



SQL Anywhere® 16

Einführung

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
Informationen zur SQL Anywhere-Dokumentation	v
Übersicht über SQL Anywhere 16	1
SQL Anywhere in Frontend-Umgebungen	1
Editionen und Lizenzierung	2
Getrennt lizenzierbare Komponenten	3
Merkmale von SQL Anywhere 16	6
Unterstützte Plattformen	7
Eingabehilfen	8
Datenverwaltungstechnologien	17
Die Komponenten eines Datenbanksystems	17
Das Konzept der relationalen Datenbank	18
Hintergrundinformationen zu SQL Anywhere	22
SQL Anywhere- und UltraLite-Zielplattformen	26
Datenbankszenarios	26
Vielstufige Datenverarbeitungsarchitektur	28
Mehrere Datenbanken auf einem einzelnen Datenbankserver ausführen ...	29
ETL-Funktionen	30
Programmierschnittstellen	31
Überblick über Datenaustauschtechnologien	35
Synchronisationstechnologie im Vergleich	36
Weitergabemethoden	40
Hinweise zur Synchronisationstechnologie	42
Design- und Administrationstools	45
Beispieldatenbanken	49

SQL Anywhere-Beispieldatenbank	49
Die Beispieldatenbankanwendung CustDB	53
 Liste der praktischen Einführungen in SQL Anywhere	 55
 Ressourcen für die erste Orientierung mit SQL Anywhere 16	 59
Ressourcen für die erste Orientierung mit SQL Anywhere Server	60
 Häufig gestellte Fragen - SQL Anywhere	 63
 Index	 69

Über diese Dokumentation

In diesem Handbuch wird SQL Anywhere 16 beschrieben. SQL Anywhere ist ein umfassendes Paket für Datenverwaltung und Datenaustausch, mit dem Sie in kürzester Zeit datenbankgestützte Anwendungen für Server-, Desktop-, Mobil- und fernverwaltete Büroumgebungen entwickeln können.

Informationen zur SQL Anywhere-Dokumentation

Die vollständige SQL Anywhere-Dokumentation ist in vier Formen verfügbar:

Konventionen in dieser Dokumentation

In diesem Abschnitt werden die in dieser Dokumentation verwendeten Konventionen beschrieben.

Betriebssysteme

Um den Verweis auf Betriebssysteme zu vereinfachen, fasst die Dokumentation die Betriebssysteme wie folgt zusammen:

- **Windows**
- **Unix**

Verzeichnis- und Dateinamen

Normalerweise sind Bezugnahmen auf Verzeichnis- und Dateinamen auf allen unterstützten Plattformen ähnlich, wobei geringfügige Unterschiede zwischen den einzelnen Formen möglich sind. In diesen Fällen gelten die Windows-Konventionen. In anderen Fällen, in denen komplexere Details erörtert werden, werden alle relevanten Formen in der Dokumentation aufgeführt.

Diese Konventionen werden verwendet, um die Dokumentation von Verzeichnis- und Dateinamen zu vereinfachen:

- **Verzeichnisse mit Groß- und Kleinbuchstaben** Unter Windows und Unix können Verzeichnis- und Dateinamen Groß- und Kleinbuchstaben enthalten.

In Windows wird die Groß-/Kleinschreibung bei Verzeichnis- und Dateinamen *nicht* berücksichtigt. Es ist durchaus üblich, Verzeichnis- und Dateinamen mit Groß- und Kleinbuchstaben zu verwenden, jedoch werden sie hauptsächlich in Kleinbuchstaben angegeben.

In Unix wird die Groß-/Kleinschreibung bei Verzeichnis- und Dateinamen *immer* berücksichtigt. Eine Mischung von Groß- und Kleinbuchstaben ist nicht üblich. Meistens werden ausschließlich Kleinbuchstaben verwendet.

- **Schrägstriche zur Trennung von Verzeichnis- und Dateinamen** Die Dokumentation verwendet Rückschrägstriche (\) als Trennzeichen für die Bestandteile des Verzeichnisnamens. Auf Unix-Plattformen ersetzen Sie den Rückschrägstrich durch einen Schrägstrich.

- **Programmdateien** Die Dokumentation weist Programmdateinamen mit Windows-Konventionen aus, wobei ein Suffix *.exe* oder *.bat* angehängt wird. Auf Unix-Plattformen haben die Namen ausführbarer Dateien kein Suffix.
- **32-Bit- und 64-Bit-Versionen** Die Dokumentation unterscheidet nicht immer zwischen 32-Bit- und 64-Bit-Versionen der Software. In manchen Fällen bietet die Dokumentation einen Beispieldateipfad mit *bin32* oder *bin64*. Die beiden sind austauschbar und hängen von der Version der Software ab, die installiert wurde. Beispiel: Die Dateipfade *C:\Program Files\SQL Anywhere 16\bin32* und *C:\Program Files\SQL Anywhere 16\bin64* sind gleichwertig.

Befehlszeilen und Shelloberflächensyntax

Die Dokumentation bietet Eingabeaufforderungsbeispiele in allgemeiner Form. Wenn diese Beispiele Zeichen enthalten, die in einer Shell als Sonderzeichen eingestuft werden, muss der Befehl für die betreffende Shell geändert werden. Die Erläuterung der Änderungen würde den Rahmen dieser Dokumentation sprengen, jedoch müssen Parameter, die solche Zeichen enthalten, im Allgemeinen in Klammern eingeschlossen werden oder dem Sonderzeichen muss ein Escapezeichen vorangestellt werden.

- **Klammern und geschweifte Klammern** Einige Eingabeaufforderungsoptionen erfordern einen Parameter, der detaillierte Werteangaben in einer Liste akzeptiert. Die Liste wird in Klammern oder geschweifte Klammern eingeschlossen. In der Dokumentation werden Klammern verwendet. Beispiel:

```
-x tcpip(host=127.0.0.1)
```

Falls Klammern Syntaxprobleme verursachen, ersetzen Sie sie durch geschweifte Klammern:

```
-x tcpip{host=127.0.0.1}
```

Falls beide Formen zu Syntaxproblemen führen, muss der gesamte Parameter in Anführungszeichen gesetzt werden wie von der Shell verlangt:

```
-x "tcpip(host=127.0.0.1)"
```

- **Semikola** Unter Unix müssen Semikola in Anführungszeichen gesetzt werden.
- **Anführungszeichen** Wenn in einem Parameterwert Anführungszeichen angegeben werden müssen, können diese mit der üblichen Verwendung von Anführungszeichen für diesen Parameter kollidieren. Um beispielsweise einen Chiffrierschlüssel anzugeben, dessen Wert Anführungszeichen enthält, müssen Sie diesen in Anführungszeichen setzen und anschließend den eingebetteten Anführungszeichen ein Escapezeichen voranstellen:

```
-ek "my \"secret\" key"
```

In vielen Shells lautet der Wert für den Schlüssel `my "secret" key`.

Online-Ressourcen und zusätzliche Hilfe

Resource	Beschreibung
SAP Sybase SQL Anywhere-Community	Die SAP SQL Anywhere-Community bietet Links zu Blogs von Führungskräften, Mitarbeitern und Entwicklern, die Meinungen und Vorschläge zur Verwendung von SQL Anywhere und damit zusammenhängenden Technologien austauschen. Siehe http://scn.sap.com/community/sybase-sql-anywhere/content .
SQL Anywhere-Forum	Das SQL Anywhere-Forum ist eine Website, auf der Sie Fragen zur SQL Anywhere-Software stellen und bearbeiten sowie Kommentare schreiben und die Fragen und Antworten anderer Benutzer bewerten können. Wenn Sie in diesem Forum eine Nachricht posten, achten Sie darauf, Ihr Problem so detailliert wie möglich darzustellen, und vergessen Sie nicht die Build-Nummer Ihrer Version von SQL Anywhere. Sie finden diese Informationen, indem Sie folgenden Befehl ausführen: dsrv16 -v Siehe http://sqlanywhere-forum.sybase.com . Es gibt auch eine Diskussionsseite der Community unter http://scn.sap.com/community/sybase-sql-anywhere/content?filterID=content~objecttype~object-type[thread]# .
DocCommentX-change	DocCommentXchange ist eine Website, auf der Sie die SAP Sybase SQL Anywhere-Dokumentation lesen, Fragen und Rückmeldungen zur Dokumentation hinterlassen sowie die Fragen und Antworten anderer Benutzer zur Dokumentation lesen können. Siehe http://dcx.sybase.com .
SQL Anywhere - Unterstützte Plattformen und Status des Technischen Supports	Von SAP Sybase SQL Anywhere unterstützte Plattformen. Siehe http://www.sybase.com/detail?id=1002288 .
SQL Anywhere Tech Corner	Die SQL Anywhere Tech Corner bietet Entwicklern einfachen Zugriff auf technische Produktdokumentation. Sie können technische White Papers, FAQs, technische Hinweise, Downloads, Techcasts und mehr durchsuchen, um Antworten auf Ihre Fragen und Lösungen zu häufigen Problemen zu finden. Weitere Hinweise finden Sie unter http://www.sybase.com/developer/library/sql-anywhere-techcorner .
SQL Anywhere .NET Developer Center	Einführung und Antworten auf Fragen im Zusammenhang mit SQL Anywhere und der .NET-Entwicklung Siehe www.sybase.com/developer/library/sql-anywhere-techcorner/microsoft-net .
PHP Developer Center	Einführung in die Skriptsprache PHP (PHP Hypertext-Präprozessor) für Abfragen in Ihrer Datenbank Siehe www.sybase.com/developer/library/sql-anywhere-techcorner/php .

Resource	Beschreibung
SQL Anywhere Windows Mobile Developer Center	Einführung und Antworten auf Fragen im Zusammenhang mit SQL Anywhere und der Windows Mobile-Entwicklung Siehe www.sybase.com/developer/library/sql-anywhere-techcorner/windows-mobile .

Top 10 der empfohlenen Whitepapers

- **SQL Anywhere-Anforderungen an I/O-Vorgänge für Windows und Linux** <http://scn.sap.com/docs/DOC-34090>
- **Kapazitätsplanung mit SQL Anywhere** <http://scn.sap.com/docs/DOC-35069>
- **Probleme der Anwendungsperformance mit SQL Anywhere diagnostizieren** <http://scn.sap.com/docs/DOC-35601>
- **Analytische Funktionen in SQL Anywhere** <http://scn.sap.com/docs/DOC-35603>
- **Optimierung der Performance von Adaptive Server Anywhere in einem WAN** <http://scn.sap.com/docs/DOC-35605>
- **Semantik und Kompatibilität von Transact-SQL Outer Joins** <http://scn.sap.com/docs/DOC-35606>

Frühere Versionen der Dokumentation

Eine Zusammenstellung der neuen Funktionen und Verhaltensänderungen von Version 6.0 zu Version 10.0.1 und später finden Sie unter <http://dcx.sybase.com/html/dbwnen10/dbwnen10.html>.

Die vollständige Dokumentation für SQL Anywhere-Versionen von 10.0.1 bis zur aktuellen Version finden Sie unter <http://dcx.sybase.com>.

Von Version 8.0 bis Version 9.0.2 wurde SQL Anywhere unter dem Namen SQL Anywhere Studio geführt. Die vollständige Dokumentation für diese Versionen finden Sie unter http://manuals.sybase.com/onlinebooks/group-sas/@Generic__CollectionView;pt=group-awarc.

Vor Version 8 hieß SQL Anywhere "Adaptive Server Anywhere". Die vollständige Dokumentation für Version 6 finden Sie unter: http://manuals.sybase.com/onlinebooks/group-awarc/@Generic__CollectionView;pt=group-sas.

Übersicht über SQL Anywhere 16

SQL Anywhere ist ein komplettes Softwarepaket mit Technologien für die Datenverwaltung und den Datenaustausch im Unternehmen, mit dem Sie in kürzester Zeit datenbankgestützte Anwendungen für Server-, Client-, Mobil- und fernverwaltete Büroumgebungen entwickeln können.

SQL Anywhere bietet folgende Funktionen:

- **Datenverwaltungstechnologien** SQL Anywhere stellt Datenbanken der Unternehmensklasse bereit, mit denen Sie alle Herausforderungen anspruchsvoller Anwendungen in unterschiedlichen Frontend-Umgebungen bewältigen können: von Datenbankservern mit höchster Performance, die bei Kunden eingerichtet werden, bei denen Software von Drittanbietern läuft, bis zu einer mobilen Datenbank, die auf Tausenden von Handhelds in einem Unternehmenssystem implementiert werden kann.
- **Datenaustauschtechnologien** Die Datenaustauschtechnologien von SQL Anywhere bewältigen hochkomplexe Anforderungen an den Datenaustausch über mobile und nicht immer verfügbare drahtlose Netze mit Backend-Datenbanken, Anwendungsservern und Nachrichtenverarbeitungssystemen. Außerdem gewährleistet SQL Anywhere mit seinen Technologien für mobiles Messaging und die Synchronisation eine sichere Beförderung und Zustellung von Nachrichten in verteilten und mobilen IT-Anwendungen.
- **Design- und Administrationstools** SQL Anywhere enthält eine Serie von Tools für die Verbesserung des Designs und der Entwicklung von datenbankgestützten Anwendungen und für eine einfachere Verwaltung von Datenbanken und Datenaustauschsystemen.

Siehe auch

- [„Das Konzept der relationalen Datenbank“ auf Seite 18](#)
- [„Überblick über Datenaustauschtechnologien“ auf Seite 35](#)
- [„Design- und Administrationstools“ auf Seite 45](#)

SQL Anywhere in Frontend-Umgebungen

SQL Anywhere-Technologien werden von mehr als 10.000 Kunden in vielen verschiedenen Systemumgebungen eingesetzt. Vier verbreitete Einsatzbereiche von SQL Anywhere sind:

- **Client/Server-Anwendungen** SQL Anywhere ist eine leistungsfähige Datenbanklösung mit Serveranwendungen für 5, 50 oder 500 Benutzer, die bei geringem Wartungsaufwand und Kosteneinsatz hohe Performance ohne komplizierten Installationsaufwand ermöglicht.

SQL Anywhere lässt sich einfach skalieren, um mehrere hundert aktive Benutzer und Hunderte von Gigabyte Daten sowie Hunderte Millionen Datensätze zu unterstützen. Und dennoch stellen die Bedienerfreundlichkeit und die Administrationsfunktionen sicher, dass die Kosten auch bei hohen Performance-Anforderungen in Grenzen bleiben.

Dieses Deploymentmodell funktioniert am besten, wenn die meisten Benutzer im Netzwerk arbeiten.

- **Anwendungen für PCs** SQL Anywhere stellt Funktionen der Unternehmensklasse ohne die komplexen Eigenschaften typischer Unternehmensdatenbanken bereit. Die robuste Zuverlässigkeit und Performance bei gleichzeitig hocheffizienter Nutzung von Speicher und Systemressourcen sorgen dafür, dass die Benutzer von Laptops und PCs oft gar nicht bemerken, dass eine Datenbank hinter einer Anwendung steckt.

Unternehmen betten SQL Anywhere-Datenbanken in ihre Anwendungen ein, weil SQL Anywhere-Datenbanken für den Einsatz in weit verstreuten Umgebungen mit minimalem Administrationsaufwand geeignet sind und geringe Anforderungen an Speicher- und Festplattenkapazität stellen.

- **Standortungebundene Büroanwendungen** Die SQL Anywhere-Architektur für den Datenaustausch ist speziell auf die Herausforderungen zugeschnitten, die bei der Administration und Übertragung von Daten innerhalb und zwischen Bürostandorten und Außendienstmitarbeitern entstehen.

Wenn sich ein Unternehmen für die Datenbank- und Datenaustauschtechnologien von SQL Anywhere entscheidet, kann es standortfernen Mitarbeitern verzögerungsfrei alle Daten zur Verfügung stellen, die sie für eine effiziente Arbeit benötigen, während gleichzeitig die Unternehmenszentrale die kritischen Informationen erhält, die für den laufenden Betrieb des Unternehmens erforderlich sind.

- **Mobile und drahtlose Anwendungen** Die weltweit als branchenführend anerkannte mobile Datenbank SQL Anywhere ermöglicht mobilen Mitarbeitern den Zugriff auf ihre Daten und Unternehmensanwendungen. Unabhängig von der Verbindungsart und dem Anwendungstyp gewährleisten Datenaustauschtechnologien von SQL Anywhere, dass mobile Mitarbeiter die benötigten Informationen jederzeit und immer dann bekommen, wenn sie sie brauchen. Die Mitarbeiter können Informationen abrufen und Transaktionen offline in Warteschlangen halten. Dadurch sparen sie Kommunikationskosten und schonen die Akkus ihrer Geräte bei gleichzeitiger Optimierung der Anwendungen.

Das Unternehmen verlässt sich auf SQL Anywhere und erwartet eine zuverlässige Verwaltung der Daten und mobilen Anwendungen auf Laptops, Handhelds und Smartphones.

Siehe auch

- „Client/Serveranwendungen“ auf Seite 27
- „PC-Anwendungen und eingebettete Datenbanken“ auf Seite 26
- „Konsolidierte und entfernte Datenbanken“ auf Seite 38.
- „UltraLite-Überblick“ [*UltraLite - Datenbankverwaltung*]
- „SQL Anywhere für Windows Mobile“ [*SQL Anywhere Server - Datenbankadministration*]

Editionen und Lizenzierung

SQL Anywhere stellt verschiedene Editionen bereit, die bestimmte getrennt lizenzierbare Komponenten enthalten und die Anzahl der vom Datenbankserver verwendeten CPUs beschränken können. Weitere Hinweise zu Editionen finden Sie unter <http://www.sybase.com/detail?id=1068247> .

Lizenzierung und CPUs

Lizenzierung pro Arbeitsplatz Eine Lizenzierung pro Arbeitsplatz begrenzt die Anzahl der Clientverbindungen zum Datenbankserver. Wenn Ihre Lizenz pro Arbeitsplatz gilt, kann der Netzwerk-Datenbankserver alle verfügbaren CPUs auf Ihrem Computer nutzen, aber nicht mehr als die für die verwendete SQL Anywhere-Edition zulässige Anzahl. Wenn eine CPU mehrere Prozessorkerne oder Threads enthält, können alle Prozessorkerne und Threads der CPU verwendet werden.

Der Personal Datenbankserver ist auf vier Prozessorkerne in einer CPU beschränkt. Wenn ein Prozessorkern mehrere Threads enthält, können alle Threads des Prozessorkerns verwendet werden.

CPU-basierte Lizenzierung Bei CPU-basierter Lizenzierung verwendet der Netzwerk-Datenbankserver maximal die Anzahl von CPUs, die in Ihrer Lizenz angegeben sind, aber nicht mehr als die erlaubte Höchstzahl der von Ihnen verwendeten SQL Anywhere-Edition. Wenn eine CPU mehrere Prozessorkerne oder Threads enthält, können alle Prozessorkerne und Threads der CPU verwendet werden.

Der Personal Datenbankserver ist auf vier Prozessorkerne in einer CPU beschränkt. Wenn ein Prozessorkern mehrere Threads enthält, können alle Threads des Prozessorkerns verwendet werden.

Sie können die Prozessoren, die der Datenbankserver verwendet, mit den folgenden Optionen weiter einschränken:

- Datenbankserveroption "-gt"
- Datenbankserveroption -gta
- Datenbankserveroption "-gtc"
- Systemprozedur "sa_server_option" mit der Option "ProcessorAffinity"

Siehe auch

- „Getrennt lizenzierbare Komponenten“ auf Seite 3
- „Dienstprogramm für die Serverlizenzierung (dblic)“ [*SQL Anywhere Server - Datenbankadministration*]
- SQL Anywhere-Lizenzierung
- „Datenbankserveroption -gt“ [*SQL Anywhere Server - Datenbankadministration*]
- „Datenbankserveroption -gta“ [*SQL Anywhere Server - Datenbankadministration*]
- „Datenbankserveroption -gtc“ [*SQL Anywhere Server - Datenbankadministration*]
- Option "ProcessorAffinity": „sa_server_option-Systemprozedur“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]

Getrennt lizenzierbare Komponenten

Die folgenden Komponenten sind getrennt lizenzierbar und müssen getrennt bestellt werden, wenn sie nicht in Ihrer Edition von SQL Anywhere enthalten sind. Weitere Hinweise zu den verschiedenen Editionen von SQL Anywhere finden Sie unter <http://www.sybase.com/detail?id=1068247>.

Wenn Sie eine separat lizenzierbare Komponente bestellen möchten, gehen Sie zu <http://www.sybase.com/detail?id=1015780>.

SQL Anywhere-Sicherheitsoption

Mit SQL Anywhere können Sie Datenbankdateien, die Synchronisation und die Transportschicht bei der Client/Serverkommunikation stark verschlüsseln.

SQL Anywhere bietet folgende Algorithmen für die starke Verschlüsselung:

Funktion	Enthalten in einer getrennt lizenzierbaren Sicherheitsoption ¹	Enthalten in SQL Anywhere ²
Datenbankverschlüsselung	FIPS-zertifizierte AES	AES
Transportschichtsicherheit	FIPS-zertifizierte RSA	RSA

¹ Software für starke Verschlüsselung mit FIPS-zertifizierter Technologie muss getrennt bestellt werden.

² Starke Verschlüsselung mit AES und RSA ist in SQL Anywhere enthalten und erfordert keine getrennte Lizenz; die Bibliotheken sind aber nicht FIPS-zertifiziert.

RSA- und FIPS-zertifizierte Verschlüsselung steht nicht auf allen Plattformen zur Verfügung. Weitere Hinweise dazu, welche Plattformen welche Verschlüsselungsmethode unterstützen, finden Sie unter <http://www.sybase.com/detail?id=1061806>.

Die Sicherheitsoption stellt Certicom-DLLs bereit, die die Verschlüsselungsalgorithmen implementieren, sowie zusätzliche DLLs, die eine Schnittstelle zwischen SQL Anywhere-Software und den Certicom-Bibliotheken bereitstellen.

Die SQL Anywhere-Sicherheitsoption enthält folgende Komponenten:

- Für Windows-Betriebssysteme: Certicom Security Builder GSE.
Weitere Hinweise finden Sie bei der Nummer 542 unter <http://csrc.nist.gov/cryptval/140-1/140val-all.htm>.
- Für Windows Mobile-Betriebssysteme: Certicom Security Builder GSE.
Weitere Hinweise finden Sie bei der Nummer 316 unter <http://csrc.nist.gov/cryptval/140-1/140val-all.htm>.

Weitere Hinweise zur Verschlüsselung finden Sie unter „Datensicherheit“ [*SQL Anywhere Server - Datenbankadministration*].

SQL Anywhere CAC-Authentifizierungsoption

Für UltraLite-Datenbanken muss die Lizenz für CAC-Authentifizierung getrennt bestellt werden. Zusätzlich erfordert diese Option die SQL Anywhere-Sicherheitsoption.

SQL Anywhere-Option In-Memory-Modus

Für SQL Anywhere-Datenbanken muss die Lizenz für die Verwendung des In-Memory-Modus getrennt bestellt werden.

Hinweise zur Plattform-Unterstützung finden Sie unter [SQL Anywhere - Unterstützte Plattformen und Support-Status](#).

Weitere Hinweise zur Verwendung des In-Memory-Modus finden Sie unter „[Datenbankserveroption - im](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

SQL Anywhere-Option Scale-Out mit Schreibschutz

Für SQL Anywhere-Datenbanken muss die Lizenz für die Verwendung des Scale-Out mit Schreibschutz getrennt bestellt werden. Scale-Out mit Schreibschutz ist eine Konfiguration, mit der Sie Berichterstellungs- oder andere Vorgänge auslagern können, die einen schreibgeschützten Zugriff auf die Datenbank erfordern. Weitere Hinweise zum Scale-Out mit Schreibschutz finden Sie unter „[SQL Anywhere-Scale-Out mit Schreibschutz](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

SQL Anywhere-Option Hochverfügbarkeit

Für SQL Anywhere-Datenbanken müssen Lizenzen für Veritas Cluster Server-Agenten und die Datenbankspiegelung für Failover getrennt bestellt werden.

Hinweise zur Plattform-Unterstützung finden Sie unter [SQL Anywhere - Unterstützte Plattformen und Support-Status](#).

Weitere Hinweise zur Datenbankspiegelung finden Sie unter „[Datenbankspiegelung](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

Weitere Hinweise zu den SQL Anywhere Veritas Cluster Server-Agenten finden Sie unter „[SQL Anywhere Veritas Cluster Server-Agenten](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

Hochverfügbarkeitsoption für MobiLink

Mit der Hochverfügbarkeitsoption für MobiLink können Sie mehrere MobiLink-Datenbankserver in Serverfarmen aus identischen Datenbankservern gruppieren, die den Shared State-Modus verwenden. Wenn MobiLink im Shared State-Modus ausgeführt wird, kann eine entfernte Datenbank daran gehindert werden, gleichzeitig mit mehreren Servern zu synchronisieren, wodurch die Datenintegrität gewährleistet ist. MobiLink im Shared State-Modus ermöglicht auch Lastverteilung und Failover bei der serverinitiierten Synchronisation.

Weitere Hinweise zu MobiLink in einer Serverfarm finden Sie unter „[MobiLink-Server in einer Serverfarm](#)“ [[MobiLink - Serveradministration](#)].

SQL Anywhere-Monitor - Produktionsversion

Die Produktionsversion SQL Anywhere-Monitor muss getrennt bestellt werden. Die Produktionsversion ist für die Verwendung in Deployments und Produktion vorgesehen. Weitere Hinweise zum Monitor finden Sie unter „[SQL Anywhere-Monitor](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

Merkmale von SQL Anywhere 16

Nachstehend werden einige der wichtigsten Merkmale von SQL Anywhere aufgelistet, die Sie mit diesem Softwarepaket nutzen können:

- **Embedding in andere Anwendungen** SQL Anywhere kann in andere Anwendungen einfach eingebettet werden. Dies kombiniert eine hohe Performance mit sehr geringem Speicherbedarf. SQL Anywhere enthält zahlreiche Funktionen für die Selbstverwaltung und Selbstwartung in Frontend-Umgebungen, wie beispielsweise Funktionen, die die Optimierung von Computerressourcen, die Selbstoptimierung für eine verbesserte Performance und die vereinfachte Installation und Unterstützung auf entfernten Computern umfassen.
- **Systemoffenheit** SQL Anywhere ist auf vielen Plattformen verfügbar, darunter Windows, Windows Mobile, Linux, Oracle Solaris, HP-UX, IBM AIX und Mac OS X. Einzigartig an SQL Anywhere ist die Tatsache, dass seine Datenbankdateien zwischen Plattformen kopiert werden können. Zusätzlich bietet SQL Anywhere Unterstützung für BlackBerry, Embedded Linux, Windows Mobile 6 und Java SE-Smartphones, wobei die UltraLite-Datenbanktechnologie für kleine Geräte verwendet wird. SQL Anywhere enthält Unterstützung für viele verbreitete Datenbankschnittstellen wie ODBC, JDBC, ADO.NET, PHP und Perl. Es können viele verbreitete Tools für die Anwendungsentwicklung eingesetzt werden, z.B.: Microsoft Visual Studio, Sybase PowerBuilder, Eclipse sowie zahlreiche Webtools. Gespeicherte Prozeduren können in C/C++, Java, .NET oder Perl geschrieben werden.
- **Optimale Performance als Designkonzept** SQL Anywhere ist speziell dafür entwickelt worden, optimale Performance ohne laufende Systemoptimierung und Administration zu gewährleisten. Funktionen wie dynamische Vorabbelegung von Cache-Speicherplatz, automatisches Generieren von Statistiken, ein ausgereifter Abfrageoptimierer, parallele Abfragenverarbeitung und materialisierte Ansichten bedeuten, dass SQL Anywhere ideal für Umgebungen geeignet ist, die eine hohe Performance erfordern, aber keinen Datenbankadministrator vor Ort haben. On-Line Analytical Processing (OLAP) von SQL Anywhere ermöglicht die Durchführung komplexer Datenanalysen innerhalb einer einzelnen SQL-Anweisung, womit der Wert der Ergebnisse gesteigert und die Performance verbessert wird, da sich die Menge der Abfragen in der Datenbank reduziert.
- **Webbetrieb** Mit der integrierten Unterstützung für HTTP-Server und Webdienste, XML-Funktionen, Volltextsuche und der PHP-Schnittstelle ist SQL Anywhere die ideale Datenbank für Anwendungen auf Webservern.
- **Mobilität** SQL Anywhere stellt Datenbanken der Unternehmensklasse für nutzungsintensive Frontline-Systeme und -Geräte bereit, unabhängig davon, ob die Verbindung mit dem Unternehmenssystem verfügbar ist oder nicht. Seine Synchronisationstechnologien stellen sicher, dass Daten über drahtlose oder verkabelte Netzwerke mit Backend-Datenbanken, Anwendungsservern und Nachrichtenverarbeitungssystemen effizient ausgetauscht werden können.
- **Sicherheit** SQL Anywhere bietet komplette Sicherheit von Ende zu Ende mit starker 128-Bit-Verschlüsselung von Datenbanktabellen, Dateien und Kommunikationsdatenströmen zwischen der Anwendung und der Datenbank sowie im MobiLink-Synchronisationsdatenstrom. SQL Anywhere kann ein Audit des Datenzugriffs durchführen, verfügt über integrierte Benutzerauthentifizierung und kann mit Authentifizierungssystemen von Drittanbietern integriert werden. SQL Anywhere bietet darüber hinaus FIPS-zertifizierte Verschlüsselung über die getrennt lizenzierbare Sicherheitsoption.

Siehe auch

- [SQL Anywhere-Sicherheitsoption auf Seite 4](#)

Unterstützte Plattformen

Die Webseite **SQL Anywhere Supported Platforms and Engineering Support Status** enthält eine Liste der unterstützten Betriebssystem-Plattformen, aufgeteilt nach SQL Anywhere-Version. Hier wird außerdem der Engineering-Unterstützungsstatus für jede SQL Anywhere-Version angegeben. Siehe <http://www.sybase.com/detail?id=1002288>.

Über diese Webseite gelangen Sie auf die folgenden Webseiten, auf denen Sie weitere Informationen zum Support finden, wie zum Beispiel:

Webseitenname	URL	Beschreibung
SQL Anywhere-Komponenten pro Plattform	http://www.sybase.com/detail?id=1061806	Stellt eine Liste der Komponenten bereit, die auf den einzelnen von SQL Anywhere unterstützten Plattformen verfügbar sind. Abgesehen von wenigen Ausnahmen werden Komponenten nicht aufgelistet, die auf allen unterstützten Plattformen verfügbar sind. Zur besseren Lesbarkeit sind die Plattformen nach Betriebssystemanbieter, Betriebssystemname und Prozessorarchitektur zusammengefasst.
SQL Anywhere - Unterstützte Linux-Plattformen	http://www.sybase.com/detail?id=1035824	Liefert Informationen über die Unterstützung von Linux-Plattformen, einschließlich Ausnahmen, Schlüsselkomponenten und getestete Linux-Distributionen.
iAnywhere.com - SQL Anywhere - Unterstützte Kerberos-Clients	http://www.sybase.com/detail?id=1061807	Stellt eine Liste von Kerberos-Clients/Laufzeitumgebungen bereit, die in Verbindung mit SQL Anywhere getestet wurden.
SQL Anywhere - Unterstützte Listener-Plattformen	http://www.sybase.com/detail?id=1061808	Stellt eine Liste der unterstützten Listener-Plattformen für SQL Anywhere bereit.
ODBC-Treiber für MobiLink	http://www.sybase.com/detail?id=1011880	Stellt eine Liste der empfohlenen ODBC-Treiber für verschiedene Versionen des MobiLink-Servers bereit.
SQL Anywhere-Client-Schnittstellen	http://www.sybase.com/detail?id=1068981	Stellt eine Liste der unterstützten Client-Schnittstellen für SQL Anywhere bereit.

Hinweise zu Software-Updates finden Sie unter „Software-Updates“ [[SQL Anywhere Server - Datenbankadministration](#)].

Unterstützung für virtualisierte Umgebungen

Sie können SQL Anywhere auf einer unterstützten Plattform in einer virtualisierten Umgebung ausführen, allerdings mit einigen Einschränkungen.

Eingabehilfen

SQL Anywhere unterstützt Abschnitt 508 des US Federal Rehabilitation Act von 1973 (29 U.S.C. 794d).

Die unten stehenden Tabellen geben an, wie SQL Anywhere die Anforderungen an Eingabehilfen gemäß Abschnitt 508 des US Federal Rehabilitation Act erfüllt, wenn die Software auf den Betriebssystemplattformen Windows x86 oder x64 läuft. Eine Deployment-Komponente für die Eingabehilfen stellt Software bereit, mit deren Hilfe Eingabehilfsfunktionen ermöglicht werden.

Für die Zwecke dieses Dokuments handelt es sich bei den folgenden Komponenten um Komponenten von SQL Anywhere:

- SQL Anywhere-Server
- Sybase Central
- Interactive SQL
- Online-Dokumentation

Komponenten von Drittherstellern

Testumgebung: SQL Anywhere 16.0 (64-Bit-Windows) verwendet sowohl die Eingabehilfetechnologien von Microsoft als auch JAWS Screen Reader von Frontier Technologies.

SQL Anywhere 16.0 Voluntary Product Accessibility Template, Zusammenfassung

Kriterien	Unterstützende Funktionen	Bemerkungen und Erläuterungen
Abschnitt 1194.21, Softwareanwendungen und Betriebssysteme auf Seite 9	Unterstützt alle Funktionen	Funktionen und Fähigkeiten werden möglicherweise auch durch Funktionen und Hilfen der Eingabehilfen von Microsoft Windows ergänzt.
Abschnitt 1194.22, Web-basierte Internet-Informationen und -Anwendungen	Nicht anwendbar	
Abschnitt 1194.23, Telekommunikationsprodukte	Nicht anwendbar	
Abschnitt 1194.24, Video- und Multimediaprodukte	Nicht anwendbar	

Kriterien	Unterstützende Funktionen	Bemerkungen und Erläuterungen
Abschnitt 1194.25, Eigenständige, geschlossene Produkte	Nicht anwendbar	
Abschnitt 1194.26, Desktop-Computer und portable Computer	Nicht anwendbar	
Abschnitt 1194.31, Funktionale Performance-Kriterien auf Seite 13	Mit Ausnahmen unterstützt	Unterstützt Eingabehilfeprodukte von Drittherstellern und Technologien, die durch Microsoft Windows unterstützt werden. Einige Produkte von Drittherstellern sind möglicherweise nicht vollständig kompatibel.
Abschnitt 1194.41, Informationen, Dokumentation und Unterstützung auf Seite 15	Unterstützt	

Abschnitt 1194.21, Softwareanwendungen und Betriebssysteme

Kriterien	Unterstützende Funktionen	Bemerkungen und Erläuterungen
(a) Wenn eine Software für ein System mit Tastatur ausgelegt ist, müssen die Funktionen über die Tastatur ausführbar sein, falls die Funktion selbst oder das Ergebnis der Durchführung einer Funktion als Text erkannt werden kann.	Unterstützt Die Tastaturbedienung ist überall in SQL Anywhere möglich. Tastenkürzel, Kurztasten und Menübefehle stehen in SQL Anywhere zur Verfügung.	SQL Anywhere bietet vollständige Unterstützung für die Eingabehilfefunktionen von Microsoft Windows.

Kriterien	Unterstützende Funktionen	Bemerkungen und Erläuterungen
<p>(b) Anwendungen dürfen aktivierte Funktionen anderer Produkte weder unterbrechen noch deaktivieren, wenn es sich dabei um Eingabehilfefunktionen handelt und diese Funktionen gemäß den Branchenstandards entwickelt und dokumentiert wurden. Außerdem dürfen Anwendungen auch keine aktivierten Funktionen eines beliebigen Betriebssystems unterbrechen oder deaktivieren, wenn es sich bei den Funktionen um Eingabehilfefunktionen handelt und die Bedienungsoberfläche des Anwendungsprogramms für diese Eingabehilfefunktionen vom Hersteller des Betriebssystems dokumentiert wurde und dem Produktentwickler zur Verfügung steht.</p>	<p>Unterstützt</p> <p>SQL Anywhere unterbricht oder deaktiviert keine Eingabehilfefunktionen des Betriebssystems.</p> <p>SQL Anywhere unterstützt große Schriften in Titelleisten, Symbolleisten und den meisten Dialogfeldern. SQL Anywhere bietet weiterhin vollständige Unterstützung für die Eingabehilfefunktionen StickyKeys, FilterKeys, MouseKeys und High Contrast.</p>	<p>Die Eingabehilfen von Microsoft Windows können mit der Software verwendet werden. Bildschirmlesegeräte können mit der Software verwendet werden. Java-basierte Komponenten erfordern Bildschirmlesegeräte, die Java Assistive Technology unterstützen.</p> <p>Die Bildschirmlupe von Microsoft Windows steht zum Vergrößern von Text in Dialogfeldern zur Verfügung.</p> <p>Wenn Sie die Tabulatortaste oder Kurztasten verwenden, wird die Bildschirmlupe möglicherweise nicht immer dem Fokus folgen.</p>
<p>(c) Es muss eine gut erkennbare Anzeige des aktuellen Fokus auf dem Bildschirm bereitgestellt werden, die sich zusammen mit den interaktiven Elementen bewegt, wenn sich der Eingabefokus ändert. Der Fokus muss programmseitig erkennbar gemacht werden, sodass Assistive Technology den Fokus und Fokusänderungen verfolgen kann.</p>	<p>Unterstützt</p> <p>In SQL Anywhere werden sowohl visuelle Bildschirmdarstellung des Fokus auf dem Bildschirm als auch Markierung und Verfolgung bereitgestellt.</p>	<p>Für die Java-basierten Komponenten in SQL Anywhere müssen Bildschirmlesegeräte von Drittherstellern die Java Assistive Technology unterstützen.</p> <p>Die Windows-Bildschirmlupe folgt bei Java-basierten Komponenten möglicherweise nicht immer dem Fokus, wenn die Tabulatortaste oder eine Kurztaste verwendet wird.</p>

Kriterien	Unterstützende Funktionen	Bemerkungen und Erläuterungen
(d) Für Assistive Technology müssen ausreichende Informationen über ein Element der Benutzeroberfläche einschließlich Identität, Vorgang und Status des Elements zur Verfügung gestellt werden. Wenn ein Programmelement durch ein Bild dargestellt wird, müssen die durch das Bild übermittelten Informationen auch in Textform zur Verfügung gestellt werden.	Unterstützt SQL Anywhere unterstützt sowohl die Eingabehilfen von Microsoft® Active Accessibility® als auch die Programmierschnittstelle Java Accessibility, die Kompatibilität mit Assistive Technology ermöglicht.	Für die Java-basierten Komponenten in SQL Anywhere müssen Bildschirmlesegeräte von Drittherstellern die Java Assistive Technology unterstützen.
(e) Wenn Bitmap-Bilder verwendet werden, um Steuerelemente, Statusanzeigen oder andere Programmelemente zu identifizieren, muss die Bedeutung der Bilder in der gesamten Anwendung einheitlich sein.	Unterstützt SQL Anywhere verwendet in allen Anwendungen standardmäßige und einheitliche Bilder.	
(f) Informationen in Textform müssen durch Betriebssystemfunktionen zum Anzeigen von Text bereitgestellt werden. Die Mindestinformationen, die verfügbar gemacht werden müssen, sind Textinhalt, Position des Texteingabecursors und Textattribute.	Unterstützt SQL Anywhere unterstützt Standard-Systemfunktionen für die Textanzeige.	
(g) Anwendungen dürfen vom Benutzer ausgewählte Kontrast- und Farbeinstellungen sowie andere individuelle Anzeigeattribute nicht überschreiben.	Unterstützt SQL Anywhere unterstützt systemweite Farbeinstellungen, die in den Betriebssystemeigenschaften festgelegt wurden.	Benutzerdefinierte Einstellungen, die in den SQL Anywhere-Dienstprogrammen für visuelle Darstellung festgelegt wurden, sind auf die Anwendung begrenzt und wirken sich nicht auf die systemweiten Einstellungen aus.

Kriterien	Unterstützende Funktionen	Bemerkungen und Erläuterungen
(h) Wenn eine Animation angezeigt wird, müssen die Informationen auf Wahl des Benutzers in mindestens einem nicht animierten Präsentationsmodus darstellbar sein.	Unterstützt SQL Anywhere verwendet nur selten Animationen.	Wo Animationen verwendet werden, sind die übermittelten Informationen nicht unbedingt erforderlich oder ergänzend.
(i) Farbkodierung darf nicht als einziges Mittel zur Übermittlung von Informationen, zur Anzeige einer Aktion, als Eingabeaufforderung oder zur Unterscheidung eines visuellen Elements benutzt werden.	Unterstützt SQL Anywhere verwendet Farbkodierung im Zusammenhang mit anderen visuellen Anzeigen zur Übermittlung von Informationen, zur Anzeige einer Aktion, als Eingabeaufforderung oder zur Unterscheidung eines visuellen Elements.	Die Anwendung zur grafischen Darstellung von Datenbankstatistiken von Sybase Central bietet sowohl eine vom Benutzer wählbare Auswahl von Farben als auch die Auswahl der Informationen im Text- und Zahlenformat.
(j) Wenn ein Produkt es dem Benutzer erlaubt, die Einstellungen für Farbe und Kontrast anzupassen, muss eine Vielzahl von Farbauswahlen zur Verfügung gestellt werden, die eine Reihe von Kontraststufen möglich machen.	Unterstützt Wenn Farbe verwendet wird, um Informationen zu übermitteln, bietet SQL Anywhere dem Benutzer eine Auswahl aus einer Vielzahl von Farben.	SQL Anywhere unterstützt Betriebssystemeinstellungen für Farbschemata mit hohem Kontrast, die das Lesen des Bildschirms für manche Benutzer vereinfachen, indem der Bildschirmkontrast durch verschiedene Farbkombinationen erhöht wird.
(k) Software darf keinen Text, keine Objekte und keine anderen Elemente verwenden, die mit einer Frequenz von mehr als 2 Hz und weniger als 77 Hz blinken.	Unterstützt SQL Anywhere verwendet auf keiner Benutzeroberfläche Text oder andere Objekte, die in diesem Frequenzbereich blinken.	

Kriterien	Unterstützende Funktionen	Bemerkungen und Erläuterungen
(l) Wenn elektronische Formulare verwendet werden, muss das Formular Benutzern, die Assistive Technology verwenden, den Zugriff auf Informationen, Feldelemente und Funktionalitäten ermöglichen, die für das Ausfüllen und das Absenden des Formulars erforderlich sind, einschließlich aller Anleitungen und Hinweise.	Unterstützt Elektronische Formulare (Dialoge) in SQL Anywhere unterstützen Assistive Technology.	Für die Java-basierten Komponenten in SQL Anywhere müssen Bildschirmlesegeräte von Drittherstellern die Java Assistive Technology unterstützen.

Abschnitt 1194.31, Funktionale Performance-Kriterien

Kriterien	Unterstützende Funktionen	Bemerkungen und Erläuterungen
(a) Es muss mindestens ein Betriebsmodus und eine Methode zum Informationsabruf zur Verfügung gestellt werden, die ein blinder Benutzer verwenden kann, oder es muss Assistive Technology für blinde oder sehbehinderte Benutzer unterstützt werden.	Unterstützt	SQL Anywhere unterstützt Technologien von Drittherstellern, die die Verwendung von Computerprogrammen für Personen erleichtern, die Assistive Technology einsetzen. SQL Anywhere unterstützt außerdem die Verwendung von Bildschirmlesegeräten für den Zugriff auf Informationen der Benutzeroberfläche. Für Funktionen auf Java-Basis sind Produkte von Drittherstellern erforderlich, die die Java Assistive Technologies unterstützen.

Kriterien	Unterstützende Funktionen	Bemerkungen und Erläuterungen
(b) Es muss mindestens ein Betriebsmodus und eine Methode (Audio oder Großbuchstaben kombiniert oder einzeln) zum Informationsabruf zur Verfügung gestellt werden, für die keine höhere Sehschärfe als 20/70 erforderlich ist, oder es muss Assistive Technology für blinde oder sehbehinderte Benutzer unterstützt werden.	Unterstützt mit Ausnahmen	SQL Anywhere unterstützt die Verwendung von Bildschirmlesegeräten für den Zugriff auf Informationen der Benutzeroberfläche. SQL Anywhere unterstützt mit wenigen Ausnahmen auch große System-schriftarten.
(a) Es muss mindestens ein Betriebsmodus und eine Methode zum Informationsabruf zur Verfügung gestellt werden, die ein gehörloser Benutzer verwenden kann, oder es muss Assistive Technology für gehörlose oder hörbehinderte Benutzer unterstützt werden.	Unterstützt Für keine Funktion der SQL Anywhere-Komponenten ist die Hörfähigkeit des Benutzers erforderlich.	In allen Fällen, in denen SQL Anywhere Informationen akustisch übermittelt, werden diese auch visuell übermittelt.
(d) Wenn akustische Informationen für die Verwendung eines Produkts wichtig sind, muss mindestens ein Betriebsmodus und eine Methode zum Informationsabruf mit besserer Hörbarkeit oder eine Unterstützung für Hörgeräte angeboten werden.	Unterstützt Für keine Funktion der SQL Anywhere-Komponenten ist die Hörfähigkeit des Benutzers erforderlich.	In allen Fällen, in denen SQL Anywhere Informationen akustisch übermittelt, werden diese auch visuell übermittelt.

Kriterien	Unterstützende Funktionen	Bemerkungen und Erläuterungen
(a) Es muss mindestens ein Betriebsmodus und eine Methode zum Informationsabruf zur Verfügung gestellt werden, für die keine Spracheingabe erforderlich ist, oder es muss Assistive Technology für behinderte Personen bereitgestellt werden.	Unterstützt SQL Anywhere-Komponenten erfordern keine Spracherkennung.	
(f) Es muss mindestens ein Betriebsmodus und eine Methode zum Informationsabruf zur Verfügung gestellt werden, für die keine feinmotorische Bedienung oder simultane Aktionen erforderlich sind und die mit eingeschränkter Reichweite und Körperkraft durchgeführt werden können.	Unterstützt SQL Anywhere unterstützt Funktionen des Betriebssystems Microsoft Windows wie MouseKeys, StickyKeys und FilterKeys sowie andere Assistive Technologies.	

Abschnitt 1194.41, Informationen, Dokumentation und Unterstützung

Kriterien	Unterstützende Funktionen	Bemerkungen und Erläuterungen
(a) Unterstützende Produktdokumentation für die Endbenutzer muss auf Wunsch ohne zusätzliche Kosten in anderen Formaten zur Verfügung gestellt werden.	Voll unterstützt	Die Dokumentation wird für Kunden in digitalem Format im Internet bereitgestellt und kann kostenlos von http://sybooks.sybase.com/ abgerufen werden.
(b) Endbenutzer müssen auf Wunsch kostenlosen Zugriff auf eine Beschreibung der Eingabehilfen und Kompatibilitätsfunktionen von Produkten in anderen Formaten erhalten.	Voll unterstützt	SQL Anywhere umfasst Dokumentation über Eingabehilfenfunktionen in Form von Hilfedateien. Die Themen in den Hilfedateien können sowohl online angezeigt als auch einzeln ausgedruckt werden.

Kriterien	Unterstützende Funktionen	Bemerkungen und Erläuterungen
(c) Die Unterstützung für Produkte muss die Kommunikationsanforderungen von Endbenutzern mit Behinderungen erfüllen.	Voll unterstützt Für keine Funktion der SQL Anywhere-Komponenten ist die Hörfähigkeit des Benutzers erforderlich.	Der technische Support bei Sybase bietet eine Reihe von Diensten einschließlich Technische Dokumente, White Papers, Produkthandbücher, Newsgroups, Online-Fallmanagement für Softwareprobleme sowie Bug Fixes und Patches. Weitere Hinweise zu diesen und anderen Support-Diensten finden Sie auf der Website für den technischen Support von Sybase http://www.sybase.com/support .

Eingabehilfen mit Java Access Bridge

SQL Anywhere 16 enthält ein Modul zur Aktivierung von Eingabehilfen. Das Modul Java Access Bridge, das bei jeder Verwendung von Sybase Central oder Interactive SQL geladen wird, unterstützt Assistive Technologies. Software anderer Hersteller, z.B. Bildschirmablesemodule, benutzen diese Komponente, um den Zugriff auf die Software zu ermöglichen.

Weitere Hinweise zur Plattformunterstützung für Eingabehilfen in SQL Anywhere finden Sie unter <http://www.sybase.com/detail?id=1061806>.

Weitere Hinweise zu Eingabehilfen von SQL Anywhere finden Sie unter „Eingabehilfen“ auf Seite 8.

Weitere Hinweise zu Eingabehilfen von Softwareprodukten von Sybase finden Sie unter <http://www.sybase.com/accessibility>.

Datenverwaltungstechnologien

SQL Anywhere stellt zwei relationale Datenbanken zur Verfügung: SQL Anywhere-Server und UltraLite.

SQL Anywhere-Server

SQL Anywhere-Server stellt alle erforderlichen Funktionalitäten für unternehmensweite Abläufe bereit, wie umfassende Transaktionsverarbeitung, referenzielle Integrität, materialisierte Ansichten, Snapshot-Isolation, Hochverfügbarkeit über Datenbankspiegelung und Server-Cluster, gespeicherte Prozeduren in SQL und Java, Trigger, Sperren auf Zeilenebene, automatische Ereignisplanung, automatische Datensicherung und Datenwiederherstellung, Unterstützung für räumliche Daten etc. Der SQL Anywhere-Server kann mehrere hundert gleichzeitige Benutzer und mehrere hundert Gigabyte Daten verwalten. Aufgrund des geringen Speicherbedarfs, des Funktionsreichtums und der automatischen Administrationsfunktionen ist SQL Anywhere ideal für die Einbettung in Server- und Clientanwendungen geeignet, die an verteilten Standorten betrieben werden.

UltraLite

Für kleinere Anwendungen, die auf einer Datenbank basieren, ist UltraLite die ideale Ergänzung. UltraLite ist ein relationales Datenbank-Managementsystem mit allen entsprechenden Funktionen, das mit seinen geringen Anforderungen an Speicherplatz und Systemressourcen speziell für den Einsatz in Handhelds und anderen mobilen Geräten, einschließlich iPhone, Android und BlackBerry, entwickelt wurde. Es bietet vollständige Transaktionsverarbeitung, mehrere Entwicklungsmodelle und einen integrierten Synchronisationsclient für den Austausch mit anderen Datenbanken.

Die Komponenten eines Datenbanksystems

Bei einem relationalen Datenbank-Managementsystem (siehe **Relationales Datenbank-Managementsystem, RDBMS**, im Glossar) handelt es sich um ein System zum Speichern und Abrufen von Daten, in dem die Daten in miteinander verknüpften Tabellen zusammengefasst sind.

Relationale Datenbank-Managementsysteme enthalten folgende Bestandteile:

- eine Datenbank
- einen Datenbankserver
- eine Anwendungsprogrammierschnittstelle (API)
- eine Clientanwendung
- **Datenbank** Eine Datenbank ist eine Sammlung von Tabellen, die durch Primär- und Fremdschlüssel verbunden sind. Die Tabellen enthalten die Informationen in der Datenbank. Die Tabellen und Schlüssel legen zusammen die Struktur der Datenbank fest. Ein Datenbank-Managementsystem greift auf diese Informationen zu.

Bei einer SQL Anywhere-Datenbank handelt es sich um eine Datei, die normalerweise die Erweiterung *.db* hat. Bei einer UltraLite-Datenbank handelt es sich ebenfalls um eine Datei, die normalerweise die Endung *.udb* hat. SQL Anywhere enthält eine Beispieldatenbank, die im SQL Anywhere-Beispielverzeichnis installiert ist: `%SQLANYAMP16%\demo.db`.

- **Datenbankserver** Der Datenbankserver verwaltet die Datenbank. Jeder Zugriff auf die Datenbank erfolgt über den Datenbankserver.

Der Datenbankserver bietet Zugriff von Clientanwendungen auf Datenbanken und verarbeitet Befehle auf sichere und effiziente Weise. Eine Datenbank kann nur jeweils von einem Server verwaltet werden. Ein SQL Anywhere-Datenbankserver kann allerdings viele Datenbanken gleichzeitig verwalten.

Es gibt zwei Versionen des SQL Anywhere-Datenbankservers: der **Personal Server** und der **Netzwerkserver**. Beide Server weisen identische Abfrageprozesse und andere interne Vorgänge auf. Der einzige Unterschied besteht in der Anzahl und im Typ der Verbindungen, die von jedem Server akzeptiert werden. Weitere Hinweise zu den Unterschieden zwischen Personal Servern und Netzwerk-Datenbankservern finden Sie unter „[Unterschiede zwischen Netzwerkserver und Personal Server](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

- **UltraLite-Laufzeitbibliothek** In UltraLite sind die Datenbank-Managementsysteme, die normalerweise in einem Datenbankserver vorhanden sind, als prozessintegrierte Laufzeitbibliothek implementiert. Die Laufzeitbibliothek und die Anwendung sind Teil desselben Prozesses.
- **Programmierschnittstelle** Anwendungen kommunizieren mit dem Datenbankserver über eine Programmierschnittstelle wie ODBC, JDBC, OLE DB, ADO.NET oder Embedded SQL.

Eine vollständige Liste der von SQL Anywhere und UltraLite unterstützten Programmierschnittstellen finden Sie unter „[Programmierschnittstellen](#)“ auf [Seite 31](#).

Jede Programmierschnittstelle beinhaltet eine Bibliothek von Funktionsaufrufen für die Kommunikation mit der Datenbank. Für ODBC und JDBC wird die Bibliothek in der Regel als **Treiber** bezeichnet. Die Bibliothek wird im Allgemeinen als gemeinsam genutzte Bibliothek auf Unix-Systemen oder als dynamische Verknüpfungsbibliothek (DLL) auf Windows-Systemen eingerichtet.

- **Clientanwendung** Clientanwendungen benutzen eine der Programmierschnittstellen, um mit dem Datenbankserver zu kommunizieren.

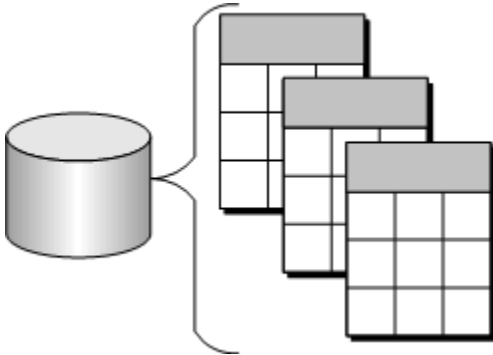
Wenn Sie eine Anwendung mit einem Tool für die vereinfachte Entwicklung (RAD, Rapid Application Development) erstellen, wie beispielsweise mit Sybase PowerBuilder, stellen Sie unter Umständen fest, dass dieses Tool eigene Methoden für die Kommunikation mit Datenbankservern verwendet und die Detailinformationen der Sprachschnittstelle verbirgt. Trotzdem benutzen alle Anwendungen eine der unterstützten Schnittstellen.

Das Konzept der relationalen Datenbank

In den folgenden Abschnitten finden Sie einen kurzen Überblick über grundlegende Konzepte der relationalen Datenbank. Sie enthalten die Definition von Tabellen, Primär- und Fremdschlüsseln sowie Datenbankobjekten.

Datenbanktabellen

In einer relationalen Datenbank werden alle Daten in **Tabellen** gespeichert, die aus **Zeilen** und **Spalten** bestehen.



Jede Tabelle hat eine oder mehrere Spalten, und jeder Spalte ist ein bestimmter Datentyp zugewiesen, wie z.B. "Ganzzahl" (Integer), "Zeichenfolge" für Text (String) oder "Datum" (Date). Jede Zeile der Tabelle enthält genau einen Wert für jede Spalte.

So könnte eine Tabelle mit Mitarbeiterdaten aussehen:

EmployeeID	Surname	GivenName	Phone
102	Huong	Zhang	1096
10693	Donaldson	Anne	7821

Eigenschaften von relationalen Tabellen

Die Tabellen einer relationalen Datenbank haben einige wichtige Eigenschaften:

- Die Reihenfolge der Spalten oder Zeilen spielt keine Rolle.
- Jede Zeile enthält immer nur genau einen Wert pro Spalte, oder sie enthält NULL, was anzeigt, dass es für diese Spalte keinen Wert gibt.
- Alle Werte einer bestimmten Spalte haben immer den gleichen Datentyp.

Die folgende Tabelle enthält einige formale und landläufige Begriffe aus dem Bereich relationaler Datenbanken, mit denen Tabellen und deren Inhalte beschrieben werden. Daneben finden Sie auch eine Gegenüberstellung der entsprechenden Termini nicht-relationaler Datenbanken wie dBase und Microsoft Visual FoxPro. In dieser Dokumentation werden die landläufigen bzw. informellen Begriffe verwendet.

Informeller relationaler Begriff	Formeller relationaler Begriff	Nicht relationaler Begriff
Tabelle	Relation	Datei

Informeller relationaler Begriff	Formeller relationaler Begriff	Nicht relationaler Begriff
Spalte	Attribut	Feld
Zeile	Tupel	Datensatz

Inhalt der Tabellen

Jede Tabelle in der Datenbank sollte Informationen über ein bestimmtes Element enthalten, wie z.B. Mitarbeiter, Produkte oder Kunden.

Wenn Sie eine Datenbank unter Beachtung dieser Vorgaben definieren, können Sie eine Struktur erstellen, die Redundanzen und dadurch verursachte Überschneidungen vermeidet. Beispiel: Die Verkaufsabteilung und die Lieferantenbuchhaltung geben Kundendaten ein und rufen sie ab. In einer relationalen Datenbank werden die Daten zu den Kunden nur einmal gespeichert, nämlich in einer Tabelle, auf die beide Abteilungen Zugriff haben.

Siehe auch

- „Datenbankerstellung“ [[SQL Anywhere Server - Datenbankadministration](#)]

Beziehungen zwischen Tabellen

Man benutzt sogenannte Primär- und Fremdschlüssel, um die Beziehungen zwischen den Daten in den einzelnen Tabellen zu beschreiben. **Primärschlüssel** bezeichnen jede einzelne Zeile einer Tabelle eindeutig; **Fremdschlüssel** geben dagegen die Beziehung zwischen einzelnen Zeilen in unterschiedlichen Tabellen an.

Mithilfe von Primär- und Fremdschlüsseln können Sie in einer relationalen Datenbank Informationen speichern, die effektiv verwaltet werden können, das heißt, dass fast keine Informationen doppelt eingegeben werden müssen.

Siehe auch

- „Fremdschlüssel“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)]
- „Primärschlüssel“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)]

Andere Datenbankobjekte

Eine relationale Datenbank besteht nicht nur aus einer Gruppe von miteinander in Beziehung stehenden Tabellen. In einer relationalen Datenbank finden Sie auch die folgenden Objekte:

- **Indizes** Indizes ermöglichen ein schnelles Auffinden von Informationen. Der Index einer Datenbank ist eigentlich nichts anderes als der Index in einem Buch. In einem Buch verweisen die Indexeinträge auf die Seiten, auf denen die entsprechenden Stichwörter vorkommen. In einer Datenbank stellen die Indexeinträge eine Verbindung der einzelnen indizierten Spaltenwerte zu dem jeweiligen physischen Standort her, an dem sich die Datenzeile befindet, die den indizierten Wert enthält.

Indizes sind ein wichtiges Designelement für eine optimale Performance. Normalerweise müssen Sie Indizes explizit erstellen. Indizes für Primärschlüssel, Fremdschlüssel und Eindeutigkeits-Integritätsregeln werden jedoch automatisch erstellt. Einmal erstellte Indizes funktionieren automatisch und ohne dass der Benutzer dies merkt. Siehe „[Indizes](#)“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*].

- **Textindizes** In Textindizes werden für jede Instanz jedes Begriffs in jeder indizierten Spalte vollständige Positionsinformationen gespeichert. Wenn Sie eine Volltextsuche durchführen, wird ein Textindex verwendet, um übereinstimmende Zeilen zu finden. Aus diesem Grund können Abfragen, die Textindizes verwenden, schneller sein als solche, die alle Werte in der Tabelle durchsuchen müssen. Siehe „[Konzepte und Referenz zu Textindizes](#)“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*] und „[Volltextsuche](#)“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*].
- **Login-Richtlinien** Login-Richtlinien bestehen aus einer Reihe von Regeln, die angewendet werden, wenn Sie eine Datenbankverbindung für einen Benutzer erstellen. Siehe „[Login-Richtlinien](#)“ [*SQL Anywhere Server - Datenbankadministration*].
- **Views** Ansichten sind temporäre Tabellen. Sie sind in den Clientanwendungen wie Tabellen angelegt, enthalten aber selbst keine Daten. Sie füllen sich vielmehr mit Daten aus den Tabellen, die ihnen zugrunde liegen, wenn ein Datenabruf erfolgt.

Die Tabellen, in denen die Daten wirklich gespeichert sind, werden manchmal auch **Basistabellen** genannt, um sie von den Ansichten zu unterscheiden. Dabei wird eine Ansicht durch eine SQL-Abfrage einer Basistabelle oder anderer Ansichten definiert.

Siehe „[Ansichten](#)“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*].

- **Materialisierte Ansichten** SQL Anywhere unterstützt auch materialisierte Ansichten. Eine materialisierte Ansicht ist eine Ansicht, deren Ergebnismenge berechnet und auf der Festplatte gespeichert wurde, ähnlich wie bei einer Basistabelle. Vom Konzept her ist eine materialisierte Ansicht sowohl eine Ansicht (sie verfügt über eine Abfragespezifikation) als auch eine Tabelle (sie verfügt über beständige materialisierte Zeilen). Sie können daher viele Vorgänge, die für Tabellen gedacht sind, auch für materialisierte Ansichten durchführen. Z.B. können Sie Indizes für materialisierte Ansichten erstellen und wieder von diesen entfernen.

Materialisierte Ansichten sind ideal für Umgebungen geeignet, in denen die Datenbank groß ist, häufige Abfragen in wiederholten Aggregat- und Join-Vorgängen zu großen Datenmengen führen und der Zugriff auf die aktuellsten Daten keine entscheidende Anforderung ist. Siehe „[Materialisierte Ansichten](#)“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*].

- **Gespeicherte Prozeduren und Trigger** Dabei handelt es sich um Routinen, die in der Datenbank gespeichert sind und die Daten in dieser Datenbank ändern.

Sie können eigene gespeicherte Prozeduren erstellen und benennen, um bestimmte Datenbankabfragen und andere Aufgaben der Datenbankverarbeitung auszuführen. Gespeicherte Prozeduren können Parameter akzeptieren und geben Ergebnismengen zurück. Beispiel: Sie können eine gespeicherte Prozedur erstellen, die die Namen aller Kunden zurückgibt, deren Ausgaben höher sind als der Parameter, der für den Aufruf der gespeicherten Prozedur festgelegt wurde.

Ein Trigger ist eine besondere gespeicherte Prozedur, die je nach der Triggerdefinition automatisch auslöst, wenn ein Benutzer Daten aktualisiert, löscht oder einfügt. Ein Trigger wird mit einer Tabelle

oder mit Spalten in einer Tabelle verknüpft. Trigger sind nützlich, wenn bestimmte Geschäftsregeln in einer Datenbank aufrechterhalten werden sollen.

Sie können in einer Datenbank auch Java-Klassen installieren. Mit Java-Klassen können Sie wirksam Logikelemente in Ihre Datenbank integrieren. Siehe „[Praktische Einführung: Java in der Datenbank verwenden](#)“ [*SQL Anywhere Server - Programmierung*].

Siehe „[Gespeicherte Prozeduren, Trigger, Batches und benutzerdefinierte Funktionen](#)“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*].

- **Benutzer und Gruppen** Jeder Benutzer einer Datenbank hat eine Benutzer-ID und ein Kennwort. Sie können Privilegien für jeden Benutzer festlegen, sodass vertrauliche Informationen geschützt werden und die Benutzer keine Änderungen vornehmen können, für die sie nicht autorisiert sind. Benutzer können Gruppen zugeordnet werden, um die Verwaltung der Privilegien zu erleichtern.

Siehe „[Benutzersicherheit \(Rollen und Privilegien\)](#)“ [*SQL Anywhere Server - Datenbankadministration*].

Zusätzlich zu diesen gemeinsamen Datenbankobjekten enthält SQL Anywhere auch erweiterte Funktionen:

- Ereignisse
- Domänen
- Publikationen
- Webdienste
- Ferndatenzugriff
- Wartungspläne
- Räumliche Bezugssysteme

Hintergrundinformationen zu SQL Anywhere

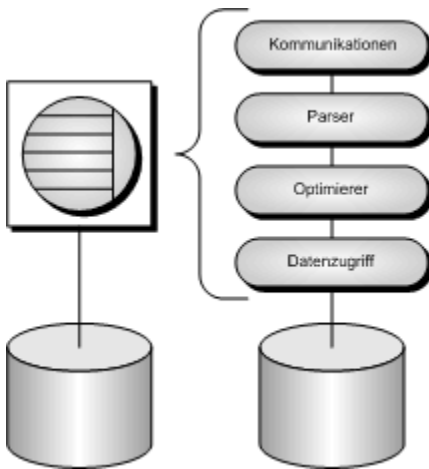
Es besteht zwar eigentlich keine dringende Notwendigkeit, die internen Vorgänge des Datenbankservers zu durchschauen, aber ein Blick hinter die Kulissen führt durchaus zu einem besseren Verständnis der Vorgänge zwischen Datenbankserver und Datenbank.

Interne Struktur des SQL Anywhere-Datenbankservers

Der SQL Anywhere-Datenbankserver verfügt über eine interne Struktur, durch die viele Anfragen effizient verwaltet werden können.

- Über eine spezielle Kommunikationsschicht wird der Austausch der Daten mit den Clientanwendungen abgewickelt. Diese Schicht erhält die Anfragen von den einzelnen Clientanwendungen und gibt die Ergebnisse an diese zurück. Das Timing der einzelnen Aktionen wird über eine Verhandlung zwischen Client und Server durchgeführt. Auf diese Weise wird sichergestellt, dass die Netzwerkbelastung auf ein Minimum begrenzt bleibt, die angeforderten Daten aber so schnell wie möglich beim Client zur Verfügung stehen.

- Der Parser überprüft jede an den Datenbankserver gesendete SQL-Anweisung und konvertiert sie in eine intern lesbare Form zur weiteren Verarbeitung.
- Je nachdem, ob es sich bei der Anforderung um eine Abfrage, eine Aktualisierung oder eine Löschanweisung handelt, gestaltet sich der Zugriff auf die Daten sehr unterschiedlich und benötigt auch unterschiedlich viel Zeit. Der Optimierer wählt die beste Methode aus, um die erforderlichen Daten möglichst schnell zu erhalten.
- Die unterste Schicht im Datenbankserver hat die Aufgabe, die Daten von der Festplatte zu lesen, sie wieder auf die Festplatte zu schreiben, ggf. Daten im Cache zwischenspeichern, um unnötige Festplattenzugriffe zu vermeiden und die Anforderungen der einzelnen Benutzer gleichmäßig aufzuteilen.



SQL Anywhere-Datenbankserver

Es gibt zwei Versionen des SQL Anywhere-Datenbankservers: **Personal Server** (dbeng16) und **Netzwerkserver** (dbsrv16). Eine mit einem Personal Datenbankserver erstellte Datenbank kann mit einem Netzwerk-Datenbankserver benutzt werden und umgekehrt. Die Engine für die Verarbeitung von Anforderungen ist beim Personal Server und Netzwerkserver identisch und beide unterstützen genau die gleiche SQL-Sprache und viele identische Datenbankfunktionen. Allerdings bestehen auch einige Unterschiede zwischen den beiden Servern.

- **Der Personal Datenbankserver** Diese Programmdatei wird für Einzelbenutzer auf demselben Computer bereitgestellt, z. B. als eingebetteter Datenbankserver. Sie unterstützt keine Client/Server-Kommunikation über ein Netzwerk.

Unter Windows-Betriebssystemen mit Ausnahme von Windows Mobile lautet der Name der Programmdatei des Personal Servers *dbeng16.exe*. Unter Unix-Betriebssystemen lautet der Name *dbeng16*. Der Personal Server wird unter Windows Mobile nicht unterstützt.

- **Der Netzwerk-Datenbankserver** Dieses Programm unterstützt Client/Serverkommunikation über ein Netzwerk und ist für den Mehrbenutzerbetrieb vorgesehen.

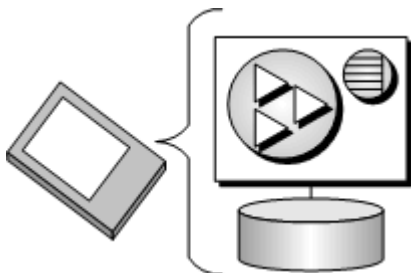
Unter Windows-Betriebssystemen einschließlich Windows Mobile lautet der Name der Programmdatei des Netzwerkservers *dsrv16.exe*. Unter Linux- und Unix-Betriebssystemen lautet der Name *dsrv16*.

Siehe auch

- „Unterschiede zwischen Netzwerkservers und Personal Server“ [[SQL Anywhere Server - Datenbankadministration](#)]

Interne Struktur von UltraLite

Um eine Datenbankanwendung auf einem kleinen System wie einem Handheld PC oder Organizer zu implementieren, kann es sinnvoll sein, UltraLite zu verwenden. In UltraLite werden die vom Server normalerweise durchgeführten Funktionen in eine Laufzeitbibliothek integriert. Die Laufzeitbibliothek ist mit der Anwendung kombiniert und wird Teil desselben Prozesses. Daher besteht eine Eins-zu-Eins-Relation zwischen der Datenbank und der Anwendung.



Für Deployments, bei denen sich mehrere Anwendungen gleichzeitig mit einer Datenbank auf demselben Gerät verbinden müssen, muss die Bibliothek als eigener Prozess existieren. In diesen Fällen wird die UltraLite-Datenbank-Engine benutzt.

Andere Funktionen

- UltraLite verfügt über eine integrierte MobiLink-Synchronisationstechnologie, die die Verbindung Ihrer Anwendung zum Informationsnetzwerk aufrechterhält.

Hinweise zur Integration von UltraLite und MobiLink finden Sie unter „[UltraLite-Clients](#)“ [[UltraLite - Datenbankverwaltung](#)].

- UltraLite unterstützt zahlreiche Betriebssysteme. Siehe „[UltraLite-Überblick](#)“ [[UltraLite - Datenbankverwaltung](#)].

Datenbankdateien

In den folgenden Abschnitten finden Sie einen Überblick über die Dateitypen wie Datenbank-, Transaktions- und temporäre Dateien, aus denen eine Datenbank besteht. Die Unterschiede in der Implementierung dieser Dateien in SQL Anywhere und UltraLite werden ebenfalls erörtert.

SQL Anywhere-Datenbankdateien

Sämtliche Informationen einer SQL Anywhere-Datenbank werden normalerweise in einer einzigen Datenbankdatei gespeichert, die nötigenfalls auf andere Positionen und Computer kopiert werden kann. Es ist zwar durchaus möglich, eine Datenbank auch auf mehrere Dateien zu verteilen, im Allgemeinen ist dies aber nur bei sehr großen Datenbanken erforderlich.

Zusätzlich zu der Datenbankdatei verwendet SQL Anywhere zwei andere Dateien, um eine Datenbank zu betreiben: das Transaktionslog und die temporäre Datei.

- **Die Datenbankdatei** Intern besteht die Datenbankdatei aus Seiten. Diese Seiten sind Abschnitte der Festplatte mit fester Größe. Die Datenzugriffsschicht liest und schreibt die Daten immer seitenweise. Viele dieser Seiten enthalten die Daten der einzelnen Datenbanktabellen, auf anderen Seiten sind unter anderem aber auch Indexinformationen sowie Informationen zur Aufteilung der Daten innerhalb der Datenbank gespeichert.
- **Das Transaktionslog** Beim Transaktionslog handelt es sich um eine separat gespeicherte Datei, die eine Aufzeichnung sämtlicher in der Datenbank durchgeführten Vorgänge enthält. Normalerweise hat das Transaktionslog denselben Namen wie die Datenbankdatei, endet aber mit dem Suffix *.log* anstelle von *.db*. Das Log hat drei wichtige Funktionen:
 - **Alle Vorgänge in der Datenbank aufzeichnen, um eine Wiederherstellung zu gewährleisten** Sollte eine Datenbankdatei einmal beschädigt werden, so können Sie die Datenbank anhand einer Sicherung und des Transaktionslogs wieder herstellen.
 - **Performance steigern** Da die Daten in das Transaktionslog geschrieben werden, kann der Datenbankserver die Anweisungen sofort verarbeiten, ohne ständig in die Datenbankdatei schreiben zu müssen.
 - **Datenbankreplikation ermöglichen** SQL Remote und die MobiLink-Synchronisation benutzen das Transaktionslog, um Änderungen in den Datenbanken zu synchronisieren.
- **Die temporäre Datei** Die temporäre Datei wird erstellt, sobald der Datenbankserver gestartet wird, und entfernt, wenn der Server herunterfährt. Wie schon der Name sagt, wird die temporäre Datei verwendet, um temporäre Daten zu speichern, während der Datenbankserver läuft. In der temporären Datei werden keine Daten gespeichert, die sitzungsübergreifend erhalten bleiben müssen.

Die temporäre UltraLite-Datei wird in demselben Verzeichnis gespeichert wie die Datenbankdatei.

Siehe „Umgebungsvariablen TMP, TMPDIR und TEMP“ [[SQL Anywhere Server - Datenbankadministration](#)].

UltraLite-Datenbankdateien

UltraLite-Datenbanken verfügen über dieselben Funktionen wie oben beschrieben, wobei folgende Ausnahmen gelten:

- UltraLite-Datenbankdateien enthalten keine Informationen über die Verteilung von Daten innerhalb der Datenbank.
- UltraLite protokolliert ihre Transaktionen intern und nicht in einer eigenen Logdatei.

- Die temporäre UltraLite-Datei wird in demselben Verzeichnis gespeichert wie die Datenbankdatei.

Siehe „[Transaktions- und Zeilenstatusverwaltung](#)“ [[UltraLite - Datenbankverwaltung](#)].

SQL Anywhere- und UltraLite-Zielplattformen

SQL Anywhere und UltraLite sind für die Datenspeicherung und den Datenzugriff von Unternehmensdatenquellen bis zu kleinen, mobilen Datenbanken konzipiert. Wenn Sie eine Anwendung entwickeln, müssen Sie die Datenbank mit den richtigen Funktionen wählen.

- Wenn Ihre Zielplattform Unix oder Mac OS X ist, müssen Sie eine SQL Anywhere-Datenbank verwenden.
- Wenn Ihre Zielplattform Embedded Linux, Android, BlackBerry bzw. iPhone oder iPad ist, müssen Sie eine UltraLite-Datenbank verwenden.
- Wenn die Zielplattform Windows 7, Windows Vista, Windows XP oder Linux ist, können Sie entweder SQL Anywhere oder UltraLite verwenden. SQL Anywhere wird oft bevorzugt, weil mehr Funktionen zur Verfügung stehen und zusätzlicher Speicherbedarf meist kein Problem darstellt.
- Wenn die Zielplattform Windows Mobile ist, etwa in einem Pocket PC oder einem Smartphone, müssen Sie auf die Speicherbeschränkungen achten und überlegen, welche Aufgaben Ihre Anwendung ausführen soll. Unter Windows Mobile benötigt SQL Anywhere etwa 6 MB Speicher plus weitere 2 MB für die Synchronisationskomponenten, während UltraLite weniger als 1 MB benötigt und die Synchronisationsfunktionen integriert sind. UltraLite ist also deutlich kleiner, bietet aber nicht dieselbe Unterstützung wie SQL Anywhere für Aufgaben wie komplexe Abfragen, Ereignisse, Prozeduren, Trigger, Ansichten, usw.

Weitere Hinweise über die Unterschiede zwischen der Kerndatenbanklösung (SQL Anywhere) und der UltraLite-Datenbanklösung finden Sie unter „[Funktionsvergleiche von UltraLite, UltraLite Java Edition und SQL Anywhere](#)“ [[UltraLite - Datenbankverwaltung](#)].

Datenbankszenarios

Datenbankanwendungen können eine Verbindung zu einem Datenbankserver herstellen, der sich auf demselben Computer befindet wie die Anwendung. Netzwerk-Datenbankserver können sich mit einem anderen Computer verbinden. Mit SQL Anywhere können Sie verteilte Datenbanken für den Fernzugriff und mobile Anwendungen einrichten, wobei physisch getrennte Datenbanken auf verschiedenen Computern gemeinsam Daten verarbeiten.

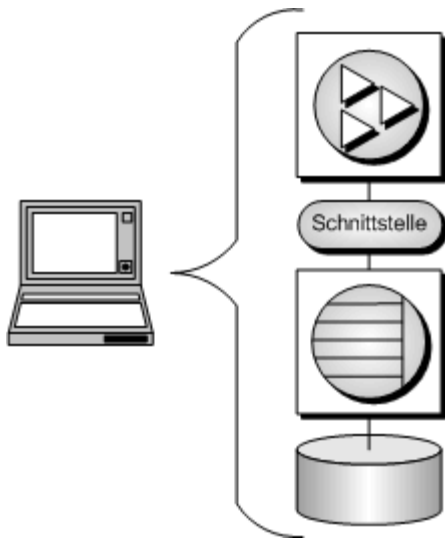
PC-Anwendungen und eingebettete Datenbanken

Mithilfe von SQL Anywhere können Sie eine komplette Anwendung und die dazu gehörende Datenbank auf einem einzelnen Computer erstellen. Im einfachsten Fall handelt es sich dabei um eine **selbstständige Anwendung** oder **persönliche Anwendung**. Diese Anwendung beinhaltet alle Daten und Objekte, und es besteht keine Verbindung zu anderen Datenbanken. Hierbei werden der Datenbankserver und die

Datenbank von der Clientanwendung gestartet. Dabei stellt die Datenbank häufig eine **eingebettete Datenbank** dar. Für den Endbenutzer ist die Datenbank ein Teil der Anwendung.

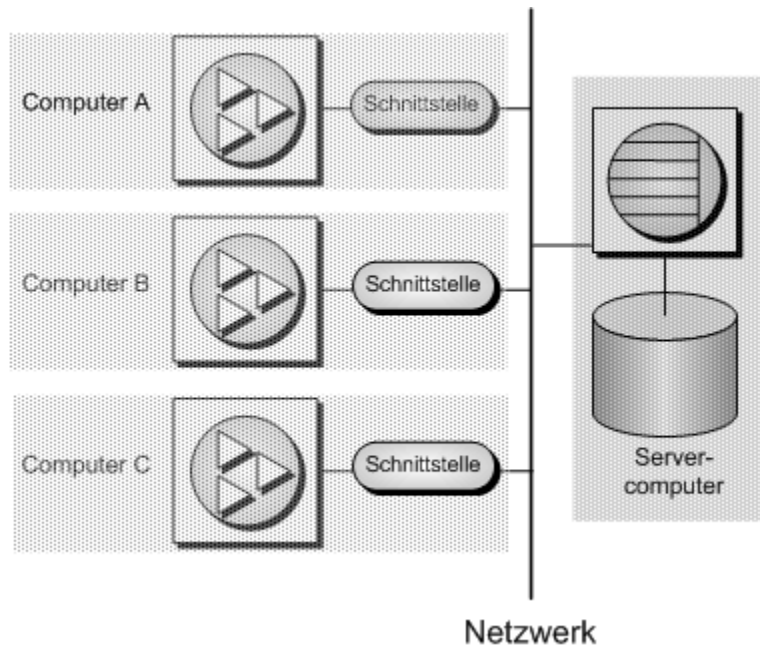
Viele relationale Datenbank-Managementsysteme müssen von erfahrenen Administratoren verwaltet werden. Ein Merkmal von SQL Anywhere-Datenbanken ist, dass sie ganz ohne Administration betrieben werden können.

Der Personal Datenbankserver von SQL Anywhere wird im Allgemeinen für eingebettete Anwendungen eingesetzt. Eingebetteten Anwendungen liegt die folgende Architektur zugrunde. Dabei wird mithilfe einer Programmierschnittstelle eine Verbindung von einer Clientanwendung zu einem Datenbankserver hergestellt, der auf demselben Computer ausgeführt wird:



Client/Serveranwendungen

Mithilfe von SQL Anywhere können Sie ein System aufbauen, bei dem viele auf unterschiedlichen Computern ausgeführte Anwendungen über ein Netzwerk auf einen einzelnen Netzwerk-Datenbankserver zugreifen, der auf einem zentralen Computer läuft. Hierbei handelt es sich um eine **Client/Server-** bzw. **Mehrbenutzer-Datenbank-Umgebung**, der die folgende Architektur zugrunde liegt. Die Interface-Bibliothek ist dabei auf jedem Clientcomputer gespeichert.



In diesem Fall ist der Datenbankserver der SQL Anywhere-Netzwerkserver, der die Netzwerkverbindungen über TCP/IP unterstützt.

Damit eine Clientanwendung in einer Client/Serverumgebung arbeiten kann, müssen Sie zusätzliche Verbindungsparameter angeben, normalerweise den Verbindungsparameter HOST und optional den Parameter ServerName.

Siehe auch

- „Deployment-Typen“ [[SQL Anywhere Server - Programmierung](#)]
- „Datenbankverbindungen“ [[SQL Anywhere Server - Datenbankadministration](#)]

Vielstufige Datenverarbeitungsarchitektur

Bei der vielstufigen Datenverarbeitung wird die Anwendungslogik auf einem Anwendungsserver wie einem Sybase EAServer, WebLogic oder WebSphere gespeichert, der sich zwischen dem Datenbankserver und der Clientanwendung befindet. Häufig greift ein einzelner Anwendungsserver auf mehrere Datenbanken und auf weitere nicht relationale Datenspeicher zu. Bei Internetanwendungen ist die Clientseite browserbasiert, und der Anwendungsserver ist im Allgemeinen eine Webserver-Erweiterung. Viele moderne, vielstufige Anwendungen benutzen eine dienstorientierte Architektur (SOA, service-oriented architecture) auf der Basis von Webdiensten.

Sybase EAServer speichert die Anwendungslogik in Form von Komponenten und stellt diese den Clientanwendungen zur Verfügung. Bei den Komponenten kann es sich um Sybase PowerBuilder-Komponenten, Java-Beans oder COM-Komponenten handeln.

Anwendungsserver können ihren Clientanwendungen auch die Transaktionslogik zur Verfügung stellen. Dabei wird sichergestellt, dass die Vorgänge gruppenweise automatisch in mehreren Datenbanken ausgeführt werden. SQL Anywhere ist für eine vielstufige Verarbeitung gut geeignet. Außerdem sind verteilte Transaktionen möglich, die mithilfe von Microsoft Distributed Transaction Coordinator koordiniert werden. Sowohl Sybase Enterprise Application Server als auch Microsoft Transaction Server verwenden DTC, um den Clientanwendungen Transaktionsdienste zur Verfügung zu stellen.

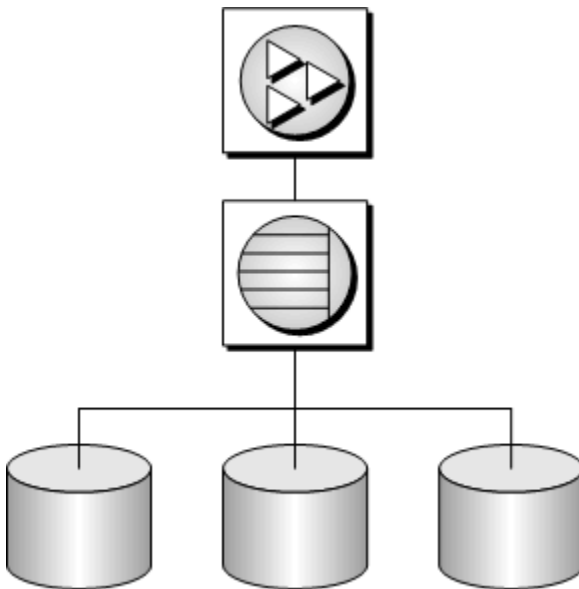
Aufgrund der integrierten Unterstützung von Webdiensten ist SQL Anywhere für viele vielstufige oder SOA-Anwendungen gut geeignet.

Siehe auch

- „Dreischichtige Datenverarbeitung und verteilte Transaktionen“ [[SQL Anywhere Server - Programmierung](#)]

Mehrere Datenbanken auf einem einzelnen Datenbankserver ausführen

SQL Anywhere Personal- und Netzwerkservers können viele Datenbanken gleichzeitig verwalten. Zwar kann eine Verbindung von einer Anwendung immer nur zu einer Datenbank hergestellt werden, es ist aber durchaus möglich, von einer Anwendung aus mehrere Verbindungen zu unterschiedlichen Datenbanken aufzubauen bzw. mit mehreren Anwendungen über ein und denselben Datenbankserver auf mehrere Datenbanken zuzugreifen.



Datenbanken können zusammen mit dem Datenbankserver, durch Herstellen einer Verbindung zu einer Datenbank mit dem Verbindungsparameter DatabaseFile oder der Anweisung START DATABASE gestartet werden.

Siehe auch

- „Syntax des SQL Anywhere-Datenbankservers“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Verbindungsparameter DatabaseFile (DBF)“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „START DATABASE-Anweisung“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]

Datenzugriff auf andere Datenbanken

Mithilfe der SQL Anywhere-Funktionen zum Ferndatenzugriff können Sie auf die Datenbanken mehrerer Datenbankserver bzw. desselben Datenbankservers zugreifen. Damit ist die Anwendung immer noch mit nur einer Datenbank verbunden. Durch das Festlegen von Fremdservern können Sie allerdings auf die in der Fremddatenbank gespeicherten Proxytabellen genauso zugreifen, als gehörten sie zu der Datenbank, mit der Sie gerade verbunden sind.

Siehe auch

- „Ferndatenzugriff“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)]

ETL-Funktionen

Extrahieren, Transformieren (Umwandeln) und Laden (ETL) ist der Prozess, bei dem große Mengen von Daten aus unterschiedlichen Datenquellen extrahiert und in einer einzigen Datenbank konsolidiert werden. In der Extraktionsphase werden die Daten syntaktisch analysiert und entsprechend ausgewertet. Während der Transformation bzw. Umwandlung werden die Daten verändert, um das für die Speicherung erforderliche Format zu erhalten. Einige übliche Umwandlungen sind die Eliminierung von unnötigen Tabellen, die Umwandlung von berechneten Werten und die Konvertierung von Werten, wie z.B. Datumsangaben, in ein gemeinsames Format, damit die Daten konsolidiert werden können. Die Daten werden anschließend in der Frequenz und dem Umfang in die Datenbank geladen, die den Anforderungen des Unternehmens entsprechen.

SQL Anywhere stellt mehrere Funktionen zur Unterstützung von ETL bereit. Beispiel:

- **OPENSTRING-Vorgang** Verwenden Sie den OPENSTRING-Vorgang in der FROM-Klausel, um Daten aus client- und serverseitigen Datenquellen umzuwandeln und zu laden. Siehe „FROM-Klausel“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)].
- **OPENXML-Operator** Verwenden Sie den Zeichenfolgenoperator OPENXML, um Daten aus XML-Dokumenten zu extrahieren. Siehe „OPENXML-Operator“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)].
- **MERGE-Anweisung** Verwenden Sie die MERGE-Anweisung, um Daten aus unterschiedlichen Quellenobjekten zusammenzuführen. Siehe „MERGE-Anweisung“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)].
- **Proxytabellen** Verwenden Sie Proxytabellen, um auf Objekte wie Tabellen, Ansichten und materialisierte Ansichten in einer entfernten Datenbank zuzugreifen. Siehe „Proxy-Tabellen“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)].

- **Systemprozeduraufrufe in der FROM-Klausel** Sie können verschiedene Systemprozeduren in der FROM-Klausel einer Abfrage verwenden, um Daten für das Laden zu extrahieren und umzuwandeln. Eine Liste der in SQL Anywhere bereitgestellten Systemprozeduren finden Sie unter „Systemprozeduren“ [*SQL Anywhere Server - SQL-Referenzhandbuch*].

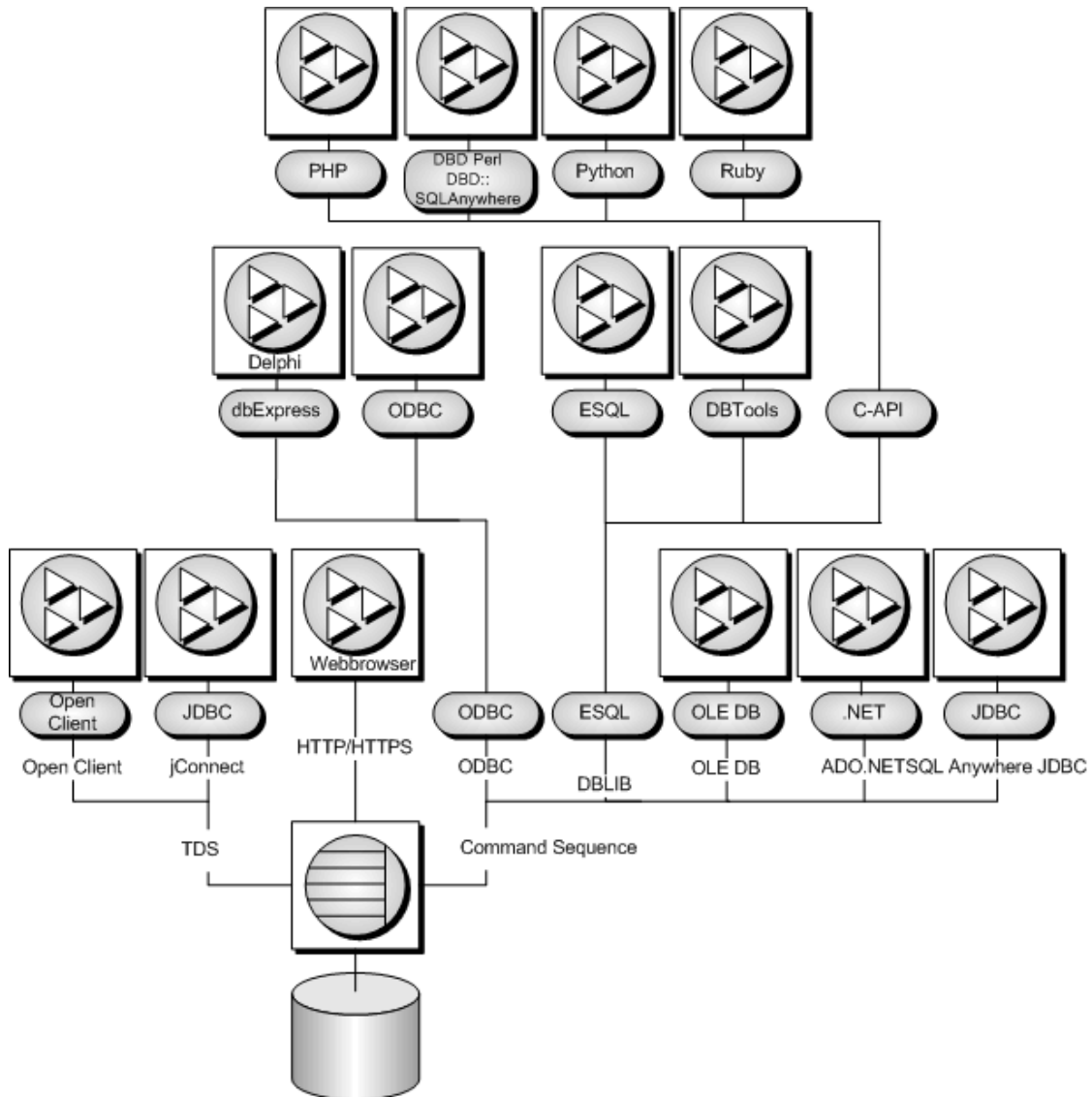
Programmierschnittstellen

SQL Anywhere unterstützt viele Programmierschnittstellen für den Datenzugriff und bietet damit größtmögliche Flexibilität für die von Ihnen genutzten Anwendungsarten und Umgebungen zur Anwendungsentwicklung.

Einen Überblick über die Architektur von Datenbank Anwendungen finden Sie unter „Datenbankszenarios“ auf Seite 26.

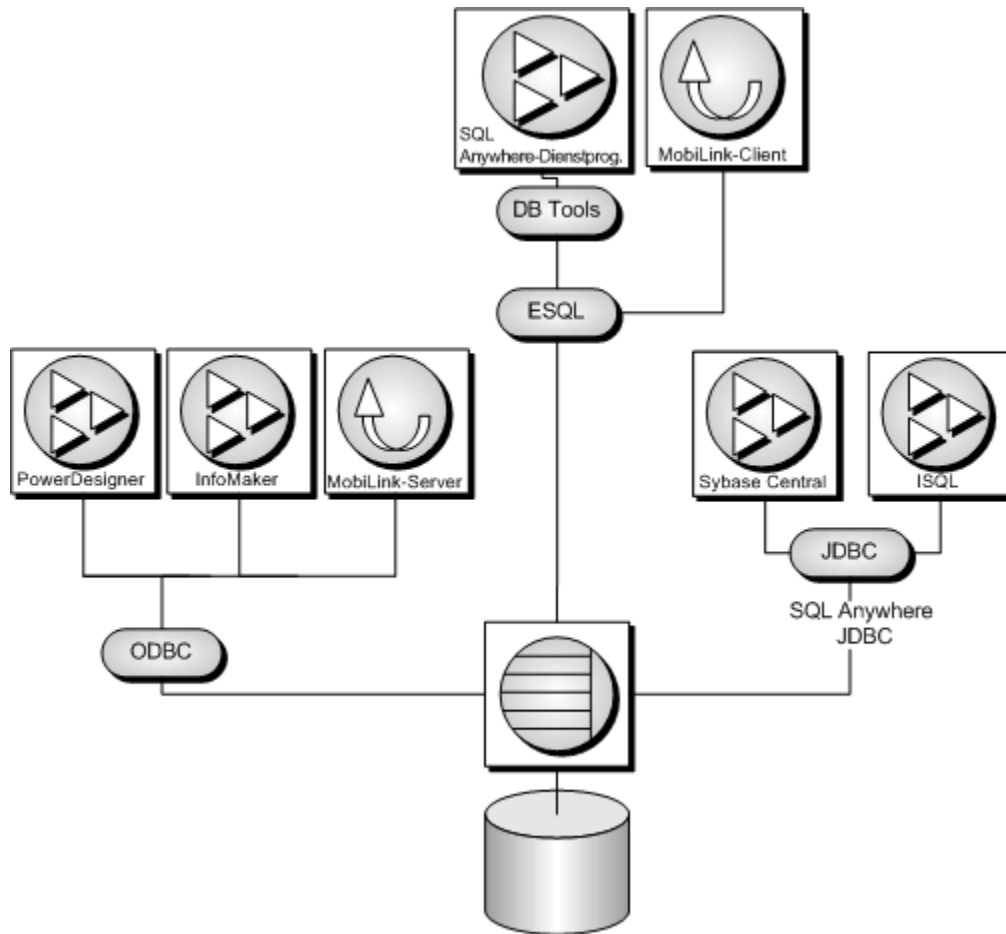
Unterstützte SQL Anywhere-Programmierschnittstellen und Protokolle

Die folgende Abbildung zeigt die unterstützten Schnittstellen sowie die verwendeten Interface-Bibliotheken. Die Namen von Schnittstellenbibliothek und Schnittstelle stimmen in der Regel überein.



SQL Anywhere-Anwendungen

Die in SQL Anywhere enthaltenen Anwendungen verwenden folgende Schnittstellen:



SQL Anywhere-Programmierschnittstellen

Einzelheiten über die SQL Anywhere-Programmierschnittstellen finden Sie in der nachstehenden Liste:

- „SQL Anywhere .NET-Unterstützung“ [[SQL Anywhere Server - Programmierung](#)]
- „ODBC-Unterstützung“ [[SQL Anywhere Server - Programmierung](#)]
- „OLE DB- und ADO-Entwicklung“ [[SQL Anywhere Server - Programmierung](#)]
- „Embedded SQL“ [[SQL Anywhere Server - Programmierung](#)]
- „JDBC-Unterstützung“ [[SQL Anywhere Server - Programmierung](#)]
- „Sybase Open Client-Unterstützung“ [[SQL Anywhere Server - Programmierung](#)]
- „SQL Anywhere-C-API-Unterstützung“ [[SQL Anywhere Server - Programmierung](#)]
- „Perl/DBI-Unterstützung“ [[SQL Anywhere Server - Programmierung](#)]
- „SQL Anywhere-PHP-Unterstützung“ [[SQL Anywhere Server - Programmierung](#)]
- „Python-Unterstützung“ [[SQL Anywhere Server - Programmierung](#)]
- „SQL Anywhere-Ruby-API-Unterstützung“ [[SQL Anywhere Server - Programmierung](#)]

UltraLite-Programmierschnittstellen

UltraLite bietet ebenfalls eine Auswahl an Programmierschnittstellen für den direkten Zugriff auf Daten.

Weitere Hinweise zu UltraLite-Programmierschnittstellen finden Sie unter „[Vorteile der UltraLite APIs für Windows Mobile](#)“ [*UltraLite - Datenbankverwaltung*].

Kommunikationsprotokolle der Interface-Bibliothek

Jede Interface-Bibliothek kommuniziert mit dem Datenbankserver über ein **Kommunikationsprotokoll**. SQL Anywhere unterstützt zwei Kommunikationsprotokolle, nämlich **Command Sequence** und **Tabular Data Stream (TDS)**. Bei beiden handelt es sich um interne Protokolle, und in den meisten Fällen ist es unerheblich, welches Sie gerade verwenden. Die Wahl der Entwicklungsumgebung wird in erster Linie von den zur Verfügung stehenden Programmen beeinflusst.

Die wesentlichen Unterschiede zeigen sich bei der Verbindungsaufnahme mit der Datenbank. Command Sequence-Anwendungen und TDS-Anwendungen verwenden unterschiedliche Verfahren zum Identifizieren einer Datenbank und eines Datenbankservers. Aus diesem Grund gestalten sich die Verbindungsfenster unterschiedlich.

- **Command Sequence** Dieses Protokoll wird von SQL Anywhere, dem SQL Anywhere-JDBC-Treiber und den APIs Embedded SQL, ODBC, OLE DB und ADO.NET verwendet.
- **TDS** Dieses Protokoll wird verwendet von Sybase Adaptive Server Enterprise, vom jConnect-JDBC-Treiber und von Sybase Open Client-Anwendungen. Weitere Hinweise zu TDS finden Sie unter „[SQL Anywhere als Open Server](#)“ [*SQL Anywhere Server - Datenbankadministration*].

Überblick über Datenaustauschtechnologien

SQL Anywhere bietet eine Reihe von Optionen für den Austausch von Daten mit bestehenden Unternehmenssystemen und mobilen Geräten durch den Einsatz von folgenden Tools:

- **MobiLink-Synchronisation** MobiLink bietet sitzungsbasierte, bidirektionale Synchronisation. Diese Art der Synchronisation ist die ideale Lösung für den Datenaustausch zwischen einer zentralen Datenbank und mehreren UltraLite- oder SQL Anywhere-Datenbanken, sowie zwischen einer zentralen, nicht-relationalen Datenquelle und vielen dezentralen UltraLite- oder SQL Anywhere-Datenbanken.

Während einer MobiLink-Synchronisation führt die entfernte Datenbank Uploads aller Änderungen durch, die seit der vorherigen Synchronisation mit dem MobiLink-Server in ihr vorgenommen wurden. Nach Erhalt dieser Daten aktualisiert der MobiLink-Server die zentrale Datenbank und überträgt anschließend alle Änderungen der zentralen Datenbank in die entfernte Datenbank. Damit gewährleistet dieses Synchronisationssystem die transaktionale Integrität der Datenbanken, wenn eine Verbindung zwischen Datenbanken abbricht, und bietet gleichzeitig einen Mechanismus für die Lösung von Konflikten bei der Änderung identischer Datensätze.

Mit den Dateiübertragungsfunktionen von MobiLink können Sie Dateien an entfernte Anwendungen über dieselbe Verbindung wie die für die Synchronisation von Daten verwendete übertragen. Dies ist vor allem beim Einlesen von Daten in neue mobile Datenbanken oder beim Upgrade der Software sinnvoll.

MobiLink bietet darüber hinaus direkte Zeilenbehandlung bei der Synchronisation entfernter Daten mit einer beliebigen zentralen Datenquelle. Zu den Datenquellen, mit denen Sie synchronisieren können, gehören u. a. Anwendung, Webserver, Webdienst, Anwendungsserver, Textdatei, Tabelle, nicht-relationale Datenbank sowie ein RDBMS (relationales Datenbank-Managementsystem), das als konsolidierte Datenbank nicht unterstützt wird.

- **SQL Remote - Replikation** SQL Remote ist eine Datenreplikationstechnologie für die Zwei-Wege-Synchronisation zwischen einer konsolidierten Datenbank und einer großen Anzahl von entfernten Datenbanken, bei denen in der Regel viele mobile Datenbanken einbezogen sind.

SQL Remote benutzt eine Store-and-Forward-Architektur zur Synchronisation von Daten mit einer Datei oder einem Nachrichtenübertragungsmechanismus wie FTP oder E-Mail.

SQL Remote gewährleistet die Integrität von Transaktionen und ist daher eine ideale Lösung für viele Geschäftsanwendungen, vor allem wenn die Verbindung unter den vernetzten Systemen nicht fortlaufend gesichert ist. Darüber hinaus wurden die Anforderungen an Arbeits- und Festplattenspeicher für alle Komponenten des Replikationssystems bewusst niedrig gehalten.

Siehe auch

- „SQL Remote-Systeme“ [[SQL Remote](#)]

Synchronisationstechnologie im Vergleich

Zu den Datenaustauschtechnologien zählen Synchronisation, Replikation, Messaging und mobile Webdienste.

Die **Synchronisation** von Daten hat die gemeinsame Nutzung von Daten zwischen physisch getrennten Datenbanken zum Ziel. Wenn eine Anwendung gemeinsam genutzte Daten in einer Datenbank ändert, werden die Änderungen durch Synchronisation in alle anderen Datenbanken übertragen. Änderungen können auf verschiedene Arten und durch unterschiedliche Kanäle übertragen werden, wobei trotz flexibler Anwendungsarchitektur die Sicherung der Datenintegrität gewährleistet wird.

SQL Anywhere bietet zwei Synchronisationstechnologien:

- **MobiLink** ist eine sitzungsbasierte Technologie für die Ein- oder Zwei-Wege-Synchronisation von Daten zwischen einer zentralen, konsolidierten Datenbank und einer großen Anzahl von entfernten Datenbanken. Sie unterstützt eine Vielzahl konsolidierter Datenbankserver und stellt eine API für die Synchronisation mit so gut wie allen anderen Datenquellen bereit. Die Verwaltung und die Anforderungen an die Ressourcen am entfernten Standort sind minimal, was MobiLink für eine Vielzahl von mobilen Anwendungen empfiehlt. Am Ende der einzelnen Synchronisationssitzungen sind die Datenbanken konsistent.
- **SQL Remote** ist eine nachrichtenbasierte Technologie für die Zwei-Wege-Replikation von Datenbanktransaktionen. Sie ist für die Zwei-Wege-Replikation ausgelegt, die einen konsolidierten Datenserver und eine große Anzahl von entfernten Datenbanken einbezieht. Die Verwaltung und die Anforderungen an die Ressourcen am entfernten Standort sind minimal, was SQL Remote für mobile Datenbanken empfiehlt.

In der folgenden Tabelle werden die Merkmale von MobiLink und SQL Remote zusammengefasst.

Synchronisations technologie	Anzahl der Datenbanken	Verbindung	Häufigkeit	Konsolidierte Datenbankentypen
MobiLink	Groß	Gelegentlich	Mittel	Viele Optionen
SQL Remote	Groß	Gelegentlich	Niedrig	SQL Anywhere

MobiLink-Merkmale

MobiLink ist für Synchronisationsinstallationen mit folgenden Anforderungen ausgelegt:

- **Große Anzahl von entfernten Datenbanken** MobiLink ist dafür konzipiert, eine große Anzahl von entfernten Datenbanken zu unterstützen. Zehntausende gleichzeitige Synchronisationen können damit abgewickelt werden.
- **Fallweise verbunden** MobiLink unterstützt Datenbanken, die nur fallweise oder indirekt mit dem Netzwerk verbunden sind, in dem der Server läuft.
- **Unterstützung konsolidierter Datenbanken** MobiLink unterstützt alle Datenspeicher als zentrale Datenquelle. Die entfernten Datenspeicher müssen SQL Anywhere- oder UltraLite-

Datenbanken sein. Das Schema der entfernten Standorte kann sich von dem der konsolidierten Datenbank unterscheiden, da Sie den Synchronisationsprozess durch das Schreiben von Skripten steuern.

- **Flexibler Synchronisationsplan** Anwendungen verbinden und synchronisieren üblicherweise in Intervallen von Sekunden, Minuten, Stunden oder Tagen.

SQL Remote-Merkmale

SQL Remote ist für Synchronisationsinstallationen mit folgenden Anforderungen ausgelegt:

- **Große Anzahl von entfernten Datenbanken** SQL Remote ist dafür konzipiert, eine große Anzahl von entfernten Datenbanken zu unterstützen. Das System kann Tausende entfernter Datenbanken in einer einzelnen Installation unterstützen, weil die Nachrichten von vielen entfernten Standorten gleichzeitig vorbereitet werden können.
- **Fallweise verbunden** SQL Remote unterstützt Datenbanken, die nur fallweise oder indirekt mit dem Netzwerk verbunden sind.
- **Niedrige bis hohe Latenzzeit** Eine hohe Latenzzeit bedeutet, dass eine große Zeitspanne zwischen der Eingabe von Daten in eine Datenbank und der Synchronisation mit jeder Datenbank im System vorliegt. Mit SQL Remote werden Replikationsnachrichten in Abständen von Sekunden, Minuten, Stunden oder Tagen versendet.
- **Niedriges bis mittleres Verkehrsaufkommen** Da Replikationsnachrichten gelegentlich versendet werden, kann eine große Transaktionsmenge an den einzelnen entfernten Standorten zu einem sehr großen Volumen von Nachrichten führen. SQL Remote ist am besten für Systeme mit relativ geringem Volumen von replizierten Daten pro entfernter Datenbank geeignet. Am konsolidierten Standort kann SQL Remote allerdings die Nachrichten effizient für mehrere Standorte gleichzeitig vorbereiten.
- **Homogene Datenbanken** SQL Remote unterstützt SQL Anywhere-Datenbanken. Jede Datenbank im System muss mit einem ähnlichen Schema aufgebaut sein.

Vorteile der Datensynchronisation

Verfügbarkeit der Daten

Einer der Hauptvorteile des Datensynchronisationssystems liegt darin, dass Daten lokal verfügbar gehalten werden, ohne dass dafür jedes Mal teure, nicht besonders zuverlässige und langsame Verbindungen zu einer einzelnen zentralen Datenbank erforderlich sind. Die Daten sind sogar ohne Verbindung zu einer zentralen Datenbank lokal verfügbar, und daher sind Sie bei einem Ausfall der Netzwerkverbindung nicht von den Daten abgeschnitten.

Antwortzeit

Die Synchronisation verbessert die Antwortzeiten bei Datenanforderungen aus zwei Gründen. Die Abfrate ist höher, weil die Anforderungen auf einem lokalen Server ohne Zugriff auf ein

Weitbereichsnetzwerk verarbeitet werden. Zusätzlich wird dem zentralen Datenbankserver Arbeit abgenommen, was die Konkurrenz um Prozessorzeit verringert.

Anforderungen an Synchronisationstechnologien

Jede Synchronisationstechnologie muss sich diversen Anforderungen stellen, die einzuhalten sind, damit der Vorteil der Flexibilität durch die Synchronisation zum Tragen kommt.

Transaktionsintegrität

Eine der wichtigsten Herausforderungen an ein Synchronisationssystem ist die Garantie, dass die Integrität der Transaktionen in den einzelnen Datenbanken zu jedem Zeitpunkt gewährleistet ist.

SQL Remote repliziert Teile des Transaktionslogs so, dass die Transaktionen während der Synchronisation aufrechterhalten werden: entweder wird eine komplette Transaktion repliziert oder gar keine. Dadurch wird die Integrität der Transaktionen in allen in das System einbezogenen Datenbanken gewährleistet.

In MobiLink können Sie sich ebenfalls für die Replikation jeder Transaktion entscheiden, aber standardmäßig mischt MobiLink mehrere Transaktionen in der entfernten Datenbank und übernimmt sie in einem einzigen Vorgang. Dies führt im Allgemeinen zu einem effizienteren Upload. In beiden Fällen behält MobiLink die Transaktionsintegrität bei.

Datenkonsistenz

Eine weitere Herausforderung, der sich ein Synchronisationssystem stellen muss, ist die Aufrechterhaltung der Datenkonsistenz im gesamten System. Synchronisationssysteme behalten eine **lose Konsistenz** im kompletten System: Das bedeutet, dass alle Änderungen zu jedem Standort irgendwann konsistent repliziert werden, dass aber zu einem gegebenen Zeitpunkt an unterschiedlichen Replikationsstandorten unterschiedliche Datenbestände vorhanden sein können.

Siehe auch

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

Konsolidierte und entfernte Datenbanken

Sowohl MobiLink als auch SQL Remote stehen für die Datensynchronisation zwischen einer konsolidierten Datenbank und einer Gruppe von entfernten Datenbanken zur Verfügung.

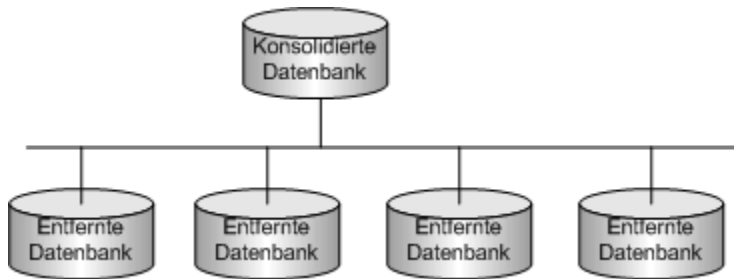
In MobiLink ist die **konsolidierte Datenbank** eines von mehreren unterstützten relationalen Datenbank-Managementsystemen. Die konsolidierte Datenbank, die in der Regel in einem Unternehmensserver untergebracht ist, protokolliert die Synchronisationsinformationen und enthält optional auch die zu replizierenden Daten. Andere zentrale Daten können in einem anderen Format gespeichert sein, etwa in einer nicht-relationalen Datenbank, einem Webdienst oder einer Textdatei.

MobiLink bietet auch die direkte Zeilenbehandlung, mit der die Datensynchronisation zu konsolidierten Datenquellen möglich ist, die nicht als relationale Datenbanken eingerichtet wurden, z.B. ERP-Systeme oder Anwendungsserver.

In SQL Remote sind alle Daten, die synchronisiert werden müssen, in einer konsolidierten SQL Anywhere-Datenbank enthalten.

Eine **entfernte Datenbank** kann entweder auf demselben Standort ausgeführt werden wie die konsolidierte Datenbank oder an einem physisch entfernten Standort, wie zum Beispiel einem Handheld. Die entfernte Datenbank kann alle bzw. einen Teil der Daten aus der konsolidierten Datenbank nutzen.

In der Abbildung wird eine schematische Darstellung einer kleinen Synchronisationsanordnung gezeigt.



Entfernte Benutzer

Ein typisches Synchronisationssystem enthält mehrere entfernte Datenbanken. Jede entfernte Datenbank enthält eine Teilmenge der Daten in der zentralen Datenbank. Jede entfernte Datenbank ist eine physisch eigenständige Datenbank, gewöhnlich auf einem eigenen Computer oder einem mobilen Gerät. Alle entfernten Datenbanken müssen mit der zentralen Datenbank konsistent sein.

Der komplette Synchronisationsaufbau kann als eine einzelne verteilte Datenbank angesehen werden, in der die Master-Kopie aller gemeinsam genutzten Daten in einer zentralen Datenbank aufbewahrt wird.

Jede entfernte Datenbank, die mit der zentralen Datenbank synchronisiert wird, ist ein entfernter Benutzer der zentralen Datenbank. Wenn ein entfernter Standort ein Server mit mehreren Benutzern ist, wird der gesamte Standort als einzelner entfernter Benutzer der zentralen Datenbank angesehen.

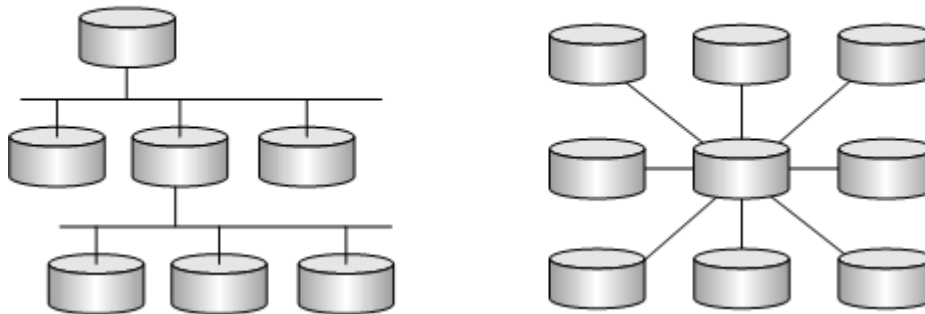
Hierarchische Datenbankkonfigurationen

Bei Datenbanken in einer **hierarchischen Konfiguration** hat jede Datenbank eine übergeordnete Datenbank, ausgenommen die konsolidierte Datenbank.

SQL Remote unterstützt hierarchische Konfigurationen von Datenbanken, nicht aber die Synchronisation unter gleichrangigen Datenbanken oder nicht-hierarchische Konfigurationen. MobiLink wird gewöhnlich ebenfalls in hierarchischen Konfigurationen verwendet, kann aber auch in anderen Konfigurationen eingesetzt werden.

Für zwei Datenbanken, die in einer hierarchischen Konfiguration direkt auf Daten zugreifen, gilt, dass eine immer der anderen über- oder untergeordnet sein muss.

Hierarchische Datenbank-konfigurationen



Datenbanken in nicht-hierarchischen Konfigurationen haben keine definierte Rangstellung über- oder untereinander.

Nicht-hierarchische Konfigurationen



In einer Installation mit MobiLink oder SQL Remote enthält jede Datenbank alle oder einen Teil der Daten, die von der übergeordneten Datenbank repliziert werden.

Entfernte Datenbanken können Tabellen enthalten, die in der konsolidierten Datenbank nicht vorkommen, sofern sie nicht in den Synchronisationsprozess einbezogen werden. SQL Remote verlangt, dass die Tabellen- und Spaltennamen in den entfernten Datenbanken mit denen in der konsolidierten Datenbank übereinstimmen. Im Gegensatz dazu erlaubt MobiLink das Speichern von Daten in anderen Spalten und Tabellen in den entfernten Datenbanken als in der konsolidierten Datenbank, was zu einer höheren Flexibilität führt.

Weitergabemethoden

Wenn eine Transaktion gemeinsam genutzte Daten in einer Datenbank ändert, muss die Transaktion bzw. müssen die Änderungen in die anderen Datenbanken im Synchronisationssystem repliziert werden. Es gibt mehrere Methoden, diese Aufgabe durchzuführen.

Zwei-Wege-Synchronisation

Alle SQL Anywhere-Synchronisationstechnologien ermöglichen die Zwei-Wege-Synchronisation. Die in der zentralen Datenbank durchgeführten Änderungen werden an entfernte Datenbanken übermittelt. Änderungen in entfernten Datenbanken werden an die zentrale Datenbank und an andere entfernte

Datenbanken übertragen. MobiLink ermöglicht eine reine Upload-Synchronisation, reine Download-Synchronisation oder Zwei-Wege-Synchronisation.

Sowohl SQL Remote als auch MobiLink erlauben es Ihnen, dieselben Daten gleichzeitig an mehreren Standorten zu ändern, und beide stellen Verfahren zur Verfügung, um auftretende Konflikte zu beheben.

Sitzungsbasierte Synchronisation: MobiLink

In einem **sitzungsbasierten** oder **synchronen** Synchronisationsschema wird die Synchronisation in Realzeit über eine direkte Kommunikationsverbindung durchgeführt. Die Verbindung kann zum Beispiel über ein Modem, ein Netzwerk oder ein Funkmodem erfolgen. Entfernte Standorte verbinden sich in Intervallen von Sekunden, Minuten, Stunden, Tagen oder Wochen.

Ein sitzungsbasierter Synchronisationsprozess ist mit einem Telefongespräch vergleichbar, bei dem auf beiden Seiten alle offenen Probleme gelöst werden. Der Prozess hält sich an ein bestimmtes Format. Ein entfernter MobiLink-Standort beginnt der Prozess, indem eine Verbindung zu einem MobiLink-Server geöffnet und eine vollständige Liste aller Änderungen heruntergeladen wird, die in der entfernten Datenbank seit der letzten Synchronisation durchgeführt wurden. Nach Erhalt der Daten aktualisiert der Server die zentrale Datenbank und sendet dann alle relevanten Änderungen zurück. Der entfernte Standort übernimmt die gesamten Änderungen, schickt eine Bestätigung zurück und schließt die Verbindung.

Nachrichtenbasierte Synchronisation: SQL Remote

SQL Remote ist ein **asynchrones Synchronisationsschema**. Es verwendet Nachrichten, um Daten zwischen Datenbanken auszutauschen. Nachrichten sind in der Regel Dateien oder speziell formatierte E-Mailnachrichten. Ein **Nachrichtenagent** in jeder Datenbank versendet Nachrichten, die eigene Datenänderungen betreffen. Derselbe Agent empfängt auch Nachrichten von einer oder mehreren Datenbanken und ändert die eigene Datenbank entsprechend dem Inhalt dieser Nachrichten.

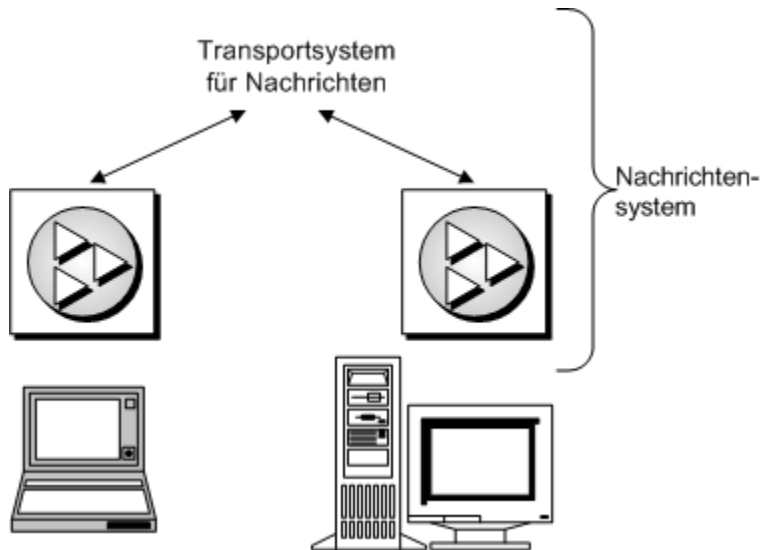
Im Datenverkehr auf Nachrichtenbasis ist jede einzelne Nachricht mit der Zieladresse und anderen Steuerungsdaten versehen, sodass für den Datenaustausch keine direkte Verbindung zwischen Anwendungen erforderlich ist. Beispiel: Eine E-Mailnachricht enthält die Zieladresse. Es gibt keine direkte Verbindung zwischen dem sendenden Server und dem Empfänger.

Nachrichtendienste benutzen die Datenübertragung über Zwischenspeicher (Store-and-Forward).

Ebenso wie sich Client/Server-Anwendungen auf Sitzungsbasis auf die Stacks eines Netzwerkprotokolls verlassen, wie z.B. TCP/IP, so verlassen sich nachrichtengesteuerte Anwendungen auf Nachrichtendienste wie Internet Simple Mail Transfer Protocol (SMTP), oder auf eine einfache Verbindung mit gemeinsam genutzten Dateien.

Die Nachrichtendienste benutzen **Store-and-Forward-Methoden** zur Übermittlung der Nachrichten an den Empfänger. Beispiel: E-Mail-Systeme speichern Nachrichten, bis der Empfänger seinen Posteingang öffnet, um seine E-Mail zu lesen. Anschließend wird die Nachricht aus dem Zwischenspeicher an den Endadressaten gestellt.

Der Aufbau eines Synchronisationssystems auf einem Nachrichtensystem bedeutet, dass ein nachrichtenbasiertes Synchronisationssystem wie SQL Remote kein eigenes Store-and-Forward-System einführen muss, um die Nachrichten an das Ziel zu bringen. Genauso wie sitzungsbasierte Client/Server-Anwendungen nicht ihre eigenen Protokoll-Stacks einrichten, um Daten zwischen dem Client und dem Server auszutauschen, benutzt SQL Remote bestehende Nachrichtensysteme, um die Nachrichten zu übermitteln.



Garantierte Zustellung

Um verlässlich zu arbeiten, muss ein nachrichtenbasiertes Synchronisationssystem sicherstellen, dass alle Nachrichten ihr Ziel erreichen und in der Reihenfolge übernommen werden, in der sie verschickt wurden. SQL Remote enthält ein Protokoll, mit dem gewährleistet wird, dass die Synchronisationsaktualisierungen in der richtigen Reihenfolge übernommen werden.

Hinweise zur Synchronisationstechnologie

Jede SQL Anywhere-Synchronisationstechnologie bietet sich für einen bestimmten Anwendungstyp an. Die folgende Beschreibung vergleicht die verschiedenen Technologien, um Sie bei der Auswahl der für Sie geeignetsten zu unterstützen.

Finden Sie heraus, welche der folgenden Überlegungen für Ihre Anwendung wichtig sind.

Ihr konsolidiertes Datenbanksystem

In einer typischen Synchronisationsumgebung dient eine große Datenbank als zentrale Sammelstelle für die Daten. Manchmal sind Sie in der Lage, ein Datenbanksystem auszuwählen, das Ihren Anforderungen entspricht. Dann wieder gibt es bereits eine zentrale Datenbank und Sie müssen das Synchronisationssystem an sie anpassen.

MobiLink kann mit vielen verbreiteten Datenbankservern arbeiten, wie zum Beispiel SQL Anywhere, Sybase Adaptive Server Enterprise, Oracle, Microsoft SQL Server und IBM DB2. Mit MobiLink-Server-APIs für .NET und Java können Sie eine Synchronisation mit jeder Datenquelle einrichten, darunter Anwendungsserver, Webserver, Textdateien und andere Datenbankprodukte.

In einem SQL Remote-System muss die zentrale Datenbank SQL Anywhere sein.

Ihr entferntes Datenbanksystem

SQL Anywhere-Synchronisationstechnologien unterscheiden sich auch in den Typen der entfernten Datenbanken, die sie unterstützen.

MobiLink unterstützt SQL Anywhere und UltraLite als entfernte Datenbanken.

SQL Remote unterstützt entfernte SQL Anywhere-Datenbanken.

Netzwerkmerkmale

MobiLink und SQL Remote sind beide für gelegentlich verbundene Umgebungen geeignet, bei denen entfernte Standorte für Stunden oder Tage isoliert arbeiten und synchronisieren, sobald eine Netzwerkverbindung verfügbar ist.

MobiLink ist sitzungsbasiert. Eine Realzeit-Verbindung ist während der Synchronisation erforderlich. Wenn diese Verbindung unterbrochen wird, bevor die Synchronisation abgeschlossen ist, wird der Prozess erst bei der nächsten Synchronisation zu Ende geführt. Im Gegensatz dazu übermittelt SQL Remote die Daten anhand von Nachrichten, die asynchron gesendet oder empfangen werden können. Diese Nachrichten nehmen die Form von Dateien auf einer Festplatte oder von E-Mailnachrichten an. Sie können abgearbeitet werden, wann immer sie eintreffen, wodurch die Synchronisation inkrementell durchgeführt werden kann.

Häufigkeit der Synchronisation

In manchen Situationen ist es unumgänglich, dass Ihre Daten sofort repliziert werden. In anderen Fällen wiederum genügt es, wenn die Synchronisation ein- bis zweimal pro Tag stattfindet. Eine häufigere Synchronisation ist möglicherweise undurchführbar, weil keine Netzwerkverbindung zur Verfügung steht.

MobiLink und SQL Remote sind vor allem für Systeme vorgesehen, in denen die Synchronisation nur gelegentlich erfolgt, etwa ein paarmal pro Tag, sie können aber auch für häufige Synchronisationen in Abständen von bis zu wenigen Sekunden verwendet werden.

Die Anzahl der entfernten Standorte

MobiLink und SQL Remote funktionieren auch mit zahlreichen entfernten Benutzern. Die Skalierbarkeit von MobiLink wird nur durch die Skalierbarkeit des konsolidierten Datenbank-Managementsystems eingeschränkt. Das nachrichtenbasierte System von SQL Remote ermöglicht es Ihnen, eine typische Installation mit Tausenden von entfernten Benutzern zu betreiben.

Es gibt bei keinem dieser Systeme eine festgelegte Obergrenze für die Anzahl entfernter Datenbanken. Die tatsächliche Anzahl hängt von der Menge der zu replizierenden Daten, der Synchronisationsfrequenz und dem Design Ihrer Anwendung ab.

Reihenfolge der Transaktionen

Bei MobiLink werden die Ergebnisse von mehreren Transaktionen in der entfernten Datenbank standardmäßig in einer Gruppe von Änderungen zusammengefasst, die dann in der konsolidierten Datenbank übernommen werden. Sie können aber auch die Option wählen, die Reihenfolge der Transaktionen beizubehalten und in einem separaten Upload zu synchronisieren. Da die Synchronisation in beiden Fällen an einer Transaktionsgrenze stattfindet, wird die referenzielle Integrität gewahrt. Nicht festgeschriebene Daten werden nie synchronisiert, daher bleibt die Datenintegrität gewährleistet.

SQL Remote repliziert Daten, indem das Transaktionslog durchsucht wird und entsprechende Nachrichten für jede Transaktion vorbereitet werden. Diese Nachrichten werden sortiert und an den entfernten oder konsolidierten Computer gesendet. Beim Verarbeiten eingehender Nachrichten geht SQL Remote immer in der gleichen Reihenfolge vor, in der diese in die andere Datenbank übernommen wurden. Falls notwendig, wird die Verarbeitung einer Nachricht automatisch solange verzögert, bis alle früheren Nachrichten übernommen wurden.

Datenkonsistenz zu einem bestimmten Zeitpunkt

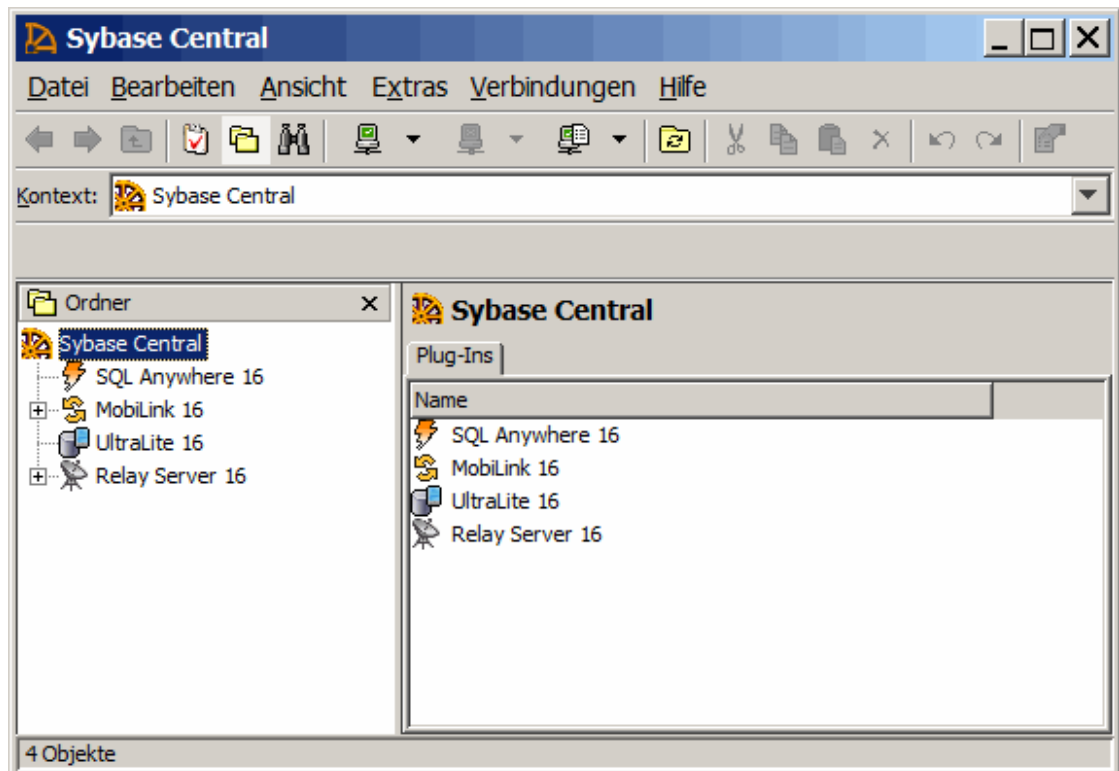
Unmittelbar nach jeder MobiLink -Synchronisationssitzung sind die Daten in den beiden Datenbanken konsistent. Die Möglichkeit, die Konsistenz der Daten an einem entfernten Standort zu einem bestimmten Zeitpunkt sicherzustellen, ist ein Vorteil der sitzungsbasierten Synchronisation mit MobiLink. Wenn es zum Beispiel wichtig ist, dass die Daten an einem entfernten Standort mit den Daten in der konsolidierten Datenbank zu einem bestimmten Zeitpunkt übereinstimmen (z.B. um Punkt 10:00 Uhr), kann das erreicht werden, indem die Synchronisation vor dieser Uhrzeit durchgeführt wird. Solange die Synchronisation erfolgreich abschließt, ist die Gleichwertigkeit der Daten am entfernten Standort sichergestellt.

Wenn die Änderungen an Daten durch einen Austausch von Nachrichten repliziert werden, ist es schwierig sicherzustellen, dass die Daten an einem entfernten Standort mit den Daten in dem konsolidierten Computer zu einem bestimmten Zeitpunkt vollständig konsistent sind. So gehen zum Beispiel manchmal Nachrichten während der Übertragung verloren. SQL Remote erkennt automatisch diesen Fehler und sendet die Nachricht erneut, aber solche Unterbrechungen können zu unerwarteten Verzögerungen führen.

Design- und Administrationstools

Nachstehend werden die Design- und Administrationstools beschrieben, die in SQL Anywhere integriert sind.

- **Sybase Central - zentrale Steuerung und Administration** Bei Sybase Central handelt es sich um ein integriertes Datenbankverwaltungs- und Administrationstool, das die Datenbankeinstellungen, Eigenschaften und Dienstprogramme in einer grafischen Benutzeroberfläche anzeigt. Über Plug-Ins kann Sybase Central benutzt werden, um den SQL Anywhere-Server, UltraLite, MobiLink, Relay Server und andere Sybase-Produkte zu verwalten.



Sybase Central erleichtert nicht nur Routineaufgaben, sondern bietet auch Performancestatistiken, Prozedurprofilerstellung, die Fehlersuche in gespeicherten Prozeduren sowie die Verwaltung von Ereignissen und Zeitplänen, Webdiensten und Verbindungsprofilen. Sybase Central hilft bei der Durchführung aller Aufgaben, die mit SQL-Anweisungen auf dem Datenbankserver oder durch SQL Anywhere-Dienstprogramme durchgeführt werden. Siehe „[Sybase Central](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

Eine Reihe von Sybase Central-Tools, die Sie bei der Analyse und Überwachung der laufenden Performance Ihrer SQL Anywhere-Datenbank unterstützen, stehen zur Verfügung. Mit diesen Tools bewältigen Sie Aufgaben wie Profilerstellung von Prozeduren, grafische Pläne, Abfrageausführung, Systemmonitor, Anforderungsprotokollierung oder Ablaufplanung. Außerdem bietet Sybase Central:

- **Unterstützung räumlicher Daten** - Räumliche Daten sind Daten, die die Position, Form und Ausrichtung von Objekten in einem definierten Raum beschreiben. Räumliche Daten in SQL Anywhere werden als 2D-Geometrien in Form von Punkten, Kurven (Linienfolgen und Folgen von Kreisbögen) und Polygonen dargestellt. Siehe [SQL Anywhere Server - Unterstützung für räumliche Daten](#).
- **Anwendungsprofilerstellung mit dem Assistenten für die Anwendungsprofilerstellung** Mit dem **Assistenten für die Anwendungsprofilerstellung** in Sybase Central können Sie folgende Aufgaben automatisch durchführen:
 - Profile von gespeicherten Prozeduren, Funktionen, Triggern und Ereignissen erstellen
 - Empfehlungen erhalten, die die Performance Ihrer Datenbankanwendung betreffen
 - Datenbankaktivität aufzeichnen, während Ihre Anwendung läuft

Siehe „Anwendungsprofilerstellung“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)].

- **Erweiterte Anwendungsprofilerstellung im Modus Anwendungsprofilerstellung** Sie können die Systemperformance verbessern, indem Sie den **Assistenten für die Datenbankprotokollierung** und den Modus **Anwendungsprofilerstellung** in Sybase Central für folgende Aufgaben verwenden:
 - Cachegröße und Indizes aufgrund gemessener Werte der Datenbankperformance anpassen
 - Ermitteln, wann Deadlocks auftreten
 - Sperraktivitäten beobachten
 - Ausführungspläne überprüfen
 - Jede einzelne Anweisung in einer Anwendung protokollieren, um sie zu diagnostizieren und zu optimieren

Siehe „Diagnoseprotokollierung“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)].

- **Auswahl und Optimierung von Indizes mit dem Indexberater** Der Indexberater analysiert die aktuelle Auslastung und gibt Empfehlungen für die Indexauswahl zur Verbesserung der Performance. Der Indexberater kann aus Sybase Central oder Interactive SQL aufgerufen werden. Siehe „Indexberater“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)].
- **Interactive SQL - SQL-Abfrage-Editor** Interactive SQL ist ein Datenbank-Dienstprogramm, mit dem Sie SQL-Anweisungen ausführen und Daten aus der Datenbank anzeigen können. Der integrierte Abfrage-Editor und andere Tools, wie beispielsweise die Anzeige des grafischen Plans, unterstützen Sie bei der Analyse, Korrektur und Optimierung von Abfragen. Siehe „Interactive SQL“ [[SQL Anywhere Server - Datenbankadministration](#)].
- **SQL Anywhere-Monitor** Der Monitor ist ein browserbasiertes Administrationstool, das Informationen über den Zustand und die Verfügbarkeit von SQL Anywhere-Datenbanken, MobiLink-Servern, MobiLink-Serverfarmen und Relay Server-Farmen bereitstellt. Der Monitor bietet eine kontinuierliche Datensammlung, E-Mail-Alarmbenachrichtigungen, eine browserbasierte Schnittstelle

und die Möglichkeit, mehrere Datenbanken, MobiLink-Server, MobiLink-Serverfarmen und Relay Server-Farmen zu überwachen. Siehe „[SQL Anywhere-Monitor](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

- **MobiLink-Monitor - Synchronisationsüberwachung** Der MobiLink-Monitor ist ein grafisches Administrationstool, das Ihnen detaillierte Informationen über die Performance Ihrer MobiLink-Synchronisationen liefert. Der MobiLink-Monitor sammelt Details und Statistiken über alle Synchronisationen, wie etwa Start- und Endzeit, Datenvolumen der Uploads und Downloads, erfolgreiche Ausführungen, Konflikte usw. Siehe „[MobiLink-Profiler](#)“ [[MobiLink - Serveradministration](#)].
- **Relay Server** Der Relay Server ermöglicht die sichere und ausgewogene Kommunikation zwischen mobilen Geräten und Backend-Servern, die über einen Webserver Daten austauschen. Zu den unterstützten Backend-Servern gehören MobiLink, SQL Anywhere, Afaria, Mobile Office, Sybase Unwired Platform und Unwired Server. Siehe [Relay Server](#).
- **Dienstprogramme** SQL Anywhere wird mit zahlreichen Dienstprogrammen für Administrationsaufgaben geliefert, beispielsweise für die Sicherung einer Datenbank oder für die Durchführung von Synchronisationen. Dienstprogramme können in Batchdateien zur Automatisierung von Aufgaben verwendet werden. Siehe:
 - „[Dienstprogramme für die Datenbankadministration](#)“ [[SQL Anywhere Server - Datenbankadministration](#)]
 - „[UltraLite-Dienstprogramme](#)“ [[UltraLite - Datenbankverwaltung](#)]
 - „[MobiLink-Dienstprogramme](#)“ [[MobiLink - Serveradministration](#)]
 - „[SQL Remote-Dienstprogramme und Optionen](#)“ [[SQL Remote](#)]
- **SAP Sybase PowerDesigner – Physical Data Model** SQL Anywhere enthält Physical Data Model, ein Modul des leistungsstarken Datenbankdesigntools SAP Sybase PowerDesigner. Dieses Modul stellt Funktionen bereit, mit denen Sie Datenbanken mithilfe der grafischen Darstellung eines Datenbankschemas erstellen bzw. modifizieren können. Sie können eine Datenbank optimieren, indem Sie alle enthaltenen Tabellen, Spalten, Indizes, Schlüssel, Ansichten, Speichermedien, Trigger und gespeicherte Prozeduren anpassen.

SAP Sybase PowerDesigner Physical Data Model enthält eine umfassende Dokumentation mit praktischen Einführungen im Videoformat. Weitere Hinweise zu SAP Sybase PowerDesigner finden Sie unter <http://www.sybase.com/products/modelingmetadata/powerdesigner>.

Hinweise zum SQL Anywhere-Datenbankdesign finden Sie unter „[Datenbankerstellung](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

Beispieldatenbanken

In diesem Abschnitt werden die Schemata der SQL Anywhere 16-Beispieldatenbanken beschrieben. Experimentieren Sie mit den Beispieldatenbanken, um mehr über SQL Anywhere 16 zu erfahren.

SQL Anywhere-Beispieldatenbank

Aus Gründen der Einfachheit und Konsistenz wird für viele Beispiele in der Dokumentation die SQL Anywhere-Beispieldatenbank `%SQLANYAMP16%\demo.db` verwendet. Standardmäßig ist diese Datei im Beispielverzeichnis von SQL Anywhere installiert (z.B. `C:\Users\Public\Documents\SQL Anywhere 16\Samples` unter Windows 7).

Für die Beispieldatenbanken gilt folgende Kombination aus Benutzer-ID und Kennwort:

- Benutzer-ID = **DBA**
- Kennwort = **sql** (Kennwörter in SQL Anywhere berücksichtigen die Groß- und Kleinschreibung)

Vorsicht

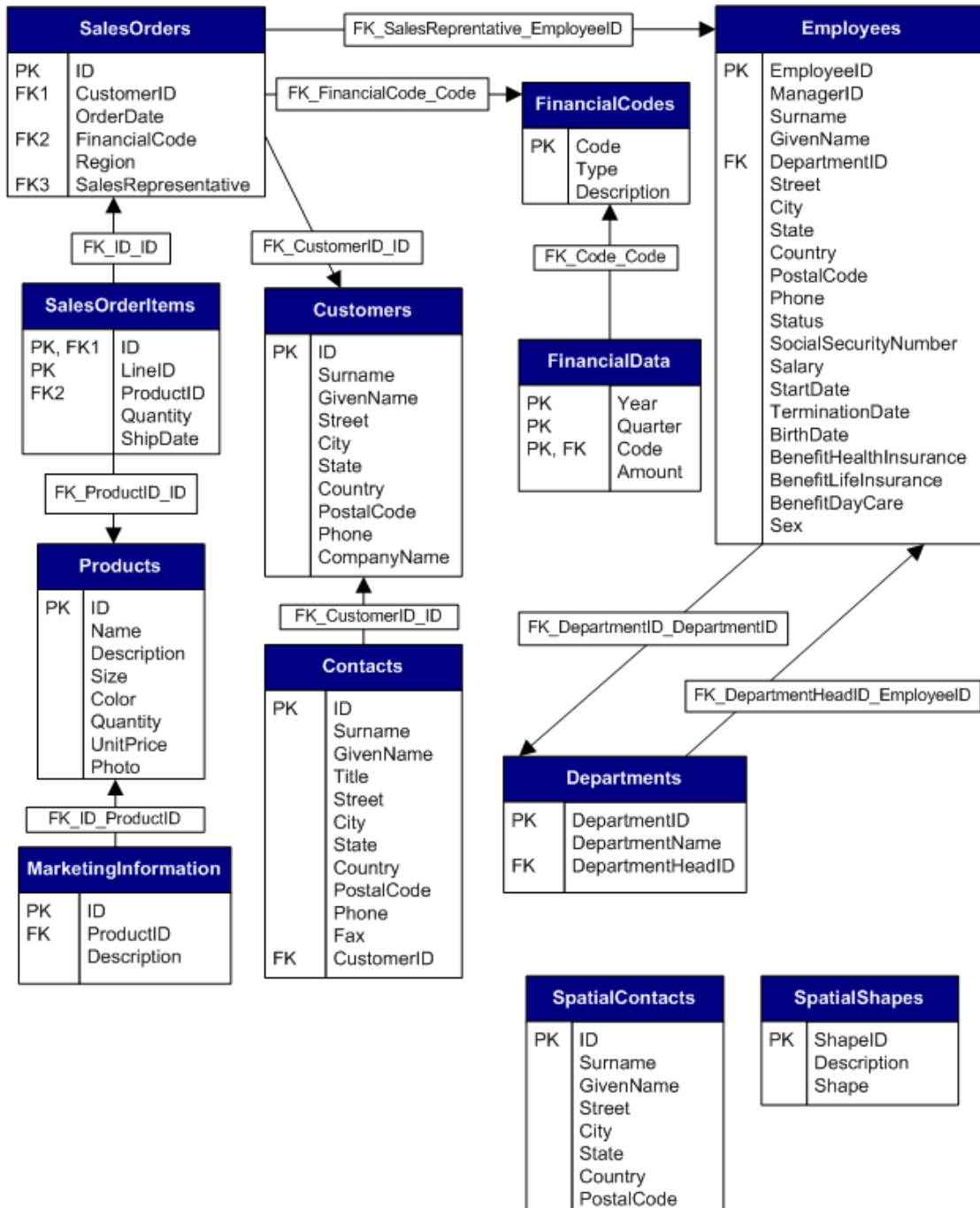
Alle Beispieldatenbanken verwenden dieselbe Kombination aus Benutzer-ID und Kennwort. Es wird empfohlen, dass Sie die Benutzer-ID DBA und das Kennwort sql ändern, um den Zugriff auf diese Datenbank zu beschränken. Siehe „[Kennwörter ändern \(Sybase Central\)](#)“ [*SQL Anywhere Server - Datenbankadministration*].

Die Beispieldatenbank verwendet folgende ODBC-Datenquelle: **SQL Anywhere 16 Demo**.

Die Beispieldatenbank stellt ein kleines amerikanisches Unternehmen dar, das Sportbekleidung verkauft. Sie enthält interne Informationen über das Unternehmen, wie Mitarbeiter (employees), Abteilungen (departments) und Finanzdaten (finances) sowie Produkt- und Verkaufsinformationen (sales orders, customers und contacts). Alle Daten in der Datenbank sind frei erfunden.

In der nachstehend abgebildeten Strukturdarstellung werden die Tabellen der Beispieldatenbank und ihre Beziehungen zueinander erläutert. Die Kästen stellen die Tabellen dar, die Pfeile zeigen die Fremdschlüsselbeziehungen zwischen den Tabellen an. PK steht für Primärschlüssel (engl. Primary Key), FK für Fremdschlüsse (engl. Foreign Key).

Hinweise zur Verbindung mit `demo.db` finden Sie unter „[Praktische Einführung: Verbindung mit der Beispieldatenbank herstellen](#)“ [*SQL Anywhere Server - Datenbankadministration*].



Rollen in der Beispieldatenbank

Die Beispieldatenbank enthält eine benutzererweiterte Rolle (GROUPO) und drei benutzerdefinierte Rollen (READ_ROLE, MODIFY_ROLE und EXEC_ROLE).

GROUPO ist ein Benutzer, der zu einer Rolle erweitert wurde. GROUPO ist der Eigentümer der oben beschriebenen Tabellen. Er ist auch der Eigentümer einer Anzahl von Beispielansichten, materialisierten Ansichten und gespeicherten Prozeduren, die in der Beispieldatenbank definiert sind. Jeder neue Benutzer, dem die Rolle GROUPO erteilt wird, kann diese Datenbankobjekte ohne den Eigentümerqualifizierer referenzieren. Er kann zum Beispiel die Tabelle "Departments" referenzieren, ohne den Eigentümerpräfix (GROUPO.Departments) anzugeben.

Jeder Benutzer, dem die Rolle READ_ROLE erteilt wird, hat Lesezugriff auf die Zeilen, die in den oben stehenden Tabellen beschrieben werden. Benutzer mit dieser Rolle können eine SELECT-Anweisung auf jede dieser Tabellen durchführen.

Jeder Benutzer, dem die Rolle MODIFY_ROLE erteilt wird, hat Schreibzugriff auf die Zeilen, die in den oben stehenden Tabellen beschrieben werden. Benutzer mit dieser Rolle können eine INSERT-, UPDATE- oder DELETE-Anweisung auf jede dieser Tabellen durchführen.

Jeder Benutzer, dem die Rolle EXEC_ROLE erteilt wird, kann jede der gespeicherten Beispielprozeduren ausführen, die in der Beispieldatenbank definiert sind, mit Ausnahme von ManageContacts (ShowContacts, ShowCustomers, ShowCustomerProducts, ShowProductInfo, ShowSalesOrders, ShowSalesOrderDetail, ShowSalesOrderItems und debugger_tutorial). Die gespeicherte Prozedur ManageContacts wird verwendet, um Kontakte in der Tabelle "Kontakte" hinzuzufügen, zu löschen oder zu aktualisieren.

Benutzer, die nur die Rollen READ_ROLE, MODIFY_ROLE oder EXEC_ROLE haben, können keine Schemaänderungen in der Datenbank durchführen. Beispiel: Sie können weder Spalten zu vorhandenen Tabellen hinzufügen oder aus diesen entfernen noch können sie Tabellen erstellen oder löschen. Dazu sind zusätzliche Privilegien erforderlich.

In einem großen Unternehmen gibt es viele solcher Rollen. Beispiel: Eine Verkaufsabteilung könnte Eigentümer der verkaufs- und kundenbezogenen Tabellen sein, eine Produktions- oder Einkaufsabteilung könnte Eigentümer der produktbezogenen Tabellen sein und eine Personalabteilung könnte Eigentümer der mitarbeiter- und abteilungsbezogenen Tabellen sein. Jede dieser Abteilungen würde ihre eigene Zusammenstellung von Rollen haben, mit verschiedenen Zugriffsrechten für die Tabellen, deren Eigentümer sie sind. Die Beispieldatenbank stellt jedoch ein kleines Unternehmen dar und die Verwendung von Rollen wurde absichtlich einfach gehalten.

Der Benutzer "DBA" hat alle Privilegien für die Beispieldatenbank und kann das Datenbankschema ändern, neue Benutzer erstellen, Rollen hinzufügen usw. Diesem Benutzer wurde die benutzererweiterte Rolle GROUPO erteilt, sodass er Beispieltabellen und gespeicherte Prozeduren referenzieren kann, ohne sie mit einer Eigentümer-ID zu qualifizieren.

Es wurden zwei weitere Benutzer erstellt, um die oben beschriebenen Rollen zu erläutern.

Dem Benutzer UPDATER (Kennwort "update") wurden die Rollen READ_ROLE, MODIFY_ROLE und EXEC_ROLE erteilt. Dieser Benutzer kann SELECT-, INSERT-, UPDATE- und DELETE-Anweisungen für alle Beispieltabellen durchführen. Dieser Benutzer hat das EXECUTE-Privileg für alle gespeicherten Beispielprozeduren. Die benutzererweiterte Rolle GROUPO wird außerdem erteilt, sodass dieser Benutzer Beispieltabellen und gespeicherte Prozeduren referenzieren kann, ohne sie mit einer Eigentümer-ID zu qualifizieren.

Dem Benutzer "BROWSER" (Kennwort "browse") wurden die Rollen READ_ROLE und EXEC_ROLE erteilt. Dieser Benutzer kann eine SELECT-Anweisung auf jede der Beispieltabellen durchführen, aber er

kann die Zeilen der Tabellen nicht ändern. Dieser Benutzer hat das EXECUTE-Privileg für alle gespeicherten Beispielprozeduren mit Ausnahme der gespeicherten Prozedur ManageContacts, da diese Prozedur zum Hinzufügen, Löschen und Aktualisieren von Kontakten in der Tabelle "Contacts" verwendet wird. (Zum Ausführen von ManageContacts ist die MODIFY_ROLE-Rolle erforderlich). Die benutzererweiterte Rolle GROUPO wird ebenfalls erteilt, sodass dieser Benutzer Beispieldatenbanken und gespeicherte Prozeduren referenzieren kann, ohne sie mit einer Eigentümer-ID zu qualifizieren.

Siehe auch

- „Beispieldatenbank starten (Windows Mobile)“ [[SQL Anywhere Server - Datenbankadministration](#)]

Neuerstellung der Beispieldatenbank (*demo.db*)

Das Testen von Funktionen und die Ausführung der praktischen Einführungen in der SQL Anywhere-Dokumentation führt manchmal zu Änderungen in der Beispieldatenbank, was die erfolgreiche Durchführung von nachfolgenden praktischen Einführungen und Tests verhindern kann. Wenn dies passiert, können Sie die Beispieldatenbank in ihrem ursprünglichen Zustand wiederherstellen. Wenn Sie die Beispieldatenbank in ihrem aktuellen Status behalten wollen, können Sie alternativ auch die Beispieldatenbank in ihrem ursprünglichen Zustand unter einem anderen Namen neu erstellen. Beide Methoden werden unten beschrieben.

- **Neuerstellung der Beispieldatenbank (Windows)** Führen Sie den folgenden Befehl aus, um *demo.db* zu löschen und eine neue Kopie der Beispieldatenbank mit Objekten und Daten zu erstellen:

```
newdemo "%SQLANYSAM16%\demo.db"
```

Wenn Sie dazu aufgefordert werden, bestätigen Sie das Löschen aller vorhandenen Dateien.

- **Neuerstellung der Beispieldatenbank (Unix)** Führen Sie den folgenden Befehl in dem Verzeichnis aus, in dem sich die Beispieldatenbank befindet, um *demo.db* zu löschen und eine neue Kopie der Beispieldatenbank mit Objekten und Daten zu erstellen:

```
newdemo.sh demo.db
```

Wenn Sie dazu aufgefordert werden, bestätigen Sie das Löschen aller vorhandenen Dateien.

- **Erstellen einer Kopie der Beispieldatenbank unter einem anderen Namen (Windows)** Führen Sie den folgenden Befehl aus, um eine Datenbank namens *mydemo.db* mit Objekten und Daten zu erstellen.

Wenn Sie keinen Pfad angeben, werden Datenbank und Transaktionslog im aktuellen Verzeichnis erstellt.

```
newdemo path\mydemo.db
```

- **Erstellen einer Kopie der Beispieldatenbank unter einem anderen Namen (Unix)** Führen Sie den folgenden Befehl aus, um eine Datenbank namens *mydemo.db* mit Objekten und Daten zu erstellen.

Wenn Sie keinen Pfad angeben, werden Datenbank und Transaktionslog im aktuellen Verzeichnis erstellt.

newdemo.sh path/mydemo.db

Die Beispieldatenbankanwendung CustDB

Die CustDB-Beispielanwendung ist ein nützliches Tool zum Lernen, wie UltraLite- und MobiLink-Anwendungen entwickelt werden. Die Beispieldatenbank ist eine Vertriebsanwendung für Hardware-Produzenten. Sie enthält Kunden-, Produkt- und Vertriebsdaten für das Unternehmen.

Die CustDB-Beispielanwendung besteht aus zwei Teilen:

- **UltraLite** Für UltraLite gilt, dass das Deployment der CustDB-Anwendung auf einem beliebigen von UltraLite unterstützten Gerät mit einer beliebigen von UltraLite unterstützten Plattform durchgeführt werden kann. Der gesamte Quellcode zum Erstellen der CustDB-Anwendung von UltraLite und zum Ausführen der Beispielanwendung wird angezeigt. Die CustDB-Beispielanwendung ist für die MobiLink-Synchronisation eingerichtet.

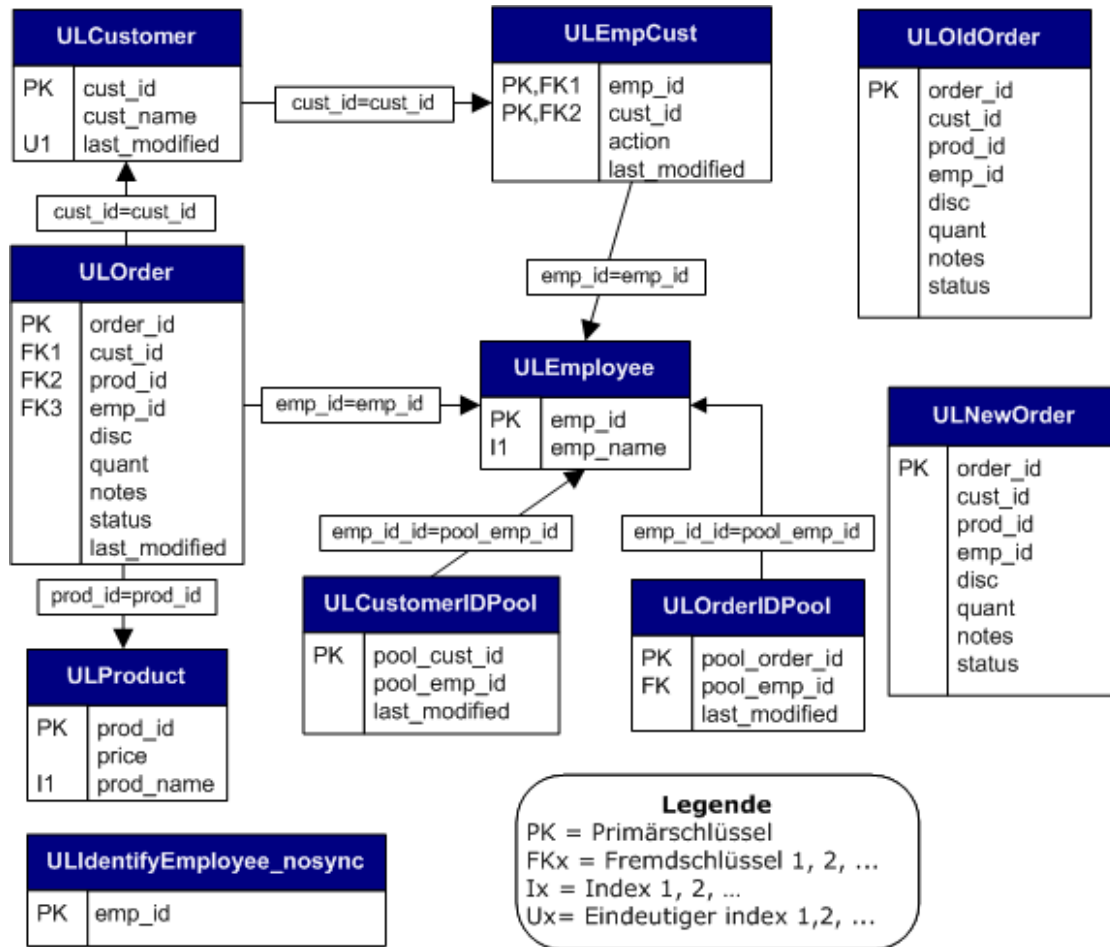
Die CustDB-Beispielanwendung von UltraLite befindet sich im Verzeichnis `%SQLANYSAMPI6%\UltraLite\CustDB\`.

- **MobiLink** Wenn Sie die MobiLink-Funktionen kennen lernen möchten, finden Sie in der konsolidierten CustDB-Datenbank ein Beispiel für eine Synchronisationslogik. Mit dieser konsolidierten Datenbank und der entfernten UltraLite-Beispieldatenbank CustDB können Sie das Beispielsynchronisationssystem ausführen.

Die konsolidierte CustDB-Datenbank von MobiLink wird erstellt, indem Skriptdateien mit einer unterstützten relationalen Datenbank (SQL Anywhere, Adaptive Server Enterprise, Oracle, Microsoft SQL Server oder DB2) ausgeführt werden. Diese Setupdateien befinden sich im Verzeichnis `%SQLANYSAMPI6%\MobiLink\CustDB\`.

Die konsolidierte CustDB-Beispieldatenbank verwendet folgende ODBC-Datenquelle: **SQL Anywhere 16 CustDB**.

Im folgenden Diagramm werden die Tabellen der CustDB-Datenbank und ihre Beziehungen zueinander erläutert.



Siehe auch

- „Praktische Einführung: Erstellen der Beispielanwendung CustDB“ [*UltraLite - Datenbankverwaltung*]
- „MobiLink-Beispiel CustDB“ [*MobiLink - Erste Orientierung*]

Liste der praktischen Einführungen in SQL Anywhere

Dieser Abschnitt enthält eine komplette Liste der praktischen Einführungen in SQL Anywhere. Sie können diese praktische Einführungen ausprobieren, um mehr über SQL Anywhere zu erfahren.

Praktische Einführungen in SQL Anywhere

- „Praktische Einführung: Verbindung mit der Beispieldatenbank herstellen“ [*SQL Anywhere Server - Datenbankadministration*]
- „Praktische Einführung: SQL Anywhere-Datenbank erstellen“ [*SQL Anywhere Server - Datenbankadministration*]
- „Praktische Einführung: Windows Mobile-Datenbanken über Sybase Central ausführen“ [*SQL Anywhere Server - Datenbankadministration*]
- „Praktische Einführung: Windows Mobile-Datenbanken mit Interactive SQL verwalten“ [*SQL Anywhere Server - Datenbankadministration*]
- „Praktische Einführung: Datenbankspiegelungssysteme erstellen“ [*SQL Anywhere Server - Datenbankadministration*]
- „Praktische Einführung: Datenbankspiegelungssysteme mit mehreren Datenbanken und einem gemeinsamen Arbiterserver erstellen“ [*SQL Anywhere Server - Datenbankadministration*]
- „Praktische Einführung: Scale-Out-System mit Schreibschutz erstellen“ [*SQL Anywhere Server - Datenbankadministration*]
- „Praktische Einführung: Überwachen von Ressourcen mit dem Monitor“ [*SQL Anywhere Server - Datenbankadministration*]
- „Praktische Einführung: Verwendung des Codebeispiels "Simple"“ [*SQL Anywhere Server - Programmierung*]
- „Praktische Einführung: Verwendung des Codebeispiels "Table Viewer"“ [*SQL Anywhere Server - Programmierung*]
- „Praktische Einführung: Eine einfache .NET-Datenbankanwendung mit Visual Studio entwickeln“ [*SQL Anywhere Server - Programmierung*]
- „Praktische Einführung: Java in der Datenbank verwenden“ [*SQL Anywhere Server - Programmierung*]
- „Praktische Einführung: Webserver erstellen und über einen Webclient darauf zugreifen“ [*SQL Anywhere Server - Programmierung*]
- „Praktische Einführung: Verwenden von SQL Anywhere für den Zugriff auf einen SOAP/DISH-Dienst“ [*SQL Anywhere Server - Programmierung*]

- „Praktische Einführung: Verwenden von Visual C# für den Zugriff auf einen SOAP/DISH-Webdienst“ [*SQL Anywhere Server - Programmierung*]
- „Praktische Einführung: Verwenden von JAX-WS für den Zugriff auf einen SOAP/DISH-Webdienst“ [*SQL Anywhere Server - Programmierung*]
- „Praktische Einführung: Mit den räumlichen Funktionen experimentieren“ [*SQL Anywhere Server - Unterstützung für räumliche Daten*]
- „Praktische Einführung: Testdatenbank für die praktischen Einführungen in die Anwendungsprofilerstellung erstellen“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Deadlocks diagnostizieren“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Langsame Anweisungen diagnostizieren“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Indexfragmentierung diagnostizieren“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Tabellenfragmentierung diagnostizieren“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Baselineing mit Prozedurprofilerstellung“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Volltextsuche in einem GENERIC-Textindex durchführen“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Eine Fuzzy-Volltextsuche durchführen“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Nicht-Fuzzy-Volltextsuche in einem NGRAM-Textindex durchführen“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Einrichten des Szenarios für die praktischen Einführungen zur Isolationsstufe“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Einführung in Dirty Reads“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Einführung in nicht-wiederholbare Lesevorgänge“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Einführung in Phantomzeilen“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Praktische Einführung: Einführung in Phantomsperren“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]

-
- „Praktische Einführung: Erste Schritte mit dem Debugger“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]

Praktische Einführungen in MobiLink

- „Praktische Einführung: Einführung in MobiLink“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: MobiLink mit einer konsolidierten SQL Anywhere-Datenbank verwenden“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: MobiLink mit einer Oracle 10g-Datenbank verwenden“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: MobiLink mit einer konsolidierten Adaptive Server Enterprise-Datenbank verwenden“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: Der Einsatz von Java oder .NET für die benutzerdefinierte Authentifizierung“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: Verwenden der direkten Zeilenbehandlung“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: Synchronisation mit Microsoft Excel“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: Synchronisation mit XML“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: Zentrale Administration von entfernten Datenbanken“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: Änderung eines Schemas mit der Skriptversionsklausel“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: Änderung eines Schemas mit der erweiterten Option ScriptVersion“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: Simulieren mehrerer MobiLink-Clients mit dem MobiLink Replay-Dienstprogramm“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: Verwenden des skriptgesteuerten Uploads“ [*MobiLink - Clientadministration*]
- „Praktische Einführung: Konfigurieren der serverinitiierten Synchronisation unter Verwendung von Lightweight-Polling“ [*MobiLink - Serverinitiierte Synchronisation*]
- „Praktische Einführung: Konfigurieren der serverinitiierten Synchronisation unter Verwendung von Gateways“ [*MobiLink - Serverinitiierte Synchronisation*]

Praktische Einführungen in UltraLite ++

- „Praktische Einführung: Erstellen der Beispielanwendung CustDB“ [*UltraLite - Datenbankverwaltung*]
- „Praktische Einführung: Eine Android-Anwendung erstellen“ [*UltraLite® – Java-Programmierung*]
- „Praktische Einführung: Eine BlackBerry-Anwendung erstellen“ [*UltraLite® – Java-Programmierung*]

- „Praktische Einführung: Erstellen einer Windows-Anwendung mit der C++-API“ [*UltraLite - C- und C++-Programmierung*]
- „Praktische Einführung: Erstellen einer Windows Mobile-Anwendung unter Verwendung von UltraLite.NET“ [*UltraLite - .NET-Programmierung*]

Praktische Einführungen in SQL Remote

- „Praktische Einführung: SQL Remote-System erstellen“ [*SQL Remote*]
- „Praktische Einführung: Einrichten eines Replikationssystems unter Verwendung des HTTP-Nachrichtensystems“ [*SQL Remote*]
- „Praktische Einführung: Einrichten eines Replikationssystems unter Verwendung des HTTP-Nachrichtensystems mit der konsolidierten Datenbank als Nachrichtenserver“ [*SQL Remote*]
- „Praktische Einführung: Einrichten eines Replikationssystems unter Verwendung des HTTP-Nachrichtensystems und mit der konsolidierten Datenbank als Nachrichtenserver über Relay Server“ [*SQL Remote*]

Ressourcen für die erste Orientierung mit SQL Anywhere 16

Nachstehend finden Sie eine Liste von einführenden Dokumentationen und praktischen Einführungen in Datenbank- und Datenaustauschtechnologien.

Die meisten praktischen Einführungen in dieser Dokumentation verwenden die SQL Anywhere-Beispieldatenbank (demo.db) oder die CustDB-Beispieldatenbank (*custdb.db*).

- **SQL Anywhere-Server** Siehe „[Ressourcen für die erste Orientierung mit SQL Anywhere Server](#)“ auf Seite 60.

In diesem Abschnitt werden die praktischen Einführungen für den Aufbau einer Verbindung zu einer SQL Anywhere Server-Datenbank und für ihre Administration beschrieben.

- **UltraLite** Siehe „[Praktische Einführung: Erstellen der Beispielanwendung CustDB](#)“ [*UltraLite - Datenbankverwaltung*].

In diesem Abschnitt wird die UltraLite-Beispieldatenbank (*custdb.db*) für folgende Demonstrationen benutzt:

- Bei einer entfernten UltraLite-Datenbank anmelden und sie auffüllen
- Clientanwendung anwenden
- UltraLite-Datenbank mit einer konsolidierten Datenbank synchronisieren
- MobiLink-Synchronisationsskripten durchsuchen

- **MobiLink** Siehe [MobiLink - Erste Orientierung](#).

Dieses Handbuch stellt MobiLink vor und bietet Ihnen zahlreiche praktische Einführungen in den Prozess zur Einrichtung eines Synchronisationssystems, das SQL Anywhere-Datenbanken verwendet.

- **SQL Remote** Siehe „[SQL Remote-Systeme](#)“ [*SQL Remote*].

In diesem Abschnitt wird SQL Remote vorgestellt und es wird beschrieben, wie Sie ein einfaches SQL Remote-Replikationssystem mit Sybase Central einrichten.

- **Relay Server** Siehe [Relay Server](#).

In diesem Handbuch wird erklärt, wie Sie den Relay Server einrichten und verwenden.

SAP Sybase PowerDesigner – Physical Data Model

Sie können das Physical Data Model von SAP Sybase PowerDesigner verwenden, um physische Datenmodelle einer Datenbank zu erstellen und danach in SQL Anywhere-Datenbanken zu konvertieren.

Siehe [SAP Sybase PowerDesigner – Physical Data Model](#) auf Seite 47.

Siehe auch

- „SQL Anywhere-Beispieldatenbank“ auf Seite 49
- „MobiLink-Beispiel CustDB“ [*MobiLink - Erste Orientierung*]
- „Praktische Einführung: Erstellen der Beispielanwendung CustDB“ [*UltraLite - Datenbankverwaltung*]

Ressourcen für die erste Orientierung mit SQL Anywhere Server

SQL Anywhere bietet drei Methoden für die Systemadministration: Sybase Central, Interactive SQL und Befehlszeilen-Dienstprogramme. In den meisten Fällen richtet sich die Wahl der besten Methode ganz einfach nach Ihrer persönlichen Vorliebe. Die meisten Aufgaben werden durch alle drei Methoden unterstützt.

Die in diesem Abschnitt angeführten praktischen Einführungen und Hinweise zum Schnelleinstieg benutzen alle drei Methoden, um SQL Anywhere-Datenbanken zu erstellen und zu verwalten.

Einen Überblick über Sybase Central, Interactive SQL und alle anderen in SQL Anywhere enthaltenen Tools finden Sie unter „*Design- und Administrationstools*“ auf Seite 45.

Datenbankserver starten

- „Praktische Einführung: Verbindung mit der Beispieldatenbank herstellen“ [*SQL Anywhere Server - Datenbankadministration*]

Diese praktische Einführung beschreibt, wie Sie Folgendes durchführen:

- Einen Datenbankserver starten
 - Mit einem Datenbankserver verbinden
 - Das Meldungsfenster des Datenbankservers anzeigen
 - Den Datenbankserver stoppen
- **Zusätzliche Informationen** Eine vollständige Liste der Serververbindungsoptionen finden Sie unter „*Syntax des SQL Anywhere-Datenbankservers*“ [*SQL Anywhere Server - Datenbankadministration*]. Als Alternative können Sie auch den folgenden Befehl ausführen, um eine Liste aller Serververbindungsoptionen in einem Fenster mit einer Eingabeaufforderung anzuzeigen:

```
dbeng16 -?
```

- **Weitere Hinweise** Weitere Beispiele für die Verbindung zur SQL Anywhere-Beispieldatenbank, zu einer eingebetteten Datenbank oder zu einer Datenbank über ein Netzwerk finden Sie unter „*Datenbankverbindungen*“ [*SQL Anywhere Server - Datenbankadministration*].

Datenbank erstellen

- „Praktische Einführung: SQL Anywhere-Datenbank erstellen“ [*SQL Anywhere Server - Datenbankadministration*]

In dieser praktischen Einführung wird gezeigt, wie Sie mit Sybase Central folgende Aufgaben durchführen können:

- Eine Datenbankdatei erstellen
- Eine Verbindung mit der Datenbank herstellen
- Eine Tabelle erstellen
- Beziehungen zwischen Tabellen herstellen
- **Zusätzliche Informationen** Weitere Hinweise zum Erstellen von Datenbankeigenschaften finden Sie unter „Datenbankerstellung“ [[SQL Anywhere Server - Datenbankadministration](#)].

Datenbanken verwalten

- „Sybase Central“ [[SQL Anywhere Server - Datenbankadministration](#)]

Dieser Abschnitt enthält eine Einführung zu Sybase Central und Informationen über die folgenden Themen:

- „Verbinden mit der Beispieldatenbank (Sybase Central)“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Verbindungsprofile erstellen“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Nach Datenbankobjekten suchen“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Entity-Relationship-Diagramme anzeigen“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Zustand und Statistik von Datenbanken“ [[SQL Anywhere Server - Datenbankadministration](#)]

SQL-Anweisungen verwenden

- „Interactive SQL“ [[SQL Anywhere Server - Datenbankadministration](#)]

Dieser Abschnitt enthält eine Einführung zu Interactive SQL und Informationen über die folgenden Themen:

- „SQL-Anweisungen ausführen (Interactive SQL)“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Abbrechen von Anweisungen in Interactive SQL“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Suchen nach Tabellen, Spalten und Prozeduren in Interactive SQL“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Interactive SQL-Verlauf“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Anweisungen in Interactive SQL protokollieren“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „SQL-Skriptdateien“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)]

Häufig gestellte Fragen - SQL Anywhere

Wie lauten die Standardwerte für Benutzer-ID und Kennwort bei *demo.db* und neu erstellten Datenbanken?

Die standardmäßige Benutzer-ID für *demo.db* und neu erstellte Datenbanken ist **DBA** und das Standardkennwort **sql**. Die meisten Beispiele in der SQL Anywhere-Dokumentation verwenden DBA/sql. Die Benutzer-ID wird ohne Berücksichtigung von Groß- und Kleinschreibung verwendet und das Kennwort mit Berücksichtigung von Groß- und Kleinschreibung. Da es sich hierbei jedoch um eine bekannte Benutzer-ID und ein bekanntes Kennwort handelt, sollten Sie das Kennwort vor dem Deployment Ihrer Datenbank ändern oder einen anderen Benutzer als DBA-Benutzer erstellen und verwenden.

Informationen zu fehlgeschlagenen Login-Versuchen für DBA-Benutzer finden Sie unter „[Automatisches Entsperren von Benutzerkonten](#)“ [*SQL Anywhere Server - Datenbankadministration*].

Wie kann ich ein DBA-Kennwort wiederherstellen?

Ein verloren gegangenes oder vergessenes DBA-Kennwort kann nicht wiederhergestellt werden. Wenn Sie Unterstützung benötigen, wenden Sie sich an den technischen Support (<http://www.sybase.com/support>).

Wie kann ich die Verbindung mit einer Datenbank herstellen?

Die für die Verbindung mit einer Datenbank erforderlichen Verbindungsparameter variieren abhängig vom relativen Speicherort der Anwendung und des Datenbankservers. In den folgenden Verbindungsszenarien wird vorausgesetzt, dass keine Verschlüsselung der Kommunikation erfolgt und dass nur eine Datenbank auf einem Datenbankserver läuft.

In den folgenden Beispielen verbinden Sie sich unter Verwendung von Interactive SQL mit der SQL Anywhere-Beispieldatenbank, %SQLANYAMP16%\demo.db.

- **Beispiel 1: Der Datenbankserver läuft nicht und die Datenbankdatei ist auf demselben Computer installiert wie die Anwendung** Sie müssen Folgendes richtig angeben: Benutzer-ID, Kennwort, Verbindungsparameter und Parameter für die Datenbankdatei. Außerdem müssen Sie möglicherweise den Namen des Datenbankservers eingeben. Um eine Verbindung zur Datenbank herzustellen, führen Sie einen Befehl wie im folgenden Beispiel aus:

```
dbisql -c "UID=DBA;PWD=sql;Server=demo;DBF=%SQLANYAMP16%\demo.db"
```

- **Beispiel 2: Die Anwendung und der Datenbankserver laufen auf demselben Computer** Sie müssen die richtige Benutzer-ID und das richtige Kennwort angeben. Überdies wird empfohlen, dass Sie den Namen des Datenbankservers angeben, damit Sie nicht eine Verbindung mit dem falschen Datenbankserver herstellen. Um eine Verbindung zur Datenbank herzustellen, führen Sie einen Befehl wie im folgenden Beispiel aus:

```
dbisql -c "UID=DBA;PWD=sql;Server=demo"
```

- **Beispiel 3: Die Anwendung und der Datenbankserver laufen auf verschiedenen Computern** Sie müssen die richtige Benutzer-ID und das richtige Kennwort angeben sowie den

Namen des Datenbankservers und die HOST-Verbindungsparameter. Um eine Verbindung zur Datenbank herzustellen, führen Sie einen Befehl wie im folgenden Beispiel aus:

```
dbisql -c "UID=DBA;PWD=sql;Server=demo;HOST=myhost"
```

Wie kann ich unter Unix eine Verbindung mit einer Datenbank herstellen?

Siehe „Verbinden mit der Beispieldatenbank unter Unix“ [[SQL Anywhere Server - Datenbankadministration](#)].

Wie kann ich unter Mac OS X eine Verbindung mit einer Datenbank herstellen?

Siehe „Verbindung zur Beispieldatenbank unter Mac OS X herstellen“ [[SQL Anywhere Server - Datenbankadministration](#)].

Wie kann ich ermitteln, welche Version von SQL Anywhere verwendet wurde, um eine Datenbank zu erstellen?

Datenbanken, die mit SQL Anywhere 9.0.1 oder höher erstellt wurden, enthalten die SYSHISTORY-Systemansicht. Die Zeilen in dieser Ansicht, die INIT- und UPDATE-Vorgänge reflektieren, geben Informationen darüber, wann und mit welcher Version die Datenbank erstellt und aktualisiert wurde. Siehe „SYSHISTORY-Systemansicht“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)].

Hinweise dazu, wie Sie die Version ermitteln, ohne die Datenbank zu starten, finden Sie unter [DBCreatedVersion-Methode \[Datenbanktools\]](#) [[SQL Anywhere Server - Programmierung](#)].

Wie kann ich die Namen der Datenbanken ermitteln, die auf einem Server laufen?

Führen Sie die folgende Anweisung aus, um die Namen der auf einem Server laufenden Datenbanken zu ermitteln:

```
SELECT DB_NAME(Number) FROM sa_db_list();
```

Führen Sie die folgende Anweisung aus, um die Anzahl von auf einem Server laufenden Datenbanken zurückzugeben:

```
SELECT * FROM sa_db_list();
```

Siehe „sa_db_list-Systemprozedur“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)] und „DB_NAME-Funktion [System]“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)].

Kann ich zwei verschiedene Versionen von SQL Anywhere auf demselben Computer installieren und ausführen?

Ja, Sie können mehrere Hauptversionen von SQL Anywhere auf demselben Computer installieren. Beispiel: SQL Anywhere 9.0.2, 10.0.1, 11.0.1 und 12.0.1 können alle installiert und unabhängig voneinander ausgeführt werden.

Sie müssen jedoch beim Start einer SQL Anywhere-Programmdatei, die in mehreren Versionen denselben Namen hat (z.B. dbisql oder dbinit), darauf achten, dass Sie die richtige Version der Anwendung starten. Sie können entweder den vollständigen absoluten Pfad mit Umgebungsvariablen wie SQLANY16 eingeben oder die gewünschte Version von SQL Anywhere in Ihrem Pfad zuerst angeben.

Siehe „Sicherstellen, dass Sie die richtige Version der Dienstprogramme ausführen, wenn mehrere Versionen installiert sind“ [[SQL Anywhere 16 - Änderungen und Upgrades](#)].

Kann ich zwei Instanzen derselben Hauptversion von SQL Anywhere auf demselben Computer installieren und ausführen?

Bei SQL Anywhere 10 und früher für Windows unter Verwendung des SQL Anywhere-Installationsprogramms Ja, Sie können mehrere Kopien derselben Version von SQL Anywhere auf demselben Computer installieren. Das SQL Anywhere-Installationsprogramm registriert jedoch einige Treiber und Komponenten in der Windows-Registrierung, von der es nur eine Kopie gibt. Aus diesem Grund werden die ODBC- und OLE DB-Treiber aus der neuesten Installation verwendet. Ebenso gibt es nur eine Gruppe von Verknüpfungen im Menü **Start**. Sie zeigen auf die zuletzt installierte Kopie von SQL Anywhere.

Bei SQL Anywhere 11 und höher für Windows unter Verwendung des SQL Anywhere-Installationsprogramms Nein, Sie können nur eine Kopie von SQL Anywhere 11 auf einem Computer installieren.

Bei SQL Anywhere 12 und früher für Linux oder Unix unter Verwendung des SQL Anywhere-Installationsprogramms Ja, Sie können mehrere Versionen an verschiedenen Orten installieren. Wenn Sie jedoch unter Linux Anwendungsmenüoptionen installieren, kann jeder Benutzer nur einen Satz von Anwendungsmenüoptionen haben. Sie zeigen auf die zuletzt installierte Kopie von SQL Anywhere.

Für eingebettete Datenbank Anwendungen, deren Deployment durchgeführt wurde Ja, bereitgestellte eingebettete Datenbank Anwendungen, die SQL Anywhere enthalten, können mit anderen SQL Anywhere-Installationen auf demselben Computer bereitgestellt werden.

Unter Windows-Betriebssystemen sollten die ODBC- und OLE DB-Treibernamen in der Registrierung den Namen der Anwendung enthalten, in der sie eingebettet sind. Zum Beispiel sollte der ODBC-Treibername **SQL Anywhere 16** umbenannt werden in *application-name* **SQL Anywhere 16**. Siehe „Deployment von Datenbanken und Anwendungen“ [[SQL Anywhere Server - Programmierung](#)] und „ODBC-Treiberkonfiguration“ [[SQL Anywhere Server - Programmierung](#)].

Ist es besser, einen Datenbankserver für jede Datenbank zu haben, oder sollte ich mehrere Datenbanken auf einem einzelnen Datenbankserver ausführen?

Sie sollten mehrere Datenbanken auf einem einzelnen Datenbankserver ausführen, da diese Konfiguration die Verwendung der Computerressourcen optimiert.

Wenn mehrere Datenbankserver auf demselben Computer laufen, kommt es möglicherweise zu einer Konkurrenz um Ressourcen und diese Konfiguration kann (mit dynamischer Anpassung der Cachegröße) zu einer verschlechterten oder nicht vorhersehbaren Performance führen. Eine verschlechterte oder nicht vorhersehbare Performance ist möglicherweise akzeptabel, wenn Sie einen Datenbankserver ohne Auswirkungen auf andere Datenbankserver für die Wartung stoppen müssen oder wenn Sie Fehler auf einen einzelnen Datenbankserver begrenzen müssen.

Sie sollten überprüfen, ob Sie über die richtige Lizenz für die gewählte Installationsoption verfügen. Weitere Hinweise finden Sie unter „[Mehrere Datenbanken auf einem einzelnen Datenbankserver ausführen](#)“ auf Seite 29.

Warum nimmt die Größe meiner Datenbank zu bzw. nicht wie erwartet ab?

Siehe „Unerwartete Änderungen der Datenbankgröße“ [[SQL Anywhere Server - Datenbankadministration](#)].

Wie erstelle ich einen effektiven Sicherungs- und Wiederherstellungsplan?

Weitere Hinweise finden Sie in den folgenden Dokumenten:

- „Entwerfen von Sicherungs- und Wiederherstellungsplänen“ [[SQL Anywhere Server - Datenbankadministration](#)]
- Weitere Hinweise finden Sie im Dokument „What Backup, Recovery, and Disaster Recovery Mean to Your SQL Anywhere Databases“ (<http://www.sybase.com/detail?id=47877>)

Was kann ich tun, wenn ein Assertierungsfehler auftritt?

Weitere Hinweise finden Sie im Dokument zum Thema Assertierungen ("I've Got An Assertion! What Should I do?" unter (<http://www.sybase.com/detail?id=1010805>).

Wie kann ich einen Fehler melden?

Um einen SQL Anywhere-Fehler niedriger Priorität zu melden, besuchen Sie <http://case-express.sybase.com>. Mit Case-Express gemeldete Fehler erhalten eine niedrigere Priorität als Fälle, die über den technischen Support geöffnet wurden.

Bei Problemen mit hoher Priorität sollten Sie einen sogenannten Case beim technischen Support öffnen (<http://www.sybase.com/support>).

Wie kann ich die Performance von Anwendung oder Datenbankserver verbessern?

Weitere Hinweise finden Sie in den folgenden Dokumenten:

- „Tools für Performanceüberwachung und Diagnose“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)]
- „Tipps zum Verbessern der Performance“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)]
- Diagnosing Application Performance Issues with SQL Anywhere (<http://www.sybase.com/detail?id=1060302>)
- Capacity Planning with SQL Anywhere (<http://www.sybase.com/detail?id=1056535>)

Wie kann ich meine SQL Anywhere-Software aktualisieren?

Siehe „SQL Anywhere-Server-Ugrades“ [[SQL Anywhere 16 - Änderungen und Upgrades](#)].

Warum läuft meine Anwendung nach einer Aktualisierung langsamer?

SQL Anywhere erreicht die beste Performance, wenn die Datenbank mit derselben Hauptversion erstellt wird wie der Datenbankserver. Erstellen Sie die Datenbank neu, wenn Sie auf Performanceprobleme gestoßen sind und die Datenbank nicht mehr in früheren Versionen von SQL Anywhere ausführen müssen. Siehe „Neuaufbau von Datenbanken“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)] und „Tools für Performanceüberwachung und Diagnose“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)].

Warum funktioniert meine Anwendung nach einer Aktualisierung nicht?

SQL Anywhere-Entwickler sind bemüht sicherzustellen, dass die Anwendungen auch nach einer Aktualisierung funktionieren. Ihre Anwendung kann jedoch von Verhaltensänderungen und der Entfernung von zuvor unterstützten Funktionen betroffen sein.

Hinweise dazu, ob Verhaltensänderungen an Ihrer Version von SQL Anywhere vorgenommen wurden sowie ob Funktionen entfernt wurden oder nicht mehr empfohlen werden, wählen Sie den für Ihre Version von SQL Anywhere relevanten Abschnitt. Siehe [SQL Anywhere 16 - Änderungen und Upgrades](#).

Wenn Ihre JDBC-Anwendungen nach der Anwendung eines Supportpakets nicht funktionieren, liegt dies wahrscheinlich daran, dass entweder Interactive SQL, Sybase Central, der Schnelllader oder andere JDBC-Anwendungen ausgeführt wurden, als das Supportpaket übernommen wurde. Sie können das Problem lösen, indem Sie alle JDBC-basierten Anwendungen herunterfahren und das Supportpaket erneut anwenden. Siehe „Fehlerbehandlung: Sicherstellen, dass während des Übernehmens eines Supportpakets keine JDBC-Anwendungen laufen“ [[SQL Anywhere 16 - Änderungen und Upgrades](#)].

Welche Betriebssysteme werden von den verschiedenen Versionen von SQL Anywhere unterstützt?

Weitere Hinweise finden Sie im Dokument "SQL Anywhere Supported Platforms and Engineering Support Status" (<http://www.sybase.com/detail?id=1002288>).

Welches sind die Lizenzierungsanforderungen für eine typische Installation von SQL Anywhere?

Eine Beschreibung der verschiedenen Lizenzierungsoptionen für SQL Anywhere 10 und höher sowie Beispiele, die Ihnen zeigen, wie Sie Ihre Lizenz anwenden, finden Sie im Dokument "SQL Anywhere Licensing" (<http://www.sybase.com/detail?id=1056242>).

Wie lautet die größte Datenbankgröße, die von SQL Anywhere unterstützt wird?

Die Größe der Datenbank, die von SQL Anywhere unterstützt werden kann, ist abhängig von Speicher, CPU und Festplattenkapazität des Computers, auf dem SQL Anywhere installiert ist. Siehe „[SQL Anywhere-Einschränkungen von Größe und Anzahl](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

Eine Reihe unserer Kunden diskutiert die Implementierung von großen Datenbanken im SQL Anywhere-Forum. Siehe Forum-Thread <http://sqlanywhere-forum.sybase.com/questions/92/whats-the-size-of-your-biggest-database>.

Wie kann ich eine Migration von einem anderen Datenbankprodukt auf SQL Anywhere durchführen?

Um auf SQL Anywhere zu migrieren, müssen Sie Ihre Daten in eine SQL Anywhere-Datenbank importieren. Siehe „[Datenbankmigration nach SQL Anywhere](#)“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)].

Es gibt Unterschiede zwischen den Produkten, z.B. verschiedene Dialekte von SQL, daher muss die Anwendung möglicherweise geändert werden.

Welche Wettbewerbsvorteile bietet SQL Anywhere?

Weitere Hinweise finden Sie im Dokument "Choosing SQL Anywhere for ISV Applications" (http://www.sybase.com/files/Data_Sheets/Sybase_iAnywhereSQLAnywhere_isvapps_ds.pdf).

Ich muss Performanceprobleme bei einer bestimmten Abfrage diagnostizieren. Wie erstelle ich einen grafischen Plan mit Statistiken?

Siehe „Grafische Pläne mit detaillierten und Knotenstatistiken erstellen“ [*SQL Anywhere Server - Datenbankadministration*].

Kann ich die OUTPUT-Anweisung in einer gespeicherten Prozedur verwenden?

Nein. Die OUTPUT-Anweisung kann nur in Interactive SQL ausgeführt werden und kann nicht in einer gespeicherten Prozedur verwendet werden. Verwenden Sie die UNLOAD-Anweisung in der gespeicherten Prozedur, um die durch eine SQL-Anweisung generierte Ergebnismenge in einer Textdatei zu speichern. Wenn Sie die UNLOAD-Anweisung in der gespeicherten Prozedur verwenden, werden Informationen vom Datenbankserver entladen und nicht von dem Clientcomputer, von dem aus die Anweisung ausgeführt wurde. Siehe „UNLOAD-Anweisung“ [*SQL Anywhere Server - SQL-Referenzhandbuch*].

Gibt es eine praktische Einführung oder eine schrittweise Beschreibung der Funktion, die mich interessiert?

Siehe „Liste der praktischen Einführungen in SQL Anywhere“ auf Seite 55.

Wo kann ich weitere Informationen über die Cloud finden?

Weitere Hinweise finden Sie in der On-Demand-Edition der Dokumentation zu SQL Anywhere unter <http://dcx.sybase.com/cloud100>.

Wo kann ich weitere Informationen finden?

Siehe:

- SQL Anywhere-Dokumentation: dcx.sybase.com/dcx_home.html
- SQL Anywhere v10 / v9.0.2 Frequently Asked Questions (<http://www.sybase.com/detail?id=1053692>)
- SQL Anywhere 11.0.1 FAQs (<http://www.sybase.com/detail?id=1062382>)
- SQL Anywhere Studio/Adaptive Server Anywhere Patches and Upgrades FAQ (<http://www.sybase.com/detail?id=1019567>)
- SQL Anywhere-Blogs (<http://iablog.sybase.com/sqlanyblogs.php>)
- SQL Anywhere-Forum <http://sqlanywhere-forum.sybase.com>

Index

Symbole

* (Stern)
(*Siehe auch* Sternchen)
-push_notifications, Option (*Siehe* -push, Option)
.NET
(*Siehe auch* ADO.NET)
.NET-Synchronisationslogik
unterstützte Plattformen,7
32-Bit
unterstützte Versionen,7
64-Bit
unterstützte Versionen,7
@environment-variable-Option (*Siehe* @data-Option)
@filename-Option (*Siehe* @data-Option)

A

Abfrage, Bypass (*Siehe* Bypass-Optimierung)
Abfrageinterne Parallelität
(*Siehe auch* abfrageübergreifende Parallelität)
Abfrageoptimierer
(*Siehe auch* Optimierer)
Abfrageoptimierung
(*Siehe auch* Optimierer)
Abfrageübergreifende Parallelität
(*Siehe auch* abfrageinterne Parallelität)
Abruf
(*Siehe auch* aktualisieren)
Abschnitt 508
Modul zur Aktivierung von Eingabehilfen,16
Unterstützung für Eingabehilfen,8
VPAT,8
Accent grave (*Siehe* Back Quotes)
Access (*Siehe* Microsoft Access)
ActiveSync
unterstützte Plattformen,7
Adaptive Server Anywhere (*Siehe* SQL Anywhere)
Administrationsdienstprogramme
(*Siehe auch* Datenbankadministration,
Dienstprogramme)
Administrationstools
unterstützte Plattformen,7
Aktualisieren
langsame Anwendung beheben,66
Anforderungen an das Betriebssystem

unterstützte Plattformen,7
Anforderungsebene, Log (*Siehe* Anforderungslog)
Anforderungsebene, Protokollierung (*Siehe*
Anforderungsprotokollierung)
Anführungszeichen
(*Siehe auch* Anführungszeichen)
ANSI-Konformität
(*Siehe auch* SQL-Standards)
Anti-SQL-Log (*Siehe* Rollback-Logs)
Anwendungsprogrammierschnittstellen
(*Siehe auch* APIs)
Anzahl der auf einem Server laufenden Datenbanken
Info,64
APIs
Info,31
Apple (*Siehe* Mac OS X)
Archive
(*Siehe auch* Sicherungen)
ARM V4T-Modus
unterstützte Plattformen,7
ARM-Chip
unterstützte Plattformen,7
ARM-Prozessor
unterstützte Plattformen,7
ASE
(*Siehe auch* Adaptive Server Enterprise)
Assertions
auf Fehler reagieren,66
Attribute
Relationen,19
Ausgabe
(*Siehe auch* Daten exportieren)
Auswahl einer Synchronisationstechnologie
Info,42
Auswahlliste (*Siehe* SELECT-Liste)

B

Back Ticks (*Siehe* Back Quotes)
Befehlsabfolge, Kommunikationsprotokoll
Diagramm,31
Info,34
Begrenzen
(*Siehe auch* Einschränkungen)
Beispieldatenbank
custdb.db für UltraLite- und MobiLink-
Anwendungen,53
demo.db für SQL Anywhere, Beispiele,49

- demo.db neu erstellen,52
- demo.db wiederherstellen,52
- Rollen,50
- Schema von custdb.db,53
- Benutzer-IDs
 - Standardwert für demo.db,63
- Berechtigungen (*Siehe Privilegien*)
- Berechtigungen für Datenbanken (*Siehe Privilegien*)
- Betriebsbereiter Server (*Siehe Partnerserver*)
- Betriebssysteme
 - unterstützte Plattformen,7
 - von SQL Anywhere unterstützt,67
 - Windows,v
- Binäre Konstanten (*Siehe binäre Literale*)
- BINARY VARYING-Datentyp (*Siehe*
VARBINARY-Datentyp)
- BIT VARYING-Datentyp (*Siehe* VARBIT-Datentyp)
- BlackBerry
 - SQL Anywhere-Datenbanken nicht unterstützt,26
 - UltraLite-Datenbankunterstützung,26
 - unterstützte Plattformen,7
- Bloom-Filter (*Siehe Hash-Filter*)
- BOM (Bytereihenfolge-Marke) (*Siehe* BYTE ORDER
MARK)
- Buch, materialisierte Ansichten (*Siehe* Buch,
Ansichten)
- Bugs
 - Anregungen mitteilen,vii
- Bypass-Abfragen
 - (*Siehe auch* einfache Abfragen)

C

- C-API (*Siehe* SQL Anywhere-C-API)
- CAC-Authentifizierung
 - getrennt lizenzierbare Komponenten,3
- Caldera
 - unterstützte Versionen,7
- CE (*Siehe* Windows Mobile)
- Certicom
 - Verschlüsselungssoftware bestellen,3
- CHAR VARYING-Datentyp (*Siehe* VARCHAR-
Datentyp)
- CHARACTER VARYING-Datentyp (*Siehe*
VARCHAR-Datentyp)
- Client/Server
 - Anwendungen und Mehrbenutzer-Datenbanken,27
- Cloud

- Zugriff auf die Dokumentation,68
- CONTAINS-Klausel (*Siehe* CONTAINS-
Suchbedingung)
- CPUs
 - lizenzieren,2
- CustDB
 - Info,53
- CustDB-Anwendung
 - Info,53

D

- Data Warehouses
 - ETL-Funktionen,30
- Daten ausgeben (*Siehe* Daten exportieren)
- Daten einfügen
 - (*Siehe auch* Daten importieren)
- Daten eingeben (*Siehe* Daten importieren)
- Daten hinzufügen
 - (*Siehe auch* Daten einfügen)
 - (*Siehe auch* Daten importieren)
- Daten importieren
 - (*Siehe auch* Daten einfügen)
- Daten übertragen
 - (*Siehe auch* Daten exportieren)
- Daten verschieben
 - (*Siehe auch* Daten einfügen)
 - (*Siehe auch* Daten exportieren)
 - (*Siehe auch* Daten importieren)
- Datenaustausch
 - Info,35
- Datenbank entladen, Dienstprogramm (dbunload)
(*Siehe* Entlade-Dienstprogramm (dbunload))
- Datenbank-Dienstprogramme
 - (*Siehe auch* Dienstprogramme)
- Datenbankadministrator (*Siehe* DBA)
- Datenbankaufräumvorgang (*Siehe*
Datenbankaufräumvorgang)
- Datenbankdateien
 - Einführung,24
- Datenbanken
 - Anzahl der auf einem Datenbankserver
laufenden,64
 - Clientanwendung,18
 - Dateien,24
 - Info,17
 - Komponenten,17
 - Objekte,20

-
- relational,18
 - replizieren,35
 - Server,17
 - Sprachschnittstelle,18
 - synchronisieren,35
 - zwischen SQL Anywhere und UltraLite wählen,26
 - Datenbanken replizieren
 - Info,35
 - Datenbanken synchronisieren
 - Info,35
 - Datenbankgröße
 - maximale von SQL Anywhere unterstützte Größe,67
 - Datenbankobjekte
 - Info,20
 - Datenbankreplikation
 - Info,35
 - Datenbankserver
 - (*Siehe auch* Datenbankserver)
 - Info,18
 - interne Vorgänge,22
 - Unterschiede zwischen Personal und Netzwerkservern,18
 - Datenbankserver, Eigenschaften (*Siehe* Servereigenschaften)
 - Datenbankspiegelung
 - getrennt lizenzierbare Komponenten,5
 - Datenbanksynchronisation
 - Info,35
 - Datenbanktabellen
 - Info,19
 - Datenbankverschlüsselung
 - Verschlüsselungssoftware bestellen,3
 - Datendefinitionsanweisungen (*Siehe* DDL)
 - Datendefinitionssprache (*Siehe* DDL)
 - Datenkonsistenz
 - Herausforderungen an die Synchronisationstechnologien,38
 - Datenmanipulationssprache (*Siehe* DML)
 - Datenverfügbarkeit
 - Vorteile der Datensynchronisation,37
 - DBA
 - Kennwort wiederherstellen,63
 - DBI-Modul (*Siehe* DBD::SQLAnywhere)
 - dbisql-Dienstprogramm
 - (*Siehe auch* Interactive SQL)
 - (*Siehe auch* Interactive SQL-Dienstprogramm (dbisql))
 - dbremote-Dienstprogramm
 - unterstützte Plattformen,7
 - DEC-Datentyp (*Siehe* DECIMAL-Datentyp)
 - Demo-Datenbank
 - Info,49
 - demo.db
 - FAQ,63
 - Info,49
 - neu erstellen,52
 - wiederherstellen,52
 - Deployment-Edition
 - unterstützte Plattformen,7
 - Deployment-Versionen
 - unterstützte Plattformen,7
 - Deployment.
 - unterstützte Plattformen,7
 - Developer Community
 - SQL Anywhere-Community,vii
 - SQL Anywhere-Forum,vii
 - Dienst überwachen (*Siehe* SQL Anywhere-Monitor)
 - Dienstprogramme
 - (*Siehe auch* Datenbank-Dienstprogramme)
 - Dokumentation
 - für frühere Versionen von SQL Anywhere,viii
 - Konventionen,v
 - SQL Anywhere,v
 - Doppelte Kennwortkontrolle
 - siehe Doppelte Kontrolle für Kennwörter,v
 - Dreischichtige Datenverarbeitung
 - Einführung,28
- ## E
- Editionen
 - getrennt lizenzierbare Komponenten bündeln,2
 - Quellen für weitere Informationen,2
 - Einfache Abfragen
 - (*Siehe auch* Bypass-Abfragen)
 - Einführung in SQL Anywhere
 - Design- und Administrationstools,45
 - Erste Orientierung,59
 - Nachrichtenverarbeitungstechnologien,35
 - Synchronisationstechnologien,35
 - Eingabeaufforderungen
 - Anführungszeichen,vi
 - geschweifte Klammern,vi
 - Klammern,vi
 - Konventionen,vi

- Semikola,vi
- Eingabehilfen
 - Java Access Bridge,16
 - Unterstützung für Abschnitt 508,8
- Eingebettete Datenbanken
 - Beispielanwendungen,1
 - Definition,26
- Ende-zu-Ende (*Siehe* Ende-zu-Ende-Verschlüsselung)
- ENG, Verbindungsparameter (*Siehe* Verbindungsparameter ServerName)
- EngineName, Verbindungsparameter (*Siehe* Verbindungsparameter ServerName)
- Engines
 - (*Siehe auch* Datenbankserver)
 - (*Siehe auch* Server)
- Entfernen
 - (*Siehe auch* löschen)
- Entitäten
 - Relationen,19
- Entscheidung zwischen SQL Anywhere- und UltraLite-Datenbanken
 - Info,26
- Entsprechung mit SQL-Standards
 - (*Siehe auch* SQL-Standards)
- Entwicklungsplattformen
 - unterstützte Plattformen,7
- Erste Orientierung
 - SQL Anywhere 16,59
- Erste Orientierung in SQL Anywhere
 - Datenbanktechnologien,17
- Erstellen, Datenbankdokumentation (*Siehe* Datenbank dokumentieren)
- Erstellung von Diensten, Dienstprogramm (*Siehe* Dienstprogramm für Dienste (dbsvc))
- ESRI-Formdateien (*Siehe* Formdateien)
- ETL
 - Info,30
 - unterstützte Funktionen,30
- Extensible Markup Language (*Siehe* XML)
- Extrahieren, Transformieren und Laden
 - ETL,30

F

- FAQ
 - Info,63
 - SQL Anywhere,63

- Federal Information Processing Standard Publication-Konformität
 - (*Siehe auch* SQL-Standards)
- Fehler
 - Assertions,66
 - melden,66
- Fehlerbehandlung
 - FAQ,63
 - SQL Anywhere-Community,vii
 - SQL Anywhere-Forum,vii
- Ferndatenzugriff
 - unterstützte Plattformen,7
- FIPS
 - Verschlüsselungssoftware bestellen,3
- FIPS, Option
 - getrennt lizenzierbare Komponente,3
- FIPS, Protokolloption
 - getrennt lizenzierbare Komponente,3
- FIPS-Konformität
 - (*Siehe auch* SQL-Standards)
- Flagger (*Siehe* SQL Flagger)
- Freie Suche (*Siehe* Volltextsuche)
- Fremdschlüssel
 - Definition,20
- Frontend-Umgebungen
 - Info,1
- Frühere Versionen
 - frühere Dokumentation für SQL Anywhere finden,viii
- FTP, Nachrichtentyp
 - von SQL Remote unterstützte Windows-Plattformen,7
- Funktionen
 - (*Siehe auch* unterstützte Plattformen)

G

- Gemeinsamer Dateizugriff, Nachrichtentyp
 - von SQL Remote unterstützte Windows-Plattformen,7
- Gespeicherte Prozeduren
 - Ergebnismengen speichern,68
- Getrennt lizenzierbare Komponenten
 - Info,3
- Grafische Pläne
 - FAQ,68
- GRANT-Option (*Siehe* WITH GRANT OPTION-Klausel)

Gruppen (*Siehe* benutzererweiterte Rollen)

H

Handheld-PC

unterstützte Plattformen,7

Häufig gestellte Fragen (*Siehe* FAQ)

Herausforderungen an die

Synchronisationstechnologien

Info,38

Herstellen einer Verbindung

SQL Anywhere-Beispiele,63

Hewlett Packard HP-UX

(*Siehe auch* HP-UX)

Hexadezimale Konstanten

(*Siehe auch* binäre Literale)

Hierarchische Datenstruktur

hierarchische Datenbankkonfigurationen,39

Hilfe

technische Unterstützung,vii

Hilfe anfordern

technische Unterstützung,vii

Hohe Verfügbarkeit

getrennt lizenzierbare Komponenten,5

Hostplattformen

unterstützte Plattformen,7

HP-UX

unterstützte Plattformen,7

I

IBM AIX

(*Siehe auch* AIX)

unterstützte Plattformen,7

Immer verfügbar

Vorteile der Datensynchronisation,37

In-Memory-Modus

getrennt lizenzierbare Komponenten,5

Initialisierungsdateien (*Siehe* INI-Dateien)

Installation

unterstützte Plattformen,7

Installationsverzeichnis (*Siehe*

Installationsverzeichnis)

INT-Datentyp (*Siehe* INTEGER-Datentyp)

Interactive SQL

(*Siehe auch* dbisql, Dienstprogramm)

(*Siehe auch* Interactive SQL-Dienstprogramm
(dbisql))

unterstützte Plattformen,7

Interactive SQL-Dienstprogramm

(*Siehe auch* dbisql, Dienstprogramm)

Interactive SQL-Dienstprogramm (dbisql)

(*Siehe auch* Interactive SQL)

Interne Vorgänge

Datenbankdateien,24

Datenbankserver,22

iPhone

unterstützte Plattformen,7

ISO-Konformität

(*Siehe auch* SQL-Standards)

J

J2EE (*Siehe* Java EE)

J2ME (*Siehe* Java ME)

J2SE (*Siehe* Java SE)

Java Access Bridge

Modul zur Aktivierung von Eingabehilfen,16

Java ME

unterstützte Versionen,7

Java-Administrationstools (*Siehe*

Administrationstools)

Java-Synchronisationslogik

unterstützte Plattformen,7

Jederzeit verfügbare Rechenleistung

SQL Anywhere-Merkmale,6

K

Kauf

getrennt lizenzierbare Komponenten,3

Kennwörter

DBA wiederherstellen,63

Standardwert für demo.db,63

Kerberos-Logins

(*Siehe auch* Kerberos)

Kernel

unterstützte Versionen,7

Knoten kopieren

(*Siehe auch* Scale-Out mit Schreibschutz)

Kollationssequenzen

(*Siehe auch* Kollationen)

Kommunikationsparameter (*Siehe*

Verbindungsparameter)

Kommunikationsprotokolle

SQL Anywhere,34

Konfigurationsdateien

(*Siehe auch* @data-Option)

Konsole (*Siehe* Meldungsfenster des Datenbankservers) (*Siehe* SQL Anywhere-Konsolendienstprogramm) (*Siehe* SQL Anywhere-Monitorkonsole)
Konsolidierte Datenbanken
 Info,38
 unterstützte RDBMS-Systeme,7
Konstante Binärdaten (*Siehe* binäre Literale)
Konstante Zeichenfolgen (*Siehe* Zeichenfolgenliterale)
Konstanten (*Siehe* binäre Literale) (*Siehe* Zeichenfolgenliterale)
Kontrollmechanismus im Mehrbenutzerbetrieb mehrerer Versionen (*Siehe* Snapshot Isolation)
Konventionen
 Betriebssysteme,v
 Dateinamen in der Dokumentation,v
 Dokumentation,v
 Eingabeaufforderungen,vi
Kopieknoten
 getrennt lizenzierbare Komponenten,5

L

Linux
 unterstützte Plattformen,7
Literalzeichenfolgen (*Siehe* Zeichenfolgenliterale)
Lizenz-Dienstprogramm (*Siehe* Dienstprogramm zur Serverlizenzierung (dblic))
Lizenzen
 getrennt lizenzierbare Komponenten,3
Lizenzieren
 CPUs,2
 Prozessoren,2
Lizenzierung
 SQL Anywhere, Anforderungen.,67
Log auf Anforderungsebene (*Siehe* Anforderungslog)
Lokale Server (*Siehe* Personal Server)
LONG BIT VARYING-Datentyp (*Siehe* LONG VARBIT-Datentyp)
Löschen von Datenbanken
 (*Siehe auch* Datenbanken löschen)

M

Mac OS X
 UltraLite nicht unterstützt,26
 unterstützte Plattformen,7
Macintosh

 (*Siehe auch* Mac OS X)
 unterstützte Plattformen,7
Mandrake
 unterstützte Versionen,7
Materialisierte Ansichten
 ETL-Funktionen,30
Mehrbenutzer-Datenbank
 Definition,27
Mehrere Datenbanken
 auf einem einzelnen Server ausführen,29
 Konkurrenz um Ressourcen,65
Mehrere Versionen
 SQL Anywhere,64
Meldungslog
 (*Siehe auch* Datenbankserver-Meldungslog)
Meldungslogdatei
 (*Siehe auch* Datenbankserver-Meldungslog)
Merkmale
 SQL Anywhere,6
MIBs
 (*Siehe auch* Managementinformations-Datenbank)
Migrieren
 von einem anderen Produkt zu SQL Anywhere,67
MIPS-Prozessor
 unterstützte Plattformen,7
Mobile Datenverarbeitung
 Beispielanwendungen,1
MobiLink
 unterstützte Plattformen,7
 Vergleich von SQL Remote und MobiLink,35
MobiLink, konsolidierte Datenbanken
 unterstützte RDBMS-Systeme,7
MobiLink, unterstützte Plattformen
 Info,7
MobiLink-Synchronisation
 Funktionsvergleich,42
 unterstützte Plattformen,7
Monitor
 (*Siehe auch* Monitor-Metrik)
 (*Siehe auch* Systemmonitor)
MVCC (*Siehe* Snapshot-Isolation)

N

n-stufige Datenverarbeitung
 Einführung,28
Nachrichtenagent (dbremote)
 unterstützte Plattformen,7

Nachrichtenbasierte Synchronisation

Einführung in die Synchronisationstechnologien,41

NATIONAL CHAR VARYING-Datentyp (*Siehe*
NVARCHAR-Datentyp)

NATIONAL CHAR-Datentyp (*Siehe* NCHAR-
Datentyp)

NATIONAL CHARACTER VARYING-Datentyp
(*Siehe* NVARCHAR-Datentyp)

NATIONAL CHARACTER-Datentyp (*Siehe*
NCHAR-Datentyp)

NCHAR VARYING-Datentyp (*Siehe* NVARCHAR-
Datentyp)

Netzwerk-Datenbankserver (*Siehe* Netzwerkservers)

Netzwerkparameter

(*Siehe auch* Verbindungsparameter)

Netzwerkservers

Überblick,18

Netzwerkservers-Monitor

(*Siehe auch* SQL Anywhere-

Konsolendienstprogramm (dbconsole))

Neuladen von Datenbanken

(*Siehe auch* Datenbanken neu aufbauen)

newdemo

Info,52

Nicht geplante Anforderungen (*Siehe* ReqStatus-
Eigenschaft) (*Siehe* UnschReq-Eigenschaft)

Numerische Konstanten (*Siehe* binäre Literale)

O

Objektbezeichner (*Siehe* OIDs)

ODBC-Datenquellen

SQL Anywhere 16 CustDB,53

SQL Anywhere 16 Demo,49

ODBC-Treiber

(*Siehe auch* SQL Anywhere ODBC-Treiber)

OIDs

(*Siehe auch* Objektbezeichner)

OLE DB-Treiber

unterstützte Prozessoren,7

OMNI (*Siehe* Ferndatenzugriff)

Online, analytische Verarbeitung (*Siehe* OLAP)

Online-Dokumentation

PDF,v

Open Client (*Siehe* Sybase Open Client)

Open Servers (*Siehe* Sybase Open Servers)

Optionen

(*Siehe auch* Datenbankoptionen)

(*Siehe auch* Optionen (UltraLite))

Oracle Solaris

(*Siehe auch* Solaris)

P

PDF

Dokumentation,v

Performance

langsame Anwendung beheben,66

verbessern, Anwendung,66

verbessern, Datenbank,66

Personal Datenbankserver (*Siehe* Personal Servers)

Personal Servers

Überblick,18

Physische Datenmodelle

SAP Sybase PowerDesigner verwenden,47

Plattformen

unterstützte Betriebssysteme,7

Plattformunterstützung

Info,7

Pocket PCs

unterstützte Plattformen,7

Prädikate

(*Siehe auch* Suchbedingungen)

Praktische Einführungen

Beispieldatenbank neu erstellen,52

Info,55

vollständige Liste,55

Primärschlüssel

Definition,20

Programmierschnittstellen

(*Siehe auch* APIs)

Info,31

Protokollierung

(*Siehe auch* Diagnoseprotokollierung)

Protokollierung auf Anforderungsebene (*Siehe*

Anforderungsprotokollierung)

Protokolloptionen

(*Siehe auch* Verbindungsparameter)

Prozessoren

lizenzieren,2

R

Räumliche Vorschau, Registerkarte

(*Siehe auch* siehe auch Spatial Viewer)

READUNCOMMITTED, Tabellen-Hint

(*Siehe auch* Festgeschriebene Anweisungen)

(*Siehe auch* Nicht festgeschriebene Anweisungen)

Reaktionszeit

Vorteile der Datensynchronisation,37

Red Hat

unterstützte Versionen,7

Rehabilitation Act

Modul zur Aktivierung von Eingabehilfen,16

Unterstützung für Eingabehilfen,8

Relationale Datenbanken

Info,19

Relationale Datenbanksysteme

Info,17

Konzepte,18

Relationen

Entitäten,19

REMOTE DBA-Berechtigung (*Siehe* SYS_RUN_REPLICATION_ROLE-Systemrolle)

REPEATABLE_READ, Tabellen-Hint

(*Siehe auch* Wiederholbare Lesevorgänge)

Replikation

Info zu SQL Remote und MobiLink,35

Vergleich von SQL Remote und MobiLink,36

Rollen

in Beispieldatenbank,50

Rückmeldung

Aktualisierung beantragen,vii

Fehler melden,vii

S

SAP Sybase PowerDesigner

Info,47

Scale-Out (*Siehe* horizontal skalieren)

getrennt lizenzierbare Komponenten,5

Scale-Out mit Schreibschutz

getrennt lizenzierbare Komponenten,5

Schlüssel

Info,20

Schnellstart

SQL Anywhere 16,59

Schnittstellen

(*Siehe auch* APIs)

Schreibschutz, Scale-Out (*Siehe* Scale-Out mit Schreibschutz)

Schrittweise Beschreibungen

FAQ,68

scRepository (*Siehe* .scRepository610)

Selbständige Anwendungen

Definition,26

Sequenzgeneratoren (*Siehe* Sequenzen)

SERIALIZABLE, Tabellen-Hint

(*Siehe auch* serialisierbar)

Server

Anzahl der laufenden Datenbanken,64

Serversuche, Dienstprogramm (*Siehe* Dienstprogramm zur Serverauflistung (dblocate))

Shells

Anführungszeichen,vi

geschweifte Klammern,vi

Klammern,vi

Konventionen,vi

Sicherheitsoption

Info,3

Sicherungspläne

erstellen,66

Simple Network Management Protocol (*Siehe* SNMP)

Sitzungsbasierte Synchronisation

Info,41

Smartphone

SQL Anywhere-Datenbankunterstützung,26

UltraLite-Datenbankunterstützung,26

unterstützte Plattformen,7

SMTP-Nachrichtenagent

unterstützte Plattformen,7

Sofortige materialisierte Ansichten (*Siehe* sofortige Ansichten)

Solaris

unterstützte Plattformen,7

Spalten

(*Siehe auch* Einschränkungen)

Info,19

Spaltenstatistiken

(*Siehe auch* Histogramme)

Spatial Viewer

(*Siehe auch* siehe auch Räumliche Vorschau, Registerkarte)

Spawn, Dienstprogramm (*Siehe* Dienstprogramm für den Serverstart im Hintergrund (dbspawn))

Sperre

(*Siehe auch* sperren)

Sperren

(*Siehe auch* Sperre)

Spiegeln, Transaktionslog (*Siehe* Transaktionslog-Spiegel)

Spiegelung

(*Siehe auch* Spiegelung der Datenbank)

- getrennt lizenzierbare Komponenten,5
- Sprachen-Dienstprogramm (*Siehe* Dienstprogramm für die Spachauswahl)
- SQL Anywhere
 - Dokumentation,v
 - Einführung,1
 - Einsatzbereiche,1
 - Erste Orientierung,60
 - FAQ,63
 - interne Vorgänge,22
 - Komponenten,1
 - Merkmale,6
 - Überblick,1
 - unterstützte Plattformen,7
 - verglichen mit UltraLite,26
 - Wettbewerbsvorteil,68
- SQL Anywhere Tech Corner
 - wenn Sie mehr wissen wollen oder technischen Support benötigen,vii
- SQL Anywhere, unterstützte Plattformen
 - Info,7
- SQL Anywhere-Beispieldatenbank
 - Info,49
 - neu erstellen,52
 - wiederherstellen,52
- SQL Anywhere-Community
 - technischer Support,vii
- SQL Anywhere-Datenbanken
 - verglichen mit UltraLite-Datenbanken,26
- SQL Anywhere-Debugger (*Siehe* Debugger)
- SQL Anywhere-Forum
 - technischer Support,vii
- SQL Anywhere-Monitor (*Siehe* Monitor)
- SQL Remote
 - unterstützte Plattformen,7
 - Vergleich von MobiLink und SQL Remote,35
- SQL Remote, unterstützte Plattformen
 - Info,7
- SQL/2003-Konformität
 - (*Siehe auch* SQL-Standards)
- SQLX (*Siehe* SQL/XML)
- SSL (*Siehe* Transportschichtsicherheit)
- Standardkennwort
 - Datenbanken,63
- Standards
 - (*Siehe auch* SQL-Standards)
- Standards und Entsprechung
 - (*Siehe auch* SQL-Standards)

- Standardwerte
 - Kennwörter,63
- Starke Verschlüsselung
 - Verschlüsselungssoftware bestellen,3
- START ENGINE-Anweisung (nicht mehr empfohlen) (*Siehe* START SERVER-Anweisung)
- Statistiken
 - (*Siehe auch* Histogramme)
- STDDEV-Funktion (*Siehe* STDDEV_SAMP-Funktion)
- STOP ENGINE-Anweisung (*Siehe* STOP SERVER-Anweisung)
- Store-and-Forward
 - Synchronisation mit SQL Remote,41
- Suchbedingungen
 - (*Siehe auch* Prädikate)
- Support
 - SQL Anywhere-Community,vii
 - SQL Anywhere-Forum,vii
- SuSE
 - unterstützte Versionen,7
- Sybase Central
 - unterstützte Plattformen,7
- Synchronisation
 - Info zu SQL Remote und MobiLink,35
 - MobiLink, Funktionsvergleich,42
 - Vergleich von SQL Remote und MobiLink,36
- SYSSERVERS (*Siehe* SYSSERVER-Systemansicht)
- Systemprivilegien
 - (*Siehe auch* Privilegien)

T

- Tabellen
 - Eigenschaften,19
 - Info,19
- Tabular Data Stream, Kommunikationsprotokoll
 - Diagramm,31
 - Info,34
- TDS-Kommunikationsprotokoll
 - (*Siehe auch* Tabular Data Stream, Kommunikationsprotokoll)
- Tech Corners
 - wenn Sie mehr wissen wollen oder technischen Support benötigen,vii
- Technischer Support
 - SQL Anywhere-Community,vii
 - SQL Anywhere-Forum,vii

- Temporäre Dateien
 - Einführung,24
- Textindizes (*Siehe* Volltextsuche)
- Textvervollständigung (*Siehe* Textvervollständigung)
- Thumb-Modus
 - unterstützt,7
- Tools
 - Design- und Administrationstools,45
- Transaktionsintegrität
 - Info,38
- Transaktionslog
 - Einführung,24
- Transaktionstechnologie
 - Herausforderungen an die Synchronisationstechnologien,38
- Transportschichtssicherheit
 - Verschlüsselungssoftware bestellen,3
- TSQL (*Siehe* Transact-SQL)
- Tupel
 - Info,19
- TurboLinux
 - unterstützte Versionen,7
- Typen von Daten (*Siehe* Datentypen)

U

- Übereinstimmung mit SQL-Standards
 - (*Siehe auch* SQL-Standards)
- Überwachen
 - (*Siehe auch* MobiLink-Monitor)
- UltraLite
 - unterstützte Plattformen,7
 - verglichen mit SQL Anywhere,26
- UltraLite, unterstützte Plattformen
 - Info,7
- UltraLite-Datenbanken
 - verglichen mit SQL Anywhere-Datenbanken,26
- Umgebungsvariable, Option (*Siehe* @data-Option)
- Unix
 - unterstützte Plattformen,7
- Unterstützte Funktionen
 - (*Siehe auch* unterstützte Plattformen)
- Unterstützte Plattformen
 - Info,7
 - MobiLink,7
 - SQL Anywhere-Server,7
 - SQL Remote,7
 - UltraLite,7

V

- V4T-Modus
 - ARM-Prozessoren,7
- VARIANCE-Funktion (*Siehe* VAR_SAMP-Funktion)
- VCS, Agenten
 - getrennt lizenzierbare Komponenten,5
- Verbindungen
 - FAQ,63
- Verbindungs-Assistent (*Siehe* Verbindungsassistent)
- Verbindungs-IDs
 - (*Siehe auch* Number, Eigenschaft)
- Verbindungsparameter
 - (*Siehe auch* Protokolloptionen)
- Veritas Cluster Server, Agenten
 - getrennt lizenzierbare Komponenten,5
- Verschlüsselung
 - (*Siehe auch* Chiffrierung)
 - Verschlüsselungssoftware bestellen,3
- Versionsnummer
 - SQL Anywhere-Version ermitteln,64
- Vielstufige Datenverarbeitung
 - Einführung,28
- Virtualisierte Umgebungen
 - von SQL Anywhere unterstützt,8
- Virtualisierung
 - Unterstützung für virtualisierte Umgebungen,8
- Vista
 - unterstützte Plattformen,7
- VM
 - SQL Anywhere in virtualisierten Umgebungen,8
- Voluntary Product Accessibility Template
 - Unterstützung für Abschnitt 508,8
- VPAT
 - Unterstützung für Abschnitt 508,8

W

- Weitere Informationen und technische Unterstützung
 - technischer Support,vii
- Weitergabemethoden
 - Einführung in die Synchronisationstechnologien,40
- Whitepapers
 - zum Lesen empfohlen,viii
- Windows
 - (*Siehe auch* Windows 2003)
 - (*Siehe auch* Windows Mobile)
 - (*Siehe auch* Windows XP)
 - Konventionen in der Dokumentation,v

- unterstützte Plattformen,7
- Windows 2003
 - unterstützte Plattformen,7
- Windows 7
 - unterstützte Plattformen,7
- Windows CE (*Siehe* Windows Mobile)
- Windows Mobile
 - unterstützte Plattformen,7
 - unterstützte Prozessoren,7
- Windows Server 2008/2008 R2
 - unterstützte Plattformen,7
- Windows Vista
 - unterstützte Plattformen,7
- Windows XP
 - unterstützte Plattformen,7
- Workgroup Computing
 - (*Siehe auch* eingebettete Datenbanken)

X

- x86-Prozessor
 - unterstützte Plattformen,7
- XScale-Prozessoren
 - unterstützte Plattformen,7

Z

- Zeichenfolgenkonstanten (*Siehe* Zeichenfolgenlitterale)
- Zeilen
 - Info,19
- Zielpattformen
 - UltraLite,7
- Zustandsdateien (*Siehe* Zustandsinformationsdateien)
- Zwei-Wege-Synchronisation
 - Einführung in die Synchronisationstechnologien,40

