



# SQL Anywhere® 16 Introduction

**Version 16.0**

**February 2013**

Version 16.0  
February 2013

© 2013 SAP AG or an SAP affiliate company. All rights reserved.

You may use, print, reproduce, and distribute this documentation (in whole or in part) subject to the following conditions: 1) you must retain this and all other proprietary notices, on all copies of the documentation or portions thereof, 2) you may not modify the documentation, 3) you may not do anything to indicate that you or anyone other than SAP is the author or source of the documentation. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark>.

---

---

# Contents

<b>About this book</b> .....	<b>v</b>
About the SQL Anywhere documentation .....	v
<b>SQL Anywhere 16 overview</b> .....	<b>1</b>
SQL Anywhere in frontline environments .....	1
Editions and licensing .....	2
Separately licensed components .....	3
Hallmarks of SQL Anywhere 16 .....	5
Supported platforms .....	6
Accessibility .....	7
<b>Data management technologies</b> .....	<b>15</b>
The parts of a database system .....	15
Relational database concepts .....	16
Inside SQL Anywhere .....	20
SQL Anywhere and UltraLite target platforms .....	23
Database scenarios .....	24
Multi-tier computing architecture .....	25
Multiple databases running on a single database server .....	26
ETL features .....	27
Programming interfaces .....	28
<b>Overview of data exchange technologies</b> .....	<b>31</b>
Synchronization technology comparison .....	31
Propagation methods .....	36
Synchronization technology considerations .....	37
<b>Design and management tools</b> .....	<b>41</b>
<b>Sample databases</b> .....	<b>45</b>

<b>SQL Anywhere sample database .....</b>	<b>45</b>
<b>The CustDB sample database application .....</b>	<b>48</b>
<b>List of SQL Anywhere tutorials .....</b>	<b>51</b>
<b>Resources for getting started with SQL Anywhere 16 .....</b>	<b>55</b>
<b>Resources for getting started with SQL Anywhere Server .....</b>	<b>56</b>
<b>Frequently asked questions - SQL Anywhere .....</b>	<b>59</b>
<b>Index .....</b>	<b>65</b>

---

# About this book

This book introduces SQL Anywhere 16, a comprehensive package that provides data management and data exchange, enabling the rapid development of database-powered applications for server, desktop, mobile, and remote office environments.

## About the SQL Anywhere documentation

The complete SQL Anywhere documentation is available in four formats:

## Documentation conventions

This section lists the conventions used in this documentation.

### Operating systems

To simplify references to operating systems, the documentation groups the supported operating systems as follows:

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

### Directory and file names

Usually references to directory and file names are similar on all supported platforms, with simple transformations between the various forms. In these cases, Windows conventions are used. Where the details are more complex, the documentation shows all relevant forms.

These conventions are used to simplify the documentation of directory and file names:

- **Uppercase and lowercase directory names** On Windows and Unix, directory and file names can contain uppercase and lowercase letters.

On Windows, references to directories and files are *not* case sensitive. Mixed case directory and file names are common, but it is common to refer to them using all lowercase letters.

On Unix, references to directories and files *are* case sensitive. Mixed case directory and file names are not common. Most use all lowercase letters.

- **Slashes separating directory and file names** The documentation uses backslashes (\) as the directory separator. On Unix, replace the backslash with the forward slash (/).
- **Executable files** The documentation shows executable file names using Windows conventions, with a suffix such as *.exe* or *.bat*. On Unix, executable file names have no suffix.
- **32-bit and 64-bit versions** The documentation does not always distinguish between 32-bit and 64-bit versions of the software. In some cases, the documentation provides an example file path that

includes *bin32* or *bin64*. The two are interchangeable, and depend on the version of the software that was installed. For example, the file paths *C:\Program Files\SQL Anywhere 16\bin32* and *C:\Program Files\SQL Anywhere 16\bin64* are equivalent.

### Command prompts and command shell syntax

The documentation provides command line examples in a generic form. If these examples contain characters that the shell considers special, the command requires modification for the specific shell. The modifications are beyond the scope of this documentation, but generally, use quotes around the parameters containing those characters or use an escape character before the special characters.

- **Parentheses and curly braces** Some command line options require a parameter that accepts detailed value specifications in a list. The list is enclosed with parentheses or curly braces. The documentation uses parentheses. For example:

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

Where parentheses cause syntax problems, substitute curly braces:

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

If both forms result in syntax problems, enclose the entire parameter in quotes as required by the shell:

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

- **Semicolons** On Unix, enclose semicolons in quotes.
- **Quotes** If you must specify quotes in a parameter value, the quotes can conflict with the traditional use of quotes to enclose the parameter. For example, to specify an encryption key whose value contains double quotes, you can enclose the key in quotes and then escape the embedded quote:

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

In many shells, the value of the key would be my "secret" key.

## Online resources and additional help

Resource	Description
<b>SAP Sybase SQL Anywhere Community</b>	The SAP Sybase SQL Anywhere Community has links to blogs maintained by executives, employees, and developers who exchange views and ideas about using SQL Anywhere and related technologies. See <a href="http://scn.sap.com/community/sybase-sql-anywhere/content">http://scn.sap.com/community/sybase-sql-anywhere/content</a> .

Resource	Description
<b>SQL Anywhere Forum</b>	The SQL Anywhere Forum is a website where you can ask and answer questions about the SQL Anywhere software, and comment and vote on the questions of others and their answers. When you post a message on the forum, always provide details about your problem, including the build number of your version of SQL Anywhere. You can find this information by running the following command: <b>dbsrv16 -v</b> . See <a href="http://sqlanywhere-forum.sybase.com">http://sqlanywhere-forum.sybase.com</a> . There is also a community discussion page at <a href="http://scn.sap.com/community/sybase-sql-anywhere/content?filterID=content~objecttype~object-type[thread]#">http://scn.sap.com/community/sybase-sql-anywhere/content?filterID=content~objecttype~object-type[thread]#</a> .
<b>DocCommentXchange</b>	DocCommentXchange is a website where you can read the SAP Sybase SQL Anywhere documentation, leave questions or feedback on the documentation, and read the documentation questions and replies of others. See <a href="http://dcx.sybase.com">http://dcx.sybase.com</a> .
<b>SQL Anywhere Supported Platforms and Engineering Support Status</b>	Platforms supported by SAP Sybase SQL Anywhere. See <a href="http://www.sybase.com/detail?id=1002288">http://www.sybase.com/detail?id=1002288</a> .
<b>SQL Anywhere Tech Corner</b>	The SQL Anywhere Tech Corner gives developers easy access to product technical documentation. You can browse technical white papers, FAQs, tech notes, downloads, techcasts, and more to find answers to your questions as well as solutions to many common issues. See <a href="http://www.sybase.com/developer/library/sql-anywhere-techcorner">http://www.sybase.com/developer/library/sql-anywhere-techcorner</a> .
<b>SQL Anywhere .NET Developer Center</b>	Get started and get answers to specific questions regarding SQL Anywhere and .NET development. See <a href="http://www.sybase.com/developer/library/sql-anywhere-techcorner/microsoft-net">www.sybase.com/developer/library/sql-anywhere-techcorner/microsoft-net</a> .
<b>PHP Developer Center</b>	An introduction to using the PHP (PHP Hypertext Preprocessor) scripting language to query your database. See <a href="http://www.sybase.com/developer/library/sql-anywhere-techcorner/php">www.sybase.com/developer/library/sql-anywhere-techcorner/php</a> .
<b>SQL Anywhere Windows Mobile Developer Center</b>	Get started and get answers to specific questions regarding SQL Anywhere and Windows Mobile development. See <a href="http://www.sybase.com/developer/library/sql-anywhere-techcorner/windows-mobile">www.sybase.com/developer/library/sql-anywhere-techcorner/windows-mobile</a> .

### Top 10 recommended whitepapers

- **SQL Anywhere I/O Requirements for Windows and Linux** <http://scn.sap.com/docs/DOC-34090>
- **Capacity Planning with SQL Anywhere** <http://scn.sap.com/docs/DOC-35069>
- **Diagnosing Application Performance Issues with SQL Anywhere** <http://scn.sap.com/docs/DOC-35601>

- **Analytic Functions in SQL Anywhere** <http://scn.sap.com/docs/DOC-35603>
- **Optimizing Adaptive Server Anywhere Performance Over a WAN** <http://scn.sap.com/docs/DOC-35605>
- **Semantics and Compatibility of Transact-SQL Outer Joins** <http://scn.sap.com/docs/DOC-35606>

### Earlier versions of the documentation

A compilation of new features and behavior changes from versions 6.0 to 10.0.1 and later can be found at <http://dcx.sybase.com/html/dbwnen10/dbwnen10.html>.

Full documentation for SQL Anywhere versions 10.0.1 up to the current release can be found at <http://dcx.sybase.com>.

From version 8.0 to 9.0.2, SQL Anywhere was called SQL Anywhere Studio. Full documentation for these versions can be found at [http://manuals.sybase.com/onlinebooks/group-sas/@Generic\\_\\_CollectionView;pt=group-awarc](http://manuals.sybase.com/onlinebooks/group-sas/@Generic__CollectionView;pt=group-awarc).

Prior to version 8, SQL Anywhere was called Adaptive Server Anywhere. Full documentation for version 6 can be found at: [http://manuals.sybase.com/onlinebooks/group-awarc/@Generic\\_\\_CollectionView;pt=group-sas](http://manuals.sybase.com/onlinebooks/group-awarc/@Generic__CollectionView;pt=group-sas).



---

# SQL Anywhere 16 overview

SQL Anywhere is a comprehensive package that provides technologies for data management and enterprise data exchange, enabling the rapid development of database-powered applications for server, desktop, mobile, and remote office environments.

SQL Anywhere offers:

- **Data management technologies** SQL Anywhere provides enterprise-caliber databases that are designed to handle the challenges of operating in many different frontline environments—from a high performance database server deployed with an independent software vendor application, to a mobile database that can be deployed to tens of thousands of handheld devices within the enterprise.
- **Data exchange technologies** SQL Anywhere offers several data exchange technologies to handle the complexities of exchanging data across unreliable wired and wireless networks to back-end databases, application servers, and messaging systems. In addition, SQL Anywhere mobile messaging and synchronization technologies guarantee secure message delivery for distributed and mobile computing.
- **Design and management tools** SQL Anywhere includes a suite of tools to improve the design and development of database-driven applications, and to simplify the management of databases and data exchange environments.

## See also

- [“Relational database concepts” on page 16](#)
- [“Overview of data exchange technologies” on page 31](#)
- [“Design and management tools” on page 41](#)

## SQL Anywhere in frontline environments

SQL Anywhere technologies are used in many different ways by over 10000 customers. Four common uses of SQL Anywhere are:

- **Client-server applications** Whether it is 5, 50, 500 users or more, SQL Anywhere is a powerful database solution for server applications, providing high performance out of the box, with low maintenance and cost.

SQL Anywhere easily scales to support hundreds of active users, hundreds of gigabytes of data, and hundreds of millions of rows. Yet many ease-of-use and administration features ensure that costs stay down as performance scales up.

This deployment model works best when the majority of users are connected to the network.

- **Desktop applications** SQL Anywhere delivers enterprise-caliber features, without the bulky characteristics of an enterprise database. Its robust reliability and performance, along with highly efficient usage of memory and system resources, ensure that the database can be hidden from laptop and desktop users.

Organizations embed SQL Anywhere databases in their applications because SQL Anywhere databases are built for use in widely deployed, minimally administered environments, and require minimal memory and disk space.

- **Remote office applications** SQL Anywhere data exchange architectures address the challenges of managing and sending data within and between offices and workers that are dispersed geographically.

Companies choose SQL Anywhere database and data exchange technologies to provide remote workers with the data they need to run their operations effectively, while providing the central office with the critical information that gives the pulse of the business.

- **Mobile and wireless applications** Recognized as the industry's leading mobile database, SQL Anywhere gives mobile workers the ability to access their data and corporate applications. Regardless of connection or application type, SQL Anywhere data exchange technologies ensure that mobile workers stay productive by having the information they need, when they need it. Workers can access information and queue up transactions offline, reducing communications costs while increasing application and battery performance.

Companies depend on SQL Anywhere for reliable management of data and mobile applications running on laptops, handheld devices, and smartphones.

#### See also

- [“Client/server applications” on page 25](#)
- [“Desktop applications and embedded databases” on page 24](#)
- [“Consolidated and remote databases” on page 34.](#)
- [“UltraLite overview” \[\*UltraLite - Database Management and Reference\*\]](#)
- [“SQL Anywhere for Windows Mobile” \[\*SQL Anywhere Server - Database Administration\*\]](#)

## Editions and licensing

SQL Anywhere offers various editions that include certain separately licensed components and that can restrict the number of CPUs used by the database server. For more information about editions, see <http://www.sybase.com/detail?id=1068247>.

### Licensing and CPUs

**Per-seat licensing** A per-seat license restricts the number of client connections to the database server. With per-seat licensing, the network database server can use all CPUs available on your computer, but not more than the maximum allowed by the SQL Anywhere edition you are running. If a CPU contains multiple cores or threads, all cores and threads of the CPU may be used.

The personal database server is limited to four cores on one CPU. If a core contains multiple threads, all threads of that core may be used.

**CPU-based licensing** With CPU-based licensing, the network database server uses up to the number of CPUs specified in your license, but not more than the maximum allowed by the SQL Anywhere edition you are running. If a CPU contains multiple cores or threads, all cores and threads of the CPU may be used.

The personal database server is limited to four cores on one CPU. If a core contains multiple threads, all threads of that core may be used.

You can further restrict the processors that the database server uses with the following options:

- -gt database server option
- -gta database server option
- -gtc database server option
- sa\_server\_option system procedure with the ProcessorAffinity option

### See also

- “Separately licensed components” on page 3
- “Server Licensing utility (dblic)” [*SQL Anywhere Server - Database Administration*]
- *SQL Anywhere Licensing*
- “-gt database server option” [*SQL Anywhere Server - Database Administration*]
- “-gta database server option” [*SQL Anywhere Server - Database Administration*]
- “-gtc database server option” [*SQL Anywhere Server - Database Administration*]
- ProcessorAffinity option: “sa\_server\_option system procedure” [*SQL Anywhere Server - SQL Reference*]

## Separately licensed components

The following components are licensed separately and may need to be ordered separately if not included in your edition of SQL Anywhere. For more information about the various SQL Anywhere editions, see <http://www.sybase.com/detail?id=1068247>.

To order a separately licensed component, go to <http://www.sybase.com/detail?id=1015780>.

### SQL Anywhere security option

With SQL Anywhere you can strongly encrypt database files, and synchronization and client-server communication transport layers.

SQL Anywhere offers the following strong encryption algorithms:

Feature	Included with a separately licensed security option <sup>1</sup>	Included with SQL Anywhere <sup>2</sup>
Database encryption	FIPS-certified AES	AES
Transport layer security	FIPS-certified RSA	RSA

<sup>1</sup> The software for strong encryption using FIPS-certified technology must be ordered separately.

<sup>2</sup> AES and RSA strong encryption are included with SQL Anywhere and do not require a separate license, but these libraries are not FIPS-certified.

RSA and FIPS-certified encryption are not available on all platforms. For information about which platforms support which encryption method, see <http://www.sybase.com/detail?id=1061806>.

The security option provides Certicom DLLs that implement the encryption algorithms, and additional DLLs that provide an interface between SQL Anywhere software and the Certicom libraries.

The SQL Anywhere security option includes the following:

- For Windows operating systems, Certicom Security Builder GSE.  
For more information, see number 542 at <http://csrc.nist.gov/cryptval/140-1/140val-all.htm>.
- For Windows Mobile operating systems, Certicom Security Builder GSE.  
For more information, see number 316 at <http://csrc.nist.gov/cryptval/140-1/140val-all.htm>.

For more information about encryption, see “Data security” [*SQL Anywhere Server - Database Administration*].

### **SQL Anywhere CAC Authentication option**

For UltraLite databases, the license for CAC Authentication must be ordered separately. In addition, this option requires the SQL Anywhere security option.

### **SQL Anywhere in-memory mode option**

For SQL Anywhere databases, the license for using the in-memory mode must be ordered separately.

For information about platform support, see [SQL Anywhere Supported Platforms and Engineering Support Status](#).

For more information about in-memory mode, see “-im database server option” [*SQL Anywhere Server - Database Administration*].

### **SQL Anywhere read-only scale-out option**

For SQL Anywhere databases, the license for using read-only scale-out must be ordered separately. Read-only scale-out is a configuration that allows you to offload reporting or other operations that require read-only access to the database. For more information about the read-only scale-out option, see “SQL Anywhere read-only scale-out” [*SQL Anywhere Server - Database Administration*].

### **SQL Anywhere high availability option**

For SQL Anywhere databases, the license for using the Veritas Cluster Server agents or for using database mirroring for failover must be ordered separately.

For information about platform support, see [SQL Anywhere Supported Platforms and Engineering Support Status](#).

For more information about database mirroring, see “Database mirroring” [*SQL Anywhere Server - Database Administration*].

For more information about the SQL Anywhere Veritas Cluster Server agents, see “SQL Anywhere Veritas Cluster Server agents” [*SQL Anywhere Server - Database Administration*].

### MobiLink high availability option

The MobiLink high availability option allows you to group multiple MobiLink servers into server farms of identical servers using the shared state mode. When MobiLink runs in shared state mode it can block the same remote database from synchronizing simultaneously with multiple servers, thereby ensuring data integrity. MobiLink shared state support also enables load balancing and failover for server initiated sync.

For more information about running MobiLink in a server farm, see [“MobiLink server in a server farm” \[MobiLink - Server Administration\]](#).

### SQL Anywhere Monitor Production Edition

The SQL Anywhere Monitor Production Edition must be ordered separately. The production edition is intended for deployment and production use. For more information about the Monitor, see [“SQL Anywhere Monitor” \[SQL Anywhere Server - Database Administration\]](#).

## Hallmarks of SQL Anywhere 16

The following is a list of SQL Anywhere hallmarks that you can take advantage of:

- **Embeddability** SQL Anywhere can be easily embedded inside other applications. It combines high performance with very small memory footprint. SQL Anywhere contains a range of features to enable self-management and maintenance in frontline environments, including features that enable optimization of computer resources, self-tuning for improved performance, and simplification of remote installation and support.
- **Interoperability** SQL Anywhere is available on many platforms, including Windows, Windows Mobile, Linux, Oracle Solaris, HP-UX, IBM AIX, and Mac OS X. Unique to SQL Anywhere, its database files can be copied between platforms. In addition, SQL Anywhere provides support for BlackBerry, Embedded Linux, Windows Mobile 6, and Java SE smartphones using its UltraLite database technology for small devices. SQL Anywhere includes support for many common database interfaces, including ODBC, JDBC, ADO.NET, PHP, and Perl. Many popular application development tools can be used, including: Microsoft Visual Studio, Sybase PowerBuilder, Eclipse, and various web tools. Stored procedures can be written in C/C++, Java, .NET, or Perl.
- **Performance out of the box** SQL Anywhere is designed to deliver outstanding performance, without ongoing tuning and administration. Features such as dynamic cache sizing, auto generation of statistics, a sophisticated query optimizer, parallel query processing, and materialized views mean that SQL Anywhere is ideal for environments that demand high performance but which have no on-site database administrator. By offering, On-Line Analytical Processing (OLAP) SQL Anywhere offers the ability to perform complex data analysis within a single SQL statement, increasing the value of the results, while improving performance by decreasing the amount of querying on the database.
- **Web operation** With a built-in HTTP server and web service support, XML features, full text search, and a PHP interface, SQL Anywhere is an ideal database for use in a web-based environment behind a web server.
- **Mobility** SQL Anywhere provides enterprise-caliber databases that operate on frontline systems and devices whether connectivity with enterprise systems is available or not. Its synchronization

technologies ensure data can be exchanged efficiently over wireless and wired networks with back-end databases, application servers, and messaging systems.

- **Security** SQL Anywhere provides full end-to-end security with 128-bit strong encryption of database tables, files, and communications streams between the application and the database, and the MobiLink synchronization stream. SQL Anywhere can audit data access, offers built-in user authentication, and can integrate with third-party authentication systems. SQL Anywhere also offers FIPS-certified encryption via a separately licensed security option.

#### See also

- [“SQL Anywhere security option” on page 3](#)

## Supported platforms

The **SQL Anywhere Supported Platforms and Engineering Support Status** web page lists the supported operating system platforms broken down by version of SQL Anywhere. It also indicates the engineering support status for each SQL Anywhere version. See <http://www.sybase.com/detail?id=1002288>.

From this web page, you can go to the following web pages for more support information, such as:

Web page name	URL	Description
<b>SQL Anywhere Components by Platforms</b>	<a href="http://www.sybase.com/detail?id=1061806">http://www.sybase.com/detail?id=1061806</a>	Provides a list of components that are available across each platform supported by SQL Anywhere. With a few exceptions, components that are available in all supported platforms are not listed. For ease of reading, the platforms are grouped by operating system vendor, operating system name and processor architecture.
<b>SQL Anywhere Supported Linux Platforms</b>	<a href="http://www.sybase.com/detail?id=1035824">http://www.sybase.com/detail?id=1035824</a>	Provides information about the Linux platform support, including support exceptions, key components, and tested Linux distributions.
<b>SQL Anywhere Supported Kerberos Clients</b>	<a href="http://www.sybase.com/detail?id=1061807">http://www.sybase.com/detail?id=1061807</a>	Provides a list of Kerberos clients/runtimes that have been tested with SQL Anywhere.
<b>SQL Anywhere Supported Listener Platforms</b>	<a href="http://www.sybase.com/detail?id=1061808">http://www.sybase.com/detail?id=1061808</a>	Provides a list of supported Listener platforms for SQL Anywhere.
<b>ODBC Drivers for MobiLink</b>	<a href="http://www.sybase.com/detail?id=1011880">http://www.sybase.com/detail?id=1011880</a>	Provides a list of the recommended ODBC drivers for different versions of the MobiLink server.

Web page name	URL	Description
SQL Anywhere Client Interfaces	<a href="http://www.sybase.com/detail?id=1068981">http://www.sybase.com/detail?id=1068981</a>	Provides a list of supported client interfaces for SQL Anywhere.

For information about software updates, see “Software updates” [[SQL Anywhere Server - Database Administration](#)].

### Support for virtualized environments

You can run SQL Anywhere on a supported platform in a virtualized environment, with some restrictions.

## Accessibility

SQL Anywhere supports Section 508 of the United States Federal Rehabilitation Act of 1973 (29 U.S.C. 794d).

The tables below identify how SQL Anywhere meets the accessibility requirements identified in Section 508 of the US Federal Rehabilitation Act when the product is run on the Windows x86 and x64 operating system platforms. An accessibility enablement component provides software that enables the use of accessibility tools.

For the purposes of this document, the following are components of SQL Anywhere:

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

### 3rd Party Tools

Testing environment: SQL Anywhere 16.0 (64-bit Windows) using both the Microsoft Accessibility technologies and the JAWS Screen Reader from Frontier Technologies.

### SQL Anywhere 16.0 Voluntary Product Accessibility Template Summary

Criteria	Supporting Features	Remarks and explanations
<a href="#">“Section 1194.21 Software Applications and Operating Systems” on page 8</a>	<b>Supports All Features</b>	Features and capabilities may also be complemented with Microsoft Windows Accessibility Features and Aids.
Section 1194.22 Web-based Internet Information and Applications	Not Applicable	

Criteria	Supporting Features	Remarks and explanations
Section 1194.23 Telecommunications Products	Not Applicable	
Section 1194.24 Video and Multi-media Products	Not Applicable	
Section 1194.25 Self-Contained, Closed Products	Not Applicable	
Section 1194.26 Desktop and Portable Computers	Not Applicable	
<a href="#">“Section 1194.31 Functional Performance Criteria” on page 11</a>	<b>Supported With Exceptions</b>	Supports 3rd party accessibility products and technologies supported on Microsoft Windows.  Some 3rd Party Products May Not Be Fully Compatible.
<a href="#">“Section 1194.41 Information, Documentation and Support” on page 13</a>	<b>Supported</b>	

**Section 1194.21 Software Applications and Operating Systems**

Criteria	Supporting Features	Remarks and explanations
(a) When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.	<b>Supports</b>  Keyboard access is provided throughout SQL Anywhere.  Keyboard shortcuts, shortcut keys, and menu commands are readily available in SQL Anywhere.	SQL Anywhere fully supports Microsoft Windows accessibility features.



Criteria	Supporting Features	Remarks and explanations
<p>(b) Applications shall not disrupt or disable activated features of other products that are identified as accessibility features, where those features are developed and documented according to industry standards. Applications also shall not disrupt or disable activated features of any operating system that are identified as accessibility features where the application programming interface for those accessibility features has been documented by the manufacturer of the operating system and is available to the product developer.</p>	<p><b>Supports</b></p> <p>SQL Anywhere does not disrupt or disable any accessibility features of the operating system.</p> <p>SQL Anywhere supports large fonts in the title bar, toolbars and most dialogs and SQL Anywhere fully supports operating system accessibility features including StickyKeys, FilterKeys, MouseKeys, and High Contrast.</p>	<p>Microsoft Windows accessibility aids may be used with the software. Screen readers may be used with the software. Java-based components require screen readers that support Java Assistive Technology.</p> <p>The Microsoft Windows magnifier application is available to magnify dialog text.</p> <p>Windows Screen Magnifier may not always follow focus when using Tab or Shortcut Keys.</p>
<p>(c) A well-defined on-screen indication of the current focus shall be provided that moves among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that Assistive Technology can track focus and focus changes.</p>	<p><b>Supports</b></p> <p>Visual on-screen focus, highlighting and tracking is provided throughout SQL Anywhere.</p>	<p>For the Java-based components in SQL Anywhere, third-party screen reader used must support Java assistive technology.</p> <p>Windows Screen Magnifier may not always follow focus for Java-based components when using Tab or Shortcut Keys.</p>
<p>(d) Sufficient information about a user interface element including the identity, operation and state of the element shall be available to Assistive Technology. When an image represents a program element, the information conveyed by the image must also be available in text.</p>	<p><b>Supports</b></p> <p>SQL Anywhere supports both the Microsoft® Active Accessibility® and Java Accessibility programming interface which enable its components to be compatible with assistive technology.</p>	<p>For the Java-based components in SQL Anywhere, third-party screen reader used must support Java assistive technology.</p>

Criteria	Supporting Features	Remarks and explanations
(e) When bitmap images are used to identify controls, status indicators, or other programmatic elements, the meaning assigned to those images shall be consistent throughout an application's performance.	<p><b>Supports</b></p> <p>SQL Anywhere utilizes standard and consistent images throughout the SQL Anywhere suite of applications.</p>	
(f) Textual information shall be provided through operating system functions for displaying text. The minimum information that shall be made available is text content, text input caret location, and text attributes.	<p><b>Supports</b></p> <p>SQL Anywhere uses standard system functions for displaying text.</p>	
(g) Applications shall not override user selected contrast and color selections and other individual display attributes.	<p><b>Supports</b></p> <p>SQL Anywhere supports system-wide colors settings as identified in the operating system display properties.</p>	Customizations made in SQL Anywhere utilities for visual representation will be specific to the application, and does not change system-wide settings.
(h) When animation is displayed, the information shall be displayable in at least one non-animated presentation mode at the option of the user.	<p><b>Supports</b></p> <p>SQL Anywhere uses animation sparingly.</p>	Where animation is used, the information conveyed is not essential, or is supplementary.
(i) Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.	<p><b>Supports</b></p> <p>SQL Anywhere uses color coding in association with other visual indicators when conveying information, indicating an action, prompting a response, or distinguishing a visual element.</p>	The Sybase Central database statistics graphing application provides a user-selectable range of colors, while providing text/numeric output of the information as well.

Criteria	Supporting Features	Remarks and explanations
(j) When a product permits a user to adjust color and contrast settings, a variety of color selections capable of producing a range of contrast levels shall be provided.	<p><b>Supports</b></p> <p>Where color is used to convey information, SQL Anywhere allows a user to select from a wide range of colors.</p>	SQL Anywhere supports operating system settings for high contrast color schemes that make the screen easier to view for some users by heightening screen contrast with alternate color combinations.
(k) Software shall not use flashing or blinking text, objects, or other elements having a flash or blink frequency greater than 2 Hz and lower than 55 Hz.	<p><b>Supports</b></p> <p>SQL Anywhere does not use flashing or blinking objects or text in any application user interface that falls within this frequency range.</p>	
(l) When electronic forms are used, the form shall allow people using Assistive Technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.	<p><b>Supports</b></p> <p>SQL Anywhere electronic forms (dialogs) support assistive technologies.</p>	For the Java-based components in SQL Anywhere, third-party screen reader used must support Java assistive technology.

### Section 1194.31 Functional Performance Criteria

Criteria	Supporting Features	Remarks and explanations
(a) At least one mode of operation and information retrieval that does not require user vision shall be provided, or support for Assistive Technology used by people who are blind or visually impaired shall be provided.	<p><b>Supports</b></p>	<p>SQL Anywhere supports 3rd party technologies that make computer programs more accessible to people who use assistive technology.</p> <p>SQL Anywhere also supports the use of screen readers to access user interface information. 3rd party products supporting Java Assistive Technologies are required for Java-Based tools.</p>

Criteria	Supporting Features	Remarks and explanations
(b) At least one mode of operation and information retrieval that does not require visual acuity greater than 20/70 shall be provided in audio and enlarged print output working together or independently, or support for Assistive Technology used by people who are visually impaired shall be provided.	<b>Supports with Exceptions</b>	SQL Anywhere supports the use of screen readers to access user interface information. SQL Anywhere, with few exceptions, also supports system large font settings.
(c) At least one mode of operation and information retrieval that does not require user hearing shall be provided, or support for Assistive Technology used by people who are deaf or hard of hearing shall be provided.	<b>Supports</b> SQL Anywhere does not require user hearing for access to any SQL Anywhere component functionality.	In all instances where SQL Anywhere provides an audio cue, it provides a visual cue as well.
(d) Where audio information is important for the use of a product, at least one mode of operation and information retrieval shall be provided in an enhanced auditory fashion, or support for assistive hearing devices shall be provided.	<b>Supports</b> SQL Anywhere does not require user hearing for access to any SQL Anywhere component functionality.	In all instances where SQL Anywhere provides an audio cue, it provides a visual cue as well.
(e) At least one mode of operation and information retrieval that does not require user speech shall be provided, or support for Assistive Technology used by people with disabilities shall be provided.	<b>Supports</b> SQL Anywhere components do not require speech recognition.	

Criteria	Supporting Features	Remarks and explanations
(f) At least one mode of operation and information retrieval that does not require fine motor control or simultaneous actions and that is operable with limited reach and strength shall be provided.	<p><b>Supports</b></p> <p>SQL Anywhere supports Microsoft Windows operating system tools such as MouseKeys, StickyKeys and FilterKeys as well as other assistive technologies.</p>	

### Section 1194.41 Information, Documentation and Support

Criteria	Supporting Features	Remarks and explanations
(a) Product support documentation provided to end-users shall be made available in alternate formats upon request, at no additional charge.	<p><b>Fully Supports</b></p>	<p>Documentation is provided in digital format for customers on the web and available for no- charge download at <a href="http://sybooks.sybase.com/">http://sybooks.sybase.com/</a>.</p>
(b) End-users shall have access to a description of the accessibility and compatibility features of products in alternate formats or alternate methods upon request, at no additional charge.	<p><b>Fully Supports</b></p>	<p>SQL Anywhere includes Help File documentation listing accessibility features. An individual may access and view Help File topics online or may select a topic for print.</p>
(c) Support services for products shall accommodate the communication needs of end-users with disabilities.	<p><b>Fully Supports</b></p> <p>SQL Anywhere does not require user hearing for access to any SQL Anywhere component functionality.</p>	<p>The Sybase Technical Support group offers a variety of services including Technical Documents, White Papers, Product Manuals, News-groups, online Software Problem Case Management, and Bug Fixes and Patches. For information on these and other support services, visit the Sybase Technical Support web site at <a href="http://www.sybase.com/support">http://www.sybase.com/support</a>.</p>

## Accessibility using the Java Access Bridge

SQL Anywhere 16 includes an accessibility enablement module. The Java Access Bridge module, which is loaded whenever you use Sybase Central or Interactive SQL, supports assistive technologies. Third-party software such as screen readers use this module to provide access to software features.

For information about platform support for the SQL Anywhere software accessibility, see <http://www.sybase.com/detail?id=1061806>.

For more information about SQL Anywhere accessibility, see “Accessibility” on page 7.

For more information about Sybase software product accessibility, see <http://www.sybase.com/accessibility>.

---

# Data management technologies

SQL Anywhere offers two relational databases: SQL Anywhere Server and UltraLite.

## SQL Anywhere Server

SQL Anywhere Server provides enterprise-caliber functionality including full transaction processing, referential integrity, materialized views, snapshot isolation, high availability via database mirroring and server clustering, SQL and Java stored procedures, triggers, row-level locking, automatic event scheduling, automatic backup and recovery, full-text searching, support for spatial data, and much more. SQL Anywhere Server can easily scale to hundreds of concurrent users and hundreds of gigabytes of data. Yet its small footprint and its many features that automate administration make it an ideal database to embed into server and desktop applications that are widely deployed in customer and remote sites.

## UltraLite

For environments that demand smaller data-driven applications, the UltraLite database is ideal. UltraLite is a full relational database management system designed specifically to minimize memory and system requirements for deployment to handhelds and other mobile devices, including the iPhone, Android, and BlackBerry. It provides full transaction-processing, a choice of development models, and a built-in synchronization client for exchanging data with other databases.

## The parts of a database system

A **relational database management system** (RDBMS) is a system for storing and retrieving data, in which the data is organized into interrelated tables.

Relational database management systems contain the following pieces:

- a database
- a database server
- an application programming interface (API)
- a client application
- **Database** A database is collection of tables that are related by primary and foreign keys. The tables hold the information in the database. The tables and keys together define the structure of the database. A database management system accesses this information.

A SQL Anywhere database is a file, usually with an extension of *.db*. An UltraLite database is also a file, usually with an extension of *.udb*. SQL Anywhere includes a sample database, and it is installed in the SQL Anywhere samples directory: `%SQLANY%SAMP16%\demo.db`.

- **Database server** The database server manages the database. All access to the database occurs through the database server.

The database server allows access to databases from client applications, and processes commands in a secure and efficient manner. A database can have only one server managing it at a time. However, a SQL Anywhere database server can manage many databases at one time.

There are two versions of the SQL Anywhere database server: the **personal server** and the **network server**. Both servers offer the same query processing and other internal operations; the only difference is in the number and types of connections each server accepts. For more information about the differences between personal and network database servers, see [“Differences between the network and personal servers” \[SQL Anywhere Server - Database Administration\]](#).

- **UltraLite runtime library** In UltraLite, the database management systems that are typically found in a database server are implemented as an in-process runtime library. The runtime library and the application are part of the same process.
- **Programming interface** Applications communicate with the database server using a programming interface such as ODBC, JDBC, OLE DB, ADO.NET, or embedded SQL.

For a complete list of supported programming interfaces in SQL Anywhere and UltraLite, see [“Programming interfaces” on page 28](#).

Each programming interface provides a library of function calls for communicating with the database. For ODBC and JDBC, the library is commonly called a **driver**. The library is typically provided as a shared library on Unix operating systems or a dynamic link library (DLL) on Windows operating systems.

- **Client application** Client applications use one of the programming interfaces to communicate with the database server.

If you develop an application using a rapid application development (RAD) tool such as Sybase PowerBuilder, you may find that the tool provides its own methods for communicating with database servers, and hides the details of the language interface. Nevertheless, all applications use one of the supported interfaces.

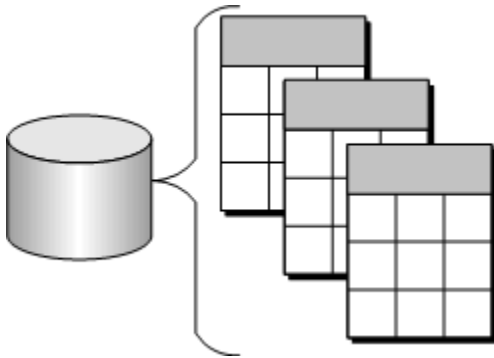
## Relational database concepts

The following sections provide a brief review of basic relational database concepts. They include definitions of tables, primary and foreign keys, and database objects.

### Database tables

In a relational database, all data is held in **tables**, which are made up of **rows** and **columns**.





Each table has one or more columns, and each column is assigned a specific data type, such as an integer, a sequence of characters (for text), or a date. Each row in the table has a single value for each column.

For example, a table containing employee information can look like the following:

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

### Characteristics of relational tables

The tables in a relational database have some important characteristics:

- There is no significance to the order of the columns or rows.
- Each row contains one and only one value for each column, or contains NULL, which indicates that there is no value for that column.
- All values for a given column have the same data type.

The following table lists some of the formal and informal relational database terms describing tables and their contents, together with their equivalent in non-relational databases such as dBase and Microsoft Visual FoxPro. This document uses the informal terms.

Informal relational term	Formal relational term	Non-relational term
Table	Relation	File
Column	Attribute	Field
Row	Tuple	Record

### What do you keep in each table?

Each table in the database should hold information about a specific kind of thing, such as employees, products, or customers.

By designing a database this way, you can set up a structure that eliminates redundancy and the possible inconsistencies caused by redundancy. For example, both the sales and accounts payable departments might enter and look up information about customers. In a relational database, the information about customers is stored only once, in a table that both departments can access.

### See also

- “Database creation” [[SQL Anywhere Server - Database Administration](#)]

## Relationships between tables

You use primary keys and foreign keys to describe relationships between the information in different tables. **Primary keys** identify each row in a table uniquely, and **foreign keys** define the relationships between rows in different tables.

Primary keys and foreign keys let you use relational databases to hold information in an efficient manner, with minimum redundancy.

### See also

- “Foreign keys” [[SQL Anywhere Server - SQL Usage](#)]
- “Primary keys” [[SQL Anywhere Server - SQL Usage](#)]

## Other database objects

There is more to a relational database than a set of related tables. You can also find the following objects in a relational database:

- **Indexes** Indexes allow quick look up of information. Conceptually, an index in a database is like an index in a book. In a book, the index relates each indexed term to the page or pages on which that word appears. In a database, the index relates each indexed column value to the physical location at which the row of data containing the indexed value is stored.

Indexes are an important design element for high performance. You usually must create indexes explicitly, but indexes for primary, foreign keys, and for unique columns are created automatically. Once created, the use of indexes is transparent to the user. See “[Indexes](#)” [[SQL Anywhere Server - SQL Usage](#)].

- **Text indexes** Text indexes store complete positional information for every instance of every term in every indexed column. When you perform a full text search, a text index is used to find matching rows. For this reason, queries that use text indexes can be faster than those that must scan all the values in the table. See “[Text index concepts and reference](#)” [[SQL Anywhere Server - SQL Usage](#)], and “[Full text search](#)” [[SQL Anywhere Server - SQL Usage](#)].
- **Login policies** Login policies consist of a set of rules that are applied when you create a database connection for a user. See “[Login policies](#)” [[SQL Anywhere Server - Database Administration](#)].

- **Views** Views are temporary tables. They look like tables to client applications, but they do not hold data. Instead, whenever they are accessed, the information in them is computed from the underlying tables.

The tables that actually hold the information are sometimes called **base tables** to distinguish them from views. A view is defined with a SQL query on base tables or other views.

See [“Views” \[SQL Anywhere Server - SQL Usage\]](#).

- **Materialized views** SQL Anywhere also supports materialized views. A materialized view is a view whose result set has been computed and stored on disk, similar to a base table. Conceptually, a materialized view is both a view (it has a query specification) and a table (it has persistent materialized rows). So, many operations that you perform on tables can be performed on materialized views as well. For example, you can build indexes on, and unload from, materialized views.

Materialized views are ideal for environments where the database is large, frequent queries result in repetitive aggregation and join operations on large amounts of data, and access to up-to-the-moment data is not a critical requirement. See [“Materialized views” \[SQL Anywhere Server - SQL Usage\]](#).

- **Stored procedures and triggers** These are routines held in the database that act on the information in the database.

You can create and name your own stored procedures to execute specific database queries and to perform other database tasks. Stored procedures can accept parameters and return result sets. For example, you might create a stored procedure that returns the names of all customers who have spent more than the amount that you specify as a parameter in the call to the procedure.

A trigger is a special stored procedure that automatically fires whenever a user updates, deletes, or inserts data, depending on how you define the trigger. You associate a trigger with a table or columns within a table. Triggers are useful for automatically maintaining business rules in a database.

You can also install Java classes into the database. Java classes provide a powerful way of building logic into your database. See [“Tutorial: Using Java in the database” \[SQL Anywhere Server - Programming\]](#).

See [“Stored procedures, triggers, batches, and user-defined functions” \[SQL Anywhere Server - SQL Usage\]](#).

- **Users and groups** Each user of a database has a user ID and password. You can set privileges for each user so that confidential information is kept private and users are prevented from making unauthorized changes. Users can be assigned to groups to make the administration of privileges easier.

See [“User security \(roles and privileges\)” \[SQL Anywhere Server - Database Administration\]](#).

In addition to these common database objects, SQL Anywhere also provides advanced features:

- Events
- Domains
- Publications
- Web services
- Remote data access
- Maintenance plans
- Spatial Reference systems

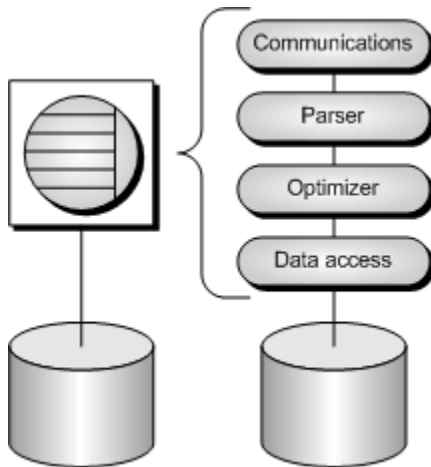
## Inside SQL Anywhere

While you never need to deal with the internals of the database server, a glimpse behind the scenes can help you understand how the database server and the database interact.

### Inside the SQL Anywhere database server

The SQL Anywhere database server has an internal structure that allows many requests to be handled efficiently.

- A communications layer handles the exchange of data with client applications. This layer receives requests from client applications, and returns results. The timing of these actions is governed by a negotiation between the client and the server to make sure that the network traffic is kept to a minimum, but that the data is made available as soon as possible on the client side.
- The parser checks each SQL statement sent to the database server, and transforms it into an internal form for processing.
- If the request is a query, an update, or delete statement, there can be many different ways of accessing the data, which may take significantly different times. The optimizer selects the best method of getting the required data quickly.
- The lowest level of the database server is concerned with reading and writing data from the disk, caching data in memory to avoid unnecessary disk access, and balancing the demands of different users.



### SQL Anywhere database servers

SQL Anywhere provides two versions of the database server: the **personal server** (`dbeng16`) and the **network server** (`dbsrv16`). A database created with a personal database server can be used with a network database server and vice versa. The request-processing engine is identical in both the personal and network servers and each one supports exactly the same SQL language and many of the same database features. However, there are a few differences between the two servers.

- **The personal database server** This executable is provided for single-user, same-computer use—for example, as an embedded database server—and does not support client/server communications across a network.

On Windows operating systems, except Windows Mobile, the name of the personal server executable is `dbeng16.exe`. On Unix operating systems its name is `dbeng16`. The personal server is not supported on Windows Mobile.

- **The network database server** This executable supports client/server communications across a network, and is intended for multi-user use.

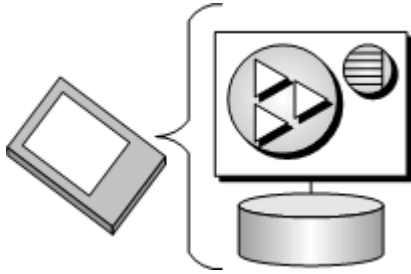
On Windows operating systems, including Windows Mobile, the name of the network server executable is `dbsrv16.exe`. On Linux and Unix operating systems, the name is `dbsrv16`.

### See also

- [“Differences between the network and personal servers” \[SQL Anywhere Server - Database Administration\]](#)

## Inside UltraLite

To provide a database application for a small device such as a handheld organizer, you may want to use UltraLite. In UltraLite, the functions performed by the server are typically placed in a runtime library. The runtime library is combined with the application to become part of the same process. So, there is a one-to-one relationship between the database and the application.



For deployments that require multiple applications to connect concurrently to one database on the same device, the library must exist as a separate process. In these cases, the UltraLite database engine is used.

### Other features

- UltraLite has built-in MobiLink synchronization technology so that your application is linked into the information network.

For information about integrating UltraLite and MobiLink, see [“UltraLite clients” \[UltraLite - Database Management and Reference\]](#).

- UltraLite supports many operating systems. See [“UltraLite overview” \[UltraLite - Database Management and Reference\]](#).

## Database files

The following sections provide an overview of the type of files, including database, transaction, and temporary files, that comprise a database. Differences between the implementation of these files in SQL Anywhere and UltraLite are also discussed.

### SQL Anywhere database files

All the information in a SQL Anywhere database is usually stored in a single database file, which can be copied from one computer to another. It is possible to have a database made up of several files, but this is generally only required for very large databases.

In addition to the database file, SQL Anywhere uses two other files when running a database: the transaction log and the temporary file.

- **The database file** Internally, the database file is composed of pages: fixed size areas of disk. The data access layer reads and writes data one page at a time. Many pages hold the data that is in the database tables, but other pages hold index information, information about the distribution of data within the database, and so on.
- **The transaction log** The transaction log is a separate file that contains a record of all the operations performed on the database. Normally, the transaction log has the same name as the database file, except that it ends with the suffix *.log* instead of *.db*. It has three important functions:
  - **Record operations on your data to enable recovery** You can recreate your database from a backup together with the transaction log if the database file is damaged.

- **Improve performance** By writing information to the transaction log, the database server can safely process your statements without writing to the database file as frequently.
- **Enable database replication** SQL Remote and MobiLink synchronization use the transaction log to synchronize changes to your other databases.
- **The temporary file** The temporary file is created when the database server starts, and is erased when the database server stops. As its name suggests, the temporary file is used to hold temporary information while the database server is running. The temporary file does not hold information that needs to be kept between sessions.

The UltraLite temporary file is stored in the same directory as the database file.

See “[TMP, TMPDIR, and TEMP environment variables](#)” [*SQL Anywhere Server - Database Administration*].

### UltraLite database files

UltraLite databases contain the same features as described above with the following exceptions:

- UltraLite database files don't contain information about the distribution of data within the database.
- UltraLite keeps track of its transactions internally, not in a separate log file.
- The UltraLite temporary file is stored in the same directory as the database file.

See “[Transaction and row state management](#)” [*UltraLite - Database Management and Reference*].

## SQL Anywhere and UltraLite target platforms

SQL Anywhere and UltraLite address data storage and data access needs from large enterprise database sources to small, mobile databases. When designing an application, you need to choose the database that is the right fit.

- If your target platform is Unix or Mac OS X, you must use a SQL Anywhere database.
- If your target platform is Embedded Linux, Android, BlackBerry, or iPhone or iPad, you must use an UltraLite database.
- If the target platform is Windows 7, Windows Vista, Windows XP, or Linux, both SQL Anywhere and UltraLite are available. SQL Anywhere is often preferred because provides a fuller feature set and its additional memory requirements are rarely an issue.
- If the target platform is Windows Mobile, such as on a Pocket PC or smartphone, you need to consider memory constraints, and possibly the tasks your application needs to perform. On Windows Mobile, SQL Anywhere requires approximately 6 MB of memory plus another 2 MB for the synchronization component, whereas UltraLite requires less than 1 MB and has synchronization built in. But, while UltraLite is considerably smaller, it does not offer the same support as SQL Anywhere for such things as complex queries, events, procedures, triggers, views, and so on.

For more information about the differences between the core database solution (SQL Anywhere) and the UltraLite database solution, see [“UltraLite, UltraLite Java edition, and SQL Anywhere feature comparisons”](#) [*UltraLite - Database Management and Reference*].

## Database scenarios

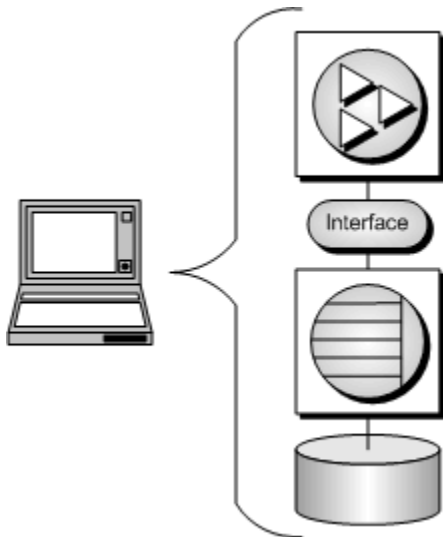
Database applications can connect to a database server located on the same computer as the application. Network database servers can connect to a different computer. In addition, with SQL Anywhere you can build distributed databases for remote office and mobile applications, with physically distinct databases on different computers sharing data.

## Desktop applications and embedded databases

You can use SQL Anywhere to build a complete application and database on a single computer. In the simplest arrangement, this is a **standalone application** or **personal application**: it is self-contained with no connection to other databases. In this case, the database server and the database can be started by the client application, and it is common to refer to the database as an **embedded database**. As far as the end user is concerned, the database is a part of the application.

Many relational database management systems require experienced staff for administration. A characteristic of SQL Anywhere databases is the ability to run entirely without administration.

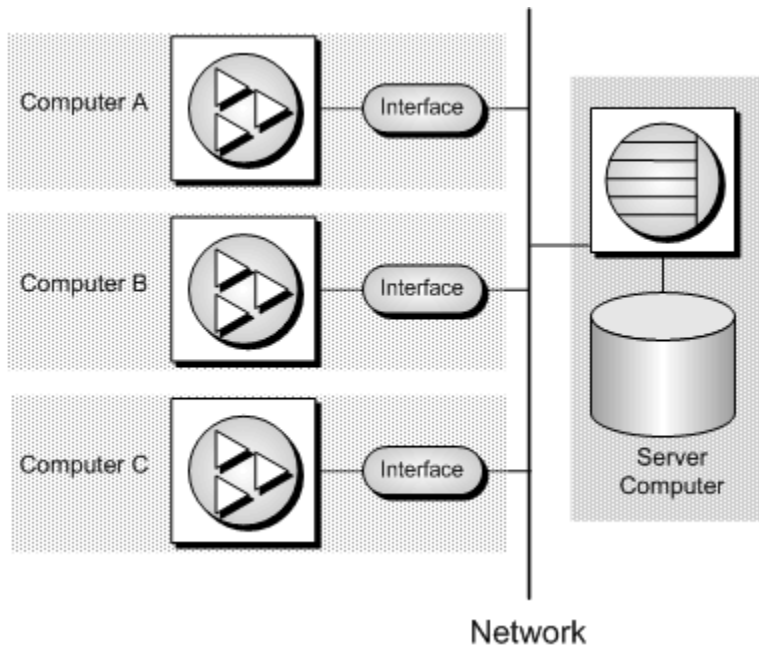
The SQL Anywhere personal database server is generally used for embedded applications. Embedded applications have the following architecture, with a client application connecting through a programming interface to a database server running on the same computer:





## Client/server applications

You can use SQL Anywhere to build an installation with many applications running on different computers, connected over a network to a single network database server running on a separate computer. This is a **client/server** or **multi-user database** environment, and has the following architecture. The interface library is located on each client computer.



In this case, the database server is the SQL Anywhere network database server, which supports network communications over TCP/IP.

For a client application to work in a client/server environment, you need to specify additional connection parameters, typically the HOST connection parameter and optionally the ServerName parameter.

### See also

- “Types of deployment” [[SQL Anywhere Server - Programming](#)]
- “Database connections” [[SQL Anywhere Server - Database Administration](#)]

## Multi-tier computing architecture

In multi-tier computing, application logic is held in an application server, such as Sybase EAServer, WebLogic, or WebSphere, which sits between the database server and the client applications. In many situations, a single application server can access multiple databases in addition to non-relational data stores. In the Internet case, client applications are browser-based, and the application server is generally a web server extension. Many modern multi-tier applications use a service-oriented architecture (SOA) based on web services.

Sybase EAServer stores application logic in the form of components, and makes these components available to client applications. The components can be Sybase PowerBuilder components, Java beans, or COM components.

Application servers can also provide transaction logic to their client applications—guaranteeing that sets of operations are executed atomically across multiple databases. SQL Anywhere is well suited to multi-tier computing, and can participate in distributed transactions coordinated by Microsoft Distributed Transaction Coordinator (MSDTC). Both Sybase Enterprise Application Server and Microsoft Transaction Server use DTC to provide transaction services to their client applications.

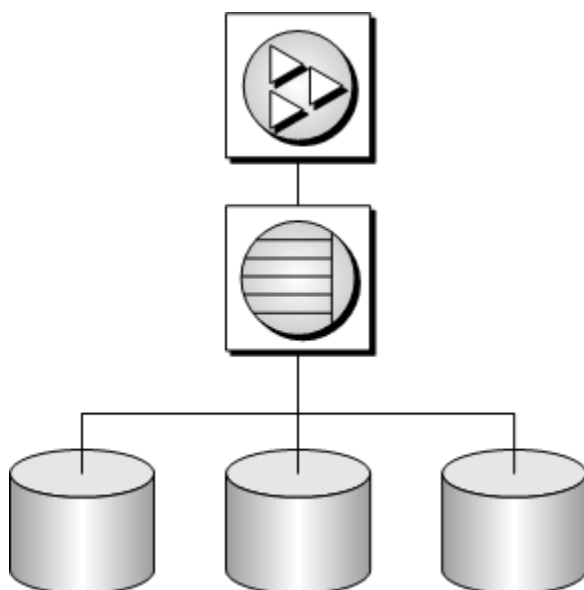
The built-in support for web services makes SQL Anywhere a good choice for many multi-tier or SOA applications.

**See also**

- [“Three-tier computing and distributed transactions” \[SQL Anywhere Server - Programming\]](#)

## Multiple databases running on a single database server

Both the SQL Anywhere personal database server and the network database server can manage many databases simultaneously. Each connection from an application must be to a single database, but an application can use separate connections to different databases, or a set of applications can work on different databases, all through the same database server.



Databases can be started when the database server is started, by connecting to a database using the DatabaseFile connection parameter, or by using the START DATABASE statement.

**See also**

- [“SQL Anywhere database server syntax” \[SQL Anywhere Server - Database Administration\]](#)
- [“DatabaseFile \(DBF\) connection parameter” \[SQL Anywhere Server - Database Administration\]](#)
- [“START DATABASE statement” \[SQL Anywhere Server - SQL Reference\]](#)

## Data access to other databases

You can access databases on multiple database servers, or even on the same server, using SQL Anywhere remote data access. The application is still connected to a single database, but by defining remote servers, you can use proxy tables that exist on the remote database as if they were in the database to which you are connected.

**See also**

- [“Remote data access” \[SQL Anywhere Server - SQL Usage\]](#)

## ETL features

Extract, Transform, and Load (ETL) is the process by which large amounts of data is extracted from disparate data sources and consolidated into a single database. In the extraction phase, data is parsed and evaluated for suitability. During transformation, data is manipulated to achieve the format required for storage. Some common transformations include the elimination of unnecessary columns, calculation of computed values, and translation of values such as dates into a common format so that the data can be consolidated. The data is then loaded into the database at a frequency and scope consistent with the organization's needs.

SQL Anywhere offers several features in support of ETL. For example:

- **OPENSTRING operation** Use the OPENSTRING operation in the FROM clause to transform and load data from client- and server-side data sources. See [“FROM clause” \[SQL Anywhere Server - SQL Reference\]](#).
- **OPENXML operator** Use the OPENXML string operator to extract data from XML documents. See [“OPENXML operator” \[SQL Anywhere Server - SQL Reference\]](#).
- **MERGE statement** Use the MERGE statement to merge data from different source objects. See [“MERGE statement” \[SQL Anywhere Server - SQL Reference\]](#).
- **Proxy tables** Use proxy tables to access objects such as tables, views, and materialized views in a remote database. See [“Proxy tables” \[SQL Anywhere Server - SQL Usage\]](#).
- **System procedure calls in the FROM clause** You can use various system procedures in the FROM clause of a query to extract and transform data for loading. For a list of system procedures offered in SQL Anywhere, see [“System procedures” \[SQL Anywhere Server - SQL Reference\]](#).

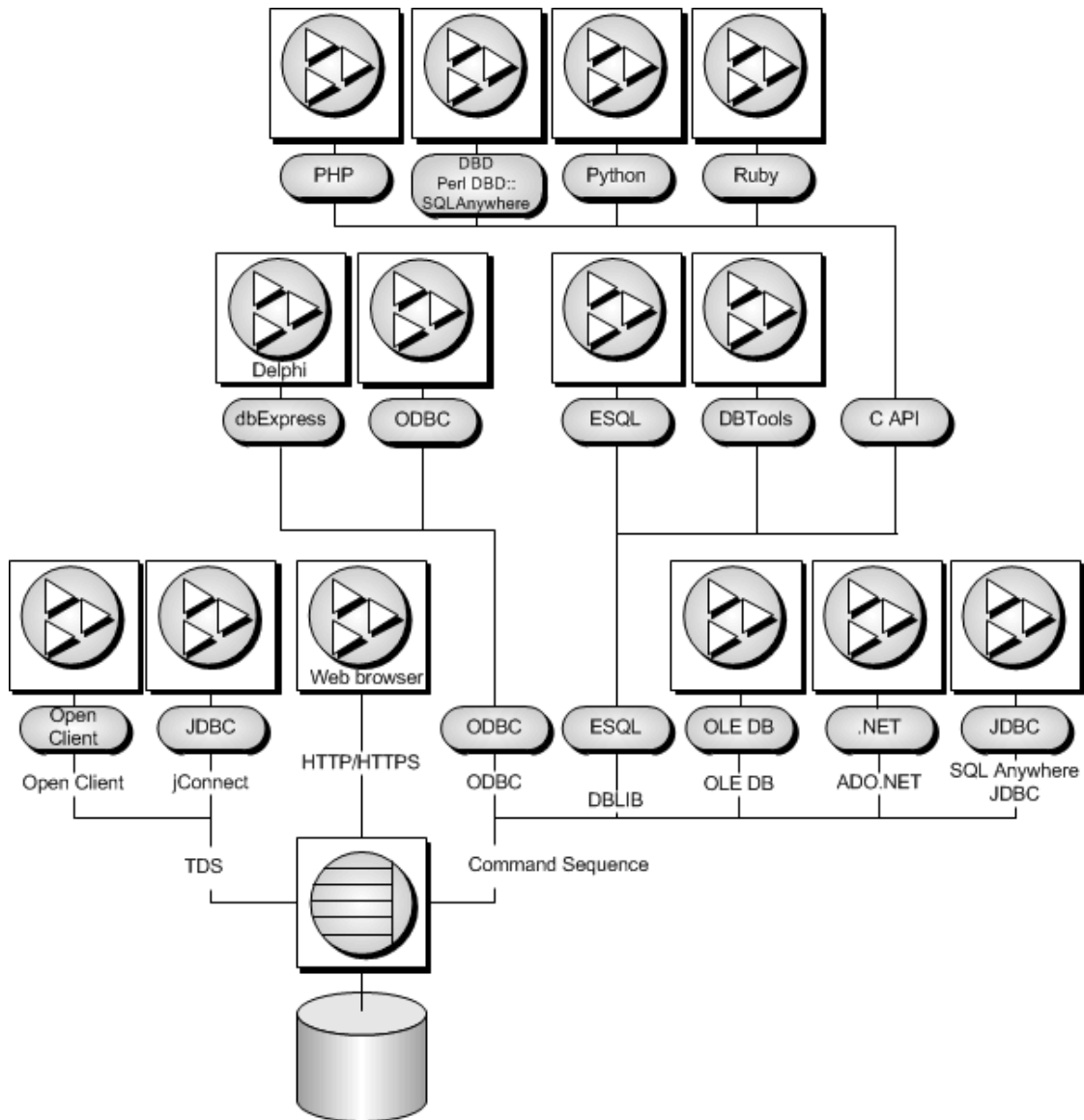
## Programming interfaces

SQL Anywhere supports a wide set of data access programming interfaces to provide flexibility in the kinds of applications and application development environments you can use.

For an overview of database application architecture, see [“Database scenarios” on page 24](#).

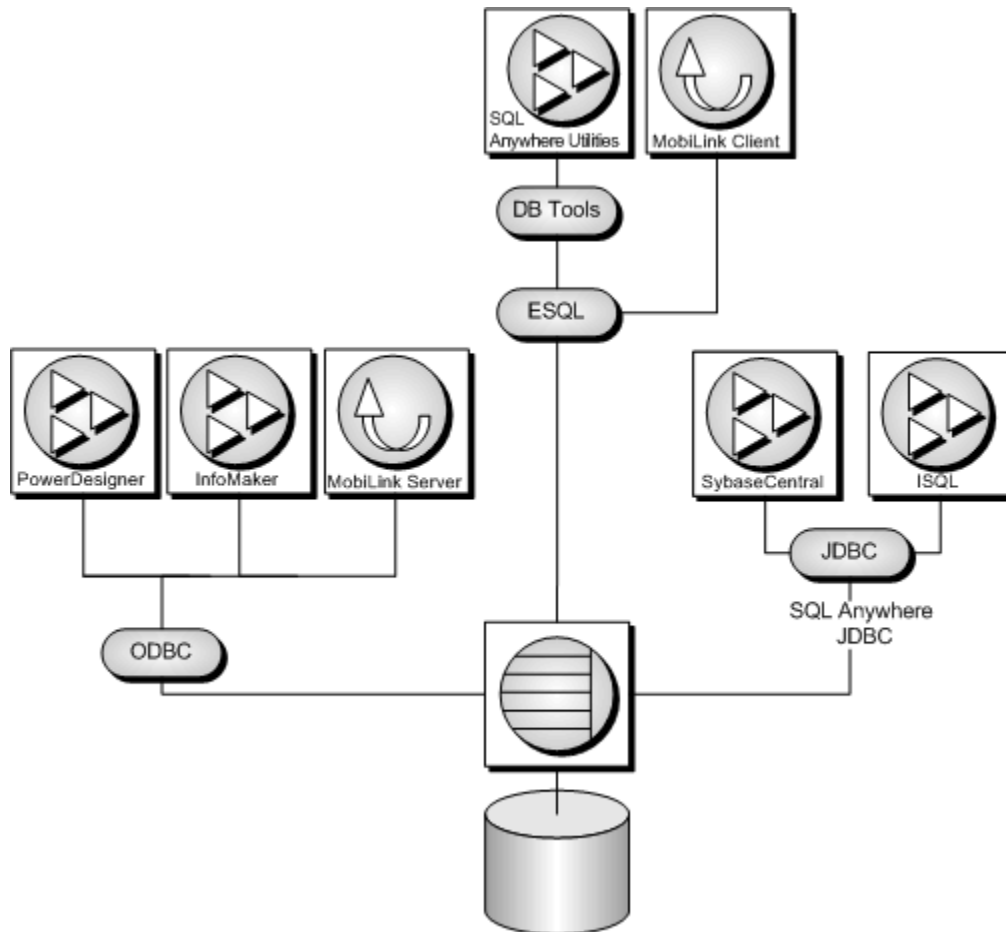
### Supported SQL Anywhere programming interfaces and protocols

The following diagram displays the supported interfaces, and the interface libraries used. The name of the interface library and the interface are usually the same.



## SQL Anywhere applications

The applications supplied with SQL Anywhere use several of these interfaces:



## SQL Anywhere programming interfaces

For specific details about the SQL Anywhere programming interfaces, see the list below:

- “SQL Anywhere .NET support” [[SQL Anywhere Server - Programming](#)]
- “ODBC support” [[SQL Anywhere Server - Programming](#)]
- “OLE DB and ADO development” [[SQL Anywhere Server - Programming](#)]
- “Embedded SQL” [[SQL Anywhere Server - Programming](#)]
- “JDBC support” [[SQL Anywhere Server - Programming](#)]
- “Sybase Open Client support” [[SQL Anywhere Server - Programming](#)]
- “SQL Anywhere C API support” [[SQL Anywhere Server - Programming](#)]
- “Perl DBI support” [[SQL Anywhere Server - Programming](#)]
- “SQL Anywhere PHP extension” [[SQL Anywhere Server - Programming](#)]
- “Python support” [[SQL Anywhere Server - Programming](#)]
- “SQL Anywhere Ruby API support” [[SQL Anywhere Server - Programming](#)]

## UltraLite programming interfaces

UltraLite also provides you with a choice of programming interfaces for straightforward access to data.

For more information about UltraLite programming interfaces, see [“Benefits of UltraLite APIs for Windows Mobile” \[UltraLite - Database Management and Reference\]](#).

## Interface library communication protocols

Each interface library communicates with the database server using a **communication protocol**. SQL Anywhere supports two communication protocols, **Command Sequence** and **Tabular Data Stream (TDS)**. These protocols are internal, and for most purposes it does not matter which one you are using. Your choice of development environment will be governed by your available tools.

The major differences are visible when connecting to the database. Command Sequence applications and TDS applications use different methods to identify a database and database server, and so connection parameters are different.

- **Command Sequence** This protocol is used by SQL Anywhere, the SQL Anywhere JDBC driver, and the embedded SQL, ODBC, OLE DB, and ADO.NET APIs.
- **TDS** This protocol is used by Sybase Adaptive Server Enterprise, the jConnect JDBC driver, and Sybase Open Client applications. For more information about TDS, see [“SQL Anywhere as an Open Server” \[SQL Anywhere Server - Database Administration\]](#).

---

# Overview of data exchange technologies

SQL Anywhere offers a wide range of options for exchanging data with existing enterprise systems and mobile devices, including such tools as:

- **MobiLink—synchronization** MobiLink offers session-based, bi-directional synchronization. It is ideal for exchanging data between a central database and many remote UltraLite or SQL Anywhere databases, or between a central, non-relational data source and many remote UltraLite or SQL Anywhere databases.

During a MobiLink synchronization, the remote database uploads changes that were made to it since the previous synchronization with the MobiLink server. On receiving this data, the MobiLink server updates the central database and then downloads changes from the central database to the remote database. It also ensures the transactional integrity of the databases in the event a connection between them is lost, and provides mechanisms for the resolution of data change conflicts.

MobiLink file transfer functionality lets you transfer files to remote applications on the same connection you use to synchronize data, which is useful when populating new remote databases or upgrading software.

In addition, MobiLink provides direct row handling for synchronizing remote data with any central data source. The data sources to which you can synchronize can include an application, web server, web service, application server, text file, spreadsheet, non-relational database, or an RDBMS that is not supported as a consolidated database.

- **SQL Remote—replication** SQL Remote is a data-replication technology designed for two-way synchronization between a consolidated database and large numbers of remote databases, typically including many mobile databases.

SQL Remote uses a store-and-forward architecture to synchronize data using a file or message transfer mechanism such as FTP or email.

SQL Remote preserves transactional integrity, making it ideal for many business applications, particularly those that operate in environments where connections are unreliable. Furthermore, memory and disk space requirements are minimal for all components of the replication system.

## See also

- “SQL Remote systems” [[SQL Remote](#)]

## Synchronization technology comparison

Data exchange technologies include synchronization, replication, messaging, and mobile web service technologies.

Data **synchronization** is the sharing of data among physically distinct databases. When an application modifies shared data at any one database, the changes are propagated to other databases in the synchronization system. Changes can be propagated by various means and through a variety of channels, allowing flexible application architecture while preserving data integrity.

SQL Anywhere offers two synchronization technologies:

- **MobiLink** is a session-based technology intended for the one- or two-way synchronization of data between a central, consolidated database and a large number of remote databases. It supports a variety of consolidated database servers, and provides an API for synchronizing with virtually any other data source. Administration and resource requirements at the remote sites are minimal, making MobiLink well suited to a variety of mobile applications. At the end of each synchronization session, the databases are consistent.
- **SQL Remote** is a message-based technology intended for the two-way replication of database transactions. It is designed for two-way replication involving a consolidated data server and large number of remote databases. Administration and resource requirements at the remote sites are minimal, making SQL Remote well suited to mobile databases.

The following table summarizes the characteristics of MobiLink and SQL Remote.

Synchronization technology	Number of databases	Connection	Frequency	Consolidated database types
MobiLink	Large	Occasional	Medium	Many options
SQL Remote	Large	Occasional	Low	SQL Anywhere

## MobiLink characteristics

MobiLink is designed for synchronization systems with the following requirements:

- **Large numbers of remote databases** MobiLink is designed to support large numbers of remote databases. It can handle tens of thousands of simultaneous synchronizations.
- **Occasionally connected** MobiLink supports databases that are occasionally connected or indirectly connected to the network on which the server is running.
- **Consolidated databases supported** MobiLink supports virtually any type of data store as the central data source. The remote data stores must be SQL Anywhere or UltraLite databases. The schema of the remote sites can be different from that of the consolidated database because you control the synchronization process by writing scripts.
- **Flexible synchronization schedule** Applications can connect and synchronize at intervals of seconds, minutes, hours, or days.

## SQL Remote characteristics

SQL Remote is designed for synchronization systems with the following requirements:

- **Large numbers of remote databases** SQL Remote is designed to support a large number of remote databases. It can support thousands of remote databases in a single installation because the messages for many remote sites can be prepared simultaneously.



- **Occasionally connected** SQL Remote supports databases that are occasionally connected or indirectly connected to the network.
- **Low to high latency** High latency means a long lag time between data being entered at one database and being replicated to each database in the system. With SQL Remote, replication messages can be sent at periods of seconds, minutes, hours, or days.
- **Low to moderate volume** As replication messages are delivered occasionally, a high transaction volume at each remote site can lead to a very large volume of messages. SQL Remote is best suited to systems with a relatively low volume of replicated data per remote database. However, at the consolidated site, SQL Remote can prepare messages efficiently by preparing messages for multiple sites simultaneously.
- **Homogeneous databases** SQL Remote supports SQL Anywhere databases. Each database in the system must have a similar schema.

## Benefits of data synchronization

### Data availability

One of the key benefits of a data synchronization system is that data is made available locally, rather than through potentially expensive, less reliable, and slow connections to a single central database. Data is accessible locally even in the absence of any connection to a central database, so you are not cut off from data in the event of a failure of a network connection.

### Response time

Synchronization improves response times for data requests for two reasons. Retrieval rates are faster because requests are processed on a local server, without accessing a wide area network. Also, local processing offloads work from a central database server so that competition for processor time is decreased.

## Challenges for synchronization technologies

Any synchronization technology must address several challenges that arise as a result of the increased flexibility permitted by synchronization.

### Transactional integrity

One of the challenges of any synchronization system is to ensure that each database always retains transactional integrity.

SQL Remote replicates portions of the transaction log in such a way that transactions are maintained during synchronization: either a whole transaction is replicated, or none of it is replicated. This ensures transactional integrity at each database in the system.

In MobiLink, you can also choose to replicate each transaction, but by default MobiLink coalesces multiple transactions on the remote database and applies them in a single transaction. This generally results in more efficient uploads. In both cases, MobiLink maintains transactional integrity.

## Data consistency

Another challenge to synchronization systems is to maintain data consistency throughout the system. Synchronization systems maintain a **loose consistency** in the system as a whole: that is, all changes are replicated to each site over time in a consistent manner, but different sites may have different copies of data at any instant.

## See also

- [“Synchronization techniques”](#) [*MobiLink - Server Administration*]

## Consolidated and remote databases

Both MobiLink and SQL Remote provide data synchronization between a central database and a set of remote databases.

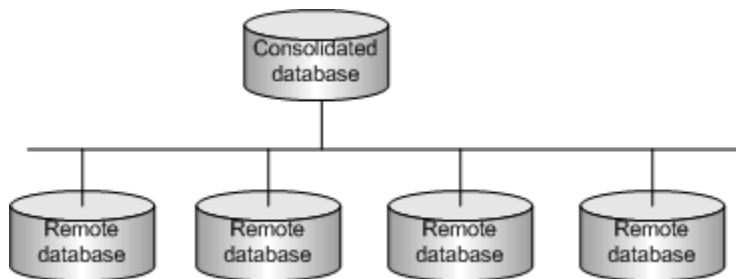
In MobiLink, the **consolidated database** is one of several supported RDBMSs. The consolidated database, which typically resides on a corporate server, tracks synchronization information and optionally contains the data to be replicated. Other central data may be stored in any other format, such as a non-relational database, web service, or text file.

MobiLink also provides direct row handling, which enables data synchronization to consolidated data sources other than relational databases including enterprise resource planning (ERP) systems or application servers.

In SQL Remote, all data that is to be synchronized is contained in a SQL Anywhere consolidated database.

A **remote database** can run either at the same site as the consolidated database or at a physically distant site such as a handheld device. The remote database can share all or some of the data in the consolidated database.

The following figure displays a schematic illustration of a small synchronization system.



## Remote users

A typical synchronization system includes many remote databases. Each remote database contains a subset of the information in the central database. Each remote database is a physically separate database, usually on a separate computer or mobile device. All remote databases must stay consistent with the central database.

The entire synchronization system may be considered a single dispersed database, with the master copy of all shared data being kept at the central database.

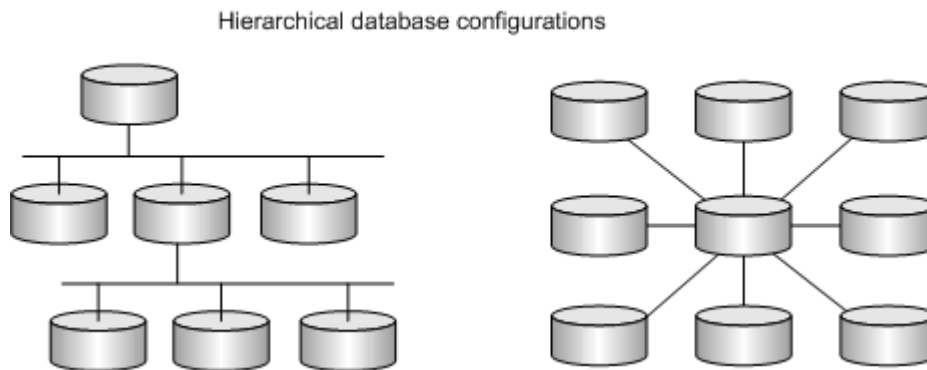
Each remote site that synchronizes with the central database is considered to be a remote user of the central database. In the case that a remote site is a multi-user server, the entire site is considered to be a single remote user of the central database.

## Hierarchical database configurations

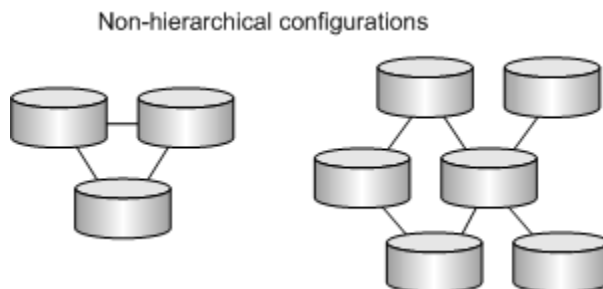
For databases in a **hierarchical configuration**, every database has a single parent database, except the consolidated database, which has no parent.

SQL Remote supports hierarchical configurations of databases; it does not support peer-to-peer synchronization or other non-hierarchical configurations. MobiLink is also normally used with a hierarchical configuration, but can also be used in other configurations.

For any two databases directly sharing data in a hierarchical configuration, one is always above or below the other in the hierarchy.



Databases in a non-hierarchical configuration do not have a well-defined notion of above or below.



In a MobiLink or SQL Remote system, each database contains all or a subset of the data replicated by the database above it in the hierarchy.

Remote databases can contain tables that are not present at the consolidated database as long as they are not involved in synchronization. SQL Remote requires that the table and column names in the remote databases match the ones in the consolidated database. In contrast, MobiLink allows data to be stored in different columns and tables in the remote databases than in the consolidated database, allowing greater flexibility.

## Propagation methods

When a transaction modifies shared data at any one database, the transaction or changes must be replicated to the other databases in the synchronization system. There are various means by which this task may be accomplished.

### Two-way synchronization

All SQL Anywhere synchronization technologies provide two-way synchronization. Changes made at the central database are propagated to remote databases. Changes made at remote databases are propagated to the central database, and to other remote databases. MobiLink allows upload-only, download-only, and two-way synchronization.

Both SQL Remote and MobiLink allow the same data to be changed simultaneously at multiple locations and both provide a means of resolving any conflicts.

## Session-based synchronization: MobiLink

In a **session-based** or **synchronous** synchronization scheme, synchronization occurs in real time over some sort of direct communications link. For example, the connection can be over a modem, network, or radio modem. Remote sites connect at intervals of seconds, minutes, hours, days, or weeks.

A session-based synchronization process is analogous to a telephone conversation in which all outstanding issues at both ends are resolved. The process follows a particular format. A MobiLink remote site begins by opening a connection to a MobiLink server and uploading a complete list of all the changes made to the remote database since the previous synchronization. Upon receiving this data, the server updates the central database, and then sends back all relevant changes. The remote site incorporates the entire set of changes, then sends back a confirmation and closes the connection.

## Message-based synchronization: SQL Remote

SQL Remote is an **asynchronous** synchronization scheme: it uses messages to exchange data between databases. Messages are typically files or specially formatted email messages. A **message agent**, attached to each database, sends messages regarding changes to its own data. The same agent also receives messages from one or more other databases and modifies the database according to the contents of the received messages.

In message-based communications, each message carries its destination address and other control information so that no direct connection is necessary between applications exchanging information. For

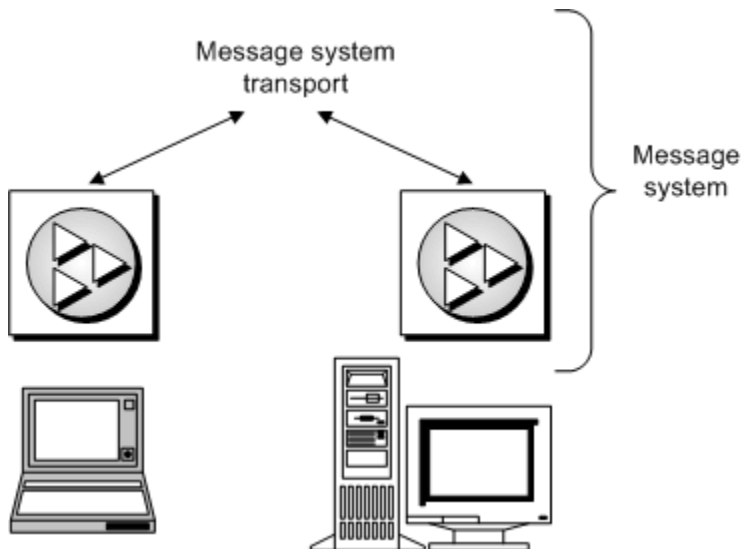
example, an email message contains the destination address; there is no direct connection between the sending server and the recipient.

### Message services use store-and-forward methods

Just as session-based client/server applications rely on network communication protocol stacks, such as TCP/IP, so message-based applications rely on message services such as Internet Simple Mail Transfer Protocol (SMTP), or a simple shared file link.

Message services use **store-and-forward** methods to get each message to its destination: for example, email systems store messages until the recipient opens their mail folder to read their mail, then the email system forwards the message.

Building a synchronization system on top of a message system means that a message-based synchronization system, such as SQL Remote, does not need to implement a store-and-forward system to get messages to their destination. Just as session-based client/server applications do not implement their own protocol stacks to pass information between client and server, so SQL Remote uses existing message systems to pass the messages.



### Guaranteed delivery

To work reliably, a message-based synchronization system must both guarantee that all messages reach their destination and that the messages are applied in the same order that they are sent. SQL Remote incorporates a protocol to guarantee application of synchronization updates in the correct order.

## Synchronization technology considerations

Each SQL Anywhere synchronization technology lends itself to particular applications. The following descriptions differentiate between the technologies and let you select the one best suited to your needs.

You should consider which of the following elements are important in your application:

### **Your consolidated database system**

In a typical synchronization environment, a large database serves as a central repository for information. Sometimes you can choose a database system that suits your needs. Other times, a central database already exists and you must adapt the synchronization system to work with it.

MobiLink can work with many popular database servers, including SQL Anywhere, Sybase Adaptive Server Enterprise, Oracle, Microsoft SQL Server, and IBM DB2. Using the MobiLink server APIs for .NET and Java, you can synchronize with any data source, including application servers, web servers, text files, and other database products.

In a SQL Remote system, the central database must be SQL Anywhere.

### **Your remote database system**

SQL Anywhere synchronization technologies also differ in the types of remote databases that they can support.

MobiLink supports SQL Anywhere and UltraLite as remote databases.

SQL Remote supports SQL Anywhere remote databases.

### **Network characteristics**

MobiLink and SQL Remote are both well suited to occasionally-connected environments, where remote sites must operate for hours or days in isolation, although more frequent synchronization is possible whenever a network connection is available.

MobiLink is session-based. A real-time connection is required during synchronization. If this connection is interrupted before synchronization is complete, the process does not complete until the next synchronization. In contrast, SQL Remote relays information via messages, which can be sent or received asynchronously. These messages may take the form of files on a hard disk, or email messages. These messages can be processed whenever they are received, allowing synchronization to occur incrementally.

### **Frequency of synchronization**

In some situations, it may be important that your information is replicated immediately. In others, synchronization once or twice a day may suffice. In fact, more frequent synchronization may be impossible when no network connection is available.

Both MobiLink and SQL Remote are primarily intended for situations where synchronization occurs infrequently, such as every few hours or days, but both can be used to synchronize as frequently as every few seconds.

### **The number of remote sites**

MobiLink and SQL Remote both work well with a very large number of remote users. MobiLink scalability is limited only by the scalability of the consolidated database management system. The SQL Remote message-based design allows a typical installation to handle thousands of remote users.

There is no hard limit on the maximum number of remote sites with any of these systems. The actual number depends on the amount of information replicated, the frequency of synchronization, and the design of your application.

### **Transaction ordering**

By default, MobiLink works by grouping the results of multiple transactions on the remote database into one set of changes to be applied to the consolidated database. Alternatively, you can choose to preserve the order of transactions and upload them separately. In both cases, synchronization always occurs at a transaction boundary, and so referential integrity is preserved. Uncommitted data is never synchronized, and so data integrity is preserved.

SQL Remote replicates data by scanning the transaction log and preparing messages, as appropriate, for each transaction. It orders these messages and sends them to the remote or consolidated site. When processing receives the messages, SQL Remote always processes them in the same order as they were applied to the other database. When necessary, it automatically delays processing a message until all earlier messages have been applied.

### **Achieving data consistency at a particular time**

Immediately following each MobiLink synchronization session, the data in the two databases is consistent. The ability to guarantee the consistency of the data at a remote site at a particular point in time is an advantage of MobiLink session-based synchronization. For example, if it is important that the data at a remote site accurately reflect the data in the consolidated database at a particular time, such as 10 o'clock in the morning, this objective can be achieved by synchronizing before this time. As long as the synchronization completes successfully, the currency of the data at the remote site is assured.

When changes to the data are replicated through an exchange of messages, it is difficult to guarantee that the data in a particular remote site is completely consistent with the data in the consolidated site at any particular point in time. For example, sometimes a message is lost in transit. SQL Remote automatically recognizes this fault and resends the message, but such interruptions can cause unexpected delays.



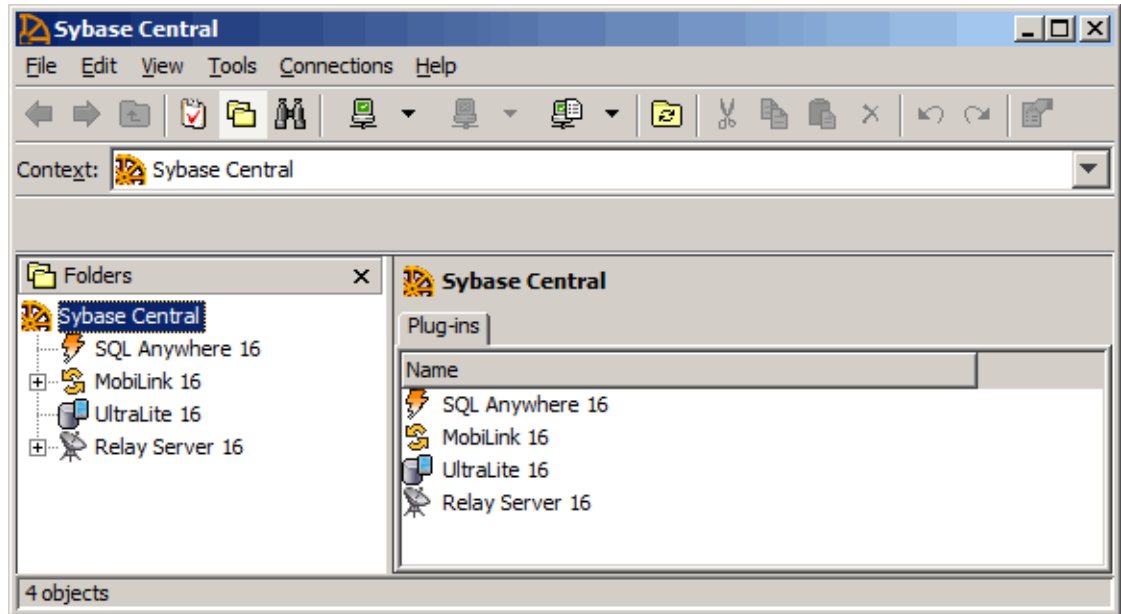


---

# Design and management tools

The following describes the design and management tools included with SQL Anywhere.

- **Sybase Central—centralized control and administration** Sybase Central is an integrated database administration and development tool that provides access to database settings, properties, and utilities in a graphical user interface. Via plug-ins, Sybase Central can be used to manage SQL Anywhere Server, UltraLite, MobiLink, Relay Server, and other Sybase products.



In addition to helping with routine tasks, Sybase Central also provides performance statistics, procedure profiling, stored procedure debugging, and the management of events and schedules, web services, and connection profiles. Sybase Central helps administer any tasks that are performed by sending SQL statements to the database server, or performed by SQL Anywhere utilities. See [“Sybase Central” \[SQL Anywhere Server - Database Administration\]](#).

A variety of Sybase Central tools is available to help you analyze and monitor the current performance of your SQL Anywhere database. These tools include procedure profiling, graphical plans, query execution, the performance monitor, request logging, and timing utilities. In addition, Sybase Central offers:

- **Support for spatial data** - Spatial data is data that describes the position, shape, and orientation of objects in a defined space. Spatial data in SQL Anywhere is represented as 2D geometries in the form of points, curves (line strings and strings of circular arcs), and polygons. See [“SQL Anywhere Server - Spatial Data Support”](#).
- **Application profiling using the Application Profiling Wizard** Use the **Application Profiling Wizard** in Sybase Central to automatically:

- Profile stored procedures, functions, triggers, and events
- Receive recommendations to help improve the performance of your database application
- Capture database activity while your application is running

See “[Application profiling](#)” [*SQL Anywhere Server - SQL Usage*].

- **Advanced application profiling in Application Profiling mode** Improve overall performance by using the **Database Tracing Wizard** and **Application Profiling** mode in Sybase Central to:

- Adjust cache size and indexes based on database performance counters
- Identify when deadlocks occur
- Look at locking activity
- Examine execution plans
- Trace each statement in an application for diagnosing and troubleshooting

See “[Diagnostic tracing](#)” [*SQL Anywhere Server - SQL Usage*].

- **Index selection and optimization using Index Consultant** The Index Consultant analyzes workloads and provides recommendations on how to select indexes to optimize performance. The Index Consultant can be run from either Sybase Central or Interactive SQL. See “[Index Consultant](#)” [*SQL Anywhere Server - SQL Usage*].

- **Interactive SQL—SQL query editor** Interactive SQL is a database utility designed to execute SQL statements and display database data. The built-in query editor and other tools, such as the graphical plan display, help you to analyze, troubleshoot, and optimize queries. See “[Interactive SQL](#)” [*SQL Anywhere Server - Database Administration*].
- **SQL Anywhere Monitor** The Monitor is a browser-based administration tool that provides you with information about the health and availability of SQL Anywhere databases, MobiLink servers, MobiLink server farms, and Relay Server farms. The Monitor provides constant data collection, email alert notifications, a browser-based interface, and the ability to monitor multiple databases, MobiLink servers, MobiLink server farms, and Relay Server Farms. See “[SQL Anywhere Monitor](#)” [*SQL Anywhere Server - Database Administration*].
- **MobiLink Monitor—synchronization monitoring** The MobiLink Monitor is a graphical administration tool that provides details about the performance of MobiLink synchronizations. The MobiLink Monitor collects details and statistical summaries about all synchronizations that occur, including start and end times, data volume uploaded and downloaded, successful completions, conflicts, and more. See “[MobiLink Profiler](#)” [*MobiLink - Server Administration*].
- **Relay Server** The Relay Server enables secure, load-balanced communication between mobile devices and back-end servers through a web server. Supported back-end servers include MobiLink, SQL Anywhere, Afaria, Mobile Office, Sybase Unwired Platform, and Unwired Server. See “[Relay Server](#)”.

- 
- **Utilities** SQL Anywhere includes various utilities for performing administration tasks such as backing up a database and performing synchronizations. Utilities are useful for including in batch files for repeated use. See:
    - “Database administration utilities” [*SQL Anywhere Server - Database Administration*]
    - “UltraLite utilities” [*UltraLite - Database Management and Reference*]
    - “MobiLink utilities” [*MobiLink - Server Administration*]
    - “SQL Remote utilities and options reference” [*SQL Remote*]
  
  - **SAP Sybase PowerDesigner Physical Data Model** SQL Anywhere includes Physical Data Model, a module of the powerful database design tool, SAP Sybase PowerDesigner. This module provides ways to generate and modify databases using a graphical representation of the database schema. You can optimize your database by customizing tables, columns, indexes, keys, views, physical storage, triggers, and stored procedures.

SAP Sybase PowerDesigner Physical Data Model includes comprehensive documentation, including video tutorials. For more information about SAP Sybase PowerDesigner, see <http://www.sybase.com/products/modelingmetadata/powerdesigner>.

For information about SQL Anywhere database design, see “Database creation” [*SQL Anywhere Server - Database Administration*].



---

# Sample databases

This section describes the schemas of the SQL Anywhere 16 sample databases. Experiment with the sample databases to learn more about SQL Anywhere 16.

## SQL Anywhere sample database

For consistency and simplicity, many of the examples throughout the documentation use the SQL Anywhere sample database, `%SQLANY%SAMP16%\demo.db`. By default this file is installed in the SQL Anywhere samples directory (for example, `C:\Users\Public\Documents\SQL Anywhere 16\Samples` on Windows 7).

The sample database uses the following default user ID and password:

- User ID = **DBA**
- Password = **sql** (passwords in SQL Anywhere are case sensitive)

### Caution

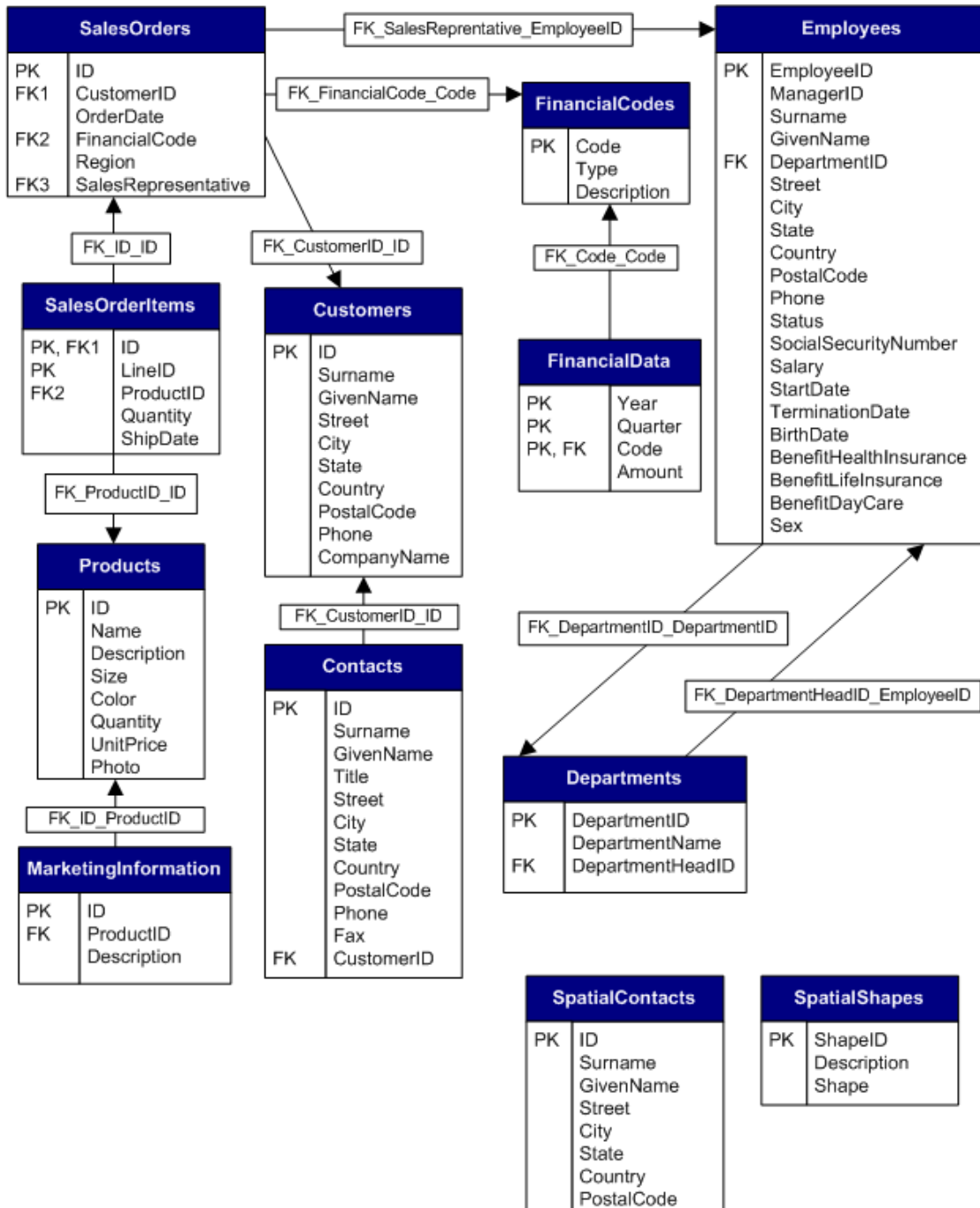
The sample database always has the same user ID and password; it is recommended that you change the DBA user ID and password to restrict access to the database. See [“Changing a password \(Sybase Central\)”](#) [*SQL Anywhere Server - Database Administration*].

The sample database uses the following ODBC data source: **SQL Anywhere 16 Demo**.

The sample database represents a small company that sells athletic clothing. It contains internal information about the company (employees, departments, and finances), product information, and sales information (sales orders, customers, and contacts). All data in the database is fictional.

The following figure displays the tables in the sample database and how they are related to each other. The boxes represent tables, and the arrows represent foreign key relationships.

For instructions on how to connect to `demo.db`, see [“Tutorial: Connecting to the sample database”](#) [*SQL Anywhere Server - Database Administration*].



### Roles in the sample database

The sample database contains one user-extended role (GROUPO) and three user-defined roles (READ\_ROLE, MODIFY\_ROLE, and EXEC\_ROLE).

GROUPO is a user that has been extended as a role. GROUPO is the owner of the tables described above. It is also the owner of a number of sample views, materialized views, and stored procedures that are defined in the sample database. Any new user that is granted the role GROUPO can reference these database objects without the owner qualifier. For example, they can refer to the Departments table without adding the owner prefix (GROUPO.Departments).

Any user that is granted the role READ\_ROLE has read access to the rows in the tables described above. Users with this role can perform a SELECT on any of these tables.

Any user that is granted the role MODIFY\_ROLE has write access to the rows in the tables described above. Users with this role can perform an INSERT, UPDATE, or DELETE on any of these tables.

Any user that is granted the role EXEC\_ROLE can execute any of the sample stored procedures that are defined in the sample database with the exception of ManageContacts (ShowContacts, ShowCustomers, ShowCustomerProducts, ShowProductInfo, ShowSalesOrders, ShowSalesOrderDetail, ShowSalesOrderItems, and debugger\_tutorial). The ManageContacts stored procedure is used to add, delete and update contacts in the Contacts table.

Any user with only roles READ\_ROLE, MODIFY\_ROLE, or EXEC\_ROLE cannot perform schema changes on the database. For example, they cannot add or remove columns from existing tables nor can they create or delete tables. Additional privileges are required for this.

In a large enterprise, there would be many such roles. For example, a Sales department might own the sales-related and customer-related tables, a Manufacturing or Purchasing department might own the products-related tables, a Human Resources department might own the employees-related and departments-related tables, and so on. Each of these departments would have their own collection of roles set up with various access rights to the tables they own. However, the sample database reflects a small enterprise and the use of roles has been deliberately kept simple.

The user "DBA" has all privileges for the sample database and can alter the database schema, create new users, add roles, and so on. This user has been granted the GROUPO user-extended role so that it can reference the sample tables and stored procedures without qualifying them with an owner ID.

Two additional users have been created to demonstrate the roles described above.

The user "UPDATER" (password "update") has been granted the roles READ\_ROLE, MODIFY\_ROLE, and EXEC\_ROLE. This user can perform a SELECT, INSERT, UPDATE, or DELETE on any of the sample tables. This user has EXECUTE privilege on all of the sample stored procedures. The GROUPO user-extended role is also granted so that this user can reference the sample tables and stored procedures without qualifying them with an owner ID.

The user "BROWSER" (password "browse") has been granted the roles READ\_ROLE, and EXEC\_ROLE. This user can perform a SELECT on any of the sample tables but cannot change the rows in the tables in any way. This user has EXECUTE privilege on all of the sample stored procedures with the exception of the ManageContacts stored procedure since this procedure is used to add, delete and update contacts in the Contacts table (the role MODIFY\_ROLE is required to execute ManageContacts). The GROUPO user-extended role is also granted so that this user can reference the sample tables and stored procedures without qualifying them with an owner ID.

## See also

- “Starting the Windows Mobile sample database” [[SQL Anywhere Server - Database Administration](#)]

## Recreate the sample database (*demo.db*)

Testing features and completing the tutorials in the SQL Anywhere documentation sometimes results in changes to the sample database that can prevent the successful completion of subsequent tutorials and tests. When this happens, you can restore the sample database to its original state. Alternatively, if you need to preserve the sample database in its current state, then you can recreate the sample database in its original state using a different name. Both methods are presented below.

- **Recreate the sample database (Windows)** Run the following command to erase *demo.db* and create a new copy of the sample database with objects and data:

```
newdemo "%SQLANYSAMP16%\demo.db"
```

When you are prompted, choose to erase any existing files.

- **Recreate the sample database (Unix)** Run the following command in the directory where the sample database is located to erase *demo.db* and create a new copy of the sample database with objects and data:

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

When you are prompted, choose to erase any existing files.

- **Create a copy of the sample database with a different name (Windows)** Run the following command to create a database called *mydemo.db* that contains objects and data.

If you do not specify a path, the database and transaction log are created in the current directory.

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

- **Create a copy of the sample database with a different name (Unix)** Run the following command to create a database called *mydemo.db* that contains objects and data.

If you do not specify a path, the database and transaction log are created in the current directory.

```
newdemo.sh path/mydemo.db
```

## The CustDB sample database application

The CustDB sample application is a useful tool for learning how to develop UltraLite and MobiLink applications. The sample database is a sales status database for a hardware supplier. It holds customer, product, and sales force information for the supplier.

There are two parts to the CustDB sample application:

- **UltraLite** For UltraLite, CustDB can be deployed on any device supported by UltraLite using any platform supported by UltraLite. You can see all the source code used to create the CustDB UltraLite



application and run the sample. The CustDB sample application is set up for MobiLink synchronization.

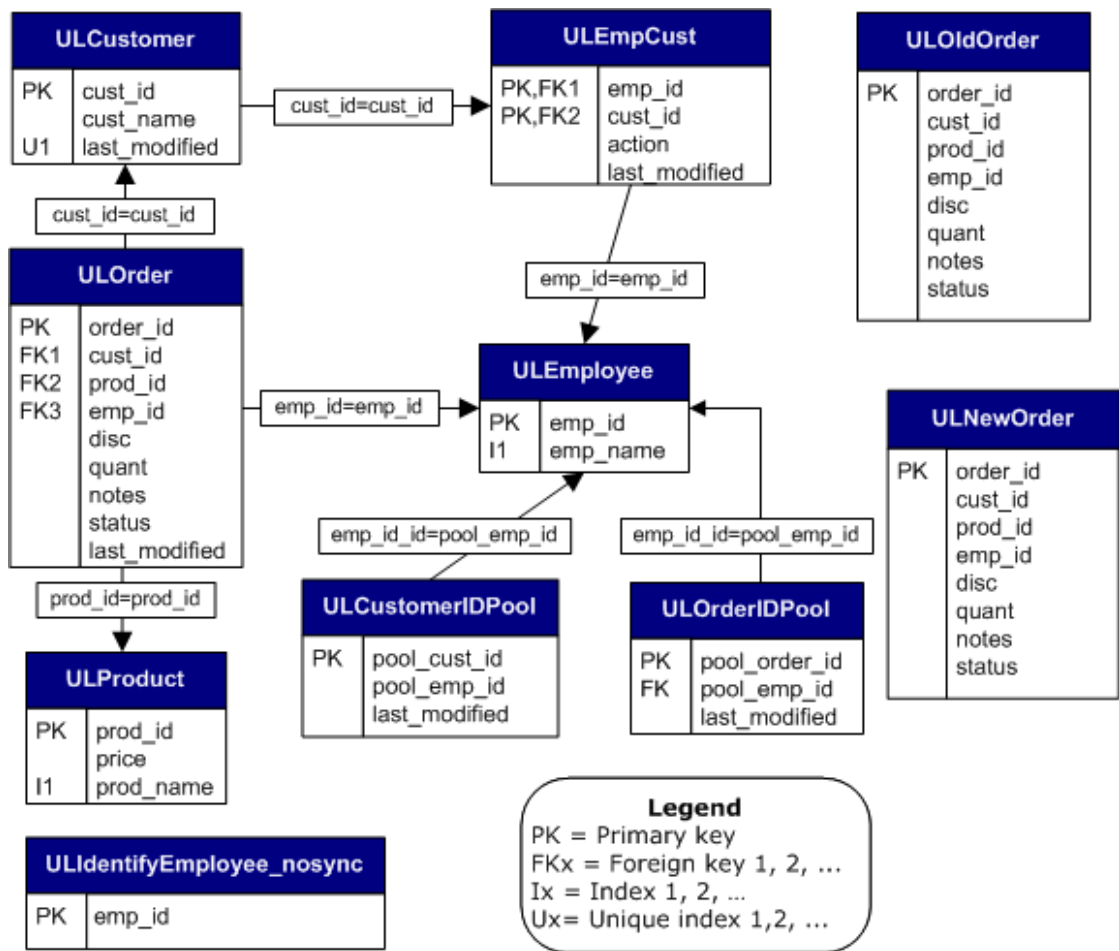
You can find the UltraLite CustDB sample application in `%SQLANYSAMPI6%\UltraLite\CustDB\`.

- MobiLink** If you are interested in exploring MobiLink features, there is a CustDB consolidated database that contains sample synchronization logic. You can use this consolidated database with the CustDB UltraLite sample remote database to run the sample synchronization system.

The MobiLink consolidated CustDB database is created by running script files against a supported relational database (SQL Anywhere, Adaptive Server Enterprise, Oracle, Microsoft SQL Server, or IBM DB2). These setup files are located in `%SQLANYSAMPI6%\MobiLink\CustDB\`.

The CustDB consolidated database uses the following ODBC data source: **SQL Anywhere 16 CustDB**.

The following diagram shows the tables in the CustDB database and how they relate to each other.



**See also**

- [“Tutorial: Building the UltraLite CustDB sample application”](#) [*UltraLite - Database Management and Reference*]
- [“CustDB sample for MobiLink”](#) [*MobiLink - Getting Started*]

---

# List of SQL Anywhere tutorials

This section contains a complete list of SQL Anywhere tutorials. You can try these tutorials to learn more about SQL Anywhere.

## SQL Anywhere tutorials

- [“Tutorial: Connecting to the sample database”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Creating a SQL Anywhere database”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Running Windows Mobile databases from Sybase Central”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Managing Windows Mobile databases with Interactive SQL”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Using Windows Mobile sample applications”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Monitoring resources with the Monitor”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Granting roles and privileges \(Sybase Central\)”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Granting roles and privileges \(SQL\)”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Creating a database mirroring system”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Creating a database mirroring system with multiple databases that share an arbiter server”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Moving the arbiter server”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Moving a partner server”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Creating a read-only scale-out system”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Using one server as both a copy node and an arbiter”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Converting a partner server to a copy node”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Adding a mirroring system to a read-only scale-out system”](#) [*SQL Anywhere Server - Database Administration*]
- [“Tutorial: Using the Simple code sample”](#) [*SQL Anywhere Server - Programming*]
- [“Tutorial: Using the Table Viewer code sample”](#) [*SQL Anywhere Server - Programming*]

- [“Tutorial: Developing a simple .NET database application with Visual Studio”](#) [*SQL Anywhere Server - Programming*]
- [“Tutorial: Using Java in the database”](#) [*SQL Anywhere Server - Programming*]
- [“Tutorial: Create a web server and access it from a web client”](#) [*SQL Anywhere Server - Programming*]
- [“Tutorial: Using SQL Anywhere to access a SOAP/DISH service”](#) [*SQL Anywhere Server - Programming*]
- [“Tutorial: Using Visual C# to access a SOAP/DISH web service”](#) [*SQL Anywhere Server - Programming*]
- [“Tutorial: Using JAX-WS to access a SOAP/DISH web service”](#) [*SQL Anywhere Server - Programming*]
- [“Tutorial: Experimenting with the spatial features”](#) [*SQL Anywhere Server - Spatial Data Support*]
- [“Tutorial: Creating a test database for the application profiling tutorials”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Diagnosing deadlocks”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Diagnosing slow statements”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Diagnosing index fragmentation”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Diagnosing table fragmentation”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Baselining with procedure profiling”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Performing a full text search on a GENERIC text index”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Performing a fuzzy full text search”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Performing a non-fuzzy full text search on an NGRAM text index”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Setting up the scenario for the isolation level tutorials”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Understanding dirty reads”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Understanding non-repeatable reads”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Understanding phantom rows”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Understanding phantom locks”](#) [*SQL Anywhere Server - SQL Usage*]
- [“Tutorial: Getting started with the debugger”](#) [*SQL Anywhere Server - SQL Usage*]

---

## MobiLink tutorials

- [“Tutorial: Introducing MobiLink”](#) [*MobiLink - Getting Started*]
- [“Tutorial: Using MobiLink with a SQL Anywhere consolidated database”](#) [*MobiLink - Getting Started*]
- [“Tutorial: Using MobiLink with an Oracle Database 10g”](#) [*MobiLink - Getting Started*]
- [“Tutorial: Using MobiLink with an Adaptive Server Enterprise consolidated database”](#) [*MobiLink - Getting Started*]
- [“Tutorial: Using Java or .NET for custom user authentication”](#) [*MobiLink - Getting Started*]
- [“Tutorial: Using direct row handling”](#) [*MobiLink - Getting Started*]
- [“Tutorial: Synchronizing with Microsoft Excel”](#) [*MobiLink - Getting Started*]
- [“Tutorial: Synchronizing with XML”](#) [*MobiLink - Getting Started*]
- [“Tutorial: Using central administration of remote databases”](#) [*MobiLink - Getting Started*]
- [“Tutorial: Changing a schema using the script version clause”](#) [*MobiLink - Getting Started*]
- [“Tutorial: Changing a schema using the ScriptVersion extended option”](#) [*MobiLink - Getting Started*]
- [“Tutorial: Simulating multiple MobiLink clients using the MobiLink Replay utility”](#) [*MobiLink - Getting Started*]
- [“Tutorial: Using scripted upload”](#) [*MobiLink - Client Administration*]
- [“Tutorial: Configuring server-initiated synchronization using light weight polling”](#) [*MobiLink - Server-Initiated Synchronization*]
- [“Tutorial: Configuring server-initiated synchronization using gateways”](#) [*MobiLink - Server-Initiated Synchronization*]

## UltraLite tutorials

- [“Tutorial: Building the UltraLite CustDB sample application”](#) [*UltraLite - Database Management and Reference*]
- [“Tutorial: Building an Android application”](#) [*UltraLite - Java Programming*]
- [“Tutorial: Building a BlackBerry application”](#) [*UltraLite - Java Programming*]
- [“Tutorial: Building a Windows application using the C++ API”](#) [*UltraLite - C and C++ Programming*]
- [“Tutorial: Building a Windows Mobile application using UltraLite.NET”](#) [*UltraLite - .NET Programming*]

## SQL Remote tutorials

- [“Tutorial: Creating a SQL Remote system”](#) [*SQL Remote*]

- “Tutorial: Setting up a replication system using the HTTP message system” [*SQL Remote*]
- “Tutorial: Setting up a replication system using the HTTP message system with the consolidated database as the message server” [*SQL Remote*]
- “Tutorial: Setting up a replication system using the HTTP message system and the consolidated database as the message server via Relay Server” [*SQL Remote*]

---

# Resources for getting started with SQL Anywhere 16

The following is a list of introductory materials and tutorials for the database and data exchange technologies.

Most of the tutorials throughout the documentation use the SQL Anywhere sample database (demo.db) or the CustDB sample database application (*custdb.db*).

- **SQL Anywhere Server** See [“Resources for getting started with SQL Anywhere Server”](#) on page 56.

This section walks you through several tutorials related to connecting to, and administering, SQL Anywhere Server databases.

- **UltraLite** See [“Tutorial: Building the UltraLite CustDB sample application”](#) [*UltraLite - Database Management and Reference*].

This section uses the UltraLite sample database (*custdb.db*) to demonstrate:

- Logging in and populating an UltraLite remote database
- Using a client application
- Synchronizing an UltraLite database with a consolidated database
- Browsing MobiLink synchronization scripts

- **MobiLink** See [“MobiLink - Getting Started”](#).

This book introduces MobiLink and contains numerous tutorials to guide you through the process of setting up a synchronization system that uses SQL Anywhere databases.

- **SQL Remote** See [“SQL Remote systems”](#) [*SQL Remote*].

This section introduces SQL Remote and describes how to set up a simple SQL Remote replication system using Sybase Central.

- **Relay Server** See [“Relay Server”](#).

This book explains how to set up and use the Relay Server.

## SAP Sybase PowerDesigner Physical Data Model

You can use SAP Sybase PowerDesigner Physical Data Model to create physical data models of a database and then convert them into SQL Anywhere databases.

See [“SAP Sybase PowerDesigner Physical Data Model”](#) on page 43.

### See also

- “SQL Anywhere sample database” on page 45
- “CustDB sample for MobiLink” [*MobiLink - Getting Started*]
- “Tutorial: Building the UltraLite CustDB sample application” [*UltraLite - Database Management and Reference*]

## Resources for getting started with SQL Anywhere Server

SQL Anywhere provides three methods to administer your system: Sybase Central, Interactive SQL, and command-line utilities. The choice of which tool to use is your preference. The majority of tasks are supported by all three methods.

The tutorials and introductory materials listed in this section use all three methods to create and manage SQL Anywhere databases.

For an overview of Sybase Central, Interactive SQL, and all other tools included with SQL Anywhere, see “[Design and management tools](#)” on page 41.

### Starting the database server

- “[Tutorial: Connecting to the sample database](#)” [*SQL Anywhere Server - Database Administration*]

This tutorial describes how to:

- Start a database server
  - Connect to a database server
  - Display the database server messages window
  - Stop the database server
- **Additional information** For a complete list of server connection options, see “[SQL Anywhere database server syntax](#)” [*SQL Anywhere Server - Database Administration*]. Alternatively, run the following command to see a list of all server connection options in a command prompt window:  

```
dbeng16 -?
```
  - **Further reading** For more examples of connecting to the SQL Anywhere sample database, connecting to an embedded database, and connecting across a network, see “[Database connections](#)” [*SQL Anywhere Server - Database Administration*].

### Creating a database

- “[Tutorial: Creating a SQL Anywhere database](#)” [*SQL Anywhere Server - Database Administration*]



This tutorial describes how to use Sybase Central to:

- Create a database file
- Connect to your database
- Create a table
- Create relationships between tables
- **Additional Information** For more information about database creation, see [“Database creation” \[SQL Anywhere Server - Database Administration\]](#).

### Managing databases

- [“Sybase Central” \[SQL Anywhere Server - Database Administration\]](#)

This section introduces Sybase Central and provides information about topics like:

- [“Connecting to the sample database \(Sybase Central\)” \[SQL Anywhere Server - Database Administration\]](#)
- [“Creating a connection profile” \[SQL Anywhere Server - Database Administration\]](#)
- [“Searching for database objects” \[SQL Anywhere Server - Database Administration\]](#)
- [“Viewing entity-relationship \(ER\) diagrams” \[SQL Anywhere Server - Database Administration\]](#)
- [“Database health and statistics” \[SQL Anywhere Server - Database Administration\]](#)

### Using SQL statements

- [“Interactive SQL” \[SQL Anywhere Server - Database Administration\]](#)

This section introduces Interactive SQL and provides information about topics like:

- [“Executing SQL statements \(Interactive SQL\)” \[SQL Anywhere Server - Database Administration\]](#)
- [“Statement canceling in Interactive SQL” \[SQL Anywhere Server - Database Administration\]](#)
- [“Looking up tables, columns, and procedures in Interactive SQL” \[SQL Anywhere Server - Database Administration\]](#)
- [“Interactive SQL history” \[SQL Anywhere Server - Database Administration\]](#)
- [“Logging statements in Interactive SQL” \[SQL Anywhere Server - Database Administration\]](#)
- [“SQL script files” \[SQL Anywhere Server - SQL Usage\]](#)

---

---

# Frequently asked questions - SQL Anywhere

## What is the default user ID and password for *demo.db* and newly created databases?

The default user ID for *demo.db* and newly created databases is **DBA** and the default password is **sql**. The majority of the examples in the SQL Anywhere documentation use DBA/sql. The user ID is case-insensitive and the password is case-sensitive. However, since this is a well known user ID and password, you should change the password before deploying the database, or create and use a different user as the DBA user.

For information about failed login attempts for DBA users, see “Automatic unlocking of user accounts” [[SQL Anywhere Server - Database Administration](#)].

## How do I recover a DBA password?

A lost or forgotten DBA password cannot be recovered. If you require assistance, contact Technical Support (<http://www.sybase.com/support>).

## How do I connect to a database?

The connection parameters required to connect to a database vary depending on the location of the application relative to the database server. The following connection scenarios assume that there is no communication encryption and there is only one database running on a database server.

In the following examples, you connect to the SQL Anywhere sample database, *%SQLANYAMP16%\demo.db* using Interactive SQL.

- **Example 1: The database server is not running, and the database file is installed on the same computer as the application** You must provide the correct user ID, password, connection and database file parameters for the database file. You may also need to specify the database server name. To connect to the database, run a command similar to the following:

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

- **Example 2: The application and the database server are running on the same computer** You must provide the correct user ID and password. It is also recommended that you provide the database server name to avoid connecting to the wrong database server. To connect to the database, run a command similar to the following:

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

- **Example 3: The application and database server are running on different computers** You must provide the correct user ID, password, database server name, and HOST connection parameters. To connect to the database, run a command similar to the following:

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

## How do I connect to a database on Unix?

See “Connecting to the sample database on Unix” [[SQL Anywhere Server - Database Administration](#)].

### How do I connect to a database on Mac OS X?

See [“Connecting to the sample database on Mac OS X” \[SQL Anywhere Server - Database Administration\]](#).

### How do I determine which version of SQL Anywhere was used to create a database?

Databases created with SQL Anywhere 9.0.1 or later include the SYSHISTORY system view. The rows in this view that reflect INIT and UPDATE operations give information about when, and with what version, the database was created and upgraded. See [“SYSHISTORY system view” \[SQL Anywhere Server - SQL Reference\]](#).

To determine the version without starting the database, see [“DBCreatedVersion method \[database tools\]” \[SQL Anywhere Server - Programming\]](#).

### How do I determine the names of the databases that are running on a server?

Execute the following statement to the names of the databases running on a server:

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

Execute the following statement to return a number of databases running on a server:

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

See [“sa\\_db\\_list system procedure” \[SQL Anywhere Server - SQL Reference\]](#) and [“DB\\_NAME function \[System\]” \[SQL Anywhere Server - SQL Reference\]](#).

### Can I install and run two different versions of SQL Anywhere on the same computer?

Yes, you can install multiple major versions of SQL Anywhere on the same computer. For example, SQL Anywhere 9.0.2, 10.0.1, 11.0.1, and 12.0.1 can all be installed and run independently.

However, you must use caution when starting a SQL Anywhere executable that has the same name in multiple versions (for example, dbisql or dbinit) to ensure that you are starting the correct version of the application. You can either specify the full absolute path using environment variables such as SQLANY16, or ensure the version of SQL Anywhere you want is specified first in your path.

See [“How to ensure that you are running the correct version of the utilities when you have multiple versions installed” \[SQL Anywhere 16 - Changes and Upgrading\]](#).

### Can I install and run two copies of the same major version of SQL Anywhere on the same computer?

**For SQL Anywhere 10 and earlier for Windows using the SQL Anywhere installer** Yes, you can install multiple copies of the same version of SQL Anywhere on the same computer. However, the SQL Anywhere installation program registers some drivers and components in the Windows Registry, and there is only one copy of the registry. As a result, the ODBC and OLE DB drivers from the most recent installation are used. Similarly, there is only one set of **Start** menu shortcuts. They point to the most recently installed copy of SQL Anywhere.

**For SQL Anywhere 11 and later for Windows using the SQL Anywhere installer** No, you can only install a single copy of SQL Anywhere 11 on a computer.

---

**For SQL Anywhere 12 and earlier for Linux or Unix using the SQL Anywhere installer** Yes, you can install multiple versions in different locations. However, on Linux, if you choose to install application menu items, each user can only have one set of application menu items. They point to the most recently installed copy of SQL Anywhere.

**For deployed embedded database applications** Yes, deployed embedded database applications that include SQL Anywhere can be deployed with other SQL Anywhere installations on the same computer.

On Windows operating systems, the ODBC and OLE DB driver names in the registry should include the name of the application in which they are embedded. For example, the ODBC driver name **SQL Anywhere 16** should be renamed *application-name SQL Anywhere 16*. See “[Database and application deployment](#)” [*SQL Anywhere Server - Programming*] and “[ODBC driver configuration](#)” [*SQL Anywhere Server - Programming*].

### **Is it better to have one database server for each database or should I run multiple databases on a single database server?**

You should run multiple databases on a single database server because this configuration optimizes the use of computer resources.

Having multiple database servers running on the same computer may result in a competition for resources, and (with dynamic cache resizing) this configuration may result in degraded or unpredictable performance. Degraded or unpredictable performance may be acceptable if you need to stop one database server for maintenance without affecting the others, or if you need to isolate errors to a single database server.

You should verify that you are correctly licensed for the installation option that you choose. For more information, see “[Multiple databases running on a single database server](#)” on page 26.

### **Why is the size of my database increasing, or not decreasing as expected?**

See “[Understanding unexpected changes in the database size](#)” [*SQL Anywhere Server - Database Administration*].

### **How do I create an effective backup and recovery plan?**

See the following:

- “[Backup and recovery plan design](#)” [*SQL Anywhere Server - Database Administration*]
- See the tech note titled [What Backup, Recovery, and Disaster Recovery Mean to Your SQL Anywhere Databases](#) (<http://www.sybase.com/detail?id=47877>)

### **What do I do when an assertion failure occurs?**

See the tech note titled “[I've Got An Assertion! What Should I Do?](#)” (<http://www.sybase.com/detail?id=1010805>).

### **How do I report a bug?**

To report low priority SQL Anywhere bugs, log in to <http://case-express.sybase.com>. Bugs reported using Case-Express are assigned a lower priority than cases opened through Technical Support.

For priority issues, open a support case with Technical Support (<http://www.sybase.com/support>).

### **How do I improve the performance of my application or database server?**

See the following documents:

- “Performance monitoring and diagnostic tools” [[SQL Anywhere Server - SQL Usage](#)]
- “Tips for improving performance” [[SQL Anywhere Server - SQL Usage](#)]
- 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>)

### **How do I upgrade my SQL Anywhere software?**

See “[SQL Anywhere Server upgrades](#)” [[SQL Anywhere 16 - Changes and Upgrading](#)].

### **Why is my application running slower after an upgrade?**

SQL Anywhere performs best when the database is created with the same major version as the database server. Rebuild the database if you are experiencing performance problems and no longer need to run the database on earlier versions of SQL Anywhere. See “[Database rebuilds](#)” [[SQL Anywhere Server - SQL Usage](#)] and “[Performance monitoring and diagnostic tools](#)” [[SQL Anywhere Server - SQL Usage](#)].

### **Why does my application not work after an upgrade?**

SQL Anywhere developers strive to ensure that applications continue to work after an upgrade. However, your application may be affected by behavior changes and the removal of previously supported features.

To determine if behavior changes have been made to your version of SQL Anywhere, or if features have been removed or deprecated, select the section relevant to your version of SQL Anywhere. See “[SQL Anywhere 16 - Changes and Upgrading](#)”.

If your JDBC applications do not work after applying an Support Package, it is likely because you had either Interactive SQL, Sybase Central, the fast launchers, or other JDBC applications running when the Support Package was applied. Shutting down all JDBC-based applications and reapplying the Support Package may resolve the problem. See “[Troubleshooting: Ensure that JDBC applications are not running when applying a Support Package](#)” [[SQL Anywhere 16 - Changes and Upgrading](#)].

### **What operating systems are supported by the different versions of SQL Anywhere?**

See the tech note [SQL Anywhere Supported Platforms and Engineering Support Status](#) (<http://www.sybase.com/detail?id=1002288>).

### **What are the licensing requirements for a typical SQL Anywhere installation?**

For descriptions of the different licensing options for SQL Anywhere 10 and later, including examples that show you how to apply your license, see [SQL Anywhere Licensing](#) (<http://www.sybase.com/detail?id=1056242>).

### **What is the largest database size supported by SQL Anywhere?**

The size of the database that can be supported by SQL Anywhere is dependent on the memory, CPU, and disk drive capacity of the computer on which SQL Anywhere is installed. See “[SQL Anywhere size and number limitations](#)” [[SQL Anywhere Server - Database Administration](#)].

---

A number of our customers discuss the implementation of large databases in the SQL Anywhere Forum. See the forum thread <http://sqlanywhere-forum.sybase.com/questions/92/whats-the-size-of-your-biggest-database>.

### **How do I migrate from another database product to SQL Anywhere?**

To migrate to SQL Anywhere, you must import your data into a SQL Anywhere database. See “[Database migration to SQL Anywhere](#)” [*SQL Anywhere Server - SQL Usage*].

There are differences between products, such as different dialects of SQL, so your application may need to be modified.

### **What competitive advantages does SQL Anywhere offer?**

See “[Choosing SQL Anywhere for ISV Applications](http://www.sybase.com/files/Data_Sheets/Sybase_iAnywhereSQLAnywhere_isvapps_ds.pdf)” ([http://www.sybase.com/files/Data\\_Sheets/Sybase\\_iAnywhereSQLAnywhere\\_isvapps\\_ds.pdf](http://www.sybase.com/files/Data_Sheets/Sybase_iAnywhereSQLAnywhere_isvapps_ds.pdf)).

### **I need to diagnose performance issues with a specific query. How do I create a graphical plan with statistics?**

See “[Creating a graphical plan with detailed and node statistics](#)” [*SQL Anywhere Server - Database Administration*].

### **Can I use the OUTPUT statement in a stored procedure?**

No. The OUTPUT statement can only be executed in Interactive SQL and cannot be used within a stored procedure. Use the UNLOAD statement within your stored procedure to save the result set generated by a SQL statement to a text file. If you use the UNLOAD statement in your stored procedure, information is unloaded from the database server and not from the client computer from which the statement was executed. See “[UNLOAD statement](#)” [*SQL Anywhere Server - SQL Reference*].

### **Is there a tutorial or walk-through related to the feature I am interested in?**

See “[List of SQL Anywhere tutorials](#)” on page 51.

### **Where can I find more information about the cloud?**

See the SQL Anywhere, on-demand edition documentation at <http://dcx.sybase.com/cloud100>.

### **Where can I find more information?**

See:

- SQL Anywhere documentation: [dcx.sybase.com/dcx\\_home.html](http://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

## Symbols

- \* (asterisk)
  - (*see also* asterisks)
- push\_notifications option (*see* -push option)
- .NET
  - (*see also* ADO.NET)
- .NET synchronization logic
  - supported platforms, 6
- 32-bit
  - versions supported, 6
- 64-bit
  - versions supported, 6
- @environment-variable option (*see* @data option)
- @filename option (*see* @data option)

## A

- Access (*see* Microsoft Access)
- accessibility
  - Java Access Bridge, 14
  - Section 508 support, 7
- ActiveSync
  - supported platforms, 6
- Adaptive Server Anywhere (*see* SQL Anywhere)
- adding data
  - (*see also* importing data)
  - (*see also* inserting data)
- administration tools
  - supported platforms, 6
- administration utilities
  - (*see also* database administration utilities)
- always available
  - benefits of data synchronization, 33
- always available computing
  - SQL Anywhere hallmarks, 5
- ANSI compliance
  - (*see also* SQL standards)
- anti SQL log (*see* rollback logs)
- APIs
  - about, 28
- Apple (*see* Mac OS X)
- application programming interfaces
  - (*see also* APIs)
- archives
  - (*see also* backups)

- ARM chip
  - supported platforms, 6
- ARM processors
  - supported platforms, 6
- ARM V4T mode
  - supported platforms, 6
- ASE
  - (*see also* Adaptive Server Enterprise)
- assertions
  - responding to failure, 61
- attributes
  - relations, 17
- authorities (*see* roles)

## B

- back ticks (*see* back quotes)
- backticks (*see* back quotes)
- backup plans
  - creating, 61
- bi-directional synchronization
  - introducing synchronization technologies, 36
- binary constants (*see* binary literals)
- BINARY VARYING data type (*see* VARBINARY data type)
- BIT VARYING data type (*see* VARBIT data type)
- BlackBerry
  - SQL Anywhere databases unsupported, 23
  - supported platforms, 6
  - UltraLite database support, 23
- bloom filters (*see* Hash filters)
- BOM (byte order mark) (*see* BYTE ORDER MARK)
- book materialized views (*see* book views)
- bugs
  - providing feedback, vi
  - reporting, 61
- bypass queries
  - (*see also* simple queries)

## C

- C API (*see* SQL Anywhere C API)
- cac authentication
  - separately licensed components, 3
- Caldera
  - versions supported, 6
- CE (*see* Windows Mobile)
- Certicom
  - ordering encryption software, 3

- challenges for synchronization technologies
    - about, 33
  - CHAR VARYING data type (*see* VARCHAR data type)
  - CHARACTER VARYING data type (*see* VARCHAR data type)
  - choosing a synchronization technology
    - about, 37
  - choosing between SQL Anywhere and UltraLite databases
    - about, 23
  - cleaner (*see* database cleaner)
  - client/server
    - applications and multi-user databases, 25
  - cloud
    - accessing documentation, 63
  - collation sequences
    - (*see also* collations)
  - column statistics
    - (*see also* histograms)
  - columns
    - about, 16
  - command prompts
    - conventions, vi
    - curly braces, vi
    - parentheses, vi
    - quotes, vi
    - semicolons, vi
  - command sequence communication protocol
    - about, 30
    - diagram, 28
  - command shells
    - conventions, vi
    - curly braces, vi
    - parentheses, vi
    - quotes, vi
  - communication parameters (*see* connection parameters)
  - communication protocols
    - SQL Anywhere, 30
  - compliance with SQL standards
    - (*see also* SQL standards)
  - configuration files
    - (*see also* @data option)
  - conformance with SQL standards
    - (*see also* SQL standards)
  - connecting
    - SQL Anywhere examples, 59
  - connection assistant (*see* connect assistant)
  - connection IDs
    - (*see also* Number property)
  - connection parameters
    - (*see also* protocol options)
  - connections
    - FAQ, 59
  - console (*see* database server messages window) (*see* SQL Anywhere Console utility) (*see* SQL Anywhere Monitor console)
  - consolidated databases
    - about, 34
    - supported RDBMSs, 6
  - constant binary (*see* binary literals)
  - constant strings (*see* string literals)
  - constants (*see* binary literals) (*see* string literals)
  - CONTAINS clause (*see* CONTAINS search condition)
  - conventions
    - command prompts, vi
    - documentation, v
    - file names in documentation, v
    - operating systems, v
  - copy nodes
    - (*see also* read only scale out)
    - separately licensed components, 4
  - CPUs
    - licensing, 2
  - CustDB
    - about, 48
  - CustDB application
    - about, 48
- ## D
- data availability
    - benefits of data synchronization, 33
  - data consistency
    - challenges for synchronization technologies, 33
  - data definition language (*see* DDL)
  - data definition statements (*see* DDL)
  - data exchange
    - about, 31
  - data manipulation language (*see* DML)
  - data warehouses
    - ETL features, 27
  - database administrator (*see* DBA)
  - database encryption

---

- ordering encryption software, 3
- database files
  - introduction, 22
- database mirroring
  - separately licensed components, 4
- database objects
  - about, 18
- database replication
  - about, 31
- database server properties (*see* server properties)
- database servers
  - about, 15
  - differences between personal and network servers, 15
  - internals, 20
- database sizes
  - largest supported SQL Anywhere, 62
- database synchronization
  - about, 31
- database tables
  - about, 16
- database utilities
  - (*see also* utilities)
- databases
  - about, 15
  - choosing between SQL Anywhere and UltraLite, 23
  - client application, 16
  - components, 15
  - files, 22
  - language interface, 16
  - number running on a database server, 60
  - objects, 18
  - relational, 16
  - replicating, 31
  - server, 15
  - synchronizing, 31
- DBA
  - recovering password, 59
- DBI module (*see* DBD::SQLAnywhere)
- dbisql utility
  - (*see also* Interactive SQL)
  - (*see also* Interactive SQL utility (dbisql))
- dbremote utility
  - supported platforms, 6
- DEC data type (*see* DECIMAL data type)
- default password
  - databases, 59
- defaults
  - passwords, 59
- deleting databases
  - (*see also* erasing databases)
- demo database
  - about, 45
- demo.db
  - about, 45
  - FAQ, 59
  - recreating, 48
  - restoring, 48
- deploying
  - supported platforms, 6
- deployment edition
  - supported platforms, 6
- deployment releases
  - supported platforms, 6
- developer community
  - SQL Anywhere Community, vi
  - SQL Anywhere Forum, vi
- development platforms
  - supported platforms, 6
- documentation
  - conventions, v
  - for earlier versions of SQL Anywhere, viii
  - SQL Anywhere, v
- dual password control
  - see* dual control passwords, v

**E**

- earlier versions
  - finding earlier documentation for SQL Anywhere, viii
- editions
  - bundling separately licensed components, 2
  - where to find more information, 2
- embedded databases
  - defined, 24
  - example applications, 1
- encrypting
  - (*see also* encryption)
- encryption
  - ordering encryption software, 3
- end to end (*see* end-to-end encryption)
- ENG connection parameter (*see* ServerName connection parameter)

EngineName connection parameter (*see* ServerName connection parameter)  
engines  
    (*see also* database servers)  
    (*see also* servers)  
entities  
    relations, 17  
environment variable option (*see* @data option)  
erasing  
    (*see also* deleting)  
    (*see also* dropping)  
ESRI shapefiles (*see* shapefiles)  
ETL  
    about, 27  
    supported features, 27  
Extensible Markup Language (*see* XML)  
extract, transform, and load  
    ETL, 27

**F**

failure  
    assertions, 61  
FAQ  
    about, 59  
    SQL Anywhere, 59  
features  
    (*see also* supported platforms)  
Federal Information Processing Standard Publication compliance  
    (*see also* SQL standards)  
feedback  
    reporting an error, vi  
    requesting an update, vi  
file sharing message type  
    SQL Remote supported Windows platforms, 6  
finding out more and requesting technical assistance  
    technical support, vi  
FIPS  
    ordering encryption software, 3  
FIPS compliance  
    (*see also* SQL standards)  
FIPS option  
    separately licensed component, 3  
FIPS protocol option  
    separately licensed component, 3  
flagger (*see* SQL Flagger)  
foreign keys

    defined, 18  
    free search (*see* full text search)  
    frequently asked questions (*see* faq)  
    frontline environments  
        about, 1  
FTP message type  
    SQL Remote supported Windows platforms, 6

**G**

generating database documentation (*see* documenting a database)  
getting help  
    technical support, vi  
getting started  
    SQL Anywhere 16, 55  
grant option (*see* WITH GRANT OPTION clause)  
graphical plans  
    FAQ, 63  
groups (*see* user-extended roles)

**H**

hallmarks  
    SQL Anywhere, 5  
Handheld PC  
    supported platforms, 6  
help  
    technical support, vi  
Hewlett Packard HP-UX  
    (*see also* HP-UX)  
hexadecimal constants  
    (*see also* binary literals)  
hierarchical data structures  
    hierarchical database configurations, 35  
high availability  
    separately licensed components, 4  
host platforms  
    supported platforms, 6  
HP-UX  
    supported platforms, 6

**I**

IBM AIX  
    (*see also* AIX)  
    supported platforms, 6  
immediate materialized views (*see* immediate views)  
importing data  
    (*see also* inserting data)

---

in memory mode  
    separately licensed components, 4

initialization files (*see* INI files)

inputting data (*see* importing data)

inserting data  
    (*see also* importing data)

install-dir (*see* installation directory)

installing  
    supported platforms, 6

INT data type (*see* INTEGER data type)

inter-query parallelism  
    (*see also* intra-query parallelism)

Interactive SQL  
    (*see also* dbisql utility)  
    (*see also* Interactive SQL utility (dbisql))  
    supported platforms, 6

Interactive SQL utility (dbisql)  
    (*see also* dbisql utility)  
    (*see also* Interactive SQL)

interfaces  
    (*see also* APIs)

internals  
    database files, 22  
    database server, 20

intra-query parallelism  
    (*see also* inter-query parallelism)

introducing SQL Anywhere  
    database technologies, 15  
    design and management tools, 41  
    getting started, 55  
    messaging technologies, 31  
    overview, 1  
    synchronization technologies, 31

iPhone  
    supported platforms, 6

ISO compliance  
    (*see also* SQL standards)

**J**

J2EE (*see* Java EE)

J2ME (*see* Java ME)

J2SE (*see* Java SE)

Java Access Bridge  
    accessibility enablement module, 14

java administration tools (*see* administration tools)

Java ME  
    versions supported, 6

Java synchronization logic  
    supported platforms, 6

## K

Kerberos logins  
    (*see also* Kerberos)

kernel  
    versions supported, 6

keys  
    about, 18

## L

language utility (*see* language selection utility)

license utility (*see* server licensing utility (dblic))

licenses  
    separately licensed components, 3

licensing  
    CPUs, 2  
    processors, 2  
    separately licensed components, 3  
    SQL Anywhere requirements, 62

limits  
    (*see also* limitations)

Linux  
    supported platforms, 6

literal strings (*see* string literals)

local servers (*see* personal server)

locking  
    (*see also* locks)

locks  
    (*see also* locking)

LONG BIT VARYING data type (*see* LONG VARBIT data type)

## M

Mac OS X  
    supported platforms, 6  
    UltraLite unsupported, 23

Macintosh  
    (*see also* Mac OS X)  
    supported platforms, 6

Mandrake  
    versions supported, 6

materialized views  
    ETL features, 27

Message Agent (dbremote)  
    supported platforms, 6

message log  
    (*see also* database server message log)  
message log file  
    (*see also* database server message log)  
message-based synchronization  
    introducing synchronization technologies, 36  
MIBs  
    (*see also* management information bases)  
migrating  
    from another product to SQL Anywhere, 63  
MIPS chip  
    supported platforms, 6  
mirror transaction log (*see* transaction log mirror)  
mirroring  
    (*see also* database mirroring)  
    separately licensed components, 4  
mobile computing  
    example applications, 1  
MobiLink  
    comparing SQL Remote and MobiLink, 31  
    supported platforms, 6  
MobiLink consolidated databases  
    supported RDBMSs, 6  
MobiLink supported platforms  
    about, 6  
MobiLink synchronization  
    features compared, 37  
    supported platforms, 6  
Monitor  
    (*see also* MobiLink Monitor)  
    (*see also* Monitor metrics)  
    (*see also* Performance Monitor)  
moving data  
    (*see also* exporting data)  
    (*see also* importing data)  
    (*see also* inserting data)  
multi version concurrency control (*see* snapshot isolation)  
multi-tier computing  
    introduction, 25  
multi-user database  
    defined, 25  
multiple databases  
    competition for resources , 61  
    running on a single server, 26  
multiple versions  
    SQL Anywhere, 60  
MVCC (*see* snapshot isolation)

## N

n-tier computing  
    introduction, 25  
NATIONAL CHAR data type (*see* NCHAR data type)  
NATIONAL CHAR VARYING data type (*see* NVARCHAR data type)  
NATIONAL CHARACTER data type (*see* NCHAR data type)  
NATIONAL CHARACTER VARYING data type (*see* NVARCHAR data type)  
NCHAR VARYING data type (*see* NVARCHAR data type)  
network database server (*see* network server)  
network parameters  
    (*see also* connection parameters)  
network server  
    overview, 15  
Network Server Monitor  
    (*see also* SQL Anywhere Console utility (dbconsole))  
newdemo  
    about, 48  
number of databases running on a server  
    about, 60  
numeric constants (*see* binary literals)

## O

object identifiers (*see* OIDs)  
ODBC data sources  
    SQL Anywhere 16 CustDB, 48  
    SQL Anywhere 16 Demo, 45  
ODBC drivers  
    (*see also* SQL Anywhere ODBC driver)  
OIDs  
    (*see also* object identifiers)  
OLE DB drivers  
    supported processors, 6  
OMNI (*see* remote data access)  
online analytical processing (*see* OLAP)  
online books  
    PDF, v  
Open Client (*see* Sybase Open Client)  
Open Server (*see* Sybase Open Server)  
operating systems  
    supported by SQL Anywhere, 62  
    supported platforms, 6

---

Windows, v  
operational servers (*see* partner servers)  
options  
    (*see also* database options)  
    (*see also* options (UltraLite))  
Oracle Solaris  
    (*see also* Solaris)  
outputting  
    (*see also* exporting data)  
outputting data (*see* exporting data)

**P**

passwords  
    default for demo.db, 59  
    recovering DBA, 59

PDF  
    documentation, v

performance  
    improving application, 62  
    improving database, 62  
    resolving slow application, 62

permissions (*see* privileges)

personal database server (*see* personal server)

personal server  
    overview, 15

physical data models  
    using SAP Sybase PowerDesigner, 43

platform support  
    about, 6

platforms  
    supported operating systems, 6

Pocket PC  
    supported platforms, 6

polling  
    (*see also* refreshing)

predicates  
    (*see also* search conditions)

primary keys  
    defined, 18

processors  
    licensing, 2

programming interfaces  
    (*see also* APIs)  
    about, 28

propagation methods  
    introducing synchronization technologies, 36

protocol options

    (*see also* connection parameters)  
purchasing  
    separately licensed components, 3

## Q

query bypass (*see* bypassing optimization)  
query optimization  
    (*see also* optimizer)  
query optimizer  
    (*see also* optimizer)  
quick start  
    SQL Anywhere 16, 55  
quotes  
    (*see also* quotation marks)

## R

range  
    (*see also* limitations)

read only scale out  
    separately licensed components, 4

read-only scale-out (*see* read only scale out)

READCOMMITTED table hint  
    (*see also* read committed)

READUNCOMMITTED table hint  
    (*see also* read uncommitted)

Red Hat  
    versions supported, 6

Rehabilitation Act  
    accessibility enablement module, 14  
    accessibility support, 7

relational database systems  
    about, 15  
    concepts, 16

relational databases  
    about, 17

relations  
    entities, 17

reloading databases  
    (*see also* rebuilding databases)

remote data access  
    supported platforms, 6

REMOTE DBA authority (*see*  
SYS\_RUN\_REPLICATION\_ROLE system role)

REPEATABLE READ table hint  
    (*see also* repeatable reads)

replicating databases  
    about, 31

- replication
  - about SQL Remote and MobiLink, 31
  - comparing SQL Remote and MobiLink, 31
- request level log (*see* request log)
- request level logging (*see* request logging)
- request-level log (*see* request log)
- request-level logging (*see* request logging)
- response time
  - benefits of data synchronization, 33
- roles
  - in sample database, 46
- rows
  - about, 16
- S**
- sample database
  - custdb.db for UltraLite and MobiLink applications, 48
  - demo.db for SQL Anywhere examples, 45
  - recreating demo.db, 48
  - restoring demo.db, 48
  - roles, 46
  - schema for custdb.db, 48
- SAP Sybase PowerDesigner
  - about, 43
- scale out
  - separately licensed components, 4
- scale-out (*see* scale out)
- scRepository (*see* .scRepository610)
- search conditions
  - (*see also* predicates)
- Section 508
  - accessibility enablement module, 14
  - accessibility support, 7
  - VPAT, 7
- security option
  - about, 3
- select-list (*see* SELECT list)
- separately licensed components
  - about, 3
- sequence generators (*see* sequences)
- SERIALIZABLE table hint
  - (*see also* serializable)
- server location utility (*see* server enumeration utility (dblocate))
- servers
  - (*see also* database servers)
  - number of databases running, 60
- service creation utility (*see* service utility (dbsvc))
- service monitor (*see* SQL Anywhere Monitor)
- session-based synchronization
  - about, 36
- simple network management protocol (*see* SNMP)
- simple queries
  - (*see also* bypass queries)
- smartphone
  - SQL Anywhere database support, 23
  - supported platforms, 6
  - UltraLite database support, 23
- SMTP message type
  - supported platforms, 6
- Solaris
  - supported platforms, 6
- Spatial Preview tab
  - (*see also see also* Spatial Viewer)
- Spatial Viewer
  - (*see also see also* Spatial Preview tab )
- spawn utility (*see* start server in background utility (dbspawn))
- SQL Anywhere
  - about, 1
  - compared to UltraLite, 23
  - competitive advantage, 63
  - components, 1
  - documentation, v
  - FAQ, 59
  - getting started, 56
  - hallmarks, 5
  - intended uses, 1
  - internals, 20
  - supported platforms, 6
- SQL Anywhere Community
  - technical support, vi
- SQL Anywhere databases
  - compared to UltraLite database, 23
- SQL Anywhere debugger (*see* debugger)
- SQL Anywhere Forum
  - technical support, vi
- SQL Anywhere Monitor (*see* Monitor)
- SQL Anywhere sample database
  - about, 45
  - recreating, 48
  - restoring, 48
- SQL Anywhere supported platforms
  - about, 6



---

SQL Anywhere Tech Corner  
 finding out more and requesting technical support, vi

SQL Remote  
 comparing MobiLink and SQL Remote, 31  
 supported platforms, 6

SQL Remote supported platforms  
 about, 6

SQL/2003 compliance  
 (*see also* SQL standards)

SQLX (*see* SQL/XML)

SSL (*see* transport-layer security)

standalone applications  
 defined, 24

standards  
 (*see also* SQL standards)

standards and compatibility  
 (*see also* SQL standards)

START ENGINE statement (deprecated) (*see* START SERVER statement)

state files (*see* state information files)

statistics  
 (*see also* histograms)

STDDEV function (*see* STDDEV\_SAMP function)

STOP ENGINE statement (*see* STOP SERVER statement)

store-and-forward  
 SQL Remote synchronization, 36

stored procedures  
 saving result sets, 63

string constants (*see* string literals)

strong encryption  
 ordering encryption software, 3

support  
 SQL Anywhere Community, vi  
 SQL Anywhere Forum, vi

supported features  
 (*see also* supported platforms)

supported platforms  
 about, 6  
 MobiLink, 6  
 SQL Anywhere server, 6  
 SQL Remote, 6  
 UltraLite, 6

SuSE  
 versions supported, 6

Sybase Central  
 supported platforms, 6

synchronization  
 about SQL Remote and MobiLink, 31  
 comparing SQL Remote and MobiLink, 31  
 MobiLink features compared, 37

synchronizing databases  
 about, 31

SYSSERVERS (*see* SYSSERVER system view)

system privileges  
 (*see also* privileges)

system requirements  
 supported platforms, 6

## T

tables  
 about, 16  
 characteristics, 16

tabular data stream communication protocol  
 about, 30  
 diagram, 28

target platforms  
 UltraLite, 6

TDS communication protocol  
 (*see also* tabular data stream communication protocol)

tech corners  
 finding out more and requesting technical support, vi

technical support  
 SQL Anywhere Community, vi  
 SQL Anywhere Forum, vi

temporary files  
 introduction, 22

text searching (*see* full text search)

three-tier computing  
 introduction, 25

thumb mode  
 supported, 6

tools  
 design and management tools, 41

tracing  
 (*see also* diagnostic tracing)

transaction log  
 introduction, 22

transactional integrity  
 about, 33

transactional technology  
 challenges for synchronization technologies, 33

- transferring data
  - (*see also* exporting data)
- transport-layer security
  - ordering encryption software, 3
- troubleshooting
  - FAQ, 59
  - SQL Anywhere Community, vi
  - SQL Anywhere Forum, vi
- TSQL (*see* Transact-SQL)
- tuples
  - about, 17
- TurboLinux
  - versions supported, 6
- tutorials
  - about, 51
  - complete listing of, 51
  - recreating the sample database, 48
- two-way synchronization
  - introducing synchronization technologies, 36
- types of data (*see* data types)
- typing completion (*see* text completion)

## U

- UltraLite
  - compared to SQL Anywhere, 23
  - supported platforms, 6
- UltraLite databases
  - compared to SQL Anywhere database, 23
- UltraLite supported platforms
  - about, 6
- Unix
  - supported platforms, 6
- unload database utility (dbunload) (*see* unload utility (dbunload) )
- unscheduled requests (*see* ReqStatus property) (*see* UnschReq property)
- upgrading
  - resolving slow application performance, 62
- user IDs
  - default for demo.db, 59
- utilities
  - (*see also* database utilities)

## V

- V4T mode
  - ARM processors, 6
- VARIANCE function (*see* VAR\_SAMP function)

- VCS agents
  - separately licensed components, 4
- Veritas Cluster Server agents
  - separately licensed components, 4
- version number
  - determining SQL Anywhere, 60
- virtualization
  - support for virtualized environments, 7
- virtualized environments
  - SQL Anywhere support, 7
- Vista
  - supported platforms, 6
- VM
  - SQL Anywhere in virtualized environments, 7
- Voluntary Product Accessibility Template
  - Section 508 support, 7
- VPAT
  - Section 508 support, 7

## W

- walk-throughs
  - FAQ, 63
- whitepapers
  - recommended reading, vii
- Windows
  - (*see also* Windows 2003)
  - (*see also* Windows Mobile)
  - (*see also* Windows XP)
  - documentation conventions, v
  - supported platforms, 6
- Windows 2003
  - supported platforms, 6
- Windows 7
  - supported platforms, 6
- Windows CE (*see* Windows Mobile)
- Windows Mobile
  - processors supported, 6
  - supported platforms, 6
- Windows Server 2008/2008 R2
  - supported platforms, 6
- Windows Vista
  - supported platforms, 6
- Windows XP
  - supported platforms, 6
- workgroup computing
  - (*see also* embedded databases)

---

## **X**

x86 chip

supported platforms, 6

XScale processors

supported platforms, 6

---