. Sie müssen Verbindungsinformationen in der Befehlszeile angeben. Zum Beispiel:

```
cons_setup.bat "DSN=MY_ODBC_DATASOURCE"
```

2. Wenn Sie den **Assistenten für das Deployment eines Synchronisationsmodells** ausgeführt und (auf der Seite **Auswählen, wie Datenbanken für die Synchronisation vorbereitet werden**) das Erstellen einer Datei zur späteren Ausführung festgelegt haben, müssen Sie die Batchdatei im Verzeichnis *project-name_deploy* ausführen. Diese Datei erstellt alle Objekte, die in der entfernten Datenbank erstellt werden sollen, einschließlich Tabellen, Publikationen, Subskriptionen und MobiLink-Benutzer.

Um diese Datei auszuführen, müssen Sie zum Verzeichnis *project-name_deploy* gehen und die Datei *remote_setup.bat* oder *remote_setup.sh* ausführen. Zum Beispiel:

```
remote_setup.bat
```

Wenn Sie eine vorhandene entfernte Datenbank verwenden, werden Sie aufgefordert, ein Kennwort einzugeben.

3. Starten Sie den MobiLink-Server, indem Sie *mlsrv.bat* ausführen. Sie müssen Verbindungsinformationen für die konsolidierte Datenbank in der Befehlszeile angeben. Zum Beispiel:

```
mlsrv.bat "DSN=MY_ODBC_DATASOURCE"
```

4. Führen Sie die Synchronisation durch.

Bei entfernten SQL Anywhere-Datenbanken:

- Erteilen Sie die SYS_RUN_REPLICATION_ROLE-Systemrolle einem Benutzer, damit dieser in der Lage ist, dbmlsync auszuführen. Führen Sie beispielsweise Folgendes in Interactive SQL aus:

```
GRANT ROLE SYS_RUN_REPLICATION_ROLE  
TO userid
```

- Verbinden Sie sich unter dem Namen dieses Benutzers mit der Datenbank.
- Starten Sie die entfernte Datenbank, die sich im Verzeichnis *project-name_deploy* befindet. Zum Beispiel:

```
dbsrv16 MyModel_remote.db
```

- Starten Sie dbmlsync, den SQL Anywhere MobiLink-Client. Führen Sie die Datei *sync.bat* im Verzeichnis *project-name_deploy* aus. Sie müssen Verbindungsinformationen in der Befehlszeile angeben. Zum Beispiel:

```
sync.bat "UID=userid;PWD=password;SERVER=MyModel_remote"
```

Bei entfernten UltraLite-Datenbanken:

- Um die Synchronisation zu testen, führen Sie die Datei mit der Endung *_ulsync.bat* im Verzeichnis *remote* aus.
- Alternativ dazu können Sie auch Ihre UltraLite-Anwendung ausführen.

Ergebnisse

Das Synchronisationsmodell wird per Deployment bereitgestellt.

Siehe auch

- „GRANT ROLE SYS_RUN_REPLICATION_ROLE-Anweisung [MobiLink] [SQL Remote]“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]
- „GRANT ROLE SYS_REPLICATION_ADMIN_ROLE-Anweisung [MobiLink] [SQL Remote]“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]
- Privilegien für dbmlsync [*MobiLink - Clientadministration*]

Einschränkungen von Synchronisationsmodellen

Im Folgenden werden einige Einschränkungen der Synchronisationsmodelle beschrieben:

- **Kein erneutes Deployment für Änderungen außerhalb des Modells möglich** Wenn Sie ein Deployment für ein Synchronisationsmodell vornehmen und es dann außerhalb des Modells ändern, werden diese Änderungen nicht im Modell gespeichert. Diese Vorgehensweise stellt kein Problem dar, wenn Sie das Modell als Ausgangspunkt verwenden, das Deployment durchführen und dann alle Änderungen außerhalb des Modells vornehmen. Um jedoch ein erneutes Deployment des Modells vorzunehmen, sollten Sie die Änderungen in Ihrem MobiLink-Projekt durchführen, sodass sie gespeichert werden und ein erneutes Deployment möglich ist.
- **Versionen** Ein Synchronisationsmodell kann nur eine Version haben. Siehe „Skriptversionen“ [*MobiLink - Serveradministration*].
- **MobiLink-Systemdatenbank** Sie können keine MobiLink-Systemdatenbank verwenden, die von der konsolidierten Datenbank getrennt wurde, wenn Sie ein Deployment eines Synchronisationsmodells vornehmen. Siehe „MobiLink-Systemdatenbank“ [*MobiLink - Serveradministration*].
- **Mehrere Publikationen** Es ist nicht möglich, mehrere Publikationen zu erstellen. Nachdem Sie ein Deployment des Modells vorgenommen haben, können Sie mithilfe von Nicht-Modell-Methoden weitere Publikationen hinzufügen, z.B. die CREATE PUBLICATION-Anweisung. Es ist jedoch nicht möglich, diese Erweiterungen in das Modell einzufügen. Siehe „Publikationen“ [*MobiLink - Clientadministration*].

- **Views** Es ist nicht möglich, eine Ansicht auszuwählen, wenn Sie konsolidierte Datenbanktabellen für Tabellenzuordnungen auswählen.
- **Berechnete Spalten** Sie können keinen Upload in berechnete Spalten in einer konsolidierten Datenbanktabelle vornehmen. Während des Deployments für ein Synchronisationsmodells mit berechneten Spalten kommt es beim Erstellen der Trigger, die für zeitstempelbasierte Downloads verwendet werden, möglicherweise zu Fehlern. Sie können die Spalte entweder aus der Synchronisation ausschließen oder die Tabelle als reine Download-Tabelle konfigurieren (und entweder den Snapshot-Download verwenden oder die generierte konsolidierte SQL-Datei bearbeiten, um die berechnete Spalte aus der Triggerdefinition zu entfernen).

Das Kopieren berechneter Spalten verursacht beim Deployment des neuen entfernten Schemas zur Erstellung einer neuen entfernten Datenbank einen Syntaxfehler. Bei der Verwendung berechneter Spalten ist einer der folgenden Schritte durchzuführen:

- Führen Sie das Deployment des Synchronisationsmodells in eine vorhandene entfernte Datenbank durch.
- Schließen Sie die berechnete Spalte aus dem entfernten Schema aus. Um eine konsolidierte Datenbanktabelle mit berechneten Spalten zu synchronisieren, können Sie kein Upload in die Tabelle durchführen.

Die Microsoft SQL Server-Beispieldatenbank AdventureWorks enthält berechnete Spalten. Legen Sie die Spalten als reine Downloadspalten fest oder schließen Sie die Spalten von der Synchronisation aus, wenn Sie diese Datenbank verwenden, um ein Modell zu erstellen.

- **Datentyp Oracle XMLTYPE** Das MobiLink-Plug-In für Sybase Central bietet die folgende Unterstützung für den Datentyp Oracle XMLTYPE.
 - Wenn eine XMLTYPE-Spalte für ein neues entferntes Schema oder beim Bearbeiten von Spaltenzuordnungen gewählt wird, lautet der Name des angezeigten **Datentyps XMLTYPE**.
 - XMLTYPE-Spalten werden in neuen entfernten SQL Anywhere-Datenbanken Spalten vom Typ XML zugeordnet. Siehe „Oracle-Datentypzuordnung“ [[MobiLink - Serveradministration](#)].
 - Die erzeugten Synchronisationsskripten funktionieren, wenn XMLTYPE-Spaltenwerte eine Größe von 4 kB oder weniger haben. Bei größeren Werten müssen Sie die generierten Skripten im Synchronisationsmodell mit Skripten überschreiben, indem Sie die Methoden verwenden, die in [XMLTYPE-Datentyp in Oracle](#) [[MobiLink - Serveradministration](#)] beschrieben werden.

Deployment-Hinweise

- **Spalten mit räumlichen Daten** Spalten mit räumlichen Daten werden kopiert, aber der räumliche Subtyp und die SRID werden möglicherweise nicht kopiert, wenn das konsolidierte RDBMS keine Metadatenunterstützung für ihr Abrufen hat, beispielsweise die ST_GEOMETRY_COLUMNS-Ansicht des SQL/MM-Standards. Die räumliche Unterstützung in UltraLite ist auf einen Typ beschränkt (ST_GEOMETRY), der nur Punktwerte und Spalten-SRID-Integritätsregeln von SRID=0 oder SRID=4326 unterstützt. Daher erhalten Sie vielleicht eine Warnung oder eine Fehlermeldung, wenn Sie ein Deployment eines nicht kompatiblen räumlichen Datentyps in einer neuen UltraLite-Datenbank vornehmen.
- **Lange Objektnamen** Die Namen der beim Deployment erstellten Datenbankobjekte können länger als die von der Datenbank unterstützte Länge sein (da die neuen Objektnamen durch das

Anfügen von Suffixen an die Namen der Basistabelle erstellt werden). In diesem Fall nehmen Sie das Deployment in eine Datei vor (nicht direkt in eine Datenbank) und bearbeiten die generierte SQL-Datei, um alle zu langen Namen zu ersetzen.

- **Neue entfernte Schemata** Wenn Sie mit dem **Assistenten zum Erstellen eines Synchronisationsmodells** ein neues entferntes Schema erstellen, enthalten die Spalten der neuen entfernten Datenbank keine Indizes für die Spalten in der konsolidierten Datenbank. Fremdschlüssel und Standardspaltenwerte werden in die neue entfernte Datenbank kopiert. Da diese Unterstützung jedoch auf Datenbankmetadaten basiert, die vom ODBC-Treiber zurückgegeben werden, können eventuelle Treiberprobleme zu Syntax- oder anderen Fehlern führen. Wenn zum Beispiel ein Treiber einen Standardspaltenwert in einem Format angibt, mit dem ein solcher Standardwert nicht in einer entfernten SQL Anywhere- oder UltraLite-Datenbank deklariert werden kann, können Fehler auftreten (darunter auch Syntaxfehler beim Deployment).

UltraLite unterstützt die Spaltentypen NCHAR(n), NVARCHAR(n) oder LONG NVARCHAR nicht. Wenn Sie das Deployment eines Synchronisationsmodells in einer neuen UltraLite-Datenbank vornehmen, werden solche Spalten im entfernten Schema in CHAR(4n), VARCHAR(4n) oder LONG VARCHAR umgewandelt. Wenn 4n größer ist als die maximale Länge für CHAR und VARCHAR, wird die maximale Länge verwendet und Sie erhalten eine Warnung.

Sie können eine bestehende entfernte Datenbank verwenden, um ein Synchronisationsmodell zu erstellen oder ein entferntes Schema in einem Modell zu aktualisieren.

- **Proxytabellen** Es ist möglich, eine Synchronisation mit konsolidierten Datenbanktabellen durchzuführen, die Proxy-Tabellen für eine andere Datenbank sind, aber Sie müssen die TIMESTAMP-Spalte sowohl der Basistabelle als auch der Proxy-Tabelle hinzufügen, wenn Sie eine TIMESTAMP-Spalte für zeitstempelbasierte Downloads verwenden. Der **Assistent für das Deployment eines Synchronisationsmodells** kann keine Spalte zu einer Proxy-Tabelle oder deren Basistabelle hinzufügen. Daher müssen Sie entweder eine Spalte verwenden, die sowohl in der Basistabelle als auch in der Proxy-Tabelle vorhanden ist, oder eine Schattentabelle bzw. einen Snapshot-Download benutzen.
- **Materialisierte Ansichten** Wenn Sie zeitstempelbasierte Downloads verwenden und gewählt haben, konsolidierten Tabellen eine TIMESTAMP-Spalte hinzuzufügen, müssen Sie vor dem Deployment alle materialisierten Ansichten deaktivieren, die von den Tabellen abhängen. Andernfalls werden möglicherweise Fehler ausgegeben, wenn versucht wird, die Tabellen zu ändern. Verwenden Sie bei konsolidierten SQL Anywhere-Datenbanken die Systemprozedur `sa_dependent_views`, um festzustellen, ob für eine Tabelle abhängige materialisierte Ansichten vorhanden sind. Siehe „[sa_dependent_views-Systemprozedur](#)“ [*SQL Anywhere Server - SQL-Referenzhandbuch*].

Sonstige Hinweise

- **Entfernte Datenbanken basierend auf einer konsolidierten Oracle-Datenbank erstellen** Wenn Sie eine konsolidierte Oracle-Datenbank als Basis für die entfernte SQL Anywhere- oder UltraLite-Datenbank verwenden, kann es sinnvoll sein, DATE-Spalten in der konsolidierten Datenbank in TIMESTAMP zu ändern. Andernfalls gehen beim Upload Informationen zu den Sekundenbruchteilen verloren.

MobiLink-Beispiel CustDB

CustDB ist eine Anwendung zum Status von Kundenbestellungen. Die CustDB-Beispieldatenbank ist eine wertvolle Quelle für den MobiLink-Entwickler. Sie enthält Beispiele für die Implementierung vieler Funktionen, die für die Entwicklung von MobiLink-Anwendungen erforderlich sind.

Diese Anwendung dient dazu, einige allgemeine Synchronisationsmethoden vorzustellen. Um den größtmöglichen Nutzen aus diesem Abschnitt zu ziehen, überprüfen Sie beim Lesen die Beispieldatenbank.

Für jedes Betriebssystem und für jeden unterstützten Datenbanktyp wird eine Version von CustDB bereitgestellt.

Ein MobiLink-Projekt, das die konsolidierte CustDB-Datenbank verwendet, wird im Verzeichnis `%SQLANYSAMPI6%\MobiLink\CustDB\Projekt.mlp` zur Verfügung gestellt. Sie können dieses Projekt zum Arbeiten mit CustDB-Projekten und zum Anzeigen von Datenbankskripten in Sybase Central öffnen.

CustDB-Szenario

In der Hauptniederlassung befindet sich eine konsolidierte Datenbank. Die folgenden Daten sind in der konsolidierten Datenbank gespeichert:

- Die MobiLink-Systemtabellen, die die Metadaten für die Synchronisation einschließlich der Synchronisationsskripten, die die Synchronisationslogik implementieren, enthalten.
- Die CustDB-Daten, einschließlich aller Kunden-, Produkt- und Bestelldaten, sind in den Zeilen von Basistabellen gespeichert.

Es gibt zwei Typen von entfernten Datenbanken, nämlich für nicht ortsgebundene Manager und für Vertriebsmitarbeiter.

Jede Datenbank eines nicht ortsgebundenen Vertriebsmitarbeiters enthält alle Produkte, aber nur die Bestellungen, die dem betreffenden Vertriebsmitarbeiter zugeordnet sind, während die Datenbank eines nicht ortsgebundenen Managers alle Produkte und alle Bestellungen für eine bestimmte Gruppe von Kunden enthält.

Synchronisationsplanung

Die Synchronisationsplanung in der Beispieldatenbank CustDB verwendet folgende Funktionen.

- **Vollständiger Download von Tabellen** Alle Zeilen und Spalten der Tabelle ULProduct werden mit den entfernten Datenbanken gemeinsam genutzt.
- **Spaltenteilmengen** Alle Zeilen, nicht jedoch alle Spalten der Tabelle ULCustomer werden mit den entfernten Datenbanken gemeinsam genutzt.
- **Zeilenteilmengen** Verschiedene entfernte Benutzer erhalten unterschiedliche Teilmengen der Zeilen aus der Tabelle ULOrder.

Weitere Hinweise zu Teilmengen von Zeilen finden Sie unter „[Partitionierte Zeilen in entfernten Datenbanken](#)“ [[MobiLink - Serveradministration](#)].

- **Synchronisation auf Zeitstempelbasis** Dies ist eine Möglichkeit, die in der konsolidierten Datenbank seit der letzten Synchronisation des Geräts vorgenommenen Änderungen festzustellen. Die Tabellen ULCustomer und ULOrder werden mit einer Methode synchronisiert, die auf Zeitstempeln basiert.

Siehe „Implementieren zeitstempelbasierter Downloads“ [[MobiLink - Serveradministration](#)].

- **Snapshot-Synchronisation** Dies ist eine einfache Synchronisationsmethode, die bei allen Synchronisationen jeweils alle Zeilen herunterlädt. Die Tabelle ULProduct wird auf diese Weise synchronisiert.

Siehe „Snapshot-Synchronisation“ [[MobiLink - Serveradministration](#)].

- **Primärschlüsselpools zur Aufrechterhaltung eindeutiger Primärschlüssel** Es ist von entscheidender Bedeutung, dass die Primärschlüsselwerte innerhalb der gesamten MobiLink-Installation eindeutig sind. Die in dieser Anwendung benutzte Methode des Primärschlüsselpools ist eine von mehreren Möglichkeiten, mit denen Sie sicherstellen können, dass die Primärschlüssel eindeutig sind.

Siehe „Primärschlüsselpools“ [[MobiLink - Serveradministration](#)].

Weitere Möglichkeiten für die Sicherstellung der Eindeutigkeit von Primärschlüsseln finden Sie unter „Eindeutige Primärschlüssel“ [[MobiLink - Serveradministration](#)].

Siehe auch

- „Praktische Einführung: Erstellen der Beispielanwendung CustDB“ [[UltraLite - Datenbankverwaltung](#)]
- „Praktische Einführung: Erstellen der Beispielanwendung CustDB“ [[UltraLite - Datenbankverwaltung](#)]
- „Die Beispieldatenbankanwendung CustDB“ [[SQL Anywhere 16 - Einführung](#)]
- „Eine konsolidierte CustDB-Datenbank einrichten“ auf Seite 61

CustDB-Dateien

In diesem Abschnitt werden die verschiedenen Teile beschrieben, aus denen sich der Code der Beispielanwendung und der Beispieldatenbank zusammensetzt. Es handelt sich dabei um die Folgenden:

- Die SQL-Beispielskripten befinden sich im Verzeichnis `%SQLANYSAMPI6%\MobiLink\CustDB`.
- Der Anwendungscode befindet sich im Verzeichnis `%SQLANYSAMPI6%\UltraLite\CustDB`.
- Der plattformspezifische Code für die Benutzeroberfläche in den jeweiligen Unterverzeichnissen von `%SQLANYSAMPI6%\UltraLite\CustDB` für die einzelnen Betriebssysteme.

Eine konsolidierte CustDB-Datenbank einrichten

Die konsolidierte CustDB-Datenbank kann eine beliebige unterstützte konsolidierte MobiLink-Datenbank sein.

SQL Anywhere 16 CustDB

Eine konsolidierte Datenbank SQL Anywhere 16 CustDB wird in `%SQLANYAMP16%\UltraLite\CustDB\custdb.db` bereitgestellt. Eine Datenquelle (DSN) namens SQL Anywhere 16 CustDB wird mit Ihrer Installation eingerichtet.

Sie können diese Datenbank mithilfe der Datei `%SQLANYAMP16%\UltraLite\CustDB\makedbs.cmd` neu aufbauen.

Wenn Sie herausfinden möchten, wie das CustDB-Beispiel erstellt wird, können Sie die Datei `%SQLANYAMP16%\MobiLink\CustDB\custdb.sql` anzeigen.

CustDB für andere RDBMS-Systeme

Die folgenden SQL-Skripten werden im Verzeichnis `%SQLANYAMP16%\MobiLink\CustDB` bereitgestellt, um die konsolidierte Datenbank CustDB auf folgenden unterstützten RDBMS erstellen zu können:

RDBMS	Custdb-Setupskript
Adaptive Server Enterprise	<i>custase.sql</i>
SQL Server	<i>custmss.sql</i>
Oracle	<i>custora.sql</i>
DB2 LUW	<i>custdb2.sql</i>
MySQL	<i>custmys.sql</i>

Weitere Hinweise zur Vorbereitung einer Datenbank als konsolidierte Datenbank finden Sie unter „[Einrichten einer konsolidierten Datenbank](#)“ [*MobiLink - Serveradministration*].

Eine konsolidierte Datenbank (Adaptive Server Enterprise, MySQL, Oracle, SQL Server) aufbauen

Die folgende Prozedur erstellt für jedes der unterstützten RDBMS eine konsolidierte CustDB-Datenbank.

Voraussetzungen

Sie müssen Zugriff auf die SQL-Skripten haben, die verwendet werden, um die konsolidierte Datenbank CustDB als eines der unterstützten RDBMS aufzubauen. Die SQL-Skripten befinden sich im Verzeichnis `%SQLANYAMP16%\MobiLink\CustDB`.

Aufgabe

1. Erstellen Sie eine Datenbank in Ihrem RDBMS.
2. Fügen Sie die MobiLink-Systemobjekte hinzu, indem Sie eines der folgenden SQL-Skripten ausführen, die sich im Unterverzeichnis `MobiLink\setup` Ihrer SQL Anywhere 16-Installation befinden:

- Für eine konsolidierte Adaptive Server Enterprise-Datenbank führen Sie *syncase.sql* aus.
 - Für eine konsolidierte MySQL-Datenbank führen Sie *syncmys.sql* aus.
 - Für eine konsolidierte Oracle-Datenbank führen Sie *syncora.sql* aus.
 - Für eine konsolidierte SQL Server-Datenbank führen Sie *syncmss.sql* aus.
3. Fügen Sie Beispielbenutzertabellen, gespeicherte Prozeduren und MobiLink-Synchronisationsskripten der CustDB-Datenbank hinzu, indem Sie eines der folgenden SQL-Skripten ausführen, die in *%SQLANYAMP16%\MobiLink\CustDB* enthalten sind:
- Für eine konsolidierte Adaptive Server Enterprise-Datenbank führen Sie *custase.sql* aus.
 - Für eine konsolidierte MySQL-Datenbank führen Sie *custmys.sql* aus.
 - Für eine konsolidierte Oracle-Datenbank führen Sie *custora.sql* aus.
 - Für eine konsolidierte SQL Server-Datenbank führen Sie *custmss.sql* aus.
4. Erstellen Sie eine ODBC-Datenquelle mit der Bezeichnung CustDB, die auf Ihre Datenbank auf dem Clientcomputer verweist.
- a. Wählen Sie **Start » Programme » SQL Anywhere 16 » Administrationstools » ODBC-Datenquellen-Administrator** (32 oder 64 Bit).
 - b. Klicken Sie auf **Hinzufügen**.
 - c. Wählen Sie den betreffenden Treiber aus der Liste aus.
Klicken Sie auf **Fertig stellen**.
 - d. Geben Sie der ODBC-Datenquelle den Namen CustDB.
 - e. Klicken Sie auf die Registerkarte **Login**. Geben Sie die **Benutzer-ID** und das **Kennwort** für Ihre Datenbank ein.
5. Klicken Sie auf **OK** und dann nochmals auf **OK**.

Ergebnisse

Eine konsolidierte CustDB-Datenbank wird für das ausgewählte RDBMS erstellt.

CustDB als konsolidierte Datenbank (DB2 LUW) aufbauen

Verwenden Sie diese Prozedur, um eine konsolidierte DB2 LUW-Datenbank einzurichten.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Erstellen Sie auf dem DB2 LUW-Server namens **CustDB** eine konsolidierte Datenbank.
2. Stellen Sie sicher, dass für den Standard-Tablespace (normalerweise mit USERSPACE1 bezeichnet) Seiten von je 8 kB verwendet werden.

Falls der Standard-Tablespace keine 8-kB-Seiten verwendet, führen Sie folgende Schritte aus:

- a. Überprüfen Sie anhand mindestens eines Ihrer Pufferpools, ob 8-kB-Seiten verwendet werden. Wenn nicht, erstellen Sie einen Pufferpool mit 8-kB-Seiten.

- b. Erstellen Sie einen neuen Tablespace und temporären Tablespace mit 8-kB-Seiten.

Weitere Hinweise finden Sie in Ihrer DB2 LUW-Dokumentation.

3. Fügen Sie die MobiLink-Systemobjekte mithilfe der Datei *MobiLink\setup\syncdb2.sql* zur konsolidierten DB2 LUW-Datenbank hinzu:

- a. Ändern Sie den Verbindungsbefehl oben in der Datei *syncdb2.sql*. Ersetzen Sie *DB2Database* durch den Namen Ihrer DB2-Datenbank oder ihres Alias. In diesem Fall wird die Datenbank CustDB genannt. Sie können auch Ihren DB2-Benutzernamen und das Kennwort wie folgt hinzufügen:

```
connect to CustDB user userid using password ~
```

- b. Öffnen Sie auf dem Server oder dem Clientcomputer ein DB2 LUW-Befehlsfenster. Führen Sie *syncdb2.sql* aus, indem Sie folgenden Befehl eingeben:

```
db2cmd db2 -c -ec -td~ +s -v -f syncdb2.sql
```

4. Hinzufügen von Datentabellen, gespeicherten Prozeduren und MobiLink-Synchronisationsskripten zu CustDB-Datenbank:

- a. Falls nötig, ändern Sie den Verbindungsbefehl in *custdb2.sql*. Sie können z.B. den Benutzernamen und das Kennwort wie folgt hinzufügen. Ersetzen Sie *userid* und *password* durch Ihren Benutzernamen und Ihr Kennwort.

```
connect to CustDB user userid using password
```

- b. Öffnen Sie auf dem Server oder dem Clientcomputer ein DB2-Befehlsfenster.

- c. Führen Sie *custdb2.sql* aus, indem Sie folgenden Befehl eingeben:

```
db2cmd db2 -c -ec -td~ +s -v -f custdb2.sql
```

- d. Wenn die Verarbeitung abgeschlossen ist, geben Sie den folgenden Befehl ein, um das Befehlsfenster wieder zu schließen.

```
exit
```

5. Erstellen Sie auf dem DB2 LUW-Client eine ODBC-Datenquelle mit der Bezeichnung "CustDB", die auf die DB2 LUW-Datenbank verweist.

- a. Starten Sie den ODBC-Datenquellen-Administrator:

Wählen Sie **Start » Programme » SQL Anywhere 16 » Administrationstools » ODBC-Datenquellen-Administrator**.

Der ODBC-Datenquellen-Administrator wird aufgerufen.

- b. Klicken Sie auf der Registerkarte **Benutzer-DSN** auf **Hinzufügen**.

- c. Im Fenster **Neue Datenquelle erstellen** wählen Sie den ODBC-Treiber für Ihre DB2 LUW-Datenbank. Wählen Sie beispielsweise den IBM DB2 UDB ODBC-Treiber. Klicken Sie auf **Fertig stellen**.

Weitere Hinweise dazu, wie Sie den ODBC-Treiber konfigurieren, finden Sie an folgenden Stellen:

- DB2 LUW-Dokumentation
- <http://www.sybase.com/detail?id=1011880>

Ergebnisse

Die konsolidierte DB2 LUW-Datenbank wird eingerichtet.

Siehe auch

- „Konsolidierte IBM DB2 LUW-Datenbank“ [*MobiLink - Serveradministration*]

Entfernte UltraLite-Datenbank einrichten

Das folgende Beispiel erstellt eine entfernte Datenbank für CustDB. Die entfernte Datenbank CustDB muss eine UltraLite-Datenbank sein.

Die Anwendungslogik für die entfernte Datenbank befindet sich in `%SQLANYSAMPI6%\UltraLite\CustDB`. Sie umfasst folgende Dateien:

- **Embedded SQL-Logik** Die Datei *custdb.sqc* enthält die SQL-Anweisungen zum Abfragen und Ändern von Daten aus der UltraLite-Datenbank sowie die Aufrufe zum Starten der Synchronisation mit der konsolidierten Datenbank.
- **C++ API-Logik** Die Datei *custdbcpp.cpp* enthält die C++ API-Logik.
- **Funktionen für die Benutzeroberflächen** Diese Funktionen sind separat in plattformspezifischen Unterverzeichnissen von *Beispielverzeichnis\UltraLite\CustDB* gespeichert.

Um die Beispielanwendung auf einem entfernten Gerät zu installieren, auf dem UltraLite läuft, führen Sie folgende Schritte aus:

1. Starten Sie die konsolidierte Datenbank.
2. Starten Sie den MobiLink-Server.
3. Installieren und starten Sie die Beispielanwendung auf dem Clientgerät.
4. Synchronisieren Sie die Beispielanwendung.

Beispiel

Das folgende Beispiel veranschaulicht, wie Sie die Beispieldatenbank CustDB auf einem Windows-Computer installieren und als konsolidierte Datenbank DB2 verwenden.

1. Prüfen Sie, ob die konsolidierte Datenbank aktiv ist.

Öffnen Sie für eine DB2 LUW-Datenbank ein DB2-Befehlsfenster. Geben Sie Folgendes ein, wobei *userid* und *password* der Benutzername und das Kennwort für die Verbindung mit der DB2 LUW-Datenbank sind:

```
db2 connect to CustDB user userid using password
```

2. Starten Sie den MobiLink-Server.

Für eine DB2 LUW-Datenbank führen Sie folgenden Befehl in der Eingabeaufforderung aus:

```
mlsrv16 -c "DSN=CustDB" -zp
```

3. Starten Sie die CustDB-Beispielanwendung:

- a. Klicken Sie auf **Start » Programme » SQL Anywhere 16 » MobiLink » Synchronisationsserver-Beispiel**.
- b. Geben Sie den Wert 50 als Mitarbeiter-ID ein und klicken Sie auf **OK**.

Die Anwendung führt automatisch eine Synchronisation durch und aus der konsolidierten CustDB-Datenbank wird eine Gruppe von Mitarbeitern, Produkten und Bestellungen in die Anwendung heruntergeladen.

4. Synchronisieren Sie die entfernte Anwendung mit der konsolidierten Datenbank.

Wählen Sie im Menü **File (Datei)** die Option **Synchronize (Synchronisieren)**.

Dieser Schritt muss nur ausgeführt werden, wenn die Datenbank verändert wurde.

Tabellen in den CustDB-Datenbanken

Die Tabellendefinitionen für die Datenbank CustDB befinden sich in plattformspezifischen Dateien im Unterverzeichnis `%SQLANYSAMPI6%\MobiLink\CustDB`.

Ein Entity-Relationship-Diagramm der CustDB-Tabellen finden Sie unter „[Die Beispieldatenbankanwendung CustDB](#)“ [[SQL Anywhere 16 - Einführung](#)].

Sowohl die konsolidierte als auch die entfernte Datenbank enthält die folgenden fünf Tabellen. Die Definitionen unterscheiden sich jedoch jeweils geringfügig voneinander.

ULCustomer

Die Tabelle ULCustomer enthält eine Liste von Kunden.

In der entfernten Datenbank besitzt ULCustomer folgende Spalten:

- **cust_id** Eine Primärschlüsselspalte mit einer eindeutigen Ganzzahlkennung für den Kunden
- **cust_name** Eine Zeichenfolge von 30 Zeichen Länge, die den Namen des Kunden enthält

In der konsolidierten Datenbank besitzt ULCustomer folgende zusätzliche Spalte:

- **last_modified** Einen Zeitstempel mit dem Zeitpunkt der letzten Änderung der Zeile. Diese Spalte wird für die Synchronisation auf Zeitstempelbasis benutzt.

ULProduct

Die Tabelle ULProduct enthält eine Liste von Produkten.

In der entfernten und der konsolidierten Datenbank enthält die Tabelle ULProduct außerdem folgende Spalten:

- **prod_id** Eine Primärschlüsselspalte mit einer eindeutigen Ganzzahlkennung für das Produkt
- **price** Eine Ganzzahl, die den Einheitenpreis angibt
- **prod_name** Eine Zeichenfolge von 30 Zeichen Länge, die den Namen des Produkts enthält

ULOrder

Die Tabelle ULOrder enthält eine Liste mit Bestellungen, einschließlich Details zu dem Kunden, der die Bestellung aufgegeben hat, dem Mitarbeiter, der die Bestellung entgegengenommen hat, und dem bestellten Produkt.

In der entfernten Datenbank besitzt ULOrder folgende Spalten:

- **order_id** Eine Primärschlüsselspalte mit einer eindeutigen Ganzzahlkennung für die Bestellung
- **cust_id** Eine Fremdspalte, die ULCustomer referenziert
- **prod_id** Eine Fremdspalte, die ULProduct referenziert
- **emp_id** Eine Fremdspalte, die ULEmployee referenziert
- **disc** Eine Ganzzahl, die den Rabatt für die Bestellung enthält
- **quant** Eine Ganzzahl, die die Anzahl der bestellten Produkte enthält
- **notes** Eine Zeichenfolge von 50 Zeichen Länge, die Hinweise zur Bestellung enthält
- **status** Eine Zeichenfolge von 20 Zeichen Länge, die den Status der Bestellung enthält

In der konsolidierten Datenbank besitzt ULOrder folgende zusätzliche Spalte:

- **last_modified** Einen Zeitstempel mit dem Zeitpunkt der letzten Änderung der Zeile. Diese Spalte wird für die Synchronisation auf Zeitstempelbasis benutzt.

ULOrderIDPool

Die Tabelle ULOrderIDPool ist ein Primärschlüsselpool für ULOrder.

In der entfernten Datenbank besitzt ULOrderIDPool folgende Spalte:

- **pool_order_id** Eine Primärschlüsselspalte mit einer eindeutigen Ganzzahl, die die Bestell-ID angibt

In der konsolidierten Datenbank besitzt ULOrderIDPool folgende zusätzliche Spalten:

- **pool_emp_id** Eine Ganzzahlspalte, die die Mitarbeiter-ID des Eigentümers der entfernten Datenbank enthält, dem die Bestell-ID zugeordnet ist
- **last_modified** Einen Zeitstempel mit dem Zeitpunkt der letzten Änderung der Zeile.

ULCustomerIDPool

Die Tabelle ULCustomerIDPool ist ein Primärschlüsselpool für ULCustomer.

In der entfernten Datenbank besitzt ULCustomerIDPool folgende Spalte:

- **pool_cust_id** Eine Primärschlüsselspalte mit einer eindeutigen Ganzzahl, die die Kunden-ID angibt

In der konsolidierten Datenbank besitzt ULCustomerIDPool folgende zusätzliche Spalten:

- **pool_emp_id** Eine Ganzzahlspalte mit der Mitarbeiter-ID, die für einen neuen auf einer entfernten Datenbank erstellten Mitarbeiter verwendet wird.
- **last_modified** Einen Zeitstempel mit dem Zeitpunkt der letzten Änderung der Zeile.

Die folgenden Tabellen sind nur in der konsolidierten Datenbank enthalten:

ULIdentifyEmployee_nosync

Die ULIdentifyEmployee_nosync-Tabelle ist sowohl in der konsolidierten Datenbank als auch in der entfernten Datenbank enthalten. Sie hat eine einzige Spalte:

- **emp_id** Diese Primärschlüsselspalte enthält eine Ganzzahl, die für eine Mitarbeiter-ID in der *entfernten* Datenbank steht.

ULEmployee

Die Tabelle ULEmployee ist nur in der konsolidierten Datenbank enthalten. Sie enthält eine Liste von Vertriebsmitarbeitern.

Die Tabelle ULEmployee hat die folgenden Spalten:

- **emp_id** Eine Primärschlüsselspalte mit einer eindeutigen Ganzzahlkennung für den Mitarbeiter
- **emp_name** Eine Zeichenfolge von 30 Zeichen Länge, die den Namen des Mitarbeiters enthält

ULEmpCust

Die Tabelle ULEmpCust steuert, welche Kundenbestellungen übertragen werden. Wenn der Mitarbeiter eine neue Kundenbestellung braucht, erzwingt die Eingabe der Mitarbeiter-ID und der Kunden-ID den Download der Bestellungen des betreffenden Kunden.

- **emp_id** Ein Fremdschlüssel für ULEmployee.emp_id
- **cust_id** Ein Fremdschlüssel für ULCustomer.cust_id. Der Fremdschlüssel umfasst emp_id und cust_id

- **action** Ein Zeichen, das ermittelt, ob ein Mitarbeiterdatensatz aus der entfernten Datenbank gelöscht werden soll. Wenn der Mitarbeiter die Bestellungen eines Kunden nicht mehr benötigt, sollte dieser Wert auf D (löschen) gesetzt werden. Wenn die Bestellungen noch gebraucht werden, sollte der Wert NULL sein.

In diesem Fall muss eine logische Löschung verwendet werden, sodass die konsolidierte Datenbank feststellen kann, welche Zeilen aus der Tabelle ULOrder gelöscht werden sollen. Wenn die Löschungen übertragen wurden, können alle Datensätze für diesen Mitarbeiter mit dem Aktionswert D ebenfalls aus der konsolidierten Datenbank gelöscht werden.

- **last_modified** Einen Zeitstempel mit dem Zeitpunkt der letzten Änderung der Zeile. Diese Spalte wird für die Synchronisation auf Zeitstempelbasis benutzt.

ULOldOrder und ULNewOrder

Diese Tabellen sind nur in der konsolidierten Datenbank vorhanden. Sie dienen zur Konfliktlösung und enthalten dieselben Spalten wie ULOrder. In SQL Anywhere und Microsoft SQL Server handelt es sich dabei um temporäre Tabellen. In Adaptive Server Enterprise sind dies normale Tabellen und @@spid. DB2 LUW und Oracle haben keine temporären Tabellen. MobiLink muss daher feststellen können, welche Zeilen dem synchronisierenden Benutzer gehören. Da es sich um Basistabellen handelt, können fünf synchronisierende Benutzer jeweils eine Zeile in diesen Tabellen haben.

Weitere Hinweise zu @@spid finden Sie unter „Variablen“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)].

Benutzer im Beispiel CustDB

Es gibt zwei Typen von Benutzern im Beispiel CustDB, nämlich Vertriebsmitarbeiter und nicht ortsgebundene Manager. Die Unterschiede sind wie folgt:

- **Vertriebsmitarbeiter** Die Benutzer-IDs 50, 51 und 52 kennzeichnen entfernte Datenbanken, die Vertriebsmitarbeitern zugeordnet sind. Vertriebsmitarbeiter können folgende Aufgaben ausführen:
 - Eine Liste der Kunden und Produkte einsehen
 - Neue Kunden hinzufügen
 - Bestellungen hinzufügen oder löschen
 - Durch die Liste der ausstehenden Bestellungen blättern
 - Änderungen mit der konsolidierten Datenbank synchronisieren
- **Nicht ortsgebundene Manager** Die Benutzer-ID 53 kennzeichnet die entfernte Datenbank, die nicht ortsgebundenen Managern zugeordnet ist. Ein nicht ortsgebundener Manager kann dieselben Aufgaben wie ein Vertriebsmitarbeiter ausführen. Zusätzlich hat der nicht ortsgebundene Manager noch folgende Möglichkeiten:
 - Bestellungen annehmen und ablehnen

Quellcode der Synchronisationslogik

Sie können die Synchronisationsskripten in der konsolidierten Datenbank mit Sybase Central untersuchen.

Skripttypen und Skriptereignisse

Die Datei *custdb.sql* fügt jedes Synchronisationsskript durch einen Aufruf von `ml_add_connection_script` bzw. `ml_add_table_script` zur konsolidierten Datenbank hinzu.

Beispiel

Die folgenden Zeilen in *custdb.sql* fügen ein Skript auf Tabellenebene für die Tabelle `ULProduct` hinzu, das während des Ereignisses `download_cursor` ausgeführt wird. Das Skript enthält eine einzelne `SELECT`-Anweisung.

```
call ml_server.ml_add_table_script(  
  'CustDB 16.0',  
  'ULProduct', 'download_cursor',  
  'SELECT prod_id, price, prod_name FROM ULProduct' )  
go
```

Synchronisieren der Bestellungen im Beispiel CustDB

Geschäftsregeln

Die Geschäftsregeln für die Tabelle `ULOrder` lauten folgendermaßen:

- Bestellungen werden nur heruntergeladen, wenn sie nicht bestätigt sind oder der Status `NULL` ist.
- Bestellungen können sowohl in der konsolidierten als auch in der entfernten Datenbank geändert werden.
- Jede entfernte Datenbank enthält nur die Bestellungen, die einem Mitarbeiter zugeordnet sind.

Downloads

Bestellungen können in der konsolidierten Datenbank eingefügt, gelöscht oder aktualisiert werden. Die entsprechenden Skripten für diese Vorgänge lauten folgendermaßen:

download_cursor Der erste Parameter im `download_cursor`-Skript ist der Zeitstempel des letzten Downloads. Er stellt sicher, dass nur Zeilen übertragen werden, die auf der entfernten oder der konsolidierten Datenbank seit der letzten Synchronisation geändert wurden. Der zweite Parameter ist die Mitarbeiter-ID. Mit ihr wird ermittelt, welche Zeilen übertragen werden sollen.

Das `download_cursor`-Skript für `CustDB` sieht wie folgt aus:

```
CALL ULOrderDownload( {ml s.last_table_download}, {ml s.username} )
```

Die `ULOrderDownload`-Prozedur für `CustDB` sieht wie folgt aus:

```
CREATE PROCEDURE ULOrderDownload ( IN LastDownload timestamp, IN EmployeeID  
integer )  
BEGIN
```

```

SELECT o.order_id, o.cust_id, o.prod_id, o.emp_id, o.disc, o.quant,
o.notes, o.status
FROM ULOrder o, ULEmpCust ec
WHERE o.cust_id = ec.cust_id
AND ec.emp_id = EmployeeID
AND ( o.last_modified >= LastDownload
OR ec.last_modified >= LastDownload)
AND ( o.status IS NULL OR o.status != 'Approved' )
AND ( ec.action IS NULL )
END

```

download_delete_cursor Das download_delete_cursor-Skript für CustDB sieht wie folgt aus:

```

SELECT o.order_id
FROM ULOrder o, dba.ULEmpCust ec
WHERE o.cust_id = ec.cust_id
AND ( ( o.status = 'Approved' AND o.last_modified >= {ml
s.last_table_download} )
OR ( ec.action = 'D' ) )
AND ec.emp_id = {ml s.username}

```

Uploads

Bestellungen können in der entfernten Datenbank eingefügt, gelöscht oder aktualisiert werden. Die entsprechenden Skripten für diese Vorgänge lauten folgendermaßen:

upload_insert Das upload_insert-Skript für CustDB sieht wie folgt aus:

```

INSERT INTO ULOrder ( order_id, cust_id, prod_id, emp_id, disc, quant,
notes, status )
VALUES( {ml r.order_id, r.cust_id, r.prod_id, r.emp_id, r.disc, r.quant,
r.notes, r.status } )

```

upload_update Das upload_update-Skript für CustDB sieht wie folgt aus:

```

UPDATE ULOrder
SET cust_id = {ml r.cust_id},
prod_id = {ml r.prod_id},
emp_id = {ml r.emp_id},
disc = {ml r.disc},
quant = {ml r.quant},
notes = {ml r.notes},
status = {ml r.status}
WHERE order_id = {ml r.order_id}

```

upload_delete Das upload_delete-Skript für CustDB sieht wie folgt aus:

```

DELETE FROM ULOrder WHERE order_id = {ml r.order_id}

```

upload_fetch Das upload_fetch-Skript für CustDB sieht wie folgt aus:

```

SELECT order_id, cust_id, prod_id, emp_id, disc, quant, notes, status
FROM ULOrder WHERE order_id = {ml r.order_id}

```

upload_old_row_insert Das upload_old_row_insert-Skript für CustDB sieht wie folgt aus:

```

INSERT INTO ULOldOrder ( order_id, cust_id, prod_id, emp_id, disc, quant,
notes, status )
VALUES( {ml r.order_id, r.cust_id, r.prod_id, r.emp_id, r.disc, r.quant,
r.notes, r.status } )

```

upload_new_row_insert Das upload_new_row_insert-Skript für CustDB sieht wie folgt aus:

```
INSERT INTO ULNewOrder ( order_id, cust_id, prod_id, emp_id, disc, quant,
notes, status )
VALUES( {ml r.order_id, r.cust_id, r.prod_id, r.emp_id, r.disc, r.quant,
r.notes, r.status } )
```

Konfliktlösung

resolve_conflict Das resolve_conflict-Skript für CustDB sieht wie folgt aus:

```
CALL ULResolveOrderConflict
```

Die ULResolveOrderConflict-Prozedur für CustDB sieht wie folgt aus:

```
CREATE PROCEDURE ULResolveOrderConflict()
BEGIN
-- approval overrides denial
IF 'Approved' = (SELECT status FROM ULNewOrder) THEN
UPDATE ULOrder o
SET o.status = n.status, o.notes = n.notes
FROM ULNewOrder n
WHERE o.order_id = n.order_id;
END IF;
DELETE FROM ULOldOrder;
DELETE FROM ULNewOrder;
END
```

Synchronisieren der Kunden im Beispiel CustDB

Geschäftsregeln

Die Geschäftsregeln für die Kunden lauten folgendermaßen:

- Kundendaten können in der konsolidierten und in der entfernten Datenbank geändert werden.
- Sowohl die entfernte als auch die konsolidierte Datenbank enthält eine vollständige Liste der Kunden.

Downloads

Kundendaten können in der konsolidierten Datenbank eingefügt und aktualisiert werden. Das entsprechende Skript für diese Vorgänge lautet folgendermaßen:

download_cursor Das folgende download_cursor-Skript lädt alle Kunden herunter, deren Daten sich seit dem letzten Download von Informationen durch den Benutzer geändert haben.

```
SELECT cust_id, cust_name FROM ULCustomer WHERE last_modified >= {ml
s.last_table_download}
```

Uploads

Kundendaten können in der entfernten Datenbank eingefügt, aktualisiert oder gelöscht werden. Die entsprechenden Skripten für diese Vorgänge lauten folgendermaßen:

upload_insert Das upload_insert-Skript für CustDB sieht wie folgt aus:

```
INSERT INTO ULCustomer( cust_id, cust_name )
VALUES( {ml r.cust_id, r.cust_name } )
```

upload_update Das upload_update-Skript für CustDB sieht wie folgt aus:

```
UPDATE ULCustomer SET cust_name = {ml r.cust_name}
WHERE cust_id = {ml r.cust_id}
```

In dieser Tabelle wird keine Konflikterkennung durchgeführt.

upload_delete Das upload_delete-Skript für CustDB sieht wie folgt aus:

```
DELETE FROM ULCustomer WHERE cust_id = {ml r.cust_id}
```

Synchronisieren der Produkte im Beispiel CustDB

Geschäftsregeln

Alle Zeilen werden für ULProduct heruntergeladen. Dies wird Snapshot-Synchronisation genannt.

Siehe „Snapshot-Synchronisation“ [[MobiLink - Serveradministration](#)].

Die Geschäftsregeln für die Tabelle ULProduct lauten folgendermaßen:

- Produkte können nur in der konsolidierten Datenbank geändert werden.
- Jede entfernte Datenbank enthält alle Produkte.

Downloads

Produktinformationen können in der konsolidierten Datenbank eingefügt, gelöscht oder aktualisiert werden. Das entsprechende Skript für diese Vorgänge lautet folgendermaßen:

download_cursor Das folgende download_cursor-Skript lädt bei jeder Synchronisation alle Zeilen und Spalten der Tabelle ULProduct herunter:

```
SELECT prod_id, price, prod_name FROM ULProduct
```

Verwalten des Primärschlüsselpools für Kunden und Bestellungen

Die Beispieldatenbank CustDB verwendet Primärschlüsselpools, um eindeutige Primärschlüssel in den Tabellen ULCustomer und ULOrder zu verwalten. Die Primärschlüsselpools sind die Tabellen ULCustomerIDPool und ULOrderIDPool.

ULCustomerIDPool

Die folgenden Skripten sind in der Tabelle ULCustomerIDPool definiert:

Downloads

begin_download Das begin_download-Skript für CustDB sieht wie folgt aus:

```
CALL ULCustomerIDPool_maintain( {ml s.username} )
```

Die ULCustomerIDPool_maintain-Prozedur für CustDB sieht wie folgt aus:

```
CREATE PROCEDURE ULCustomerIDPool_maintain ( IN syncuser_id INTEGER )
BEGIN
  DECLARE pool_count INTEGER;
  -- Determine how many ids to add to the pool
  SELECT COUNT(*) INTO pool_count
  FROM ULCustomerIDPool
  WHERE pool_emp_id = syncuser_id;
  -- Top up the pool with new ids
  WHILE pool_count < 20 LOOP
    INSERT INTO ULCustomerIDPool ( pool_emp_id )
    VALUES ( syncuser_id );
    SET pool_count = pool_count + 1;
  END LOOP;
END
```

download_cursor

```
SELECT pool_cust_id FROM ULCustomerIDPool
WHERE last_modified >= {ml s.last_table_download}
AND pool_emp_id = {ml s.username}
```

Uploads

upload_delete Das upload_delete-Skript für CustDB sieht wie folgt aus:

```
DELETE FROM ULCustomerIDPool
WHERE pool_cust_id = {ml r.pool_cust_id}
```

ULOrderIDPool

Die folgenden Skripten sind in der Tabelle ULOrderIDPool definiert:

Downloads

begin_download Das begin_download-Skript für CustDB sieht wie folgt aus:

```
CALL ULOrderIDPool_maintain( {ml s.username} )
```

Die ULOrderIDPool_maintain-Prozedur für CustDB sieht wie folgt aus:

```
ALTER PROCEDURE ULOrderIDPool_maintain ( IN syncuser_id INTEGER )
BEGIN
  DECLARE pool_count INTEGER;
  -- Determine how many ids to add to the pool
  SELECT COUNT(*) INTO pool_count
  FROM ULOrderIDPool
  WHERE pool_emp_id = syncuser_id;
  -- Top up the pool with new ids
  WHILE pool_count < 20 LOOP
    INSERT INTO ULOrderIDPool ( pool_emp_id )
    VALUES ( syncuser_id );
    SET pool_count = pool_count + 1;
  END LOOP;
END
```

download_cursor Das download_cursor-Skript für CustDB sieht wie folgt aus:


```
SELECT pool_order_id FROM ULOrderIDPool
WHERE last_modified >= {ml s.last_table_download}
AND pool_emp_id = {ml s.username}
```

Uploads

upload_delete Das upload_delete-Skript für CustDB sieht wie folgt aus:

```
DELETE FROM ULOrderIDPool
WHERE pool_order_id = {ml r.pool_order_id}
```

CustDB-Datenbank wiederherstellen

Um das Beispiel wiederherzustellen, führen Sie vom Verzeichnis *%SQLANYSAMPI6%\UltraLite\CustDB* den folgenden Befehl aus:

```
makedbs
```

MobiLink-Beispielanwendung "Contact"

Die Beispielanwendung Contact ist eine wertvolle Quelle für den MobiLink-Entwickler. Sie veranschaulicht die Implementierung vieler Funktionen, die für die Entwicklung von MobiLink-Anwendungen erforderlich sind.

Die Beispielanwendung Contact enthält eine konsolidierte SQL Anywhere-Datenbank und zwei entfernte SQL Anywhere-Datenbanken. Sie stellt mehrere übliche Synchronisationsmethoden vor. Um den größtmöglichen Nutzen aus diesem Abschnitt zu ziehen, überprüfen Sie beim Lesen die Beispielanwendung.

Die konsolidierte Datenbank ist zwar SQL Anywhere, die Synchronisationsskripten bestehen jedoch aus SQL-Anweisungen, die mit minimalen Änderungen auch mit anderen Datenbank-Managementsystemen eingesetzt werden können.

Die Beispielanwendung Contact befindet sich in *%SQLANYSAMPI6%\MobiLink\Contact*.

Synchronisationsplanung

Die Synchronisationsplanung in der Beispielanwendung Contact verwendet folgende Funktionen.

- **Spaltenteilmengen** Auf eine Teilmenge der Spalten der Tabellen Customer, Product, SalesRep und Contact in der konsolidierten Datenbank wird gemeinsam mit den entfernten Datenbanken zugegriffen.
- **Zeilenteilmengen** Auf alle Spalten, aber nur auf eine der Zeilen der Tabelle SalesRep in der konsolidierten Datenbank wird gemeinsam mit den verschiedenen entfernten Datenbanken zugegriffen. Siehe „[Partitionierte Zeilen in entfernten Datenbanken](#)“ [*MobiLink - Serveradministration*].
- **Synchronisation auf Zeitstempelbasis** Dies ist eine Möglichkeit, die in der konsolidierten Datenbank seit der letzten Synchronisation des Geräts vorgenommenen Änderungen festzustellen. Die Tabellen Customer, Contact und Product werden mit einer Methode synchronisiert, die auf

Zeitstempeln basiert. Siehe „[Implementieren zeitstempelbasierter Downloads](#)“ [*MobiLink - Serveradministration*].

Privilegien

Sie müssen die folgenden Rollen und Privilegien haben:

- SYS_REPLICATION_ADMIN_ROLE-Systemrolle
- SYS_RUN_REPLICATION_ROLE-Systemrolle
- CREATE ANY TRIGGER-Systemprivileg

Einrichten der Beispielanwendung Contact

Eine Windows-Batchdatei mit dem Namen *build.bat* wird mitgeliefert, damit Sie die Contact-Beispieldatenbanken aufbauen können. Unter Unix-Betriebssystemen heißt die Datei *build.sh*.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Kontext und Bemerkungen

Sie können den Inhalt der Batchdatei anzeigen. Sie führt folgende Aktionen aus:

- Erstellt Definitionen für ODBC-Datenquellen für eine konsolidierte Datenbank und für jede der beiden entfernten Datenbanken.
- Erstellt eine Datei der konsolidierten Datenbank mit dem Namen *consol.db* und lädt die MobiLink-Systemtabellen, einige Daten, Synchronisationsskripten und MobiLink-Benutzernamen in die Datenbank.
- Erstellt zwei entfernte Datenbanken, die jeweils *remote.db* heißen, in Unterverzeichnissen mit den Namen *remote_1* und *remote_2*. Lädt Informationen in beide Datenbanken und passt diese an. Zu diesen Anpassungen gehören eine globale Datenbankkennung, ein MobiLink-Benutzername sowie Subskriptionen für zwei Publikationen.

Aufgabe

1. Navigieren Sie über die Eingabeaufforderung zu `%SQLANYSAMPI6%\MobiLink\Contact`.
2. Führen Sie *build.bat* (Windows) oder *build.sh* (Unix) aus.

Ergebnisse

Die Contact-Beispieldatenbank wird erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Ausführen des Beispiels Contact](#)“ auf Seite 77.

Ausführen des Beispiels Contact

Das Beispiel Contact umfasst Batchdateien, die die Erstsynchronisation ausführen, und veranschaulicht den MobiLink-Server und das Befehlszeilenprogramm dbmlsync.

Voraussetzungen

Stellen Sie sicher, dass *build.bat* ausgeführt wurde, um die Contact-Beispieldatenbank zu erstellen.

Kontext und Bemerkungen

Sie können den Inhalt der folgenden Batchdateien, die sich im Unterverzeichnis *%SQLANYSAMPI6%\MobiLink\Contact* befinden, in einem Texteditor einsehen:

- *step1.bat*
- *step2.bat*
- *step3.bat*

Aufgabe

1. Starten Sie den MobiLink-Server.
 - a. Navigieren Sie über die Eingabeaufforderung zu *%SQLANYSAMPI6%\MobiLink\Contact*.
 - b. Führen Sie den folgenden Befehl aus:

```
step1
```

Der Befehl führt eine Batchdatei aus, die den MobiLink-Server im ausführlichen Modus startet. Dieser Modus ist in der Entwicklungsphase oder bei der Fehlersuche besonders nützlich, beeinträchtigt jedoch die Performance in erheblichem Maße und sollte daher nicht in einer Produktionsumgebung benutzt werden.

2. Synchronisieren Sie die beiden entfernten Datenbanken.
 - a. Navigieren Sie über die Eingabeaufforderung zu *%SQLANYSAMPI6%\MobiLink\Contact*.
 - b. Führen Sie den folgenden Befehl aus:

```
step2
```

Dies ist eine Batchdatei, die die beiden entfernten Datenbanken synchronisiert.

3. Fahren Sie den MobiLink-Server herunter.
 - a. Navigieren Sie über die Eingabeaufforderung zu *%SQLANYSAMPI6%\MobiLink\Contact*.
 - b. Führen Sie den folgenden Befehl aus:

```
step3
```

Dies ist eine Batchdatei, die den MobiLink-Server herunterfährt.

Ergebnisse

Die Contact-Beispieldatenbank wird ausgeführt und die entfernten und konsolidierten Datenbanken werden synchronisiert.

Nächste Schritte

Wenn Sie herausfinden wollen, wie die Synchronisation im Beispiel Contact funktioniert, können Sie mit Interactive SQL die Daten in der entfernten und der konsolidierten Datenbank ändern und dann mit den Batchdateien die Installation synchronisieren.

Tabellen in den Contact-Datenbanken

Die Tabellendefinitionen für die Contact-Datenbank befinden sich in den folgenden Dateien im Beispielverzeichnis:

- *MobiLink\Contact\build_consol.sql*
- *MobiLink\Contact\build_remote.sql*

Sowohl die konsolidierte als auch die entfernte Datenbank enthält die folgenden drei Tabellen. Die Definitionen unterscheiden sich jedoch jeweils geringfügig.

Tabelle SalesRep

Jeder Vertriebsmitarbeiter belegt eine Zeile in der Tabelle SalesRep. Jede entfernte Datenbank gehört einem Vertriebsmitarbeiter.

In den jeweiligen entfernten Datenbanken besitzt SalesRep folgende Spalten:

- **rep_id** Eine Primärschlüsselspalte mit einer Kennung für den Vertriebsmitarbeiter.
- **name** Den Namen des Vertriebsmitarbeiters.

Nur in der konsolidierten Datenbank gibt es außerdem die Spalte ml_username mit dem MobiLink-Benutzernamen für den Vertriebsmitarbeiter.

Tabelle Customer

Diese Tabelle enthält für jeden Kunden eine Zeile. Jeder einzelne Kunde ist ein Unternehmen, mit dem ein Vertriebsmitarbeiter Geschäfte abwickelt. Es liegt eine Eins-zu-Viele-Beziehung zwischen den Tabellen SalesRep und Customer vor.

In den jeweiligen entfernten Datenbanken besitzt Customer folgende Spalten:

- **cust_id** Eine Primärschlüsselspalte mit einer Kennung für den Kunden.
- **name** Der Name des Kunden. Dies ist der Name des Unternehmens.
- **rep_id** Eine Fremdspalte, die die Tabelle SalesRep referenziert. Sie kennzeichnet den Vertriebsmitarbeiter, der dem Kunden zugeordnet ist.

In der konsolidierten Datenbank sind außerdem die beiden Spalten `last_modified` und `active` enthalten:

- **last_modified** Zeitpunkt der letzten Änderung der Zeile. Diese Spalte wird für die Synchronisation auf Zeitstempelbasis benutzt.
- **active** Eine BIT-Spalte, die angibt, ob der Kunde derzeit aktiv ist (1) oder ob das Unternehmen keine Aufträge mehr mit diesem Kunden abwickelt (0). Wenn die Spalte als inaktiv (0) gekennzeichnet ist, werden alle Zeilen in den entfernten Datenbanken gelöscht, die im Zusammenhang mit diesem Ansprechpartner stehen.

Tabelle Contact

Diese Tabelle enthält für jeden Ansprechpartner eine Zeile. Ein Ansprechpartner ist eine Person, die im Kundenunternehmen arbeitet. Es gibt eine Eins-zu-Viele-Beziehung zwischen den Tabellen `Customer` und `Contact`.

In den jeweiligen entfernten Datenbanken besitzt `Contact` folgende Spalten:

- **contact_id** Eine Primärschlüsselspalte mit einer Kennung für den Kontakt.
- **name** Den Namen des Ansprechpartners.
- **cust_id** Die Kennung des Kunden, für den der Ansprechpartner arbeitet.

In der konsolidierten Datenbank enthält die Tabelle außerdem die folgenden Spalten:

- **last_modified** Zeitpunkt der letzten Änderung der Zeile. Diese Spalte wird für die Synchronisation auf Zeitstempelbasis benutzt.
- **active** Eine BIT-Spalte, die angibt, ob der Ansprechpartner derzeit aktiv ist (1) oder ob das Unternehmen keine Aufträge mehr mit diesem Ansprechpartner abwickelt (0). Wenn die Spalte als inaktiv (0) gekennzeichnet ist, wird die Zeile in den entfernten Datenbanken gelöscht, die im Zusammenhang mit diesem Ansprechpartner stehen.

Tabelle Product

Jedes Produkt, das vom Unternehmen verkauft wird, belegt eine Zeile in der Tabelle `Product`. Die Tabelle `Product` wird in einer separaten Publikation geführt, sodass entfernte Datenbanken die Tabelle separat synchronisieren können.

In den jeweiligen entfernten Datenbanken besitzt `Product` folgende Spalten:

- **id** Eine Primärschlüsselspalte mit einer Kennung für das Produkt.
- **name** Der Name des Artikels.
- **size** Die Größe des Artikels.
- **quantity** Die Anzahl der Artikel im Lagerbestand. Wenn ein Vertriebsmitarbeiter einen Auftrag entgegennimmt, wird der Wert in dieser Spalte aktualisiert.
- **unit_price** Den Preis pro Einheit des Produkts.

In der konsolidierten Datenbank enthält die Tabelle außerdem die folgenden Spalten:

- **supplier** Das Unternehmen, das das Produkt herstellt.
- **last_modified** Zeitpunkt der letzten Änderung der Zeile. Diese Spalte wird für die Synchronisation auf Zeitstempelbasis benutzt.
- **active** Eine BIT-Spalte, die angibt, ob das Produkt derzeit aktiv ist (1). Wenn die Spalte als inaktiv (0) gekennzeichnet ist, werden die Zeilen aus den entfernten Datenbanken gelöscht, die mit diesem Produkt übereinstimmen.

Zusätzlich zu diesen Tabellen wird nur für die konsolidierte Datenbank eine Reihe weiterer Tabellen erstellt. Dazu gehört die Tabelle `product_conflict`, eine temporäre Tabelle, die bei der Fehlersuche im Falle von Konflikten eingesetzt wird, und eine Reihe von Tabellen zur Überwachung der MobiLink-Aktivitäten mit dem Eigentümer `mlmaint`. Skripten zur Erstellung der MobiLink-Überwachungstabellen befinden sich in der Datei `%SQLANYSAMPI6%\MobiLink\Contact\mlmaint.sql`.

Benutzer im Beispiel Contact

Das Beispiel Contact enthält mehrere verschiedene Datenbankbenutzer-IDs und MobiLink-Benutzernamen.

Datenbankbenutzer-IDs

Die beiden entfernten Datenbanken sind den Vertriebsmitarbeitern Samuel Singer (`rep_id` 856) und Pamela Savarino (`rep_id` 949) zugewiesen.

Wenn sich diese Benutzer mit ihrer entfernten Datenbank verbinden, verwenden sie sowohl die Standardbenutzer-ID **dba** von SQL Anywhere als auch das Kennwort **SQL**.

Jede entfernte Datenbank hat auch eine Benutzer-ID **sync_user** mit dem Kennwort **sync_user**. Diese Benutzer-ID wird ausschließlich in der `dbmsync`-Befehlszeile verwendet. **sync_user** muss über die `SYS_RUN_REPLICATION_ROLE`-Systemrolle verfügen und kann alle Vorgänge ausführen, wenn er mit `dbmsync` verbunden ist. Er verfügt jedoch über keinerlei Berechtigungen, wenn er sich von einer anderen Anwendung aus anmeldet. Die Verwendung der ID **sync_user** und des entsprechenden Kennworts sollte also kein Problem darstellen.

Für die konsolidierte Datenbank gibt es einen Benutzer mit dem Namen **mlmaint**. Dieser Benutzer ist der Eigentümer der Tabellen, die zur Überwachung der MobiLink-Synchronisationsstatistiken und der eventuell auftretenden Fehler eingesetzt werden. Der Benutzer **mlmaint** hat keine Berechtigung, eine Verbindung herzustellen. Die Zuordnung der Tabellen zu einer separaten Benutzer-ID erfolgt nur, um die Objekte von den anderen im Schema zu trennen und um die Verwaltung in Sybase Central und in anderen Dienstprogrammen zu vereinfachen.

MobiLink-Benutzername

MobiLink-Benutzernamen unterscheiden sich von den Datenbankbenutzer-IDs. Jedes entfernte Gerät hat zusätzlich zur Benutzer-ID einen MobiLink-Benutzernamen, der beim Herstellen der Verbindung mit der Datenbank verwendet wird. Der MobiLink-Benutzername von Samuel Singer ist `SSinger`. Der MobiLink-

Benutzer für Pamela Savarino lautet PSavarino. Der MobiLink-Benutzername wird an folgenden Stellen gespeichert bzw. eingesetzt:

- In der entfernten Datenbank wird der MobiLink-Benutzer mit der Anweisung CREATE SYNCHRONIZATION USER hinzugefügt.
- In der konsolidierten Datenbank werden Benutzername und Kennwort mit dem Dienstprogramm "mluser" hinzugefügt.
- Während der Synchronisation wird das MobiLink-Kennwort des Benutzers, der die Verbindung herstellt, über die Befehlszeile von dbmlsync angegeben, die in *MobiLink\Contact\step2.bat* ausgeführt wird.
- Der MobiLink-Server übergibt den MobiLink-Benutzernamen während der Synchronisation bei vielen Skripten als Parameter.
- Die Tabelle SalesRep in der konsolidierten Datenbank enthält eine Spalte ml_username. Die Synchronisationsskripten vergleichen den MobiLink-Benutzernamen mit dem Wert in dieser Spalte.

Synchronisieren der Vertriebsmitarbeiter im Beispiel Contact

Die Synchronisationsskripten für die SalesRep-Tabelle veranschaulicht die **Snapshot-Synchronisation**. Unabhängig davon, ob sich die Daten des Benutzers geändert haben, werden sie heruntergeladen.

Siehe „Snapshot-Synchronisation“ [[MobiLink - Serveradministration](#)].

Geschäftsregeln

Die Geschäftsregeln für die Tabelle SalesRep lauten folgendermaßen:

- Die Tabelle darf in der entfernten Datenbank nicht geändert werden.
- MobiLink-Benutzername und rep_id-Wert eines Vertriebsmitarbeiters dürfen nicht geändert werden.
- Jede entfernte Datenbank enthält eine Zeile aus der Tabelle SalesRep, die dem MobiLink-Benutzernamen des Eigentümers der entfernten Datenbank entspricht.

Downloads

- **download_cursor** In jeder entfernten Datenbank enthält die Tabelle SalesRep eine einzige Zeile. Der Download einer einzelnen Zeile verursacht sehr wenig Aufwand, darum wird ein einfaches download_cursor-Skript mit einem Snapshot verwendet:

```
SELECT rep_id, name
FROM SalesRep
WHERE ? IS NOT NULL
AND ml_username = ?
```

Der erste Parameter im Skript ist der Zeitstempel des letzten Downloads. Er wird nicht verwendet. Der Ausdruck IS NOT NULL ist eine Ausdrucksattrappe, die dazu dient, dass der Parameter benutzt wird. Der zweite Parameter ist der MobiLink-Benutzername.

Uploads

Diese Tabelle darf nicht in der entfernten Datenbank aktualisiert werden, daher gibt es keine Upload-Skripten für die Tabelle.

Synchronisieren der Kunden im Beispiel Contact

Die Synchronisationsskripten für die Tabelle Customer veranschaulichen die **zeitstempelbasierte Synchronisation** und das Verteilen von Zeilen. Beide Methoden minimieren die während der Synchronisation übertragenen Datenmengen und gewährleisten konsistente Tabellendaten.

Siehe:

- „Implementieren zeitstempelbasierter Downloads“ [[MobiLink - Serveradministration](#)]
- „Partitionierte Zeilen in entfernten Datenbanken“ [[MobiLink - Serveradministration](#)]

Geschäftsregeln

Die Geschäftsregeln für die Kunden lauten folgendermaßen:

- Kundendaten können in der konsolidierten und in der entfernten Datenbank geändert werden.
- In gewissen Zeitabständen können Kunden den Vertriebsmitarbeitern neu zugeordnet werden. Dieses Verfahren wird im Allgemeinen "Gebietsregulierung" genannt.
- Jede entfernte Datenbank enthält nur die Kunden, die ihr zugeordnet wurden.

Downloads

- **download_cursor** Das folgende download_cursor-Skript lädt nur aktive Kunden herunter, deren Daten sich seit dem letzten erfolgreichen Download geändert haben. Hier werden auch Kunden nach Vertriebsmitarbeitern gefiltert.

```
SELECT cust_id, Customer.name, Customer.rep_id
FROM Customer key join SalesRep
WHERE Customer.last_modified >= ?
AND SalesRep.ml_username = ?
AND Customer.active = 1
```

- **download_delete_cursor** Das folgende download_delete_cursor-Skript lädt nur Kunden herunter, deren Daten sich seit dem letzten erfolgreichen Download geändert haben. Es löscht aus der entfernten Datenbank alle als inaktiv markierten Kunden bzw. solche, die nicht dem Vertriebsmitarbeiter zugeordnet sind.

```
SELECT cust_id
FROM Customer key join SalesRep
WHERE Customer.last_modified >= ?
AND ( SalesRep.ml_username != ? OR Customer.active = 0 )
```


Wenn Zeilen aus der Tabelle Customer in der konsolidierten Datenbank gelöscht werden, erscheinen sie nicht in dieser Ergebnismenge und werden also nicht aus den entfernten Datenbanken gelöscht. Stattdessen werden solche Kunden als inaktiv markiert.

Im Falle einer Gebietsregulierung werden mit diesem Skript die Kunden gelöscht, die dem betreffenden Vertriebsmitarbeiter nicht mehr zugeordnet sind. Es löscht auch Kunden, die an andere Vertriebsmitarbeiter übertragen wurden. Solche zusätzlichen Löschungen werden mit dem SQLCODE 100 gekennzeichnet, haben jedoch keinen Einfluss auf die Synchronisation. Es könnte ein komplexeres Skript entwickelt werden, das nur die Kunden abrufen, die dem aktuellen Vertriebsmitarbeiter entzogen wurden.

Der MobiLink-Client führt kaskadierende Löschungen in der entfernten Datenbank aus, sodass dieses Skript auch alle Ansprechpartner löscht, die bei Kunden arbeiten, die anderen Vertriebsmitarbeitern zugewiesen wurden.

Uploads

Kundendaten können in der entfernten Datenbank eingefügt, aktualisiert oder gelöscht werden. Die entsprechenden Skripten für diese Vorgänge lauten folgendermaßen:

- **upload_insert** Das folgende upload_insert-Skript fügt eine Zeile in die Customer-Tabelle ein und markiert den Kunden als aktiv:

```
INSERT INTO Customer(  
  cust_id, name, rep_id, active )  
VALUES ( ?, ?, ?, 1 )
```

- **upload_update** Das folgende upload_update-Skript ändert die Kundendaten in der konsolidierten Datenbank. In dieser Tabelle wird keine Konflikterkennung durchgeführt.

```
UPDATE Customer  
SET name = ?, rep_id = ?  
WHERE cust_id = ?
```

- **upload_delete** Das folgende upload_delete-Skript markiert die Kunden in der konsolidierten Datenbank als inaktiv. Es löscht keine Zeile.

```
UPDATE Customer  
SET active = 0  
WHERE cust_id = ?
```

Synchronisieren der Ansprechpartner im Beispiel Contact

Die Tabelle Contact enthält den Namen einer Person, die bei einem Kunden arbeitet, sowie einen Fremdschlüssel für den Kunden und eine eindeutige Ganzzahl zur Kennzeichnung des Ansprechpartners. Sie enthält außerdem einen last_modified-Zeitstempel und eine Markierung, die anzeigt, ob der Kontakt aktiv ist.

Geschäftsregeln

Die Geschäftsregeln für diese Tabelle lauten folgendermaßen:

- Ansprechpartnerdaten können in der konsolidierten und in der entfernten Datenbank geändert werden.
- Jede entfernte Datenbank enthält nur die Ansprechpartner, die für Kunden arbeiten, denen sie zugeordnet sind.
- Wenn die Kundenzuordnung zwischen Vertriebsmitarbeitern geändert wird, müssen Ansprechpartner ebenfalls neu zugeordnet werden.

Trigger

Ein Trigger in der Customer-Tabelle wird dafür verwendet, die Ansprechpartner aufzunehmen, wenn Angaben zu einem Kunden geändert werden. Der Trigger ändert ausdrücklich die Spalte `last_modified` der jeweiligen Ansprechpartner, wenn der betreffende Kunde geändert wird:

```
CREATE TRIGGER UpdateCustomerForContact
AFTER UPDATE OF rep_id ORDER 1
ON DBA.Customer
REFERENCING OLD AS old_cust NEW as new_cust
FOR EACH ROW
BEGIN
    UPDATE Contact
    SET Contact.last_modified = new_cust.last_modified
    FROM Contact
    WHERE Contact.cust_id = new_cust.cust_id
END
```

Da alle Datensätze der Ansprechpartner aktualisiert werden, wenn ein Kunde geändert wird, verbindet der Trigger den Kunden und die ihm zugewiesenen Ansprechpartner. Wenn ein Kunde geändert wird, werden alle zugeordneten Ansprechpartner ebenfalls geändert. Der Kunde und die zugeordneten Ansprechpartner werden bei der nächsten Synchronisation zusammen heruntergeladen.

Downloads

- **download_cursor** Das `download_cursor`-Skript für Contact sieht wie folgt aus:

```
SELECT contact_id, contact.name, contact.cust_id
FROM ( contact JOIN customer ) JOIN salesrep
ON contact.cust_id = customer.cust_id
AND customer.rep_id = salesrep.rep_id
WHERE Contact.last_modified >= ?
AND salesrep.ml_username = ?
AND Contact.active = 1
```

Dieses Skript ruft alle aktiven Ansprechpartner ab, die seit dem letzten Download durch den Vertriebsmitarbeiter geändert wurden (entweder ausdrücklich oder durch Änderung des betreffenden Kunden) und die dem Vertriebsmitarbeiter zugeordnet sind. Ein Join mit der Customer- und der SalesRep-Tabelle ist erforderlich, um die diesem Vertriebsmitarbeiter zugeordneten Ansprechpartner zu ermitteln.

- **download_delete_cursor** Das `download_delete_cursor`-Skript für Contact sieht wie folgt aus:

```
SELECT contact_id
FROM ( Contact JOIN Customer ) JOIN SalesRep
ON Contact.cust_id = Customer.cust_id
AND Customer.rep_id = SalesRep.rep_id
WHERE Contact.last_modified >= ?
AND Contact.active = 0
```

Die automatische Verwendung kaskadierender referenzieller Integrität durch den MobiLink-Client löscht Ansprechpartner, wenn der entsprechende Kunde aus der entfernten Datenbank gelöscht wird. Das `download_delete_cursor`-Skript muss somit nur die Ansprechpartner löschen, die als inaktiv markiert sind.

Uploads

Ansprechpartnerdaten können in der entfernten Datenbank eingefügt, aktualisiert oder gelöscht werden. Die entsprechenden Skripten für diese Vorgänge lauten folgendermaßen:

- **upload_insert** Das folgende `upload_insert`-Skript fügt eine Zeile in die `Contact`-Tabelle ein und markiert den Ansprechpartner als aktiv:

```
INSERT INTO Contact (
  contact_id, name, cust_id, active )
VALUES ( ?, ?, ?, 1 )
```

- **upload_update** Das folgende `upload_update`-Skript ändert die Daten des Ansprechpartners in der konsolidierten Datenbank:

```
UPDATE Contact
SET name = ?, cust_id = ?
WHERE contact_id = ?
```

In dieser Tabelle wird keine Konflikterkennung durchgeführt.

- **upload_delete** Das folgende `upload_delete`-Skript markiert den Ansprechpartner in der konsolidierten Datenbank als inaktiv. Es löscht keine Zeile.

```
UPDATE Contact
SET active = 0
WHERE contact_id = ?
```

Synchronisieren der Produkte im Beispiel Contact

Die Skripten für die `Product`-Tabelle veranschaulichen die Konflikterkennung und -lösung.

Die `Product`-Tabelle befindet sich in einer eigenen Publikation, getrennt von den anderen Tabellen, damit sie separat übertragen werden kann. Wenn z.B. Preisänderungen vorgenommen werden und der Vertriebsmitarbeiter über eine langsame Verbindung synchronisiert, kann er die Produktänderungen per Download übertragen, ohne die eigenen Änderungen in den Tabellen `Customer` und `Contact` per Upload übertragen zu müssen.

Geschäftsregeln

Die einzige Änderung, die in der entfernten Datenbank vorgenommen werden kann, ist die Änderung in der `Mengenspalte`, wenn ein Auftrag angenommen wird.

Downloads

- **download_cursor** Das folgende `download_cursor`-Skript lädt alle Zeilen herunter, die seit der letzten Synchronisation der entfernten Datenbank geändert wurden:

```
SELECT id, name, size, quantity, unit_price
FROM product
```

```
WHERE last_modified >= ?  
AND active = 1
```

- **download_delete_cursor** Das folgende download_delete_cursor-Skript entfernt alle Produkte, die vom Unternehmen nicht mehr vertrieben werden. Diese Produkte sind in der konsolidierten Datenbank als inaktiv markiert:

```
SELECT id, name, size, quantity, unit_price  
FROM product  
WHERE last_modified >= ?  
AND active = 0
```

Uploads

Nur UPDATE-Vorgänge werden von der entfernten Datenbank per Upload übertragen. Die wichtigste Funktion dieser Upload-Skripten ist die Prozedur zur Erkennung und Lösung von Konflikten.

Wenn zwei Vertriebsmitarbeiter Aufträge entgegennehmen und dann synchronisieren, werden beide Aufträge von dem Wert in der Spalte quantity der Tabelle Product subtrahiert. Wenn Samuel Singer beispielsweise eine Bestellung für 20 Baseballkappen (Produkt-ID 400) aufnimmt, ändert er die Menge von 90 zu 70. Wenn Pamela Savarino eine Bestellung für 10 Baseballkappen erhält, ändert sie die Spalte in ihrer Datenbank von 90 zu 80.

Wenn Samuel Singer seine Änderungen synchronisiert, wird der Wert in der Mengenspalte in der konsolidierten Datenbank von 90 auf 70 geändert. Wenn Pamela Savarino ihre Änderungen synchronisiert, wäre der richtige Vorgang, den Wert auf 60 zu setzen. Diese Änderung wird dadurch erreicht, dass der Konflikt erkannt wird.

Das Konflikterkennungsschema umfasst die folgenden Skripten.

- **upload_update** Das folgende upload_update-Skript führt eine einfache UPDATE-Anweisung in der konsolidierten Datenbank aus:

```
UPDATE product  
SET name = ?, size = ?, quantity = ?, unit_price = ?  
WHERE product.id = ?
```

- **upload_fetch** Das folgende upload_fetch-Skript ruft eine einzelne Zeile aus der Tabelle Product ab und vergleicht sie mit den alten Werten der per Upload übertragenen Zeile. Wenn die beiden Zeilen verschieden sind, wird ein Konflikt erkannt.

```
SELECT id, name, size, quantity, unit_price  
FROM Product  
WHERE id = ?
```

- **upload_old_row_insert** Wenn ein Konflikt erkannt wird, werden die alten Werte in die Tabelle product_conflict eingefügt, damit das resolve_conflict-Skript darauf zugreifen kann. Die Zeile wird mit dem Wert "O" (für Old – Alt) in der Spalte row_type eingefügt.

```
INSERT INTO DBA.product_conflict(  
id, name, size, quantity, unit_price, row_type )  
VALUES( ?, ?, ?, ?, ?, 'O' ) )
```

- **upload_new_row_insert** Das folgende Skript fügt die neuen Werte der per Upload übertragenen Zeile in die Tabelle product_conflict ein, damit das resolve_conflict-Skript darauf zugreifen kann:

```
INSERT INTO DBA.product_conflict(
  id, name, size, quantity, unit_price, row_type )
VALUES( ?, ?, ?, ?, ?, 'N' )
```

Konfliktlösung

- **resolve_conflict** Das folgende Skript löst den Konflikt, indem es die Differenz zwischen neuen und alten Zeilen dem Mengenwert in der konsolidierten Datenbank hinzurechnet.

```
UPDATE Product
SET p.quantity = p.quantity
                - old_row.quantity
                + new_row.quantity
FROM Product p,
     DBA.product_conflict old_row,
     DBA.product_conflict new_row
WHERE p.id = old_row.id
      AND p.id = new_row.id
      AND old_row.row_type = 'O'
      AND new_row.row_type = 'N'
```

Statistiken und Fehlerüberwachung im Beispiel Contact

Das Beispiel Contact umfasst einige einfache Skripten zur Fehlerberichterstattung und zur Überwachung. Die SQL-Anweisungen für die Erstellung dieser Skripten befinden sich in der Datei *MobiLink\Contact\mlmaint.sql*.

Die Skripten fügen Zeilen in Tabellen ein, die für die Werte erstellt wurden. Zur Vereinfachung der Situation ist ein anderer Benutzer, nämlich mlmaint, der Eigentümer.

Praktische Einführungen in MobiLink

Dieser Abschnitt enthält praktische Einführungen in die Einrichtung und Verwendung der MobiLink-Technologie. Diese reichen von praktischen Einführungen für neue Benutzer bis hin zu Beispielen, wie Sie erweiterte Funktionen benutzen.

Weitere praktische Einführungen für MobiLink stehen online zur Verfügung. Siehe <http://www.sybase.com/detail?id=1081144>.

Hinweis

Die Online-Einführungen basieren auf Version 12.0.0 von SQL Anywhere. Einige Darstellungen und Prozeduren unterscheiden sich möglicherweise von SQL Anywhere 16.0.

Praktische Einführung: Einführung in MobiLink

Diese praktische Einführung zeigt Ihnen die grundlegenden Schritte für das Schreiben von Synchronisationsskripten, das Interpretieren von MobiLink-Logs und die Überwachung von Synchronisationen zwischen einer konsolidierten Datenbank und zwei entfernten Datenbanken mit dem MobiLink-Profiler. Sie enthält Anweisungen zum Einrichten der Datenbanken und der Synchronisationen mit Sybase Central.

Erforderliche Software

- SQL Anywhere 16

Privilegien

Sie müssen die folgenden Rollen und Privilegien für die konsolidierte Datenbank haben:

- SYS_AUTH_DBA_ROLE-Kompatibilitätsrolle

Sie müssen die folgenden Rollen und Privilegien für die entfernte Datenbank haben:

- SYS_AUTH_DBA_ROLE-Systemrolle
- SYS_RUN_REPLICATION_ROLE-Systemrolle

Kenntnisse und Erfahrungen

Hierzu benötigen Sie Folgendes:

- Grundkenntnisse über MobiLink-Ereignisskripten

Übersicht

In dieser praktischen Einführung wird Folgendes veranschaulicht:

- Migrieren eines Schemas der konsolidierten Datenbank in entfernte Datenbanken

- Erstellen der für die Synchronisation erforderlichen Basisskripten mithilfe von Sybase Central und Speichern der Skripten in der konsolidierten Datenbank
- Starten des MobiLink-Servers
- Synchronisation mithilfe von Logdateien und MobiLink-Profiler überwachen

Siehe auch

- „Interactive SQL“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Sybase Central“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „MobiLink-Synchronisation“ auf Seite 1
- „Schreiben von Synchronisationsskripten“ [[MobiLink - Serveradministration](#)]
- „Synchronisationsmethoden“ [[MobiLink - Serveradministration](#)]
- „Synchronisationsereignisse“ [[MobiLink - Serveradministration](#)]
- „MobiLink-Profiler“ [[MobiLink - Serveradministration](#)]
- „Konfliktverarbeitung“ [[MobiLink - Serveradministration](#)]
- <http://www.sybase.com/detail?id=1058600#319> (Sie benötigen ein Login für Sybase.com, um diese Seite anzuzeigen.)
- <http://sqlanywhere-forum.sybase.com>

Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank

In dieser Lektion richten Sie eine konsolidierte SQL Anywhere-Datenbank ein, indem Sie sie erstellen und eine ODBC-Datenquelle definieren.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Klicken Sie auf **Start » Programme » SQL Anywhere 16 » Administrationstools » Sybase Central**.
2. Klicken Sie auf **Extras » SQL Anywhere 16 » Datenbank erstellen**.
3. Klicken Sie auf **Weiter**.
4. Behalten Sie die Standardeinstellung für **Datenbank auf diesem Computer erstellen** bei und klicken Sie auf **Weiter**.
5. Im Feld **Haupt-Datenbankdatei in folgender Datei speichern** geben Sie `c:\MLintro\MLconsolidated.db` ein. Klicken Sie auf **Weiter**. Wenn das Verzeichnis nicht vorhanden ist, werden Sie aufgefordert, es zu erstellen. Klicken Sie auf **Ja**.

6. Befolgen Sie die übrigen Anweisungen des **Assistenten zum Erstellen einer Datenbank** und akzeptieren Sie die Standardwerte. Wenn Sie dazu aufgefordert werden, eine Benutzer-ID und ein Kennwort für den DBA-Benutzer anzugeben, geben Sie **DBA** und **sql** ein.

Auf der Seite **Mit Datenbank verbinden** aktivieren Sie die Option **Datenbank nach der letzten getrennten Verbindung herunterfahren**.

7. Klicken Sie auf **Fertig stellen**.

Die **MLconsolidated**-Datenbank wird erstellt.

8. Klicken Sie im Fenster **Datenbank erstellen** auf **Schließen**, wenn das Fenster nicht automatisch geschlossen wurde.
9. Klicken Sie auf **Verbindungen » Trennen**, um den Personal Datenbankserver zu stoppen.
10. Klicken Sie auf **Extras » SQL Anywhere 16 » ODBC-Administrator öffnen**.
11. Klicken Sie auf der Registerkarte **Benutzer-DSN** auf **Hinzufügen**.
12. Klicken Sie im Fenster **Neue Datenquelle erstellen** auf **SQL Anywhere 16** und auf **Fertig stellen**.
13. Führen Sie im Fenster **ODBC-Konfiguration für SQL Anywhere** folgende Aufgaben aus:
 - a. Klicken Sie auf die Registerkarte **ODBC**.
 - b. Im Feld **Datenquellennamen** geben Sie **mlintro_consdb** ein.
 - c. Klicken Sie auf die Registerkarte **Login**.
 - d. Lassen Sie in der Dropdown-Liste **Authentifizierung** die Standardoption **Datenbank** ausgewählt, um eine Verbindung mit Ihrer Benutzer-ID und Ihrem Kennwort einzuleiten.
 - e. Im Feld **Benutzer-ID** geben Sie **DBA** ein.
 - f. Im Feld **Kennwort** geben Sie **sql** ein.
 - g. Behalten Sie in der Dropdown-Liste **Aktion** die Standardeinstellung **Eine Datenbank auf diesem Computer starten und eine Verbindung herstellen** bei.
 - h. Geben Sie im Feld **Datenbankdatei** **c:\MLintro\MLconsolidated.db** ein.
 - i. Im Feld **Servername** geben Sie **MLconsolidated** ein.
 - j. Geben Sie im Feld **Startzeile** **"dbsrv16"** ein.
 - k. Klicken Sie auf die Registerkarte **ODBC** und klicken Sie auf **Verbindung testen**, um zu überprüfen, ob die Verbindung erfolgreich ist. Klicken Sie auf **OK**.
 - l. Klicken Sie auf **OK** und dann nochmals auf **OK**, um das Fenster **ODBC-Datenquellen-Administrator** zu schließen.
14. Klicken Sie auf **Verbindungen » Verbinden mit SQL Anywhere 16**.
15. Wählen Sie in der Dropdown-Liste **Aktion** die Option **Mit ODBC-Datenquelle verbinden**.
16. Im Feld **ODBC-Datenquellennamen** geben Sie **mlintro_consdb** ein.

17. Klicken Sie auf **Verbinden**.

Ergebnisse

Eine konsolidierte SQL Anywhere-Datenbank und eine ODBC-Datenquelle werden erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 2: Erstellen und Füllen einer Tabelle in der konsolidierten Datenbank von MobiLink](#)“ auf Seite 92.

Siehe auch

- „MobiLink - konsolidierte Datenbanken“ [[MobiLink - Serveradministration](#)]
- „Dienstprogramm Initialisierung (dbinit)“ [[SQL Anywhere Server - Datenbankadministration](#)]

Lektion 2: Erstellen und Füllen einer Tabelle in der konsolidierten Datenbank von MobiLink

In dieser Lektion erstellen Sie die Tabelle **Product** und fügen Beispieldaten in die konsolidierte MobiLink-Datenbank ein.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 90.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Einführung in MobiLink](#)“ auf Seite 89.

Aufgabe

1. Verbinden Sie sich über Interactive SQL mit Ihrer konsolidierten Datenbank. Führen Sie an der Eingabeaufforderung folgenden Befehl aus:

Sie können Interactive SQL über Sybase Central oder eine Eingabeaufforderung starten.

- Klicken Sie in Sybase Central mit der rechten Maustaste auf die Datenbank **MLconsolidated - DBA** und klicken Sie dann auf **Open Interactive SQL**.
- `dbisql -c "DSN=mlintro_consdb"`

2. Führen Sie die nachstehende SQL-Anweisung aus, um die Tabelle **Product** zu füllen:

```
CREATE TABLE Product (  
  name VARCHAR(128) NOT NULL PRIMARY KEY,  
  quantity INTEGER  
);
```

Die **Product**-Tabelle enthält folgende Spalten:

Spalte	Beschreibung
name	Name des Produkts.
quantity	Anzahl der verkauften Artikel

Füllen Sie nach dem Erstellen der Tabellen die **Product**-Tabelle mit Beispieldaten.

- Um die Tabelle **Product** mit Beispieldaten zu füllen, führen Sie folgende SQL-Anweisungen in Interactive SQL aus:

```
INSERT INTO Product(name, quantity)
VALUES ( 'Screwmaster Drill', 10);

INSERT INTO Product(name, quantity)
VALUES ( 'Drywall Screws 10lb', 30);

INSERT INTO Product(name, quantity)
VALUES ( 'Putty Knife x25', 12);

COMMIT;
```

- Vergewissern Sie sich, dass die **Product**-Tabelle die im vorherigen Schritt eingefügten Daten enthält.

Führen Sie die folgende SQL-Anweisung aus, um den Inhalt zu prüfen:

```
SELECT * FROM Product
```

Der Inhalt der **Product**-Tabelle sollte in Interactive SQL erscheinen.

- Schließen Sie Interactive SQL. Sie müssen Ihre SQL-Anweisungen nicht speichern.

Ergebnisse

Die Tabelle "Products" wird in der konsolidierten Datenbank erstellt.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 3: Erstellen eines MobiLink-Projekts und eines Synchronisationsmodells“](#) auf Seite 93.

Siehe auch

- „Interactive SQL“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „CREATE TABLE-Anweisung“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]

Lektion 3: Erstellen eines MobiLink-Projekts und eines Synchronisationsmodells

In dieser Lektion erstellen Sie ein neues MobiLink-Projekt mithilfe des **Assistenten zum Erstellen eines Projekts**. Der **Assistent zum Erstellen eines Projekts** erstellt auch ein Synchronisationsmodell mithilfe der Standardwerte, die später bearbeitet werden können.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 90.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Einführung in MobiLink](#)“ auf Seite 89.

Aufgabe

1. Klicken Sie in Sybase Central auf **Extras » MobiLink 16 » Neues Projekt**.

Der **Assistent zum Erstellen eines Projekts** wird angezeigt.

2. Im Feld **Name** geben Sie **mlintro_project** ein.
3. Geben Sie im Feld **Speicherort** den Pfad **C:\MLintro** ein und klicken Sie auf **Weiter**.
4. Im Feld **Anzeigename der Datenbank** geben Sie **mlintro_consdb** ein.
5. Geben Sie im Feld **Verbindungszeichenfolge** **dsn=mlintro_consdb** ein und klicken Sie dann auf **Weiter**.
6. Stellen Sie sicher, dass die Tabelle **Produkt** aus der Liste **Welche konsolidierten Datenbanktabellen und Spalten sollen in der entfernten Datenbank enthalten sein?** gewählt ist und klicken Sie dann auf **Weiter**.
7. Wählen Sie die Option **Dem Projekt einen Namen für ein entferntes Schema hinzufügen** aus.
8. Im Feld **Wie lautet der Name des neuen entfernten Schemas?** geben Sie **sync_mlintro** ein. Klicken Sie anschließend auf **Weiter**.
9. Wählen Sie **SQL Anywhere** für die Option **Welcher entfernte Datenbanktyp soll verwendet werden** und klicken Sie auf **Fertig stellen**.
10. Klicken Sie auf **Ja**, um die MobiLink-Systemtabellen zu installieren, und anschließend auf **OK**.

Ein Synchronisationsmodell mit demselben Namen wie das entfernte Schema wird erstellt.

11. Wählen Sie die Tabellenzeile **Produkt** auf der Registerkarte **Zuordnungen**.
12. Wählen Sie auf der Registerkarte **Downloadtyp** im Fensterausschnitt **Details** die Option **Zeitstempel** als **Downloadtyp**.
13. Wählen Sie **Schattentabelle für Zeitstempelspalte verwenden**.

Die Verwendung von Schattentabellen wird häufig bevorzugt, da hierbei keine Änderung von vorhandenen Tabellen erforderlich ist. Im Gegensatz dazu fügt die Standardeinstellung eine Zeitstempelspalte zu Ihrer Tabelle hinzu und hat normalerweise eine bessere Performance.

14. Um die Änderungen des Synchronisationsmodells zu speichern, wählen Sie **Datei » Speichern**.

Ergebnisse

Das MobiLink-Projekt, der Name des entfernten Schemas und ein Synchronisationsmodell werden erstellt.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 4: Synchronisationsmodell testen“ auf Seite 95](#).

Siehe auch

- „Synchronisationsmodelle“ auf Seite 30
- „Einrichten einer konsolidierten Datenbank“ [*MobiLink - Serveradministration*]
- „Systemtabellen des MobiLink-Servers“ [*MobiLink - Serveradministration*]
- „Systemprozeduren des MobiLink-Servers“ [*MobiLink - Serveradministration*]
- „download_delete_cursor-Skripten“ [*MobiLink - Serveradministration*]
- „Konfliktverarbeitung“ [*MobiLink - Serveradministration*]
- „Konfliktlösung“ [*MobiLink - Serveradministration*]
- „Publikationen“ [*MobiLink - Clientadministration*]
- „Synchronisationsmodellaufgaben“ auf Seite 33
- „Downloadtyp ändern“ auf Seite 38
- „Konflikterkennung und -lösung ändern“ auf Seite 46
- „Tabellen- und Spaltenzuordnungen“ auf Seite 34

Lektion 4: Synchronisationsmodell testen

In dieser Lektion wird gezeigt, wie Sie Ihr Synchronisationsmodell schnell testen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank“ auf Seite 90](#).

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Einführung in MobiLink“ auf Seite 89](#).

Aufgabe

1. Erweitern Sie im linken Fensterausschnitt von Sybase Central unter **MobiLink 16** die Elemente **mlintro_project**, **Synchronisationsmodelle** und dann **sync_mlintro**.
2. Klicken Sie auf **Datei » Test**.
3. Klicken Sie auf **OK**, um die Warnung zu übergehen, die darauf hinweist, dass Ihre konsolidierte Datenbank geändert wird.

Das Synchronisationsmodell wird in der konsolidierten Datenbank bereitgestellt und eine entfernte Datenbank wird für Testzwecke erstellt. Der MobiLink-Server wird gestartet.

4. Klicken Sie auf die Registerkarte **Daten**. Der obere Fensterausschnitt zeigt Zeilen aus der Produkt-Tabelle in der konsolidierten Datenbank. Der untere Fensterausschnitt zeigt, dass die entfernte Datenbank zurzeit keine Zeilen in der Produkttabelle enthält.

Wenn mehr als eine Tabelle definiert wurde, können Sie in der Dropdown-Liste **Anzeigen** auswählen, welche Tabelle angezeigt werden soll.

5. Klicken Sie auf **Synchronisieren**. Die drei Zeilen erscheinen nun in der entfernten Datenbank.
6. Wählen Sie die Registerkarte **Client-Log**. Scannen Sie das Log auf Fehlermeldungen oder Warnungen. Führen Sie auf der Registerkarte **MobiLink-Log** das Gleiche durch.
7. Klicken Sie auf die Registerkarte **Daten**. Klicken Sie im unteren Fensterausschnitt mit der rechten Maustaste auf **Drywall Screws 10lb**, wählen Sie **Zeile bearbeiten** und geben Sie die Menge 99 ein.
8. Klicken Sie auf **Synchronisieren**. Die Änderung in der entfernten Datenbank wurde mit der konsolidierten Datenbank synchronisiert.
9. Klicken Sie auf **Aktionen** und wählen Sie dann **Interactive SQL in der konsolidierten Datenbank öffnen**. Das Interactive SQL-Fenster wird mit einer Verbindung zur konsolidierten Datenbank geöffnet.
10. Führen Sie hierzu die folgende SQL-Anweisung aus und schließen Sie dann Interactive SQL:

```
update product set quantity = quantity + 1;  
commit;
```

11. Klicken Sie auf der Registerkarte **Daten** in Sybase Central auf **Aktionen » Daten aktualisieren**. Sie können die aktualisierten Zeilen in der konsolidierten Datenbank sehen.
12. Klicken Sie auf **Synchronisieren**. Die entfernte Datenbank wird mit den neuen Mengenwerten aktualisiert.
13. Schließen Sie das Fenster **Test**.

Ergebnisse

Sie haben das Synchronisationsmodell erfolgreich getestet.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 5: Synchronisationsmodell erstellen](#)“ auf Seite 97.

Siehe auch

- „[Deployment des Synchronisationsmodells](#)“ auf Seite 50

Lektion 5: Synchronisationsmodell erstellen

In dieser Lektion wird gezeigt, wie Sie Ihr Synchronisationsmodell ändern können und welche Möglichkeiten Ihnen bei der Entwicklung eines Modells zur Verfügung stehen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 90.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Einführung in MobiLink](#)“ auf Seite 89.

Aufgabe

1. Erweitern Sie im linken Fensterausschnitt von Sybase Central unter **MobiLink 16** die Elemente **mlintro_project » Synchronisationsmodelle » sync_mlintro**.
2. Wählen Sie auf der Registerkarte **Zuordnungen** die Zeile für die Produkttabelle.
3. Öffnen Sie im unteren Fensterausschnitt die Registerkarte **Konfliktverarbeitung**.
4. Wählen Sie für **Konfliktlösung Konsolidiert**. Durch diese Option wird festgelegt, dass bei der Änderung einer Zeile sowohl in der konsolidierten als auch in der entfernten Datenbank der Wert der konsolidierten Datenbank als der korrekte Wert betrachtet wird.
5. Klicken Sie auf der Registerkarte **Deployment** auf **Test**. Wenn Sie aufgefordert werden, das Synchronisationsmodell zu speichern, klicken Sie auf **Ja**.
6. Wählen Sie die Registerkarte **Daten** und klicken Sie dann auf **Synchronisieren**, um die entfernte Datenbank zu aktualisieren.
7. Bearbeiten Sie die Zeile mit dem Namen **Screwmaster Drill** und setzen Sie die Menge der konsolidierten Datenbank auf 20 und die Menge der entfernten Datenbank auf 10.
8. Klicken Sie auf **Synchronisieren**. Sowohl die entfernten als auch die konsolidierten Datenbanken sollten eine Menge von 20 anzeigen.
9. Schließen Sie das Fenster **Test**.
10. Wählen Sie die Registerkarte **Ereignisse** für das Modell **sync_mlintro**. Diese Seite zeigt die SQL-Anweisungen, die der MobiLink-Server für dieses Synchronisationsmodell ausführt. Die grünen Balken am linken Rand weisen darauf hin, dass die SQL-Anweisungen automatisch auf der Basis der Auswahlen generiert werden, die Sie auf der Registerkarte **Zuordnungen** getroffen haben.

Es gibt Fälle, in denen die Optionen, die auf der Registerkarte **Zuordnungen** zur Verfügung stehen, nicht für Ihr System ausreichen. In diesen Fällen können Sie Ihre Synchronisationsskripte auf der Registerkarte **Ereignisse** weiter anpassen.

11. Prüfen Sie auf der Registerkarte **Ereignisse** das upload_insert-Ereignis für die Produkttabelle. Dies zeigt die SQL-Anweisungen, die der MobiLink-Server ausführt, wenn er eine neue Zeile von einer entfernten Datenbank empfängt. Ändern Sie dieses Ereignis, um alle neuen Bestellungen auf eine maximale Menge von 50 zu beschränken. Der Text **{ml r."quantity"}** steht für die entfernte Mengenspalte, die hochgeladen wurde. Ändern Sie dies folgendermaßen:

```
If {ml r."quantity"} < 50 then {ml r."quantity"} else 50 end if
```

Das vollständige upload_insert-Ereignis sollte nun wie folgt aussehen:

```
Product (DBA): upload_insert
/* Insert the row into the consolidated database. */
INSERT INTO "DBA"."Product" ( "name", "quantity" )
VALUES ( {ml r."name"}, If {ml r."quantity"} < 50 then {ml r."quantity"}
else 50 end if )
```

Beachten Sie, dass die Fortschrittsanzeige am Rand für das upload_insert-Ereignis nun gelb ist, womit angezeigt wird, dass die Einstellungen auf der Registerkarte **Zuordnungen** überschrieben wurden.

12. Wählen Sie die Registerkarte **Deployment** und klicken Sie auf **Test**. Wählen Sie die Registerkarte **Daten** und klicken Sie auf die Option **Synchronisieren**. Klicken Sie mit der rechten Maustaste im unteren Fensterausschnitt für die entfernte Datenbank. Wählen Sie **Zeile hinzufügen** und fügen Sie eine Zeile mit dem Namen **Hammer** und einer Menge von **200** hinzu. Klicken Sie auf **Synchronisieren**. Jetzt enthalten sowohl die konsolidierte als auch die entfernte Datenbank einen Wert von 50 für die Zeile **Hammer**.
13. Schließen Sie das Fenster **Test** und kehren Sie zurück zur Registerkarte **Ereignisse**. Klicken Sie mit der rechten Maustaste auf das upload_insert-Ereignis-Skript und wählen Sie dann **Skript 'Product (DBA): upload_insert' wiederherstellen**. Die Fortschrittsanzeige am Rand ist nun wieder grün und die zuvor durchgeführten Anpassungen wurden rückgängig gemacht.

Ergebnisse

Die Änderungen werden am Synchronisationsmodell durchgeführt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 6: MobiLink-Serveroptionen wählen](#)“ auf Seite 98.

Siehe auch

- „[Deployment des Synchronisationsmodells](#)“ auf Seite 50

Lektion 6: MobiLink-Serveroptionen wählen

In dieser Lektion wird gezeigt, wie Sie Optionen wählen, die später für die Ausführung des MobiLink-Servers verwendet werden, wenn Sie das Deployment für Ihr Synchronisationsmodell vornehmen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 90.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Einführung in MobiLink“ auf Seite 89](#).

Aufgabe

1. Erweitern Sie im linken Fensterausschnitt von Sybase Central unter **MobiLink 16 mlintro_project Server-Befehlszeilen**. Wenn Sie aufgefordert werden, Ihr Synchronisationsmodell zu speichern, klicken Sie auf **Ja**.
2. Rechtsklicken Sie im rechten Fensterausschnitt auf **Standardwert** und wählen Sie **Eigenschaften** aus.
3. Führen Sie im Fenster **Standardeigenschaften der MobiLink-Server-Befehlszeile** die folgenden Aufgaben aus:
 - Wählen Sie die Registerkarte **Allgemein** aus. Wählen Sie aus der Liste **Ausführlichkeitsstufe** die Option **Hoch (-v+)**.
 - Wählen Sie die Registerkarte **Erweitert**
 - Setzen Sie die Option **-dl** auf **TRUE**.
 - Setzen Sie die Option **-o** auf **mlsrv.mls**.
 - Setzen Sie die Option **-zf** auf **TRUE**.
 - Setzen Sie die Option **-zu** auf **TRUE**.

Hinweis

Die Option **-zf** sollte nur für Debugging und Entwicklungsarbeiten verwendet werden. In dieser praktischen Einführung ist die Option **-zf** erforderlich, damit Sie den MobiLink-Server nicht herunterfahren müssen, wenn in einer späteren Lektion der konsolidierten Datenbank neue Skripten hinzugefügt werden. Mit der Option **-zu+** werden automatisch neue MobiLink-Benutzer zur Synchronisationsumgebung hinzugefügt.

4. Klicken Sie auf **OK**.

Ergebnisse

Die Optionen für das Ausführen des MobiLink-Servers beim Deployment des Synchronisationsmodells wurden eingestellt.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 7: Deployment des Synchronisationsmodells“ auf Seite 100](#).

Siehe auch

- [„Deployment des Synchronisationsmodells“ auf Seite 50](#)

Lektion 7: Deployment des Synchronisationsmodells

In dieser Lektion stellen Sie das Synchronisationsmodell aus der vorherigen Lektion mit dem **Assistenten für das Deployment eines Synchronisationsmodells** bereit, um die konsolidierte Datenbank für die Synchronisation zu konfigurieren. Außerdem erstellen Sie die entfernte Datenbank und stellen sie per Deployment bereit.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 90.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Einführung in MobiLink](#)“ auf Seite 89.

Aufgabe

1. Erweitern Sie im linken Fensterausschnitt von Sybase Central unter **MobiLink 16** die Elemente **mlintro_project » Synchronisationsmodelle » sync_mlintro**.
2. Klicken Sie auf **Datei » Speichern**.
3. Akzeptieren Sie die Standardeinstellung für das Feld **Wählen Sie den Ordner für vom Assistenten generierte Dateien** und klicken Sie auf **Weiter**.
4. Akzeptieren Sie die Standardeinstellungen für die Seite **Client-Netzwerkoptionen** und klicken Sie auf **Weiter**.
5. Auf der Seite **MobiLink-Benutzer und -Kennwort** unter **Welchen Benutzer und welches Kennwort für MobiLink möchten Sie verwenden?** wählen Sie **Diese verwenden** und führen die folgenden Aufgaben durch:
 - a. Geben Sie im Feld **MobiLink-Benutzer** den Wert **mlintro_user** ein.
 - b. Geben Sie im Feld **MobiLink-Kennwort** den Wert **sql** ein.
 - c. Wählen Sie **Registrieren Sie diesen Benutzer in der konsolidierten Datenbank. Registrierte Benutzer sind berechtigt, zu synchronisieren**. Klicken Sie auf **Weiter**.
6. Geben Sie auf der Seite **Synchronisationsprofil** den Wert **mlintro_remote_syncprofile** im Feld **Synchronisationsprofilname** ein. Klicken Sie auf **Durchsuchen** und setzen Sie **Ausführlichkeitsstufe** auf **Hoch**. Klicken Sie auf **OK** und klicken Sie dann auf **Weiter**.
7. Führen Sie die folgenden Aufgaben auf der Seite **Auswählen, wie Datenbanken für die Synchronisation vorbereitet werden** durch:
 - a. Für **Was möchten Sie mit dem SQL-Skript tun, das erstellt wird, um die konsolidierte Datenbank für die Synchronisation vorzubereiten?** wählen Sie **Mit der konsolidierten Datenbank ausführen**.

- b. Für **Was möchten Sie mit dem SQL-Skript tun, das erstellt wird, um die entfernte Datenbank für die Synchronisation vorzubereiten?** wählen Sie **Mit einer neuen entfernten Datenbank ausführen**.
 - c. Klicken Sie auf **Weiter**.
8. Auf der Seite **Auswahl überprüfen** können Sie die Auswahlen überprüfen, die Sie im Assistenten getroffen haben. Sie können auch die SQL-Anweisungen überprüfen, die für Ihre Datenbanken mit den Schaltflächen **Ansicht** ausgeführt werden.

Klicken Sie auf **Fertig stellen**.

Eine entfernte Datenbank namens *sync_mlintro_remote.db* wird erstellt.

Ergebnisse

Sie haben die entfernte Datenbank erfolgreich erstellt und bereitgestellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 8: Starten des MobiLink-Servers](#)“ auf Seite 101.

Siehe auch

- „[Deployment des Synchronisationsmodells](#)“ auf Seite 50
- „[SQL Anywhere-Datenbanken als entfernte Datenbanken verwenden](#)“ [*MobiLink - Clientadministration*]

Lektion 8: Starten des MobiLink-Servers

In dieser Lektion starten Sie den MobiLink-Server mit der `mlsrv16`-Option `-c`, um eine Verbindung mit der konsolidierten Datenbank herzustellen. Sie verwenden zusätzliche Optionen, um das MobiLink-Serververhalten zu konfigurieren.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 90.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Einführung in MobiLink](#)“ auf Seite 89.

Aufgabe

1. Gehen Sie über die Eingabeaufforderung zum Verzeichnis `c:\MLintro\mlintro_project\sync_mlintro_deploy`.
2. Führen Sie den folgenden Befehl aus:

```
mlsrv.bat
```

Hinweis

Die Option -zu verringert die Sicherheit Ihres Servers, da allen Benutzern eine Verbindung erlaubt wird. Daher sollte diese Option in einer Produktionsumgebung nicht verwendet werden.

Ergebnisse

Der MobiLink-Server wird gestartet.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 9: Starten der MobiLink-Clients](#)“ auf Seite 102.

Lektion 9: Starten der MobiLink-Clients

In dieser Lektion starten Sie die entfernte Datenbank, um sie für die Synchronisation vorzubereiten. In dieser Lektion wird davon ausgegangen, dass sich die entfernte Datenbank, die konsolidierte Datenbank und der MobiLink-Server auf demselben Computer befinden.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 90.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Einführung in MobiLink](#)“ auf Seite 89.

Aufgabe

1. Gehen Sie über die Eingabeaufforderung zum Verzeichnis `c:\MLintro\mlintro_project\sync_mlintro_deploy`.
2. Führen Sie den folgenden Befehl aus, um die `sync_mlintro_remote`-Datenbank zu starten:

```
dbsrv16 sync_mlintro_remote
```

Ergebnisse

Die entfernte Datenbank wird gestartet.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 10: MobiLink-Profiler starten](#)“ auf Seite 102.

Lektion 10: MobiLink-Profiler starten

In dieser Lektion starten und konfigurieren Sie den MobiLink-Profiler, um Synchronisationen zu überwachen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 90.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Einführung in MobiLink](#)“ auf Seite 89.

Kontext und Bemerkungen

Der MobiLink-Profiler kann verwendet werden, um statistische Informationen zu Synchronisationen zu sammeln. Das Diagramm zeigt Aufgaben auf der vertikalen Achse und den Zeitablauf auf der horizontalen Achse an. Mit dem MobiLink-Profiler können Sie Synchronisationen, die zu Fehlern führen bzw. bestimmte Bedingungen erfüllen, schnell identifizieren. Da sich der MobiLink-Profiler nicht signifikant auf die Performance auswirkt, wird seine Verwendung sowohl für die Entwicklung als auch für Produktionsumgebungen empfohlen.

Aufgabe

1. Klicken Sie auf **Start » Programme » SQL Anywhere 16 » Administrationstools » MobiLink-Profiler**.
2. Verbinden Sie den MobiLink-Profiler mit dem MobiLink-Server. Wählen Sie **Datei » Profiler-Sitzung beginnen**.

Das Fenster **Mit dem MobiLink-Server verbinden** wird angezeigt.

3. Akzeptieren Sie im Feld **Benutzer** den Standard-Benutzernamen **monitor_user**. Im Feld **Kennwort** geben Sie **sql** ein. Klicken Sie auf **OK**.

Ergebnisse

Dieser Benutzer wird automatisch hinzugefügt, weil Sie den MobiLink-Server in einer früheren Lektion mit der Option -zu+ gestartet haben. Der Profiler beginnt mit der Sammlung von Synchronisationsdaten.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 11: Synchronisieren](#)“ auf Seite 103.

Siehe auch

- „[MobiLink-Profiler](#)“ [[MobiLink - Serveradministration](#)]

Lektion 11: Synchronisieren

In dieser Lektion synchronisieren Sie die entfernte Datenbank mit der konsolidierten Datenbank mithilfe des Dienstprogramms dbmlsync.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 90.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Einführung in MobiLink](#)“ auf Seite 89.

Aufgabe

1. Gehen Sie über die Eingabeaufforderung zum Verzeichnis `c:\MLintro\mlintro_project\sync_mlintro_deploy`.
2. Führen Sie den folgenden Befehl aus, um die **sync_mlintro_remote**-Datenbank zu synchronisieren:

```
sync.bat "server=sync_mlintro_remote;UID=DBA;PWD=sql"
```

Ein Fenster mit sämtlichen Informationen über die **sync_mlintro_remote**-Clientsynchronisation mit der konsolidierten Datenbank wird angezeigt. Die Informationen werden in der Datei `remote.dbs` gespeichert, auf die nach dem Schließen des Client-Synchronisationsfensters zugegriffen werden kann.

3. Schließen Sie das Client-Synchronisationsfenster.

Klicken Sie auf **Herunterfahren**.

4. Klicken Sie im MobiLink-Profiler auf die Schaltfläche **Pause**, um das horizontale Blättern zu stoppen, und verwenden Sie dann den unteren Fensterausschnitt, um zur letzten Synchronisation zurückzublättern.
5. Sehen Sie sich die Eigenschaften der letzten Synchronisation an.

Doppelklicken Sie auf den farbigen Senkrechtrich, um die Synchronisationseigenschaften anzuzeigen.

Ergebnisse

Die entfernte Datenbank wird mit der konsolidierten Datenbank synchronisiert.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 12: Prüfen auf Fehler und Warnungen mit der Logdatei-Anzeige für MobiLink-Server](#)“ auf Seite 105.

Siehe auch

- „[MobiLink SQL Anywhere Client-Dienstprogramm \(dbmlsync\)](#)“ [[MobiLink - Clientadministration](#)]

Lektion 12: Prüfen auf Fehler und Warnungen mit der Logdatei-Anzeige für MobiLink-Server

Nachdem die Tabellen synchronisiert wurden, können Sie den Fortschritt der Synchronisationen in den Meldungslogdateien anzeigen, die Sie über die Befehlszeilen erstellt haben, also *mlsrv.mls* und *remote.dbs*. Standardmäßig werden die Dateien in dem Verzeichnis gespeichert, in dem der Befehl ausgeführt wurde.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 90.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Einführung in MobiLink](#)“ auf Seite 89.

Aufgabe

1. In Sybase Central klicken Sie auf **Extras » MobiLink 16 » Anzeige von MobiLink-Server-Logdateien**.

2. Öffnen Sie Ihre Logdatei in einem Texteditor.

Gehen Sie zu `c:\MLintro\mlintro_project\sync_mlintro_deploy\mlsrv.mls` und klicken Sie auf **Öffnen**.

Das Fenster **Logdatei-Anzeige für MobiLink-Server** wird angezeigt.

3. Klicken Sie auf die Registerkarte **Synchronisationen**, um nach Fehlern und Warnungen zu suchen, die während der Synchronisationen aufgetreten sind.
4. Klicken Sie auf die Registerkarte **Meldungen**, um nach Fehlern und Warnungen zu suchen, die vom MobiLink-Server gemeldet wurden.

Deaktivieren Sie die Option **Informationen anzeigen** und klicken Sie auf **Übernehmen**.

Nur Synchronisationen, die Fehler und Warnungen enthalten, werden im Fensterausschnitt **Meldungen** angezeigt. Beispiel: Sie erhalten eine Warnung, die Folgendes anzeigt:

```
[10093] Der MobiLink-Server wird aktuell mit -zf ausgeführt, wodurch  
seine Performance verringert wird
```

5. Klicken Sie auf die Registerkarte **Übersichten**, um nach allgemeinen Statistiken zu suchen, die in der Logdatei aufgelistet sind.
6. Öffnen Sie eine Clientlogdatei, wie etwa *remote.dbs* in einem Texteditor.
7. Durchsuchen Sie die linke Seite der Datei von oben nach unten. Ein Fehler ist aufgetreten, wenn Sie eine Zeile sehen, die mit einem **E.** beginnt. Die Synchronisation wurde erfolgreich abgeschlossen, wenn die Logdatei keine Fehler enthält.

Ergebnisse

Sie haben den Fortschritt der Synchronisationen in den Meldungslogdateien angezeigt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 13: Überwachung von Ressourcen mit dem SQL Anywhere-Monitor](#)“ auf Seite 106.

Siehe auch

- „[Protokollierung für den MobiLink-Server](#)“ [*MobiLink - Serveradministration*]

Lektion 13: Überwachung von Ressourcen mit dem SQL Anywhere-Monitor

In dieser Lektion haben Sie die Überwachung eines MobiLink-Servers und einer MobiLink-Serverfarm eingerichtet. Diese praktische Einführung verwendet die Monitor Developer Edition.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 90.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Einführung in MobiLink](#)“ auf Seite 89.

Aufgabe

1. Starten Sie den Monitor. Die folgenden Schritte gelten unter der Annahme, dass der Monitor derzeit nicht im Hintergrund ausgeführt wird.

So starten Sie die Monitor Developer Edition (Windows) Klicken Sie auf **Start » Programme » SQL Anywhere 16 » Administrationstools » SQL Anywhere-Monitor**.

So starten Sie Monitor Developer Edition (Linux) Führen Sie das Skript `samonitor.sh` im Verzeichnis `bin32` oder `bin64` im Monitor-Installationsverzeichnis aus:

```
samonitor.sh launch
```

Der Monitor startet die Sammlung der Metriken und ein Browser öffnet die Standard-URL, über die Sie sich beim Monitor anmelden können: `http://localhost:4950`.

Hinweis

Wenn Sie über ein Netzwerk auf den Monitor zugreifen, navigieren Sie zu **`http://computer-name:4950`**, wobei `computer-name` der Name des Computers ist, auf dem der Monitor läuft.

2. Melden Sie sich beim Monitor als *Standardadministrator* an.

Im Feld **Benutzername** geben Sie **admin** und im Feld **Kennwort** geben Sie **admin** ein.

Hinweis

Sie müssen beim Monitor als Administrator angemeldet sein, um die folgenden Schritte auszuführen. Benutzer ohne Schreibrechte und Operatoren haben nicht die erforderlichen Privilegien zum Ausführen aller Aufgaben.

Prüfen Ihres Monitor-Benutzertyps

1. Melden Sie sich beim Monitor an.
2. Klicken Sie auf **Extras » Benutzereinstellungen** und überprüfen Sie die Einstellung **Benutzertyp**.

Siehe „Monitor-Benutzer“ [[SQL Anywhere Server - Datenbankadministration](#)].

3. Hinzufügen einer MobiLink-Serverressource zum Monitor:
 - a. Im linken Navigationsmenü klicken Sie auf **Extras » Administration**.
 - b. Klicken Sie auf **Ressourcen** und auf **Hinzufügen**.
 - c. Klicken Sie auf **MobiLink-Server** und auf **Weiter**.
 - d. Im Feld **Name** geben Sie **MobiLinkServerSample** ein und klicken auf **Weiter**.
 - e. Im Feld **Host** geben Sie **localhost** ein und klicken auf **Weiter**.
 - f. Wenn Sie zur Eingabe der Autorisierung aufgefordert werden, geben Sie im Feld **Benutzername** einen Benutzernamen wie **monitor_user** und im Feld **Kennwort** ein Kennwort wie **sql** ein.

Diese Anmeldeinformationen werden verwendet, um einen Benutzer auf dem MobiLink-Server zu erstellen. Der Monitor speichert diese Benutzer-ID und das Kennwort und verwendet sie für die Verbindung mit dem zu überwachenden MobiLink-Server.
 - g. Klicken Sie auf **Erstellen**.
 - h. Die neue Ressource **MobiLinkServerSample** wird erstellt und die Überwachung beginnt.
 - i. Klicken Sie auf **Schließen**.
 - j. Klicken Sie auf **Schließen**.
 - k. Klicken Sie auf **Übersicht » Ressourcenliste**. Klicken Sie auf **MobiLinkServerSample**, um ein Dashboard für die Ressource zu erstellen und zu öffnen.
4. In dieser Lektion fügen Sie eine MobiLink-Serverfarm-Ressource hinzu, um zwei MobiLink-Server zu überwachen:
 - a. Fügen Sie zwei MobiLink-Server als zu überwachende Ressourcen hinzu. Verwenden Sie als erste Ressource die Ressource **MobiLinkServerSample**, die Sie in der vorherigen Lektion hinzugefügt haben.

Fügen Sie eine zweite MobiLink-Serverressource hinzu:

- i. Führen Sie an einer Eingabeaufforderung den folgenden Befehl aus, um einen MobiLink-Server zu starten, der an Port 8039 auf Verbindungsanforderungen wartet:

```
mlsrv16 -vcrs -zu+ -c "DSN=mlintro_consdb" -ot ml_tcpip.txt -zs  
ml_tcpip -x tcpip{port=8039}
```

- ii. Klicken Sie auf **Extras » Administration**.
- iii. Klicken Sie auf **Ressourcen** und auf **Hinzufügen**.
- iv. Klicken Sie auf **MobiLink-Server** und auf **Weiter**.
- v. Im Feld **Name** geben Sie **ml_tcpip** ein und klicken auf **Weiter**.
- vi. Im Feld **Host** geben Sie **localhost** ein.

Im Feld **Port** geben Sie **8039** ein und klicken auf **Weiter**.

- vii. Wenn Sie zur Eingabe der Autorisierung aufgefordert werden, geben Sie im Feld **Benutzername** einen Benutzernamen wie **monitor_user** und im Feld **Kennwort** ein Kennwort wie **sql** ein.

Diese Anmeldeinformationen werden verwendet, um einen Benutzer auf dem MobiLink-Server zu erstellen. Der Monitor speichert diese Benutzer-ID und das Kennwort und verwendet sie für die Verbindung mit dem zu überwachenden MobiLink-Server.

- viii. Klicken Sie auf **Erstellen**.

Die Ressource **ml_tcpip** wird der **Ressourcenliste** im Dashboard **Übersicht** hinzugefügt.

- ix. Klicken Sie auf **Schließen**.
- x. Klicken Sie auf **Schließen**.
- b. Fügen Sie die MobiLink-Serverfarm-Ressource hinzu. Weitere Hinweise zu MobiLink-Serverfarmen finden Sie unter „[MobiLink-Server in einer Serverfarm](#)“ [[MobiLink - Serveradministration](#)].
 - i. Öffnen Sie das Fenster **Administration**.

Klicken Sie auf **Extras » Administration**.

- ii. Klicken Sie auf **Ressourcen** und auf **Hinzufügen**.
- iii. Klicken Sie auf **MobiLink-Server-Farm** und anschließend auf **Weiter**.
- iv. Im Feld **Name** geben Sie **MobiLink_Test_Farm** ein und klicken auf **Weiter**.
- v. Klicken Sie auf **MobiLinkServerSample** und **ml_tcpip**, und klicken Sie auf **Erstellen**.
- vi. Klicken Sie auf **Schließen**.
- vii. Klicken Sie auf **Schließen**.
- c. Klicken Sie auf **Dashboards » Übersicht**.

Die Ressource **MobiLink_Test_Farm** erscheint in der **Ressourcenliste**.

Die MobiLink-Serverressourcen bleiben in der **Ressourcenliste**.

 - d. Klicken Sie auf den Pfeil links neben **MobiLink_Test_Farm**, um die Liste der MobiLink-Serverressourcen anzuzeigen, die in der Farm enthalten sind.

- e. Klicken Sie auf **MobiLink_Test_Farm**, um das **MobiLink_Test_Farm**-Dashboard zu öffnen und die gesammelten Metriken anzuzeigen.

Die Widgets **Alarmliste**, **Ressourcen-Widget** und **Serverinformationen** sollten angezeigt werden.

5. Hinweise zum Testen eines Alarms und zu zusätzlichen Monitor-Funktionen finden Sie unter [„Praktische Einführung: Überwachen von Ressourcen mit dem Monitor“ \[SQL Anywhere Server - Datenbankadministration\]](#).

Ergebnisse

Sie haben den SQL Anywhere-Monitor zum Überwachen eines MobiLink-Servers und einer MobiLink-Serverfarm verwendet.

Nächste Schritte

Gehen Sie weiter zu [„Aufräumen“ auf Seite 109](#).

Aufräumen

Entfernen Sie alle Daten der praktischen Einführung von Ihrem Computer.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank“ auf Seite 90](#).

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Einführung in MobiLink“ auf Seite 89](#).

Aufgabe

1. Schließen Sie alle Instanzen der folgenden Anwendungen:
 - MobiLink-Profiler
 - SQL Anywhere-Monitor für MobiLink
 - Sybase Central
 - Interactive SQL
2. Schließen Sie SQL Anywhere, MobiLink und den Synchronisationsclient.
3. Löschen Sie alle Datenquellen im Zusammenhang mit der praktischen Einführung:
 - a. Starten Sie den ODBC-Datenquellen-Administrator.
 - b. Wählen Sie **Start » Programme » SQL Anywhere 16 » Administrationstools » ODBC-Datenquellen-Administrator**.

- c. In der Liste **Benutzerdatenquellen** wählen Sie **mlintro_consdb** und klicken Sie dann auf **Entfernen**.
4. Löschen Sie das Verzeichnis mit Ihren konsolidierten und entfernten Datenbanken.

Ergebnisse

Die Daten der praktischen Einführung werden von Ihrem Computer entfernt.

Praktische Einführung: MobiLink mit einer konsolidierten SQL Anywhere-Datenbank verwenden

Diese praktische Einführung zeigt, wie Sie eine SQL Anywhere-Datenbank mit MobiLink für die mobile Verwendung einrichten. Dabei wird die Synchronisation zwischen einer konsolidierten SQL Anywhere-Datenbank und einer entfernten UltraLite-Datenbank eingerichtet. Sie können auch eine entfernte SQL Anywhere-Datenbank verwenden.

Der Zweck dieser praktischen Einführung ist die Einrichtung einer mobilen Verwendung von Daten für ein Mobilfunkunternehmen, das in vielen Regionen tätig ist. In diesem Szenario gilt Folgendes für jede einzelne Region:

- Es handelt sich um eine entfernte Synchronisationsumgebung.
- Sie weist eine lokale UltraLite-Datenbank auf, die mit der konsolidierten SQL Anywhere-Datenbank an einem zentralen Standort mit MobiLink synchronisiert wird.
- Sie kann an ihrem Standort auf Produktinformationen zugreifen und die Daten aus der entfernten Datenbank manipulieren, wenn ein neuer Kunde ein Konto aktiviert oder bestehender Kunde ein neues mobiles Gerät aktiviert.

Erforderliche Software

- SQL Anywhere 16

Privilegien

Sie müssen die folgenden Rollen und Privilegien für die konsolidierte Datenbank haben:

- SYS_AUTH_DBA_ROLE-Kompatibilitätsrolle

Übersicht

In dieser praktischen Einführung wird Folgendes veranschaulicht:

- Berücksichtigen wichtiger Überlegungen beim Planen eines entfernten Schemas, wie etwa die Synchronisationsrichtungen für entfernte Tabellen.
- Hinzufügen eindeutiger Primärschlüssel zu konsolidierten und entfernten Datenbanken
- Einrichten der Synchronisation zwischen einer konsolidierten Datenbank und einer entfernten Datenbank mithilfe des **Assistenten zum Erstellen eines Synchronisationsmodells**

- Passen Sie die Synchronisationseinstellungen mit Sybase Central an.
- Deployment einer konsolidierten Datenbank und einer entfernten Datenbank mithilfe des **Assistenten für das Deployment eines Synchronisationsmodells**
- Synchronisieren des entfernten Clients mit der konsolidierten Datenbank

Siehe auch

- [„MobiLink-Synchronisation“ auf Seite 1](#)

Lektion 1: Planen von Schemata

In dieser praktischen Einführung wird davon ausgegangen, dass die Beispieldatenbank auf dem Computer installiert ist, auf dem SQL Anywhere ausgeführt wird.

Die Beispieldatenbank wird als konsolidierte Datenbank verwendet. Die folgende Tabelle enthält eine Beschreibung der einzelnen Tabellen in der konsolidierten SQL Anywhere-Datenbank:

Tabelle	Beschreibung
Customers	Kunden, deren Informationen gespeichert sind
SalesOrders	Datensätze von Kontoaktivierungen.
Products	Datensätze aller erhältlichen Produkte.
CustomerProducts	Eine Liste der Produkte, die den einzelnen Kunden gehören.

Planung des entfernten Schemas

Es wäre nicht effizient, wenn jede Region eine Kopie der gesamten konsolidierten Datenbank hätte, und dies ist auch nicht erforderlich. Das entfernte Schema verwendet dieselben Tabellennamen, doch es enthält nur die für eine bestimmte Region relevanten Informationen. Um diese Konfiguration zu erreichen, ist das entfernte Schema als Teilmenge der konsolidierten Datenbank wie folgt eingerichtet:

Konsolidierte Tabelle	Entfernte Tabelle
Customers	Filtern Sie nach Region.
SalesOrders	Filtern Sie nach Kunden-ID für Kunden in der entsprechenden Region.
Products	Beziehen Sie alle Zeilen ein.
CustomerProducts	Filtern Sie nach Kunden-ID für Kunden in der entsprechenden Region.

Jeder Vertriebsmitarbeiter muss Informationen über die Kunden speichern, die sich in ihrer Region befinden, sowie die Produkte, die allen Kunden angeboten werden. Jedoch benötigt ein Vertriebsmitarbeiter keine Informationen zu Kunden in anderen Regionen, sodass diese Informationen

nicht mit jeder regionalen Niederlassung synchronisiert werden. Aus diesem Grund werden die Zeilen basierend auf der Regions-ID gefiltert.

Hinweis

Sie können auch eine Teilmenge von Spalten aus einer Tabelle verwenden, wenn bestimmte Spalten in den entfernten Datenbanken nicht benötigt werden.

Im nächsten Schritt wird die Synchronisationsrichtung der einzelnen Tabellen ausgewählt. Sie sollten untersuchen, welche Informationen von einer entfernten Datenbank gelesen werden müssen und welche Informationen eine entfernte Datenbank erstellen, ändern oder entfernen muss. In diesem Beispiel muss eine Region Zugriff auf die Liste der Produkte haben, die Kunden angeboten werden, doch sie gibt nie ein neues Produkt in das System ein. Sie legt die Einschränkung fest, dass Produkte immer über die konsolidierte Datenbank am zentralen Standort in das System eingegeben werden müssen. Ein Vertriebsmitarbeiter muss jedoch in der Lage sein, regelmäßig neue Kontoaktivierungen einzugeben. Diese Faktoren führen zu folgenden Synchronisationsrichtungen für die Tabellen:

Tabelle	Synchronisation
Customers	Nur in konsolidierte Datenbank hochladen.
SalesOrders	Nur in konsolidierte Datenbank hochladen.
Products	Nur auf entfernte Datenbank herunterladen.
CustomerProducts	Nur in konsolidierte Datenbank hochladen.

Gehen Sie weiter zu [„Lektion 2: Vorbereiten der konsolidierten Datenbank“](#) auf Seite 112.

Lektion 2: Vorbereiten der konsolidierten Datenbank

In dieser Lektion verbinden Sie sich mit der konsolidierten Datenbank, erstellen die CustomerProducts-Tabelle und ändern die Customers-Tabelle dahingehend, dass sie regionale Informationen enthält.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Planen von Schemata“](#) auf Seite 111.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: MobiLink mit einer konsolidierten SQL Anywhere-Datenbank verwenden“](#).

Kontext und Bemerkungen

In einem Synchronisationssystem ist der Primärschlüssel einer Tabelle die einzige Möglichkeit, eine Zeile in einer anderen Datenbank eindeutig zu identifizieren, und die einzige Möglichkeit, Konflikte zu erkennen. Jede für die mobile Verwendung eingerichtete Tabelle muss einen Primärschlüssel haben. Der

Primärschlüssel darf nie geändert werden. Sie müssen außerdem sicherstellen, dass ein in einer Datenbank eingefügter Primärschlüsselwert in keiner weiteren Datenbank eingefügt wird.

In einer späteren Lektion wird das entfernte Schema aus dem konsolidierten Schema erstellt, sodass das entfernte Schema dieselben Primärschlüssel enthält wie das konsolidierte Schema.

Spalten wurden gezielt ausgewählt, um eindeutige Primärschlüssel für alle Datenbanken sicherzustellen. Für die Customers-Tabelle besteht der Primärschlüssel aus der ID-Spalte. Jeder in der entfernten Customers-Tabelle eingefügte Wert muss eine eindeutige Kunden-ID-Nummer haben (der Region-Wert ist immer gleich). Dadurch wird die Eindeutigkeit in jeder entfernten Customers-Tabelle sichergestellt. Der Primärschlüssel in der konsolidierten Customers-Tabelle verhindert Konflikte, wenn mehrere Vertriebsmitarbeiter gleichzeitig Daten einlesen. Jeder Upload aus einer Region ist gegenüber anderen Regionen eindeutig, da ihre Region-Werte unterschiedlich sind.

Aufgabe

1. Klicken Sie auf **Start » Programme » SQL Anywhere 16 » Administrationstools » Sybase Central**.
2. Klicken Sie auf **Verbindungen » Verbinden mit SQL Anywhere 16**.
3. Führen Sie im Fenster **Verbinden** die folgenden Aufgaben durch:
 - a. Wählen Sie in der Dropdown-Liste **Aktion** die Option **Mit einer ODBC-Datenquelle verbinden**.
 - b. Im Feld **ODBC-Datenquellenname** geben Sie **SQL Anywhere 16 Demo** ein.
 - c. Klicken Sie auf **Verbinden**.
4. Verbinden Sie sich über Interactive SQL mit Ihrer konsolidierten Datenbank.

Führen Sie an der Eingabeaufforderung folgenden Befehl aus:

```
dbisql -c "DSN=SQL Anywhere 16 Demo"
```

5. Führen Sie in Interactive SQL die folgenden Anweisungen aus, um Daten zu erstellen und in der CustomerProducts-Tabelle einzufügen:

```
CREATE TABLE CustomerProducts
  (ID int default AUTOINCREMENT PRIMARY KEY,
   SalesOrderID int NOT NULL,
   CustomerID int NOT NULL,
   ProductID int);

INSERT INTO CustomerProducts (SalesOrderID, CustomerID, ProductID)
  SELECT SalesOrders.ID, SalesOrders.CustomerID,
  SalesOrderItems.ProductID
  FROM SalesOrders, SalesOrderItems
  WHERE SalesOrders.ID = SalesOrderItems.ID;
```

6. Führen Sie in Interactive SQL die folgenden Anweisungen aus, um der Customers-Tabelle regionale Informationen für jeden Kunden hinzuzufügen:

```
ALTER TABLE Customers
  ADD Region VARCHAR(255);
```

```
UPDATE Customers
  SET Region = (SELECT TOP 1 SalesOrders.Region
                FROM SalesOrders
                WHERE Customers.ID = SalesOrders.CustomerID
                ORDER BY Region);
COMMIT;
```

Ergebnisse

Eine Verbindung wird zur SQL Anywhere-Demo-Datenbank hergestellt, eine Tabelle namens CustomerProducts wird erstellt und die Customers-Tabelle wird so geändert, dass sie regionale Informationen enthält.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 3: Erstellen eines Synchronisationsmodells](#)“ auf Seite 114.

Siehe auch

- „MobiLink - konsolidierte Datenbanken“ [[MobiLink - Serveradministration](#)]
- „Eindeutige Primärschlüssel“ [[MobiLink - Serveradministration](#)]

Lektion 3: Erstellen eines Synchronisationsmodells

In dieser Lektion erstellen Sie ein neues MobiLink-Projekt mithilfe des **Assistenten zum Erstellen eines Projekts**. Der **Assistent zum Erstellen eines Projekts** erstellt auch ein Synchronisationsmodell mithilfe der Standardwerte, die später bearbeitet werden können.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: MobiLink mit einer konsolidierten SQL Anywhere-Datenbank verwenden](#)“.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 111.

Aufgabe

1. Klicken Sie in **Sybase Central** auf **Extras » MobiLink 16 » Neues Projekt**.
2. Der **Assistent zum Erstellen eines Projekts** wird angezeigt.
3. Im Feld **Wie lautet der Name des neuen Projekts?** geben Sie **mlsqla_project** ein.
4. Geben Sie im Feld **Wo soll das neue Projekt gespeichert werden?** den Pfad **C:\mlsqla** ein und klicken Sie auf **Weiter**.
5. Im Feld **Anzeigename der Datenbank** geben Sie **demo** ein.

6. Klicken Sie auf **Bearbeiten**.
7. Führen Sie die folgenden Aufgaben auf der Seite **Mit einer allgemeinen ODBC-Datenbank verbinden** durch:
 - a. Klicken Sie im Feld **ODBC-Datenquellennamen** auf **Durchsuchen** und wählen Sie **SQL Anywhere 16 Demo**.
 - b. Klicken Sie auf **OK** und auf **Speichern**.
8. Wählen Sie die Option **Kennwort speichern** aus und klicken Sie auf **Weiter**.
9. Wählen Sie auf der Seite **Neues entferntes Datenbankschema** in der Liste **Welche konsolidierten Datenbanktabellen und Spalten sollen in der entfernten Datenbank enthalten sein?** die folgenden Tabellen:
 - CustomerProducts
 - Customers
 - Products
 - SalesOrders

Klicken Sie auf **Weiter**.
10. Wählen Sie die Option **Dem Projekt einen Namen für ein entferntes Schema hinzufügen** aus.
11. Im Feld **Wie lautet der Name des neuen entfernten Schemas?** geben Sie **mysql_remote_schema** ein. Klicken Sie anschließend auf **Weiter**.
12. Wählen Sie **UltraLite** für **Welcher entfernter Datenbanktyp soll verwendet werden** aus und klicken Sie auf **Fertig stellen**.
13. Klicken Sie bei Aufforderung zur Installation der MobiLink-Setupskripten auf **Ja**.
14. Klicken Sie auf **Ja**, wenn Sie dazu aufgefordert werden, das entfernte Schema zu importieren.
15. Klicken Sie auf **OK**.
16. Klicken Sie mit der rechten Maustaste auf das neue Synchronisationsmodell und wählen Sie **Eigenschaften**.
 - a. Geben Sie **sync_mysql** in das erste Feld ein.
 - b. Geben Sie **sync_mysql_publication** in das Feld **Publikationsname** ein.
 - c. Geben Sie **sync_mysql_scriptversion** in das Feld **Skriptversion** ein.

Die Publikation ist das Objekt in der entfernten Datenbank, das festlegt, welche Daten synchronisiert werden. MobiLink-Serverskripten legen fest, wie hochgeladene Daten von entfernten Datenbanken in der konsolidierten Datenbank verwendet werden und wie Skriptversionen Skripten gruppieren. Sie können unterschiedliche Skriptversionen für verschiedene Anwendungen verwenden. Auf diese Weise können Sie einen einzelnen MobiLink-Server verwalten und gleichzeitig mehrere Anwendungen synchronisieren.
 - d. Klicken Sie auf **Übernehmen** und dann auf **OK**.

17. Führen Sie im rechten Fensterausschnitt von Sybase Central die folgenden Aufgaben aus:

- a. Klicken Sie auf die Registerkarte **Ereignisse**.
- b. Aktualisieren Sie den CustomerProducts-Download-Cursor, um nur Produkte für Kunden in der Region "Eastern" herunterzuladen.

Ersetzen Sie das bestehende SQL-Skript für das download_cursor-Ereignis für die CustomerProducts-Tabelle durch die folgende Abfrage:

```
SELECT "DBA"."CustomerProducts"."ID",
       "DBA"."CustomerProducts"."SalesOrderID",
       "DBA"."CustomerProducts"."CustomerID",
       "DBA"."CustomerProducts"."ProductID"
FROM   "DBA"."CustomerProducts"
INNER JOIN "GROUPO"."Customers" ON "GROUPO"."Customers"."ID" =
       "DBA"."CustomerProducts"."CustomerID"
WHERE  "GROUPO"."Customers"."Region" = 'Eastern';
```

- c. Aktualisieren Sie den Download-Cursor für die Customers-Tabelle, um nur Kundendaten in der Region "Eastern" herunterzuladen.

Ersetzen Sie das bestehende SQL-Skript für das download_cursor-Ereignis für die Customers-Tabelle durch die folgende Abfrage:

```
SELECT "GROUPO"."Customers"."ID",
       "GROUPO"."Customers"."Surname",
       "GROUPO"."Customers"."GivenName",
       "GROUPO"."Customers"."Street",
       "GROUPO"."Customers"."City",
       "GROUPO"."Customers"."State",
       "GROUPO"."Customers"."Country",
       "GROUPO"."Customers"."PostalCode",
       "GROUPO"."Customers"."Phone",
       "GROUPO"."Customers"."CompanyName",
       "GROUPO"."Customers"."Region"
FROM   "GROUPO"."Customers"
WHERE  Region = 'Eastern';
```

- d. Aktualisieren Sie den SalesOrders-Download-Cursor, um nur Bestellinformationen für Kunden in der Region "Eastern" herunterzuladen.

Ersetzen Sie das bestehende SQL-Skript für das download_cursor-Ereignis für die SalesOrders-Tabelle durch die folgende Abfrage:

```
SELECT "GROUPO"."SalesOrders"."ID",
       "GROUPO"."SalesOrders"."CustomerID",
       "GROUPO"."SalesOrders"."OrderDate",
       "GROUPO"."SalesOrders"."FinancialCode",
       "GROUPO"."SalesOrders"."Region",
       "GROUPO"."SalesOrders"."SalesRepresentative"
FROM   "GROUPO"."SalesOrders"
WHERE  "GROUPO"."SalesOrders"."Region" = 'Eastern'
AND    "GROUPO"."SalesOrders"."ID" IN
      (SELECT "DBA"."CustomerProducts"."SalesOrderID"
       FROM   "DBA"."CustomerProducts");
```

18. Speichern Sie das Synchronisationsmodell.

Klicken Sie auf **Datei » Speichern**.

Das Synchronisationsmodell ist vollständig und kann nun per Deployment bereitgestellt werden.

Ergebnisse

Ein MobiLink-Projekt- und -Synchronisationsmodell wird erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 4: Erstellen eines Synchronisationsmodells](#)“ auf Seite 117.

Siehe auch

- „Synchronisationsmodelle“ auf Seite 30
- „Einrichten einer konsolidierten Datenbank“ [*MobiLink - Serveradministration*]
- „Systemtabellen des MobiLink-Servers“ [*MobiLink - Serveradministration*]
- „Systemprozeduren des MobiLink-Servers“ [*MobiLink - Serveradministration*]
- „download_delete_cursor-Skripten“ [*MobiLink - Serveradministration*]
- „Publikationen“ [*MobiLink - Clientadministration*]
- „Synchronisationsmodellaufgaben“ auf Seite 33
- „Downloadtyp ändern“ auf Seite 38
- „Tabellen- und Spaltenzuordnungen“ auf Seite 34

Lektion 4: Erstellen eines Synchronisationsmodells

Sie können mit dem **Assistenten für das Deployment eines Synchronisationsmodells** das Deployment der konsolidierten Datenbank und der entfernten Datenbank vornehmen. Sie können die beiden Datenbanken entweder einzeln oder zusammen bereitstellen. Der **Assistent für das Deployment eines Synchronisationsmodells** führt Sie durch die Schritte für die Konfiguration der Optionen für das Deployment.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: MobiLink mit einer konsolidierten SQL Anywhere-Datenbank verwenden](#)“.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 111.

Aufgabe

1. Erweitern Sie im linken Fensterausschnitt von Sybase Central unter **MobiLink 16** die Elemente **mlsqla_project**, **Synchronisationsmodelle** und dann **sync_mlsqla**.

2. Klicken Sie auf **Datei » Speichern**.

Der **Assistent zum Deployment eines Synchronisationsmodells** wird geöffnet. Wählen Sie den Standardspeicherort für die generierten Dateien aus und klicken Sie auf **Weiter**.

3. Akzeptieren Sie die Standardwerte auf der Seite **Client-Netzwerkoptionen** und klicken Sie auf **Weiter**.

4. Führen Sie auf der Seite **MobiLink-Benutzer und -Kennwort** die folgenden Aufgaben aus:
 - a. Bei **Welchen Benutzer und welches Kennwort für MobiLink möchten Sie verwenden?** wählen Sie **Diese verwenden**
 - b. Geben Sie im Feld **MobiLink-Benutzer** den Wert **mlsqli_remote** ein.
 - c. Geben Sie im Feld **MobiLink-Kennwort** den Wert **mlsqli_pass** ein.
 - d. Wählen Sie **Registrieren Sie diesen Benutzer in der konsolidierten Datenbank. Registrierte Benutzer sind berechtigt, die Synchronisation auszuführen.**
 - e. Klicken Sie auf **Weiter**.
5. Ändern Sie den Namen des Synchronisationsprofils in **mlsqli_remote_syncprofile** und klicken Sie auf **Weiter**.
6. Wählen Sie auf der Seite **Auswählen, wie Datenbanken für die Synchronisation vorbereitet werden** die Auswahl **Mit der konsolidierten Datenbank ausführen** für die Option **Was möchten Sie mit dem SQL-Skript tun, das erstellt wird, um die konsolidierte Datenbank für die Synchronisation vorzubereiten?**
7. Für die Option **Was möchten Sie mit dem SQL-Skript tun, das erstellt wird, um die entfernte Datenbank für die Synchronisation vorzubereiten?** wählen Sie **Mit einer neuen entfernten Datenbank ausführen**. Klicken Sie auf **Weiter**.
8. Um die SQL-Skripten zu überprüfen, die mit dem Assistenten erstellt wurden, klicken Sie auf **Ansicht**.
9. Klicken Sie auf **Fertig stellen**.
10. Klicken Sie auf **Schließen**.

Ergebnisse

Ihre konsolidierte Datenbank ist für die Synchronisation mit mehreren entfernten Clients konfiguriert und Sie haben erfolgreich ein Deployment auf einen entfernten Client vorgenommen. Wenn Sie andere entfernte Clients bereitstellen möchten, können Sie diesen Assistenten erneut ausführen. Denken Sie daran, einen neuen MobiLink-Benutzer zu erstellen und die Auswahl des Deployments von konsolidierter Datenbank und MobiLink-Server aufzuheben. Da die konsolidierten und entfernten Datenbanken bereits bereitgestellt wurden, müssen Sie nur noch das Deployment anderer entfernter Synchronisationsclients vornehmen.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 5: Starten des MobiLink-Servers“ auf Seite 119](#).

Siehe auch

- [„Deployment des Synchronisationsmodells“ auf Seite 50](#)
- [„SQL Anywhere-Datenbanken als entfernte Datenbanken verwenden“ \[MobiLink - Clientadministration\]](#)
- [„MobiLink-Benutzer“ \[MobiLink - Clientadministration\]](#)

Lektion 5: Starten des MobiLink-Servers

In dieser Lektion starten Sie den MobiLink-Server mit der `mlsrv16`-Option `-c`, um eine Verbindung mit der konsolidierten Datenbank herzustellen. Sie können zusätzliche Optionen verwenden, um das MobiLink-Serververhalten zu konfigurieren.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: MobiLink mit einer konsolidierten SQL Anywhere-Datenbank verwenden](#)“.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 111.

Starten des MobiLink-Servers

1. Gehen Sie über die Eingabeaufforderung zum Verzeichnis `c:\mlsqla`.
2. Stellen Sie eine Verbindung mit der konsolidierten Datenbank her, indem Sie den folgenden Befehl ausführen:

```
mlsrv16 -c "DSN=SQL Anywhere 16 Demo" -o mlsrv.mls -v+ -dl -zf -zu+ -x  
tcpip
```

Das MobiLink-Server-Meldungsfenster wird eingeblendet.

Nachfolgend finden Sie eine Beschreibung der einzelnen MobiLink-Serveroptionen, die in dieser praktischen Einführung verwendet werden. Die Optionen `-o`, `-v` und `-dl` bieten Informationen zur Fehlersuche und -behebung. Die Verwendung dieser Protokollierungsoptionen empfiehlt sich in einer Entwicklungsumgebung. Aus Gründen der Performance werden `-v+` und `-dl` in Produktionsumgebungen normalerweise nicht verwendet. Eine vollständige Liste der MobiLink-Serveroptionen finden Sie unter „[MobiLink-Serveroptionen](#)“ [[MobiLink - Serveradministration](#)].

Option	Beschreibung
-c	Steht vor der Verbindungszeichenfolge
-o	Legt die Meldungslogdatei <i>mlsrv.mls</i> fest.
-v+	Gibt an, welche Informationen protokolliert werden. Mit <code>-v+</code> wird die maximale ausführliche Protokollierung aktiviert.
-dl	Alle Lognachrichten auf dem Bildschirm anzeigen.
-zf	Gibt an, dass der MobiLink-Server zu Beginn jeder Synchronisation eine Prüfung auf Skriptänderungen durchführen soll.
-zu+	Fügt neue Benutzer automatisch hinzu.

Option	Beschreibung
-x	Legt das Kommunikationsprotokoll und Parameter für MobiLink-Clients fest

Hinweis

Die Optionen -zf und -zu+ sollten nur für Debugging und Entwicklungsarbeiten verwendet werden. In dieser praktischen Einführung ist die Option -zf erforderlich, damit Sie den Server nicht herunterfahren müssen, wenn in einer späteren Lektion der konsolidierten Datenbank neue Skripten hinzugefügt werden. Mit der Option -zu+ werden automatisch neue MobiLink-Benutzer zur Synchronisationsumgebung hinzugefügt.

Ergebnisse

Der MobiLink-Server wird gestartet und mit der konsolidierten Datenbank verbunden.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 6: Synchronisieren“ auf Seite 120](#).

Lektion 6: Synchronisieren

In dieser Lektion synchronisieren Sie den MobiLink-Client mit dem MobiLink-Server unter Verwendung des ulsync-Dienstprogramms zum Initiieren der Synchronisation.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: MobiLink mit einer konsolidierten SQL Anywhere-Datenbank verwenden“](#).

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Planen von Schemata“ auf Seite 111](#).

Aufgabe

1. Wechseln Sie zum Verzeichnis `c:\msqla\msqla_project\sync_msqla_deploy` und führen Sie den folgenden Befehl aus, um die Datenbank **sync_msqla_remote** zu synchronisieren:

```
ulsync -c "DBF=sync_msqla_remote.udb"  
"Publications=sync_msqla_publication;MobiLinkId=msqla_remote;MobiLinkPw  
d=msqla_pass;ScriptVersion=sync_msqla_scriptversion;Stream=tcip{port=24  
39}"
```

- **DBF** Gibt an, welche Datenbankdatei Sie laden und mit welcher Datenbankdatei Sie sich verbinden wollen, wenn eine Datenbank gestartet wird, die nicht läuft.
- **Publikationen** Die Publikation auf dem entfernten Gerät, das zur Ausführung der Synchronisation verwendet wird. (Diese Publikation wurde vom **Assistenten zum Erstellen eines Synchronisationsmodells** erstellt.)

- **MobiLinkUid** Der Benutzername, der zur Authentifizierung mit dem MobiLink-Server verwendet wird.
- **MobiLinkPwd** Das Kennwort, das zur Authentifizierung mit dem MobiLink-Server verwendet wird.
- **ScriptVersion** Die Skriptversion auf dem entfernten Gerät, das zur Ausführung der Synchronisation verwendet wird. (Diese Publikation wurde vom **Assistenten zum Erstellen eines Synchronisationsmodells** erstellt.)
- **Stream** Legt Optionen zum Konfigurieren des Netzwerkprotokolls fest.

Der Verarbeitungsfortschritt der Synchronisation wird im Meldungsfenster des MobiLink-Servers angezeigt. Wenn dieser Befehl erfolgreich ausgeführt wird, füllt die Anwendung ulsync die entfernte Datenbank mit einer Teilmenge der Informationen aus der konsolidierten Datenbank.

Wenn die Synchronisation fehlschlägt, überprüfen Sie die Verbindungsinformationen, die Sie an die ulsync-Anwendung übermittelt haben, sowie den MobiLink-Benutzernamen und das Kennwort. Wenn diese Informationen richtig sind, überprüfen Sie den verwendeten Publikationsnamen und stellen Sie sicher, dass die konsolidierte Datenbank und der MobiLink-Server ausgeführt werden. Sie können auch den Inhalt des Synchronisationslogs (Server und Client) überprüfen.

Hinweis

Wenn Sie die Anwendung ulsync auf einem anderen Computer als dem MobiLink-Server ausführen, müssen Sie Argumente mit der Position des MobiLink-Servers übergeben.

Nach der erfolgreichen Synchronisation des entfernten Clients mit der konsolidierten Datenbank durch den MobiLink-Server sollte die entfernte Datenbank mit den Informationen einer Region gefüllt sein. Sie können überprüfen, ob die Datenbank mit Daten gefüllt wurde, indem Sie in Sybase Central das SQL Anywhere 16-Plug-In verwenden.

2. Öffnen Sie Sybase Central.
3. Stellen Sie eine Verbindung zur entfernten Datenbank her:
 - a. Klicken Sie im linken Fensterausschnitt mit der rechten Maustaste auf **UltraLite 16** und klicken Sie auf **Verbinden**.
 - b. Geben Sie **DBA** als **Benutzer-ID** und **sql** als **Kennwort** ein.
 - c. Geben Sie im Feld **Datenbankdatei** Folgendes ein: **C:\mssqla\mssqla_project\sync_mssqla_deploy\sync_mssqla_remote.udb**.
 - d. Klicken Sie auf **Verbinden**.
4. Erweitern Sie im linken Fensterausschnitt **UltraLite 16**, **sync_mssqla_remote**, **Tabellen** und wählen Sie **Customers**.
5. Klicken Sie im rechten Fensterausschnitt auf das Register **Daten**.

Alle Datensätze in den Customers-Tabellen sind für die Kunden, die zur Region "Eastern" gehören. Diese einzelne Region hat nichts mit den Kundeninformationen anderer Regionen zu tun. Aus diesem Grund legen Sie die Synchronisationsskripten so fest, dass Zeilen anhand der Region gefiltert werden,

und Sie setzen die entfernte ID dieser Datenbank auf den Wert einer bestimmten Region-ID. Die Datenbank der betreffenden Region benötigt weniger Speicherplatz und weniger Zeit für die Synchronisation. Da die entfernte Datenbank auf eine minimale Größe begrenzt wird, können häufige Vorgänge, wie etwa die Eingabe eines neuen Kunden oder die Verarbeitung einer Änderung eines mobilen Geräts, schneller und effizienter ausgeführt werden.

Ergebnisse

Die Daten werden zwischen der entfernten und in der konsolidierten Datenbank synchronisiert.

Nächste Schritte

Gehen Sie weiter zu [„Aufräumen“ auf Seite 122](#).

Siehe auch

- [„UltraLite-Synchronisationsdienstprogramm \(ulsync\)“ \[UltraLite - Datenbankverwaltung\]](#)

Aufräumen

Entfernen Sie die Daten der praktischen Einführung von Ihrem Computer.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: MobiLink mit einer konsolidierten SQL Anywhere-Datenbank verwenden“](#).

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Planen von Schemata“ auf Seite 111](#).

Entfernen der Daten der praktischen Einführung von Ihrem Computer

1. Schließen Sie alle Instanzen der folgenden Anwendungen:
 - Sybase Central
 - Interactive SQL
2. Löschen Sie das Verzeichnis `C:\mssqla` mit Ihren konsolidierten und entfernten Datenbanken.
3. Führen Sie den folgenden Befehl aus, um die Beispieldatenbank zu löschen und eine neue Kopie der Beispieldatenbank mit ihren ursprünglichen Objekten und Daten zu erstellen:

```
newdemo "%SQLANYAMP16%\demo.db"
```

Wenn Sie dazu aufgefordert werden, bestätigen Sie das Löschen aller vorhandenen Dateien.

Ergebnisse

Die Daten der praktischen Einführung werden von Ihrem Computer entfernt.

Nächste Schritte

Keine.

Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden

Diese praktische Einführung veranschaulicht, wie Sie eine Oracle 10g-Datenbank mit MobiLink für die mobile Verwendung einrichten. Dabei wird die Synchronisation zwischen einer Oracle 10g-Datenbank und einer entfernten SQL Anywhere-Datenbank eingerichtet. Sie könnten auch eine entfernte UltraLite-Datenbank einrichten.

Der Zweck dieser praktischen Einführung ist, die Daten eines Verkaufsteams mobil verfügbar zu machen. In diesem Szenario ist jeder Vertriebsmitarbeiter ein entfernter Synchronisationsclient. Jeder Vertriebsmitarbeiter hat eine lokale SQL Anywhere-Datenbank, die mithilfe von MobiLink mit einer Oracle-Unternehmensdatenbank in der Zentrale synchronisiert wird. Jeder Vertriebsmitarbeiter kann auf die Unternehmensdaten über seinen Laptop oder sein mobiles Gerät zugreifen und Daten aus der entfernten Datenbank bearbeiten.

In dieser praktischen Einführung wird davon ausgegangen, dass Sie eine grundlegende Installation von Oracle Database 10g durchgeführt haben, bei der eine Starterdatenbank namens orcl erstellt wurde. Die Datenbank orcl verfügt über die Beispielschemata Order Entry (OE) und Human Relations (HR). Alternativ dazu können Sie eine neue Datenbank mithilfe des Oracle-Datenbank-Konfigurationsassistenten einrichten und die Beispielschemata erstellen oder die Beispielschemata manuell in eine leere Datenbank über SQL * Plus installieren. Die SQL-Dateien des Beispielschemas sind in separaten **Example** Download-Paketen direkt von Oracle verfügbar. Weitere Hinweise zum Installieren der beiden Beispielschemata finden Sie unter <http://www.oracle.com/technetwork/testcontent/connection11g-088156.html>.

In dieser praktischen Einführung wird davon ausgegangen, dass Sie als SYS-Benutzer eine Verbindung mit SYSDBA-Privilegien herstellen können. Dies ist erforderlich, wenn Sie die Berechtigung für die Oracle-Systemansicht GV_\$TRANSACTION erteilen. Das Kennwort für den SYS-Benutzer wird bei der Installation einer Oracle-Datenbank festgelegt.

Erforderliche Software

- SQL Anywhere 16
- Oracle Database 10g Release 2 oder später

Übersicht

In dieser praktischen Einführung wird Folgendes veranschaulicht:

- Berücksichtigen wichtiger Überlegungen beim Planen eines entfernten Schemas, wie etwa die Synchronisationsrichtungen für entfernte Tabellen.
- Hinzufügen eindeutiger Primärschlüssel zu konsolidierten und entfernten Datenbanken

- Erstellen einer ODBC-Datenquelle, die MobiLink mit einer Oracle 10g-Datenbank verbindet
- Einrichten der Synchronisation zwischen einer konsolidierten Datenbank und einer entfernten Datenbank mithilfe des **Assistenten zum Erstellen eines Synchronisationsmodells**
- Anpassen eines Synchronisationsmodells mit Sybase Central
- Deployment einer konsolidierten Datenbank und einer entfernten Datenbank mithilfe des **Assistenten für das Deployment eines Synchronisationsmodells**
- Synchronisieren des entfernten Clients mit der konsolidierten Datenbank

Berechtigungen und Privilegien

Sie müssen die folgenden Berechtigungen in der Oracle-Datenbank haben:

- SELECT für SYS.GV_\$TRANSACTION
- SELECT für SYS.GV_\$SESSION
- SELECT für SYS.V_\$SESSION
- SELECT für SYS.GV_\$LOCK
- EXECUTE für SYS.DBMS_UTILITY.
- SELECT für DBA_OBJECTS

Sie müssen die folgenden Rollen und Privilegien für die entfernte Datenbank haben:

- SYS_REPLICATION_ADMIN_ROLE-Systemrolle
- SYS_RUN_REPLICATION_ROLE-Systemrolle

Siehe auch

- [„MobiLink-Synchronisation“ auf Seite 1](#)
- [„MobiLink-Plug-In für Sybase Central“ auf Seite 22](#)

Lektion 1: Planen von Schemata

In dieser praktischen Einführung wird davon ausgegangen, dass die Beispielschemata Order Entry (OE) und Human Relations (HR) installiert sind. Das OE-Schema wird als konsolidierte Datenbank verwendet. Es beinhaltet Informationen über Angestellte, Bestellungen, Kunden und Produkte. In dieser praktischen Einführung befassen Sie sich in erster Linie mit dem OE-Schema.

Im Folgenden finden Sie eine Beschreibung der relevanten Tabellen im OE-Schema:

Tabelle	Beschreibung
CUSTOMERS	Kunden, deren Informationen gespeichert sind
INVENTORIES	Menge der in den einzelnen Lagerhäusern gespeicherten Produkte
ORDER_ITEMS	Liste der in den verschiedenen Bestellungen enthaltenen Produkte
ORDERS	Datensatz eines Verkaufs zwischen einem Vertriebsmitarbeiter und einem Kunden an einem bestimmten Datum
PRODUCT_DESCRIPTIONS	Beschreibung der einzelnen Produkte in verschiedenen Sprachen
PRODUCT_INFORMATION	Datensatz der einzelnen Produkte im System

Planung des entfernten Schemas

Es wäre nicht effizient, wenn jeder Vertriebsmitarbeiter eine Kopie der gesamten konsolidierten Datenbank hätte, und dies ist auch nicht erforderlich. Das entfernte Schema ist so angelegt, dass es nur Informationen über einen bestimmten Vertriebsmitarbeiter enthält. Hierzu wurde das entfernte Schema folgendermaßen eingerichtet:

Konsolidierte Tabelle	Entfernte Tabelle
CUSTOMERS	Enthält alle Zeilen
INVENTORIES	In entfernter Datenbank nicht enthalten
ORDER_ITEMS	Filtert nach sales_rep_id
ORDERS	Enthält alle Zeilen
PRODUCT_DESCRIPTIONS	In entfernter Datenbank nicht enthalten
PRODUCT_INFORMATION	Enthält alle Zeilen

Jeder Vertriebsmitarbeiter muss Datensätze aller Kunden und Produkte speichern, sodass jedes Produkt an jeden Kunden verkauft werden kann. In dieser praktischen Einführung wird davon ausgegangen, dass ein Vertriebsmitarbeiter immer dieselbe Sprache wie der Kunde spricht, sodass Sie die Tabelle PRODUCT_DESCRIPTIONS nicht benötigen. Jeder Vertriebsmitarbeiter benötigt Informationen über Bestellungen, aber nicht über Bestellungen anderer Vertriebsmitarbeiter. Hierzu werden die Zeilen basierend auf der Vertriebsmitarbeiter-ID gefiltert.

Im nächsten Schritt wird die Synchronisationsrichtung der einzelnen Tabellen ausgewählt. Sie sollten untersuchen, welche Informationen von einer entfernten Datenbank gelesen werden müssen und welche Informationen eine entfernte Datenbank erstellen, ändern oder entfernen muss. In diesem Beispiel benötigt ein bestimmter Vertriebsmitarbeiter eine Liste von Produkten und Kunden. Er gibt allerdings nie

ein neues Produkt in das System ein. Sie legen die Einschränkung fest, dass Produkte und Kunden immer über die konsolidierte Datenbank in der Zentrale in das System eingegeben werden. Ein Vertriebsmitarbeiter muss jedoch in der Lage sein, regelmäßig neue Bestellungen einzugeben. Diese Faktoren führen zu folgenden Entscheidungen über die Synchronisation in jeder Tabelle:

Tabelle	Synchronisation
CUSTOMERS	Nur auf entfernte Datenbank herunterladen.
ORDER_ITEMS	Download und Upload
ORDER	Download und Upload
PRODUCT_INFORMATION	Nur auf entfernte Datenbank herunterladen.

Gehen Sie weiter zu „[Lektion 2: Vorbereiten der konsolidierten Datenbank](#)“ auf Seite 126.

Lektion 2: Vorbereiten der konsolidierten Datenbank

Die OE-Datenbank muss für die Verwendung mit MobiLink geändert werden. Spalten werden gelöscht, da sie als benutzerdefinierte Typen erstellt wurden. Sie könnten diese benutzerdefinierten Typen in Typen konvertieren, die von SQL Anywhere erkannt werden, doch dies ist für diese praktische Einführung nicht relevant. Außerdem müssen Sie dem OE-Benutzer das Privileg erteilen, Trigger zu erstellen, weil MobiLink Trigger unter Verwendung der OE-Anmeldeinformationen erstellen muss.

Voraussetzungen

In dieser praktischen Einführung wird davon ausgegangen, dass Sie die Beispieldatenbank Order Entry (OE) installiert haben. Informationen über die Installation des Beispielschemas für Oracle 10g finden Sie in der Oracle-Dokumentation zu http://docs.oracle.com/cd/B14117_01/server.101/b10771.pdf.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 124.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden](#)“ auf Seite 123.

Kontext und Bemerkungen

Informationen über die Installation des Beispielschemas für Oracle 11g finden Sie in der Oracle-Dokumentation zu http://docs.oracle.com/cd/B28359_01/server.111/b28328.pdf.

Aufgabe

1. Stellen Sie eine Verbindung als SYS-Benutzer mit SYSDBA-Privilegien mithilfe der Oracle SQL Plus-Anwendung her. Führen Sie an der Eingabeaufforderung folgenden Befehl aus:

```
sqlplus SYS/your password for sys as SYSDBA
```

2. Um Spalten zu löschen, die als benutzerdefinierte Typen erstellt wurden, führen Sie folgende Anweisungen aus:

```
ALTER TABLE OE.CUSTOMERS DROP COLUMN CUST_ADDRESS;  
ALTER TABLE OE.CUSTOMERS DROP COLUMN PHONE_NUMBERS;  
ALTER TABLE OE.CUSTOMERS DROP COLUMN CUST_GEO_LOCATION;  
ALTER TABLE OE.PRODUCT_INFORMATION DROP COLUMN WARRANTY_PERIOD;
```

3. Um den OE-Benutzer freizugeben und das Kennwort auf sql einzustellen, führen Sie folgende Anweisung aus:

```
ALTER USER OE IDENTIFIED BY sql ACCOUNT UNLOCK;
```

4. Um dem OE-Benutzer das Erstellen von Triggern zu erlauben, führen Sie folgende Anweisung aus:

```
GRANT CREATE ANY TRIGGER TO OE;
```

5. Um den Fremdschlüssel orders_customer zu löschen und einen neuen Fremdschlüssel zu erstellen, der die Kunden-ID customer_id in der Tabelle customers referenziert, führen Sie folgende Befehle aus:

```
ALTER TABLE OE.ORDERS DROP CONSTRAINT ORDERS_CUSTOMER_ID_FK;  
ALTER TABLE OE.ORDERS ADD CONSTRAINT ORDERS_CUSTOMER_ID_FK  
FOREIGN KEY (CUSTOMER_ID) REFERENCES OE.CUSTOMERS (CUSTOMER_ID);
```

Ergebnisse

Spalten, die als benutzerdefinierte Typen erstellt wurden, werden gelöscht und der OE-Benutzer kann nun Trigger erstellen.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 3: Der konsolidierten Datenbank eindeutige Schlüssel hinzufügen](#)“ auf Seite 127.

Siehe auch

- „MobiLink - konsolidierte Datenbanken“ [*MobiLink - Serveradministration*]
- „Konsolidierte Oracle-Datenbank“ [*MobiLink - Serveradministration*]
- „Eindeutige Primärschlüssel“ [*MobiLink - Serveradministration*]

Lektion 3: Der konsolidierten Datenbank eindeutige Schlüssel hinzufügen

In einem Synchronisationssystem ist der Primärschlüssel einer Tabelle die einzige Möglichkeit, eine Zeile in einer anderen Datenbank eindeutig zu identifizieren, und die einzige Möglichkeit, Konflikte zu erkennen. Jede für die mobile Verwendung eingerichtete Tabelle muss einen Primärschlüssel haben. Der Primärschlüssel darf nie geändert werden. Sie müssen außerdem sicherstellen, dass ein in einer Datenbank eingefügter Primärschlüsselwert in keiner weiteren Datenbank eingefügt wird.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 124.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden](#)“ auf Seite 123.

Kontext und Bemerkungen

Es gibt mehrere Möglichkeiten, eindeutige Primärschlüssel zu generieren. Der Einfachheit halber wird in dieser praktischen Einführung die Methode der zusammengesetzten Primärschlüssel verwendet. Diese Methode erstellt Primärschlüssel mit mehreren Spalten, die in konsolidierten und entfernten Datenbanken eindeutig sind.

Aufgabe

1. Führen Sie an der Eingabeaufforderung folgenden Befehl aus:

```
sqlplus SYS/your password for sys as SYSDBA
```

2. Werte, die der SALES_REP_ID hinzugefügt werden, müssen in der Tabelle HR.EMPLOYEES vorhanden sein. Der Fremdschlüssel ORDERS_SALES_REP_FK erzwingt diese Regel. Führen Sie die folgende Anweisung zum Löschen der Fremdschlüssel aus:

```
ALTER TABLE OE.ORDERS  
DROP CONSTRAINT ORDERS_SALES_REP_FK;
```

3. Die Spalte SALES_REP_ID kann nicht als Primärschlüssel hinzugefügt werden, da sie Nullwerte enthält. Für diese praktische Einführung ersetzen Sie die Nullwerte durch den Wert 1. Führen Sie die folgende Anweisung aus:

```
UPDATE OE.ORDERS  
SET SALES_REP_ID = 1  
WHERE SALES_REP_ID IS NULL;
```

4. Die Spalte ORDER_ID ist der aktuelle Primärschlüssel der Tabelle ORDERS. Löschen Sie den aktuellen Primärschlüssel mit der folgenden Anweisung:

```
ALTER TABLE OE.ORDERS  
DROP PRIMARY KEY CASCADE;
```

5. Der zusammengesetzte Primärschlüssel umfasst die Spalten SALES_REP_ID und ORDER_ID. Fügen Sie den zusammengesetzten Primärschlüssel mit der folgenden Anweisung hinzu:

```
ALTER TABLE OE.ORDERS  
ADD CONSTRAINT salesrep_order_pk PRIMARY KEY (sales_rep_id, order_id);
```

Ergebnisse

Nachdem Sie diese Anweisungen ausgeführt haben, stellt der MobiLink-Server eine Verbindung mit der konsolidierten Datenbank her und richtet die Synchronisation für eine beliebige Anzahl entfernter Datenbanken ein.

In einer späteren Lektion wird das entfernte Schema aus dem konsolidierten Schema erstellt, sodass das entfernte Schema dieselben Primärschlüssel enthält wie das konsolidierte Schema.

Spalten wurden gezielt ausgewählt, um eindeutige Primärschlüssel für alle Datenbanken sicherzustellen. Für die Tabelle ORDERS besteht der Primärschlüssel aus den Spalten SALES_REP_ID und ORDER_ID. Jeder in die entfernte Vertriebstabelle eingegebene Wert muss eine eindeutige Bestellnummer haben (der Wert SALES_REP_ID ist immer gleich). Dadurch wird die Eindeutigkeit in jeder entfernten ORDERS-Tabelle sichergestellt. Der Primärschlüssel in der konsolidierten ORDERS-Tabelle verhindert Konflikte, wenn mehrere Vertriebsmitarbeiter gleichzeitig Daten hochladen. Jeder Upload eines Vertriebsmitarbeiters ist eindeutig und von den Uploads anderer Vertriebsmitarbeiter unterscheidbar, da ihre SALES_REP_ID-Werte unterschiedlich sind.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 4: Herstellen der Verbindung mit MobiLink“ auf Seite 129](#).

Lektion 4: Herstellen der Verbindung mit MobiLink

In dieser Lektion erstellen Sie eine ODBC-Datenquelle, die MobiLink mit der konsolidierten Datenbank verbindet.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Planen von Schemata“ auf Seite 124](#).

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: [„Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden“ auf Seite 123](#).

Aufgabe

1. Erstellen Sie eine ODBC-Datenquelle.

Verwenden Sie den SQL Anywhere 16 - Oracle-Oracle-Treiber, der mit SQL Anywhere 16 geliefert wird. Verwenden Sie die folgenden Konfigurationseinstellungen:

Felder der Registerkarte ODBC	Werte
Data Source Name (Datenquellename)	oracle_cons
Benutzer-ID	OE
Kennwort	sql
TNS-Dienstname	orcl
Die Prozedur gibt Ergebnisse zurück oder verwendet VARRAY-Parameter	ausgewählt

Felder der Registerkarte ODBC	Werte
Arraygröße	60000

In dieser praktischen Einführung wird davon ausgegangen, dass Sie eine grundlegende Installation von Oracle Database 10g durchgeführt haben, bei der eine Starterdatenbank namens orcl erstellt wurde. Das OE-Schema (Order Entry) wird von orcl automatisch installiert. Wenn Sie das OE-Schema auf einer anderen Datenbank installiert haben, verwenden Sie den Namen der Datenbank als TNS-Dienstnamen.

2. Klicken Sie auf **Verbindung testen**, um die ODBC-Verbindung zu testen.

Ergebnisse

Die ODBC-Datenquelle wird erstellt und getestet.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 5: Erstellen eines MobiLink-Projekts und eines Synchronisationsmodells](#)“ auf Seite 130.

Siehe auch

- <http://www.sybase.com/detail?id=1011880>
- „SQL Anywhere 16 - Oracle-ODBC-Treiber“ [[MobiLink - Serveradministration](#)]

Lektion 5: Erstellen eines MobiLink-Projekts und eines Synchronisationsmodells

In dieser Lektion verbinden Sie sich mit der konsolidierten Datenbank, indem Sie ein neues MobiLink-Projekt erstellen. Ein Synchronisationsmodell wird automatisch erstellt, wenn Sie ein neues Projekt erstellen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 124.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden](#)“ auf Seite 123.

Aufgabe

1. Klicken Sie auf **Start » Programme » SQL Anywhere 16 » Administrationstools » Sybase Central**.
2. Klicken Sie auf **Extras » MobiLink 16 » Neues Projekt**.

Der **Assistent zum Erstellen eines Projekts** wird angezeigt.

3. Im Feld **Wie lautet der Name des neuen Projekts?** geben Sie **oracle_project** ein.
4. Im Feld **Wo soll das neue Projekt gespeichert werden?** geben Sie **C:\mlora** ein und klicken auf **Weiter**.
5. Im Feld **Anzeigename der Datenbank** geben Sie **oracle_cons** ein.
6. Klicken Sie auf **Bearbeiten**.
7. Führen Sie die folgenden Aufgaben auf der Seite **Mit einer allgemeinen ODBC-Datenbank verbinden** durch:
 - a. Im Feld **Benutzer-ID** geben Sie **OE** ein.
 - b. Im Feld **Kennwort** geben Sie das Kennwort **sql** ein.
 - c. Klicken Sie im Feld **ODBC-Datenquellennamen** auf **Durchsuchen** und wählen Sie **oracle_cons**.
 - d. Klicken Sie auf **OK** und auf **Speichern**.
8. Wählen Sie die Option **Kennwort speichern** aus und klicken Sie auf **Weiter**.
9. Im Fenster **Eigentümer von konsolidierten Schemas** wählen Sie **Datenbankschema nur für ausgewählte Eigentümer laden** und aktivieren **OE-Benutzer**.
10. Wählen Sie auf der Seite **Neues entferntes Datenbankschema** in der Liste **Welche konsolidierten Datenbanktabellen und Spalten sollen in der entfernten Datenbank enthalten sein?** die folgenden Tabellen aus und klicken Sie dann auf **Weiter**:
 - CUSTOMERS
 - ORDERS
 - ORDER_ITEMS
 - PRODUCT_INFORMATION
11. Wählen Sie die Option **Dem Projekt einen Namen für ein entferntes Schema hinzufügen**.
12. Geben Sie **oracle_remote_schema** für den Namen des entfernten Schemas ein und klicken Sie dann auf **Weiter**.
13. Wählen Sie **SQL Anywhere** und klicken Sie dann auf **Fertig stellen**.

Wenn die konsolidierte Datenbank zum ersten Mal von MobiLink verwendet wird, werden Sie aufgefordert, die MobiLink-Systemeinrichtung zu installieren. Durch die Installation der MobiLink-Systemeinrichtung werden MobiLink-Systemtabellen und -Prozeduren hinzugefügt. Klicken Sie auf **Ja** und dann auf **OK**.

Ergebnisse

Ein MobiLink-Projekt- und -Synchronisationsmodell wird erstellt.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 6: Ein Synchronisationsmodell ändern“](#) auf Seite 132.

Lektion 6: Ein Synchronisationsmodell ändern

In dieser Lektion ändern Sie das Synchronisationsmodell für Ihre konsolidierte Datenbank, die Sie bei der Erstellung eines neuen MobiLink-Projekts erstellt haben.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Planen von Schemata“](#) auf Seite 124.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: [„Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden“](#) auf Seite 123.

Aufgabe

1. Klicken Sie mit der rechten Maustaste auf das Synchronisationsmodell **oracle_remote_schema** und wählen Sie dann **Eigenschaften**.
2. Geben Sie **sync_oracle** in das erste Feld ein.
3. Führen Sie die folgenden Aufgaben durch:
 - a. Geben Sie im Feld **Publikationsname** **sync_oracle_publication** ein.
 - b. Geben Sie im Feld **Skriptversion** **sync_oracle_scriptversion** ein.

Die Publikation ist das Objekt in der entfernten Datenbank, das festlegt, welche Daten synchronisiert werden. MobiLink-Serverskripten legen fest, wie hochgeladene Daten von entfernten Datenbanken in der konsolidierten Datenbank verwendet werden und Skriptversionen Skripten gruppieren. Sie können unterschiedliche Skriptversionen für verschiedene Anwendungen verwenden. Auf diese Weise können Sie einen einzelnen MobiLink-Server verwalten und gleichzeitig mehrere Anwendungen synchronisieren.
 - c. Klicken Sie auf **Übernehmen** und dann auf **OK**.
4. Legen Sie für jede Tabelle im Synchronisationsmodell die Richtung fest, in der die Daten synchronisiert werden.

Klicken Sie auf die Registerkarte **Zuordnungen** im rechten Fensterausschnitt und legen Sie die Zeilen in der Spalte **Zuordnungsrichtung** wie folgt fest:

- Die Tabellen **ORDERS** und **ORDER_ITEMS** müssen mit **Bidirektional** (Upload und Download) festgelegt werden.
- Die restlichen Tabellen müssen auf **Nur auf entfernte Datenbank heruntergeladen** festgelegt werden.

5. Wenn ein Fenster angezeigt wird, das angibt, dass das Laden des konsolidierten Schemas für alle Eigentümer lange Zeit in Anspruch nehmen kann, laden Sie das Datenbankschema für die Benutzer **HR** und **OE**.
6. Filtern Sie die Zeilen, die in die entfernte Datenbank heruntergeladen wurden, nach der entfernten ID.
 - a. Wählen Sie die Zeile mit der **ORDERS**-Tabelle und klicken Sie auf die Registerkarte **Download-Teilmenge** unten im rechten Fensterausschnitt.
 - b. Ändern Sie die Spalte **Download-Teilmenge** auf **Benutzerdefiniert**.
 - c. Filtern Sie die Zeilen nach entfernten IDs, die die entfernte Datenbank eindeutig identifizieren, indem Sie eine Einschränkung der WHERE-Klausel des download_cursor-Skripts hinzufügen.
Geben Sie im Feld **In der WHERE-Klausel des Download-Cursors zu verwendender SQL-Ausdruck** eine Suchbedingung ein. Das folgende SQL-Skript kann beispielsweise für die **ORDERS**-Tabelle verwendet werden:

```
OE.ORDERS.SALES_REP_ID = {ml s.remote_id}
```

Das Download-Cursorskript legt fest, welche Spalten und Zeilen von den einzelnen Tabellen in die entfernte Datenbank heruntergeladen werden. Die Suchbedingung stellt sicher, dass Sie nur Informationen über einen einzigen Vertriebsmitarbeiter herunterladen, nämlich den Mitarbeiter, dessen ID mit der entfernten Datenbank-ID übereinstimmt.

- d. Klicken Sie auf die Registerkarte **Download-Löschen-Teilmenge** und ändern Sie die **Download-Löschen-Teilmenge** von **Identisch** auf **Keine**.
7. Speichern Sie das Synchronisationsmodell.

Klicken Sie auf **Datei » Speichern**.

Ergebnisse

Das Synchronisationsmodell ist vollständig und kann nun per Deployment bereitgestellt werden.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 7: Deployment des Synchronisationsmodells](#)“ auf Seite 134.

Siehe auch

- „Einrichten einer konsolidierten Datenbank“ [[MobiLink - Serveradministration](#)]
- „Systemtabellen des MobiLink-Servers“ [[MobiLink - Serveradministration](#)]
- „Systemprozeduren des MobiLink-Servers“ [[MobiLink - Serveradministration](#)]
- „Schreiben von Synchronisationsskripten“ [[MobiLink - Serveradministration](#)]
- „download_delete_cursor-Skripten“ [[MobiLink - Serveradministration](#)]
- „Konfliktverarbeitung“ [[MobiLink - Serveradministration](#)]
- „Konfliktlösung“ [[MobiLink - Serveradministration](#)]
- „Publikationen“ [[MobiLink - Clientadministration](#)]
- „Synchronisationsereignisse“ [[MobiLink - Serveradministration](#)]
- „Synchronisationsmodellaufgaben“ auf Seite 33
- „Synchronisationsmethoden“ [[MobiLink - Serveradministration](#)]
- „Downloadtyp ändern“ auf Seite 38
- „Konflikterkennung und -lösung ändern“ auf Seite 46
- „Tabellen- und Spaltenzuordnungen“ auf Seite 34

Lektion 7: Deployment des Synchronisationsmodells

Sie können mit dem **Assistenten für das Deployment eines Synchronisationsmodells** das Deployment der konsolidierten Datenbank und der entfernten Datenbank vornehmen. Sie können die beiden Datenbanken entweder einzeln oder zusammen bereitstellen. Der **Assistent für das Deployment eines Synchronisationsmodells** führt Sie durch die Schritte für die Konfiguration der Optionen für das Deployment.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 124.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden](#)“ auf Seite 123.

Aufgabe

1. Erweitern Sie im linken Fensterausschnitt von Sybase Central unter **MobiLink 16** die Elemente **oracle_project**, **Synchronisationsmodelle** und dann **sync_oracle**.
2. Klicken Sie auf **Datei » Deployment vornehmen**.

Der **Assistent zum Deployment eines Synchronisationsmodells** wird geöffnet.

3. Akzeptieren Sie die Standardeinstellung für das Feld **Wählen Sie den Ordner für vom Assistenten generierte Dateien** und klicken Sie auf **Weiter**.
4. Akzeptieren Sie die Standardeinstellungen für die Seite **Client-Netzwerkoptionen** und klicken Sie auf **Weiter**.

5. Auf der Seite **MobiLink-Benutzer und -Kennwort** unter **Welchen Benutzer und welches Kennwort für MobiLink möchten Sie verwenden?** wählen Sie **Diese verwenden** und führen die folgenden Aufgaben aus:
 - a. Im Feld **MobiLink-Benutzer** geben Sie **oracle_remote** ein.
 - b. Im Feld **MobiLink-Kennwort** geben Sie **oracle_pass** ein.
 - c. Wählen Sie **Registrieren Sie diesen Benutzer in der konsolidierten Datenbank. Registrierte Benutzer sind berechtigt, die Synchronisation auszuführen.** und klicken Sie auf **Weiter** und dann nochmals auf **Weiter**.
6. Auf der Seite **Synchronisationsprofil** lassen Sie den Standardwert für den Synchronisationsprofilnamen als **sync_oracle_publication_oracle_remote**. Klicken Sie auf **Weiter**.
7. Auf der Seite **Auswählen, wie Datenbanken für die Synchronisation vorbereitet werden** wählen Sie die folgenden Optionen:
 - a. xBei **Was möchten Sie mit dem SQL-Skript tun, das erstellt wird, um die konsolidierte Datenbank für die Synchronisation vorzubereiten?** wählen Sie **Mit der konsolidierten Datenbank ausführen**.
 - b. Bei **Was möchten Sie mit dem SQL-Skript tun, das erstellt wird, um die konsolidierte Datenbank für die Synchronisation vorzubereiten?** wählen Sie **Mit einer neuen entfernten Datenbank ausführen**.
 - c. Klicken Sie auf **Weiter**.
8. Klicken Sie auf **Fertig stellen**.
9. Klicken Sie auf **Schließen**.

Ergebnisse

Ihre konsolidierte Datenbank ist für die Synchronisation mit mehreren entfernten Clients konfiguriert und Sie haben erfolgreich ein Deployment auf einen entfernten Client vorgenommen. Um Deployments auf andere entfernte Clients vorzunehmen, können Sie diesen Assistenten erneut ausführen. Hierzu müssen Sie einen neuen MobiLink-Benutzer erstellen und die Auswahl des Deployments der konsolidierten Datenbank und des MobiLink-Servers aufheben. Da die konsolidierte Datenbank und der MobiLink-Server bereits bereitgestellt wurden, müssen Sie nur noch das Deployment anderer entfernter Synchronisationsclients vornehmen.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 8: Starten von Server und Client](#)“ auf Seite 136.

Siehe auch

- „[Deployment des Synchronisationsmodells](#)“ auf Seite 50
- „[SQL Anywhere-Datenbanken als entfernte Datenbanken verwenden](#)“ [*MobiLink - Clientadministration*]
- „[MobiLink-Benutzer](#)“ [*MobiLink - Clientadministration*]

Lektion 8: Starten von Server und Client

In einer früheren Lektion haben Sie das Download-Cursor-Skript geändert, um Informationen zu einem Vertriebsmitarbeiter herunterzuladen. In dieser Lektion geben Sie den Vertriebsmitarbeiter an, indem Sie die entfernte ID auf die ID des Vertriebsmitarbeiters festlegen und die konsolidierte und entfernte Datenbank von MobiLink starten.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 124.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden](#)“ auf Seite 123.

Kontext und Bemerkungen

Standardmäßig verwendet MobiLink die Snapshot/READ COMMITTED-Isolationsstufe für Uploads und Downloads. Damit der MobiLink-Server die Snapshot-Isolation optimal nutzen kann, muss das vom MobiLink-Server verwendete Oracle-Konto Zugriff auf die Oracle-Systemansicht GV_\$TRANSACTION haben. Ist dies nicht der Fall, wird eine Warnung ausgegeben und die Zeilen werden unter Umständen beim Download ausgelassen.

Aufgabe

1. Stellen Sie eine Verbindung als SYS-Benutzer mit SYSDBA-Privilegien mithilfe der Oracle SQL Plus-Anwendung her. Führen Sie an der Eingabeaufforderung folgenden Befehl aus:

```
sqlplus SYS/your password for sys as SYSDBA
```

2. Um die Zugriffsberechtigung auf die Oracle-Systemansicht GV_\$TRANSACTION zu erteilen, führen Sie folgende Anweisung aus:

```
GRANT SELECT ON SYS.GV_$TRANSACTION TO OE;
```

3. Um die Zugriffsberechtigung auf die Oracle-Systemansichten V\$SESSION und GV_\$SESSION zu erteilen, führen Sie die folgende Anweisung aus:

```
GRANT SELECT ON SYS.V_$SESSION TO OE;  
GRANT SELECT ON SYS.GV_$SESSION TO OE;
```

4. Um die Zugriffsberechtigung auf andere Systemobjekte zu erteilen, führen Sie die folgende Anweisung aus:

```
GRANT SELECT ON SYS.GV_$LOCK TO OE;  
GRANT EXECUTE ON SYS.DBMS_UTILITY TO OE;  
GRANT SELECT ON DBA_OBJECTS TO OE;
```

5. Navigieren Sie an einer Eingabeaufforderung zu dem Verzeichnis, in dem Sie das Synchronisationsmodell erstellt haben. (Dies ist das Stammverzeichnis, das Sie im ersten Schritt des **Assistenten zum Erstellen eines Synchronisationsmodells** gewählt haben.)

Wenn Sie die vorgeschlagenen Verzeichnisnamen verwendet haben, navigieren Sie zum folgenden Verzeichnis: `mlorc\oracle_project\sync_oracle_deploy\`.

6. Führen Sie zum Start des MobiLink-Servers folgenden Befehl aus:

```
mlsrv.bat "DSN=oracle_cons;UID=OE;PWD=sql"
```

- **mlsrv.bat** Die Befehlsdatei, die zum Starten des MobiLink-Servers erstellt wurde.
- **DSN** Der ODBC-Datenquellenname.
- **UID** Der Benutzername, mit dem Sie eine Verbindung mit der konsolidierten Datenbank herstellen.
- **PWD** Das Kennwort, mit dem Sie eine Verbindung mit der konsolidierten Datenbank herstellen.

Wenn dieser Befehl erfolgreich ausgeführt wurde, erscheint im Meldungsfenster des MobiLink-Servers die Meldung `MobiLink-Server gestartet`.

Wenn der MobiLink-Server nicht gestartet werden kann, überprüfen Sie die Verbindungsinformationen für die konsolidierte Datenbank.

7. Navigieren Sie an der Eingabeaufforderung zu dem Verzeichnis, in dem der **Assistent für das Deployment eines Synchronisationsmodells** die entfernte Datenbank erstellt hat.

Wenn Sie die vorgeschlagenen Verzeichnisnamen verwendet haben, navigieren Sie zum folgenden Verzeichnis: `mlorc\oracle_project\sync_oracle_deploy\`.

8. Starten Sie die entfernte SQL Anywhere-Datenbank mit folgendem Befehl:

```
dbsrv16 -n remote_eng sync_oracle_remote.db -n remote_db
```

- **dbsrv16** Der Datenbankserver, der zum Start der SQL Anywhere-Datenbank verwendet wird.
- **remote_eng** Der Name des Datenbankservers.
- **sync_oracle_remote.db** Die Datenbankdatei, die auf `remote_eng` gestartet wurde.
- **remote_db** Der Name der Datenbank auf `remote_eng`.

Ergebnisse

Wenn dieser Befehl erfolgreich ausgeführt wurde, wird ein SQL Anywhere-Datenbankserver namens `remote_eng` gestartet und die Datenbank `remote_db` wird geladen.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 9: Einstellen der entfernten ID](#)“ auf Seite 138.

Siehe auch

- „Syntax des SQL Anywhere-Datenbankservers“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Per Deployment bereitgestellte Synchronisationsmodelle“ auf Seite 54
- „MobiLink-Server“ [[MobiLink - Serveradministration](#)]
- „Entfernte IDs“ [[MobiLink - Clientadministration](#)]

Lektion 9: Einstellen der entfernten ID

Im entfernten Schema repräsentiert jede entfernte Datenbank einen Vertriebsmitarbeiter. Die von Ihnen erstellten Synchronisationsskripten enthielten Logik, die den MobiLink-Server angewiesen haben, eine Teilmenge der Daten basierend auf der entfernten ID der entfernten Datenbank herunterzuladen. Sie müssen die entfernte ID der Datenbank vor der ersten Synchronisation auf den Wert einer gültigen Vertriebsmitarbeiter-ID setzen, da das entfernte Gerät bei der ersten Synchronisation alle Informationen zu dem gewählten Vertriebsmitarbeiter herunterlädt.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 124.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden](#)“ auf Seite 123.

Aufgabe

1. Wählen Sie eine gültige Vertriebsmitarbeiter-ID aus:
 - a. Stellen Sie eine Verbindung als SYS-Benutzer mit SYSDBA-Privilegien mithilfe der Oracle SQL Plus-Anwendung her. Führen Sie an der Eingabeaufforderung folgenden Befehl aus:

```
sqlplus SYS/your-password-for-sys as SYSDBA
```

- b. Um eine Liste gültiger Vertriebsmitarbeiter-IDs in der Tabelle ORDERS anzuzeigen, führen Sie folgende Anweisung aus:

```
SELECT COUNT( SALES_REP_ID ), SALES_REP_ID  
FROM OE.ORDERS GROUP BY SALES_REP_ID;
```

In diesem Beispiel repräsentiert die entfernte Datenbank einen Vertriebsmitarbeiter mit der SALES_REP_ID 154.

- c. Um Oracle SQL Plus zu beenden, führen Sie folgenden Befehl aus:

```
exit
```

2. Um die entfernte ID der Datenbank auf den Wert 154 festzulegen, führen Sie folgenden Befehl aus:

```
dbisql  
-c "SERVER=remote_eng;DBN=remote_db;UID=DBA;PWD=sql"  
"SET OPTION PUBLIC.ml_remote_id='154';"
```

- **dbisql** Die Anwendung, die zur Ausführung von SQL-Befehlen in einer SQL Anywhere-Datenbank verwendet wurde.
- **ENG** Legt den Datenbankservernamen auf remote_eng fest.
- **DBN** Legt den Datenbanknamen auf remote_db fest.
- **UID** Der Benutzername, mit dem eine Verbindung mit der entfernten Datenbank hergestellt wird.

- **PWD** Das Kennwort, mit dem eine Verbindung mit der entfernten Datenbank hergestellt wird.
- **SET OPTION PUBLIC.ml_remote_id='154'** Die SQL-Anweisung, mit der die entfernte ID auf 154 festgelegt wird.

Ergebnisse

Die entfernte ID der Datenbank wird auf den Wert einer gültigen Vertriebsmitarbeiter-ID gesetzt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 10: Den entfernten Client synchronisieren](#)“ auf Seite 139.

Lektion 10: Den entfernten Client synchronisieren

In dieser Lektion synchronisieren Sie den entfernten Client erstmals. Dies erfolgt mit dem Dienstprogramm dbmlsync. Dbmlsync stellt eine Verbindung mit der entfernten Datenbank her, authentifiziert sich selbst beim MobiLink-Server und führt alle Uploads und Downloads aus, die zur Synchronisation der entfernten und konsolidierten Datenbanken, die auf einer Publikation in der entfernten Datenbank basieren, erforderlich sind.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 124.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden](#)“ auf Seite 123.

Aufgabe

- Führen Sie an der Eingabeaufforderung folgenden Befehl aus:

```
dbmlsync
-c "SERVER=remote_eng;DBN=remote_db;UID=DBA;PWD=sql"
-n sync_oracle_publication
-u oracle_remote -mp oracle_pass
```

- **dbmlsync** Die Synchronisationsanwendung.
- **SERVER** Gibt den Namen des Fremdservers an.
- **DBN** Der Name der entfernten Datenbank.
- **UID** Gibt den Benutzernamen an, mit dem eine Verbindung mit der entfernten Datenbank hergestellt wird.
- **PWD** Gibt das Kennwort an, mit dem eine Verbindung mit der entfernten Datenbank hergestellt wird.
- **sync_oracle_publication** Die Publikation auf dem entfernten Gerät, das zur Ausführung der Synchronisation verwendet wird. (Diese Publikation wurde vom **Assistenten zum Erstellen eines Synchronisationsmodells** erstellt.)

- **oracle_remote** Der Benutzername, der zur Authentifizierung mit dem MobiLink-Server verwendet wird.
- **oracle_pass** Das Kennwort, das zur Authentifizierung mit dem MobiLink-Server verwendet wird.

Hinweis

Wenn Sie die Anwendung dbmlsync auf einem anderen Computer als dem MobiLink-Server ausführen, müssen Sie Argumente mit der Position des MobiLink-Servers übergeben.

Ergebnisse

Der Verarbeitungsfortschritt der Synchronisation wird im **Meldungsfenster des SQL Anywhere MobiLink-Clients** angezeigt. Wenn dieser Befehl erfolgreich ausgeführt wird, füllt die Anwendung dbmlsync die entfernte Datenbank mit einer Teilmenge der Informationen aus der konsolidierten Datenbank.

Wenn die Synchronisation fehlschlägt, überprüfen Sie die Verbindungsinformationen, die Sie an die dbmlsync-Anwendung übermitteln, sowie den MobiLink-Benutzernamen und das Kennwort. Wenn diese Informationen richtig sind, überprüfen Sie den verwendeten Publikationsnamen und stellen Sie sicher, dass die konsolidierte Datenbank und der MobiLink-Server ausgeführt werden. Sie können auch den Inhalt des Synchronisationslogs (Server und Client) überprüfen.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 11: Einfügen von Daten in die entfernte Datenbank](#)“ auf Seite 140.

Siehe auch

- „Der Synchronisationsprozess“ auf Seite 14
- „dbmlsync-Syntax“ [*MobiLink - Clientadministration*]

Lektion 11: Einfügen von Daten in die entfernte Datenbank

Nach der erfolgreichen Synchronisation des entfernten Clients mit der konsolidierten Datenbank durch den MobiLink-Server sollte die entfernte Datenbank mit den Informationen eines Vertriebsmitarbeiters gefüllt sein. Sie können überprüfen, ob die Datenbank korrekt mit Daten gefüllt wurde, indem Sie in Sybase Central das SQL Anywhere 16-Plug-In verwenden.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 124.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden](#)“ auf Seite 123.

Aufgabe

1. Starten Sie Sybase Central.
2. Stellen Sie eine Verbindung zur entfernten Datenbank her:
 - a. Klicken Sie im linken Fensterausschnitt mit der rechten Maustaste auf **SQL Anywhere 16** und klicken Sie auf **Verbinden**.
 - b. Klicken Sie in der Dropdown-Liste **Authentifizierung** auf die Option **Datenbank** und geben Sie **DBA** als **Benutzer-ID** sowie **sql** als **Kennwort** ein.
 - c. Klicken Sie in der Dropdown-Liste **Aktion** auf die Option **Mit einer laufenden Datenbank auf diesem Computer verbinden** und geben Sie dann **remote_eng** als **Servernamen** und **remote_db** als **Datenbanknamen** ein.
 - d. Klicken Sie auf **Verbinden**.
3. Klicken Sie im linken Fensterausschnitt von Sybase Central unter **remote_db - DBA** auf das Element **Tabellen** und klicken Sie auf die Tabelle **ORDERS** und dann auf der Registerkarte **Daten** in den rechten Fensterausschnitt.

In der Tabelle **ORDERS** werden alle Datensätze für den Vertriebsmitarbeiter mit der ID 154 angezeigt. Dieser Vertriebsmitarbeiter hat nichts mit den Verkaufsinformationen anderer Vertriebsmitarbeiter zu tun. Aus diesem Grund legen Sie die Synchronisationsskripten so fest, dass Zeilen anhand der entfernten ID gefiltert werden, und legen die entfernte ID dieser Datenbank auf den Wert einer bestimmten Vertriebsmitarbeiter-ID fest. Nun benötigt die Datenbank des betreffenden Vertriebsmitarbeiters weniger Speicherplatz und weniger Zeit für die Synchronisation. Da die Größe der entfernten Datenbank auf ein Minimum beschränkt wird, können häufige Vorgänge, z.B. die Eingabe eines neuen Verkaufs oder die Verarbeitung einer Rückvergütung für einen vorherigen Verkauf, schneller und effizienter ausgeführt werden.

Ergebnisse

Die entfernte Datenbank wird nur mit Daten gefüllt, die sich auf den Vertriebsmitarbeiter mit der ID 154 beziehen.

Nächste Schritte

Gehen Sie weiter zu [„Aufräumen“ auf Seite 141](#).

Aufräumen

Erstellen Sie die Datenbank Order Entry neu, und entfernen Sie das gesamte Material der praktischen Einführung von Ihrem Computer.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Planen von Schemata“ auf Seite 124](#).

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden“ auf Seite 123.

Aufgabe

1. Befolgen Sie die Schritte in <http://www.oracle.com/technetwork/testcontent/obeconnection-099425.html> zur Neuerstellung der Oracle-Beispieldatenbank.
2. MobiLink-Projekt löschen
 - a. Starten Sie Sybase Central.
 - b. Doppelklicken Sie im rechten Fensterausschnitt auf **MobiLink 16**.
 - c. **oracle_project** wird im rechten Fensterausschnitt angezeigt.
 - d. Klicken Sie mit der rechten Maustaste auf **oracle_project** und klicken Sie auf **Löschen**.
 - e. Im Fenster **Projekt löschen** wählen Sie **Aus der Liste und vom Computer entfernen** und klicken auf **Ja**.

Ergebnisse

Die Order Entry-Datenbank wird generiert und das gesamte Material der praktischen Einführung wird von Ihrem Computer entfernt.

Nächste Schritte

Keine.

Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden

Diese praktische Einführung zeigt, wie Sie eine Adaptive Server Enterprise-Datenbank mit MobiLink für die mobile Verwendung einrichten. Die Synchronisation zwischen einer konsolidierten Adaptive Server Enterprise-Datenbank und einer entfernten SQL Anywhere-Datenbank wird eingerichtet. Sie könnten auch UltraLite-Clients verwenden.

Zweck dieser praktischen Einführung ist, Daten für eine Buchhandlungskette für die mobile Verwendung bereitzustellen. Jede Buchhandlung in diesem Szenario ist eine entfernte Synchronisationsumgebung. Jede Buchhandlung hat eine lokale SQL Anywhere-Datenbank, die mit der Adaptive Server Enterprise-Datenbank in der Zentrale synchronisiert wird. Jede Buchhandlung kann mehrere Computer haben, die auf die Daten der entfernten Datenbank zugreifen und sie ändern können.

In dieser praktischen Einführung wird davon ausgegangen, dass das Beispielschema pubs2 auf einem Adaptive Server Enterprise 15.7-Server installiert ist. Das Beispielschema pubs2 wird mit Adaptive Server Enterprise 15.0 bereitgestellt. Es ist ein optionaler Bestandteil der Installation. In dieser

praktischen Einführung wird es als konsolidierte Datenbank verwendet. Informationen über dieses Beispiel finden Sie in der Adaptive Server Enterprise-Dokumentation oder können online unter <http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc31654.1570/html/sag1/X21150.htm> angezeigt werden.

Diese praktische Einführung verwendet das **sa**-Standardkonto. Wenn Adaptive Server Enterprise installiert ist, hat das **sa**-Konto ein Nullkennwort. In dieser praktischen Einführung wird davon ausgegangen, dass Sie das Nullkennwort durch ein gültiges Kennwort ersetzt haben. Weitere Hinweise zur Änderung des Nullkennworts in Adaptive Server Enterprise finden Sie unter <http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc36272.1570/html/commands/CHDGIFDG.htm>.

Erforderliche Software

- SQL Anywhere 16
- Adaptive Server Enterprise 15.7

Übersicht

In dieser praktischen Einführung wird Folgendes veranschaulicht:

- Berücksichtigen wichtiger Überlegungen beim Planen eines entfernten Schemas, wie etwa die Synchronisationsrichtungen für entfernte Tabellen.
- Hinzufügen eindeutiger Primärschlüssel zu konsolidierten und entfernten Datenbanken
- Erstellen einer ODBC-Datenquelle, die MobiLink mit einer Adaptive Server Enterprise-Datenbank verbindet.
- Einrichten der Synchronisation zwischen einer konsolidierten Datenbank und einer entfernten Datenbank mithilfe des **Assistenten zum Erstellen eines Synchronisationsmodells**
- Passen Sie die Synchronisationseinstellungen mit Sybase Central an.
- Deployment einer konsolidierten Datenbank und einer entfernten Datenbank mithilfe des **Assistenten für das Deployment eines Synchronisationsmodells**
- Synchronisieren des entfernten Clients mit der konsolidierten Datenbank

Berechtigungen und Privilegien

Sie müssen die folgenden Rollen und Privilegien für die konsolidierte Datenbank haben:

- SELECT-Berechtigung für MASTERSYSTRANSACTIONS und MASTERSYSPROCESSES

Sie müssen die folgenden Rollen und Privilegien für die entfernte SQL Anywhere-Datenbank haben:

- SYS_REPLICATION_ADMIN_ROLE-Systemrolle
- SYS_RUN_REPLICATION_ROLE-Systemrolle

Siehe auch

- „MobiLink-Synchronisation“ auf Seite 1

Lektion 1: Planen von Schemata

Das Beispielschema pubs2 wird als konsolidiertes Datenbankschema verwendet. Es enthält Informationen über Buchhandlungen, Titel, Autoren, Verlage und den Vertrieb. Die folgende Tabelle enthält eine Beschreibung der einzelnen Tabellen in der Adaptive Server Enterprise-Datenbank:

Tabelle	Beschreibung
au_pix	Bilder der Autoren
authors	Die Autoren der verschiedenen <i>Titel</i> (titles) im System.
discounts	Informationen über verschiedene Rabatte in verschiedenen <i>Buchhandlungen</i> (stores).
sales	Jeder Verkaufsdatensatz bezieht sich auf einen Verkauf durch eine bestimmte Buchhandlung.
salesdetail	Informationen über die verschiedenen <i>Titel</i> (titles), die in einem bestimmten Verkauf enthalten waren.
stores	Jeder Buchhandlungs-Datensatz bezieht sich auf eine Buchhandlung oder eine Zweigstelle im System.
titleauthor	Informationen darüber, welche <i>Titel</i> (titles) von welchem <i>Autor</i> (authors) geschrieben wurden.
titles	Datensätze aller Bücher im System.
blurbs, publishers und roysched	Informationen, die in dieser praktischen Einführung nicht benötigt werden.

Planung des entfernten Schemas

Es wäre nicht effizient, wenn jede Buchhandlung eine Kopie der gesamten konsolidierten Datenbank hätte, und dies ist auch nicht erforderlich. Das entfernte Schema verwendet dieselben Tabellennamen, doch es enthält nur die für eine bestimmte Buchhandlung relevanten Informationen. Um diese Konfiguration zu erreichen, ist das entfernte Schema als Teilmenge der konsolidierten Datenbank wie folgt eingerichtet:

Konsolidierte Tabelle	Entfernte Tabelle
au_pix	Enthält alle Zeilen
authors	Enthält alle Zeilen

Konsolidierte Tabelle	Entfernte Tabelle
discounts	Filterung nach stor_id
sales	Filterung nach stor_id
salesdetail	Filterung nach stor_id
stores	Filterung nach stor_id
titleauthor	Enthält alle Zeilen
titles	Enthält alle Zeilen
blurbs	In entfernter Datenbank nicht enthalten
publishers	In entfernter Datenbank nicht enthalten
roysched	In entfernter Datenbank nicht enthalten

Jede Buchhandlung muss Datensätze aller Titel und Autoren besitzen, sodass Kunden das Lager der Buchhandlung durchsuchen können. Eine Buchhandlung braucht jedoch keine Informationen über Verlage oder Tantiemen, sodass diese Informationen nicht für jede Buchhandlung synchronisiert werden. Jede Buchhandlung benötigt Informationen über Verkaufszahlen und Rabatte, jedoch nicht über Verkaufszahlen und Rabatte anderer Buchhandlungen. Dieses Verhalten wird erreicht, indem die Zeilen basierend auf der Buchhandlungs-ID gefiltert werden.

Hinweis

Sie können auch eine Teilmenge von Spalten aus einer Tabelle verwenden, wenn bestimmte Spalten in den entfernten Datenbanken nicht benötigt werden.

Im nächsten Schritt wird die Synchronisationsrichtung der einzelnen Tabellen ausgewählt. Sie sollten untersuchen, welche Informationen von einer entfernten Datenbank gelesen werden müssen und welche Informationen eine entfernte Datenbank erstellen, ändern oder entfernen muss. In diesem Beispiel muss eine Buchhandlung Zugriff auf die Liste der Autoren und Titel haben, doch sie gibt nie einen neuen Autor in das System ein. Daher gibt es die Einschränkung, dass Autoren und Titel immer über die konsolidierte Datenbank in der Zentrale eingegeben werden müssen. Eine Buchhandlung muss jedoch in der Lage sein, regelmäßig neue Verkaufszahlen einzugeben. Diese Faktoren führen zu folgenden Synchronisationsrichtungen für die Tabellen:

Tabelle	Synchronisation
titleauthor	Nur auf entfernte Datenbank herunterladen.
authors	Nur auf entfernte Datenbank herunterladen.
au_pix	Nur auf entfernte Datenbank herunterladen.

Tabelle	Synchronisation
titles	Nur auf entfernte Datenbank herunterladen.
stores	Nur auf entfernte Datenbank herunterladen.
discounts	Nur auf entfernte Datenbank herunterladen.
sales	Download und Upload
salesdetail	Download und Upload

Gehen Sie weiter zu „[Lektion 2: Die konsolidierte Datenbank vorbereiten](#)“ auf Seite 146.

Lektion 2: Die konsolidierte Datenbank vorbereiten

In dieser Lektion erhöhen Sie die Größe der konsolidierten Datenbank für die MobiLink-Synchronisation und erstellen eindeutige Primärschlüssel.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 144.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden](#)“ auf Seite 142.

Kontext und Bemerkungen

MobiLink muss der Datenbank pubs2 Systemtabellen und andere Objekte für die Synchronisation hinzufügen. Wenn Sie diese Objekte hinzufügen, muss die Datenbank pubs2 vergrößert werden.

Aufgabe

1. Stellen Sie mithilfe des Dienstprogramms isql in Adaptive Server Enterprise als **sa** eine Verbindung mit der Datenbank pubs2 her. Führen Sie an einer Eingabeaufforderung folgenden Befehl in einer einzigen Zeile aus:

```
isql  
-U sa  
-P your-password-for-sa-account  
-D pubs2
```

Wenn Sie entfernt auf Adaptive Server Enterprise zugreifen, geben Sie den Servernamen mit der Option -S an.

2. Um die erforderliche Berechtigung für die Erhöhung der Größe einer Datenbank zu haben, müssen Sie auf die Master-Datenbank zugreifen. Führen Sie folgenden Befehl in isql aus:


```
use master
go
sp_dboption pubs2, "select into/bulkcopy/pllsort", true
go
```

3. In Adaptive Server Enterprise wird eine Datenbank auf einer Festplatte oder einem Teil einer Festplatte gespeichert. Führen Sie zur Erweiterung der Datenbank pubs2 die folgende Anweisung aus (Sie müssen die Festplatte angeben, auf der pubs2 gespeichert ist):

```
ALTER DATABASE pubs2 ON disk-name = 33
```

Ergebnisse

Ergebnis

Nächste Schritte

Gehen Sie weiter zu „[Lektion 3: Der konsolidierten Datenbank eindeutige Schlüssel hinzufügen](#)“ auf Seite 147.

Siehe auch

- „[Konsolidierte Adaptive Server Enterprise-Datenbank](#)“ [[MobiLink - Serveradministration](#)]

Lektion 3: Der konsolidierten Datenbank eindeutige Schlüssel hinzufügen

In dieser Lektion werden eindeutige Primärschlüssel zur konsolidierten Datenbank hinzugefügt.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 144.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden](#)“ auf Seite 142.

Kontext und Bemerkungen

In einem Synchronisationssystem ist der Primärschlüssel einer Tabelle die einzige Möglichkeit, eine Zeile in einer anderen Datenbank eindeutig zu identifizieren, und die einzige Möglichkeit, Konflikte zu erkennen. Jede für die mobile Verwendung eingerichtete Tabelle muss einen Primärschlüssel haben. Der Primärschlüssel darf nie geändert werden. Sie müssen außerdem sicherstellen, dass ein in einer Datenbank eingefügter Primärschlüsselwert in keiner weiteren Datenbank eingefügt wird.

Es gibt mehrere Möglichkeiten, eindeutige Primärschlüssel zu generieren. Der Einfachheit halber wird in dieser praktischen Einführung die Methode der zusammengesetzten Primärschlüssel verwendet. Diese Methode erstellt Primärschlüssel mit mehreren Spalten, die in konsolidierten und entfernten Datenbanken eindeutig sind.

Aufgabe

1. Stellen Sie mithilfe des Dienstprogramms isql in Adaptive Server Enterprise als **sa** eine Verbindung mit der Datenbank pubs2 her. Führen Sie an einer Eingabeaufforderung folgenden Befehl in einer einzigen Zeile aus:

```
isql
-U sa
-P your-password-for-sa-account
-D pubs2
```

Wenn Sie entfernt auf Adaptive Server Enterprise zugreifen, geben Sie den Servernamen mit der Option -S an.

2. Die folgenden Zeilen sind nicht eindeutig und basieren auf dem zusammengesetzten Primärschlüssel, der für die Tabelle salesdetail erstellt wurde. Der Einfachheit halber löschen Sie die Zeilen, indem Sie folgende Anweisungen ausführen:

```
DELETE FROM salesdetail
WHERE stor_id = '5023'
AND ord_num = 'NF-123-ADS-642-9G3'
AND title_id = 'PC8888'

DELETE FROM salesdetail
WHERE stor_id = '5023'
AND ord_num = 'ZS-645-CAT-415-1B2'
AND title_id = 'BU2075'
```

3. Die folgenden Indizes beeinträchtigen die Erstellung von Primärschlüsseln in einem vorherigen Schritt. Zum Löschen der Indizes führen Sie folgende Anweisungen aus:

```
DROP INDEX authors.aidind
DROP INDEX titleauthor.taind
DROP INDEX titles.titleidind
DROP INDEX sales.salesind
```

4. Fügen Sie eindeutige Primärschlüssel hinzu, indem Sie folgende Anweisungen ausführen:

```
ALTER TABLE au_pix ADD PRIMARY KEY (au_id)
ALTER TABLE authors ADD PRIMARY KEY (au_id)
ALTER TABLE titleauthor ADD PRIMARY KEY (au_id, title_id)
ALTER TABLE titles ADD PRIMARY KEY (title_id)
ALTER TABLE discounts ADD PRIMARY KEY (discounttype)
ALTER TABLE stores ADD PRIMARY KEY (stor_id)
ALTER TABLE sales ADD PRIMARY KEY (stor_id, ord_num)
ALTER TABLE salesdetail ADD PRIMARY KEY (stor_id, ord_num, title_id)
```

Nachdem Sie diese Anweisungen ausgeführt haben, stellt der MobiLink-Server eine Verbindung mit der konsolidierten Datenbank her und richtet die Synchronisation für eine beliebige Anzahl entfernter Datenbanken ein.

Hinweis

Es ist möglich, Daten mit konsolidierten Datenbanken, die keine Primärschlüssel haben, zu synchronisieren. Sie müssen jedoch eigene Synchronisationsereignisse erstellen, die in Schattentabellen Aktionen ausführen, mit denen Zeilen in anderen Tabellen eindeutig identifiziert werden.

In einer späteren Lektion wird das entfernte Schema aus dem konsolidierten Schema erstellt, sodass das entfernte Schema dieselben Primärschlüssel enthält wie das konsolidierte Schema.

Spalten wurden gezielt ausgewählt, um eindeutige Primärschlüssel für alle Datenbanken sicherzustellen. Für die Tabelle sales besteht der Primärschlüssel aus den Spalten stor_id und ord_num. Jeder in die entfernte Tabelle sales eingegebene Wert muss eine eindeutige Bestellnummer haben (der stor_id-Wert ist immer gleich). Mit diesem Verfahren wird die Eindeutigkeit in jeder entfernten Sales-Tabelle sichergestellt. Der Primärschlüssel in der konsolidierten Sales-Tabelle verhindert Konflikte, wenn mehrere Buchhandlungen gleichzeitig Daten hochladen. Jeder Upload einer Buchhandlung ist von Uploads anderer Buchhandlungen eindeutig zu unterscheiden, da ihre stor_id-Werte unterschiedlich sind.

Für die Tabelle salesdetail besteht der Primärschlüssel aus den Spalten stor_id, ord_num und title_id. Eine Bestellung kann mehrere Buchtitel enthalten. Zeilen der entfernten Sales-Tabellen können dieselben Werte für stor_id und ord_num enthalten, doch sie müssen unterschiedliche Werte für title_id haben. Mit dieser Konfiguration wird die Eindeutigkeit in jeder entfernten Salesdetail-Tabelle sichergestellt. Ähnlich wie bei der sales-Tabelle ist jeder Upload einer Buchhandlung in die konsolidierte Datenbank aufgrund der unterschiedlichen stor_id eindeutig.

Ergebnisse

Zeilen, die nicht eindeutig sind, werden gelöscht und der konsolidierten Datenbank werden eindeutige Primärschlüssel hinzugefügt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 4: Herstellen der Verbindung mit MobiLink](#)“ auf Seite 149.

Siehe auch

- „Eindeutige Primärschlüssel“ [[MobiLink - Serveradministration](#)]

Lektion 4: Herstellen der Verbindung mit MobiLink

In dieser Lektion erstellen Sie eine ODBC-Datenquelle, die MobiLink mit der konsolidierten Datenbank verbindet.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 144.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden](#)“ auf Seite 142.

Aufgabe

1. Erstellen Sie eine ODBC-Datenquelle.

Sie sollten den von Adaptive Server Enterprise bereitgestellten ODBC-Treiber verwenden. Verwenden Sie für diese praktische Einführung folgende Konfigurationseinstellungen:

Felder der Registerkarte "Allgemein"	Wert
Data Source Name (Datenquellename)	ase_cons
Description (Beschreibung)	
Server Name (ASE Host Name) (Servername, ASE-Hostname)	localhost
Server Port (Serverport)	5000
Database Name (Datenbankname)	pubs2
Logon ID (Anmeldekennung)	sa
Use Cursors (Cursor verwenden)	nicht ausgewählt

Feld der Registerkarte "Transaction" (Transaktion)	Wert
Server Initiated Transactions (Serverinitiierte Transaktionen)	nicht ausgewählt

2. Testen Sie die ODBC-Verbindung:
 - a. Klicken Sie auf der Registerkarte **General (Allgemein)** auf **Test Connection (Verbindung testen)**.
Das Adaptive Server Enterprise-Anmeldefenster wird angezeigt.
 - b. Geben Sie das Kennwort für das Konto **sa** ein und klicken Sie auf **OK**.
Die Meldung **Anmeldung erfolgreich** wird angezeigt.

Ergebnisse

Die ODBC-Datenquelle wird erstellt und getestet.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 5: Erstellen eines MobiLink-Projekts und eines Synchronisationsmodells“](#) auf Seite 151.

Nachdem Sie die ODBC-Datenquelle konfiguriert haben, können Sie das MobiLink 16-Plug-In verwenden, um sich mit der konsolidierten Datenbank zu verbinden und ein Synchronisationsmodell zu erstellen.

Siehe auch

- <http://www.sybase.com/detail?id=1011880>

Lektion 5: Erstellen eines MobiLink-Projekts und eines Synchronisationsmodells

In dieser Lektion verbinden Sie sich mit der konsolidierten Datenbank, indem Sie ein neues MobiLink-Projekt erstellen. Ein Synchronisationsmodell wird automatisch erstellt, wenn Sie ein neues Projekt erstellen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 144.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden](#)“ auf Seite 142.

Aufgabe

1. Klicken Sie auf **Start » Programme » SQL Anywhere 16 » Administrationstools » Sybase Central**.

2. Klicken Sie auf **Extras » MobiLink 16 » Neues Projekt**.

Der **Assistent zum Erstellen eines Projekts** wird angezeigt.

3. Im Feld **Name** geben Sie **ase_project** ein.
4. Im Feld **Speicherort** geben Sie **C:\mlase** ein und klicken auf **Weiter**.
5. Im Feld **Anzeigename der Datenbank** geben Sie **ase_cons** ein.
6. Klicken Sie auf **Bearbeiten**.
7. Führen Sie die folgenden Aufgaben auf der Seite **Mit einer allgemeinen ODBC-Datenbank verbinden** durch:
 - a. Im Feld **Benutzer-ID** geben Sie **sa** ein.
 - b. Geben Sie im Feld **Kennwort** das Kennwort für das **sa**-Konto ein.
 - c. Klicken Sie im Feld **ODBC-Datenquellenname** auf **Durchsuchen** und wählen Sie **ase_cons**.
 - d. Klicken Sie auf **OK** und auf **Speichern**.
8. Wählen Sie die Option **Kennwort speichern** aus und klicken Sie auf **Weiter**.

9. Wählen Sie auf der Seite **Neues entferntes Datenbankschema** in der Liste **Welche konsolidierten Datenbanktabellen und Spalten sollen in der entfernten Datenbank enthalten sein?** die folgenden Tabellen:

- au_pix
- authors
- discounts
- sales
- salesdetail
- stores
- titleauthor
- titles

Klicken Sie auf **Weiter**.

10. Wählen Sie die Option **Dem Projekt einen Namen für ein entferntes Schema hinzufügen** aus.

11. Geben Sie **ase_remote_schema** für den Namen des entfernten Schemas ein und klicken Sie dann auf **Weiter**.

12. Aktivieren Sie das Kontrollkästchen **SQL Anywhere** und klicken Sie dann auf **Fertig stellen**.

Wenn die konsolidierte Datenbank zum ersten Mal von MobiLink verwendet wird, werden Sie aufgefordert, die MobiLink-Systemeinrichtung zu installieren. Durch die Installation der MobiLink-Systemeinrichtung werden MobiLink-Systemtabellen und -Prozeduren hinzugefügt. Klicken Sie auf **Ja** und dann auf **OK**.

Ergebnisse

Das MobiLink-Projekt- und Synchronisationsmodell wird erstellt.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 6: Ein Synchronisationsmodell ändern“ auf Seite 152](#).

Lektion 6: Ein Synchronisationsmodell ändern

In dieser Lektion ändern Sie ein Synchronisationsmodell.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Planen von Schemata“ auf Seite 144](#).

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: [„Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden“ auf Seite 142](#).

Aufgabe

1. Klicken Sie mit der rechten Maustaste auf das `ase_remote_schema`-Synchronisationsmodell und wählen Sie dann **Eigenschaften**.
2. Führen Sie die folgenden Aufgaben durch:
 - a. Geben Sie im ersten Feld **sync_ase** ein.
 - b. Geben Sie im Feld **Publikationsname** **sync_ase_publication** ein.
 - c. Geben Sie im Feld **Skriptversion** **sync_ase_scriptversion** ein.

Die Publikation ist das Objekt in der entfernten Datenbank, das festlegt, welche Daten synchronisiert werden. MobiLink-Serverskripten legen fest, wie hochgeladene Daten von entfernten Datenbanken in der konsolidierten Datenbank verwendet werden und Skriptversionen Skripten gruppieren. Sie können unterschiedliche Skriptversionen für verschiedene Anwendungen verwenden. Auf diese Weise können Sie einen einzelnen MobiLink-Server verwalten und gleichzeitig mehrere Anwendungen synchronisieren.
 - d. Klicken Sie auf **Übernehmen** und dann auf **OK**.
3. Legen Sie für jede Tabelle im Synchronisationsmodell die Richtung fest, in der die Daten synchronisiert werden.

Klicken Sie auf die Registerkarte **Zuordnungen** im rechten Fensterausschnitt und legen Sie die Zeilen in der Spalte **Zuordnungsrichtung** wie folgt fest:

- Die Tabellen **sales** und **salesdetail** müssen mit **Bidirektional** (Upload und Download) festgelegt werden.
- Die restlichen Tabellen müssen auf **Nur auf entfernte Datenbank herunterladen** festgelegt werden.

4. Filtern Sie die Zeilen, die in die entfernte Datenbank heruntergeladen wurden, nach der entfernten ID.
 - a. Wählen Sie die Zeile, in der die Tabelle **stores** enthalten ist, und klicken Sie dann auf die Registerkarte **Download-Teilmenge**.
 - b. Ändern Sie die **Download-Teilmenge** auf **Benutzerdefiniert**.
 - c. Filtern Sie die Zeilen nach entfernten IDs, die die entfernte Datenbank eindeutig identifizieren, indem Sie eine Einschränkung der WHERE-Klausel des `download_cursor`-Skripts hinzufügen.

Geben Sie im Feld **In der WHERE-Klausel des Download-Cursors zu verwendender SQL-Ausdruck** eine Suchbedingung ein. Das folgende SQL-Skript kann beispielsweise für die **stores**-Tabelle verwendet werden:

```
"dbo"."stores"."stor_id" = {ml s.remote_id}
```

Das Download-Cursorskript legt fest, welche Spalten und Zeilen von den einzelnen Tabellen in die entfernte Datenbank heruntergeladen werden. Die Suchbedingung stellt sicher, dass Sie nur Informationen über einen einzigen Vertriebsmitarbeiter herunterladen, nämlich den Vertriebsmitarbeiter, dessen ID mit der entfernten ID der Datenbank übereinstimmt.

- d. Klicken Sie auf die Registerkarte **Download-Löschen-Teilmenge** und ändern Sie die **Download-Löschen-Teilmenge** von **Identisch** auf **Keine**.

5. Wiederholen Sie den vorherigen Schritt für die Zeilen mit den Tabellen **sales**, **salesdetail** und **discounts**.

Hinweis

Sie müssen die im SQL-Skript benannte Tabelle auf den Tabellennamen in der Zeile ändern, die Sie bearbeiten.

Verwenden Sie das folgende WHERE-Klauselskript für die Tabelle **sales**:

```
"dbo"."sales"."stor_id" = {ml s.remote_id}
```

Verwenden Sie das folgende WHERE-Klauselskript für die Tabelle **salesdetail**:

```
"dbo"."salesdetail"."stor_id" = {ml s.remote_id}
```

Verwenden Sie das folgende WHERE-Klauselskript für die Tabelle **discounts**:

```
"dbo"."discounts"."stor_id" = {ml s.remote_id}
```

6. Speichern Sie das Synchronisationsmodell.

Klicken Sie auf **Datei » Speichern**.

Ergebnisse

Das Synchronisationsmodell ist vollständig und kann nun per Deployment bereitgestellt werden.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 7: Deployment des Synchronisationsmodells](#)“ auf Seite 154.

Siehe auch

- „Einrichten einer konsolidierten Datenbank“ [*MobiLink - Serveradministration*]
- „Systemtabellen des MobiLink-Servers“ [*MobiLink - Serveradministration*]
- „Systemprozeduren des MobiLink-Servers“ [*MobiLink - Serveradministration*]
- „download_delete_cursor-Skripten“ [*MobiLink - Serveradministration*]
- „Konfliktverarbeitung“ [*MobiLink - Serveradministration*]
- „Konfliktlösung“ [*MobiLink - Serveradministration*]
- „Publikationen“ [*MobiLink - Clientadministration*]
- „Synchronisationsmodellaufgaben“ auf Seite 33
- „Downloadtyp ändern“ auf Seite 38
- „Konflikterkennung und -lösung ändern“ auf Seite 46
- „Tabellen- und Spaltenzuordnungen“ auf Seite 34

Lektion 7: Deployment des Synchronisationsmodells

Sie können mit dem **Assistenten für das Deployment eines Synchronisationsmodells** das Deployment der konsolidierten Datenbank und der entfernten Datenbank vornehmen. Sie können das Deployment

dieser Datenbanken einzeln oder zusammen durchführen. Der **Assistent für das Deployment eines Synchronisationsmodells** führt Sie durch die Schritte für die Konfiguration der Optionen für das Deployment.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 144.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden](#)“ auf Seite 142.

Aufgabe

1. Erweitern Sie im linken Fensterausschnitt von Sybase Central unter **MobiLink 16** die Elemente **ase_project**, **Synchronisationsmodelle** und dann **sync_ase**.
2. Klicken Sie auf **Datei » Speichern**.
3. Akzeptieren Sie die Standardeinstellung für das Feld **Wählen Sie den Ordner für vom Assistenten generierte Dateien** und klicken Sie auf **Weiter**.
4. Akzeptieren Sie die Standardeinstellungen für die Seite **Client-Netzwerkoptionen** und klicken Sie auf **Weiter**.
5. Auf der Seite **MobiLink-Benutzer und -Kennwort** unter **Welchen Benutzer und welches Kennwort für MobiLink möchten Sie verwenden?** wählen Sie **Diese verwenden** und führen die folgenden Aufgaben aus:
 - Geben Sie im Feld **MobiLink-Benutzer** **ase_remote** ein.
 - Geben Sie im Feld **MobiLink-Kennwort** **ase_pass** ein.
 - Wählen Sie **Registrieren Sie diesen Benutzer in der konsolidierten Datenbank. Registrierte Benutzer sind berechtigt, zu synchronisieren**. Klicken Sie auf **Weiter**.Klicken Sie auf **Weiter** und anschließend nochmals auf **Weiter**.
6. Auf der Seite **Synchronisationsprofil** geben Sie **sync_ase_profile** im Feld **Synchronisationsprofilname** ein und klicken auf **Weiter**.
7. Führen Sie die folgenden Aufgaben auf der Seite **Auswählen, wie Datenbanken für die Synchronisation vorbereitet werden** durch:
 - a. Für **Was möchten Sie mit dem SQL-Skript tun, das erstellt wird, um die konsolidierte Datenbank für die Synchronisation vorzubereiten?** wählen Sie **Mit der konsolidierten Datenbank ausführen**.
 - b. Bei **Was möchten Sie mit dem SQL-Skript tun, das erstellt wird, um die konsolidierte Datenbank für die Synchronisation vorzubereiten?** wählen Sie **Mit einer neuen entfernten Datenbank ausführen**. Klicken Sie auf **Weiter**.

8. Klicken Sie auf **Fertig stellen**.

9. Klicken Sie auf **Schließen**.

Ergebnisse

Ihre konsolidierte Datenbank ist vollständig für die Synchronisation mit mehreren entfernten Clients konfiguriert und Sie haben erfolgreich ein Deployment auf einen entfernten Client vorgenommen. Um Deployments auf andere entfernte Clients vorzunehmen, können Sie diesen Assistenten erneut ausführen. Hierzu müssen Sie einen neuen MobiLink-Benutzer erstellen und die Auswahl des Deployments der konsolidierten Datenbank und des MobiLink-Servers aufheben. Da sie bereits per Deployment bereitgestellt wurden, müssen Sie nur noch das Deployment anderer entfernter Synchronisationsclients vornehmen.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 8: Starten von Server und Client“ auf Seite 156](#).

Siehe auch

- [„Deployment des Synchronisationsmodells“ auf Seite 50](#)
- [„SQL Anywhere-Datenbanken als entfernte Datenbanken verwenden“ \[*MobiLink - Clientadministration*\]](#)
- [„MobiLink-Benutzer“ \[*MobiLink - Clientadministration*\]](#)

Lektion 8: Starten von Server und Client

In dieser Lektion starten Sie den MobiLink-Server und die entfernte Datenbank.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Planen von Schemata“ auf Seite 144](#).

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: [„Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden“ auf Seite 142](#).

Kontext und Bemerkungen

In einer früheren Lektion haben Sie das Download-Cursor-Skript geändert, um Informationen zu einer Buchhandlung herunterzuladen. In dieser Lektion legen Sie die Buchhandlung fest, indem Sie die entfernte ID auf die ID der Buchhandlung setzen.

Aufgabe

1. Navigieren Sie an einer Eingabeaufforderung zu dem Verzeichnis, in dem Sie das Synchronisationsmodell erstellt haben. (Dies ist das Stammverzeichnis, das Sie im ersten Schritt des **Assistenten zum Erstellen eines Synchronisationsmodells** gewählt haben.)

Wenn Sie die vorgeschlagenen Verzeichnisnamen verwendet haben, navigieren Sie zum folgenden Verzeichnis: *mlase\ase_project\sync_ase_deploy*.

2. Führen Sie zum Start des MobiLink-Servers folgenden Befehl aus:

```
mlsrv.bat "DSN=ase_cons;UID=sa;PWD=sa;"
```

- **mlsrv.bat** Die Befehlsdatei, die zum Starten des MobiLink-Servers erstellt wurde.
- **dsn** Der Name der ODBC-Datenquelle.
- **uid** Der Benutzername, den Sie verwenden, um eine Verbindung mit der konsolidierten Datenbank herzustellen (der Standard-Benutzername für Adaptive Server Enterprise ist **sa**).
- **pwd** Das Kennwort, das Sie für die Verbindung als **sa** verwenden.

Wenn dieser Befehl erfolgreich ausgeführt wurde, erscheint im Meldungsfenster des MobiLink-Servers die Meldung **MobiLink-Server gestartet**.

Wenn der MobiLink-Server nicht gestartet werden kann, überprüfen Sie die Verbindungsinformationen für die konsolidierte Datenbank.

3. Navigieren Sie an der Eingabeaufforderung zu dem Verzeichnis, in dem der **Assistent für das Deployment eines Synchronisationsmodells** die entfernte Datenbank erstellt hat.

Wenn Sie die vorgeschlagenen Verzeichnisnamen verwendet haben, navigieren Sie zum folgenden Verzeichnis: *mlase\ase_project\sync_ase_deploy*.

4. Führen Sie zum Start der SQL Anywhere-Datenbank folgenden Befehl aus:

```
dbsrv16 -n remote_eng sync_ase_remote.db -n remote_db
```

- **dbsrv16** Der Datenbankserver, der zum Start der SQL Anywhere-Datenbank verwendet wird.
- **remote_eng** Der Name des Datenbankservers.
- **sync_ase_remote.db** Die Datenbankdatei, die auf remote_eng gestartet wurde.
- **remote_db** Der Name der Datenbank auf remote_eng.

Ergebnisse

Ein SQL Anywhere-Datenbankserver namens remote_eng wird gestartet und die Datenbank remote_db wird geladen.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 9: Einstellen der entfernten ID](#)“ auf Seite 158.

Siehe auch

- „Syntax des SQL Anywhere-Datenbankservers“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Per Deployment bereitgestellte Synchronisationsmodelle“ auf Seite 54
- „MobiLink-Server“ [[MobiLink - Serveradministration](#)]

Lektion 9: Einstellen der entfernten ID

Im entfernten Schema repräsentiert jede entfernte Datenbank eine Buchhandlung. Die von Ihnen erstellten Synchronisationsskripten enthalten Logik, die den MobiLink-Server anweisen, eine Teilmenge der Daten basierend auf der entfernten ID der entfernten Datenbank herunterzuladen. Sie müssen die entfernte ID der Datenbank auf den Wert einer gültigen Buchhandlungs-ID festlegen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 144.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden](#)“ auf Seite 142.

Kontext und Bemerkungen

Es ist wichtig, diesen Schritt vor der ersten Synchronisation auszuführen, da das entfernte Gerät bei der ersten Synchronisation alle Informationen zu der gewählten Buchhandlung (in diesem Fall Thoreau Reading Discount Chain) herunterlädt.

Aufgabe

1. Wählen Sie eine gültige Buchhandlungs-ID aus:
 - a. Stellen Sie mithilfe des Dienstprogramms isql in Adaptive Server Enterprise als **sa** eine Verbindung mit der Datenbank pubs2 her. Führen Sie an einer Eingabeaufforderung folgenden Befehl in einer einzigen Zeile aus:

```
isql
-U sa
-P your-password-for-sa-account
-D pubs2
```

Wenn Sie entfernt auf Adaptive Server Enterprise zugreifen, geben Sie den Servernamen mit der Option -S an.

- b. Um eine Liste gültiger Buchhandlungs-IDs in der Tabelle stores anzuzeigen, führen Sie folgende Anweisung aus:

```
SELECT * FROM stores
```

In dieser praktischen Einführung repräsentiert die entfernte Datenbank die Buchhandlung Thoreau Reading Discount Chain, deren Buchhandlungs-ID den Wert 5023 hat.

- c. Führen Sie zum Beenden von isql folgenden Befehl aus:

```
exit
```

2. Um die entfernte ID der Datenbank auf 5023 festzulegen, führen Sie folgenden Befehl aus (in einer einzigen Zeile):

```
dbisql
-c "SERVER=remote_eng;DBN=remote_db;UID=DBA;PWD=sql"
"SET OPTION PUBLIC.ml_remote_id='5023'"
```

- **dbisql** Die Anwendung, die zur Ausführung von SQL-Befehlen in einer SQL Anywhere-Datenbank verwendet wurde.
- **server** Legt den Datenbankservernamen auf remote_eng fest.
- **dbn** Legt den Datenbanknamen auf remote_db fest.
- **uid** Gibt den Benutzernamen an, mit dem eine Verbindung mit der entfernten Datenbank hergestellt wird.
- **pwd** Gibt das Kennwort an, mit dem eine Verbindung mit der entfernten Datenbank hergestellt wird.
- **SET OPTION PUBLIC.ml_remote_id='5023'** Der SQL-Befehl, mit dem die entfernte ID auf 5023 festgelegt wird.

Ergebnisse

Setzen Sie die entfernte ID der Datenbank auf 5023, wobei es sich um den Wert der Speicher-ID handelt.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 10: Synchronisation“ auf Seite 159](#).

Siehe auch

- [„Entfernte IDs“ \[MobiLink - Clientadministration\]](#)

Lektion 10: Synchronisation

In dieser Lektion synchronisieren Sie den entfernten Client erstmals mithilfe des Dienstprogramms dbmlsync.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Planen von Schemata“ auf Seite 144](#).

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: [„Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden“ auf Seite 142](#).

Kontext und Bemerkungen

Dbmlsync stellt eine Verbindung mit der entfernten Datenbank her, lädt die Synchronisationsdaten aus der entfernten Datenbank, liest die Transaktionslogs und generiert die Upload-Daten. Dbmlsync stellt eine Verbindung zum MobiLink-Server her, authentifiziert sich selbst beim MobiLink-Server und führt dann

alle Uploads und Downloads aus, die zur Synchronisation der entfernten und konsolidierten Datenbanken, die auf einer Publikation in der entfernten Datenbank basieren, erforderlich sind.

Aufgabe

- Führen Sie an einer Eingabeaufforderung folgenden Befehl in einer einzigen Zeile aus:

```
dbmlsync -c "SERVER=remote_eng;DBN=remote_db;UID=DBA;PWD=sql;"  
-n sync_ase_publication  
-u ase_remote -mp ase_pass
```

- **dbmlsync** Die Synchronisationsanwendung.
- **SERVER** Gibt den Namen des Fremdservers an.
- **DBN** Der Name der entfernten Datenbank.
- **UID** Gibt den Benutzernamen an, mit dem eine Verbindung mit der entfernten Datenbank hergestellt wird.
- **PWD** Gibt das Kennwort an, mit dem eine Verbindung mit der entfernten Datenbank hergestellt wird.
- **sync_ase_publication** Die Publikation auf dem entfernten Gerät, das zur Ausführung der Synchronisation verwendet wird. (Diese Publikation wurde mit dem **Assistenten zum Erstellen eines Synchronisationsmodells** erstellt.)
- **ase_remote** Der Benutzername, der zur Authentifizierung mit dem MobiLink-Server verwendet wird.
- **ase_pass** Das Kennwort, das zur Authentifizierung mit dem MobiLink-Server verwendet wird.

Hinweis

Wenn Sie die Anwendung dbmlsync auf einem anderen Computer als dem MobiLink-Server ausführen, müssen Sie auch Argumente mit der Position des MobiLink-Servers übergeben.

Ergebnisse

Der Verarbeitungsfortschritt der Synchronisation wird im **Meldungsfenster des SQL Anywhere MobiLink-Clients** angezeigt. Wenn dieser Befehl erfolgreich ausgeführt wird, füllt die Anwendung dbmlsync die entfernte Datenbank mit einer Teilmenge der Informationen aus der konsolidierten Datenbank.

Wenn die Synchronisation fehlschlägt, überprüfen Sie die Verbindungsinformationen, die Sie an die dbmlsync-Anwendung übermitteln, sowie den MobiLink-Benutzernamen und das Kennwort. Wenn das Problem weiterhin besteht, überprüfen Sie den verwendeten Publikationsnamen. Stellen Sie sicher, dass die konsolidierte Datenbank und der MobiLink-Server ausgeführt werden. Sie können auch den Inhalt des Synchronisationslogs (Server und Client) überprüfen.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 11: Einfügen von Daten in die entfernte Datenbank](#)“ auf Seite 161.

Siehe auch

- „Der Synchronisationsprozess“ auf Seite 14
- „dbmlsync-Syntax“ [*MobiLink - Clientadministration*]

Lektion 11: Einfügen von Daten in die entfernte Datenbank

Nach der erfolgreichen Synchronisation des entfernten Clients mit der konsolidierten Datenbank durch den MobiLink-Server sollte die entfernte Datenbank mit den Informationen einer Buchhandlung gefüllt sein. Sie können den Inhalt der entfernten Datenbank in Sybase Central mit dem SQL Anywhere 16-Plug-In überprüfen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Planen von Schemata](#)“ auf Seite 144.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: „[Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden](#)“ auf Seite 142.

Aufgabe

1. Starten Sie Sybase Central.
2. Stellen Sie eine Verbindung zur entfernten Datenbank her:
 - a. Klicken Sie im linken Fensterausschnitt mit der rechten Maustaste auf **SQL Anywhere 16** und wählen Sie **Verbinden**.
 - b. Wählen Sie in der Dropdown-Liste **Authentifizierung** die Option **Datenbank** und führen Sie die folgenden Schritte aus:
 - i. Im Feld **Benutzer-ID** geben Sie **DBA** ein.
 - ii. Im Feld **Kennwort** geben Sie **sql** ein.
 - c. Wählen Sie in der Dropdown-Liste **Aktion** die Option **Mit einer laufenden Datenbank auf diesem Computer verbinden**.
 - d. Geben Sie im Feld **Servername** den Wert **remote_eng** und im Feld **Datenbankname** den Wert **remote_db** ein.
 - e. Klicken Sie auf **Verbinden**.
3. Wenn die Tabellen, die anhand der konsolidierten Datenbank erstellt wurden, nicht zu sehen sind, führen Sie folgende Schritte aus:
 - a. Klicken Sie mit der rechten Maustaste auf **remote_db** und dann auf **Eigentümerfilter konfigurieren**.
 - b. Wählen Sie **dbo** aus und klicken Sie auf **OK**.

Die anhand der konsolidierten Datenbank erstellten Tabellen werden im linken Fensterausschnitt angezeigt. Die Eigentümerschaft dieser Tabellen durch den dbo wird in der entfernten Datenbank beibehalten.

4. Wählen Sie eine beliebige entfernte Tabelle aus und klicken Sie dann auf der Registerkarte **Daten** in den rechten Fensterausschnitt.

In den Tabellen sales, salesdetail und stores beziehen sich alle Datensätze auf die Buchhandlung mit der ID 5023. Diese einzelne Buchhandlung hat nichts mit den Verkaufsinformationen anderer Buchhandlungen zu tun. Aus diesem Grund legen Sie die Synchronisationsskripten so fest, dass Zeilen anhand der entfernten ID gefiltert werden, und setzen die entfernte ID dieser Datenbank auf den Wert einer bestimmten Buchhandlungs-ID. Nun benötigt die Datenbank der betreffenden Buchhandlung weniger Speicherplatz und weniger Zeit für die Synchronisation. Da die entfernte Datenbank auf eine minimale Größe begrenzt wird, können häufige Vorgänge, wie etwa die Eingabe eines neuen Verkaufs oder die Verarbeitung einer Rückvergütung eines früheren Verkaufs, schneller und effizienter ausgeführt werden.

Ergebnisse

Die Daten in der entfernten Datenbank werden angezeigt.

Nächste Schritte

Gehen Sie weiter zu [„Aufräumen“ auf Seite 162](#).

Aufräumen

Erstellen Sie die Datenbank pubs2 neu und entfernen Sie das gesamte Material der praktischen Einführung von Ihrem Computer.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Planen von Schemata“ auf Seite 144](#).

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien besitzen, die im Abschnitt "Berechtigungen und Privilegien" am Anfang dieser praktischen Einführung aufgeführt werden: [„Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden“ auf Seite 142](#).

Aufgabe

1. Erstellen Sie die Datenbank pubs2 neu.

Geben Sie zur Ausführung des Skripts, das die Datenbank pubs2 installiert, folgenden Befehl ein:

```
isql
-U sa
-P your-password-for-sa-account
-i %SYBASE%\%SYBASE_ASE%\scripts\instpbs2
```


Wenn Sie entfernt auf Adaptive Server Enterprise zugreifen, geben Sie den Servernamen mit der Option -S an. Außerdem müssen Sie die Datei instpbs2 lokal auf Ihren Computer kopieren. Die Option -i muss aktualisiert werden, sodass sie den neuen Speicherort der Datei instpbs2 angibt.

2. Löschen Sie das Synchronisationsmodell.
 - a. Starten Sie Sybase Central.
 - b. Doppelklicken Sie auf **MobiLink 16** im rechten Fensterausschnitt.
Das Modell **sync_ase** wird angezeigt.
 - c. Rechtsklicken Sie auf **sync_ase** und wählen Sie **Löschen**.
3. Löschen Sie die entfernte Datenbank mit dem Dienstprogramm dberase.

Führen Sie den folgenden Befehl aus:

```
dberase sync_ase\remote\sync_ase_remote.db
```

Ergebnisse

Die Datenbank pubs2 wird regeneriert und das gesamte Material der praktischen Einführung wird von Ihrem Computer gelöscht.

Nächste Schritte

Keine.

Praktische Einführung: Der Einsatz von Java oder .NET für die benutzerdefinierte Authentifizierung

MobiLink-Synchronisationsskripten können in SQL, Java oder .NET geschrieben werden. Sie können Java oder .NET verwenden, um zu jedem Zeitpunkt einer Synchronisation benutzerdefinierte Aktionen hinzuzufügen.

In dieser praktischen Einführung fügen Sie dem Verbindungsereignis `authenticate_user` eine Java- oder .NET-Methode hinzu. Mit dem Ereignis `authenticate_user` können Sie ein benutzerdefiniertes Authentifizierungsschema erstellen, das die in MobiLink integrierte Clientauthentifizierung aufhebt.

Erforderliche Software

- SQL Anywhere 16
- Java Software Development Kit oder Microsoft .NET Framework

Kenntnisse und Erfahrungen

Hierzu benötigen Sie Folgendes:

- Java- oder .NET-Kenntnisse
- Grundkenntnisse über MobiLink-Ereignisskripten

Privilegien

Sie müssen die folgenden Privilegien für die Datenbank CustDB haben:

- SYS_REPLICATION_ADMIN_ROLE-Systemrolle
- CREATE ANY TRIGGER-Systemprivileg
- CREATE ANY VIEW-Systemprivileg
- EXECUTE ANY PROCEDURE-Systemprivileg

Übersicht

In dieser praktischen Einführung wird Folgendes veranschaulicht:

- Kompilieren einer Quelldatei mit API-Referenzen des MobiLink-Servers
- Festlegen von Klassenmethoden für bestimmte Ereignisse auf Tabellenebene
- Ausführen des MobiLink-Servers (mlsrv16) mit der -sl-Option
- Testen der Synchronisation mit einer Windows-Beispiel-Clientanwendung

Ziele

Sie gewinnen Kenntnisse bzw. Vertrautheit mit folgenden Komponenten:

- Benutzerdefinierte MobiLink-Authentifizierung

Siehe auch

- „Benutzerauthentifizierungsverfahren“ [[MobiLink - Clientadministration](#)]
- „Authentifizierung bei externen Servern“ [[MobiLink - Clientadministration](#)]
- „Synchronisationsskripten in .NET“ [[MobiLink - Serveradministration](#)]
- „Schreiben eines Synchronisationsskripts in Java“ [[MobiLink - Serveradministration](#)]

Lektion 1: Eine Java-Klasse für die benutzerdefinierte Authentifizierung (serverseitig) erstellen

In dieser Lektion kompilieren Sie eine Klasse mit Java-Logik für die benutzerdefinierte Authentifizierung.

Voraussetzungen

Der MobiLink-Server muss Zugriff auf die Klassen in *mlscript.jar* haben, damit die Java-Synchronisationslogik ausgeführt werden kann. *mlscript.jar* enthält eine Sammlung von MobiLink Java-

Server-API-Klassen zur Verwendung in Ihren Java-Methoden. Wenn Sie Ihre Java-Klasse kompilieren, referenzieren Sie *mlscript.jar*.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Der Einsatz von Java oder .NET für die benutzerdefinierte Authentifizierung](#)“ auf Seite 163.

Kontext und Bemerkungen

Hinweise zum Erstellen von .NET-Klassen für die benutzerdefinierte Authentifizierung finden Sie unter „[Lektion 1: Eine .NET-Klasse für die benutzerdefinierte Authentifizierung \(serverseitig\) erstellen](#)“ auf Seite 166.

Aufgabe

1. Erstellen Sie eine Klasse mit dem Namen `MobiLinkAuth` und schreiben Sie eine `authenticateUser`-Methode.

Die Klasse `MobiLinkAuth` umfasst die Methode `authenticateUser` für das Synchronisationsereignis `authenticate_user`. Das Ereignis `authenticate_user` enthält Parameter für Benutzer und Kennwort. Sie geben das Authentifizierungsergebnis im INOUT-Parameter `authentication_status` zurück.

Hinweis

Sie registrieren die Methode `authenticateUser` für das Synchronisationsereignis `authenticate_user` in „[Lektion 2: Registrieren von Java- oder .NET-Skripten für das Ereignis `authenticate_user`](#)“ auf Seite 168

Verwenden Sie den folgenden Code für Ihre Serveranwendung:

```
import ianywhere.ml.script.*;

public class MobiLinkAuth {
    public void authenticateUser (
        ianywhere.ml.script.InOutInteger authentication_status,
        String user,
        String pwd,
        String newPwd ) {

        if (user.startsWith("128")) {
            // success: an auth status code of 1000
            authentication_status.setValue(1000);
        } else {
            // fail: an authentication_status code of 4000
            authentication_status.setValue(4000);
        }
    }
}
```

Dieser Programmcode zeigt einen einfachen Fall einer angepassten Benutzerauthentifizierung. Die Authentifizierung ist erfolgreich, wenn der Client auf die konsolidierte Datenbank über einen Benutzernamen zugreift, der mit **128** beginnt.

2. Speichern Sie den Programmcode.

In dieser praktischen Einführung wird davon ausgegangen, dass *c:\MLauth* das Arbeitsverzeichnis für serverseitige Komponenten ist. Speichern Sie die Datei als *MobiLinkAuth.java* in diesem Verzeichnis.

3. Kompilieren Sie Ihre Klassendatei.
 - a. Navigieren Sie zu dem Verzeichnis, das Ihre Java-Datei enthält.
 - b. Kompilieren Sie die *MobiLinkAuth*-Klasse und verweisen Sie auf die *MobiLink-Server Java API-Bibliothek*.

Führen Sie den folgenden Befehl aus, wobei Sie *C:\Program Files\SQL Anywhere 16* durch Ihr *SQL Anywhere 16*-Verzeichnis ersetzen:

```
javac MobiLinkAuth.java -classpath "C:\Program Files\SQL Anywhere 16\java\mlscript.jar"
```

Ergebnisse

Die Datei *MobiLinkAuth.class* wird generiert.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 2: Registrieren von Java- oder .NET-Skripten für das Ereignis *authenticate_user*](#)“ auf Seite 168.

Siehe auch

- „*authenticate_user* (Verbindungsereignis)“ [*MobiLink - Serveradministration*]
- „Benutzerauthentifizierung mit Java und .NET“ [*MobiLink - Clientadministration*]
- „Java-Synchronisationsbeispiel“ [*MobiLink - Serveradministration*]
- „.NET-Synchronisationsbeispiel“ [*MobiLink - Serveradministration*]

Lektion 1: Eine .NET-Klasse für die benutzerdefinierte Authentifizierung (serverseitig) erstellen

In dieser Lektion kompilieren Sie eine Klasse mit .NET-Logik für die benutzerdefinierte Authentifizierung.

Voraussetzungen

Der *MobiLink-Server* muss Zugriff auf die Klassen in *iAnywhere.MobiLink.Script.dll* haben, damit die .NET-Synchronisationslogik ausgeführt werden kann. *iAnywhere.MobiLink.Script.dll* enthält eine Sammlung von *MobiLink .NET-Server-API-Klassen* zur Verwendung in Ihren .NET-Methoden. Wenn Sie Ihre .NET-Klasse kompilieren, referenzieren Sie *iAnywhere.MobiLink.Script.dll*.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Der Einsatz von Java oder .NET für die benutzerdefinierte Authentifizierung](#)“ auf Seite 163.

Kontext und Bemerkungen

Hinweise zum Erstellen von .NET-Klassen für die benutzerdefinierte Authentifizierung finden Sie unter [„Lektion 1: Eine Java-Klasse für die benutzerdefinierte Authentifizierung \(serverseitig\) erstellen“](#) auf Seite 164.

Aufgabe

1. Erstellen Sie eine Klasse mit dem Namen `MobiLinkAuth` und schreiben Sie eine `authenticateUser`-Methode.

Die Klasse `MobiLinkAuth` umfasst die Methode `authenticateUser` für das Synchronisationsereignis `authenticate_user`. Das Ereignis `authenticate_user` enthält Parameter für Benutzer und Kennwort. Sie geben das Authentifizierungsergebnis im INOUT-Parameter `authentication_status` zurück.

Hinweis

Sie registrieren die Methode `authenticateUser` für das Synchronisationsereignis `authenticate_user` in [„Lektion 2: Registrieren von Java- oder .NET-Skripten für das Ereignis `authenticate_user`“](#) auf Seite 168

Verwenden Sie den folgenden Code für Ihre Serveranwendung:

```
using iAnywhere.MobiLink.Script;

public class MobiLinkAuth {
    public void authenticateUser(
        ref int authentication_status,
        string user,
        string pwd,
        string newPwd ) {

        if(user.StartsWith("128")) {
            // success: an auth status code of 1000
            authentication_status = 1000;
        } else {
            // fail: and authentication_status code of 4000
            authentication_status = 4000;
        }
    }
}
```

Dieser Programmcode zeigt einen einfachen Fall einer angepassten Benutzerauthentifizierung. Die Authentifizierung ist erfolgreich, wenn der Client auf die konsolidierte Datenbank über einen Benutzernamen zugreift, der mit **128** beginnt.

2. Speichern Sie den Programmcode.

In dieser praktischen Einführung wird davon ausgegangen, dass `c:\MLauth` das Arbeitsverzeichnis für serverseitige Komponenten ist. Speichern Sie die Datei als `MobiLinkAuth.cs` in diesem Verzeichnis.

3. Kompilieren Sie Ihre Klassendatei.

- a. Navigieren Sie zu dem Verzeichnis, das Ihre C#-Datei enthält.

- b. Kompilieren Sie die MobiLinkAuth-Klasse und verweisen Sie auf die MobiLink-Server .NET API-Bibliothek.

Führen Sie den folgenden Befehl aus, wobei Sie *C:\Program Files\SQL Anywhere 16* durch Ihr SQL Anywhere 16-Verzeichnis ersetzen:

```
csc /out:MobiLinkAuth.dll /target:library /reference:"C:\Program Files
\SQL Anywhere 16\Assembly\v2\iAnywhere.MobiLink.Script.dll"
MobiLinkAuth.cs
```

Ergebnisse

Die Assembly *MobiLinkAuth.dll* wird generiert.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 2: Registrieren von Java- oder .NET-Skripten für das Ereignis authenticate_user](#)“ auf Seite 168.

Siehe auch

- „authenticate_user (Verbindungsereignis)“ [[MobiLink - Serveradministration](#)]
- „Benutzerauthentifizierung mit Java und .NET“ [[MobiLink - Clientadministration](#)]
- „Java-Synchronisationsbeispiel“ [[MobiLink - Serveradministration](#)]
- „.NET-Synchronisationsbeispiel“ [[MobiLink - Serveradministration](#)]

Lektion 2: Registrieren von Java- oder .NET-Skripten für das Ereignis authenticate_user

SQL Anywhere wird mit einer SQL Anywhere-Beispieldatenbank (CustDB) geliefert, die bereits für die Synchronisation eingerichtet ist. Die CustDB-Tabelle ULCustomer ist zum Beispiel eine synchronisierte Tabelle, die eine Reihe von Skripten auf Tabellenebene unterstützt. In dieser Lektion registrieren Sie die Methode authenticateUser von MobiLinkAuth für das Synchronisationsereignis authenticate_user. Sie fügen dieses Skript zur CustDB hinzu (Beispieldatenbank von MobiLink).

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Eine Java-Klasse für die benutzerdefinierte Authentifizierung \(serverseitig\) erstellen](#)“ auf Seite 164.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Der Einsatz von Java oder .NET für die benutzerdefinierte Authentifizierung](#)“ auf Seite 163.

Kontext und Bemerkungen

CustDB ist ein konsolidierter Datenbankserver für sowohl UltraLite- als auch SQL Anywhere-Clients. Die CustDB-Datenbank hat eine ODBC-Datenquelle namens **SQL Anywhere 16 CustDB**.

Aufgabe

1. Verbinden Sie sich mit der Beispieldatenbank über Interactive SQL.

Führen Sie den folgenden Befehl aus:

```
dbisql -c "DSN=SQL Anywhere 16 CustDB"
```

2. Verwenden Sie die gespeicherte Prozedur `ml_add_java_connection_script` oder `ml_add_dnet_connection_script`, um die Methode `authenticateUser` für das Ereignis `authenticate_user` zu registrieren.

Für Java führen Sie die folgenden Anweisungen aus:

```
CALL ml_server.ml_add_java_connection_script(  
    'custdb 16.0',  
    'authenticate_user',  
    'MobiLinkAuth.authenticateUser');
```

```
COMMIT;
```

Für .NET führen Sie die folgenden Anweisungen aus:

```
CALL ml_add_dnet_connection_script(  
    'custdb 16.0',  
    'authenticate_user',  
    'MobiLinkAuth.authenticateUser');
```

```
COMMIT;
```

Ergebnisse

Die `authenticateUser`-Methode wird für das Ereignis `authenticate_user` registriert.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 3: Starten des MobiLink-Servers“](#) auf Seite 170.

Siehe auch

- „Hinzufügungen und Löschungen von Skripten“ [[MobiLink - Serveradministration](#)]
- „Schreiben eines Synchronisationsskripts in Java“ [[MobiLink - Serveradministration](#)]
- „Synchronisationsskripten in .NET“ [[MobiLink - Serveradministration](#)]
- „Java-Synchronisationsbeispiel“ [[MobiLink - Serveradministration](#)]
- „.NET-Synchronisationsbeispiel“ [[MobiLink - Serveradministration](#)]
- „Debuggen von Java-Klassen“ [[MobiLink - Serveradministration](#)]
- „Fehlersuche in der .NET-Synchronisationslogik“ [[MobiLink - Serveradministration](#)]
- „Schreiben von Synchronisationsskripten“ [[MobiLink - Serveradministration](#)]
- „Synchronisationsereignisse“ [[MobiLink - Serveradministration](#)]
- „`ml_add_java_connection_script`-Systemprozedur“ [[MobiLink - Serveradministration](#)]
- „`ml_add_dnet_connection_script`-Systemprozedur“ [[MobiLink - Serveradministration](#)]

Lektion 3: Starten des MobiLink-Servers

In dieser Lektion führen Sie den MobiLink-Server mit der Option `-sl` aus, um Verzeichnisse für die Suche nach kompilierten Dateien festzulegen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Eine Java-Klasse für die benutzerdefinierte Authentifizierung \(serverseitig\) erstellen](#)“ auf Seite 164.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Der Einsatz von Java oder .NET für die benutzerdefinierte Authentifizierung](#)“ auf Seite 163.

Aufgabe

- Verbinden Sie sich mit der CustDB-Beispieldatenbank und laden Sie in der `mlsrv16`-Befehlszeile Ihre Java-Klasse oder das .NET-Assembly.

Ersetzen Sie `c:\MLauth` durch den Pfad Ihrer Quelldateien.

Für Java führen Sie den folgenden Befehl aus:

```
mlsrv16 -c "DSN=SQL Anywhere 16 CustDB" -o serverOut.txt -v+ -sl java(-cp  
c:\MLauth)
```

Für .NET führen Sie den folgenden Befehl aus:

```
mlsrv16 -c "DSN=SQL Anywhere 16 CustDB" -o serverOut.txt -v+ -sl dnet(-  
MLAutoLoadPath=c:\MLauth)
```

Ergebnisse

Die `MobiLinkAuth`-Methode wird ausgeführt, sobald das Synchronisationsereignis `authenticate_user` eintritt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 4: Testen der Authentifizierung](#)“ auf Seite 170.

Siehe auch

- „[MobiLink-Serveroptionen](#)“ [[MobiLink - Serveradministration](#)]
- „`mlsrv16`-Option `-sl java`“ [[MobiLink - Serveradministration](#)]
- „`mlsrv16`-Option `-sl dnet`“ [[MobiLink - Serveradministration](#)]

Lektion 4: Testen der Authentifizierung

UltraLite verfügt über einen Windows-Beispielclient, der eine Synchronisation initiieren kann. In dieser Lektion führen Sie die Anwendung mit der CustDB-Datenbank aus, die Sie in der vorangehenden Lektion gestartet haben.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Eine Java-Klasse für die benutzerdefinierte Authentifizierung \(serverseitig\) erstellen](#)“ auf Seite 164.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Der Einsatz von Java oder .NET für die benutzerdefinierte Authentifizierung](#)“ auf Seite 163.

Aufgabe

1. Starten Sie die Beispielanwendung.

Klicken Sie auf **Start » Programme » SQL Anywhere » UltraLite » Windows-Beispielanwendung**.

2. Geben Sie eine ungültige Mitarbeiter-ID ein und führen Sie eine Synchronisation durch.

In dieser Anwendung ist die Mitarbeiter-ID auch der MobiLink-Benutzername. Wenn der Benutzername nicht mit **128** beginnt, lässt die Logik die Synchronisation scheitern. Geben Sie den Wert **50** als Mitarbeiter-ID ein und klicken Sie auf **OK**.

Ergebnisse

Ein Fehler, in dem festgestellt wird, dass das Skript `authenticate_user` den Code 4000 zurückgibt, erscheint im Meldungsfenster des MobiLink-Servers.

Im Fenster der **UltraLite CustDB-Beispieldatenbank** wird der Synchronisationsfehler `SQLCODE -1497` gemeldet, der auf eine ungültige Benutzer-ID bzw. ein ungültiges Kennwort hinweist. Siehe „[Ungültige Benutzer-ID bzw. ungültiges Kennwort](#)“ [[Fehlermeldungen](#)].

Nächste Schritte

Gehen Sie weiter zu „[Aufräumen](#)“ auf Seite 171.

Siehe auch

- „[MobiLink-Beispiel CustDB](#)“ auf Seite 60

Aufräumen

Entfernen Sie das Material der praktischen Einführung von Ihrem Computer und setzen Sie die Datenbank für die Windows-Beispielanwendung zurück.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Eine Java-Klasse für die benutzerdefinierte Authentifizierung \(serverseitig\) erstellen](#)“ auf Seite 164.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Der Einsatz von Java oder .NET für die benutzerdefinierte Authentifizierung](#)“ auf Seite 163.

Aufgabe

1. Löschen Sie Ihre Java- oder .NET-Quelldateien.

Löschen Sie zum Beispiel das Verzeichnis `c:\mlauth`.

Vorsicht

Stellen Sie sicher, dass sich in diesem Verzeichnis nur Daten befinden, die sich auf die praktische Einführung beziehen.

2. Schließen Sie Interactive SQL und die UltraLite Windows-Clientanwendung.

Klicken Sie auf **Datei » Beenden** in jeder Anwendung.

3. Schließen Sie SQL Anywhere, MobiLink und den Synchronisationsclient.

Klicken Sie mit der rechten Maustaste auf die einzelnen Anzeigen in der Taskleiste und dann auf **Schließen**.

4. Setzen Sie die Datenbank für die Windows-Beispielanwendung zurück.

Führen Sie über das Verzeichnis `%SQLANYSDMP16%\UltraLite\CustDB` den folgenden Befehl aus:

```
makedbs
```

Ergebnisse

Alle Daten der praktischen Einführung werden von Ihrem Computer entfernt und die Datenbank für die Windows-Beispielanwendung wird zurückgesetzt.

Nächste Schritte

Keine.

Praktische Einführung: Verwenden der direkten Zeilenbehandlung

Die direkte Zeilenbehandlung ermöglicht es Ihnen, entfernte Daten an beliebige zentrale Datenquellen, Anwendungen oder Webdienste zu übermitteln, bei denen es sich um keine unterstützten konsolidierten Datenbanken handelt.

In dieser praktischen Einführung erfahren Sie, wie Sie die MobiLink-Server-APIs für Java und .NET zur einfachen direkten Zeilenbehandlung einsetzen können. Sie lernen auch, die RemoteOrders-Tabelle des

Clients mit der konsolidierten Datenbank zu synchronisieren und die direkte Zeilenbehandlung zur Verarbeitung von Zeilen in der OrderComments-Tabelle einzusetzen.

Erforderliche Software

- SQL Anywhere 16
- Java Software Development Kit oder Microsoft .NET Framework

Kenntnisse und Erfahrungen

Hierzu benötigen Sie Folgendes:

- Java- oder .NET-Kenntnisse
- Grundkenntnisse über MobiLink-Ereignisskripten

Privilegien

Sie müssen die folgenden Rollen und Privilegien für die konsolidierte Datenbank haben:

- SYS_AUTH_RESOURCE_ROLE-Kompatibilitätsrolle
- MONITOR-Systemprivileg

Sie müssen die folgenden Rollen und Privilegien für die entfernte Datenbank haben:

- SYS_REPLICATION_ADMIN_ROLE-Systemrolle
- SYS_RUN_REPLICATION_ROLE-Systemrolle

Übersicht

In dieser praktischen Einführung wird Folgendes veranschaulicht:

- Verwenden der MobiLink-Server-APIs für Java oder .NET
- Erstellen von Methoden zur direkten Zeilenbehandlung in MobiLink

Siehe auch

- „MobiLink-Synchronisation“ auf Seite 1
- „Synchronisationsmethoden“ [*MobiLink - Serveradministration*]
- „Direkte Zeilenbehandlung“ [*MobiLink - Serveradministration*]
- <http://www.sybase.com/detail?id=1058600#319> (Sie benötigen ein Login für Sybase.com, um diese Seite anzuzeigen.)
- <http://sqlanywhere-forum.sybase.com>

Lektion 1: Einrichten einer Textdatei-Datenquelle

In dieser Lektion erstellen Sie eine neue Textdatei, um Bestellinformationen zu speichern.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Verwenden der direkten Zeilenbehandlung](#)“ auf Seite 172.

Aufgabe

1. Erstellen Sie eine neue leere Textdatei.
2. Fügen Sie die folgenden durch Tabstops getrennten Werte für comment_id, order_id und order_comment in die Datei ein:

```
786    34    OK, ship promotional material.  
787    35    Yes, the product is going out of production.  
788    36    No, your commission cannot be increased...
```

3. Speichern Sie die Datei in Ihrem Arbeitsverzeichnis.

In dieser praktischen Einführung wird davon ausgegangen, dass *c:\MLdirect* das Arbeitsverzeichnis für serverseitige Komponenten ist. Speichern Sie die Datei als *orderResponses.txt* in diesem Verzeichnis.

Ergebnisse

Die Textdatei wird erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 2: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 174.

Lektion 2: Einrichten einer konsolidierten MobiLink-Datenbank

In dieser Lektion erstellen Sie eine Datenbank und definieren eine ODBC-Datenquelle.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Textdatei-Datenquelle](#)“ auf Seite 173.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Verwenden der direkten Zeilenbehandlung](#)“ auf Seite 172.

Kontext und Bemerkungen

Die konsolidierte MobiLink-Datenbank ist ein zentrales Daten-Repository mit MobiLink-Systemtabellen und gespeicherten Prozeduren, die für die Verwaltung des Synchronisationsprozesses eingesetzt werden.

Die direkte Zeilenbehandlung ermöglicht es Ihnen, Synchronisationen mit einer anderen Datenquelle als mit einer konsolidierten Datenbank durchzuführen. Auch in diesem Fall benötigen Sie jedoch eine konsolidierte Datenbank, um die Informationen zu verwalten, auf die der MobiLink-Server zugreift.

Hinweis

Wenn Sie bereits eine konsolidierte MobiLink-Datenbank mit MobiLink-Systemobjekten und einer ODBC-Datenquelle eingerichtet haben, können Sie diese Lektion überspringen.

Aufgabe

1. Klicken Sie auf **Start » Programme » SQL Anywhere 16 » Administrationstools » Sybase Central**.
2. Klicken Sie auf **Extras » SQL Anywhere 16 » Datenbank erstellen**.
3. Klicken Sie auf **Weiter**.
4. Akzeptieren Sie die Standardeinstellung **Datenbank auf diesem Computer erstellen** und klicken Sie auf **Weiter**.
5. Im Feld **Haupt-Datenbankdatei in folgender Datei speichern** geben Sie den Dateinamen und den Pfad für die Datenbank ein. Beispiel: `c:\MLdirect\MLconsolidated.db`.
6. Befolgen Sie die übrigen Anweisungen des **Assistenten zum Erstellen einer Datenbank** und akzeptieren Sie die Standardwerte. Deaktivieren Sie auf der Seite **Mit der Datenbank verbinden** die Option **Datenbank nach der letzten getrennten Verbindung herunterfahren**.
7. Klicken Sie auf **Fertig stellen**.

Die Datenbank MLconsolidated wird in Sybase Central angezeigt.

8. Klicken Sie auf **Schließen** im Fenster **Datenbank erstellen**, wenn das Fenster nicht automatisch geschlossen wurde.
9. Verwenden Sie den SQL Anywhere 16-Treiber, um eine ODBC-Datenquelle für die Datenbank MLconsolidated zu definieren.

Klicken Sie in Sybase Central auf **Extras » SQL Anywhere 16 » ODBC-Administrator öffnen**.

10. Klicken Sie auf der Registerkarte **Benutzer-DSN** auf **Hinzufügen**.
11. Klicken Sie im Fenster **Neue Datenquelle erstellen** auf **SQL Anywhere 16** und auf **Fertig stellen**.
12. Führen Sie im Fenster **ODBC-Konfiguration für SQL Anywhere** folgende Aufgaben aus:
 - a. Klicken Sie auf die Registerkarte **ODBC**.
 - b. Im Feld **Datenquellenname** geben Sie `mldirect_db` ein.
 - c. Klicken Sie auf die Registerkarte **Login**.
 - d. Im Feld **Benutzer-ID** geben Sie `DBA` ein.

- e. Im Feld **Kennwort** geben Sie **sql** ein.
- f. Im Feld **Servername** geben Sie **MLconsolidated** ein.
- g. Klicken Sie auf **OK**.

13. Schließen Sie den ODBC-Datenquellen-Administrator.

Klicken Sie auf **OK** im Fenster **ODBC-Datenquellen-Administrator**.

Ergebnisse

Die konsolidierte Datenbank und die ODBC-Datenquelle werden erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 3: Erstellen einer Tabelle in Ihrer konsolidierten MobiLink-Datenbank](#)“ auf Seite 176.

Siehe auch

- „Dienstprogramm Initialisierung (dbinit)“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „MobiLink - konsolidierte Datenbanken“ [[MobiLink - Serveradministration](#)]

Lektion 3: Erstellen einer Tabelle in Ihrer konsolidierten MobiLink-Datenbank

In der nachfolgenden Lektion erstellen Sie die Tabelle RemoteOrders in der konsolidierten MobiLink-Datenbank.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Textdatei-Datenquelle](#)“ auf Seite 173.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Verwenden der direkten Zeilenbehandlung](#)“ auf Seite 172.

Kontext und Bemerkungen

Die RemoteOrders-Tabelle wird erstellt und enthält folgende Spalten:

- order_id** Eindeutiger Bezeichner für Bestellungen
- product_id** Eindeutiger Bezeichner für Produkte
- quantity** Anzahl der verkauften Artikel
- order_status** Bestellstatus

last_modified Datum, an dem eine Zeile zuletzt geändert wurde. Diese Spalte verwenden Sie für zeitstempelbasierte Downloads, eine gängige Technik zur Filterung von Zeilen für eine effiziente Synchronisation.

Erstellen Sie die Tabelle RemoteOrders

1. Verbinden Sie sich mit Ihrer Datenbank mittels Interactive SQL.

Sie können Interactive SQL über Sybase Central oder eine Eingabeaufforderung starten.

- Um Interactive SQL aus Sybase Central zu starten, rechtsklicken Sie auf die **MLconsolidated - DBA**-Datenbank und auf **Interactive SQL öffnen**.
- Um Interactive SQL an einer Eingabeaufforderung zu starten, führen Sie folgenden Befehl aus:

```
dbisql -c "DSN=mldirect_db"
```

2. Um die Tabelle RemoteOrders zu erstellen, führen Sie folgende SQL-Anweisung in Interactive SQL aus.

```
CREATE TABLE RemoteOrders (  
    order_id          INTEGER NOT NULL,  
    product_id        INTEGER NOT NULL,  
    quantity          INTEGER,  
    order_status       VARCHAR(10) DEFAULT 'new',  
    last_modified      TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY(order_id)  
);
```

Interactive SQL erstellt die Tabelle RemoteOrders in Ihrer konsolidierten Datenbank.

3. Führen Sie die folgende Anweisung in Interactive SQL aus, um MobiLink-Systemtabellen und gespeicherte Prozeduren zu erstellen.

Ersetzen Sie *C:\Program Files\SQL Anywhere 16* durch den Pfad Ihrer SQL Anywhere 16-Installation.

```
READ "C:\Program Files\SQL Anywhere 16\MobiLink\setup\syncsa.sql";
```

Interactive SQL wendet *syncsa.sql* auf Ihre konsolidierte Datenbank an. Durch die Ausführung von *syncsa.sql* wird eine Reihe von Systemtabellen und gespeicherten Prozeduren mit dem Präfix **ml__** erstellt. Der MobiLink-Server arbeitet während des Synchronisationsprozesses mit diesen Tabellen und gespeicherten Prozeduren.

Ergebnisse

Die RemoteOrders-Tabelle wird erstellt und MobiLink-Systemtabellen und gespeicherte Prozeduren werden zur konsolidierten Datenbank hinzugefügt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 4: Hinzufügen von Synchronisationsskripten](#)“ auf Seite 178.

Siehe auch

- „CREATE TABLE-Anweisung“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]

Lektion 4: Hinzufügen von Synchronisationsskripten

Sie verwenden die direkte Zeilenbehandlung, um mit gespeicherten Prozeduren Ihrer konsolidierten MobiLink-Datenbank Synchronisationsskriptinformationen hinzuzufügen. In dieser Lektion registrieren Sie Methodennamen, die den Ereignissen `handle_UploadData`, `handle_DownloadData`, `end_download`, `download_cursor` und `download_delete_cursor` entsprechen. Sie erstellen Ihre eigene Java- oder .NET-Klasse in einer späteren Lektion.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Textdatei-Datenquelle](#)“ auf Seite 173.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Verwenden der direkten Zeilenbehandlung](#)“ auf Seite 172.

Kontext und Bemerkungen

Mit der SQL-Zeilenbehandlung können Sie entfernte Daten mit Tabellen in Ihrer konsolidierten MobiLink-Datenbank synchronisieren. SQL-basierte Skripten legen Folgendes fest:

- Wie Upload-Daten von einem MobiLink-Client in die konsolidierte Datenbank übernommen werden
- Welche Daten von der konsolidierten Datenbank heruntergeladen werden

Die folgenden SQL-basierten Upload- und Download-Ereignisse werden erstellt:

- **upload_insert** Dieses Ereignis definiert, wie neue Bestellungen aus einer entfernten Clientdatenbank in die konsolidierte Datenbank übernommen werden.
- **download_cursor** Dieses Ereignis definiert die Bestellungen, die auf entfernte Clients heruntergeladen werden sollen.
- **download_delete_cursor** Dieses Ereignis ist erforderlich, wenn Sie Synchronisationsskripten verwenden, die nicht nur einen reinen Upload ausführen. Konfigurieren Sie den MobiLink-Server so, dass er dieses Ereignis für die vorliegende praktische Einführung ignoriert.

Aufgabe

1. Verbinden Sie sich über Interactive SQL mit Ihrer konsolidierten Datenbank, falls dies nicht schon geschehen ist, indem Sie folgenden Befehl ausführen:

```
dbisql -c "DSN=mldirect_db"
```

2. Verwenden Sie die gespeicherte Prozedur `ml_add_table_script`, um SQL-basierte Tabellenskripten für die Ereignisse `upload_insert`, `download_cursor` und `download_delete_cursor` hinzuzufügen.

Führen Sie hierzu die folgende SQL-Anweisung in Interactive SQL aus. Das upload_insert-Skript fügt order_id, product_id, quantity und order_status in die konsolidierte MobiLink-Datenbank ein. Das download_cursor-Skript verwendet eine Filterung auf Zeitstempelbasis für den Download aktualisierter Zeilen auf entfernte Clients.

```
CALL ml_add_table_script( 'default', 'RemoteOrders',
    'upload_insert',
    'INSERT INTO RemoteOrders( order_id, product_id, quantity,
    order_status)
    VALUES( {ml r.order_id}, {ml r.product_id}, {ml r.quantity}, {ml
    r.order_status} )' );

CALL ml_add_table_script( 'default', 'RemoteOrders',
    'download_cursor',
    'SELECT order_id, product_id, quantity, order_status
    FROM RemoteOrders WHERE last_modified >= {ml
    s.last_table_download}');

CALL ml_add_table_script( 'default', 'RemoteOrders',
    'download_delete_cursor', '--{ml_ignore}');

COMMIT;
```

3. Registrieren Sie eine Java- oder .NET-Methode für das end_download-Ereignis.

Mit dieser Methode geben Sie Speicherressourcen frei, wenn der MobiLink-Server das end_download-Verbindungsereignis ausführt.

Für Java führen Sie die folgende Anweisung in Interactive SQL aus:

```
CALL ml_add_java_connection_script( 'default',
    'end_synchronization',
    'MobiLinkOrders.EndSync' );
```

Für .NET führen Sie die folgende Anweisung in Interactive SQL aus:

```
CALL ml_add_dnet_connection_script( 'default',
    'end_synchronization',
    'MobiLinkOrders.EndSync' );
```

Interactive SQL registriert die benutzerdefinierte EndDownload-Methode für das end_download-Ereignis.

4. Registrieren Sie Java- oder .NET-Methoden für die Ereignisse handle_UploadData und handle_DownloadData.

Für Java führen Sie die folgenden Anweisungen in Interactive SQL aus:

```
CALL ml_add_java_connection_script( 'default',
    'handle_UploadData',
    'MobiLinkOrders.GetUpload' );

CALL ml_add_java_connection_script( 'default',
    'handle_DownloadData',
    'MobiLinkOrders.SetDownload' );
```

Für .NET führen Sie die folgenden Anweisungen in Interactive SQL aus:

```
CALL ml_add_dnet_connection_script( 'default',  
  'handle_UploadData',  
  'MobiLinkOrders.GetUpload' );  
  
CALL ml_add_dnet_connection_script( 'default',  
  'handle_DownloadData',  
  'MobiLinkOrders.SetDownload' );
```

Interactive SQL registriert die benutzerdefinierten GetUpload- und SetDownload-Methoden jeweils für die Ereignisse handle_UploadData und handle_DownloadData. Sie erstellen diese Methoden in einer folgenden Lektion.

5. Registrieren Sie download_cursor- und download_delete_cursor-Ereignisse.

Führen Sie hierzu die folgende SQL-Anweisung in Interactive aus:

```
CALL ml_add_table_script( 'default', 'OrderComments',  
  'download_cursor', '--{ml_ignore}');  
  
CALL ml_add_table_script( 'default', 'OrderComments',  
  'download_delete_cursor', '--{ml_ignore}');
```

Die Ereignisse download_cursor und download_delete_cursor müssen für die OrderComments-Tabelle registriert werden, wenn Sie Skripten verwenden, weil die Synchronisation in beiden Richtungen erfolgt und nicht nur ein reiner Upload ist. Siehe „Erforderliche Skripten“ [[MobiLink - Serveradministration](#)].

6. Schreiben Sie Ihre Änderungen fest.

Führen Sie die folgende Anweisung in Interactive SQL aus:

```
COMMIT;
```

7. Schließen Sie Interactive SQL.

Ergebnisse

Die Methodennamen zu den Ereignissen handle_UploadData, handle_DownloadData, end_download, download_cursor und download_delete_cursor werden registriert.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 5: Erstellen von Java- oder .NET-Klassen für die direkte Zeilenbehandlung in MobiLink](#)“ auf Seite 181.

Siehe auch

- „Überblick über MobiLink-Ereignisse“ [[MobiLink - Serveradministration](#)]
- „Hinzufügungen und Löschungen von Skripten“ [[MobiLink - Serveradministration](#)]
- „Skripten für den Upload von Zeilen“ [[MobiLink - Serveradministration](#)]
- „Skripten für den Download von Zeilen“ [[MobiLink - Serveradministration](#)]
- „upload_insert (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „upload_update (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „upload_delete (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „download_cursor (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „download_delete_cursor (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „Direkte Zeilenbehandlung“ [[MobiLink - Serveradministration](#)]
- „Direkte Uploads“ [[MobiLink - Serveradministration](#)]
- „Direkte Downloads“ [[MobiLink - Serveradministration](#)]
- „Implementieren zeitstempelbasierter Downloads“ [[MobiLink - Serveradministration](#)]
- „Partitionierte Zeilen in entfernten Datenbanken“ [[MobiLink - Serveradministration](#)]

Lektion 5: Erstellen von Java- oder .NET-Klassen für die direkte Zeilenbehandlung in MobiLink

In dieser Lektion verwenden Sie die direkte Zeilenbehandlung für die Verarbeitung von Zeilen in der OrderComments-Tabelle Ihrer Clientdatenbank.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Textdatei-Datenquelle](#)“ auf Seite 173.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Verwenden der direkten Zeilenbehandlung](#)“ auf Seite 172.

Kontext und Bemerkungen

Sie fügen die folgende Methode für die direkte Zeilenbehandlung hinzu:

- **GetUpload** Verwenden Sie diese Methode für das handle_UploadData-Ereignis. GetUpload schreibt Upload-Kommentare in die Datei *orderComments.txt*.
- **SetDownload** Verwenden Sie diese Methode für das handle_DownloadData-Ereignis. SetDownload verwendet die *orderResponses.txt*-Datei zum Download von Antworten auf entfernte Clients.
- **EndDownload** Verwenden Sie diese Methode für das end_download-Ereignis. EndDownload gibt Speicherressourcen frei.

In der folgenden Prozedur wird gezeigt, wie Sie eine Java- oder .NET-Klasse erstellen, die Ihre Verarbeitungsmethoden enthält. Eine vollständige Auflistung finden Sie unter „[Vollständige](#)“

[MobiLinkOrders-Auflistung \(Java\)](#)“ auf Seite 188 oder „[Vollständige MobiLinkOrders-Auflistung \(.NET\)](#)“ auf Seite 190.

Aufgabe

1. Erstellen Sie eine Klasse mit dem Namen MobiLinkOrders in Java oder .NET.

Für Java verwenden Sie den folgenden Code:

```
import ianywhere.ml.script.*;
import java.io.*;
import java.sql.*;

public class MobiLinkOrders {
```

Für .NET verwenden Sie den folgenden Code:

```
using iAnywhere.MobiLink.Script;
using System.IO;
using System.Data;
using System.Text;

public class MobiLinkOrders {
```

2. Deklarieren Sie eine DBConnectionContext-Instanzvariable auf Klassenebene.

Für Java verwenden Sie den folgenden Code:

```
// Class level DBConnectionContext
DBConnectionContext _cc;
```

Für .NET verwenden Sie den folgenden Code:

```
// Class level DBConnectionContext
private DBConnectionContext _cc = null;
```

Der MobiLink-Server übergibt eine DBConnectionContext-Instanz an Ihren Klassenkonstruktor. DBConnectionContext umfasst Informationen zur aktuellen Verbindung mit der konsolidierten MobiLink-Datenbank.

3. Deklarieren Sie Objekte für die Dateieingabe und Dateiausgabe.

Für Java deklarieren Sie einen java.io.FileWriter und java.io.BufferedReader wie folgt:

```
// Java objects for file i/o
FileWriter my_writer;
BufferedReader my_reader;
```

Für .NET deklarieren Sie einen StreamWriter und StreamReader wie folgt:

```
// Instances for file I/O
private static StreamWriter my_writer = null;
private static StreamReader my_reader = null;
```

4. Erstellen Sie Ihren Klassenkonstruktor.

Ihr Klassenkonstruktor legt Ihre DBConnectionContext-Instanz auf Klassenebene fest.

Für Java verwenden Sie den folgenden Code:

```
public MobiLinkOrders( DBConnectionContext cc )
    throws IOException, FileNotFoundException
{
    // Declare a class-level DBConnectionContext
    _cc = cc;
}
```

Für .NET verwenden Sie den folgenden Code:

```
public MobiLinkOrders(DBConnectionContext cc) {
    _cc = cc;
}
```

5. Schreiben Sie die GetUpload-Methode

Die GetUpload-Methode ruft eine UploadedTableData-Klasseninstanz ab, die die OrderComments-Tabelle darstellt. Die OrderComments-Tabelle enthält spezielle Kommentare von entfernten Vertriebsmitarbeitern. Sie erstellen diese Tabelle in einer späteren Lektion.

Die UploadedTableData getInserts-Methode gibt eine Ergebnismenge für neue Bestellkommentare zurück. Die writeOrderComment-Methode gibt jede Zeile in der Ergebnismenge in einer Textdatei aus.

Für Java verwenden Sie den folgenden Code:

```
public void writeOrderComment( int _commentID, int _orderID, String
    _comments )
    throws IOException
{
    if (my_writer == null)
        // A FileWriter for writing order comments
        my_writer = new FileWriter( "C:\\MLdirect\\
\\orderComments.txt",true);

    // Write out the order comments to remoteOrderComments.txt
    my_writer.write(_commentID + "\\t" + _orderID + "\\t" + _comments);
    my_writer.write( "\\n" );
    my_writer.flush();
}

// Method for the handle_UploadData synchronization event
public void GetUpload( UploadData ut )
    throws SQLException, IOException
{
    // Get an UploadedTableData for OrderComments
    UploadedTableData orderCommentsTbl =
    ut.getUploadedTableByName("OrderComments");

    // Get inserts uploaded by the MobiLink client
    ResultSet insertResultSet = orderCommentsTbl.getInserts();

    while ( insertResultSet.next() )
    {
        // Get order comments
        int _commentID = insertResultSet.getInt("comment_id");
        int _orderID = insertResultSet.getInt("order_id");
        String _specialComments =
        insertResultSet.getString("order_comment");
    }
}
```

```
        if (_specialComments != null) {
            writeOrderComment(_commentID, _orderID, _specialComments);
        }
    }
    insertResultSet.close();
}
```

Für .NET verwenden Sie den folgenden Code:

```
public void WriteOrderComment(int comment_id,
    int order_id,
    string comments)
{
    if (my_writer == null) {
        my_writer = new StreamWriter("c:\\MLdirect\\
\\orderComments.txt");
    }
    my_writer.WriteLine("{0}\\t{1}\\t{2}", comment_id, order_id,
comments);
    my_writer.Flush();
}

// Method for the handle_UploadData synchronization event.
public void GetUpload(UploadData ut)
{
    // Get UploadedTableData for remote table called OrderComments
    UploadedTableData order_comments_table_data =
        ut.GetUploadedTableByName("OrderComments");

    // Get inserts uploaded by the MobiLink client
    using( IDataReader new_comment_reader =
order_comments_table_data.GetInserts() ) {
        while (new_comment_reader.Read()) {
            // Columns are
            // 0 - "order_comment"
            // 1 - "comment_id"
            // 2 - "order_id"
            // You can look up these values using the
            DataTable returned by:
            //
            order_comments_table_data.GetSchemaTable().
            // In this example, you just use the
            known column order to
            // determine the column indexes;
            alternatively, you could use
            // the column names

            // Only process this insert if the
            order_comment is not null
            if (!new_comment_reader.IsDBNull(2)) {
                int comment_id =
                    new_comment_reader.GetInt32(0);
                int order_id =
                    new_comment_reader.GetInt32(1);
                string comments =
                    new_comment_reader.GetString(2);
                WriteOrderComment(comment_id,
order_id, comments);
            }
        }
    }
}
```

6. Erstellen Sie die SetDownload-Methode:

- a. Rufen Sie eine Klasseninstanz ab, die die OrderComments-Tabelle darstellt.

Verwenden Sie die DBConnectionContext getDownloadData-Methode, um eine DownloadData-Instanz zu erhalten. Verwenden Sie die DownloadData getDownloadTableByName-Methode zur Rückgabe einer DownloadTableData-Instanz für die OrderComments-Tabelle.

Für Java verwenden Sie den folgenden Code:

```
public void SetDownload()
    throws SQLException, IOException
{
    DownloadData download_d = _cc.getDownloadData();

    DownloadTableData download_td =
download_d.getDownloadTableByName( "OrderComments" );
```

Für .NET verwenden Sie den folgenden Code:

```
private const string read_file_path =
    "c:\\MLdirect\\orderResponses.txt";

// Method for the handle_DownloadData synchronization event
public void SetDownload() {
    if ((my_reader == null) && !File.Exists(read_file_path)) {
        System.Console.Out.Write("There is no file to read.");
        return;
    }
    DownloadTableData comments_for_download =

    _cc.GetDownloadData().GetDownloadTableByName( "OrderComments" );
```

Hinweis

Sie erstellen die Tabelle OrderComments in der entfernten Datenbank in „[Lektion 7: Einrichten der konsolidierten MobiLink-Clientdatenbank](#)“ auf Seite 194.

- b. Rufen Sie eine vorbereitete Anweisung auf, die Ihnen das Hinzufügen oder Einfügen von Vorgängen zum bzw. in den Download sowie das Aktualisieren dieser Vorgänge ermöglicht.

Für Java verwenden Sie die DownloadTableData getUpsertPreparedStatement-Methode, um eine java.sql.PreparedStatement-Instanz zurückzugeben, wie folgt:

```
PreparedStatement update_ps =
download_td.getUpsertPreparedStatement();
```

Für .NET verwenden Sie die DownloadTableData GetUpsertCommand-Methode wie folgt:

```
// Add upserts to the set of operation that are going to be
// applied at the remote database
IDbCommand comments_upsert =
    comments_for_download.GetUpsertCommand();
```

- c. Legen Sie die Downloaddaten für jede Zeile fest.

Dieser Code durchläuft die *orderResponses.txt*-Datei und fügt Daten zum MobiLink-Download hinzu.

Für Java verwenden Sie den folgenden Code:

```
try {
    // A BufferedReader for reading in responses
```

```
        if (my_reader == null)
            my_reader = new BufferedReader(new FileReader("C:\\MLdirect\\
\\orderResponses.txt"));

        // Get the next line from orderResponses
        String commentLine;
        commentLine = my_reader.readLine();

        // Send comment responses down to clients
        while (commentLine != null) {
            // Get the next line from orderResponses.txt
            String[] response_details = commentLine.split("\\t");

            if (response_details.length != 3) {
                System.err.println("Error reading from
orderResponses.txt");
                System.err.println("Error setting direct row handling
download");
                return;
            }
            int comment_id = Integer.parseInt(response_details[0]);
            int order_id = Integer.parseInt(response_details[1]);
            String updated_comment = response_details[2];

            // Set an order comment response in the MobiLink download
            update_ps.setInt(1, comment_id);
            update_ps.setInt(2, order_id);
            update_ps.setString(3, updated_comment);
            update_ps.executeUpdate();

            // Get next line
            commentLine = my_reader.readLine();
        }
    }
```

Für .NET verwenden Sie den folgenden Code:

```
if (my_reader == null) {
    my_reader = new StreamReader(read_file_path);
}
string comment_line;
while ((comment_line = my_reader.ReadLine()) != null) {
    // Three values are on each line separated by '\t'
    string[] response_details = comment_line.Split('\t');
    if (response_details.Length != 3) {
        throw (new SynchronizationException(
            "Error reading from orderResponses.txt"));
    }
    int comment_id = System.Int32.Parse(response_details[0]);
    int order_id = System.Int32.Parse(response_details[1]);
    string comments = response_details[2];

    // Parameters of the correct number and type have
    // already been added so you just need to set the
    // values of the IDataParameter
    ((IDataParameter)(comments_upsert.Parameters[0])).Value =
        comment_id;
    ((IDataParameter)(comments_upsert.Parameters[1])).Value =
        order_id;
    ((IDataParameter)(comments_upsert.Parameters[2])).Value =
        comments;
    // Add the upsert operation
    comments_upsert.ExecuteNonQuery();
}
```



```
    }
}
```

- d. Schließen Sie die vorbereitete Anweisung für das Hinzufügen von Einfüge- oder Aktualisierungsvorgängen zum Download.

Für Java verwenden Sie den folgenden Code:

```
finally {
    update_ps.close();
}
```

Für .NET müssen Sie IDbCommand nicht schließen. Das Objekt wird am Ende des Downloads automatisch zerstört.

7. Erstellen Sie die EndDownload-Methode.

Diese Methode verarbeitet das end_download-Verbindungsereignis und bietet Ihnen die Möglichkeit, Ressourcen freizugeben.

Für Java verwenden Sie den folgenden Code:

```
public void EndSync()
    throws IOException
{
    // Close i/o resources
    if (my_reader != null) {
        my_reader.close();
        my_reader = null;
    }
    if (my_writer != null) {
        my_writer.close();
        my_writer = null;
    }
}
```

Für .NET verwenden Sie den folgenden Code:

```
public void EndSync()
{
    if (my_writer != null) {
        my_writer.Close();
        my_writer = null;
    }
    if (my_reader != null) {
        my_reader.Close();
        my_reader = null;
    }
}
```

8. Speichern Sie den Programmcode.

Für Java speichern Sie Ihren Code als *MobiLinkOrders.java* in Ihrem Arbeitsverzeichnis *c:\MLdirect*.

Für .NET speichern Sie Ihren Code als *MobiLinkOrders.cs* in Ihrem Arbeitsverzeichnis *c:\MLdirect*.

9. Hinweise zur Überprüfung des Codes finden Sie unter „[Vollständige MobiLinkOrders-Auflistung \(Java\)](#)“ auf Seite 188 oder „[Vollständige MobiLinkOrders-Auflistung \(.NET\)](#)“ auf Seite 190.

10. Kompilieren Sie Ihre Klassendatei.

- Navigieren Sie zum Verzeichnis mit Ihren Java- oder .NET-Quelldateien.
- Kompilieren Sie MobiLinkOrders mit Referenzen zur API-Bibliothek des MobiLink-Servers für Java oder .NET.

Für Java müssen Sie *mlscript.jar* unter *%SQLANY16%\java* referenzieren.

Für Java führen Sie den folgenden Befehl aus und ersetzen *C:\Program Files\SQL Anywhere 16* durch Ihr SQL Anywhere 16-Verzeichnis:

```
javac -classpath "C:\Program Files\SQL Anywhere 16\java\mlscript.jar"  
MobiLinkOrders.java
```

Für .NET führen Sie den folgenden Befehl aus und ersetzen *C:\Program Files\SQL Anywhere 16* durch Ihr SQL Anywhere 16-Verzeichnis:

```
csc /out:MobiLinkServerCode.dll /target:library /reference:"C:  
\Program Files\SQL Anywhere 16\Assembly  
\v2\iAnywhere.MobiLink.Script.dll" MobiLinkOrders.cs
```

Hinweis

Dieses Beispiel gewährleistet nicht, dass die Primärschlüsselwerte eindeutig sind. Siehe „Eindeutige Primärschlüssel“ [[MobiLink - Serveradministration](#)].

Ergebnisse

Die Java- oder .NET-Klasse für die direkte Zeilenbehandlung in MobiLink wird erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 6: Starten des MobiLink-Servers](#)“ auf Seite 192.

Siehe auch

- „Direkte Zeilenbehandlung“ [[MobiLink - Serveradministration](#)]
- „Schreiben eines Synchronisationsskripts in Java“ [[MobiLink - Serveradministration](#)]
- „Synchronisationsskripten in .NET“ [[MobiLink - Serveradministration](#)]

Vollständige MobiLinkOrders-Auflistung (Java)

Nachfolgend finden Sie die vollständige MobiLinkOrders-Liste für die direkte Zeilenbehandlung mit Java. Eine schrittweise Erklärung finden Sie unter „[Lektion 5: Erstellen von Java- oder .NET-Klassen für die direkte Zeilenbehandlung in MobiLink](#)“ auf Seite 181.

```
import ianywhere.ml.script.*;  
import java.io.*;  
import java.sql.*;  
  
public class MobiLinkOrders {  
  
    // Class level DBConnectionContext  
    DBConnectionContext _cc;
```

```

// Java objects for file i/o
FileWriter my_writer;
BufferedReader my_reader;

public MobiLinkOrders( DBConnectionContext cc )
    throws IOException, FileNotFoundException
{
    // Declare a class-level DBConnectionContext
    _cc = cc;
}

public void writeOrderComment( int _commentID, int _orderID, String
_comments )
    throws IOException
{
    if (my_writer == null)
        // A FileWriter for writing order comments
        my_writer = new FileWriter( "C:\\\\MLdirect\\
\\orderResponses.txt",true);

    // Write out the order comments to remoteOrderComments.txt
    my_writer.write(_commentID + "\\t" + _orderID + "\\t" + _comments);
    my_writer.write( "\\n" );
    my_writer.flush();
}

// Method for the handle_UploadData synchronization event
public void GetUpload( UploadData ut )
    throws SQLException, IOException
{
    // Get an UploadedTableData for OrderComments
    UploadedTableData orderCommentsTbl =
ut.getUploadedTableByName("OrderComments");

    // Get inserts uploaded by the MobiLink client
    ResultSet insertResultSet = orderCommentsTbl.getInserts();

    while ( insertResultSet.next() )
    {
        // Get order comments
        int _commentID = insertResultSet.getInt("comment_id");
        int _orderID = insertResultSet.getInt("order_id");
        String _specialComments =
insertResultSet.getString("order_comment");
        if (_specialComments != null) {
            writeOrderComment(_commentID,_orderID,_specialComments);
        }
    }
    insertResultSet.close();
}

public void SetDownload()
    throws SQLException, IOException
{
    DownloadData download_d = _cc.getDownloadData();

    DownloadTableData download_td =
download_d.getDownloadTableByName( "OrderComments" );

    PreparedStatement update_ps =
download_td.getUpsertPreparedStatement();
    try {
        // A BufferedReader for reading in responses
        if (my_reader == null)

```

```
        my_reader = new BufferedReader(new FileReader("C:\\MLdirect\\
\\orderResponses.txt"));

        // Get the next line from orderResponses
        String commentLine;
        commentLine = my_reader.readLine();

        // Send comment responses down to clients
        while (commentLine != null) {
            // Get the next line from orderResponses.txt
            String[] response_details = commentLine.split("\\t");

            if (response_details.length != 3) {
                System.err.println("Error reading from orderResponses.txt");
                System.err.println("Error setting direct row handling download");
                return;
            }
            int comment_id = Integer.parseInt(response_details[0]);
            int order_id = Integer.parseInt(response_details[1]);
            String updated_comment = response_details[2];

            // Set an order comment response in the MobiLink download
            update_ps.setInt(1, comment_id);
            update_ps.setInt(2, order_id);
            update_ps.setString(3, updated_comment);
            update_ps.executeUpdate();

            // Get next line
            commentLine = my_reader.readLine();
        }
    } finally {
        update_ps.close();
    }
}

public void EndDownload()
    throws IOException
{
    // Close i/o resources
    if (my_reader != null) {
        my_reader.close();
        my_reader = null;
    }
    if (my_writer != null) {
        my_writer.close();
        my_writer = null;
    }
}
}
```

Vollständige MobiLinkOrders-Auflistung (.NET)

Nachfolgend finden Sie die vollständige MobiLinkOrders-Liste für die direkte Zeilenbehandlung mit .NET. Eine schrittweise Erklärung finden Sie unter „[Lektion 5: Erstellen von Java- oder .NET-Klassen für die direkte Zeilenbehandlung in MobiLink](#)“ auf Seite 181.

```
using iAnywhere.MobiLink.Script;
using System.IO;
using System.Data;
using System.Text;
```

```

public class MobiLinkOrders {
    // Class level DBConnectionContext
    private DBConnectionContext _cc = null;

    // Instances for file I/O
    private static StreamWriter my_writer = null;
    private static StreamReader my_reader = null;

    public MobiLinkOrders(DBConnectionContext cc) {
        _cc = cc;
    }

    public void WriteOrderComment(int comment_id,
        int order_id,
        string comments)
    {
        if (my_writer == null) {
            my_writer = new StreamWriter("c:\\MLdirect\\orderComments.txt");
        }
        my_writer.WriteLine("{0}\\t{1}\\t{2}", comment_id, order_id, comments);
        my_writer.Flush();
    }

    // Method for the handle_UploadData synchronization event.
    public void GetUpload(UploadData ut)
    {
        // Get UploadedTableData for remote table called OrderComments
        UploadedTableData order_comments_table_data =
            ut.GetUploadedTableByName("OrderComments");

        // Get inserts uploaded by the MobiLink client
        IDataReader new_comment_reader =
            order_comments_table_data.GetInserts();

        while (new_comment_reader.Read()) {
            // Columns are
            // 0 - "order_comment"
            // 1 - "comment_id"
            // 2 - "order_id"
            // You can look up these values using the DataTable returned by:
            // order_comments_table_data.GetSchemaTable().
            // In this example, you just use the known column order to
            // determine the column indexes

            // Only process this insert if the order_comment is not null
            if (!new_comment_reader.IsDBNull(2)) {
                int comment_id = new_comment_reader.GetInt32(0);
                int order_id = new_comment_reader.GetInt32(1);
                string comments = new_comment_reader.GetString(2);
                WriteOrderComment(comment_id, order_id, comments);
            }
        }
        // Always close the reader when you are done with it!
        new_comment_reader.Close();
    }

    private const string read_file_path =
        "c:\\MLdirect\\orderResponses.txt";

    // Method for the handle_DownloadData synchronization event
    public void SetDownload() {
        if ((my_reader == null) && !File.Exists(read_file_path)) {
            System.Console.Out.Write("There is no file to read.");
        }
    }
}

```

```

        return;
    }
    DownloadTableData comments_for_download =
        _cc.GetDownloadData().GetDownloadTableByName("OrderComments");

    // Add upserts to the set of operation that are going to be
    // applied at the remote database
    IDbCommand comments_upsert =
        comments_for_download.GetUpsertCommand();

    if (my_reader == null) {
        my_reader = new StreamReader(read_file_path);
    }
    string comment_line;
    while ((comment_line = my_reader.ReadLine()) != null) {
        // Three values are on each line separated by '\t'
        string[] response_details = comment_line.Split('\t');
        if (response_details.Length != 3) {
            throw (new SynchronizationException(
                "Error reading from orderResponses.txt"));
        }
        int comment_id = System.Int32.Parse(response_details[0]);
        int order_id = System.Int32.Parse(response_details[1]);
        string comments = response_details[2];

        // Parameters of the correct number and type have
        // already been added so you just need to set the
        // values of the IDataParameter
        ((IDataParameter)(comments_upsert.Parameters[0])).Value =
            comment_id;
        ((IDataParameter)(comments_upsert.Parameters[1])).Value =
            order_id;
        ((IDataParameter)(comments_upsert.Parameters[2])).Value =
            comments;
        // Add the upsert operation
        comments_upsert.ExecuteNonQuery();
    }
}

public void EndDownload()
{
    if (my_writer != null) {
        my_writer.Close();
        my_writer = null;
    }
    if (my_reader != null) {
        my_reader.Close();
        my_reader = null;
    }
}
}

```

Lektion 6: Starten des MobiLink-Servers

In dieser Lektion starten Sie den MobiLink-Server. Starten Sie den MobiLink-Server (mlsrv16) mit der Option -c, um eine Verbindung mit der konsolidierten Datenbank herzustellen. Verwenden Sie die Option -sl java oder -sl dnet, um Ihre Java- oder .NET-Klasse zu laden.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Textdatei-Datenquelle](#)“ auf Seite 173.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Verwenden der direkten Zeilenbehandlung](#)“ auf Seite 172.

Aufgabe

- Verbinden Sie sich mit Ihrer konsolidierten Datenbank und laden Sie die Klasse auf der mlsrv16-Befehlszeile.

Ersetzen Sie `c:\MLdirect` durch den Pfad Ihrer Quelldateien.

Für Java führen Sie den folgenden Befehl aus:

```
mlsrv16 -c "DSN=mldirect_db" -o serverOut.txt -v+ -dl -zu+ -x tcpip -sl
java (-cp c:\MLdirect)
```

Für .NET führen Sie den folgenden Befehl aus:

```
mlsrv16 -c "DSN=mldirect_db" -o serverOut.txt -v+ -dl -zu+ -x tcpip -sl
dnet (-MLAutoLoadPath=c:\MLdirect)
```

Das MobiLink-Server-Meldungsfenster wird eingeblendet.

Nachfolgend finden Sie eine Beschreibung der einzelnen MobiLink-Serveroptionen, die in dieser praktischen Einführung verwendet werden. Die Optionen -o, -v und -dl bieten Informationen zur Fehlersuche und -behebung. Die Verwendung dieser Protokollierungsoptionen empfiehlt sich in einer Entwicklungsumgebung. Aus Gründen der Performance werden -v+ und -dl in der Produktion normalerweise nicht verwendet.

Option	Beschreibung
-c	Steht vor der Verbindungszeichenfolge
-o	Legt die Meldungslogdatei <i>serverOut.txt</i> fest
-v+	Die Option -v legt fest, welche Informationen protokolliert werden. Mit -v+ wird die maximale ausführliche Protokollierung aktiviert.
-dl	Alle Lognachrichten auf dem Bildschirm anzeigen.
-zu+	Fügt neue Benutzer automatisch hinzu.
-x	Legt das Kommunikationsprotokoll und Parameter für MobiLink-Clients fest
-sl java	Bestimmt eine Verzeichnisgruppe für die Suche nach Klassendateien und erzwingt das Laden der Java Virtual Machine beim Start des Servers

Option	Beschreibung
-sl dnet	Legt den Speicherort der .NET-Assemblys fest und erzwingt das Laden der externen Umgebung CLR beim Serverstart.

Ergebnisse

Der MobiLink-Server wird mit einer Verbindung zu Ihrer konsolidierten Datenbank gestartet und die von Ihnen erstellte Klasse wird geladen.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 7: Einrichten der konsolidierten MobiLink-Clientdatenbank](#)“ auf Seite 194.

Siehe auch

- „MobiLink-Serveroptionen“ [[MobiLink - Serveradministration](#)]
- „mlsrv16-Option -sl java“ [[MobiLink - Serveradministration](#)]
- „mlsrv16-Option -sl dnet“ [[MobiLink - Serveradministration](#)]

Lektion 7: Einrichten der konsolidierten MobiLink-Clientdatenbank

In dieser Lektion verwenden Sie eine SQL Anywhere-Datenbank für Ihre konsolidierte Datenbank und den MobiLink-Client. Dabei wird davon ausgegangen, dass sich Ihr MobiLink-Client, die konsolidierte Datenbank und der MobiLink-Server auf demselben Computer befinden.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Textdatei-Datenquelle](#)“ auf Seite 173.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Verwenden der direkten Zeilenbehandlung](#)“ auf Seite 172.

Kontext und Bemerkungen

Um die MobiLink-Clientdatenbank einzurichten, erstellen Sie die Tabellen RemoteOrders und OrderComments. Die RemoteOrders-Tabelle entspricht der RemoteOrders-Tabelle in der konsolidierten Datenbank. Der MobiLink-Server verwendet SQL-basierte Skripten für die Synchronisation der entfernten Bestellungen. Die OrderComments-Tabelle wird nur in Clientdatenbanken verwendet. Der MobiLink-Server verarbeitet die OrderComments-Tabelle mithilfe spezieller Ereignisse.

Erstellen Sie anschließend einen Synchronisationsbenutzer, eine Publikation und eine Subskription in der Clientdatenbank. Publikationen identifizieren die Tabellen und Spalten in Ihrer entfernten Datenbank, die synchronisiert werden sollen. Diese Tabellen und Spalten heißen **Artikel**. Eine Synchronisationssubskription subskribiert einen MobiLink-Benutzer für eine Publikation.

Aufgabe

1. Erstellen Sie Ihre MobiLink Clientdatenbank mit dem Befehlszeilen-Dienstprogramm dbinit.

Führen Sie den folgenden Befehl aus:

```
dbinit -i -k -dba DBA,sql remotel
```

Die Optionen -i und -k ignorieren die jConnect-Unterstützung und Watcom-SQL-Kompatibilitätsansichten.

2. Starten Sie Ihre MobiLink-Clientdatenbank mit dem Befehlszeilen-Dienstprogramm dbeng16.

Führen Sie den folgenden Befehl aus:

```
dbsrv16 remotel
```

3. Führen Sie den folgenden Befehl aus, um eine Verbindung zur MobiLink-Clientdatenbank mithilfe von Interactive SQL herzustellen:

```
dbisql -c "SERVER=remotel;UID=DBA;PWD=sql"
```

4. Erstellen Sie die RemoteOrders-Tabelle, indem Sie folgende SQL-Anweisung in Interactive SQL ausführen:

```
CREATE TABLE RemoteOrders (
    order_id          INTEGER NOT NULL,
    product_id        INTEGER NOT NULL,
    quantity          INTEGER,
    order_status       VARCHAR(10) DEFAULT 'new',
    PRIMARY KEY(order_id)
);
```

5. Erstellen Sie die OrderComments-Tabelle, indem Sie folgende Anweisung in Interactive SQL ausführen:

```
CREATE TABLE OrderComments (
    comment_id        INTEGER NOT NULL,
    order_id           INTEGER NOT NULL,
    order_comment      VARCHAR(255),
    PRIMARY KEY(comment_id),
    FOREIGN KEY(order_id) REFERENCES RemoteOrders(order_id)
);
```

6. Führen Sie die folgenden Anweisungen in Interactive SQL aus, um Ihren MobiLink-Synchronisationsbenutzer sowie die Publikation und die Subskription zu erstellen:

```
CREATE SYNCHRONIZATION USER ml_sales1;
CREATE PUBLICATION order_publ (TABLE RemoteOrders, TABLE OrderComments);
CREATE SYNCHRONIZATION SUBSCRIPTION TO order_publ FOR ml_sales1
    TYPE TCPIP ADDRESS 'host=localhost';
```

Hinweis

Mit den Klauseln TYPE und ADDRESS in der CREATE SYNCHRONIZATION SUBSCRIPTION-Anweisung legen Sie fest, wie die Verbindung zum MobiLink-Server hergestellt wird.

Sie können Publikationen verwenden, um zu bestimmen, welche Daten synchronisiert werden. In diesem Fall legen Sie die Tabellen RemoteOrders und OrderComments vollständig fest.

Ergebnisse

Die entfernte SQL Anywhere-Datenbank wird erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 8: Synchronisieren](#)“ auf Seite 196.

Siehe auch

- „Dienstprogramm Initialisierung (dbinit)“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „MobiLink-Clients“ [[MobiLink - Clientadministration](#)]
- „CREATE SYNCHRONIZATION USER-Anweisung [MobiLink]“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]
- „CREATE PUBLICATION-Anweisung [MobiLink] [SQL Remote]“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]
- „CREATE SYNCHRONIZATION SUBSCRIPTION-Anweisung [MobiLink]“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]

Lektion 8: Synchronisieren

Das dbmlsync-Dienstprogramm startet die MobiLink-Synchronisation für entfernte SQL Anywhere-Datenbanken. Bevor Sie dbmlsync starten, fügen Sie Bestelldaten und -kommentare zu Ihrer entfernten Datenbank hinzu.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Textdatei-Datenquelle](#)“ auf Seite 173.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Verwenden der direkten Zeilenbehandlung](#)“ auf Seite 172.

Aufgabe

1. Verbinden Sie sich über Interactive SQL mit der MobiLink-Clientdatenbank, falls dies nicht schon geschehen ist, und führen Sie folgenden Befehl aus:

```
dbisql -c "SERVER=remotel;UID=DBA;PWD=sql"
```

2. Führen Sie die folgende Anweisung aus, um der RemoteOrders-Tabelle in der Clientdatenbank eine Bestellung hinzuzufügen:

```
INSERT INTO RemoteOrders (order_id, product_id, quantity, order_status)
VALUES (1,12312,10,'new');
```

3. Fügen Sie der OrderComments-Tabelle in der Clientdatenbank einen Kommentar hinzu, indem Sie folgende Anweisung in Interactive SQL ausführen:

```
INSERT INTO OrderComments (comment_id, order_id, order_comment)
VALUES (1,1,'send promotional material with the order');
```

4. Führen Sie die folgende Anweisung in Interactive SQL aus, um Ihre Änderungen festzuschreiben:

```
COMMIT;
```

5. Führen Sie den folgenden Befehl aus:

```
dbmlsync -c "SERVER=remotel;UID=DBA;PWD=sql" -o rem1.txt -v+
```

Die folgende Tabelle enthält eine Beschreibung für jede dbmlsync-Option, die in dieser Lektion verwendet wurde:

Option	Beschreibung
-c	Legt die Verbindungszeichenfolge fest.
-o	Legt die Meldungslogdatei <i>rem1.txt</i> fest
-v+	Die Option -v legt fest, welche Informationen protokolliert werden. Mit -v+ wird die maximale ausführliche Protokollierung aktiviert.

Sobald Sie den MobiLink-Synchronisationsclient gestartet haben, werden Sie darüber informiert, dass die Synchronisation durchgeführt wurde. Bei der SQL-basierten Synchronisation werden Zeilen aus der RemoteOrders-Tabelle des Clients in die RemoteOrders-Tabelle der konsolidierten Datenbank übertragen.

Bei der Java- oder .NET-Verarbeitung wurden Ihre Kommentare in *orderComments.txt* eingefügt.

6. Schließen Sie alle SQL Anywhere MobiLink-Clientfenster.
7. Fügen Sie eine Antwort in *orderResponses.txt* ein, um den Download zur entfernten Datenbank durchzuführen. Diese Aktion wird serverseitig durchgeführt.

Fügen Sie den folgenden Text zu *orderResponses.txt* hinzu. Trennen Sie einzelne Einträge mit dem Tabulatorzeichen. Am Zeilenende drücken Sie die Eingabetaste.

```
1    1    Promotional material shipped
```

8. Führen Sie die Synchronisation mithilfe des dbmlsync-Clientdienstprogramms aus.

Diese Aktion wird clientseitig durchgeführt.

Führen Sie den folgenden Befehl aus:

```
dbmlsync -c "SERVER=remotel;UID=DBA;PWD=sql" -o rem1.txt -v+
```

Das MobiLink-Clientdienstprogramm wird geöffnet.

Hinweis

Zeilen, die mithilfe der direkten Zeilenbehandlung heruntergeladen wurden, werden nicht mit der mlsrv16-Option -v+ ausgegeben, sondern mit der dbmlsync-Option -v+ in die entfernte Logdatei geschrieben.

9. In Interactive SQL überprüfen Sie mit der OrderComments-Tabelle, ob die Zeile heruntergeladen wurde.

Führen Sie hierzu die folgende SQL-Anweisung aus:

```
SELECT * FROM OrderComments;
```

Ergebnisse

Die entfernte SQL Anywhere-Datenbank wird aktualisiert und mit der konsolidierten Datenbank synchronisiert.

Nächste Schritte

Gehen Sie weiter zu „Aufräumen“ auf Seite 198.

Siehe auch

- „SQL Anywhere-Clients“ [[MobiLink - Clientadministration](#)]
- „MobiLink SQL Anywhere Client-Dienstprogramm (dbmlsync)“ [[MobiLink - Clientadministration](#)]

Aufräumen

Daten der praktischen Einführung von Ihrem Computer entfernen

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Textdatei-Datenquelle](#)“ auf Seite 173.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Verwenden der direkten Zeilenbehandlung](#)“ auf Seite 172.

Aufgabe

1. Schließen Sie alle Instanzen von Interactive SQL.
2. Schließen Sie SQL Anywhere, MobiLink und den Synchronisationsclient.
3. Löschen Sie alle Datenquellen im Zusammenhang mit der praktischen Einführung:
 - a. Starten Sie den ODBC-Administrator.

Führen Sie den folgenden Befehl aus:

odbcad32

- b. Entfernen Sie die Datenquelle **mldirect_db**.
4. Löschen Sie die konsolidierten und entfernten Datenbanken:
 - a. Navigieren Sie zum Verzeichnis mit Ihren konsolidierten und entfernten Datenbanken.
 - b. Löschen Sie *MLconsolidated.db*, *MLconsolidated.log*, *remote1.db* und *remote1.log*.

Ergebnisse

Die Daten der praktischen Einführung werden von Ihrem Computer entfernt.

Nächste Schritte

Keine.

Praktische Einführung: Synchronisation mit Microsoft Excel

Die direkte Zeilenbehandlung ermöglicht es Ihnen, entfernte Daten an beliebige zentrale Datenquellen, Anwendungen oder Webdienste zu übermitteln.

Diese praktische Einführung zeigt Ihnen die grundlegenden Schritte bei der Verwendung der direkten Zeilenbehandlung zur Synchronisation von Daten in einer Microsoft Excel-Tabellenkalkulation mit MobiLink-Clients. Sie zeigt Ihnen an einem Beispiel, wie die direkte Zeilenbehandlung von MobiLink mit einer Java-Implementierung erfolgt, damit Sie eine andere Datenquelle als eine unterstützte konsolidierte Datenbank verwenden können.

Erforderliche Software

- SQL Anywhere 16
- Java Software Development Kit
- Microsoft Excel 2007 oder höher

Kenntnisse und Erfahrungen

Hierzu benötigen Sie Folgendes:

- Kenntnisse über Java
- Microsoft Excel-Kenntnisse
- Grundkenntnisse über MobiLink-Ereignisskripten

Privilegien

Sie müssen die folgenden Rollen und Privilegien für die konsolidierte Datenbank haben:

- SYS_AUTH_RESOURCE_ROLE-Kompatibilitätsrolle
- MONITOR-Systemprivileg

Sie müssen die folgenden Rollen und Privilegien für die entfernte Datenbank haben:

- SYS_REPLICATION_ADMIN_ROLE-Systemrolle
- SYS_RUN_REPLICATION_ROLE-Systemrolle

Übersicht

In dieser praktischen Einführung wird Folgendes veranschaulicht:

- Verwenden der MobiLink-Server-API für Java
- Erstellen von Methoden zur direkten Zeilenbehandlung in MobiLink
- Zugriff auf Daten einer Microsoft Excel-Tabelle mithilfe von Java

Siehe auch

- „MobiLink-Synchronisation“ auf Seite 1
- „Synchronisationsmethoden“ [*MobiLink - Serveradministration*]
- „Direkte Zeilenbehandlung“ [*MobiLink - Serveradministration*]
- <http://www.sybase.com/detail?id=1058600#319> (Sie benötigen ein Login für Sybase.com, um diese Seite anzuzeigen.)
- <http://sqlanywhere-forum.sybase.com>

Lektion 1: Einrichten einer Excel-Tabelle

In dieser Lektion erstellen Sie eine Excel-Tabelle und verwenden den Microsoft Excel-Treiber zum Festlegen einer ODBC-Datenquelle. In der Excel-Tabelle werden Kommentare zu Bestellungen gespeichert.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit Microsoft Excel](#)“ auf Seite 199.

Der Microsoft Excel-Treiber ist ein 32-Bit-Treiber, sodass die 32-Bit-Version des ODBC-Datenquellen-Administrators für diese praktische Einführung erforderlich ist.

Aufgabe

1. Erstellen Sie ein Arbeitsverzeichnis namens *c:\MLobjexcel* für serverseitige Komponenten.
2. Öffnen Sie Microsoft Excel und erstellen Sie eine neue Arbeitsmappe.

3. Geben Sie im Standardarbeitsblatt die folgenden Inhalte unter den Spaltentiteln **A, B, C** ein:

comment_id	order_id	order_comment
2	1	Versendetes Werbematerial
3	1	Weitere Informationen über erforderliches Material

4. Ändern Sie den Namen des Standardarbeitsblatts **Sheet1** in **order_sheet**.

- Doppelklicken Sie auf die Registerkarte **Sheet1**.
- Geben Sie **order_sheet** ein.

5. Speichern Sie die Excel-Arbeitsmappe.

Speichern Sie die Arbeitsmappe unter dem Namen *order_central.xlsx* im Arbeitsverzeichnis *c:\MObjexcel*.

6. Verwenden Sie den Microsoft Excel-Treiber, um eine ODBC-Datenquelle zu erstellen:

- Wählen Sie **Start » Programme » SQL Anywhere 16 » Administrationstools » ODBC-Datenquellen-Administrator**.
- Klicken Sie auf die Registerkarte **Benutzer-DSN**.
- Klicken Sie auf **Hinzufügen**.
- Klicken Sie auf **Microsoft Excel-Treiber (*.xls, *.xlsx, *.xlsm, *.xlsb)**.
- Klicken Sie auf **Fertig stellen**.
- Im Feld **Datenquellenname** geben Sie **excel_datasource** ein.
- Klicken Sie auf **Arbeitsmappe auswählen** und navigieren Sie zu *c:\MObjexcel\order_central.xlsx* (der Datei mit der Arbeitsmappe).
- Deaktivieren Sie die Option **Schreibgeschützt**.
- Klicken Sie auf **OK** auf allen offenen ODBC Datenquellen-Administrator-Fenstern.

Ergebnisse

Eine Excel-Arbeitsmappe und eine ODBC-Datenquelle werden erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 2: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 201.

Lektion 2: Einrichten einer konsolidierten MobiLink-Datenbank

In dieser Lektion erstellen Sie eine Datenbank und definieren eine ODBC-Datenquelle.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Excel-Tabelle](#)“ auf Seite 200.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit Microsoft Excel](#)“ auf Seite 199.

Kontext und Bemerkungen

Die konsolidierte MobiLink-Datenbank ist ein zentrales Daten-Repository mit MobiLink-Systemtabellen und gespeicherten Prozeduren, die für die Verwaltung des Synchronisationsprozesses eingesetzt werden. Die direkte Zeilenbehandlung ermöglicht es Ihnen, Synchronisationen mit einer anderen Datenquelle als mit einer konsolidierten Datenbank durchzuführen. Auch in diesem Fall benötigen Sie jedoch eine konsolidierte Datenbank, um die Informationen zu verwalten, auf die der MobiLink-Server zugreift.

Hinweis

Wenn Sie bereits eine konsolidierte MobiLink-Datenbank mit MobiLink-Systemobjekten und einem DSN eingerichtet haben, können Sie diese Lektion überspringen.

Aufgabe

1. Klicken Sie auf **Start » Programme » SQL Anywhere 16 » Administrationstools » Sybase Central**.
2. Klicken Sie auf **Extras » SQL Anywhere 16 » Datenbank erstellen**.
3. Klicken Sie auf **Weiter**.
4. Behalten Sie die Standardeinstellung **Datenbank auf diesem Computer erstellen** bei und klicken Sie dann auf **Weiter**.
5. Im Feld **Haupt-Datenbankdatei in folgender Datei speichern** geben Sie den Dateinamen und den Pfad für die Datenbank ein. Beispiel: *c:\MLobjexcel\MLconsolidated.db*.
6. Befolgen Sie die übrigen Anweisungen des **Assistenten zum Erstellen einer Datenbank** und akzeptieren Sie die Standardwerte. Wenn Sie dazu aufgefordert werden, eine Benutzer-ID und ein Kennwort für den DBA-Benutzer anzugeben, geben Sie **DBA** und **sql** ein.

Deaktivieren Sie auf der Seite **Mit der Datenbank verbinden** die Option **Datenbank nach der letzten getrennten Verbindung herunterfahren**.

7. Klicken Sie auf **Fertig stellen**.

Die Datenbank MLconsolidated wird in Sybase Central angezeigt.

8. Klicken Sie in Sybase Central auf **Extras » SQL Anywhere 16 » ODBC-Administrator öffnen**.

9. Klicken Sie auf der Registerkarte **Benutzer-DSN** auf **Hinzufügen**.
10. Klicken Sie im Fenster **Neue Datenquelle erstellen** auf **SQL Anywhere 16** und auf **Fertig stellen**.
11. Führen Sie im Fenster **ODBC-Konfiguration für SQL Anywhere** folgende Aufgaben aus:
 - a. Klicken Sie auf die Registerkarte **ODBC**.
 - b. Im Feld **Datenquellenname** geben Sie **mlexcel_db** ein.
 - c. Klicken Sie auf die Registerkarte **Login**.
 - d. Im Feld **Benutzer-ID** geben Sie **DBA** ein.
 - e. Im Feld **Kennwort** geben Sie **sql** ein.
 - f. Klicken Sie in der Dropdown-Liste **Aktion** auf **Mit einer laufenden Datenbank auf diesem Computer verbinden**.
 - g. Im Feld **Servername** geben Sie **MLconsolidated** ein.
 - h. Klicken Sie auf **OK**.
12. Schließen Sie den ODBC-Datenquellen-Administrator.

Klicken Sie auf **OK** im Fenster **ODBC-Datenquellen-Administrator**.

Ergebnisse

Eine konsolidierte Datenbank und eine ODBC-Datenquelle für die konsolidierte Datenbank werden erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 3: Erstellen einer Tabelle in Ihrer konsolidierten MobiLink-Datenbank](#)“ auf Seite 203.

Siehe auch

- „Dienstprogramm Initialisierung (dbinit)“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „CREATE TABLE-Anweisung“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]
- „MobiLink - konsolidierte Datenbanken“ [[MobiLink - Serveradministration](#)]

Lektion 3: Erstellen einer Tabelle in Ihrer konsolidierten MobiLink-Datenbank

In der nachfolgenden Lektion erstellen Sie die Tabelle RemoteOrders in der konsolidierten MobiLink-Datenbank.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Excel-Tabelle](#)“ auf Seite 200.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit Microsoft Excel](#)“ auf Seite 199.

Kontext und Bemerkungen

Die RemoteOrders-Tabelle, die Sie in dieser Lektion erstellen, enthält folgende Spalten:

order_id Eindeutiger Bezeichner für Bestellungen

product_id Eindeutiger Bezeichner für Produkte

quantity Anzahl der verkauften Artikel

order_status Bestellstatus

last_modified Datum, an dem eine Zeile zuletzt geändert wurde. Diese Spalte verwenden Sie für zeitstempelbasierte Downloads, eine gängige Technik zur Filterung von Zeilen für eine effiziente Synchronisation.

Aufgabe

1. Verbinden Sie sich mit Ihrer Datenbank mittels Interactive SQL.

Sie können Interactive SQL über Sybase Central oder eine Eingabeaufforderung starten.

- Um Interactive SQL aus Sybase Central zu starten, rechtsklicken Sie auf die **MLconsolidated - DBA**-Datenbank und auf **Interactive SQL öffnen**.
- Um Interactive SQL an einer Eingabeaufforderung zu starten, führen Sie folgenden Befehl aus:

```
dbisql -c "DSN=mlexcel_db"
```

2. Führen Sie folgendes SQL-Skript in Interactive SQL aus, um die Tabelle RemoteOrders zu erstellen.

```
CREATE TABLE RemoteOrders (  
    order_id          INTEGER NOT NULL,  
    product_id        INTEGER NOT NULL,  
    quantity          INTEGER,  
    order_status      VARCHAR(10) DEFAULT 'new',  
    last_modified      TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY(order_id)  
);
```

Interactive SQL erstellt die Tabelle RemoteOrders in Ihrer konsolidierten Datenbank.

3. Führen Sie die folgende SQL-Anweisung in Interactive SQL aus, um MobiLink-Systemtabellen und gespeicherte Prozeduren zu erstellen.

Ersetzen Sie *C:\Program Files\SQL Anywhere 16* durch den Pfad Ihrer SQL Anywhere 16-Installation.

```
READ "C:\Program Files\SQL Anywhere 16\MobiLink\setup\syncsa.sql";
```

Interactive SQL wendet *syncsa.sql* auf Ihre konsolidierte Datenbank an. Durch die Ausführung von *syncsa.sql* wird eine Reihe von Systemtabellen und gespeicherten Prozeduren mit dem Präfix **ml_** erstellt. Der MobiLink-Server arbeitet während des Synchronisationsprozesses mit diesen Tabellen und gespeicherten Prozeduren.

Ergebnisse

Die RemoteOrders-Tabelle wird in der konsolidierten Datenbank erstellt.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 4: Hinzufügen von Synchronisationsskripten“](#) auf Seite 205.

Lektion 4: Hinzufügen von Synchronisationsskripten

In dieser Lektion fügen Sie Skripten zu Ihrer konsolidierten Datenbank für die SQL-Zeilenbehandlung und die direkte Zeilenbehandlung hinzu.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Einrichten einer Excel-Tabelle“](#) auf Seite 200.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Synchronisation mit Microsoft Excel“](#) auf Seite 199.

Kontext und Bemerkungen

Mit der SQL-Zeilenbehandlung können Sie entfernte Daten mit Tabellen in Ihrer konsolidierten MobiLink-Datenbank synchronisieren. SQL-basierte Skripten legen Folgendes fest:

- Wie Upload-Daten von einem MobiLink-Client in die konsolidierte Datenbank übernommen werden
- Welche Daten von der konsolidierten Datenbank heruntergeladen werden

In dieser Lektion schreiben Sie Synchronisationsskripten für die folgenden SQL-basierten Upload- und Download-Ereignisse:

- **upload_insert** Dieses Ereignis definiert, wie neue Bestellungen aus einer entfernten Clientdatenbank in die konsolidierte Datenbank übernommen werden.
- **download_cursor** Dieses Ereignis definiert die Bestellungen, die auf entfernte Clients heruntergeladen werden sollen.
- **download_delete_cursor** Dieses Ereignis ist erforderlich, wenn Sie Synchronisationsskripten verwenden, die nicht nur einen reinen Upload ausführen. Der MobiLink-Server wird konfiguriert, dieses Ereignis für diese praktische Einführung zu ignorieren.

Sie fügen mithilfe der direkten Zeilenbehandlung spezielle Verarbeitungsschritte zu einem SQL-basierten Synchronisationssystem hinzu. Dabei registrieren Sie Methodennamen, die den Ereignissen `handle_UploadData`, `handle_DownloadData`, `download_cursor` und `download_delete_cursor` entsprechen. Sie erstellen Ihre eigene Java -Klasse in einer späteren Lektion.

Aufgabe

1. Verbinden Sie sich über Interactive SQL mit Ihrer konsolidierten Datenbank, falls dies nicht schon geschehen ist.

Führen Sie den folgenden Befehl aus:

```
dbisql -c "DSN=mlexcel_db"
```

2. Verwenden Sie die gespeicherte Prozedur `ml_add_table_script`, um SQL-basierte Tabellenskripten für die Ereignisse `upload_insert`, `download_cursor` und `download_delete_cursor` hinzuzufügen.

Führen Sie hierzu die folgende SQL-Anweisung in Interactive SQL aus. Das `upload_insert`-Skript fügt `order_id`, `product_id`, `quantity` und `order_status` in die konsolidierte MobiLink-Datenbank ein. Das `download_cursor`-Skript verwendet eine Filterung auf Zeitstempelbasis für den Download aktualisierter Zeilen auf entfernte Clients.

```
CALL ml_add_table_script( 'default', 'RemoteOrders',
    'upload_insert',
    'INSERT INTO RemoteOrders( order_id, product_id, quantity,
    order_status)
    VALUES( {ml r.order_id}, {ml r.product_id}, {ml r.quantity}, {ml
    r.order_status} )' );

CALL ml_add_table_script( 'default', 'RemoteOrders',
    'download_cursor',
    'SELECT order_id, product_id, quantity, order_status
    FROM RemoteOrders WHERE last_modified >= {ml s.last_table_download}');

CALL ml_add_table_script( 'default', 'RemoteOrders',
    'download_delete_cursor', '--{ml_ignore}');

COMMIT
```

3. Registrieren Sie Java -Methoden für die Ereignisse `handle_UploadData` und `handle_DownloadData`.

Führen Sie hierzu die folgende SQL-Anweisung in Interactive SQL aus:

```
CALL ml_add_java_connection_script( 'default',
    'handle_UploadData',
    'MobiLinkOrders.GetUpload' );

CALL ml_add_java_connection_script( 'default',
    'handle_DownloadData',
    'MobiLinkOrders.SetDownload' );
```

Interactive SQL registriert die `GetUpload`- und `SetDownload`-Methoden jeweils für die Ereignisse `handle_UploadData` und `handle_DownloadData`. Sie erstellen diese Methoden in einer folgenden Lektion.

4. Registrieren Sie die Ereignisse `download_cursor` und `download_delete_cursor`.

Führen Sie folgendes SQL-Skript in Interactive SQL aus:

```
CALL ml_add_table_script( 'default', 'OrderComments',  
    'download_cursor', '--{ml_ignore}');  
  
CALL ml_add_table_script( 'default', 'OrderComments',  
    'download_delete_cursor', '--{ml_ignore}');
```

Die Ereignisse download_cursor und download_delete_cursor müssen für die OrderComments-Tabelle registriert werden, wenn Sie Skripten verwenden, weil die Synchronisation in beiden Richtungen erfolgt und nicht nur ein reiner Upload ist. Siehe „Erforderliche Skripten“ [[MobiLink - Serveradministration](#)].

5. Schreiben Sie Ihre Änderungen fest.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
COMMIT;
```

Ergebnisse

Die Ereignisse upload_insert, download_cursor und download_delete_cursor werden der Datenbank hinzugefügt. Die Methodennamen zu den Ereignissen handle_UploadData, handle_DownloadData, download_cursor und download_delete_cursor werden registriert.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 5: Erstellen einer Java-Klasse mit direkter Zeilenbehandlung in MobiLink](#)“ auf Seite 208.

Siehe auch

- „Überblick über MobiLink-Ereignisse“ [[MobiLink - Serveradministration](#)]
- „Hinzufügungen und Löschungen von Skripten“ [[MobiLink - Serveradministration](#)]
- „Skripten für den Upload von Zeilen“ [[MobiLink - Serveradministration](#)]
- „Skripten für den Download von Zeilen“ [[MobiLink - Serveradministration](#)]
- „upload_insert (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „upload_update (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „upload_delete (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „download_cursor (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „download_delete_cursor (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „Direkte Zeilenbehandlung“ [[MobiLink - Serveradministration](#)]
- „Direkte Uploads“ [[MobiLink - Serveradministration](#)]
- „Direkte Downloads“ [[MobiLink - Serveradministration](#)]
- „Implementieren zeitstempelbasierter Downloads“ [[MobiLink - Serveradministration](#)]
- „Partitionierte Zeilen in entfernten Datenbanken“ [[MobiLink - Serveradministration](#)]

Lektion 5: Erstellen einer Java-Klasse mit direkter Zeilenbehandlung in MobiLink

In dieser Lektion verwenden Sie die direkte Zeilenbehandlung für die Verarbeitung von Zeilen in der OrderComments-Tabelle Ihrer Clientdatenbank.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Excel-Tabelle](#)“ auf Seite 200.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit Microsoft Excel](#)“ auf Seite 199.

Kontext und Bemerkungen

In dieser Lektion fügen Sie die folgende Methode für die direkte Zeilenbehandlung hinzu:

- **GetUpload** Diese Methode verwenden Sie für das handle_UploadData-Ereignis. GetUpload schreibt hochgeladene Kommentare in die Excel-Tabelle *order_central.xlsx*.
- **SetDownload** Diese Methode verwenden Sie für das handle_DownloadData-Ereignis. SetDownload fragt die in der Excel-Tabelle *order_central.xlsx* gespeicherten Daten ab und sendet sie an entfernte Clients.

Mit dem nachfolgenden Verfahren können Sie eine Java-Klasse inklusive Ihrer Methoden für die Prozessverarbeitung erstellen. Eine vollständige Auflistung finden Sie unter „[Vollständige MobiLinkOrders-Auflistung \(Java\)](#)“ auf Seite 212.

Aufgabe

1. Schreiben Sie eine neue Klasse mit dem Namen MobiLinkOrders.

Schreiben Sie den folgenden Programmcode:

```
import anywhere.ml.script.*;
import java.io.*;
import java.sql.*;

public class MobiLinkOrders {
```

2. Deklarieren Sie eine DBConnectionContext-Instanzvariable auf Klassenebene.

Fügen Sie den folgenden Code hinzu:

```
// Class level DBConnectionContext
DBConnectionContext _cc;
```

Der MobiLink-Server übergibt eine DBConnectionContext-Instanz an Ihren Klassenkonstruktor. DBConnectionContext umfasst Informationen zur aktuellen Verbindung mit der konsolidierten MobiLink-Datenbank.

3. Erstellen Sie Ihren Klassenkonstruktor.

Ihr Klassenkonstruktor legt Ihre DBConnectionContext-Instanz auf Klassenebene fest.

Fügen Sie den folgenden Code hinzu:

```
public MobiLinkOrders( DBConnectionContext cc )
    throws IOException, FileNotFoundException {
    // Declare a class-level DBConnectionContext
    _cc = cc;
}
```

4. Schreiben Sie die GetUpload-Methode.

Die GetUpload-Methode ruft eine UploadedTableData-Klasseninstanz ab, die die OrderComments-Tabelle darstellt. Die OrderComments-Tabelle enthält spezielle Kommentare von entfernten Vertriebsmitarbeitern. Sie erstellen diese Tabelle in einer späteren Lektion.

Die UploadedTableData getInserts-Methode gibt eine Ergebnismenge für neue Bestellkommentare zurück.

Fügen Sie den folgenden Code hinzu:

```
// Method for the handle_UploadData synchronization event
public void GetUpload( UploadData ut )
    throws SQLException, IOException {
    // Get an UploadedTableData for OrderComments
    UploadedTableData orderCommentsTbl =
    ut.getUploadedTableByName("OrderComments");

    // Get inserts uploaded by the MobiLink client
    ResultSet insertResultSet = orderCommentsTbl.getInserts();

    try {
        // Connect to the excel worksheet through ODBC
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection con =
        DriverManager.getConnection( "jdbc:odbc:excel_datasource" );

        while( insertResultSet.next() ) {
            // Get order comments
            int _commentID = insertResultSet.getInt("comment_id");
            int _orderID = insertResultSet.getInt("order_id");
            String _specialComments =
            insertResultSet.getString("order_comment");

            // Execute an insert statement to add the order comment
            to the worksheet
            PreparedStatement st = con.prepareStatement("INSERT INTO
            [order_sheet$]"
            + "(order_id, comment_id, order_comment) VALUES
            (?, ?, ?)" );
            st.setString( 1, Integer.toString(_orderID) );
            st.setString( 2, Integer.toString(_commentID) );
            st.setString( 3, _specialComments );
            st.executeUpdate();
            st.close();
        }
        con.close();
    } catch(Exception ex) {
```

```

        System.err.print("Exception: ");
        System.err.println(ex.getMessage());
    } finally {
        insertResultSet.close();
    }
}

```

5. Erstellen Sie die SetDownload-Methode:

- a. Rufen Sie eine Klasseninstanz ab, die die OrderComments-Tabelle darstellt.

Verwenden Sie die DBConnectionContext getDownloadData-Methode, um eine DownloadData-Instanz zu erhalten. Verwenden Sie die DownloadData getDownloadTableByName-Methode zur Rückgabe einer DownloadTableData-Instanz für die OrderComments-Tabelle.

Fügen Sie den folgenden Code hinzu:

```

public void SetDownload() throws SQLException, IOException {
    DownloadData download_d = _cc.getDownloadData();
    DownloadTableData download_td =
download_d.getDownloadTableByName( "OrderComments" );
}

```

Hinweis

Die Erstellung dieser Tabelle in der entfernten Datenbank wird in „[Lektion 7: Einrichten der konsolidierten MobiLink-Clientdatenbank](#)“ auf Seite 215 beschrieben.

- b. Rufen Sie eine vorbereitete Anweisung auf, die Ihnen das Hinzufügen von Einfüge- oder Aktualisierungsvorgängen zum Download ermöglicht.

Verwenden Sie die DownloadTableData getUpsertPreparedStatement-Methode, um eine java.sql.PreparedStatement-Instanz zurückzugeben.

Fügen Sie den folgenden Code hinzu:

```

// Prepared statement to compile upserts (inserts or updates).
PreparedStatement download_upserts =
download_td.getUpsertPreparedStatement();

```

- c. Legen Sie die Downloaddaten für jede Zeile fest.

Der folgende Code durchläuft die Tabelle *order_central.xlsx* und fügt dem MobiLink-Download Daten hinzu.

Fügen Sie den folgenden Code hinzu:

```

try {
    // Connect to the excel worksheet through ODBC
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection con =
DriverManager.getConnection( "jdbc:odbc:excel_datasource" );

    // Retrieve all the rows in the worksheet
    Statement st = con.createStatement();
    ResultSet Excel_rs = st.executeQuery( "select * from
[order_sheet$]" );

    while (Excel_rs.next()) {
        // Retrieve the row data
        int Excel_comment_id = Excel_rs.getInt(1);
        int Excel_order_id = Excel_rs.getInt(2);
        String Excel_comment = Excel_rs.getString(3);
    }
}

```



```

        // Add the Excel data to the MobiLink download.
        download_upserts.setInt( 1, Excel_comment_id );
        download_upserts.setInt( 2, Excel_order_id );
        download_upserts.setString( 3, Excel_comment );
        download_upserts.executeUpdate();
    }

    // Close the excel result set, statement, and connection.
    Excel_rs.close();
    st.close();
    con.close();
} catch (Exception ex) {
    System.err.print("Exception: ");
    System.err.println(ex.getMessage());
}
}

```

- d. Schließen Sie die vorbereitete Anweisung, die für das Hinzufügen oder Aktualisieren im Download verwendet wurde, beenden Sie die Methode und die Klasse.

Fügen Sie den folgenden Code hinzu:

```

        finally {
            download_upserts.close();
        }
    }
}

```

6. Speichern Sie den Java-Code unter dem Namen *MobiLinkOrders.java* in Ihrem Arbeitsverzeichnis *c:\MLobjexcel*.

Hinweise zur Überprüfung des Codes in *MobiLinkOrders.java* finden Sie unter „[Vollständige MobiLinkOrders-Auflistung \(Java\)](#)“ auf Seite 212.

7. Kompilieren Sie Ihre Klassendatei.

- Navigieren Sie zum Verzeichnis mit Ihren Java-Quelldateien.
- Kompilieren Sie *MobiLinkOrders* mit Referenzen zur *MobiLink-Server-API-Bibliothek* für Java. Sie müssen *mlscript.jar* (in *%SQLANY16%\Java*) referenzieren.

Führen Sie den folgenden Befehl aus, wobei Sie *C:\Program Files\SQL Anywhere 16* durch Ihr SQL Anywhere 16-Verzeichnis ersetzen:

```

javac -classpath "C:\Program Files\SQL Anywhere 16\java\mlscript.jar"
MobiLinkOrders.java

```

Ergebnisse

Zeilen in der *OrderComments*-Tabelle Ihrer Clientdatenbank werden aktualisiert.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 6: Starten des MobiLink-Servers](#)“ auf Seite 213.

Siehe auch

- „Direkte Zeilenbehandlung“ [[MobiLink - Serveradministration](#)]
- „Schreiben eines Synchronisationsskripts in Java“ [[MobiLink - Serveradministration](#)]

Vollständige MobiLinkOrders-Auflistung (Java)

Die folgende Auflistung zeigt den vollständigen Quellcode der Java MobiLinkOrders-Klasse, die für diese praktische Einführung verwendet wird. Eine schrittweise Erklärung finden Sie unter „[Lektion 5: Erstellen einer Java-Klasse mit direkter Zeilenbehandlung in MobiLink](#)“ auf Seite 208.

```
import ianywhere.ml.script.*;
import java.io.*;
import java.sql.*;

public class MobiLinkOrders {

    // Class level DBConnectionContext
    DBConnectionContext _cc;

    public MobiLinkOrders( DBConnectionContext cc )
        throws IOException, FileNotFoundException {
        // Declare a class-level DBConnectionContext
        _cc = cc;
    }

    // Method for the handle_UploadData synchronization event
    public void GetUpload( UploadData ut )
        throws SQLException, IOException {
        // Get an UploadedTableData for OrderComments
        UploadedTableData orderCommentsTbl =
            ut.getUploadedTableByName( "OrderComments" );

        // Get inserts uploaded by the MobiLink client
        ResultSet insertResultSet = orderCommentsTbl.getInserts();

        try {
            // Connect to the excel worksheet through ODBC
            Class.forName( "sun.jdbc.odbc.JdbcOdbcDriver" );
            Connection con =
                DriverManager.getConnection( "jdbc:odbc:excel_datasource" );

            while( insertResultSet.next() ) {
                // Get order comments
                int _commentID = insertResultSet.getInt( "comment_id" );
                int _orderID = insertResultSet.getInt( "order_id" );
                String _specialComments =
                    insertResultSet.getString( "order_comment" );

                // Execute an insert statement to add the order comment to
                the worksheet
                PreparedStatement st = con.prepareStatement( "INSERT INTO
[order_sheet$]"
                    + "(order_id, comment_id, order_comment) VALUES
(?,?,?)" );
                st.setString( 1, Integer.toString(_orderID) );
                st.setString( 2, Integer.toString(_commentID) );
                st.setString( 3, _specialComments );
                st.executeUpdate();
                st.close();
            }
        }
```

```

        con.close();
    } catch (Exception ex) {
        System.err.print("Exception: ");
        System.err.println(ex.getMessage());
    } finally {
        insertResultSet.close();
    }
}

public void SetDownload() throws SQLException, IOException {
    DownloadData download_d = _cc.getDownloadData();
    DownloadTableData download_td =
download_d.getDownloadTableByName( "OrderComments" );

    // Prepared statement to compile upserts (inserts or updates).
    PreparedStatement download_upserts =
download_td.getUpsertPreparedStatement();

    try {
        // Connect to the excel worksheet through ODBC
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection con =
DriverManager.getConnection( "jdbc:odbc:excel_datasource" );

        // Retrieve all the rows in the worksheet
        Statement st = con.createStatement();
        ResultSet Excel_rs = st.executeQuery( "select * from [order_sheet
$]" );

        while (Excel_rs.next()) {
            // Retrieve the row data
            int Excel_comment_id = Excel_rs.getInt(1);
            int Excel_order_id = Excel_rs.getInt(2);
            String Excel_comment = Excel_rs.getString(3);

            // Add the Excel data to the MobiLink download.
            download_upserts.setInt( 1, Excel_comment_id );
            download_upserts.setInt( 2, Excel_order_id );
            download_upserts.setString( 3, Excel_comment );
            download_upserts.executeUpdate();
        }

        // Close the excel result set, statement, and connection.
        Excel_rs.close();
        st.close();
        con.close();
    } catch (Exception ex) {
        System.err.print("Exception: ");
        System.err.println(ex.getMessage());
    } finally {
        download_upserts.close();
    }
}
}

```

Lektion 6: Starten des MobiLink-Servers

In dieser Lektion starten Sie den MobiLink-Server (mlsrv16) mit der Option -c, um eine Verbindung zu Ihrer konsolidierten Datenbank herzustellen, und mit der Option -sl java, um Ihre Java-Klasse zu laden.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Excel-Tabelle](#)“ auf Seite 200.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit Microsoft Excel](#)“ auf Seite 199.

Aufgabe

- Verbinden Sie sich mit Ihrer konsolidierten Datenbank und laden Sie die Klasse auf der mlsrv16-Befehlszeile.

Führen Sie den folgenden Befehl aus. Ersetzen Sie *c:\MLobjexcel* durch den Pfad Ihrer Java-Quelldateien.

```
mlsrv16 -c "DSN=mlexcel_db" -o serverOut.txt -v+ -dl -zu+ -x tcpip -sl  
java (-cp c:\MLobjexcel)
```

Das MobiLink-Server-Meldungsfenster wird eingeblendet.

Nachfolgend finden Sie eine Beschreibung der einzelnen MobiLink-Serveroptionen, die in dieser praktischen Einführung verwendet werden. Die Optionen -o, -v und -dl bieten Informationen zur Fehlersuche und -behebung. Die Verwendung dieser Protokollierungsoptionen empfiehlt sich in einer Entwicklungsumgebung. Aus Gründen der Performance werden -v+ und -dl in der Produktion normalerweise nicht verwendet.

Option	Beschreibung
-c	Steht vor der Verbindungszeichenfolge
-o	Legt die Meldungslogdatei <i>serverOut.txt</i> fest
-v+	Die Option -v legt fest, welche Informationen protokolliert werden. Mit -v+ wird die maximale ausführliche Protokollierung aktiviert.
-dl	Alle Lognachrichten auf dem Bildschirm anzeigen.
-zu+	Fügt neue Benutzer automatisch hinzu.
-x	Legt das Kommunikationsprotokoll und Parameter für MobiLink-Clients fest
-sl java	Bestimmt eine Verzeichnisgruppe für die Suche nach Klassendateien und erzwingt das Laden der Java Virtual Machine beim Start des Servers

Ergebnisse

Der MobiLink-Server wird gestartet und ist bereit für die direkte Zeilenbehandlung.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 7: Einrichten der konsolidierten MobiLink-Clientdatenbank](#)“ auf Seite 215.

Siehe auch

- „MobiLink-Serveroptionen“ [[MobiLink - Serveradministration](#)]
- „mlsrv16-Option -sl java“ [[MobiLink - Serveradministration](#)]

Lektion 7: Einrichten der konsolidierten MobiLink-Clientdatenbank

In dieser Lektion verwenden Sie eine SQL Anywhere-Datenbank für Ihre konsolidierte Datenbank und den MobiLink-Client.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Excel-Tabelle](#)“ auf Seite 200.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit Microsoft Excel](#)“ auf Seite 199.

Kontext und Bemerkungen

Dabei wird davon ausgegangen, dass sich Ihr MobiLink-Client, die konsolidierte Datenbank und der MobiLink-Server auf demselben Computer befinden.

Um die MobiLink-Clientdatenbank einzurichten, erstellen Sie die Tabellen RemoteOrders und OrderComments. Die RemoteOrders-Tabelle entspricht der RemoteOrders-Tabelle in der konsolidierten Datenbank. Der MobiLink-Server verwendet SQL-basierte Skripten für die Synchronisation der entfernten Bestellungen. Die OrderComments-Tabelle wird nur in Clientdatenbanken verwendet. Der MobiLink-Server verarbeitet die OrderComments-Tabelle mithilfe spezieller Ereignisse.

Aufgabe

1. Erstellen Sie Ihre MobiLink Clientdatenbank mit dem Befehlszeilen-Dienstprogramm dbinit.

Führen Sie den folgenden Befehl aus:

```
dbinit -i -k -dba DBA,sql remotel
```

Die Optionen -i und -k ignorieren die jConnect-Unterstützung und Watcom-SQL-Kompatibilitätsansichten.

2. Starten Sie Ihre MobiLink-Clientdatenbank mit dem Befehlszeilen-Dienstprogramm dbeng16.

Führen Sie den folgenden Befehl aus:

```
dbsrv16 remotel
```

3. Verbinden Sie sich über Interactive SQL mit Ihrem MobiLink-Client.

Führen Sie den folgenden Befehl aus:

```
dbisql -c "SERVER=remotel;UID=DBA;PWD=sql"
```

4. Erstellen Sie die Tabelle RemoteOrders.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
CREATE TABLE RemoteOrders (  
    order_id            INTEGER NOT NULL,  
    product_id          INTEGER NOT NULL,  
    quantity            INTEGER,  
    order_status        VARCHAR(10) DEFAULT 'new',  
    PRIMARY KEY(order_id)  
);
```

5. Erstellen Sie die OrderComments-Tabelle.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
CREATE TABLE OrderComments (  
    comment_id          INTEGER NOT NULL,  
    order_id            INTEGER NOT NULL,  
    order_comment        VARCHAR(255),  
    PRIMARY KEY(comment_id),  
    FOREIGN KEY(order_id) REFERENCES RemoteOrders(order_id)  
);
```

6. Erstellen Sie Ihren MobiLink-Synchronisationsbenutzer sowie die Publikation und Subskription.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
CREATE SYNCHRONIZATION USER ml_sales1;  
CREATE PUBLICATION order_publ (TABLE RemoteOrders, TABLE OrderComments);  
CREATE SYNCHRONIZATION SUBSCRIPTION TO order_publ FOR ml_sales1  
    TYPE TCPIP ADDRESS 'host=localhost';
```

Hinweis

Mit den Klauseln TYPE und ADDRESS in der CREATE SYNCHRONIZATION SUBSCRIPTION-Anweisung legen Sie fest, wie die Verbindung zum MobiLink-Server hergestellt wird.

Sie können Publikationen verwenden, um zu bestimmen, welche Daten synchronisiert werden. In diesem Fall legen Sie die gesamten RemoteOrders- und OrderComments-Tabellen fest.

Ergebnisse

Der SQL Anywhere-Clientdatenbank wird erstellt und für die Synchronisation vorbereitet.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 8: Synchronisieren](#)“ auf Seite 217.

Siehe auch

- „Dienstprogramm Initialisierung (dbinit)“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „MobiLink-Clients“ [[MobiLink - Clientadministration](#)]
- „CREATE SYNCHRONIZATION USER-Anweisung [MobiLink]“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]
- „CREATE PUBLICATION-Anweisung [MobiLink] [SQL Remote]“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]
- „CREATE SYNCHRONIZATION SUBSCRIPTION-Anweisung [MobiLink]“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]

Lektion 8: Synchronisieren

Das dbmlsync-Dienstprogramm startet die MobiLink-Synchronisation für entfernte SQL Anywhere-Datenbanken. Bevor Sie dbmlsync starten, fügen Sie Bestelldaten und -kommentare zu Ihrer entfernten Datenbank hinzu.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Excel-Tabelle](#)“ auf Seite 200.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit Microsoft Excel](#)“ auf Seite 199.

Aufgabe

1. Verbinden Sie sich über Interactive SQL mit der MobiLink-Clientdatenbank, falls dies nicht schon geschehen ist.

Führen Sie den folgenden Befehl aus:

```
dbisql -c "SERVER=remotel;UID=DBA;PWD=sql"
```

2. Fügen Sie der RemoteOrders-Tabelle in der Clientdatenbank eine Bestellung hinzu.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
INSERT INTO RemoteOrders (order_id, product_id, quantity, order_status)
VALUES (1,12312,10,'new');
```

3. Fügen Sie der OrderComments-Tabelle in der Clientdatenbank einen Kommentar hinzu.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
INSERT INTO OrderComments (comment_id, order_id, order_comment)
VALUES (1,1,'send promotional material with the order');
```

4. Schreiben Sie Ihre Änderungen fest.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
COMMIT;
```

5. Führen Sie den folgenden Befehl an einer Eingabeaufforderung aus:

```
dbmlsync -c "SERVER=remotel;UID=DBA;PWD=sql" -o rem1.txt -v+
```

Die folgende Tabelle enthält eine Beschreibung für jede dbmlsync-Option, die in dieser Lektion verwendet wurde:

Option	Beschreibung
-c	Legt die Verbindungszeichenfolge fest.
-o	Legt die Meldungslogdatei <i>rem1.txt</i> fest
-v+	Die Option -v legt fest, welche Informationen protokolliert werden. Mit -v+ wird die maximale ausführliche Protokollierung aktiviert.

Sobald Sie den MobiLink-Synchronisationsclient gestartet haben, werden Sie darüber informiert, dass die Synchronisation durchgeführt wurde. Bei der SQL-basierten Synchronisation wurden Zeilen aus der RemoteOrders-Tabelle des Clients in die RemoteOrders-Tabelle der konsolidierten Datenbank übertragen.

Die Java-Verarbeitung hat Ihren Kommentar in die Tabelle *order_central.xlsx* eingefügt. Die in der Tabelle *order_central.xlsx* gespeicherten Informationen werden auf den Client heruntergeladen.

6. In Interactive SQL überprüfen Sie mit der OrderComments-Tabelle, ob die Zeile heruntergeladen wurde.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
SELECT * FROM OrderComments;
```

Hinweis

Zeilen, die mithilfe der direkten Zeilenbehandlung heruntergeladen wurden, werden nicht von der mlsrv16-Option -v+ gedruckt, sondern von der dbmlsync-Option -v+ in der entfernten Logdatei.

Ergebnisse

Die entfernte Datenbank wird mit Bestelldaten und -kommentaren aktualisiert und die entfernte Datenbank und die konsolidierte Datenbank werden synchronisiert.

Nächste Schritte

Gehen Sie weiter zu „[Aufräumen](#)“ auf Seite 219.

Siehe auch

- „SQL Anywhere-Clients“ [[MobiLink - Clientadministration](#)]
- „MobiLink SQL Anywhere Client-Dienstprogramm (dbmlsync)“ [[MobiLink - Clientadministration](#)]

Aufräumen

Daten der praktischen Einführung von Ihrem Computer entfernen

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer Excel-Tabelle](#)“ auf Seite 200.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit Microsoft Excel](#)“ auf Seite 199.

Aufgabe

1. Schließen Sie alle Instanzen der folgenden Anwendungen:
 - Interactive SQL
 - Microsoft Excel
2. Löschen Sie die Excel-Arbeitsmappe *order_central.xlsx*.
3. Schließen Sie SQL Anywhere, MobiLink und den Synchronisationsclient.
4. Löschen Sie alle ODBC-Datenquellen im Zusammenhang mit der praktischen Einführung.
 - a. Starten Sie den ODBC-Administrator.
Führen Sie den folgenden Befehl aus:

```
odbcad32
```
 - b. Entfernen Sie die Datenquellen **excel_datasource** und **mlexcel_db**.
5. Löschen Sie die konsolidierten und entfernten Datenbanken.
 - a. Wechseln Sie zum Verzeichnis mit Ihren konsolidierten und entfernten Datenbanken.
 - b. Löschen Sie *MLconsolidated.db*, *MLconsolidated.log*, *remote1.db* und *remote1.log*.

Ergebnisse

Die Daten der praktischen Einführung werden von Ihrem Computer entfernt.

Nächste Schritte

Keine.

Praktische Einführung: Synchronisation mit XML

Diese praktische Einführung veranschaulicht, wie Sie eine Synchronisation von Daten in einer XML-Datei auf entfernten Clients vornehmen.

Die direkte Zeilenbehandlung ermöglicht es Ihnen, entfernte Daten an beliebige zentrale Datenquellen, Anwendungen oder Webdienste zu übermitteln.

In dieser praktischen Einführung implementieren Sie die direkte Zeilenbehandlung in MobiLink. Damit können Sie Datenquellen verwenden, bei denen es sich nicht um unterstützte konsolidierte Datenquellen handelt. Diese praktische Einführung verwendet eine Java-Implementierung als Beispiel.

Erforderliche Software

- SQL Anywhere 16
- Java Software Development Kit
- XML DOM-Bibliothek

Kenntnisse und Erfahrungen

Hierzu benötigen Sie Folgendes:

- Kenntnisse über Java
- XML-Kenntnisse
- XML DOM-Kenntnisse
- Grundkenntnisse über MobiLink-Ereignisskripten

Privilegien

Sie müssen die folgenden Rollen und Privilegien für die konsolidierte Datenbank haben:

- SYS_AUTH_RESOURCE_ROLE-Kompatibilitätsrolle
- MONITOR-Systemprivileg

Sie müssen die folgenden Rollen und Privilegien für die entfernte Datenbank haben:

- SYS_REPLICATION_ADMIN_ROLE-Systemrolle
- SYS_RUN_REPLICATION_ROLE-Systemrolle

Übersicht

In dieser praktischen Einführung wird Folgendes veranschaulicht:

- Verwenden Sie die MobiLink-Server-API für Java
- Erstellen von Methoden zur direkten Zeilenbehandlung in MobiLink

Siehe auch

- „MobiLink-Synchronisation“ auf Seite 1
- „Synchronisationsmethoden“ [*MobiLink - Serveradministration*]
- „Direkte Zeilenbehandlung“ [*MobiLink - Serveradministration*]
- <http://www.sybase.com/detail?id=1058600#319> (Sie benötigen ein Login für Sybase.com, um diese Seite anzuzeigen.)
- <http://sqlanywhere-forum.sybase.com>

Lektion 1: Einrichten einer XML-Datenquelle

In dieser Lektion erstellen Sie eine XML-Datei zum Speichern von Bestellinformationen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit XML](#)“ auf Seite 219.

Aufgabe

1. Erstellen Sie eine XML-Datei mit folgendem Inhalt:

```
<?xml version="1.0" encoding="UTF-8"?>
<orders></orders>
```

2. Speichern Sie die XML-Datei.

In dieser praktischen Einführung wird davon ausgegangen, dass *c:\Mlobjxml* das Arbeitsverzeichnis für serverseitige Komponenten ist. Speichern Sie die XML-Datei als *order_comments.xml* in diesem Verzeichnis.

Ergebnisse

Die XML-Datei wird erstellt

Nächste Schritte

Gehen Sie weiter zu „[Lektion 2: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ auf Seite 221.

Lektion 2: Einrichten einer konsolidierten MobiLink-Datenbank

In dieser Lektion erstellen Sie eine Datenbank und definieren eine ODBC-Datenquelle.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer XML-Datenquelle](#)“ auf Seite 221.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit XML](#)“ auf Seite 219.

Kontext und Bemerkungen

Die konsolidierte MobiLink-Datenbank ist ein zentrales Daten-Repository mit MobiLink-Systemtabellen und gespeicherten Prozeduren, die für die Verwaltung des Synchronisationsprozesses eingesetzt werden. Die direkte Zeilenbehandlung ermöglicht es Ihnen, Synchronisationen mit einer anderen Datenquelle als mit einer konsolidierten Datenbank durchzuführen. Auch in diesem Fall benötigen Sie jedoch eine konsolidierte Datenbank, um die Informationen zu verwalten, auf die der MobiLink-Server zugreift.

Hinweis

Wenn Sie bereits eine konsolidierte MobiLink-Datenbank mit MobiLink-Systemobjekten und einem DSN eingerichtet haben, können Sie diese Lektion überspringen.

Aufgabe

1. Starten Sie Sybase Central.

Klicken Sie auf **Start » Programme » SQL Anywhere 16 » Administrationstools » Sybase Central**.

2. Klicken Sie auf **Extras » SQL Anywhere 16 » Datenbank erstellen**.
3. Klicken Sie auf **Weiter**.
4. Behalten Sie die Standardeinstellung für **Datenbank auf diesem Computer erstellen** bei und klicken Sie auf **Weiter**.
5. Im Feld **Haupt-Datenbankdatei in folgender Datei speichern** geben Sie den Dateinamen und den Pfad für die Datenbank ein. Beispiel: `c:\MLobjxm\MLconsolidated.db`. Klicken Sie auf **Weiter**.
6. Befolgen Sie die übrigen Anweisungen des **Assistenten zum Erstellen einer Datenbank** und akzeptieren Sie die Standardwerte. Wenn Sie dazu aufgefordert werden, eine Benutzer-ID und ein Kennwort für den DBA-Benutzer anzugeben, geben Sie **DBA** und **sql** ein.

Deaktivieren Sie auf der Seite **Mit der Datenbank verbinden** die Option **Datenbank nach der letzten getrennten Verbindung herunterfahren**.

7. Klicken Sie auf **Fertig stellen**.

Die Datenbank MLconsolidated wird in Sybase Central angezeigt.

8. Klicken Sie auf **Schließen** im Fenster **Datenbank erstellen**, wenn das Fenster nicht automatisch geschlossen wurde.
9. Klicken Sie auf **Extras » SQL Anywhere 16 » ODBC-Administrator öffnen**.

10. Klicken Sie auf der Registerkarte **Benutzer-DSN** auf **Hinzufügen**.
11. Klicken Sie im Fenster **Neue Datenquelle erstellen** auf **SQL Anywhere 16** und auf **Fertig stellen**.
12. Führen Sie im Fenster **ODBC-Konfiguration für SQL Anywhere** folgende Aufgaben aus:
 - a. Klicken Sie auf die Registerkarte **ODBC**.
 - b. Im Feld **Datenquellenname** geben Sie **mlxml_db** ein.
 - c. Klicken Sie auf die Registerkarte **Login**.
 - d. Im Feld **Benutzer-ID** geben Sie **DBA** ein.
 - e. Im Feld **Kennwort** geben Sie **sql** ein.
 - f. Im Feld **Servername** geben Sie **MLconsolidated** ein.
 - g. Klicken Sie auf **OK**.
13. Schließen Sie den ODBC-Datenquellen-Administrator.

Klicken Sie auf **OK** im Fenster **ODBC-Datenquellen-Administrator**.

Ergebnisse

Eine Datenbank wird erstellt und die ODBC-Datenquelle wird definiert.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 3: Erstellen einer Tabelle in Ihrer konsolidierten MobiLink-Datenbank](#)“ auf Seite 223.

Siehe auch

- „Dienstprogramm Initialisierung (dbinit)“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „MobiLink - konsolidierte Datenbanken“ [[MobiLink - Serveradministration](#)]

Lektion 3: Erstellen einer Tabelle in Ihrer konsolidierten MobiLink-Datenbank

Mit der nachfolgenden Prozedur erstellen Sie die Tabelle RemoteOrders in der konsolidierten MobiLink-Datenbank.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer XML-Datenquelle](#)“ auf Seite 221.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit XML](#)“ auf Seite 219.

Kontext und Bemerkungen

Die RemoteOrders-Tabelle enthält folgende Spalten:

order_id Eindeutiger Bezeichner für Bestellungen

product_id Eindeutiger Bezeichner für Produkte

quantity Anzahl der verkauften Artikel

order_status Bestellstatus

last_modified Datum, an dem eine Zeile zuletzt geändert wurde. Diese Spalte verwenden Sie für zeitstempelbasierte Downloads, eine gängige Technik zur Filterung von Zeilen für eine effiziente Synchronisation.

Aufgabe

1. Verwenden Sie Interactive SQL, um sich mit Ihrer Datenbank zu verbinden.

Sie können Interactive SQL über Sybase Central oder eine Eingabeaufforderung starten.

- Um Interactive SQL aus Sybase Central zu starten, rechtsklicken Sie auf die **MLconsolidated - DBA**-Datenbank und auf **Interactive SQL öffnen**.
- Um Interactive SQL an einer Eingabeaufforderung zu starten, führen Sie folgenden Befehl aus:

```
dbisql -c "DSN=mlxml_db"
```

2. Um die Tabelle RemoteOrders zu erstellen, führen Sie folgendes SQL-Skript in Interactive SQL aus:

```
CREATE TABLE RemoteOrders (  
    order_id          INTEGER NOT NULL,  
    product_id        INTEGER NOT NULL,  
    quantity          INTEGER,  
    order_status      VARCHAR(10) DEFAULT 'new',  
    last_modified      TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY(order_id)  
);
```

Interactive SQL erstellt die Tabelle RemoteOrders in Ihrer konsolidierten Datenbank.

3. Führen Sie die folgende Anweisung in Interactive SQL aus, um MobiLink-Systemtabellen und gespeicherte Prozeduren zu erstellen.

Ersetzen Sie *C:\Program Files\SQL Anywhere 16* durch den Pfad Ihrer SQL Anywhere 16-Installation.

```
READ "C:\Program Files\SQL Anywhere 16\MobiLink\setup\syncsa.sql";
```

Interactive SQL wendet *syncsa.sql* auf Ihre konsolidierte Datenbank an. Durch die Ausführung von *syncsa.sql* wird eine Reihe von Systemtabellen und gespeicherten Prozeduren mit dem Präfix **ml_** erstellt. Der MobiLink-Server arbeitet während des Synchronisationsprozesses mit diesen Tabellen und gespeicherten Prozeduren.

Ergebnisse

Die RemoteOrders-Tabelle wird erstellt und die MobiLink-Systemtabellen und gespeicherten Prozeduren werden installiert.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 4: Hinzufügen von Synchronisationsskripten“ auf Seite 225](#).

Siehe auch

- [„CREATE TABLE-Anweisung“ \[SQL Anywhere Server - SQL-Referenzhandbuch\]](#)

Lektion 4: Hinzufügen von Synchronisationsskripten

In dieser Lektion fügen Sie Skripten zu Ihrer konsolidierten Datenbank für die SQL-Zeilenbehandlung und die direkte Zeilenbehandlung hinzu.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Einrichten einer XML-Datenquelle“ auf Seite 221](#).

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Synchronisation mit XML“ auf Seite 219](#).

Kontext und Bemerkungen

Mit der SQL-Zeilenbehandlung können Sie entfernte Daten mit Tabellen in Ihrer konsolidierten MobiLink-Datenbank synchronisieren. SQL-basierte Skripten legen Folgendes fest:

- Wie Upload-Daten von einem MobiLink-Client in die konsolidierte Datenbank übernommen werden
- Welche Daten von der konsolidierten Datenbank heruntergeladen werden

In dieser Lektion schreiben Sie Synchronisationsskripten für die folgenden SQL-basierten Upload- und Download-Ereignisse:

- **upload_insert** Dieses Ereignis definiert, wie neue Bestellungen aus einer entfernten Clientdatenbank in die konsolidierte Datenbank übernommen werden.
- **download_cursor** Dieses Ereignis definiert die Bestellungen, die auf entfernte Clients heruntergeladen werden sollen.
- **download_delete_cursor** Dieses Ereignis ist erforderlich, wenn Sie Synchronisationsskripten verwenden, die nicht nur einen reinen Upload ausführen. Der MobiLink-Server wird konfiguriert, dieses Ereignis für diese praktische Einführung zu ignorieren.

Sie fügen mithilfe der direkten Zeilenbehandlung spezielle Verarbeitungsschritte zu einem SQL-basierten Synchronisationssystem hinzu. In dieser Lektion registrieren Sie Methodennamen, die den Ereignissen

handle_UploadData, download_cursor und download_delete_cursor entsprechen. Sie erstellen Ihre eigene Java -Klasse in „[Lektion 5: Erstellen einer Java-Klasse mit direkter Zeilenbehandlung in MobiLink](#)“ auf Seite 227.

Aufgabe

1. Verbinden Sie sich über Interactive SQL mit Ihrer konsolidierten Datenbank, falls dies nicht schon geschehen ist.

Führen Sie den folgenden Befehl aus:

```
dbisql -c "DSN=mlxml_db"
```

2. Verwenden Sie die gespeicherte Prozedur ml_add_table_script, um SQL-basierte Tabellenskripten für die Ereignisse upload_insert, download_cursor und download_delete_cursor hinzuzufügen.

Führen Sie die folgende Anweisung in Interactive SQL aus. Das upload_insert-Skript fügt order_id, product_id, quantity und order_status in die konsolidierte MobiLink-Datenbank ein. Das download_cursor-Skript verwendet eine Filterung auf Zeitstempelbasis für den Download aktualisierter Zeilen auf entfernte Clients.

```
CALL ml_add_table_script( 'default', 'RemoteOrders',
    'upload_insert',
    'INSERT INTO RemoteOrders( order_id, product_id, quantity,
    order_status)
    VALUES( {ml r.order_id}, {ml r.product_id}, {ml r.quantity}, {ml
    r.order_status} )' );

CALL ml_add_table_script( 'default', 'RemoteOrders',
    'download_cursor',
    'SELECT order_id, product_id, quantity, order_status
    FROM RemoteOrders WHERE last_modified >= {ml s.last_table_download}');

CALL ml_add_table_script( 'default', 'RemoteOrders',
    'download_delete_cursor', '--{ml_ignore}');

COMMIT;
```

3. Registrieren Sie die Java -Methode für das Ereignis handle_UploadData.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
CALL ml_add_java_connection_script( 'default',
    'handle_UploadData', 'MobiLinkOrders.GetUpload' );
```

Interactive SQL registriert die GetUpload-Methode für das handle_UploadData-Ereignis. Sie erstellen die GetUpload-Methode, die eingefügte Daten aus der OrderComments-Tabelle in der MobiLink-Clientdatenbank abrufen, in einer späteren Lektion.

4. Registrieren Sie die Ereignisse download_cursor und download_delete_cursor.

Führen Sie hierzu die folgende SQL-Anweisung in Interactive SQL aus:

```
CALL ml_add_table_script( 'default', 'OrderComments',
    'download_cursor', '--{ml_ignore}');
```



```
CALL ml_add_table_script( 'default', 'OrderComments',  
  'download_delete_cursor', '--{ml_ignore}');
```

Die Ereignisse download_cursor und download_delete_cursor müssen für die OrderComments-Tabelle registriert werden, wenn Sie Skripten verwenden, weil die Synchronisation in beiden Richtungen erfolgt und nicht nur ein reiner Upload ist. Siehe „Erforderliche Skripten“ [[MobiLink - Serveradministration](#)].

5. Schreiben Sie Ihre Änderungen fest.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
COMMIT;
```

6. Schließen Sie Interactive SQL.

Ergebnisse

Die Methodennamen zu den Ereignissen handle_UploadData, handle_DownloadData, end_download, download_cursor und download_delete_cursor werden registriert.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 5: Erstellen einer Java-Klasse mit direkter Zeilenbehandlung in MobiLink](#)“ auf Seite 227.

Siehe auch

- „Überblick über MobiLink-Ereignisse“ [[MobiLink - Serveradministration](#)]
- „Hinzufügungen und Löschungen von Skripten“ [[MobiLink - Serveradministration](#)]
- „Skripten für den Upload von Zeilen“ [[MobiLink - Serveradministration](#)]
- „Skripten für den Download von Zeilen“ [[MobiLink - Serveradministration](#)]
- „upload_insert (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „upload_update (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „upload_delete (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „download_cursor (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „download_delete_cursor (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „Direkte Zeilenbehandlung“ [[MobiLink - Serveradministration](#)]
- „Direkte Uploads“ [[MobiLink - Serveradministration](#)]
- „Direkte Downloads“ [[MobiLink - Serveradministration](#)]
- „Implementieren zeitstempelbasierter Downloads“ [[MobiLink - Serveradministration](#)]
- „Partitionierte Zeilen in entfernten Datenbanken“ [[MobiLink - Serveradministration](#)]

Lektion 5: Erstellen einer Java-Klasse mit direkter Zeilenbehandlung in MobiLink

In dieser Lektion verwenden Sie die direkte Zeilenbehandlung für die Verarbeitung von Zeilen in der OrderComments-Tabelle Ihrer Clientdatenbank. Sie fügen die GetUpload-Methoden für die direkte Zeilenbehandlung für das handle_UploadData-Ereignis hinzu. GetUpload schreibt hochgeladene Kommentare in die XML-Datei.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer XML-Datenquelle](#)“ auf Seite 221.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit XML](#)“ auf Seite 219.

Kontext und Bemerkungen

Mit dem nachfolgenden Verfahren können Sie eine Java-Klasse inklusive Ihrer Methoden für die Prozessverarbeitung erstellen. Eine vollständige Auflistung finden Sie unter „[MobiLinkOrders-Java-Code-Auflistung](#)“ auf Seite 232.

Aufgabe

1. Erstellen Sie eine Klasse namens MobiLinkOrders.

Schreiben Sie den folgenden Programmcode:

```
import ianywhere.ml.script.*;
import java.io.*;
import java.sql.*;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.xml.sax.SAXException;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

// For write operation
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.TransformerConfigurationException;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

public class MobiLinkOrders {
```

2. Deklarieren Sie auf Klassenebene die Instanzvariablen für DBConnectionContext und Document. Document ist eine Klasse, die ein XML-Dokument als Objekt repräsentiert.

Schreiben Sie den folgenden Programmcode:

```
// Class level DBConnectionContext
DBConnectionContext _cc;
Document _doc;
```

Der MobiLink-Server übergibt eine DBConnectionContext-Instanz an Ihren Klassenkonstruktor. DBConnectionContext umfasst Informationen zur aktuellen Verbindung mit der konsolidierten MobiLink-Datenbank.

3. Erstellen Sie Ihren Klassenkonstruktor.

Ihr Klassenkonstruktor legt Ihre DBConnectionContext-Instanz auf Klassenebene fest.

Schreiben Sie den folgenden Programmcode:

```
public MobiLinkOrders( DBConnectionContext cc ) throws IOException,
FileNotFoundException {
    // Declare a class-level DBConnectionContext
    _cc = cc;
}
```

4. Schreiben Sie die GetUpload-Methode.

Die GetUpload-Methode ruft eine UploadedTableData-Klasseninstanz ab, die die OrderComments-Tabelle darstellt. Die OrderComments-Tabelle enthält spezielle Kommentare von entfernten Vertriebsmitarbeitern. Sie erstellen diese Tabelle in einer späteren Lektion.

Die UploadedTableData getInserts-Methode gibt eine Ergebnismenge für neue Bestellkommentare zurück.

- a. Erstellen Sie die Methodendeklaration.

Schreiben Sie den folgenden Programmcode:

```
// Method for the handle_UploadData synchronization event
public void GetUpload( UploadData ut ) throws SQLException,
IOException {
```

- b. Schreiben Sie Programmcode, der hochgeladene Einfügungen aus dem MobiLink-Client abruft.

Schreiben Sie den folgenden Programmcode:

```
// Get an UploadedTableData for the remote table
UploadedTableData remoteOrdersTable =
ut.getUploadedTableByName( "OrderComments" );

// Get inserts uploaded by the MobiLink client
// as a java.sql.ResultSet
ResultSet insertResultSet = remoteOrdersTable.getInserts();
```

- c. Schreiben Sie den Programmcode, der die bestehende XML-Datei **order_comments.xml** liest.

Schreiben Sie den folgenden Programmcode:

```
try {
    readDom( "order_comments.xml" );
```

- d. Schreiben Sie den Programmcode, der alle hochgeladenen Einfügungen in die XML-Datei einfügt.

Schreiben Sie den folgenden Programmcode:

```
// Write out each insert in the XML file
while( insertResultSet.next() ) {
    buildXML(insertResultSet);
}
```

- e. Schreiben Sie den Programmcode, der die Ausgabe in die XML-Datei vornimmt.
Schreiben Sie den folgenden Programmcode:

```
        writeXML();  
    }
```

- f. Schreiben Sie den Programmcode, der das ResultSet-Objekt schließt.
Schreiben Sie den folgenden Programmcode:

```
        finally {  
            // Close the result set of uploaded inserts  
            insertResultSet.close();  
        }  
    }
```

5. Erstellen Sie die buildXML-Methode.

Schreiben Sie den folgenden Programmcode:

```
private void buildXML( ResultSet rs ) throws SQLException {  
    int order_id = rs.getInt(1);  
    int comment_id = rs.getInt(2);  
    String order_comment = rs.getString(3);  
  
    // Create the comment object to be added to the XML file  
    Element comment = _doc.createElement("comment");  
    comment.setAttribute("id", Integer.toString(comment_id));  
    comment.appendChild(_doc.createTextNode(order_comment));  
  
    // Get the root element (orders)  
    Element root = _doc.getDocumentElement();  
  
    // Get each individual order  
    NodeList rootChildren = root.getChildNodes();  
  
    for(int i = 0; i < rootChildren.getLength(); i++) {  
        // If the order exists, add the comment to the order  
        Node n = rootChildren.item(i);  
        if(n.getNodeType() == Node.ELEMENT_NODE) {  
            Element e = (Element) n;  
            int idIntVal = Integer.parseInt(e.getAttribute("id"));  
  
            if(idIntVal == order_id) {  
                e.appendChild(comment);  
                // The comment has been added to the file, so exit  
                // the function.  
                return;  
            }  
        }  
    }  
  
    // If the order did not exist already, create it  
    Element order = _doc.createElement("order");  
    order.setAttribute("id", Integer.toString(order_id));  
  
    // Add the comment to the new order  
    order.appendChild(comment);  
    root.appendChild(order);  
}
```

6. Erstellen Sie die writeXML-Methode.

Schreiben Sie den folgenden Programmcode:

```
private void writeXML() {
    try {
        // Use a Transformer for output
        TransformerFactory tFactory =
TransformerFactory.newInstance();
        Transformer transformer = tFactory.newTransformer();

        // The XML source is _doc
        DOMSource source = new DOMSource(_doc);
        // Write the xml data to order_comments.xml
        StreamResult result = new StreamResult(new
File("order_comments.xml"));
        transformer.transform(source, result);
    } catch (TransformerConfigurationException tce) {
        // Error generated by the parser
        System.out.println ("\n** Transformer Factory error");
        System.out.println("    " + tce.getMessage() );

        // Use the contained exception, if any
        Throwable x = tce;
        if (tce.getException() != null) x = tce.getException();
        x.printStackTrace();
    } catch (TransformerException te) {
        // Error generated by the parser
        System.out.println ("\n** Transformation error");
        System.out.println("    " + te.getMessage() );

        // Use the contained exception, if any
        Throwable x = te;
        if (te.getException() != null) x = te.getException();
        x.printStackTrace();
    }
}
```

7. Erstellen Sie die readDOM-Methode.

Schreiben Sie den folgenden Programmcode:

```
private void readDom(String filename) {
    DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();

    try {
        //parse the Document data into _doc
        DocumentBuilder builder = factory.newDocumentBuilder();
        _doc = builder.parse( new File(filename) );

    } catch (SAXException sxe) {
        // Error generated during parsing
        Exception x = sxe;
        if (sxe.getException() != null) x = sxe.getException();
        x.printStackTrace();

    } catch (ParserConfigurationException pce) {
        // Parser with specified options can't be built
        pce.printStackTrace();

    } catch (IOException ioe) {
```

```
        // I/O error
        ioe.printStackTrace();
    }
}
```

8. Speichern Sie den Java-Code unter dem Namen *MobiLinkOrders.java* in Ihrem Arbeitsverzeichnis *c:\MLobjxml*.

Hinweise zur Überprüfung des Codes in *MobiLinkOrders.java* finden Sie unter „[MobiLinkOrders-Java-Code-Auflistung](#)“ auf Seite 232.

9. Kompilieren Sie Ihre Klassendatei.
 - a. Navigieren Sie zum Verzeichnis mit Ihren Java-Quelldateien.
 - b. Kompilieren Sie *MobiLinkOrders* mit Referenzen zur *MobiLink-Server-API-Bibliothek* für Java. Sie müssen *mlscript.jar* in *%SQLANY16%\Java* referenzieren und sicherstellen, dass die XML DOM-Bibliothek korrekt installiert ist.

Führen Sie den folgenden Befehl aus, wobei Sie *C:\Program Files\SQL Anywhere 16* durch Ihr SQL Anywhere 16-Verzeichnis ersetzen:

```
javac -classpath "C:\Program Files\SQL Anywhere 16\java\mlscript.jar"
MobiLinkOrders.java
```

Ergebnisse

Die Java-Klasse für direkte Zeilenbehandlung in *MobiLink* wird erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 6: Starten des MobiLink-Servers](#)“ auf Seite 235.

Siehe auch

- „[Direkte Zeilenbehandlung](#)“ [*MobiLink - Serveradministration*]
- „[Schreiben eines Synchronisationsskripts in Java](#)“ [*MobiLink - Serveradministration*]

MobiLinkOrders-Java-Code-Auflistung

Die folgende Auflistung zeigt den vollständigen Quellcode der Java *MobiLinkOrders*-Klasse, die für diese praktische Einführung verwendet wird. Eine schrittweise Erklärung finden Sie unter „[Lektion 5: Erstellen einer Java-Klasse mit direkter Zeilenbehandlung in MobiLink](#)“ auf Seite 227.

```
import ianywhere.ml.script.*;
import java.io.*;
import java.sql.*;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.xml.sax.SAXException;
```

```

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

// For write operation
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.TransformerConfigurationException;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

public class MobiLinkOrders {

    // Class level DBConnectionContext
    DBConnectionContext _cc;
    Document _doc;

    public MobiLinkOrders( DBConnectionContext cc ) throws IOException,
FileNotFoundException {
        // Declare a class-level DBConnectionContext
        _cc = cc;
    }

    // Method for the handle_UploadData synchronization event
    public void GetUpload( UploadData ut ) throws SQLException, IOException {
        // Get an UploadedTableData for the remote table
        UploadedTableData remoteOrdersTable =
ut.getUploadedTableByName("OrderComments");

        // Get inserts uploaded by the MobiLink client
        // as a java.sql.ResultSet
        ResultSet insertResultSet = remoteOrdersTable.getInserts();

        try {
            readDom("order_comments.xml");

            // Write out each insert in the XML file
            while( insertResultSet.next() ) {
                buildXML(insertResultSet);
            }
            writeXML();
        } finally {
            // Close the result set of uploaded inserts
            insertResultSet.close();
        }
    }

    private void buildXML( ResultSet rs ) throws SQLException {
        int order_id = rs.getInt(1);
        int comment_id = rs.getInt(2);
        String order_comment = rs.getString(3);

        // Create the comment object to be added to the XML file
        Element comment = _doc.createElement("comment");
        comment.setAttribute("id", Integer.toString(comment_id));
        comment.appendChild(_doc.createTextNode(order_comment));

        // Get the root element (orders)
        Element root = _doc.getDocumentElement();

        // Get each individual order

```

```

        NodeList rootChildren = root.getChildNodes();

        for(int i = 0; i < rootChildren.getLength(); i++) {
            // If the order exists, add the comment to the order
            Node n = rootChildren.item(i);
            if(n.getNodeType() == Node.ELEMENT_NODE) {
                Element e = (Element) n;
                int idIntVal = Integer.parseInt(e.getAttribute("id"));

                if(idIntVal == order_id) {
                    e.appendChild(comment);
                    // The comment has been added to the file, so exit
                    // the function
                    return;
                }
            }
        }

        // If the order did not exist already, create it
        Element order = _doc.createElement("order");
        order.setAttribute("id", Integer.toString(order_id));

        // Add the comment to the new order
        order.appendChild(comment);
        root.appendChild(order);
    }

    private void writeXML() {
        try {
            // Use a Transformer for output
            TransformerFactory tFactory = TransformerFactory.newInstance();
            Transformer transformer = tFactory.newTransformer();

            // The XML source is _doc
            DOMSource source = new DOMSource(_doc);
            // Write the xml data to order_comments.xml
            StreamResult result = new StreamResult(new
File("order_comments.xml"));
            transformer.transform(source, result);
        } catch (TransformerConfigurationException tce) {
            // Error generated by the parser
            System.out.println ("\n** Transformer Factory error");
            System.out.println("    " + tce.getMessage() );

            // Use the contained exception, if any
            Throwable x = tce;
            if (tce.getException() != null) x = tce.getException();
            x.printStackTrace();
        } catch (TransformerException te) {
            // Error generated by the parser
            System.out.println ("\n** Transformation error");
            System.out.println("    " + te.getMessage() );

            // Use the contained exception, if any
            Throwable x = te;
            if (te.getException() != null) x = te.getException();
            x.printStackTrace();
        }
    }

    private void readDom(String filename) {
        DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
    }

```



```

try {
    //parse the Document data into _doc
    DocumentBuilder builder = factory.newDocumentBuilder();
    _doc = builder.parse( new File(filename) );

} catch (SAXException sxe) {
    // Error generated during parsing
    Exception x = sxe;
    if (sxe.getException() != null) x = sxe.getException();
    x.printStackTrace();

} catch (ParserConfigurationException pce) {
    // Parser with specified options can't be built
    pce.printStackTrace();

} catch (IOException ioe) {
    // I/O error
    ioe.printStackTrace();

}
}
}

```

Lektion 6: Starten des MobiLink-Servers

In dieser Lektion starten Sie den MobiLink-Server. Sie starten den MobiLink-Server (mlsrv16) mit der Option -c, um eine Verbindung zu Ihrer konsolidierten Datenbank herzustellen, und mit der Option -sl java, um Ihre Java-Klasse zu laden.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer XML-Datenquelle](#)“ auf Seite 221.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit XML](#)“ auf Seite 219.

Aufgabe

- Verbinden Sie sich mit Ihrer konsolidierten Datenbank und laden Sie die Klasse auf der mlsrv16-Befehlszeile.

Ersetzen Sie `c:\MLobjxml` durch den Pfad Ihrer Quelldateien und führen Sie anschließend den folgenden Befehl aus:

```
mlsrv16 -c "DSN=mlxml_db" -o serverOut.txt -v+ -dl -zu+ -x tcpip -sl
java (-cp c:\MLobjxml)
```

Das MobiLink-Server-Meldungsfenster wird eingeblendet.

Nachfolgend finden Sie eine Beschreibung der einzelnen MobiLink-Serveroptionen, die in dieser praktischen Einführung verwendet werden. Die Optionen -o, -v und -dl bieten Informationen zur Fehlersuche und -behebung. Die Verwendung dieser Protokollierungsoptionen empfiehlt sich in einer Entwicklungsumgebung. Aus Gründen der Performance werden -v+ und -dl in der Produktion normalerweise nicht verwendet.

Option	Beschreibung
-c	Steht vor der Verbindungszeichenfolge
-o	Legt die Meldungslogdatei <i>serverOut.txt</i> fest
-v+	Die Option -v legt fest, welche Informationen protokolliert werden. Mit -v+ wird die maximale ausführliche Protokollierung aktiviert.
-dl	Alle Lognachrichten auf dem Bildschirm anzeigen.
-zu+	Fügt neue Benutzer automatisch hinzu.
-x	Legt das Kommunikationsprotokoll und Parameter für MobiLink-Clients fest
-sl java	Bestimmt eine Verzeichnisgruppe für die Suche nach Klassendateien und erzwingt das Laden der Java Virtual Machine beim Start des Servers

Ergebnisse

Der MobiLink-Server wird gestartet.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 7: Einrichten der konsolidierten MobiLink-Clientdatenbank](#)“ auf Seite 236.

Siehe auch

- „MobiLink-Serveroptionen“ [[MobiLink - Serveradministration](#)]
- „mlsrv16-Option -sl java“ [[MobiLink - Serveradministration](#)]

Lektion 7: Einrichten der konsolidierten MobiLink-Clientdatenbank

In dieser Lektion verwenden Sie eine SQL Anywhere-Datenbank für Ihre konsolidierte Datenbank und den MobiLink-Client. Dabei wird davon ausgegangen, dass sich Ihr MobiLink-Client, die konsolidierte Datenbank und der MobiLink-Server auf demselben Computer befinden.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer XML-Datenquelle](#)“ auf Seite 221.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit XML](#)“ auf Seite 219.

Kontext und Bemerkungen

Um die MobiLink-Clientdatenbank einzurichten, erstellen Sie die Tabellen RemoteOrders und OrderComments. Die RemoteOrders-Tabelle entspricht der RemoteOrders-Tabelle in der konsolidierten Datenbank. Der MobiLink-Server verwendet SQL-basierte Skripten für die Synchronisation der entfernten Bestellungen. Die OrderComments-Tabelle wird nur in Clientdatenbanken verwendet. Der MobiLink-Server verarbeitet die OrderComments-Tabelle mithilfe spezieller Ereignisse.

Aufgabe

1. Erstellen Sie Ihre MobiLink Clientdatenbank mit dem Befehlszeilen-Dienstprogramm dbinit.

Navigieren Sie zu `c:\Mlobjxml` und führen Sie anschließend den folgenden Befehl aus:

```
dbinit -i -k -dba DBA,sql remotel
```

Die Optionen `-i` und `-k` ignorieren die jConnect-Unterstützung und Watcom-SQL-Kompatibilitätsansichten.

2. Starten Sie Ihre MobiLink-Clientdatenbank mit dem Befehlszeilen-Dienstprogramm dbeng16.

Führen Sie den folgenden Befehl aus:

```
dbsrv16 remotel
```

3. Verbinden Sie sich über Interactive SQL mit Ihrem MobiLink-Client.

Führen Sie den folgenden Befehl aus:

```
dbisql -c "SERVER=remotel;UID=DBA;PWD=sql"
```

4. Erstellen Sie die Tabelle RemoteOrders.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
CREATE TABLE RemoteOrders (  
    order_id          INTEGER NOT NULL,  
    product_id        INTEGER NOT NULL,  
    quantity          INTEGER,  
    order_status      VARCHAR(10) DEFAULT 'new',  
    PRIMARY KEY(order_id)  
);
```

5. Erstellen Sie die OrderComments-Tabelle.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
CREATE TABLE OrderComments (  
    comment_id        INTEGER NOT NULL,  
    order_id          INTEGER NOT NULL,  
    order_comment      VARCHAR(255),  
    PRIMARY KEY(comment_id),  
    FOREIGN KEY(order_id) REFERENCES RemoteOrders(order_id)  
);
```

6. Erstellen Sie Ihren MobiLink-Synchronisationsbenutzer sowie die Publikation und Subskription.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
CREATE SYNCHRONIZATION USER ml_sales1;  
CREATE PUBLICATION order_publ (TABLE RemoteOrders, TABLE OrderComments);  
CREATE SYNCHRONIZATION SUBSCRIPTION TO order_publ FOR ml_sales1  
TYPE TCPIP ADDRESS 'host=localhost';
```

Hinweis

Mit den Klauseln TYPE und ADDRESS in der CREATE SYNCHRONIZATION SUBSCRIPTION-Anweisung legen Sie fest, wie die Verbindung zum MobiLink-Server hergestellt wird.

Sie können Publikationen verwenden, um zu bestimmen, welche Daten synchronisiert werden. In diesem Fall legen Sie die gesamten RemoteOrders- und OrderComments-Tabellen fest.

Ergebnisse

Die entfernte Datenbank wird erstellt und für die Synchronisation eingerichtet.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 8: Synchronisieren](#)“ auf Seite 238.

Siehe auch

- „Dienstprogramm Initialisierung (dbinit)“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „MobiLink-Clients“ [[MobiLink - Clientadministration](#)]
- „CREATE SYNCHRONIZATION USER-Anweisung [MobiLink]“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]
- „CREATE PUBLICATION-Anweisung [MobiLink] [SQL Remote]“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]
- „CREATE SYNCHRONIZATION SUBSCRIPTION-Anweisung [MobiLink]“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]

Lektion 8: Synchronisieren

In dieser Lektion verwenden Sie das dbmlsync-Dienstprogramm, um die MobiLink-Synchronisation zu initialisieren. Bevor Sie dbmlsync starten, fügen Sie Bestelldaten und -kommentare zu Ihrer entfernten Datenbank hinzu.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer XML-Datenquelle](#)“ auf Seite 221.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit XML](#)“ auf Seite 219.

Aufgabe

1. Verbinden Sie sich über Interactive SQL mit der MobiLink-Clientdatenbank, falls dies nicht schon geschehen ist.

Führen Sie den folgenden Befehl aus:

```
dbisql -c "SERVER=remotel;UID=DBA;PWD=sql"
```

2. Fügen Sie der RemoteOrders-Tabelle in der Clientdatenbank eine Bestellung hinzu.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
INSERT INTO RemoteOrders (order_id, product_id, quantity, order_status)
VALUES (1,12312,10,'new');
```

3. Fügen Sie der OrderComments-Tabelle in der Clientdatenbank einen Kommentar hinzu.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
INSERT INTO OrderComments (comment_id, order_id, order_comment)
VALUES (1,1,'send promotional material with the order');
```

4. Schreiben Sie Ihre Änderungen fest.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
COMMIT;
```

5. Führen Sie den folgenden Befehl an einer Eingabeaufforderung aus:

```
dbmlsync -c "SERVER=remotel;UID=DBA;PWD=sql" -o rem1.txt -v+
```

Die folgende Tabelle enthält eine Beschreibung für jede dbmlsync-Option, die in dieser Lektion verwendet wurde:

Option	Beschreibung
-c	Legt die Verbindungszeichenfolge fest.
-o	Legt die Meldungslogdatei <i>rem1.txt</i> fest
-v+	Die Option -v legt fest, welche Informationen protokolliert werden. Mit -v+ wird die maximale ausführliche Protokollierung aktiviert.

Sobald Sie den MobiLink-Synchronisationsclient gestartet haben, werden Sie darüber informiert, dass die Synchronisation durchgeführt wurde.

6. Bei der SQL-basierten Synchronisation wurden Zeilen aus der RemoteOrders-Tabelle des Clients in die RemoteOrders-Tabelle der konsolidierten Datenbank übertragen.

Führen Sie die folgenden Schritte aus, um zu überprüfen, ob die Daten, die der RemoteOrders-Tabelle des Clients hinzugefügt wurden, in die RemoteOrders-Tabelle in der konsolidierten Datenbank übertragen wurden:

- a. Um Interactive SQL an einer Eingabeaufforderung zu starten, führen Sie folgenden Befehl aus:

```
dbisql -c "DSN=mlxml_db"
```

- b. Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
SELECT * FROM RemoteOrders;
```

7. Die Java -Verarbeitung hat Ihren Kommentar in die XML-Datei eingefügt.

Gehen Sie zu `c:\MLobjxml` und öffnen Sie `order_comments.xml` in einem Texteditor, um zu überprüfen, ob der Kommentar eingefügt wurde.

Ergebnisse

Die entfernten und konsolidierten Datenbanken werden synchronisiert.

Nächste Schritte

Gehen Sie weiter zu „[Aufräumen](#)“ auf Seite 240.

Siehe auch

- „SQL Anywhere-Clients“ [[MobiLink - Clientadministration](#)]
- „MobiLink SQL Anywhere Client-Dienstprogramm (dbmlsync)“ [[MobiLink - Clientadministration](#)]

Aufräumen

Entfernen Sie die Daten der praktischen Einführung von Ihrem Computer.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten einer XML-Datenquelle](#)“ auf Seite 221.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Synchronisation mit XML](#)“ auf Seite 219.

Aufgabe

1. Schließen Sie alle Instanzen von Interactive SQL.
2. Schließen Sie SQL Anywhere, MobiLink und den Synchronisationsclient.
3. Löschen Sie alle ODBC-Datenquellen im Zusammenhang mit der praktischen Einführung.
 - a. Starten Sie den ODBC-Administrator.
Führen Sie den folgenden Befehl aus:

```
odbcad32
```

- b. Entfernen Sie die Datenquelle **mlxml_db**.
4. Löschen Sie die konsolidierten und entfernten Datenbanken.
 - a. Wechseln Sie zum Verzeichnis mit Ihren konsolidierten und entfernten Datenbanken.
 - b. Löschen Sie *MLconsolidated.db*, *MLconsolidated.log*, *remote1.db* und *remote1.log*.

Ergebnisse

Die Daten der praktischen Einführung werden von Ihrem Computer entfernt.

Nächste Schritte

Keine.

Praktische Einführung: Zentrale Administration von entfernten Datenbanken

Diese praktische Einführung führt Sie durch den Prozess der Einrichtung einer zentralen Administration entfernter Datenbanken und zeigt, wie Sie einige allgemeine Vorgänge durchführen.

Sie können diese praktische Einführung durcharbeiten, um die zentrale Administration vollständig neu einzurichten oder einem bestehenden Synchronisationssystem eine zentrale Administration hinzuzufügen. In der praktischen Einführung wird die Prozedur erläutert und festgehalten, an welcher Stelle Sie die zentrale Administration einem bestehenden Synchronisationssystem hinzufügen können.

Es stehen mehrere einführende Videos zur zentralen Administration entfernter Datenbanken online zur Verfügung. Weitere Hinweise finden Sie unter <http://www.sybase.com/detail?id=1081142>.

Hinweis

Die Einführungsvideos basieren auf Version 12.0.0 von SQL Anywhere. Einige Darstellungen und Prozeduren unterscheiden sich möglicherweise von SQL Anywhere 16.0.

Erforderliche Software

In dieser praktischen Einführung wird davon ausgegangen, dass SQL Anywhere (einschließlich MobiLink und Sybase Central) auf Ihrem lokalen Computer, auf dem Sie diese praktische Einführung ausführen, vollständig installiert ist.

Weitere Hinweise zum Deployment des MobiLink-Agenten finden Sie unter „Bereitstellung von SQL Anywhere MobiLink-Clients“ [[MobiLink - Serveradministration](#)] und „Deployment des MobiLink-Clients für UltraLite“ [[MobiLink - Serveradministration](#)].

Privilegien

Sie müssen die folgenden Rollen und Privilegien für die konsolidierte Datenbank haben:

- SYS_AUTH_RESOURCE_ROLE-Kompatibilitätsrolle

- MONITOR-Systemprivileg

Sie müssen die folgenden Rollen und Privilegien für die entfernte Datenbank haben:

- SYS_REPLICATION_ADMIN_ROLE-Systemrolle
- SYS_RUN_REPLICATION_ROLE-Systemrolle

Übersicht

In dieser praktischen Einführung wird Folgendes veranschaulicht:

- Eine konsolidierte Datenbank und ein MobiLink-Projekt erstellen
- Den MobiLink-Server starten, einen MobiLink-Benutzer und einen Agenten definieren und den Agenten auf dem entfernten Gerät konfigurieren
- Ein Synchronisationsmodell erstellen und es per Deployment bereitstellen
- Mit entfernten Aufgaben arbeiten

Siehe auch

- „Zentrale Administration von entfernten Datenbanken“ [[MobiLink - Serveradministration](#)]

Lektion 1: Erstellen einer konsolidierten Datenbank

In dieser Lektion richten Sie eine konsolidierte Datenbank ein. Wenn Sie ein vorhandenes Synchronisationssystem haben, gehen Sie weiter zu „[Lektion 2: Erstellen des MobiLink-Projekts](#)“ auf Seite 243.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Aufgabe

1. Führen Sie die folgenden Befehle aus, um Verzeichnisse zu erstellen, die in dieser praktischen Einführung verwendet werden. Das Verzeichnis der konsolidierten Datenbank enthält alle Datenbankdateien und anderen Dateien, die auf dem zentralen Server liegen.

```
md c:\cadmin_demo
md c:\cadmin_demo\consolidated
```

2. Erstellen Sie eine konsolidierte SQL Anywhere-Datenbank und eine ODBC-Datenquelle, um sich mit ihr zu verbinden.

```
cd c:\cadmin_demo\consolidated
dbinit -dba DBA,sql consol.db
start dbsrv16 consol.db
```



```
dbdsn -w cadmin_tutorial_consol consol -y -c  
"UID=DBA;PWD=sql;DBF=consol.db;SERVER=consol"  
cd ..
```

3. Verbinden Sie sich über Interactive SQL mit einer Datenbank. Führen Sie den folgenden Befehl aus:

```
dbisql -c "DSN=cadmin_tutorial_consol"
```

4. Führen Sie die folgenden Anweisung in Interactive SQL aus, um mit dem Skript *syncsa.sql* MobiLink-Systemtabellen und gespeicherte Prozeduren zu erstellen. Ersetzen Sie *C:\Programme\SQL Anywhere 16* durch den Speicherort Ihrer SQL Anywhere 16-Installation.

```
READ "C:\Program Files\SQL Anywhere 16\MobiLink\setup\syncsa.sql";
```

5. Schließen Sie **Interactive SQL**. Sie müssen Ihre SQL-Anweisungen nicht speichern.

Ergebnisse

Eine SQL Anywhere-Datenbank wird erstellt und Sie sind mit ihr verbunden.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 2: Erstellen des MobiLink-Projekts](#)“ auf Seite 243.

Lektion 2: Erstellen des MobiLink-Projekts

Wenn Sie eine zentrale Administration einrichten wollen, müssen Sie ein MobiLink-Projekt erstellen. Das Projekt ist ein Behälter für die verschiedenen Objekte, die Sie für die zentrale Administration definieren.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Aufgabe

1. Klicken Sie zum Starten von Sybase Central auf **Start » Programme » SQL Anywhere 16 » Administrationstools » Sybase Central**.
2. Klicken Sie auf **Extras » MobiLink 16 » Neues Projekt**.

Der **Assistent zum Erstellen eines Projekts** wird angezeigt.

3. Ändern Sie auf der Seite **Willkommen** den Projektnamen auf **Central Admin Tutorial** und geben Sie *c:\cadmin* als Pfad für die Projektdatei ein. Klicken Sie auf **Weiter**.
4. Geben Sie auf der Seite **Konsolidierte Datenbank angeben** den Wert **Tutorial** als **Anzeigename der Datenbank** ein.

5. Geben Sie die folgenden Werte für die **Verbindungszeichenfolge** ein:

`UID=DBA;PWD=sql;DSN=cadmin_tutorial_consol`

6. Wählen Sie **Kennwort speichern** und klicken Sie auf **Weiter**.

Unter Umständen erhalten Sie eine Warnung, dass kein Synchronisationsmodell erstellt wird, weil die Datenbank keine Tabellen enthält. Klicken Sie auf **OK**.

7. Wählen Sie **Dem Projekt einen Namen für ein entferntes Schema hinzufügen** und Typ **Tutorial Application v1.0** für den Schemanamen. Klicken Sie auf **Weiter**.

8. Wählen Sie **SQL Anywhere** als Datenbanktyp aus und klicken Sie auf **Fertig stellen**.

Wenn die konsolidierte Datenbank zum ersten Mal von MobiLink verwendet wird, werden Sie aufgefordert, die MobiLink-Systemeinrichtung zu installieren. Durch die Installation der MobiLink-Systemeinrichtung werden MobiLink-Systemtabellen und -Prozeduren hinzugefügt. Klicken Sie auf **Ja** und dann auf **OK**.

Ergebnisse

Das MobiLink-Projekt wird erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 3: Starten des MobiLink-Servers](#)“ auf Seite 244.

Lektion 3: Starten des MobiLink-Servers

In dieser Lektion starten Sie den MobiLink-Server. Sie brauchen den MobiLink-Server sowohl für die Synchronisation der Daten von Ihrer entfernten Datenbank als auch zur Synchronisation von Aufgaben und Aufgabenergebnissen zwischen der konsolidierten Datenbank und jedem entfernten Gerät.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Kontext und Bemerkungen

Wenn Sie bereits ein Synchronisationssystem haben, können Sie diese Lektion überspringen, da der Server bereits läuft. Sie sollten aber die Befehlszeile Ihres Serveraufrufs überprüfen und sicherstellen, dass die Optionen `-ftr` und `-ftru` festgelegt sind. Diese Optionen sind erforderlich, um Dateien auf entfernte Geräte herunterzuladen und Dateien von entfernten Geräten hochzuladen.

Aufgabe

- Führen Sie an der Eingabeaufforderung folgende Befehle aus:

```
md c:\cadmin_demo\consolidated\upload
md c:\cadmin_demo\consolidated\download
cd c:\cadmin_demo\consolidated
start mlsrv16.exe -c "DSN=cadmin_tutorial_consol;UID=DBA;PWD=sql" -ftr
download -ftru upload -x tcpip(port=2439) -v+ -ot mlsrv.txt
cd ..
```

Es folgt eine Übersicht über die verwendeten Optionen:

- **-c** Gibt die Verbindungsparameter an, die MobiLink verwendet, um eine Verbindung mit der konsolidierten Datenbank herzustellen.
- **-ftr** Legt das Verzeichnis fest, in dem MobiLink nach Dateien für den Download sucht.
- **-ftru** Legt das Verzeichnis fest, in dem MobiLink die hochgeladenen Dateien ablegt.
- **-x** Legt die Kommunikationsparameter fest, die definieren, wie die Synchronisationsclients eine Verbindung mit dem MobiLink-Server aufnehmen.
- **-v+** Legt die maximale Ausführlichkeitsstufe fest. Diese Einstellung ist bei der Fehlersuche nützlich, kann aber die Leistung in einer Produktionsumgebung reduzieren.
- **-ot** Legt die Datei fest, in der MobiLink-Ausgabemeldungen protokolliert werden.

Ergebnisse

Dieser MobiLink-Server und die Upload- und Downloadverzeichnisse werden für die Dateien erstellt, die aus entfernten Geräten hochgeladen oder in diese Geräte heruntergeladen werden.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 4: Definieren eines MobiLink-Benutzers“](#) auf Seite 245.

Lektion 4: Definieren eines MobiLink-Benutzers

In dieser Lektion definieren Sie einen MobiLink-Benutzer, den der Agent verwenden soll. Sie können diese Lektion überspringen, wenn Sie über ein bestehendes Synchronisationssystem verfügen und möchten, dass der MobiLink-Agent einen Ihrer bestehenden MobiLink-Benutzer zur Synchronisation verwendet.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Erstellen einer konsolidierten Datenbank“](#) auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Zentrale Administration von entfernten Datenbanken“](#) auf Seite 241.

Kontext und Bemerkungen

Wenn ein Agent seine Agentendatenbank synchronisiert, muss er sich am MobiLink-Server authentifizieren. Seine Authentifizierung erfolgt selbsttätig durch Übergabe des MobiLink-Benutzernamens und eines Kennworts. Normalerweise verwenden Sie denselben MobiLink-Benutzer und dasselbe Kennwort zur Synchronisation Ihrer entfernten Datenbanken, die der Agent verwendet, um die Agentendatenbank zu synchronisieren.

Aufgabe

1. Klicken Sie in Sybase Central auf **Ansicht » Ordner**.
2. Erweitern Sie unter MobiLink 16 **Central Admin Tutorial, Konsolidierte Datenbanken, Tutorial**.
3. Rechtsklicken Sie auf **Benutzer** und klicken Sie auf **Neu » Benutzer**.

Der **Assistent zum Erstellen von Benutzern** wird angezeigt.

4. Geben Sie auf der **Willkommenseite** als Namen des neuen Besitzers **JOHN** ein und klicken Sie auf **Weiter**
5. Aktivieren Sie auf der Seite **Authentifizierung** die Option **Dieser Benutzer benötigt ein Verbindungskennwort, wenn er die Standard-MobiLink-Authentifizierung verwendet** und geben Sie **sql** in die Felder **Kennwort** und **Kennwort bestätigen** ein. Klicken Sie auf **Fertig stellen**.

Wenn Sie Agenten, die eine Synchronisation versuchen wollen, nicht authentifizieren möchten, überspringen Sie diesen Schritt und fügen Sie die -zu+-Option in die MobiLink-Server-Befehlszeile ein. Bei der -zu+-Option wird jeder MobiLink-Benutzer beim ersten Synchronisationsversuch registriert. Siehe „[mlsrv16-Option -zu](#)“ [[MobiLink - Serveradministration](#)].

Ergebnisse

Der MobiLink-Benutzer wird erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 5: Definieren eines Agenten](#)“ auf Seite 246.

Lektion 5: Definieren eines Agenten

In dieser Lektion definieren Sie einen Agenten. Dieser Agent repräsentiert eine Instanz des MobiLink-Agenten, der auf einem entfernten Gerät läuft.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Kontext und Bemerkungen

Sie müssen einen separaten Agenten für die einzelnen entfernten Geräte erstellen, die Sie verwalten.

Aufgabe

1. Erweitern Sie unter MobiLink 16 **Central Admin Tutorial, Konsolidierte Datenbanken, Tutorial**.
2. Rechtsklicken Sie auf **Agenten** und klicken Sie auf **Neu » Agent**.

Der **Assistent zum Erstellen von MobiLink-Agenten** wird angezeigt.

3. Wählen Sie auf der Seite **Willkommen** die Option **Einzelnen Agenten erstellen** und klicken Sie auf **Weiter**.
4. Geben Sie auf der Seite **Agenten-ID** **AID_JOHN** für die **Agenten-ID** ein. Die Agenten-ID kann jeden beliebigen Wert haben, jeder Agent muss aber eine eindeutige ID haben. Agent-IDs beginnen im Allgemeinen mit dem Präfix **AID_** und der zweite Teil der Agenten-ID ist in der Regel der MobiLink-Benutzername, der vom Agenten verwendet wird. Optional können Sie eine Beschreibung für den Agenten im Feld **Beschreibung** eingeben. Klicken Sie auf **Weiter**.
5. Auf der Seite **Entfernte Datenbank** können Sie eine entfernte Datenbank definieren, die von diesem Agenten verwaltet wird. Damit wird die Datenbank noch nicht erstellt. Dies erfolgt später. Für **Name des entfernten Schemas** wählen Sie **Tutorial Application v1.0**, also den Namen, den Sie in der vorherigen Lektion definiert haben, aus der Dropdown-Liste.
6. Geben Sie die nachstehende Zeichenfolge im Feld **Verbindungszeichenfolge für die Datenbank** ein.

```
start=dbsrv16;SERVER=tutorial_v1;DBF={db_location}  
\tutorial_v1.db;UID=DBA;PWD=sql
```

Dieser Zeichenfolgenwert verwendet den Makro {db_location}. Dieser Makro wird durch das Verzeichnis auf dem entfernten Gerät ersetzt, in dem die Anwendungsdatenbanken gespeichert sind. Klicken Sie auf **Weiter**.

7. Geben Sie auf der Seite **Agentenkonfiguration** den Wert **30** ein und wählen Sie **Sekunden** für das **Synchronisationsintervall**. Das Synchronisationsintervall steuert, wie häufig der Agent seine Agentendatenbank synchronisiert. Durch die Synchronisation der Agentendatenbank empfängt der Agent neue Aufgaben und lädt die Ergebnisse der Aufgaben hoch, die er bereits durchgeführt hat.
8. Geben Sie auf der Seite **Agentenkonfiguration** den Wert **10** ein und wählen Sie **Sekunden** für das **Administrations-Abrufintervall**. Das Administrations-Abrufintervall legt fest, wie häufig der Agent nachsieht, ob Anforderungen vom Server eingetroffen sind, eine Synchronisation oder andere Aktionen durchzuführen.

Hinweis

Wenn für das Synchronisationsintervall und das Administrations-Abrufintervall kurze Zeitspannen gewählt werden, reagiert der Agent sehr schnell. Dies ist etwa bei Demos oder bei der Fehlerbehandlung besonders wichtig. Wenn Sie hingegen diese kurzen Zeitspannen in einem Produktionssystem verwenden, wird die Verarbeitungslast auf Ihrem Server stark erhöht und dies reduziert die Performance.

9. Klicken Sie auf **Fertig stellen**.

Ergebnisse

Der Agent wird erstellt und konfiguriert.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 6: Konfigurieren des Agenten auf dem entfernten Gerät](#)“ auf Seite 248.

Lektion 6: Konfigurieren des Agenten auf dem entfernten Gerät

In dieser Lektion führen Sie den MobiLink-Agenten aus. Der MobiLink-Agent muss auf jedem entfernten Gerät laufen, das zentral administriert wird. Für diese praktische Einführung wird der Agent auf dem Computer ausgeführt, auf dem auch der MobiLink-Server läuft.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Aufgabe

1. Erstellen Sie ein Verzeichnis, das die Dateien enthält, die sich normalerweise auf dem entfernten Gerät befinden.

```
md c:\cadmin_demo\remote
cd c:\cadmin_demo\remote
```

2. Führen Sie den MobiLink-Agenten im Konfigurationsmodus wie folgt aus:

```
mlagent -cr -db . -x tcpip{host=localhost;port=2439} -a AID_JOHN -u JOHN
-p sql
```

Bei diesem Schritt wird eine Agentendatenbank erstellt. Darin werden bestimmte Konfigurationsinformationen gespeichert. Wenn die angegebenen Optionen in der Datenbank gespeichert sind, fährt der Agent herunter. Es folgt eine Übersicht der verwendeten Optionen:

- **-cr** Legt fest, dass der Agent im Konfigurationsmodus ausgeführt werden und alle Einstellungen verwerfen soll, die während früherer Instanzen im Konfigurationsmodus gespeichert wurden.
 - **-db** Legt fest, wo der Agent Anwendungsdatenbanken erstellen soll. Dies wird der Wert für den {db_location}-Makro.
 - **-x** Legt fest, wie sich der Agent mit dem MobiLink-Server verbinden soll, um seine Agentendatenbank zu synchronisieren (zur Entgegennahme neuer Aufgaben und Übermittlung der Ergebnisse der bereits durchgeführten). Wenn Sie einem bestehenden Synchronisationssystem die zentrale Administration hinzufügen, müssen Sie den für diese Option angegebenen Wert auf eine geeignete Zeichenfolge für die Verbindung mit Ihrem MobiLink-Server ändern.
 - **-a** Legt die Agenten-ID für diesen Agenten fest. Sie haben dieselbe Agenten-ID eingegeben, die Sie vorher in der konsolidierten Datenbank in Sybase Central erstellt haben.
 - **-u** Legt den MobiLink-Benutzer fest, den der Agent verwendet, wenn die Agentendatenbank synchronisiert wird. Dieser Wert wird vom MobiLink-Server hauptsächlich zur Authentifizierung des Agenten verwendet.
 - **-p** Legt das Kennwort fest, das zu dem MobiLink-Benutzer gehört, der mit der -u-Option festgelegt wurde.
3. Starten Sie den MobiLink-Agenten auf dem entfernten Gerät. Für diese praktische Einführung starten Sie den Agenten explizit wie folgt:

```
start mlagent -v9 -ot agent.txt
```

Es folgt eine Übersicht über die verwendeten Optionen zum Ausführen des Agenten in dieser Lektion:

- **-v9** Verwendet die maximale Ausführlichkeitsstufe. Die Verwendung dieser Protokollierungsoptionen empfiehlt sich in einer Entwicklungsumgebung. Aus Gründen der Performance wird -v in einer Produktionsumgebung normalerweise nicht eingesetzt.
 - **-ot** Legt die Datei fest, in die die Protokolle des Agenten geschrieben werden.
4. Der MobiLink-Agent läuft jetzt und die Synchronisation sollte nun erfolgreich durchgeführt werden. Um dies zu überprüfen, kehren Sie zu Sybase Central zurück. Erweitern Sie in der Ansicht **Ordner** unter **MobiLink 16 Central Admin Tutorial » Konsolidierte Datenbanken » Tutorial » Agenten**. Wählen Sie **AID_JOHN** und sehen Sie sich die Registerkarte **Ereignisse** im rechten Fensterausschnitt an. Sie sehen jetzt einen Eintrag, der die erste Synchronisation des Agenten anzeigt.

Hinweis

Hinweise zur Agentenkonfiguration in Produktionsumgebungen Beachten Sie Folgendes, wenn Sie die zentrale Administration in einer Produktionsumgebung verwenden:

- Sie müssen eventuell die Werte für die Optionen -u und -p ändern, damit eine geeignete Kombination aus MobiLink-Benutzer und Kennwort für Ihr Synchronisationssystem verwendet wird.
- Die -on-Option wird empfohlen, um die Größe der vom Agenten erstellten Logdatei zu begrenzen.
- Ein entferntes Gerät kann nur entfernt administriert werden, wenn der MobiLink-Agent darauf läuft. Daher sollten Sie sicherstellen, dass der Agent immer läuft. Dazu können Sie beispielsweise den Agenten als Service ausführen oder in der Registrierung als Autostartprogramm definieren.

Ergebnisse

Der MobiLink-Agent wird ausgeführt und führt die Synchronisation durch.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 7: Erstellen eines Synchronisationsmodells“ auf Seite 250](#).

Lektion 7: Erstellen eines Synchronisationsmodells

In dieser Lektion erstellen Sie ein Synchronisationsmodell.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Erstellen einer konsolidierten Datenbank“ auf Seite 242](#).

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Zentrale Administration von entfernten Datenbanken“ auf Seite 241](#).

Kontext und Bemerkungen

Wenn Sie die zentrale Administration einem bestehenden Synchronisationssystem hinzufügen, gehen Sie weiter zu [„Lektion 8: Erstellen eines Synchronisationsmodells“ auf Seite 251](#).

Aufgabe

1. Definieren Sie die Tabellen für die entfernte Datenbank in der konsolidierten Datenbank. Erweitern Sie in der Ansicht **Ordner** von Sybase Central unter **MobiLink 16** den Eintrag **Central Admin Tutorial, Konsolidierte Datenbanken**. Rechtsklicken Sie auf **Tutorial - DBA** und klicken Sie auf **Interactive SQL öffnen**.
2. Geben Sie im Fensterausschnitt **SQL-Anweisungen** folgende Anweisungen ein:


```
CREATE TABLE customer(  
    cust_id      INTEGER PRIMARY KEY,  
    f_name       VARCHAR(100),  
    l_name       VARCHAR(100)  
)
```

3. Drücken Sie F5, um die SQL-Anweisung auszuführen. Schließen Sie Interactive SQL. Sie müssen Ihre SQL-Anweisungen nicht speichern.
4. Klicken Sie mit der rechten Maustaste in der Ansicht **Ordner** von Sybase Central auf **Central Admin Tutorial » Neu » Synchronisationsmodell**.
5. Geben Sie auf der **Willkommenseite** als Namen des neuen Synchronisationsmodells **tutorial1** ein. Klicken Sie auf **Weiter**.
6. Aktivieren Sie auf der Seite **Primärschlüssel-Anforderungen** alle drei Kontrollkästchen, um zu bestätigen, dass Ihr Schema die Anforderungen für Synchronisationen erfüllt. Klicken Sie auf **Weiter**.
7. Wählen Sie auf der Seite **Schema der konsolidierten Datenbank** auf die **Tutorial**-Datenbank und klicken Sie auf **Weiter**.
8. Wählen Sie auf der Seite **Schema der entfernten Datenbank** die Option **Nein, entferntes Datenbankschema erstellen** und klicken Sie dann auf **Weiter**.
9. Auf der Seite **Neues entferntes Datenbankschema** vergewissern Sie sich, dass die Customer-Tabelle aktiviert ist, und klicken Sie auf **Weiter**. Klicken Sie auf **Fertig stellen**.

Ergebnisse

Sie haben jetzt ein Synchronisationsmodell erstellt, das eine einzelne Tabelle namens **customer** enthält, die zwischen der entfernten und der konsolidierten Datenbank synchronisiert werden kann. Der nächste Schritt besteht darin, ein Deployment dieses Modells vorzunehmen, um Synchronisationsobjekte in der konsolidierten Datenbank zu erstellen und SQL-Anweisungen zur Erstellung einer entfernten Datenbank zu generieren.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 8: Erstellen eines Synchronisationsmodells](#)“ auf Seite 251.

Lektion 8: Erstellen eines Synchronisationsmodells

In dieser Lektion nehmen Sie das Deployment des Synchronisationsmodells vor, das Sie in der vorherigen Lektion erstellt haben.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Aufgabe

1. Erweitern Sie in der Ansicht **Ordner** von Sybase Central unter **MobiLink 16** den Eintrag **Central Admin Tutorial » Synchronisationsmodelle**. Klicken Sie mit der rechten Maustaste auf **tutorial1** und klicken Sie auf **Deployment**.

Der **Assistent zum Deployment eines Synchronisationsmodells** wird geöffnet.

2. Auf der Seite **Willkommen** akzeptieren Sie den Standardspeicherort für die Dateien, die vom Assistenten erstellt wurden, und klicken Sie auf **Weiter**.
3. Wählen Sie auf der Seite **Client-Netzwerkoptionen** die folgenden Optionen aus und klicken Sie auf **Weiter**.

- **Protokoll** TCP/IP
- **Host** localhost
- **Port** 2439

4. Wählen Sie auf der Seite **MobiLink-Benutzer und -Kennwort** die Option **Für entfernte Aufgaben geeignete Makrowerte verwenden**:

Die Makrowerte **{ml_username}** und **{ml_password}** werden in den generierten SQL-Dateien verwendet und durch den MobiLink-Benutzer und das Kennwort ersetzt, die vom MobiLink-Agent verwendet werden, wenn die SQL-Anweisungen auf dem entfernten Gerät ausgeführt werden. Ein Synchronisationsprofil wird automatisch mit dem Namen **tutorial1_{ml_username}** erstellt, wobei der Makro **{ml_username}** durch den Namen des MobiLink-Benutzers (in diesem Fall JOHN) ersetzt wird.

5. Auf der Seite **Synchronisationsprofil** geben Sie **tutorial1_JOHN** in das Feld **Synchronisationsprofilname** ein.
6. Klicken Sie auf **Weiter**, bis Sie zur Seite **Auswählen, wie Datenbanken für die Synchronisation vorbereitet werden** kommen und führen Sie dort die folgenden Aufgaben aus:

- Für **Was möchten Sie mit dem SQL-Skript tun, das erstellt wird, um die konsolidierte Datenbank für die Synchronisation vorzubereiten?** wählen Sie **Mit der konsolidierten Datenbank ausführen**.
- Für **Was möchten Sie mit dem SQL-Skript tun, das erstellt wird, um die entfernte Datenbank für die Synchronisation vorzubereiten?** wählen Sie **Nicht ausführen**.
- Klicken Sie auf **Weiter** und auf **Fertig stellen**.

Wenn Sie vom Synchronisationsmodell wegnavigieren, werden Sie aufgefordert, Ihre Änderungen zu speichern. Klicken Sie auf **Ja**.

Ergebnisse

Sie haben jetzt die Erstellung und das Deployment eines Synchronisationsmodells abgeschlossen. Als Sie das Deployment des Modells vorgenommen haben, wurden Skripten der konsolidierten Datenbank hinzugefügt, damit entfernte Datenbanken synchronisieren können. Sie haben auch eine SQL-Datei im c:\admin_demo\Central Admin Tutorial\tutorial1_deploy\ -Verzeichnis erstellt, die verwendet werden kann, um eine entfernte Datenbank zu erstellen. Sie können sich diese Dateien jetzt ansehen.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 9: Erstellen einer entfernten Aufgabe](#)“ auf Seite 253.

Lektion 9: Erstellen einer entfernten Aufgabe

In dieser Lektion erstellen wir eine entfernte Aufgabe, um die Nachricht "Hello World" auf dem entfernten Gerät anzuzeigen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Kontext und Bemerkungen

Die meisten Aktionen in der zentralen Administration werden mit entfernten Aufgaben durchgeführt. Eine entfernte Aufgabe ist eine Sammlung von Befehlen, die von einem Administrator erstellt wird. Sie kann einem oder mehr Agenten zugeordnet werden. Wenn sie einmal einem Agenten zugewiesen ist, wird die entfernte Aufgabe per Download zum Agenten geschickt, wenn der Agent seine Agentendatenbank das nächste Mal synchronisiert. Der Agent führt dann die Aufgabe zum geeigneten Zeitpunkt aus und übersendet die Informationen über die Ausführung per Upload.

Aufgabe

1. Erstellen Sie eine neue entfernte Aufgabe. Erweitern Sie in der Ansicht **Ordner** von Sybase Central unter **MobiLink 16** den Eintrag **Central Admin Tutorial**, klicken Sie mit der rechten Maustaste auf **Entfernte Aufgaben** und klicken Sie auf **Neu » Entfernte Aufgabe**.

Der **Assistent für die Erstellung einer entfernten Aufgabe** wird angezeigt.

2. Geben Sie auf der **Willkommenseite** den Wert **Hello World** im Feld **Name** ein. Klicken Sie auf **Weiter**.
3. Aktivieren Sie auf der Seite **Auslösemechanismen** das Kontrollkästchen **Beim Empfang von einem Agenten** und klicken Sie auf **Fertig stellen**, um den Assistenten abzuschließen.

4. Klicken Sie auf die neu erstellte **Hello World**-Aufgabe in der Ansicht **Ordner**. Im rechten Fensterausschnitt sehen Sie die Registerkarte **Befehle**, auf der Sie Ihrer Aufgabe Befehle hinzufügen können.
5. Auf der Registerkarte **Befehle** wählen Sie **Bedieneraufforderung** aus der Dropdown-Liste **Befehlstyp**. Im Feld **Nachricht** geben Sie **Hello World** ein.
6. Um einen zweiten Befehl zur Aufgabe hinzuzufügen, drücken Sie entweder zweimal die Tabulatortaste, bis ein neuer Befehl angezeigt wird, oder klicken Sie auf die Schaltfläche **Befehl hinzufügen**. Legen Sie den Befehlstyp für den zweiten Befehl mit **Eingabeaufforderung** fest und geben Sie **Hello Again** in das Feld **Nachricht** ein.

Die gerade erstellte Hello World-Aufgabe ist eine Planungszeitaufgabe. Sie wird im Projekt auf Ihrem lokalen Computer gespeichert. Bevor Sie die Aufgabe einem Agenten zuweisen können, müssen Sie sie in die konsolidierte Datenbank kopieren, indem Sie ein Deployment vornehmen.

Ergebnisse

Die entfernte Aufgabe wurde erstellt und ist bereit für das Deployment.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 10: Deployment einer entfernten Aufgabe](#)“ auf Seite 254.

Lektion 10: Deployment einer entfernten Aufgabe

In dieser Lektion nehmen Sie ein Deployment der entfernten Aufgabe vor, damit sie einem Agenten zugewiesen werden kann.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Aufgabe

1. Erweitern Sie in der Ansicht **Ordner** von Sybase Central unter **MobiLink 16** die Elemente **Central Admin Tutorial**, **Entfernte Aufgaben**, rechtsklicken Sie auf die Aufgabe **Hello World** und klicken Sie auf **Deployment**.

Der **Assistent für das Deployment einer entfernten Aufgabe** wird angezeigt.

2. Akzeptieren Sie die Standardwerte auf der Seite **Aufgabenname und Ziel** und klicken Sie auf **Weiter**.

Damit erhält die bereitgestellte Aufgabe denselben Namen wie die Planungszeitaufgabe. Das ist in der Regel beabsichtigt, außer wenn Sie ein zweites Deployment derselben Planungszeitaufgabe vornehmen. In diesem Fall müssen Sie den Namen der bereitgestellten Aufgabe ändern.

3. Auf der Seite **Empfänger** weisen Sie die bereitgestellte Aufgabe bestehenden Agenten zu. Sie können dies auch später in einem eigenen Schritt tun. Wählen Sie in der Dropdown-Liste **Empfänger** die Option **Spezifische Agenten**. Wählen Sie in der Liste **Agent** den Eintrag **AID_JOHN** und klicken Sie auf **Weiter**.
4. Aktivieren Sie auf der Seite **Zustelloptionen** die Option **Bei der nächsten Synchronisation des Agenten** und klicken Sie auf **Weiter**.
5. Aktivieren Sie auf der Seite **Bericht über Ergebnisse und Status** für beide Fragen das Kontrollkästchen **Ergebnisse und Status sofort senden**. Damit stellen Sie sicher, dass Sie zeitnahe Ausführungsinformationen erhalten, wenn Ihre Aufgabe ausgeführt wird. Für Routineaufgaben und laufend wiederholte Aufgaben können Sie die Rückmeldung in längeren Abständen definieren (vor allem wenn sie erfolgreich verlaufen), da damit die Häufigkeit der Synchronisationen der Agentendatenbank und die Verarbeitungslast auf dem MobiLink-Server reduziert werden.
6. Klicken Sie auf **Fertig stellen**.

Wenn der **AID_JOHN**-Agent seine Agentendatenbank das nächste Mal synchronisiert, erhält er die Aufgabe und führt sie aus. Klicken Sie **OK** auf den Meldungsfeldern mit dem Text **Hello World** und **Hello Again**.

In der Ansicht **Ordner** können Sie sehen, dass nun zwei Kopien der **Hello World**-Aufgabe in der Liste stehen. Die bereitgestellte Kopie kann in der Ansicht **Ordner** unter **Entfernte Aufgaben** » **Bereitgestellte Aufgaben** eingesehen werden. Hierbei handelt es sich um die Kopie in der konsolidierten Datenbank. Die bereitgestellte Kopie der Aufgabe kann nicht mehr geändert werden. Die Planungszeitkopie der Aufgabe ist unter **Entfernte Aufgaben** weiterhin sichtbar. Diese Aufgabe kann geändert und erneut mit einem anderen Namen bereitgestellt werden.

Sie können eine bereitgestellte Aufgabe weiteren Agenten zuweisen, indem Sie mit der rechten Maustaste darauf klicken und **Empfänger hinzufügen** auswählen.

Ergebnisse

Die entfernte Aufgabe wird ausgeführt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 11: Überprüfen des Status einer entfernten Aufgabe](#)“ auf Seite 255.

Lektion 11: Überprüfen des Status einer entfernten Aufgabe

In dieser Lektion überprüfen Sie den Status Ihrer entfernten Aufgabe in Sybase Central.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Aufgabe

1. Erweitern Sie in der Ansicht **Ordner** von Sybase Central unter **MobiLink 16** den Eintrag **Central Admin Tutorial » Entfernte Aufgaben » Bereitgestellte Aufgaben**. Klicken Sie auf die bereitgestellte Version der **Hello World**-Aufgabe und wählen Sie die Registerkarte **Ergebnisse** im rechten Fensterausschnitt. Warten Sie, bis die Registerkarte automatisch aktualisiert wird, oder drücken Sie F5, um sofort zu aktualisieren. Auf der Registerkarte **Ergebnisse** befindet sich eine Zeile für jeden Befehl in der Aufgabe. Ein **Ergebniscode** zeigt an, ob der Befehl erfolgreich verlaufen oder fehlgeschlagen ist. Ein **Ergebniscode** 0 bedeutet Erfolg.
2. Um die Ergebnisse der Ausführung einer Aufgabe auf verschiedene Arten anzuzeigen, wählen Sie die Registerkarte **Empfänger** für die bereitgestellte Aufgabe oder ziehen die Registerkarte **Ereignisse** bzw. **Aufgaben** des Agenten zu Rate, der die Aufgabe ausgeführt hat.

Ergebnisse

Die Ergebnisse der Aufgabenausführung werden angezeigt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 12: Erstellen einer entfernten Datenbank auf einem entfernten Gerät](#)“ auf Seite 256.

Lektion 12: Erstellen einer entfernten Datenbank auf einem entfernten Gerät

In dieser Lektion verwenden Sie eine entfernte Aufgabe zur Erstellung einer neuen entfernten Datenbank auf dem entfernten Gerät. Wenn Sie die zentrale Verwaltung einem bestehenden Synchronisationssystem hinzufügen, gehen Sie weiter zu „[Lektion 13: Zeitplanung für die Synchronisation](#)“ auf Seite 258.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Aufgabe

1. Erstellen Sie eine neue entfernte Aufgabe. Erweitern Sie in der Ansicht **Ordner** von Sybase Central unter **MobiLink 16** das Element **Central Admin Tutorial**. Klicken Sie mit der rechten Maustaste auf **Entfernte Aufgaben** und klicken Sie auf **Neu » Entfernte Aufgabe**.

Der **Assistent für die Erstellung einer entfernten Aufgabe** wird angezeigt.

2. Geben Sie auf der **Willkommenseite** den Wert **Create DB** im Feld **Name** ein. Anders als die vorher erstellte Aufgabe erstellt oder führt diese Aufgabe Handlungen in einer entfernten Datenbank aus. Daher müssen Sie **Diese Aufgabe benötigt oder erstellt eine entfernte Datenbank** aktivieren und den Namen für entfernte Schemata **Tutorial Application v1.0** wählen. Damit wird die entfernte Datenbank bezeichnet, in der die Datenbankaktionen dieser Aufgabe ablaufen werden. Klicken Sie auf **Weiter**.
3. Aktivieren Sie auf der Seite **Auslösemechanismen** das Kontrollkästchen **Beim Empfang von einem Agenten** und klicken Sie dann auf **Fertig stellen**, um den Assistenten abzuschließen.
4. Klicken Sie auf die neu erstellte **Create DB**-Aufgabe in der Ansicht **Ordner**.
5. Geben Sie einige Befehle in die entfernte Aufgabe ein und verwenden Sie dazu den Bereich **Befehle** auf der rechten Seite.
 - a. Der erste Befehl erstellt eine neue, leere Datenbank auf dem entfernten Gerät. Legen Sie den Befehlstyp mit **Datenbank erstellen** fest.
 - b. Legen Sie den Dateinamen mit **{db_location}\tutorial_v1.db** fest. Dieser Dateiname entspricht dem Dateinamen in der Verbindungszeichenfolge, die Sie bei der Konfiguration des Agenten eingegeben haben.
 - c. Drücken Sie einige Male die Tabulatortaste, bis ein neuer Befehl erscheint.
 - d. Der zweite Befehl erstellt das Schema in der neuen Datenbank. Legen Sie den Befehlstyp mit **SQL ausführen** fest. Klicken Sie auf **Importieren**.
 - e. Wählen Sie im Fenster **Öffnen** die Datei *c:\cadmin\Central Admin Tutorial\tutorial1_deploy\remote_setup.sql* und klicken Sie auf **Öffnen**. Damit wird der SQL-Code für die Initialisierung der entfernten Datenbank importiert, der generiert wurde, als Sie das Synchronisationsmodell in einer Befehlsdatei bereitgestellt haben.
6. Die entfernte Aufgabe ist damit abgeschlossen. Nehmen Sie das Deployment der Aufgabe vor und ordnen Sie sie dem Agenten **AID_JOHN** zu:
 - a. Klicken Sie mit der rechten Maustaste auf die **Create DB**-Aufgabe in der Ansicht **Ordner** und klicken Sie auf **Deployment**.

Der **Assistent für das Deployment einer entfernten Aufgabe** wird angezeigt.
 - b. Akzeptieren Sie die Standardwerte auf der Seite **Aufgabenname und Ziel** und klicken Sie auf **Weiter**.
 - c. Wählen Sie in der Dropdown-Liste **Empfänger** die Option **Spezifische Agenten**. Wählen Sie in der Liste **Agent** den Eintrag **AID_JOHN** und klicken Sie auf **Weiter**.
 - d. Aktivieren Sie auf der Seite **Zustellungsoptionen** die Option **Bei der nächsten Synchronisation des Agenten** und klicken Sie auf **Weiter**.

- e. Wählen Sie auf der Seite **Bericht über Ergebnisse und Status** für beide Fragen die Option **Ergebnisse und Status sofort senden**.
 - f. Klicken Sie auf **Fertig stellen**.
7. Prüfen Sie, ob die Aufgabe erfolgreich war:
- a. Erweitern Sie in der Ansicht **Ordner** von Sybase Central unter **MobiLink 16** die Elemente **Central Admin Tutorial**, **Konsolidierte Datenbanken**, **Tutorial**, **Agenten** und klicken Sie auf **AID_JOHN**.
 - b. Wählen Sie die Registerkarte **Ereignisse** und suchen Sie nach der **Create DB**-Aufgabe. Warten Sie, bis die Registerkarte automatisch aktualisiert wird oder drücken Sie F5, um sofort zu aktualisieren.

Ergebnisse

Die neue entfernte Datenbank wird auf dem entfernten Gerät erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 13: Zeitplanung für die Synchronisation](#)“ auf Seite 258.

Lektion 13: Zeitplanung für die Synchronisation

Der nächste Schritt besteht darin, den MobiLink-Agenten so zu konfigurieren, dass seine entfernte Datenbank in regelmäßigen Abständen synchronisiert wird. Dies erfolgt, indem eine entfernte Aufgabe erstellt wird, die aufgrund eines Zeitplans ausgeführt wird und die Datenbank jedes Mal synchronisiert, wenn sie ausgeführt wird.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Kontext und Bemerkungen

Diese Aufgabe unterscheidet sich von den anderen Aufgaben, die Sie erstellt haben, weil die anderen Aufgaben nur einmal ausgeführt werden. Diese Aufgabe bleibt auf dem entfernten Gerät und wird in regelmäßigen Abständen ausgeführt, bis Sie sie stoppen.

Aufgabe

1. Erstellen Sie eine neue entfernte Aufgabe. Erweitern Sie in der Ansicht **Ordner** von Sybase Central unter **MobiLink 16** das Element **Central Admin Tutorial**. Klicken Sie mit der rechten Maustaste auf **Entfernte Aufgaben** und klicken Sie auf **Neu » Entfernte Aufgabe**.

Der **Assistent für die Erstellung einer entfernten Aufgabe** wird angezeigt.

2. Geben Sie auf der **Willkommenseite** den Wert **Sync** im Feld **Name** ein. Aktivieren Sie die Option **Diese Aufgabe benötigt oder erstellt eine entfernte Datenbank** und wählen Sie den Namen für entfernte Schemata **Tutorial Application v1.0**. Damit wird die entfernte Datenbank bezeichnet, in der die Datenbankaktionen dieser Aufgabe ablaufen werden. Klicken Sie auf **Weiter**.
3. Wählen Sie auf der Seite **Auslösemechanismen** die Option **Basiert auf einem Zeitplan** und klicken Sie auf **Weiter**.
4. Akzeptieren Sie die Standardwerte auf der Seite **Startzeit und Startdatum**. Damit kann die Aufgabe sofort ausgeführt werden. Klicken Sie auf **Weiter**.
5. Aktivieren Sie auf der Seite **Wiederholung** das Kontrollkästchen **Wiederholung alle** und setzen Sie das Intervall auf eine Minute. Klicken Sie auf **Fertig stellen**, damit der Assistent abgeschlossen wird.
6. Klicken Sie auf die neu erstellte Aufgabe **Sync** in der Ansicht **Ordner**.
7. Fügen Sie der Aufgabe einen einzelnen Befehl hinzu, der die Synchronisation bewirkt.
 - a. Legen Sie auf der Seite **Befehle** den **Befehlstyp** für den ersten Befehl mit **Synchronisieren** fest.
 - b. Als **Synchronisationsprofil** geben Sie **tutorial1_JOHN** ein. Hierbei handelt es sich um das Synchronisationsprofil, das erstellt wurde, als Sie das Synchronisationsmodell bereitgestellt haben.
8. Die Synchronisationsaufgabe ist damit abgeschlossen. Klicken Sie mit der rechten Maustaste auf **Sync** und klicken Sie auf **Deployment**. Klicken Sie auf **Weiter**.
9. Klicken Sie in der Dropdown-Liste **Empfänger** auf **Spezifische Agenten** und weisen Sie die Aufgabe dem Agenten **AID_JOHN** zu. Klicken Sie auf **Weiter** und dann nochmals auf **Weiter**.
10. Setzen Sie auf der Seite **Bericht über Ergebnisse und Status** die Option **Bei Erfolg der Aufgabe auf Status später senden** und **Bei Fehlschlag der Aufgabe auf Ergebnisse und Status sofort senden**.

Da diese Aufgabe häufig wiederholt wird, empfiehlt es sich, die Rückmeldung einzuschränken, um die Performance nicht zu beeinträchtigen.
11. Klicken Sie auf **Fertig stellen**.

Ergebnisse

Wenn der Agent diese neue Aufgabe erhält, beginnt er, seine entfernte Datenbank in Minutenabständen zu synchronisieren.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 14: Ändern von geplanten Synchronisationen](#)“ auf Seite 260.

Lektion 14: Ändern von geplanten Synchronisationen

In der vorherigen Lektion haben Sie eine entfernte Aufgabe erstellt, um die entfernte Datenbank in Minutenabständen zu synchronisieren. In dieser Lektion ändern Sie das Synchronisationsintervall auf einmal pro Stunde.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Kontext und Bemerkungen

Wenn eine entfernte Aufgabe bereitgestellt wurde, kann die bereitgestellte Version nicht mehr geändert werden. Erstellen Sie stattdessen eine neue entfernte Aufgabe mit den gewünschten Änderungen, stornieren Sie die aktuelle Aufgabe und führen Sie ein Deployment der neuen Aufgabe durch, um die alte zu ersetzen.

Aufgabe

1. Zuerst erstellen Sie eine neue entfernte Aufgabe mit dem gewünschten Wiederholungsintervall unter Verwendung der bestehenden bereitgestellten Aufgabe als Vorlage.
 - a. Erweitern Sie in der Ansicht **Ordner** von Sybase Central unter **MobiLink 16** den Eintrag **Central Admin Tutorial » Entfernte Aufgaben » Bereitgestellte Aufgaben**. Rechtsklicken Sie auf **Sync** und wählen Sie **Kopieren**, um die Aufgabe in die Zwischenablage zu übernehmen.
 - b. Klicken Sie mit der rechten Maustaste auf **Entfernte Aufgaben** und wählen Sie **Einfügen**. In einem Fenster werden Sie ersucht, die entfernte Aufgabe umzubenennen. Geben Sie **Sync every hour** ein und klicken Sie auf **OK**.
 - c. Klicken Sie mit der rechten Maustaste auf die neue **Sync every hour**-Aufgabe und wählen Sie **Deployment**. Ändern Sie auf der Seite **Wiederholung** des Eigenschaftsfensters den Wert **Wiederholung alle** von **1 Minuten** in **1 Stunden** und klicken Sie auf **OK**.
2. Als nächstes stornieren Sie die bestehende entfernte Aufgabe, die die Synchronisation in Minutenabständen auslöst.
 - a. Klicken Sie in der Ansicht **Ordner** auf die bereitgestellte Version der **Sync**-Aufgabe und klicken Sie auf die Registerkarte **Empfänger** im rechten Fensterausschnitt.
 - b. Klicken Sie mit der rechten Maustaste in den Eintrag in der Tabelle für den **AID_JOHN**-Agenten und klicken Sie auf **Abbrechen**.
3. Zuletzt führen Sie das Deployment der **Sync every hour**-Aufgabe durch und weisen sie dem **AID_JOHN**-Agenten zu.
 - a. Klicken Sie mit der rechten Maustaste auf **Sync every hour** und klicken Sie auf **Deployment**. Klicken Sie auf **Weiter**.

- b. Klicken Sie in der Dropdown-Liste **Empfänger** auf **Spezifische Agenten** und weisen Sie die Aufgabe dem Agenten **AID_JOHN** zu. Klicken Sie auf **Weiter** und dann nochmals auf **Weiter**.
- c. Legen Sie auf der Seite **Bericht über Ergebnisse und Status** die Option **Bei Erfolg der Aufgabe** auf **Nur Status später senden** fest und **Bei Fehlschlag der Aufgabe** auf **Ergebnisse und Status sofort senden**.
- d. Klicken Sie auf **Fertig stellen**.

Ergebnisse

Wenn der Agent diese neue Aufgabe erhält, beginnt er, seine entfernte Datenbank in Stundenabständen und nicht mehr in Minutenabständen zu synchronisieren.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 15: Erzwingen einer sofortigen Synchronisation“ auf Seite 261](#).

Lektion 15: Erzwingen einer sofortigen Synchronisation

In der vorhergehenden Lektion haben Sie die entfernte Datenbank so eingerichtet, dass die Synchronisation einmal pro Stunde durchgeführt wird. Diese Lektion zeigt, wie Sie eine serverinitiierte entfernte Aufgabe (SIRT für "server-initiated remote task") verwenden, um eine Synchronisation zu erzwingen, bevor das Stundenintervall abgelaufen ist. Diese Technik ist nützlich, wenn Sie zentral steuern möchten, wann eine Aufgabe ausgeführt wird.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Erstellen einer konsolidierten Datenbank“ auf Seite 242](#).

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Zentrale Administration von entfernten Datenbanken“ auf Seite 241](#).

Aufgabe

- Erweitern Sie in der Ansicht **Ordner** von Sybase Central unter **MobiLink 16** den Eintrag **Central Admin Tutorial » Entfernte Aufgaben » Bereitgestellte Aufgaben**. Rechtsklicken Sie auf **Sync every hour** und klicken Sie **Für alle Empfänger einleiten**.

Ergebnisse

Alle Empfänger der Aufgabe werden angewiesen, die Aufgabe sofort auszuführen, wenn sie das nächste Mal den Server abrufen. Die Häufigkeit des Serverabrufs durch die Agenten wird durch die Eigenschaft **Administrations-Abrufintervall** des Agenten gesteuert.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 16: Ändern des entfernten Schemas“ auf Seite 262](#).

Lektion 16: Ändern des entfernten Schemas

In dieser praktischen Einführung ändern Sie das Schema der entfernten Datenbank, indem Sie eine neue Tabelle hinzufügen.

In dieser Lektion ändern Sie das Schema der entfernten Datenbank. In dieser praktischen Einführung tritt eine Schemaänderung auf, wenn Sie den Namen des entfernten Schemas der Datenbank ändern. Sie werden nie gezwungen sein, den Namen des entfernten Schemas zu ändern. Dies bleibt immer Ihrer freien Entscheidung überlassen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Kontext und Bemerkungen

Sie sollten sicherstellen, dass eine entfernte Aufgabe, die Sie in einer entfernten Datenbank ausführen können, in jeder anderen entfernten Datenbank mit demselben Namen des entfernten Schemas ausgeführt werden kann. Sie sollten den Namen des entfernten Schemas einer Datenbank ändern, wenn Sie die Datenbank auf eine Art ändern, die zu einem Fehlschlag oder zu einem Erfolg einer Aufgabe führen könnte. Die einzigen Befehle in einer Aufgabe, die vom Status der entfernten Datenbank betroffen sind, sind die Befehle **Synchronisieren** und **SQL ausführen**.

Synchronisieren-Befehle hängen vom Vorhandensein von Synchronisationsprofilen in der entfernten Datenbank ab, daher müssen Sie Namen entfernter Schemata immer ändern, wenn Sie ein Synchronisationsprofil hinzufügen oder entfernen.

SQL ausführen-Befehle hängen vom Status vieler Datenbankobjekte ab, die Sie normalerweise zum Schema gehörig ansehen. Einige Beispiele für Änderungen, die **SQL ausführen**-Befehle betreffen würden und daher eine Änderung des Namens eines entfernten Schemas bedingen, sind das Hinzufügen oder Entfernen von Tabellen in der Datenbank, die Änderung der Definition von Tabellen in der Datenbank und das Hinzufügen oder Entfernen von gespeicherten Prozeduren.

Aufgabe

1. Kehren Sie zur Ansicht **Ordner** in Sybase Central zurück. Erweitern Sie unter **MobiLink 16** das Element **Central Admin Tutorial**, klicken Sie mit der rechten Maustaste auf **Namen für entfernte Schemata** und klicken Sie auf **Neu » Name für entfernte Schemata**.

Der **Assistent zur Erstellung eines Namens für ein entferntes Schema** wird angezeigt.

2. Geben Sie **Tutorial Application v2.0** für den Namen des Schemas ein und klicken Sie auf **Fertig stellen**.

3. Erstellen Sie eine neue entfernte Aufgabe. Klicken Sie mit der rechten Maustaste in der Ansicht **Order** auf Sybase Central unter **Central Admin Tutorial** auf **Entfernte Aufgaben** und klicken Sie auf **Neu » Entfernte Aufgaben**. Der **Assistent für die Erstellung einer entfernten Aufgabe** wird angezeigt.
4. Geben Sie auf der **Willkommensseite** den Wert **Schema Upgrade** im Feld **Name** ein.
5. Aktivieren Sie **Diese Aufgabe benötigt oder erstellt eine entfernte Datenbank** und legen Sie die Option **Name des entfernten Schemas** mit **Tutorial Application v1.0** fest.
6. Aktivieren Sie **Diese Aufgabe aktualisiert das Schema der verwalteten entfernten Datenbank** und legen Sie **Neuer Name für das entfernte Schema** mit **Tutorial Application v2.0** fest. Klicken Sie auf **Fertig stellen**.
7. Wählen Sie in der Registerkarte **Befehle** die Option **SQL ausführen** aus der Dropdown-Liste **Befehlstyp** aus. Im Feld **SQL** geben Sie folgende Daten ein:

```
CREATE TABLE product (
    prod_id integer primary key,
    name varchar( 100 )
);
```

Die Schemaänderung ist damit abgeschlossen.

Bevor Sie die Aufgabe zum Ändern des Schemas bereitstellen, müssen Sie Aufgaben berücksichtigen, die dem entfernten Gerät bereits zugewiesen sind. Nachdem die **Schema Upgrade**-Aufgabe abgeschlossen wurde, ist der Name für das entfernte Schema der Datenbank **Tutorial Application v2.0**. Alle Aufgaben auf dem entfernten Gerät, die dem alten Namen des entfernten Schemas **Tutorial Application v1.0** zugewiesen sind, können nicht mehr ausgeführt werden und werden vom Agenten verworfen. Um die von diesen Aufgaben bereitgestellten Funktionen zu erhalten, müssen Sie neue Versionen der Aufgaben erstellen und mit dem neuen Namen für das entfernte Schema verknüpfen.

8. Klicken Sie in der Ansicht **Ordner** unter **Central Admin Tutorial » Konsolidierte Datenbanken » Tutorial » Agenten** auf **AID_JOHN**. Wählen Sie die Registerkarte **Aufgaben** im rechten Fensterausschnitt. Nur aktive Aufgaben werden weiterhin vom Agenten ausgeführt. Dies sind die einzigen Aufgaben, für die Sie eventuell neue Versionen erstellen müssen. In diesem Fall ist die einzige aktive Aufgabe die **Sync every hour**-Aufgabe.

Sie können ermitteln, ob diese Aufgabe mit dem alten Namen des entfernten Schemas verbunden ist, indem Sie die Spalte **Name des entfernten Schemas** auf der Registerkarte **Aufgaben** aktivieren. Diese Aufgabe zeigt, dass der **Name des entfernten Schemas** der **Sync every hour**-Aufgabe **Tutorial Application v1.0** und daher mit dem alten Namen des entfernten Schemas verbunden ist. Um die Synchronisation nach der Schemaänderung fortzusetzen, müssen Sie eine neue Version dieser Aufgabe erstellen und dem Agenten zuweisen.

9. Klicken Sie mit der rechten Maustaste auf die **Sync every hour**-Aufgabe und klicken Sie auf **Gehe zu Aufgabe**.
10. Klicken Sie mit der rechten Maustaste auf die bereitgestellte Aufgabe **Sync every hour** und klicken Sie auf **Kopieren**.

11. Klicken Sie mit der rechten Maustaste auf **Entfernte Aufgaben** und klicken Sie auf **Einfügen**. Wenn Sie aufgefordert werden, einen Namen für die kopierte Aufgabe einzugeben, geben Sie **Sync every hour v2** ein und klicken Sie auf **OK**.
12. Sie müssen darüber nachdenken, ob Befehle in der Aufgabe Änderungen erfordern, damit sie mit dem neuen Schema verwendet werden können. In diesem Fall ist die Antwort nein. Es ist nur ein Befehl vorhanden und er hängt nur vom **tutorial1_JOHN**-Synchronisationsprofil ab, das Sie bei dieser Schemaänderung nicht geändert haben.
13. Markieren Sie die Aufgabe als mit dem neuen Namen des entfernten Schemas verbunden. Klicken Sie mit der rechten Maustaste auf die **Sync every hour v2**-Aufgabe und wählen Sie **Eigenschaften**. Wählen Sie auf der Seite **Allgemein** des Eigenschaftsfensters **Tutorial Application v2.0** für **Name des entfernten Schemas** und klicken Sie auf **OK**.
14. Um das Deployment der neuen Aufgabe vorzunehmen, klicken Sie mit der rechten Maustaste auf die **Sync every hour v2**-Aufgabe und wählen Sie **Deployment**. Klicken Sie auf **Weiter**.
15. Klicken Sie unter **Empfänger** auf **Spezifische Agenten** und wählen Sie dann den Agenten **AID_JOHN** aus. Klicken Sie auf **Weiter** und auf **Fertig stellen**.
16. Klicken Sie mit der rechten Maustaste auf die **Schema Upgrade**-Aufgabe und klicken Sie auf **Deployment**. Klicken Sie auf **Weiter**.
17. Klicken Sie in der Dropdown-Liste **Empfänger** auf **Spezifische Agenten** und weisen Sie die Aufgabe dem Agenten **AID_JOHN** zu. Klicken Sie auf **Weiter** und auf **Fertig stellen**.

Ergebnisse

Das Ausführen der **Schema Upgrade**-Aufgabe dürfte erfolgreich verlaufen. Danach wird die **Sync every hour v2**-Aufgabe einmal pro Stunde ausgeführt und die **Sync every hour**-Aufgabe funktioniert nicht mehr.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 17: Abfrage in der entfernten Datenbank](#)“ auf Seite 264.

Lektion 17: Abfrage in der entfernten Datenbank

In dieser Lektion führen Sie eine Abfrage in der entfernten Datenbank durch und geben die Ergebnisse an den Server zurück. Dies ist besonders nützlich, wenn Sie eine Fehlersuche durchführen, weil sie genau feststellen können, in welchem Status sich die entfernte Datenbank befindet.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Kontext und Bemerkungen

Die Tabellen, die Sie in dieser praktischen Einführung der Datenbank hinzugefügt haben, enthalten keine Daten, daher werden Sie eine Abfrage der Systemtabellen der Datenbank durchführen. Zwar fragen Sie in diesem Beispiel eine Systemtabelle ab, es besteht aber kein Unterschied zur Abfrage einer Benutzertabelle.

Angenommen, Sie möchten bestätigen, dass die Schemaänderung, die Sie in der vorherigen Lektion vorgenommen haben, wie erwartet erfolgt ist und die Product-Tabelle mit den richtigen Spalten erstellt wurde. Sie können diese Bestätigung erhalten, indem Sie die Systemtabellen systable und systabcol abfragen.

Aufgabe

1. Erweitern Sie in der Ansicht **Ordner** von Sybase Central unter **MobiLink 16** den Eintrag **Central Admin Tutorial**, klicken Sie mit der rechten Maustaste auf **Entfernte Aufgaben** und klicken Sie auf **Neu » Entfernte Aufgabe**.

Der **Assistent für die Erstellung einer entfernten Aufgabe** wird angezeigt.

2. Geben Sie auf der **Willkommenseite** den Wert **Table Query** im Feld **Name** ein.
3. Aktivieren Sie **Diese Aufgabe benötigt oder erstellt eine entfernte Datenbank**, legen Sie die Option **Name des entfernten Schemas** mit **Tutorial Application v2.0** fest und klicken Sie auf **Weiter**.
4. Aktivieren Sie auf der Seite **Auslösemechanismen** das Kontrollkästchen **Beim Empfang von einem Agenten** und klicken Sie auf **Fertig stellen**.
5. Fügen Sie der Aufgabe einen **SQL ausführen**-Befehl mit folgenden SQL-Code hinzu:

```
SELECT * FROM systable WHERE table_name = 'product'
go
SELECT * FROM systabcol ORDER BY table_id
```
6. Klicken Sie mit der rechten Maustaste auf die neue **Table Query**-Aufgabe und wählen Sie **Deployment**. Klicken Sie auf **Weiter**.
7. Wählen Sie für **Empfänger** die Option **Spezifische Agenten**, wählen Sie den Agenten **AID_JOHN** aus und klicken Sie auf **Weiter** und dann nochmals auf **Weiter**.
8. Legen Sie auf der Seite **Bericht über Ergebnisse und Status** sowohl **Bei Erfolg der Aufgabe** als auch **Bei Fehlschlag der Aufgabe** mit **Ergebnisse und Status sofort senden** fest. Klicken Sie auf **Fertig stellen** und warten Sie, bis die Aufgabe ausgeführt wird.
9. Klicken Sie auf die bereitgestellte Kopie der **Table Query**-Aufgabe in der Ansicht **Ordner** und klicken Sie auf die Registerkarte **Ergebnisse**. Wenn Sie auf der Registerkarte keine Ergebnisse sehen, warten Sie, bis die Registerkarte automatisch aktualisiert wird, oder drücken Sie F5 für sofortige Aktualisierung.
10. Klicken Sie mit der rechten Maustaste auf die Zeile in der Tabelle für die Anweisung **SQL ausführen** und wählen Sie **Details** aus.

Das Fenster **Befehlsergebnis** erscheint.

11. Klicken Sie im Fenster auf die Registerkarte **Ergebnisse**. Diese Registerkarte zeigt die Ergebnisse aller ausgeführten Abfragen. Die Dropdown-Liste **Ergebnis** im oberen Teil des Fensters ermöglicht es Ihnen, zwischen den Ergebnissen für die beiden Abfragen umzuschalten. Klicken Sie auf **Schließen**.

Ergebnisse

Die Ergebnisse aus der entfernten Datenbank werden angezeigt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 18: Upload von Dateien mit SIRT](#)“ auf Seite 266.

Lektion 18: Upload von Dateien mit SIRT

In dieser Lektion laden Sie Dateien aus dem entfernten Gerät hoch, indem Sie eine serverinitiierte entfernte Aufgabe (SIRT) verwenden. Der Upload von Dateien aus dem entfernten Gerät ist nützlich für die Fehlerbehandlung, da ein Administrator die Dateien auf Problemstellen untersuchen kann.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Kontext und Bemerkungen

Als Sie den MobiLink-Agenten auf dem entfernten Gerät gestartet haben, wiesen Sie ihn an, Meldungen in die Datei *agent.txt* zu protokollieren. Sie werden jetzt die Datei vom entfernten Gerät abrufen und überprüfen.

Aufgabe

1. Erweitern Sie in der Ansicht **Ordner** von Sybase Central unter **MobiLink 16** den Eintrag **Central Admin Tutorial**, klicken Sie mit der rechten Maustaste auf **Entfernte Aufgaben** und klicken Sie auf **Neu » Entfernte Aufgabe**.

Der **Assistent für die Erstellung einer entfernten Aufgabe** wird angezeigt.

2. Geben Sie auf der **Willkommenseite** den Wert **Upload Agent Log** im Feld **Name** ein.
3. Deaktivieren Sie die Option **Diese Aufgabe benötigt oder erstellt eine entfernte Datenbank**, falls sie aktiviert ist, und klicken Sie auf **Fertig stellen**.
4. Klicken Sie auf die neue Aufgabe in der Ansicht **Ordner** und fügen Sie der Aufgabe einen Befehl hinzu. Legen Sie den **Befehlstyp** mit **Datei hochladen** fest.

5. Legen Sie **Serverdateiname** auf *{agent_id}\agent.txt* und **Name der entfernten Datei** mit *{agent_log}* fest. Sie können die Schaltfläche mit den drei Punkten im Befehlseditor verwenden, um die Makrowerte einfach einzugeben.

Der *{agent_log}*-Makro wird durch den Namen der Logdatei ersetzt, die vom MobiLink-Agenten auf dem entfernten Gerät geführt wird.

Im Feld **Serverdateiname** haben Sie mit dem *{agent_id}*-Makro das Verzeichnis angegeben, in dem die Datei gespeichert wird. Dies ist besonders wichtig. Wenn Sie bei der Angabe des Serverdateinamens keinen Makro verwenden, legt jeder Agent, der die Aufgabe ausführt, die hochgeladene Datei am selben Standort ab und überschreibt dabei die vom vorherigen Agenten hochgeladene Datei. Mit einem Makro wird sichergestellt, dass jeder Agent seine Logdatei an einen anderen Speicherort auf dem Server hochlädt, sodass Sie alle Logdateien anzeigen können.

6. Klicken Sie mit der rechten Maustaste auf die neue **Upload Agent Log**-Aufgabe und wählen Sie **Deployment**. Klicken Sie auf **Weiter**.
7. Klicken Sie unter **Empfänger** auf **Spezifische Agenten** und wählen Sie dann den Agenten **AID_JOHN** aus. Klicken Sie auf **Weiter**.
8. Klicken Sie auf der Seite **Zustelloptionen** auf **Bei der nächsten Synchronisation des Agenten** und klicken Sie auf **Weiter**.
9. Legen Sie auf der Seite **Bericht über Ergebnisse und Status** sowohl **Bei Erfolg der Aufgabe** als auch **Bei Fehlschlag der Aufgabe** auf **Ergebnisse und Status sofort senden** fest. Klicken Sie auf **Fertig stellen**.
10. Die Aufgabe muss vom Administrator in Sybase Central initiiert werden. Um die Aufgabe zu initiieren, gehen Sie zu **AID_JOHN** unter **Agenten**. Klicken Sie im Fensterausschnitt mit der rechten Maustaste auf der Registerkarte **Aufgaben** auf die Aufgabe **Upload Agent Log** und wählen Sie **Einleiten**. Warten Sie, bis die Aufgabe ausgeführt ist.

Die hochgeladene Datei wird in das MobiLink-Uploadverzeichnis gestellt, das mit der -ftru-Option in der MobiLink-Befehlszeile festgelegt wurde. Sie haben *c:\cadmin_demo\consolidated\upload* als Uploadverzeichnis angegeben. Sehen Sie sich dieses Verzeichnis in einer Eingabeaufforderung oder im Windows-Explorer an. Das Unterverzeichnis **AID_JOHN** müsste vorhanden sein. In diesem Unterverzeichnis befindet sich die hochgeladene *agent.txt*-Datei.

Ergebnisse

Ergebnis

Nächste Schritte

Gehen Sie weiter zu „Aufräumen“ auf Seite 267.

Aufräumen

Entfernen Sie alle Daten der praktischen Einführung von Ihrem Computer.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen einer konsolidierten Datenbank](#)“ auf Seite 242.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Zentrale Administration von entfernten Datenbanken](#)“ auf Seite 241.

Entfernen der Daten der praktischen Einführung von Ihrem Computer

1. Schließen Sie die Fenster von Interactive SQL, Sybase Central, MobiLink und vom Synchronisationsclient, indem Sie mit der rechten Maustaste auf die einzelnen Anzeigen in der Taskleiste klicken und Schließen wählen.
2. Löschen Sie alle Datenquellen im Zusammenhang mit der praktischen Einführung:
 - a. Starten Sie den ODBC-Datenquellen-Administrator.
 - b. Wählen Sie **Start » Programme » SQL Anywhere 16 » Administrationstools » ODBC-Datenquellen-Administrator**.
 - c. Wählen Sie **cadmin_tutorial_consol** aus der Liste der **Benutzerdatenquellen** und klicken Sie auf **Entfernen**.

Ergebnisse

Die Ressourcen der praktischen Einführung werden entfernt.

Nächste Schritte

Keine.

Praktische Einführung: Änderung eines Schemas mit der Skriptversionsklausel

In dieser praktischen Einführung wird beschrieben, wie eine Schemaänderung in einer entfernten Datenbank durchgeführt wird, die zu einem Synchronisationssystem gehört, in der die erweiterte dbmsync-Option ScriptVersion nicht verwendet wird. In dieser praktischen Einführung richten Sie ein Synchronisationssystem ein, das eine einzige Tabelle synchronisiert und dann eine Schemaänderung durchführt, um eine Spalte der synchronisierenden Tabelle hinzuzufügen und die Synchronisation fortzusetzen.

Erforderliche Software

In dieser praktischen Einführung wird davon ausgegangen, dass SQL Anywhere (einschließlich MobiLink) auf Ihrem lokalen Computer, auf dem Sie diese praktische Einführung ausführen, vollständig installiert ist.

Privilegien

Sie müssen die folgenden Rollen und Privilegien für die konsolidierte Datenbank haben:

- SYS_AUTH_RESOURCE_ROLE-Kompatibilitätsrolle
- MONITOR-Systemprivileg

Sie müssen die folgenden Rollen und Privilegien für die entfernte Datenbank haben:

- SYS_REPLICATION_ADMIN_ROLE-Systemrolle
- SYS_RUN_REPLICATION_ROLE-Systemrolle

Übersicht

In dieser praktischen Einführung wird Folgendes veranschaulicht:

- „Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank“
- „Lektion 2: Erstellen und Konfigurieren der entfernten Datenbank“
- „Lektion 3: Synchronisieren der entfernten Datenbank“
- „Lektion 4: Einfügen von Daten in die entfernte Datenbank“
- „Lektion 5: Durchführen einer Schemaänderung in der konsolidierten Datenbank“
- „Lektion 6: Durchführen einer Schemaänderung in der entfernten Datenbank“
- „Lektion 7: Daten einfügen“
- „Lektion 8: Synchronisieren“

Siehe auch

- „Schemaänderungen in entfernten MobiLink-Clients“ [*MobiLink - Clientadministration*]

Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank

In dieser Lektion richten Sie eine konsolidierte Datenbank für die Synchronisation ein.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Änderung eines Schemas mit der Skriptversionsklausel](#)“ auf Seite 268.

Aufgabe

1. Führen Sie die folgenden Befehle aus, um eine konsolidierte Datenbank zu erstellen und zu starten.

```
md c:\cons
cd c:\cons
dbinit -dba DBA,sql consol.db
dbsrv16 consol.db
```

2. Führen Sie den folgenden Befehl aus, um eine ODBC-Datenquelle für die konsolidierte Datenbank zu definieren.

```
dbdsn -w dsn_consol -y -c "UID=DBA;PWD=sql;DBF=consol.db;SERVER=consol"
```

3. Um eine Datenbank als konsolidierte MobiLink-Datenbank zu verwenden, müssen Sie ein Setup-Skript ausführen, das Systemtabellen, Ansichten und gespeicherte Prozeduren hinzufügt, die von MobiLink verwendet werden. Der folgende Befehl richtet *consol.db* als konsolidierte Datenbank ein.

```
dbisql -c "DSN=dsn_consol" %SQLANY16%\MobiLink\setup\syncsa.sql
```

4. Öffnen Sie Interactive SQL und stellen Sie eine Verbindung zu *consol.db* mit der Datenquelle *dsn_consol* her.

```
dbisql -c "DSN=dsn_consol"
```

5. Führen Sie hierzu die folgende SQL-Anweisung aus: Sie erstellen die Customer-Tabelle in der konsolidierten Datenbank und die erforderlichen Synchronisationsskripten.

```
CREATE TABLE customer (
    id      unsigned integer primary key,
    name    varchar( 256),
    phone   varchar( 12 )
);

CALL ml_add_column('my_ver1', 'customer', 'id', null );
CALL ml_add_column('my_ver1', 'customer', 'name', null );
CALL ml_add_column('my_ver1', 'customer', 'phone', null );

CALL ml_add_table_script( 'my_ver1', 'customer', 'upload_insert',
    'INSERT INTO customer ( id, name, phone ) '
    || 'VALUES ({ml r.id}, {ml r.name}, {ml r.phone}) ' );

CALL ml_add_table_script( 'my_ver1', 'customer', 'download_cursor',
    'SELECT id, name, phone from customer' );

CALL ml_add_table_script( 'my_ver1', 'customer',
    'download_delete_cursor', '--{ml_ignore}' );
COMMIT;
```

Nachdem Sie die SQL-Anweisungen ausgeführt haben, lassen Sie Interactive SQL aktiviert und mit der Datenbank verbunden, als würden Sie weitere SQL-Anweisungen in der Datenbank ausführen, während Sie die praktische Einführung durcharbeiten.

6. Starten Sie den MobiLink-Server mit dem folgenden Befehl.

```
start mlsrv16 -c "DSN=dsn_consol" -v+ -ot mlsrv.txt -zu+
```

Ergebnisse

Die konsolidierte Datenbank wird erstellt und so eingerichtet, dass sie mit MobiLink eingesetzt werden kann.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 2: Erstellen und Konfigurieren der entfernten Datenbank](#)“ auf Seite 271.

Lektion 2: Erstellen und Konfigurieren der entfernten Datenbank

In dieser Lektion richten Sie eine entfernte Datenbank für die Synchronisation ein.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Änderung eines Schemas mit der Skriptversionsklausel](#)“ auf Seite 268.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank](#)“ auf Seite 269.

Aufgabe

1. Führen Sie die folgenden Befehle aus, um eine entfernte Datenbank zu erstellen und zu starten:

```
cd..
md c:\remote
cd c:\remote
dbinit -dba DBA,sql remote.db
dbsrv16 remote.db
```

2. Öffnen Sie eine weitere Instanz von Interactive SQL und verbinden Sie sich mit *remote.db*.

```
dbisql -c "SERVER=remote;DBF=remote.db;UID=DBA;PWD=sql"
```

3. Führen Sie die folgende SQL-Anweisung in Interactive SQL aus, um die Tabelle zu erstellen, die synchronisiert werden soll.

```
CREATE TABLE customer (
  id      UNSIGNED INTEGER PRIMARY KEY,
  name    VARCHAR( 256),
  phone   VARCHAR( 12 )
);
```

4. In der weiterhin offenen Interactive SQL-Instanz und bei aufrechter Verbindung mit der entfernten Datenbank erstellen Sie eine Publikation, einen MobiLink-Benutzer und eine Subskription. Die Skriptversion ist über die SCRIPT VERSION-Klausel mit der Subskription verbunden. Dies ist sehr wichtig, da die in dieser praktischen Einführung gezeigte Schema-Upgradeprozedur nur für Subskriptionen funktioniert, bei denen mit der SCRIPT VERSION-Klausel die Skriptversionfunktion aktiviert ist.

```
CREATE PUBLICATION p1 (  
    TABLE customer  
);  
  
CREATE SYNCHRONIZATION USER u1;  
  
CREATE SYNCHRONIZATION SUBSCRIPTION my_sub  
TO p1  
FOR u1  
SCRIPT VERSION 'my_ver1';
```

Nachdem Sie die SQL-Anweisungen ausgeführt haben, lassen Sie Interactive SQL aktiviert und mit der Datenbank verbunden, als würden Sie weitere SQL-Anweisungen in der Datenbank ausführen, während Sie die praktische Einführung durcharbeiten.

Ergebnisse

Die entfernte Datenbank wird erstellt und konfiguriert.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 3: Synchronisieren der entfernten Datenbank“ auf Seite 272](#).

Lektion 3: Synchronisieren der entfernten Datenbank

Sie haben jetzt ein funktionierendes Synchronisationssystem eingerichtet. In dieser Lektion testen Sie es durch Einfügen einiger Daten und nachfolgende Synchronisation.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Änderung eines Schemas mit der Skriptversionsklausel“ auf Seite 268](#).

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank“ auf Seite 269](#).

Aufgabe

1. Verwenden Sie die Instanz von Interactive SQL, die mit der konsolidierten Datenbank verbunden ist, und führen Sie die folgende SQL-Anweisung aus, um eine Zeile in die customer-Tabelle einzufügen.

```
INSERT INTO customer VALUES( 100, 'John Jones', '519-555-1234' );  
COMMIT;
```

2. Verwenden Sie die Instanz von Interactive SQL, die mit der entfernten Datenbank verbunden ist, und führen Sie die folgende SQL-Anweisung aus, um eine Zeile in die Customer-Tabelle einzufügen.

```
INSERT INTO customer VALUES( 1, 'Willie Lowman', '705-411-6372' );  
COMMIT;
```

3. Synchronisieren Sie, indem Sie den folgenden Befehl ausführen.

```
dbmsync -v+ -ot sync1.txt -c UID=DBA;PWD=sql;SERVER=remote -s my_sub -k
```

Ergebnisse

Die entfernte Datenbank wird synchronisiert.

Den Erfolg der Synchronisation erkennen Sie durch den Vergleich der Inhalte der Customer-Tabelle in der entfernten und in der konsolidierten Datenbank. Auch ein Blick in das dbmsync-Log *sync1.txt* ist sinnvoll, um es auf Fehlermeldungen zu prüfen.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 4: Einfügen von Daten in die entfernte Datenbank“](#) auf Seite 273.

Lektion 4: Einfügen von Daten in die entfernte Datenbank

In dieser Lektion fügen Sie Daten in die entfernte Datenbank ein, um zu zeigen, dass eine Schemaänderung auch durchgeführt werden kann, wenn Vorgänge in der entfernten Datenbank vorhanden sind, die hochgeladen werden müssen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Änderung eines Schemas mit der Skriptversionsklausel“](#) auf Seite 268.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank“](#) auf Seite 269.

Aufgabe

- Verwenden Sie die Instanz von Interactive SQL, die mit der entfernten Datenbank verbunden ist, und führen Sie die folgende SQL-Anweisung aus, um eine Zeile in die Customer-Tabelle einzufügen.

```
INSERT INTO customer VALUES( 2, 'Sue Slow', '602-411-5467' );  
COMMIT;
```

Ergebnisse

Die Daten werden in die entfernte Datenbank eingefügt.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 5: Durchführen einer Schemaänderung in der konsolidierten Datenbank“](#) auf Seite 274.

Lektion 5: Durchführen einer Schemaänderung in der konsolidierten Datenbank

In dieser Lektion führen Sie eine Schemaänderung in der konsolidierten Datenbank durch.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Änderung eines Schemas mit der Skriptversionsklausel“](#) auf Seite 268.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank“](#) auf Seite 269.

Aufgabe

1. Fügen Sie eine neue Spalte in die Customer-Tabelle ein, um die Mobiltelefonnummer des Kunden zu speichern. Zuerst fügen Sie die neue Spalte der konsolidierten Datenbank hinzu, indem Sie die folgende SQL-Anweisung in der Instanz von Interactive SQL ausführen, die mit der konsolidierten Datenbank verbunden ist.

```
ALTER TABLE customer ADD cell_phone VARCHAR(12) DEFAULT NULL;
```

2. Erstellen Sie eine neue Skriptversion mit dem Namen **my_ver2**, um Synchronisationen von entfernten Datenbanken mit dem neuen Schema abzuwickeln. Entfernte Datenbanken mit dem alten Schema verwenden weiterhin die alte Skriptversion **my_ver1**. Führen Sie die folgende SQL-Anweisung für die konsolidierte Datenbank aus.

```
CALL ml_add_column('my_ver2', 'customer', 'id', null );
CALL ml_add_column('my_ver2', 'customer', 'name', null );
CALL ml_add_column('my_ver2', 'customer', 'phone', null );
CALL ml_add_column('my_ver2', 'customer', 'cell_phone', null );

CALL ml_add_table_script( 'my_ver2', 'customer', 'upload_insert',
    'INSERT INTO customer ( id, name, phone, cell_phone ) '
    || 'VALUES ({ml r.id}, {ml r.name}, {ml r.phone}, {ml '
    r.cell_phone})' );

CALL ml_add_table_script( 'my_ver2', 'customer', 'download_cursor',
    'SELECT id, name, phone, cell_phone from customer' );

CALL ml_add_table_script( 'my_ver2', 'customer',
    'download_delete_cursor', '--{ml_ignore}' );
COMMIT;
```

Ergebnisse

Das Schema der konsolidierten Datenbank wird aktualisiert.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 6: Durchführen einer Schemaänderung in der entfernten Datenbank“](#) auf Seite 275.

Lektion 6: Durchführen einer Schemaänderung in der entfernten Datenbank

In dieser Lektion ändern Sie die entfernte Datenbank, um eine neue Spalte in die Customer-Tabelle einzufügen und die zur Synchronisation verwendete Skriptversion zu ändern.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Änderung eines Schemas mit der Skriptversionsklausel](#)“ auf Seite 268.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank](#)“ auf Seite 269.

Aufgabe

1. Starten Sie eine Änderung des Synchronisationsschemas. Dies ist erforderlich für die meisten Schemaänderungen, die synchronisierende Tabellen betreffen. Diese Anweisung ändert die Skriptversion, die verwendet wird, um die Subskription zu synchronisieren, und sperrt die betroffene Tabelle, damit die Schemaänderung sicher durchgeführt werden kann.

Führen Sie die folgende SQL-Anweisung in der entfernten Datenbank über die Instanz von Interactive SQL aus, die mit der entfernten Datenbank verbunden ist.

```
START SYNCHRONIZATION SCHEMA CHANGE  
FOR TABLES customer  
SET SCRIPT VERSION = 'my_ver2';
```

2. Fügen Sie die neue Spalte mit der nachfolgenden SQL-Anweisung der Customer-Tabelle hinzu.

```
ALTER TABLE customer ADD cell_phone VARCHAR(12) DEFAULT NULL;
```

3. Schließen Sie die Schemaänderung ab, sodass die Tabellen entsperrt werden.

```
STOP SYNCHRONIZATION SCHEMA CHANGE;
```

Ergebnisse

In der entfernten Datenbank wird eine Änderung des Synchronisationsschemas durchgeführt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 7: Daten einfügen](#)“ auf Seite 275.

Lektion 7: Daten einfügen

In dieser Lektion fügen Sie mit dem neuen Schema weitere Daten in die entfernte und konsolidierte Datenbank ein.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Änderung eines Schemas mit der Skriptversionsklausel“](#) auf Seite 268.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank“](#) auf Seite 269.

Einfügen von Daten in die entfernte Datenbank

1. Über Interactive SQL führen Sie folgende SQL-Anweisungen in der entfernten Datenbank aus.

```
INSERT INTO customer VALUES( 3, 'Mo Hamid', '613-411-9999',  
'613-502-1212' );  
COMMIT;
```

2. Über Interactive SQL führen Sie folgende SQL-Anweisungen in der konsolidierten Datenbank aus.

```
INSERT INTO customer VALUES( 101, 'Theo Tug', '212-911-7677',  
'212-311-3900' );  
COMMIT;
```

Ergebnisse

Die Daten werden mit dem neuen Schema in die entfernte und die konsolidierte Datenbank eingeführt.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 8: Synchronisieren“](#) auf Seite 276.

Lektion 8: Synchronisieren

In dieser Lektion synchronisieren Sie noch einmal mit den Schemaänderungen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Änderung eines Schemas mit der Skriptversionsklausel“](#) auf Seite 268.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank“](#) auf Seite 269.

Aufgabe

- Jetzt synchronisieren Sie erneut, indem Sie den folgenden Befehl ausführen:

```
dbmlsync -v+ -ot sync2.txt -c UID=DBA;PWD=sql;SERVER=remote -s my_sub -k
```

Die Zeile für **Sue Slow**, die vor der Schemaänderung eingefügt wurde, wird mit der Skriptversion **my_ver1** hochgeladen. Die Zeile für **Mo Hamid**, die nach der Schemaänderung eingefügt wurde,

wird mit der Skriptversion **my_ver2** hochgeladen. Zeilen werden mit dem Download-Cursor für **my_ver2** heruntergeladen.

Ergebnisse

Die Änderung des Schemas ist nun abgeschlossen und Sie können die Synchronisation auf die übliche Weise fortsetzen.

Nächste Schritte

Keine.

Praktische Einführung: Änderung eines Schemas mit der erweiterten Option ScriptVersion

Diese praktische Einführung zeigt, wie eine Schemaänderung durchgeführt wird, wenn Sie die erweiterte Option ScriptVersion verwenden.

Hinweis

Es wird empfohlen, die erweiterte Option ScriptVersion möglichst zu vermeiden. Verbinden Sie Ihre Skriptversion mit Ihrer Subskription besser über die SCRIPT VERSION-Klausel in der CREATE SYNCHRONIZATION SUBSCRIPTION-Anweisung oder der SET SCRIPT VERSION-Klausel in der ALTER SYNCHRONIZATION SUBSCRIPTION-Anweisung. Durch diese Implementierungen erhalten Sie mehr Flexibilität für die Durchführung von Schema-Upgrades.

Erforderliche Software

In dieser praktischen Einführung wird davon ausgegangen, dass SQL Anywhere (einschließlich MobiLink) auf Ihrem lokalen Computer, auf dem Sie diese praktische Einführung ausführen, vollständig installiert ist.

Privilegien

Sie müssen die folgenden Rollen und Privilegien für die konsolidierte Datenbank haben:

- SYS_AUTH_RESOURCE_ROLE-Kompatibilitätsrolle
- MONITOR-Systemprivileg

Sie müssen die folgenden Rollen und Privilegien für die entfernte Datenbank haben:

- SYS_REPLICATION_ADMIN_ROLE-Systemrolle
- SYS_RUN_REPLICATION_ROLE-Systemrolle

Übersicht

In dieser praktischen Einführung wird Folgendes veranschaulicht:

- „Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank“
- „Lektion 2: Erstellen und Konfigurieren der entfernten Datenbank“
- „Lektion 3: Synchronisieren der entfernten Datenbank“
- „Lektion 4: Durchführen der Schemaänderung in der konsolidierten Datenbank“
- „Lektion 5: Durchführen einer Schemaänderung in der entfernten Datenbank“

Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank

In dieser Lektion richten Sie eine konsolidierte Datenbank für die Synchronisation ein.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Änderung eines Schemas mit der erweiterten Option ScriptVersion“](#) auf Seite 277.

Erstellen und Konfigurieren einer konsolidierten Datenbank

1. Führen Sie folgenden Befehl aus, um eine konsolidierte Datenbank zu erstellen und zu starten.

```
md c:\cons
cd c:\cons
dbinit -dba DBA,sql consol.db
dbsrv16 consol.db
```

2. Führen Sie den folgenden Befehl aus, um eine ODBC-Datenquelle für die konsolidierte Datenbank zu definieren:

```
dbdsn -w dsn_consol -y -c "UID=DBA;PWD=sql;DBF=consol.db;SERVER=consol"
```

3. Um eine Datenbank als konsolidierte MobiLink-Datenbank zu verwenden, müssen Sie ein Setup-Skript ausführen, das Systemtabellen, Ansichten und gespeicherte Prozeduren hinzufügt, die von MobiLink verwendet werden. Der folgende Befehl richtet *consol.db* als konsolidierte Datenbank ein.

```
dbisql -c "DSN=dsn_consol" %SQLANY16%\MobiLink\setup\syncsa.sql
```

4. Öffnen Sie Interactive SQL und stellen Sie mithilfe des **dsn_consol**-DSN eine Verbindung zu *consol.db* her.

```
dbisql -c "DSN=dsn_consol"
```

5. Führen Sie hierzu die folgende SQL-Anweisung in Interactive SQL aus. Sie erstellen die Customer-Tabelle in der konsolidierten Datenbank und die erforderlichen Synchronisationsskripte.

```
CREATE TABLE customer (
  id      unsigned integer primary key,
  name    varchar( 256),
```

```
    phone    varchar( 12 )
);

CALL ml_add_column('my_ver1', 'customer', 'id', null );
CALL ml_add_column('my_ver1', 'customer', 'name', null );
CALL ml_add_column('my_ver1', 'customer', 'phone', null );

CALL ml_add_table_script( 'my_ver1', 'customer', 'upload_insert',
    'INSERT INTO customer ( id, name, phone ) '
    || 'VALUES ({ml r.id}, {ml r.name}, {ml r.phone} )' );

CALL ml_add_table_script( 'my_ver1', 'customer', 'download_cursor',
    'SELECT id, name, phone from customer' );

CALL ml_add_table_script( 'my_ver1', 'customer',
    'download_delete_cursor', '--{ml_ignore}' );
COMMIT;
```

Nachdem Sie die SQL-Anweisungen ausgeführt haben, lassen Sie Interactive SQL aktiviert und mit der Datenbank verbunden, als würden Sie weitere SQL-Anweisungen in der Datenbank ausführen, während Sie die praktische Einführung durcharbeiten.

6. Starten Sie den MobiLink-Server mit dem folgenden Befehl:

```
start mlsrv16 -c "DSN=dsn_consol" -v+ -ot mlsrv.txt -zu+
```

Ergebnisse

Die konsolidierte Datenbank wird für die Synchronisation erstellt und konfiguriert.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 2: Erstellen und Konfigurieren der entfernten Datenbank“ auf Seite 279](#).

Lektion 2: Erstellen und Konfigurieren der entfernten Datenbank

In dieser Lektion richten Sie eine entfernte Datenbank für die Synchronisation ein.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Änderung eines Schemas mit der erweiterten Option ScriptVersion“ auf Seite 277](#).

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank“ auf Seite 278](#).

Aufgabe

1. Führen Sie die folgenden Befehle zum Erstellen und Starten einer konsolidierten Datenbank.

```
cd..
md c:\remote
```

```
cd c:\remote
dbinit -dba DBA,sql remote.db
dbeng16 remote.db
```

2. Öffnen Sie eine weitere Instanz von Interactive SQL und verbinden Sie sich mit *remote.db*.

```
dbisql -c "SERVER=remote;DBF=remote.db;UID=DBA;PWD=sql"
```

3. Führen Sie die folgende SQL-Anweisung in Interactive SQL aus, um die Tabelle zu erstellen, die synchronisiert werden soll.

```
CREATE TABLE customer (
    id      unsigned integer primary key,
    name    varchar( 256),
    phone   varchar( 12 )
);
```

4. Publikationen, MobiLink-Benutzer und Subskriptionen erstellen.

```
CREATE PUBLICATION p1 (
    TABLE customer
);

CREATE SYNCHRONIZATION USER u1;

CREATE SYNCHRONIZATION SUBSCRIPTION my_sub
TO p1
FOR u1
OPTION ScriptVersion='my_ver1';
```

Nachdem Sie die SQL-Anweisungen ausgeführt haben, lassen Sie Interactive SQL aktiviert und mit der Datenbank verbunden, als würden Sie weitere SQL-Anweisungen in der Datenbank ausführen, während Sie die praktische Einführung durcharbeiten.

Ergebnisse

Die entfernte Datenbank wird für die Synchronisation erstellt und konfiguriert.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 3: Synchronisieren der entfernten Datenbank“](#) auf Seite 280.

Lektion 3: Synchronisieren der entfernten Datenbank

Sie haben jetzt ein funktionierendes Synchronisationssystem eingerichtet. In dieser Lektion testen Sie es durch Einfügen einiger Daten und nachfolgende Synchronisation.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Änderung eines Schemas mit der erweiterten Option ScriptVersion“](#) auf Seite 277.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank“](#) auf Seite 278.

Aufgabe

1. Verwenden Sie die Instanz von Interactive SQL, die mit der konsolidierten Datenbank verbunden ist, und führen Sie die folgende SQL-Anweisung aus, um eine Zeile in die customer-Tabelle einzufügen.

```
INSERT INTO customer VALUES( 100, 'John Jones', '519-555-1234' );  
COMMIT;
```

2. Verwenden Sie die Instanz von Interactive SQL, die mit der entfernten Datenbank verbunden ist, und führen Sie die folgende SQL-Anweisung aus, um eine Zeile in die Customer-Tabelle einzufügen.

```
INSERT INTO customer VALUES( 1, 'Willie Lowman', '705-411-6372' );  
COMMIT;
```

3. Synchronisieren Sie, indem Sie den folgenden Befehl ausführen.

```
dbmlsync -v+ -ot sync1.txt -c UID=DBA;PWD=sql;SERVER=remote -s my_sub -k
```

Den Erfolg der Synchronisation erkennen Sie durch den Vergleich der Inhalte der Customer-Tabelle in der entfernten und in der konsolidierten Datenbank. Auch ein Blick in das dbmlsync-Log *sync1.txt* ist sinnvoll, um es auf Fehlermeldungen zu prüfen.

Ergebnisse

Die konsolidierten und entfernten Datenbanken werden synchronisiert.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 4: Durchführen der Schemaänderung in der konsolidierten Datenbank](#)“ auf Seite 281.

Lektion 4: Durchführen der Schemaänderung in der konsolidierten Datenbank

In dieser Lektion fügen Sie eine neue Spalte in die Customer-Tabelle ein, um die Mobiltelefonnummer des Kunden zu speichern.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Änderung eines Schemas mit der erweiterten Option ScriptVersion](#)“ auf Seite 277.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank](#)“ auf Seite 278.

Aufgabe

1. Verwenden Sie die Instanz von Interactive SQL, die mit der konsolidierten Datenbank verbunden ist, und führen Sie die folgende SQL-Anweisung aus, um eine Zeile in die customer-Tabelle einzufügen.

```
ALTER TABLE customer ADD cell_phone VARCHAR(12) DEFAULT NULL;
```

2. Erstellen Sie eine neue Skriptversion mit dem Namen my_ver2, um Synchronisationen von entfernten Datenbanken mit dem neuen Schema abzuwickeln. Entfernte Datenbanken mit dem alten Schema verwenden weiterhin die alte Skriptversion my_ver1.

Führen Sie die folgenden SQL-Anweisungen für die konsolidierte Datenbank aus:

```
CALL ml_add_column('my_ver2', 'customer', 'id', null );
CALL ml_add_column('my_ver2', 'customer', 'name', null );
CALL ml_add_column('my_ver2', 'customer', 'phone', null );
CALL ml_add_column('my_ver2', 'customer', 'cell_phone', null );

CALL ml_add_table_script( 'my_ver2', 'customer', 'upload_insert',
    'INSERT INTO customer ( id, name, phone, cell_phone ) '
    || 'VALUES ({ml r.id}, {ml r.name}, {ml r.phone}, {ml '
    || 'r.cell_phone})' );

CALL ml_add_table_script( 'my_ver2', 'customer', 'download_cursor',
    'SELECT id, name, phone, cell_phone from customer' );

CALL ml_add_table_script( 'my_ver2', 'customer',
    'download_delete_cursor', '--{ml_ignore}' );

COMMIT;
```

Ergebnisse

An der konsolidierten Datenbank werden Änderungen vorgenommen und eine neue Skriptversion wird erstellt, um die Schemaänderung zu behandeln.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 5: Durchführen einer Schemaänderung in der entfernten Datenbank“](#) auf Seite 282.

Lektion 5: Durchführen einer Schemaänderung in der entfernten Datenbank

In dieser Lektion ändern Sie die entfernte Datenbank, um eine neue Spalte in die Customer-Tabelle einzufügen und die zur Synchronisation verwendete Skriptversion zu ändern.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Änderung eines Schemas mit der erweiterten Option ScriptVersion“](#) auf Seite 277.

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Erstellen und Konfigurieren der konsolidierten Datenbank“](#) auf Seite 278.

Bevor Sie die entfernte Datenbank und die zur Synchronisation verwendete Skriptversion ändern, müssen Sie sicherstellen, dass keine Vorgänge für die Customer-Tabelle laufen, die hochgeladen werden muss.

Die einfachste Möglichkeit hierzu ist das Ausführen der Änderung im `sp_hook_dbmlsync_schema_upgrade`-Hook. Wenn Sie diesen Hook verwenden, gewährleistet `dbmlsync`, dass die Schemaänderung sicher erfolgt, indem die Synchronisationstabellen beim Start der Synchronisation gesperrt und die Sperren so lange beibehalten werden, bis die Änderung abgeschlossen ist.

Kontext und Bemerkungen

Vorsicht

Wenn Sie das Schema ändern, obwohl Vorgänge hochgeladen werden müssen, kann die entfernte Datenbank nach einer Schemaänderung keine Synchronisation durchführen.

Aufgabe

1. Erstellen Sie einen `sp_hook_dbmlsync_schema_upgrade`-Hook mit den folgenden SQL-Anweisungen in der entfernten Datenbank. Der Hook fügt eine neue Spalte der Customer-Tabelle hinzu und ändert den Wert der erweiterten Option `ScriptVersion`, die mit der Subskription gespeichert ist. Der Hook wird von `dbmlsync` gelöscht, nachdem er ausgeführt wurde.

```
CREATE PROCEDURE sp_hook_dbmlsync_schema_upgrade()  
BEGIN  
    ALTER TABLE customer  
        ADD cell_phone varchar(12) default null;  
  
    ALTER SYNCHRONIZATION SUBSCRIPTION my_sub  
        ALTER OPTION ScriptVersion='my_ver2';  
  
    UPDATE #hook_dict  
        SET value = 'always'  
        WHERE name = 'drop hook';  
END;
```

2. Synchronisieren Sie, um Vorgänge hochzuladen, die hochgeladen werden müssen, und um die Schemaänderung durch Ausführen des `sp_hook_dbmlsync_schema_change`-Hooks durchzuführen. Führen Sie den folgenden Befehl aus.

```
dbmlsync -v+ -ot sync2.txt -c UID=DBA;PWD=sql;SERVER=remote -s my_sub -k
```

Nach dieser Synchronisation empfiehlt sich ein Blick in das `dbmlsync`-Log `sync2.txt`, um sicherzugehen, dass keine Fehlermeldungen einen Fehlschlag der Schemaänderung anzeigen.

Ergebnisse

Die Änderung des Schemas ist nun abgeschlossen und Sie können die Synchronisation auf die übliche Weise fortsetzen.

Nächste Schritte

Keine.

Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm

Diese praktische Einführung zeigt, wie Sie mit dem mlreplay-Dienstprogramm mehrere MobiLink-Clients auf einem einzelnen Computer simulieren.

Erforderliche Software

- SQL Anywhere 16

Privilegien

Sie müssen die folgenden Rollen und Privilegien für die konsolidierte Datenbank haben:

- SYS_AUTH_RESOURCE_ROLE-Kompatibilitätsrolle
- MONITOR-Systemprivileg

Sie müssen die folgenden Rollen und Privilegien für die entfernte Datenbank haben:

- SYS_REPLICATION_ADMIN_ROLE-Systemrolle
- SYS_RUN_REPLICATION_ROLE-Systemrolle

Kenntnisse und Erfahrungen

Hierzu benötigen Sie Folgendes:

- Grundkenntnisse über MobiLink-Ereignisskripten

Übersicht

In dieser praktischen Einführung wird Folgendes veranschaulicht:

- Einrichten einer konsolidierten Datenbank
- Starten des MobiLink-Servers zur Aufzeichnung und Wiedergabe von Synchronisationen
- Verwenden des mlreplay-Dienstprogramms zum Simulieren von MobiLink-Clients

Siehe auch

- „Dienstprogramm MobiLink Replay (mlreplay)“ [[MobiLink - Serveradministration](#)]

Lektion 1: Einrichten der konsolidierten MobiLink-Datenbank

In dieser Lektion richten Sie eine konsolidierte MobiLink-Datenbank ein.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm](#)“ auf Seite 284.

Aufgabe

1. Erstellen Sie ein neues Arbeitsverzeichnis zum Speichern aller Beispieldateien, die Sie in dieser praktischen Einführung erstellt haben.

Diese praktische Einführung geht vom Pfad `c:\mlreplay` aus.

2. Wechseln Sie an einer Eingabeaufforderung zum Verzeichnis `c:\mlreplay`.

In dieser praktischen Einführung wird davon ausgegangen, dass alle Befehle über dieses Verzeichnis ausgeführt werden.

3. Führen Sie zum Erstellen einer konsolidierten SQL Anywhere-Datenbank den Befehl `cons.db` aus.

```
dbinit -dba DBA,sql cons.db
```

4. Führen Sie den folgenden Befehl aus, um die konsolidierte Datenbank zu starten:

```
dbsrv16 cons.db
```

5. Wählen Sie **Start » Programme » SQL Anywhere 16 » Administrationstools » ODBC-Datenquellen-Administrator**.

6. Klicken Sie auf der Registerkarte **Benutzer-DSN** auf **Hinzufügen**.

7. Klicken Sie im Fenster **Neue Datenquelle erstellen** auf **SQL Anywhere 16** und auf **Fertig stellen**.

8. Führen Sie im Fenster **ODBC-Konfiguration für SQL Anywhere** folgende Aufgaben aus:

- a. Klicken Sie auf die Registerkarte **ODBC**.
- b. Im Feld **Datenquellenname** geben Sie `cons` ein.
- c. Klicken Sie auf die Registerkarte **Login**.
- d. Im Feld **Benutzer-ID** geben Sie `DBA` ein.
- e. Im Feld **Kennwort** geben Sie `sql` ein.
- f. Wählen Sie in der Dropdown-Liste **Aktion** die Option **Mit einer laufenden Datenbank auf diesem Computer verbinden**.
- g. Im Feld **Servename** geben Sie `cons` ein.
- h. Im Feld **Datenbankname** geben Sie `cons` ein.
- i. Klicken Sie auf **OK**.

9. Schließen Sie den ODBC-Datenquellen-Administrator.

Klicken Sie auf **OK** im Fenster **ODBC-Datenquellen-Administrator**.

10. Verbinden Sie sich über Interactive SQL mit Ihrer konsolidierten Datenbank.

Führen Sie den folgenden Befehl aus:

```
dbisql -c "DSN=cons"
```

11. Führen Sie die folgende Anweisung in Interactive SQL aus, um mit dem Skript *syncsa.sql* MobiLink-Systemtabellen und gespeicherte Prozeduren zu erstellen. Ersetzen Sie *C:\Program Files\SQL Anywhere 16* durch den Pfad Ihrer SQL Anywhere 16-Installation.

```
READ "C:\Program Files\SQL Anywhere 16\MobiLink\setup\syncsa.sql";
```

Interactive SQL wendet *syncsa.sql* auf Ihre konsolidierte Datenbank an.

Durch das Ausführen von *syncsa.sql* wird eine Reihe von Systemtabellen und gespeicherten Prozeduren mit dem Präfix **ml_** erstellt. Der MobiLink-Server arbeitet während des Synchronisationsprozesses mit diesen Tabellen und gespeicherten Prozeduren.

12. Führen Sie die nachstehende SQL-Anweisung aus, um die Tabelle **T1** zu füllen:

```
CREATE TABLE T1 (  
    pk1      INTEGER,  
    pk2      INTEGER,  
    c1       VARCHAR(30000),  
    PRIMARY KEY(pk1, pk2)  
);
```

Interactive SQL erstellt die **T1**-Tabelle in Ihrer konsolidierten Datenbank.

13. Schließen Sie Interactive SQL.

Ergebnisse

Die konsolidierte Datenbank wird eingerichtet.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 2: Erstellen des MobiLink-Projekts](#)“ auf Seite 286.

Lektion 2: Erstellen des MobiLink-Projekts

In dieser Lektion verbinden Sie sich mit der konsolidierten Datenbank, indem Sie ein neues MobiLink-Projekt erstellen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten der konsolidierten MobiLink-Datenbank](#)“ auf Seite 284.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm](#)“ auf Seite 284.

Aufgabe

1. Klicken Sie auf **Start » Programme » SQL Anywhere 16 » Administrationstools » Sybase Central**.
2. Klicken Sie auf **Extras » MobiLink 16 » Neues Projekt**.
3. Im Feld **Name** geben Sie **mlreplay_project** ein.
4. Geben Sie im Feld **Speicherort** den Pfad **C:\mlreplay** ein und klicken Sie dann auf **Weiter**.
5. Im Feld **Anzeigenname der Datenbank** geben Sie **cons** ein.
6. Klicken Sie auf **Bearbeiten**. Das Fenster **Mit einer allgemeinen ODBC-Datenbank verbinden** wird angezeigt.
7. Im Feld **Benutzer-ID** geben Sie **DBA** ein.
8. Im Feld **Kennwort** geben Sie **sql** ein.
9. Klicken Sie im Feld **ODBC-Datenquellenname** auf **Durchsuchen** und wählen Sie **cons**.
10. Klicken Sie auf **OK** und auf **Speichern**.
11. Prüfen Sie die Option **Kennwort speichern** und klicken Sie auf **Weiter**. Akzeptieren Sie dabei alle Standardwerte bis zum Ende des Assistenten.
12. Klicken Sie auf **Fertig stellen**.

Ergebnisse

Das MobiLink-Projekt wird erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 3: Hinzufügen von Synchronisationsskripten](#)“ auf Seite 287.

Lektion 3: Hinzufügen von Synchronisationsskripten

Jedes Skript ist einer bestimmten Skriptversion zugeordnet. Vor dem Hinzufügen von Skripten müssen Sie der konsolidierten Datenbank eine Skriptversion hinzufügen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten der konsolidierten MobiLink-Datenbank](#)“ auf Seite 284.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm](#)“ auf Seite 284.

Kontext und Bemerkungen

Sie können Synchronisationsskripten mithilfe von Sybase Central anzeigen, schreiben und ändern. In dieser Lektion schreiben Sie die folgenden Synchronisationsskripten:

- **upload_insert** Dieses Ereignis definiert, wie neue clientseitige Daten in die konsolidierte Datenbank übernommen werden sollen.
- **download_cursor** Dieses Ereignis definiert die Daten, die auf entfernte Clients heruntergeladen werden sollen.
- **download_delete_cursor** Dieses Ereignis ist erforderlich, wenn Sie Synchronisationsskripten verwenden, die nicht nur einen reinen Upload ausführen. Der MobiLink-Server wird konfiguriert, dieses Ereignis für diese praktische Einführung zu ignorieren.

Aufgabe

1. Klicken Sie auf **Ansicht » Ordner**.
2. Erweitern Sie im linken Fensterausschnitt von Sybase Central unter **MobiLink 16 mlreplay_project, Konsolidierte Datenbanken, cons - DBA**.
3. Rechtsklicken Sie auf **Versionen** und wählen Sie **Neu » Version**.
4. Im Feld **Wie soll der Name der neuen Skriptversion lauten?** geben Sie **MLReplayDemo** ein.
5. Klicken Sie auf **Fertig stellen**.
6. Erweitern Sie im linken Fensterausschnitt von Sybase Central unter **MobiLink 16 mlreplay_project, Konsolidierte Datenbanken, cons - DBA**.
7. Rechtsklicken Sie auf **Synchronisierte Tabellen** und klicken Sie auf **Neu » Synchronisierte Tabelle**.
8. Klicken Sie auf die Option **Eine Tabelle in der konsolidierten Datenbank mit dem Namen der entfernten Tabelle wählen**.
9. Klicken Sie in der Liste **Welchem Benutzer gehört die Tabelle, die synchronisiert werden soll?** auf **DBA**.
10. Klicken Sie in der Liste **Welche Tabelle soll synchronisiert werden?** auf **T1**.
11. Klicken Sie auf **Fertig stellen**.

Die Tabelle **T1** wird als Synchronisationstabelle registriert und Sie können Skripten dieser Tabelle hinzufügen.

12. Erweitern Sie im linken Fensterausschnitt von Sybase Central unter **MobiLink 16** die Ordner **mlreplay_project, Konsolidierte Datenbanken, cons - DBA, Synchronisierte Tabellen**.
13. Klicken Sie mit der rechten Maustaste auf die Tabelle **T1** und klicken Sie auf **Neu » Tabellenskript**.

14. Klicken Sie in der Liste **Für welche Version soll das Tabellenskript erstellt werden?** auf **MLReplayDemo**.
15. Klicken Sie in der Liste **Bei welchem Ereignis soll das Tabellenskript ausgeführt werden?** auf **upload_insert** und anschließend auf **Weiter**.
16. Klicken Sie auf **Fertig stellen**.
17. Verwenden Sie im rechten Fensterausschnitt von Sybase Central das folgende SQL-Skript für das **upload_insert**-Ereignis:

```
INSERT INTO T1 VALUES( cast({ml s.remote_id} as INTEGER), {ml r.2}, {ml r.3} );
```

Das Ereignis **upload_insert** legt fest, wie in die entfernte Datenbank eingefügte Daten in der konsolidierten Datenbank angewendet werden. Siehe „[upload_insert \(Tabellenereignis\)](#)“ [[MobiLink - Serveradministration](#)].

18. Klicken Sie auf **Datei » Speichern**.
19. Wiederholen Sie die Schritte 13 bis 16 und geben Sie das **download_cursor**-Ereignis anstelle des **upload_insert**-Ereignisses in Schritt 15 an.
20. Verwenden Sie im rechten Fensterausschnitt von Sybase Central das folgende SQL-Skript für das **download_cursor**-Ereignis:

```
SELECT pk1, pk2, c1 FROM T1;
```

Das Skript **download_cursor** bestimmt den Cursor für die Auswahl der Zeilen aus der konsolidierten Datenbank, die in die entfernte Datenbank heruntergeladen und eingefügt oder aktualisiert werden. Weitere Hinweise zu **download_cursor** finden Sie unter „[download_cursor \(Tabellenereignis\)](#)“ [[MobiLink - Serveradministration](#)].

21. Klicken Sie auf **Datei » Speichern**.
22. Wiederholen Sie die Schritte 13 bis 16 und geben Sie das **download_delete_cursor**-Ereignis anstelle des **upload_insert**-Ereignisses in Schritt 15 an.
23. Verwenden Sie im rechten Fensterausschnitt von Sybase Central das folgende SQL-Skript für das **download_delete_cursor**-Ereignis:

```
--{ml_ignore}
```

24. Klicken Sie auf **Datei » Speichern**.

Ergebnisse

Die Synchronisationsskripten werden erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 4: Starten des MobiLink-Servers für die Aufzeichnung](#)“ auf Seite 290.

Siehe auch

- „Überblick über MobiLink-Ereignisse“ [[MobiLink - Serveradministration](#)]
- „Hinzufügungen und Löschungen von Skripten“ [[MobiLink - Serveradministration](#)]
- „Skripten für den Upload von Zeilen“ [[MobiLink - Serveradministration](#)]
- „Skripten für den Download von Zeilen“ [[MobiLink - Serveradministration](#)]
- „upload_insert (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „upload_update (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „upload_delete (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „download_cursor (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „download_delete_cursor (Tabellenereignis)“ [[MobiLink - Serveradministration](#)]
- „Direkte Zeilenbehandlung“ [[MobiLink - Serveradministration](#)]
- „Direkte Uploads“ [[MobiLink - Serveradministration](#)]
- „Direkte Downloads“ [[MobiLink - Serveradministration](#)]
- „Implementieren zeitstempelbasierter Downloads“ [[MobiLink - Serveradministration](#)]
- „Partitionierte Zeilen in entfernten Datenbanken“ [[MobiLink - Serveradministration](#)]

Lektion 4: Starten des MobiLink-Servers für die Aufzeichnung

In dieser Lektion starten Sie den MobiLink-Server (mlsrv16) mit der Option -c, um eine Verbindung mit der konsolidierten Datenbank herzustellen.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten der konsolidierten MobiLink-Datenbank](#)“ auf Seite 284.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm](#)“ auf Seite 284.

Aufgabe

- Geben Sie folgenden Befehl ein, um eine Verbindung mit der konsolidierten Datenbank herzustellen:

```
mlsrv16 -c "DSN=cons" -zu+ -zs mlreplay_svr -x tcpip -ot mlsrv.mls -v+ -rp .
```

Nachfolgend finden Sie eine Beschreibung der einzelnen in dieser Lektion verwendeten MobiLink-Serveroptionen. Die Optionen -ot - und -v bieten Informationen zur Fehlersuche und -behandlung. Diese Protokollierungsoptionen sind für Entwicklungsumgebungen geeignet. Aus Gründen der Performance wird -v bei der Produktion normalerweise nicht eingesetzt.

Option	Beschreibung
-c	Steht vor der Verbindungszeichenfolge

Option	Beschreibung
-ot	Legt die Meldungslogdatei <i>mlsrv.mls</i> fest.
-v+	Gibt an, welche Informationen protokolliert werden. Mit -v+ wird die maximale ausführliche Protokollierung aktiviert.
-rp	Gibt das Verzeichnis an, in dem Synchronisationen für die Wiedergabe aufgezeichnet werden.
-x	Legt das Protokoll fest, das verwendet wird, um Synchronisationsanforderungen abzuheören.
-zs	Legt einen MobiLink-Servernamen fest.
-zu+	Fügt neue Benutzer automatisch hinzu.

Ergebnisse

Der MobiLink-Server wird gestartet und mit der konsolidierten Datenbank verbunden. Das MobiLink-Server-Meldungsfenster wird eingeblendet.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 5: Einrichten der konsolidierten MobiLink-Clientdatenbank](#)“ auf Seite 291.

Siehe auch

- „MobiLink-Serveroptionen“ [[MobiLink - Serveradministration](#)]

Lektion 5: Einrichten der konsolidierten MobiLink-Clientdatenbank

MobiLink ist für Synchronisationen vorgesehen, an denen ein konsolidierter Datenbankserver und zahlreiche mobile Datenbanken beteiligt sind. In dieser Lektion erstellen Sie eine entfernte Datenbank, eine T1-Tabelle, die Sie mit der konsolidierten Datenbank synchronisieren sowie eine Synchronisationspublikation, einen Benutzer und eine Subskription.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten der konsolidierten MobiLink-Datenbank](#)“ auf Seite 284.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm](#)“ auf Seite 284.

Kontext und Bemerkungen

In dieser Lektion verwenden Sie eine SQL Anywhere-Datenbank für Ihre konsolidierte Datenbank und den MobiLink-Client. Dabei wird davon ausgegangen, dass sich Ihr MobiLink-Client, die konsolidierte Datenbank und der MobiLink-Server auf demselben Computer befinden.

Um die MobiLink-Clientdatenbank einzurichten, erstellen Sie eine **T1**-Tabelle für die entfernte Datenbank. Die **T1**-Tabelle entspricht der **T1**-Tabelle in der konsolidierten Datenbank. Der MobiLink-Server verwendet SQL-basierte Skripten für die Synchronisation von Produktmengen.

Sie erstellen einen Synchronisationsbenutzer, eine Publikation und eine Subskription in der Clientdatenbank, nachdem Sie die Tabellen erstellt haben. Publikationen identifizieren die Tabellen und Spalten in Ihrer entfernten Datenbank, die synchronisiert werden sollen. Diese Tabellen und Spalten heißen **Artikel**. Eine Synchronisationssubskription subskribiert einen MobiLink-Benutzer für eine Publikation.

Aufgabe

1. Erstellen Sie Ihre MobiLink Clientdatenbanken mit dem Befehlszeilen-Dienstprogramm dbinit.

Geben Sie den folgenden Befehl ein, um die **remote**-Datenbank zu erstellen:

```
dbinit -dba DBA,sql remote.db
```

2. Starten Sie Ihre MobiLink-Clientdatenbank mit dem Befehlszeilen-Dienstprogramm dbeng16.

Führen Sie den folgenden Befehl aus, um die **remote**-Datenbank zu starten:

```
dbeng16 remote
```

3. Stellen Sie mit Interactive SQL eine Verbindung mit der entfernten Datenbank her.

Führen Sie den folgenden Befehl aus:

```
dbisql -c "SERVER=remote;UID=DBA;PWD=sql"
```

4. Erstellen Sie die **T1**-Tabelle für die **remote**-Datenbank.

Führen Sie hierzu die folgende SQL-Anweisung in Interactive SQL aus:

```
CREATE TABLE T1 (  
    pk1      INTEGER,  
    pk2      INTEGER,  
    c1       VARCHAR(30000),  
    PRIMARY KEY(pk1,pk2)  
);  
  
SET OPTION PUBLIC.ml_remote_id = '0';
```

5. Erstellen Sie Ihren MobiLink-Synchronisationsbenutzer sowie die Subskription für die **remote**-Datenbank.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
CREATE PUBLICATION P1 ( TABLE T1 );
CREATE SYNCHRONIZATION USER U1;
CREATE SYNCHRONIZATION SUBSCRIPTION TO P1 FOR U1 TYPE 'TCPIP' ADDRESS
'host=localhost;port=2439';
```

6. Lassen Sie Interactive SQL für die nächste Lektion geöffnet.

Ergebnisse

Eine entfernte Datenbank, eine **T1**-Tabelle, eine Synchronisationspublikation, ein Benutzer und eine Subskription werden erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 6: Aufzeichnen der Synchronisation](#)“ auf Seite 293.

Siehe auch

- „Dienstprogramm Initialisierung (dbinit)“ [*SQL Anywhere Server - Datenbankadministration*]
- „MobiLink-Clients“ [*MobiLink - Clientadministration*]
- „CREATE SYNCHRONIZATION USER-Anweisung [MobiLink]“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]
- „CREATE PUBLICATION-Anweisung [MobiLink] [SQL Remote]“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]
- „CREATE SYNCHRONIZATION SUBSCRIPTION-Anweisung [MobiLink]“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]

Lektion 6: Aufzeichnen der Synchronisation

In dieser Lektion führen Sie das dbmlsync-Dienstprogramm aus, um die MobiLink-Synchronisation für entfernte SQL Anywhere-Datenbanken zu initiieren.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten der konsolidierten MobiLink-Datenbank](#)“ auf Seite 284.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm](#)“ auf Seite 284.

Aufgabe

1. Führen Sie die erste aufgezeichnete Synchronisation aus, damit das Schema im Cache auf dem MobiLink-Server zwischengespeichert wird.

Führen Sie den folgenden Befehl aus, um die **remote**-Datenbank zu synchronisieren:

```
dbmlsync -c "SERVER=remote;UID=DBA;PWD=sql" -ot remotel.mls -e
"sv=MLReplayDemo" -v+
```

Die folgende Tabelle enthält eine Beschreibung für jede dbmlsync-Option, die in dieser Lektion verwendet wurde:

Option	Beschreibung
-c	Legt die Verbindungszeichenfolge fest.
-ot	Gibt die Datei zur Protokollierung von Meldungen an.
-e	Gibt die Skriptversion für die Synchronisation an.
-v+	Gibt an, welche Informationen protokolliert werden. Mit -v+ wird die maximale ausführliche Protokollierung aktiviert.

Eine Ausgabeanzeige erscheint und weist darauf hin, dass die Synchronisation erfolgreich verlaufen ist. Bei der SQL-basierten Synchronisation wurden Zeilen aus der **T1**-Tabelle des Clients in die **T1**-Tabelle der konsolidierten Datenbank übertragen.

2. Bereiten Sie die entfernte Datenbank für die Einfügung von Daten vor, sodass eine zweite Synchronisation durchgeführt wird.

Sie sollten jedoch immer noch eine Verbindung mit der **entfernten** Datenbank mittels Interactive SQL aufrecht erhalten. Wenn dies nicht der Fall ist, führen Sie den folgenden Befehl aus, um eine Verbindung zur Datenbank **remote** herzustellen:

```
dbisql -c "SERVER=remote;UID=DBA;PWD=sql"
```

3. Laden Sie Daten in die **remote**-Datenbank für den Upload zum MobiLink-Server während der Wiedergabesitzung.

Führen Sie die folgende SQL-Anweisung in Interactive SQL aus:

```
INSERT INTO T1 (pk1,pk2,c1) values (0,1,'data1');
INSERT INTO T1 (pk1,pk2,c1) values (0,2,'data2');
INSERT INTO T1 (pk1,pk2,c1) values (0,3,'data3');
INSERT INTO T1 (pk1,pk2,c1) values (0,4,'data4');
INSERT INTO T1 (pk1,pk2,c1) values (0,5,'data5');
INSERT INTO T1 (pk1,pk2,c1) values (0,6,'data6');
INSERT INTO T1 (pk1,pk2,c1) values (0,7,'data7');
INSERT INTO T1 (pk1,pk2,c1) values (0,8,'data8');
INSERT INTO T1 (pk1,pk2,c1) values (0,9,'data9');
INSERT INTO T1 (pk1,pk2,c1) values (0,10,'data10');
COMMIT;
```

4. Führen Sie die zweite aufgezeichnete Synchronisation aus. Dies ist das Protokoll, das wiedergegeben wird.

Führen Sie den folgenden Befehl aus, um die **remote**-Datenbank zu synchronisieren:

```
dbmlsync -c "SERVER=remote;UID=DBA;PWD=sql" -ot remote2.mls -e
"sv=MLReplayDemo" -v+
```

Ergebnisse

Die Datenbanken werden synchronisiert.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 7: Starten von MobiLink-Server für die Wiedergabe](#)“ auf Seite 295.

Siehe auch

- „SQL Anywhere-Clients“ [[MobiLink - Clientadministration](#)]
- „MobiLink SQL Anywhere Client-Dienstprogramm (dbmlsync)“ [[MobiLink - Clientadministration](#)]

Lektion 7: Starten von MobiLink-Server für die Wiedergabe

In dieser Lektion stoppen Sie den MobiLink-Server, um die Aufzeichnung zu beenden, und Sie starten dann den Server neu, ohne die Option -rp zur Vorbereitung des Servers für die Wiedergabe zu verwenden.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten der konsolidierten MobiLink-Datenbank](#)“ auf Seite 284.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm](#)“ auf Seite 284.

Aufgabe

1. Führen Sie den folgenden Befehl aus, um den MobiLink-Server **mlreplay_svr** zu stoppen:

```
mlstop -w -t 1m mlreplay_svr
```

Der MobiLink-Server wird zusammen mit der Aufzeichnung der Synchronisation beendet.

Die folgende Tabelle enthält eine Beschreibung für jede Option, die in dieser Lektion verwendet wurde:

Option	Beschreibung
-w	Wartet, bis der Server herunterfährt, bevor eine Rückkehr zur Eingabeaufforderung erfolgt.
-t	Gibt an, dass der Server entweder nach einer Minute oder nach Abschluss der aktuellen Synchronisationen heruntergefahren werden soll, je nachdem, was früher eintritt.

2. Geben Sie folgenden Befehl ein, um eine Verbindung mit der konsolidierten Datenbank herzustellen:

```
mlsrvl6 -c "DSN=cons" -zu+ -zs mlreplay_svr -x tcpip -ot  
server_replay.mls -v+
```

Das MobiLink-Server-Meldungsfenster wird eingeblendet.

Nachfolgend finden Sie eine Beschreibung der einzelnen in dieser Lektion verwendeten MobiLink-Serveroptionen. Die Optionen -ot - und -v bieten Informationen zur Fehlersuche und -behandlung. Diese Protokollierungsoptionen sind für Entwicklungsumgebungen geeignet. Aus Gründen der Performance wird -v bei der Produktion normalerweise nicht eingesetzt.

Option	Beschreibung
-c	Legt die Verbindungszeichenfolge fest.
-ot	Legt die Meldungslogdatei <i>server_replay.mls</i> fest.
-v+	Gibt an, welche Informationen protokolliert werden. Mit -v+ wird die maximale ausführliche Protokollierung aktiviert.
-x	Legt das Protokoll fest, das verwendet wird, um Synchronisationsanforderungen abzuheören.
-zs	Legt einen MobiLink-Servernamen fest.
-zu+	Fügt neue Benutzer automatisch hinzu.

Ergebnisse

Der MobiLink-Server wird gestoppt und dann wieder gestartet.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 8: Wiedergabe der Synchronisation“ auf Seite 296](#).

Siehe auch

- [„MobiLink-Serveroptionen“ \[MobiLink - Serveradministration\]](#)

Lektion 8: Wiedergabe der Synchronisation

In dieser Lektion führen Sie eine Synchronisation aus, damit das Schema im Cache auf dem MobiLink-Server zwischengespeichert wird. Sie erstellen die simulierte Clientinformationsdatei, um die MobiLink-Protokollinformationen über die simulierten Clients wiederzugeben.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Einrichten der konsolidierten MobiLink-Datenbank“ auf Seite 284](#).

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: [„Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm“ auf Seite 284](#).

Kontext und Bemerkungen

Die Informationsdatei für simulierte Clients ist nur erforderlich, wenn die Wiedergabe des aufgezeichneten Protokolls für mehrere simulierte Clients gleichzeitig erfolgt.

Aufgabe

1. Führen Sie den folgenden Befehl aus, um die **remote**-Datenbank zu synchronisieren:

```
dbmlsync -c "SERVER=remote;UID=DBA;PWD=sql" -ot remote3.mls -e  
"sv=MLReplayDemo" -v+
```

Die folgende Tabelle enthält eine Beschreibung für jede dbmlsync-Option, die in dieser Lektion verwendet wurde:

Option	Beschreibung
-c	Legt die Verbindungszeichenfolge fest.
-ot	Gibt die Datei zur Protokollierung von Meldungen an.
-e	Gibt die Skriptversion für die Synchronisation an.
-v+	Gibt an, welche Informationen protokolliert werden. Mit -v+ wird die maximale ausführliche Protokollierung aktiviert.

Eine Ausgabeanzeige erscheint und weist darauf hin, dass die Synchronisation erfolgreich verlaufen ist. Bei der SQL-basierten Synchronisation wurden Zeilen aus der **T1**-Tabelle des Clients in die **T1**-Tabelle der konsolidierten Datenbank übertragen.

2. Erstellen Sie eine Informationsdatei für simulierte Clients zur Verwendung mit dem mlreplay-Dienstprogramm.

Erstellen Sie eine neue Textdatei und schreiben Sie die folgende Liste mit Trennkommas wie angezeigt:

```
mlreplay1,,1,  
mlreplay2,,2,  
mlreplay3,,3,  
mlreplay4,,4,  
mlreplay5,,5,  
mlreplay6,,6,  
mlreplay7,,7,  
mlreplay8,,8,  
mlreplay9,,9,  
mlreplay10,,10,
```

3. Speichern Sie die Datei als *mlreplay.csv* in Ihrem Arbeitsverzeichnis.

Mit der Informationsdatei für Clients können zehn entfernte Clients simuliert werden.

4. Geben Sie die aufgezeichnete Synchronisation mit simulierten Clients wieder.

Führen Sie den folgenden Befehl aus:

```
mlreplay -ap -x tcpip -ot mlreplay.mls -sci mlreplay.csv  
recorded_protocol_mlreplay_svr_2.mlr
```

Die folgende Tabelle enthält eine Beschreibung für jede Option, die in dieser Lektion verwendet wurde:

Option	Beschreibung
-ap	Passt den Fortschritt der in einer Wiedergabesitzung wiedergegebenen Synchronisationen an, damit das mlreplay-Dienstprogramm keine Warnungen wegen nicht übereinstimmenden Offsets auf dem MobiLink-Server verursacht.
-x	Legt das Protokoll fest, das verwendet wird, um Synchronisationsanforderungen abzuheören.
-ot	Gibt die Datei zur Protokollierung von Meldungen an.
-sci	Gibt den Speicherort der Informationsdatei des Clients an.

Das mlreplay-Dienstprogramm speichert Informationen vom Anfang bis zum Ende der Verbindung in einer aufgezeichneten Protokolldatei namens *recorded_protocol_mlreplay_svr_2.mlr*.

- Öffnen Sie die *mlreplay.mls*-Logdatei mit einem Texteditor, um das Ergebnis der MobiLink-Wiedergabe anzusehen.

Ergebnisse

Eine Synchronisation wird durchgeführt, das Schema wird im Cache auf dem MobiLink-Server zwischengespeichert und eine Informationsdatei für simulierte Clients wird erstellt, um die MobiLink-Protokolldaten über die simulierten Clients wiederzugeben.

Nächste Schritte

Gehen Sie weiter zu „[Aufräumen](#)“ auf Seite 298.

Siehe auch

- „Dienstprogramm MobiLink Replay (mlreplay)“ [[MobiLink - Serveradministration](#)]

Aufräumen

Entfernen Sie die Daten der praktischen Einführung von Ihrem Computer.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Einrichten der konsolidierten MobiLink-Datenbank](#)“ auf Seite 284.

In dieser Lektion wird davon ausgegangen, dass Sie die Rollen und Privilegien haben, die im Abschnitt "Privilegien" am Anfang dieser praktischen Einführung aufgeführt sind: „[Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm](#)“ auf Seite 284.

Aufgabe

1. Schließen Sie die Fenster von Interactive SQL, SQL Anywhere, MobiLink und vom Synchronisationsclient, indem Sie mit der rechten Maustaste auf die einzelnen Anzeigen in der Taskleiste klicken und **Schließen** wählen.
2. Löschen Sie alle Datenquellen im Zusammenhang mit der praktischen Einführung:
 - a. Starten Sie den ODBC-Datenquellen-Administrator.
 - b. Wählen Sie **Start » Programme » SQL Anywhere 16 » Administrationstools » ODBC-Datenquellen-Administrator**.
 - c. In der Liste **Benutzerdatenquellen** wählen Sie **cons** und klicken Sie dann auf **Entfernen**.
3. Löschen Sie das Verzeichnis mit Ihren konsolidierten und entfernten Datenbanken.

Ergebnisse

Die Daten der praktischen Einführung werden von Ihrem Computer entfernt.

Index

Symbole

.NET

MobiLink-Server-API, Vorteile,12

.NET-Synchronisationslogik

Info,12

A

Aktualisieren von Schemata

erneutes Deployment eines Modells,54

Aktualisieren, Schemata

Assistent zur Schemaaktualisierung,49

Änderung des entfernten Schemas, praktische Einführung

SCRIPT VERSION-Klausel,268

ScriptVersion, erweiterte Option,277

Anwendungen

gelegentlich verbunden,7

Smart Client,7

Assistent zur Schemaaktualisierung

Info,49

authenticate_user

Sybase Central,48

Authentifizierung auf externen Servern

MobiLink,48

B

Batchdateien

Synchronisationsmodell,54

Beispielanwendungen

MobiLink, CustDB-Anwendung,60

Beispieldatenbank

MobiLink, CustDB-Anwendung,60

Beispiele

Contact, MobiLink-Beispiel,75

MobiLink,7

MobiLink, CustDB-Anwendung,60

C

CodeXchange

MobiLink-Beispiele,7

COMMIT-Anweisung

MobiLink-Warnung,17

Contact, MobiLink-Beispiel

ausführen,77

Benutzer,80

Contact, Tabelle,83

Customer, Tabelle,82

einrichten,76

Product, Tabelle,85

SalesRep, Tabelle,81

Statistiken überwachen,87

Tabellen,78

custase.sql

Position,61

CustDB

MobiLink, Beispielanwendung,60

MobiLink, Benutzer,69

MobiLink, ULProduct, Tabelle,73

SQL Anywhere CustDB, konsolidierte Datenbank,61

wiederherstellen,75

CustDB, Anwendung

DB2,62

Synchronisationsskripten,61

CustDB, MobiLink-Beispiel

Tabellen,66

ULCustomer, Tabelle,72

ULOrder, Tabelle,70

custdb.sqc

Position,65

custmss.sql

Position,61

custora.sql

Position,61

D

Datenbanken

mit MobiLink synchronisieren,1

Datenbewegungstechnologien

MobiLink-Synchronisation,1

DB2

CustDB-Anwendung,62

Deadlocks

Upload-Verarbeitung durch MobiLink,19

Deployment eines Synchronisationsmodells, Assistent Info,50

Deployment vornehmen

Synchronisationsmodell-Batchdateien,54

Deployment, erneutes

Synchronisationsmodelle,54

- Direkte Zeilenbehandlung
 - praktische Einführung,172
 - praktische Einführung in XML,219
 - praktische Einführung, Microsoft Excel,199
- Download-Teilmenge
 - Info,41
- Download-Teilmengen
 - ändern in einem Synchronisationsmodell,42
- Downloads
 - MobiLink-Definition,14
 - MobiLink-Transaktionen,17
- Downloadtypen
 - ändern in einem Synchronisationsmodell,38

E

- Einrichten
 - MobiLink mit dem Assistenten zum Erstellen eines Synchronisationsmodells,30
- Einrichten eines MobiLink-Projekts
 - Sybase Central-Aufgabe,24
- Einrichten eines Synchronisationsmodells
 - Sybase Central, Aufgabe,30
- Einrichtung
 - MobiLink-Synchronisation,5
- Entfernte Schemata
 - Synchronisationsmodelle,32
- Ereignisse
 - MobiLink, Einführung,15
- Ereignisse-Registerkarte
 - Synchronisationsmodell,47
- Erste Orientierung
 - MobiLink,5
- Erstellen eines Synchronisationsmodells, Assistent
 - Syntax,30
- Excel
 - MobiLink, praktische Einführung,199
- Externe Server
 - Authentifizierung bei einem Synchronisationsmodell,48

F

- Fehler
 - MobiLink-Synchronisation, Wiederherstellung,18
- Fehlerbehandlung
 - MobiLink-Synchronisationsfehler,18
- Fehlertoleranz
 - Info,18

G

- Gelegentlich verbundene Anwendungen
 - MobiLink,7

I

- IBM DB2
 - CustDB-Anwendung,62
- IMAP-Authentifizierung
 - Synchronisationsmodell,48

J

- Java
 - MobiLink-Server-API, Vorteile,12
 - Synchronisationslogik,12
- Java-Synchronisationslogik
 - Info,12

K

- Kaskadierendes Löschen
 - MobiLink-Synchronisation,20
- Kommunikationsfehler
 - MobiLink-Synchronisation, Wiederherstellung,18
- Konflikterkennung und Konfliktlösung
 - ändern in einem Synchronisationsmodell,46
 - in einem Synchronisationsmodell,45
- Konfliktlösung
 - Contact, Beispiel,85
 - CustDB, Beispiel,73
- Konsolidierte Datenbanken
 - ändern,26
 - einrichten,25

L

- LDAP
 - Synchronisationsmodell,48
- Logdateien
 - MobiLink,105
- Löschungen
 - ändern in einem Synchronisationsmodell,39

M

- makedbs.cmd
 - Speicherort,61
- Microsoft Excel
 - MobiLink, praktische Einführung,199
- mlreplay-Dienstprogramm

- praktische Einführung,284
- MobiLink
 - Architektur,1
 - ASE, praktische Einführung,142
 - Beispiele,7
 - CustDB-Anwendung,60
 - Funktionen,2
 - Info,1
 - mlreplay, praktische Einführung,284
 - MobiLink mit praktischer Einführung in SQL Anywhere,110
 - Optionen zur Erstellung der Synchronisationslogik,12
 - praktische Oracle-Einführung,123
 - Prozessüberblick,14
 - Prüfung der Systemkonfiguration,26
 - Schnellstart,5
 - Synchronisationsgrundlagen,1
 - zentrale Administration, praktische Einführung,241
- MobiLink 16-Plug In für Sybase Central
 - Info,22
- MobiLink, direkte Zeilenbehandlung
 - praktische Einführung,172
- MobiLink, direkte Zeilenbehandlung mit Microsoft Excel
 - praktische Einführung,199
- MobiLink, direkte Zeilenbehandlung mit XML
 - praktische Einführung,219
- MobiLink-Anwendungen
 - Methodenentwicklung,11
 - Planung,7
- MobiLink-Beispiel "Contact"
 - Info,75
- MobiLink-Download
 - Definition,14
- MobiLink-Ereignisse
 - Einführung,15
- MobiLink-Funktionen
 - Info,2
- MobiLink-Projekt
 - erstellen,24
 - Info,22
- MobiLink-Server
 - erste Orientierung,5
- MobiLink-Server-API für .NET
 - Vorteile,13
- MobiLink-Server-API für Java
 - Vorteile,13

- MobiLink-Server-APIs
 - Vorteile,13
- MobiLink-Skripten
 - Info,15
- MobiLink-Synchronisation
 - custdb-Beispieldatenbank,60
- MobiLink-Synchronisationsprozess
 - Info,5
- MobiLink-Systemkonfiguration
 - Prüfung,26
- MobiLink-Upload
 - Definition,14
 - Verarbeitung,19
- Modelle
 - MobiLink-Synchronisation,30
- Modelle in MobiLink
 - Einschränkungen,57

O

- Online-Anwendungen
 - MobiLink,7
- Optionen zum Schreiben von Synchronisationslogik
 - Info,12
- Oracle
 - MobiLink, praktische Einführung,123

P

- Parallelität
 - Upload-Verarbeitung durch MobiLink,19
- Performance
 - Upload-Verarbeitung durch MobiLink,19
- Plug-Ins
 - MobiLink,22
- POP3-Authentifizierung
 - Synchronisationsmodell,48
- Praktische Einführung
 - MobiLink mit Oracle,123
- Praktische Einführungen
 - Änderung des Schemas mit der erweiterten Option ScriptVersion,277
 - Änderung des Schemas mit Skriptversionsklausel,268
 - Java oder .NET für die benutzerdefinierte Authentifizierung,163
 - mehrere MobiLink-Clients simulieren,284
 - MobiLink mit ASE,142
 - MobiLink mit SQL Anywhere,110

- MobiLink, direkte Zeilenbehandlung,172
- MobiLink, direkte Zeilenbehandlung mit Microsoft Excel,199
- MobiLink, direkte Zeilenbehandlung mit XML,219
- Überwachung von MobiLink-Skripten und Konfliktlösung,89
- zentrale Administration entfernter Datenbanken,241
- Projekte
 - erstellen,24
 - konsolidierte Datenbanken hinzufügen,25
- Protokolle
 - MobiLink-Synchronisation,1
- Prüfung der Systemkonfiguration
 - MobiLink,26
- R**
- Referenzielle Integrität
 - MobiLink-Synchronisation,20
- Referenzielle Integrität und Synchronisation
 - MobiLink-Clients,20
- S**
- Schemaänderungen
 - erneutes Deployment eines Synchronisationsmodells,54
- Schemata
 - erneutes Deployment eines Synchronisationsmodells,54
- Schnellstart
 - MobiLink,5
- SCRIPT VERSION-Klausel
 - Änderung des entfernten Schemas, praktische Einführung,268
- ScriptVersion, erweiterte Option
 - Änderung des entfernten Schemas, praktische Einführung,277
- Sicherheit
 - MobiLink-Überblick,22
- Skripten
 - ändern in einem Synchronisationsmodell,47
 - für eine Tabelle suchen,47
 - MobiLink, Einführung,15
- Smart Client, Anwendungen
 - MobiLink,7
- SQL-Synchronisationslogik
 - Alternativen,12
 - SQL_ROW_DELETED_TO_MAINTAIN_REFERENTIAL_INTEGRITY
 - UltraLite-Synchronisation,20
- Synchronisation
 - Architektur des MobiLink-Systems,1
 - Info zu MobiLink,1
 - MobiLink ASE, praktische Einführung,142
 - MobiLink mit praktischer Einführung in SQL Anywhere,110
 - MobiLink, praktische Oracle-Einführung,123
 - MobiLink-Performance,19
 - MobiLink-Prozessüberblick,14
 - MobiLink-Transaktionen,17
 - Optionen zur Erstellung der Synchronisationslogik,12
 - Schnellstart,5
 - Zeitstempel in MobiLink,17
- Synchronisations-Upload
 - MobiLink-Verarbeitung,19
- Synchronisationsgrundlagen
 - Info,1
- Synchronisationslogik
 - Optionen zur Erstellung,12
- Synchronisationsmethoden
 - custdb-Beispielanwendung,60
 - MobiLink, Beispielanwendung Contact,75
- Synchronisationsmodell
 - testen,51
- Synchronisationsmodelle
 - Aufgaben,33
 - Einführung,30,50
 - Einschränkungen,57
 - entfernen,32
 - entfernte Datenbanken,32
 - erstellen,30
 - Info,30
- Synchronisationsprozess
 - Info,14
- Synchronisationssystem
 - Bestandteile,1
- Synchronisieren
 - per Deployment bereitgestellte Modelle,54
- Synchronisierte Tabellen
 - Hinzufügen von Zuordnungen,34
- Systemausfälle
 - MobiLink-Synchronisation, Wiederherstellung,18
- Systemeinrichtung
 - MobiLink-Prüfung,26

T

Tabellenzuordnungen

Info,34

Teilmengen

ändern in einem Synchronisationsmodell,42

Info,41

Testfenster

MobiLink,51

Transaktionen

MobiLink festschreiben und zurücksetzen,17

während der MobiLink-Synchronisation,17

U

upload_delete

Contact, Beispiel,85

CustDB, Beispiel,73

Uploads

MobiLink-Definition,14

MobiLink-Transaktionen,17

MobiLink-Verarbeitung,19

V

Verarbeitung des Uploads

Info,19

Verfahren bei Synchronisationsfehlern

MobiLink,18

Verteilte Datenbanken

MobiLink-Synchronisation,1

Z

Zentrale Administration von entfernten Datenbanken

Rollen und Privilegien,241

Zuordnen

Tabellen und Spalten,34

Zurücksetzen

MobiLink-Warnung,17

