

SYBASE®

Introduction to Sybase® IQ

Sybase® IQ

12.6

DOCUMENT ID: DC38159-01-1260-02

LAST REVISED: December 2004

Copyright © 1991-2004 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase, the Sybase logo, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Warehouse, Afaia, Answers Anywhere, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Translator, APT-Library, AvantGo Mobile Delivery, AvantGo Mobile Inspection, AvantGo Mobile Marketing Channel, AvantGo Mobile Pharma, AvantGo Mobile Sales, AvantGo Pylon, AvantGo Pylon Application Server, AvantGo Pylon Conduit, AvantGo Pylon PIM Server, AvantGo Pylon Pro, Backup Server, BizTracker, ClearConnect, Client-Library, Client Services, Convoy/DM, Copernicus, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DataWindow .NET, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, e-ADK, E-Anywhere, e-Biz Impact, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, GlobalFIX, iAnywhere, iAnywhere Solutions, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, M2M Anywhere, Mail Anywhere Studio, MainframeConnect, Maintenance Express, Manage Anywhere Studio, M-Business Channel, M-Business Network, M-Business Server, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, mFolio, Mirror Activator, My AvantGo, My AvantGo Media Channel, My AvantGo Mobile Marketing, MySupport, Net-Gateway, Net-Library, New Era of Networks, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, Orchestration Studio, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, PocketBuilder, Pocket PowerBuilder, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, QAnywhere, Rapport, RemoteWare, RepConnector, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Report-Execute, Report Workbench, Resource Manager, RFID Anywhere, RW-DisplayLib, RW-Library, S-Designor, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase IQ, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SybFlex, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, TradeForce, Transact-SQL, Translation Toolkit, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server, XcelleNet, and XP Server are trademarks of Sybase, Inc. 10/04

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book	vii
CHAPTER 1 Introducing Sybase IQ	1
Welcome to Sybase IQ	1
The IQ approach	1
Why Sybase IQ?	2
Sybase IQ benefits	3
Sybase IQ components	4
The Sybase IQ database server	4
Administration utilities	5
Using the documentation	5
Using accessibility features	6
The Sybase IQ program group	9
The sample database	10
Preparing for the tutorial	12
CHAPTER 2 Terms and Concepts	13
Sybase IQ and data warehousing terms	13
Adaptive Server Anywhere	13
Anywhere database	13
Catalog Store	14
Component Integration Services (CIS)	14
Connection Profile	14
Dbpace	14
Enterprise database	14
Freelist	15
IQ database	15
IQ Local Store	15
IQ Message Log	15
IQ Store	15
IQ Temporary Store	16
IQ Transaction log	16
Join index	16

- Metadata 16
- Multiplex 17
- Object 17
- Proxy database 17
- Synchronize command 17
- Synchronization 17
- Database computing concepts 18
 - Sybase IQ database files 19
 - How the pieces fit together 20
 - SQL and database computing 21
- Relational database concepts 21
 - Database tables 21
 - Primary and foreign keys 23
 - Other database objects 24
 - Queries 25
 - Other SQL statements 26
 - The system tables 26

- CHAPTER 3**
- Running and Connecting to Servers 27**
 - Understanding database servers 27
 - Sybase Central and database management 28
 - Starting and stopping Sybase Central 29
 - The Sybase Central interface 29
 - Registering a plug-in 30
 - Stopping Sybase Central 30
 - Running the IQ Agent 30
 - Copying the sample database 31
 - Starting the database server 32
 - Start Server Wizard: Options 33
 - Start Server Wizard: Connection Parameters 33
 - Specifying the Agent Port 35
 - Start Server Wizard: Database Path 35
 - Start Server Wizard: Summary 36
 - Navigating the main Sybase Central window 37
 - Connecting to a database 37
 - Viewing a database schema 37
 - Viewing the tables in a database 38
 - Viewing other database objects 39
 - Disconnecting databases and users 40
 - Connecting to databases from DBISQL 41
 - Sending commands to the database 42
 - Stopping a database server 43

CHAPTER 4	Managing Databases	45
	Managing databases.....	45
	Creating databases	45
	Viewing database properties	58
	Viewing database options	58
	Deleting databases.....	58
	Managing tables.....	59
	Creating a table and adding columns with Sybase Central	59
	Editing existing tables.....	60
	Deleting tables with Sybase Central.....	61
	Managing procedures	62
	Viewing stored procedure code with Sybase Central	62
	Managing users and groups.....	63
	Adding a group to the database	64
	Adding a user to the database	65
	Adding a user to a group	67
	Managing connected users	68
CHAPTER 5	Indexing and Loading Data	71
	Creating column indexes.....	71
	Creating join indexes.....	74
	Updating join index data	76
	Loading data	76
CHAPTER 6	Managing Dbspaces	79
	Overview of dbspaces.....	79
	Creating dbspaces	80
	Specifying dbspace name and store	80
	Specifying dbspace path	81
	Altering dbspaces.....	81
	Deleting dbspaces.....	82
	Copying dbspaces.....	82
CHAPTER 7	Updating Databases	83
	Migrating data from databases from earlier versions	83
	Adding rows to a table.....	84
	Canceling changes.....	85
	Making changes permanent	85
	Ensuring data consistency	86
	Deleting rows	89
	Updating rows	90
	Synchronizing join indexes.....	90

	Deleting dbspaces, tables, and indexes.....	91
CHAPTER 8	Using Views.....	93
	Using views	93
	Defining a view	93
	Managing views with Sybase Central.....	95
	View restrictions and advanced uses	96
	Using views for security	96
	Restrictions on views	98
Index		99

About This Book

This book provides an overview of Sybase[®] IQ, a high-performance decision support server designed specifically for data warehouses and data marts.

Audience

This guide is for system administrators, managers, or anyone who will be involved in setting up or using Sybase IQ. This guide assumes no technical knowledge of the Sybase products.

How to use this book

Read this guide before you begin using Sybase IQ, to establish a foundation for using the product.

This book combines overview information with a hands-on introduction to Sybase IQ. *As you use the product, you may encounter situations that are exceptions to the general rules and explanations documented here.* For detailed information, this book refers the reader to other books in the documentation set.

Note Because Sybase IQ is an extension of the Adaptive Server[®] Anywhere product, IQ and Anywhere support many of the same features. The IQ documentation set refers the reader to Anywhere documentation where appropriate.

It is a good idea to read this entire book soon after installation, before your site sets up a production server.

The following table shows which chapters fit a particular interest or need.

Table 1: Guide to using this book

To do this ...	Read ...
Understand the product overview	Chapters 1-2
Complete a product tutorial	Chapters 1-8
Complete a system administration tutorial	Chapters 3-8
Run queries	Chapter 8

Related documents

Documentation for Sybase IQ:

- *New Features in Sybase IQ 12.6*
Read just before or after purchasing Sybase IQ for a list of new features.
- *Sybase IQ Performance and Tuning Guide*
Read to understand query optimization, design, and tuning issues for very large databases.
- *Sybase IQ Reference Manual*
Read for a full description of the SQL language, stored procedures, data types, and system tables supported by Sybase IQ.
- *Sybase IQ System Administration Guide*
Read to manage the IQ Store.
- *Sybase IQ Troubleshooting and Error Messages Guide*
Read to solve problems, perform system recovery and database repair, and understand both IQ error messages, which are referenced by SQLCode, SQLState and message text, and SQL preprocessor errors and warnings.
- *Sybase IQ Utility Guide*
Read for Sybase IQ utility program reference material, such as available syntax, parameters, and options.
- *Large Objects Management in Sybase IQ*
Read to understand storage and retrieval of Binary Large Objects (BLOBs) and Character Large Objects (CLOBs) within the Sybase IQ data repository. You need a separate license to install this product option.
- *Sybase IQ Installation and Configuration Guide*
Read the edition for your platform before and while installing Sybase IQ, when migrating to a new version of Sybase IQ, or when configuring Sybase IQ for a particular platform.

- *Sybase IQ Release Bulletin*

Read just before or after purchasing Sybase IQ for last minute changes to the product and documentation. Read for help if you encounter a problem.

Note Because Sybase IQ is an extension of Adaptive Server® Anywhere, a component of SQL Anywhere® Studio, IQ supports many of the same features as Adaptive Server Anywhere. The IQ documentation set refers you to SQL Anywhere Studio documentation where appropriate.

Documentation for Adaptive Server Anywhere:

- *Adaptive Server Anywhere Programming Guide*

Intended for application developers writing programs that directly access the ODBC, Embedded SQL™, or Open Client™ interfaces, this book describes how to develop applications for Adaptive Server Anywhere.

- *Adaptive Server Anywhere Database Administration Guide*

Intended for all users, this book covers material related to running, managing, and configuring databases and database servers.

- *Adaptive Server Anywhere Error Messages*

This book lists all Adaptive Server Anywhere error messages with diagnostic information.

- *Adaptive Server Anywhere SQL Reference Manual*

Intended for all users, this book provides a complete reference for the SQL language used by Adaptive Server Anywhere. It also describes the Adaptive Server Anywhere system tables and procedures.

You can also refer to the Adaptive Server Anywhere documentation in the SQL Anywhere Studio 9.0.1 collection on the Sybase Product Manuals Web site. To access this site, go to Product Manuals at <http://www.sybase.com/support/manuals/>.

Other sources of information

Use the Sybase Getting Started CD, the SyBooks CD, and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.

-
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

To access the Sybase Product Manuals Web site, go to Product Manuals at <http://www.sybase.com/support/manuals/>.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ Finding the latest information on product certifications

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Select Products from the navigation bar on the left.
- 3 Select a product name from the product list and click Go.
- 4 Select the Certification Report filter, specify a time frame, and click Go.
- 5 Click a Certification Report title to display the report.

❖ Creating a personalized view of the Sybase Web site (including support pages)

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

❖ Finding the latest information on EBFs and software maintenance

- 1 Point your Web browser to the Sybase Support Page at <http://www.sybase.com/support>.
- 2 Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
- 3 Select a product.
- 4 Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the “Technical Support Contact” role to your MySybase profile.

- 5 Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Syntax conventions

This documentation uses the following syntax conventions in syntax descriptions:

- **Keywords** SQL keywords are shown in UPPER CASE. However, SQL keywords are case insensitive, so you can enter keywords in any case you wish. SELECT is the same as Select, which is the same as select.
- **Placeholders** Items that must be replaced with appropriate identifiers or expressions are shown in *italics*.
- **Continuation** Lines beginning with ... are a continuation of the statements from the previous line.
- **Repeating items** Lists of repeating items are shown with an element of the list followed by an ellipsis (three dots). One or more list elements are allowed. If more than one is specified, they must be separated by commas.
- **Optional portions** Optional portions of a statement are enclosed by square brackets. For example:

```
RELEASE SAVEPOINT [ savepoint-name ]
```

It indicates that the *savepoint-name* is optional. Do not type the square brackets.

- **Options** When none or only one of a list of items must be chosen, the items are separated by vertical bars and the list enclosed in square brackets. For example:

```
[ ASC | DESC ]
```

The vertical bar indicates that you can choose one of ASC, DESC, or neither. Do not type the square brackets.

- **Alternatives** When precisely one option must be chosen, the alternatives are enclosed in curly braces. For example:

```
QUOTES { ON | OFF }
```

Curly braces indicate that exactly one of ON or OFF must be provided. Do not type the braces.

Typographic conventions

Table 2 lists the typographic conventions used in this documentation.

Table 2: Typographic conventions

Item	Description
Code	SQL and program code is displayed in a mono-spaced (fixed-width) font.
User entry	Text entered by the user is shown in bold serif type.
<i>emphasis</i>	Emphasized words are shown in italic.
<i>file names</i>	File names are shown in italic.
database objects	Names of database objects, such as tables and procedures, are shown in bold, san-serif type in print, and in italic online.

The sample database

Sybase IQ includes a sample database, which many of the examples in the IQ documentation use.

The sample database represents a small company. It contains internal information about the company (employees, departments, and financial data), as well as product information (products), sales information (sales orders, customers, and contacts), and financial information (*fin_code*, *fin_data*).

The sample database is held in a file named *asiqdemo.db*, located in the directory *\$ASDIR/demo* on UNIX systems and *%ASDIR%\demo* on Windows systems.

Accessibility features

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.

Sybase IQ 12.6 and the HTML documentation have been tested for compliance with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

For information about accessibility support in the Sybase IQ plug-in for Sybase Central, see “Using accessibility features” on page 6. The online help for this product, which you can navigate using a screen reader, also describes accessibility features, including Sybase Central keyboard shortcuts.

Note You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool and see “Using screen readers” on page 7.

For information about how Sybase supports accessibility, see Sybase Accessibility at <http://www.sybase.com/accessibility>. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

For a Section 508 compliance statement for Sybase IQ, go to Sybase Accessibility at <http://www.sybase.com/products/accessibility>.

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.



Introducing Sybase IQ

About this chapter

This chapter introduces Sybase IQ and explains how it fits into the Sybase data warehouse product family.

Welcome to Sybase IQ

Sybase IQ is a high-performance decision support server designed specifically for data warehousing.

Sybase IQ is part of the Adaptive Server® product family that includes **Adaptive Server® Enterprise** for enterprise transaction and mixed workload environments and **Adaptive Server Anywhere**, a small footprint version of Adaptive Server, often used for mobile and occasionally connected computing. **Component Integration Services** are embedded within Sybase IQ and all Adaptive Server configurations to provide IQ users with direct access to relational or nonrelational databases on the mainframe, UNIX, or Windows servers.

There are slight differences in the Structured Query Language (SQL, pronounced “sequel”) syntax supported by Sybase IQ, Adaptive Server Anywhere, and Adaptive Server Enterprise. *Sybase IQ Reference Manual* provides an overview of compatibility of the SQL used by IQ, Anywhere and Enterprise.

The IQ approach

An IQ database is fundamentally different from a conventional relational database because it focuses on readers, not writers. In a conventional database, it is most important to allow many users to update the database instantly and accurately, without interfering with each other. By contrast, in an IQ database, fast query response for many users is most important.

This reader-oriented approach drives the design of IQ architecture and features. The IQ way means:

- Data is stored in columns, not rows.
- Placing indexes on all columns gives you a performance advantage.
- A large page size gives you a performance advantage.
- A large temporary cache gives you a performance advantage for most operations.
- Access to data is controlled at the table level, using table-level locks rather than row locks, and a technique called table-level versioning.
- Most query results focus on data at the table level.
- Most insertions and deletions write data for an entire table, not for a single row.

To get the best performance from your IQ database, keep these fundamental differences in mind. For details on setting up IQ to get the best performance, see *Sybase IQ System Administration Guide* and *Sybase IQ Performance and Tuning Guide*.

Why Sybase IQ?

A **decision support system** is a software application designed to let an organization analyze data in order to support business decision making. You can use two different architectures to design decision support systems:

- A centralized architecture contains definitions for the entire enterprise, providing a consolidated view
- A set of databases, one for each business function or department, specify data using the same time and geographic dimensions, providing consistent comparisons

You can use Sybase IQ to implement either architecture, according to your analysis goals.

While specific requirements may vary, most databases used in decision support systems are specially adapted. Conventional relational databases used for running business processes are tuned for OLTP (On-Line Transaction Processing) and are not optimal for analysis, as shown in the following table:

Table 1-1: Database adaptations for business goals

To run process	To analyze business
Clerical users	Managerial, clerical and analytical users
Current data	Historical data
Detailed data	Summarized and detailed data
Highly changeable data	Stable data
Supports day-to-day operations	Supports strategic decisions
Transaction driven	Analysis driven
Optimized for structured queries	Optimized for ad hoc queries

When the database supports a specific functional department rather than being enterprise-wide, it is often referred to as a **data mart**. You can also think of a data mart as an application-specific database that focuses on a specific business problem.

Data marts can deliver the business intelligence required to gain competitive advantage at a modest cost and with exceptional ease of administration. Companies may have a number of business-oriented data marts supported by a central data model along with a central data staging and consolidation warehouse.

Decision support analysis has become less batch-oriented and more interactive. The business market demands rapid response to queries. With data accessible, more queries can be generated and decision making is improved.

Sybase IQ benefits

Sybase IQ provides benefits that support the interactive approach to decision support including:

- Intelligent query processing: Sybase IQ uses index-only access plans to process only the data needed to satisfy any type of query.
- Truly interactive ad hoc query performance on a uniprocessor as well as on parallel systems. An **ad hoc query** is one about which the system has no prior knowledge and for which no explicit tuning is required. Ad hoc queries are distinguished from standard or production reports, where only pre-defined variables, such as dates, are used to generate pre-defined reports on a regular basis.

- Multiplex capability for managing large query loads in a multi-server configuration. For more information, see the *Sybase IQ System Administration Guide*.
- Fully-flexible schema support.
- Efficient query execution without query-specific tuning by the System Administrator (under most circumstances).
- Fast initial and incremental loading.
- Fast aggregations, counts, comparisons of data.
- Parallel processing optimized for multi-user environments.
- Stored procedures.
- Increased productivity due to reduced query time.
- Entire database and indexing stored in less space than raw data.
- Reduced I/O.

Sybase IQ components

Sybase IQ consists of a database server and a set of administration utilities.

The Sybase IQ database server

Sybase IQ is an extended version of the Adaptive Server Anywhere network server. The network server supports client/server communications across a network and multiple users.

To start the server, run the appropriate startup utility for your operating system:

- On UNIX operating systems, use *start_asiq*.
- On Windows operating systems, use *start_asiq.exe*.

Administration utilities

Sybase IQ includes a set of applications to help you manage databases. These applications can run either on the same machine as your IQ database server, or can run across a network against a network server.

The particular Sybase IQ components you have installed depend on which operating system you use, what choices you made when installing the software, and whether you installed Sybase IQ as part of another product.

Not all components are available on all operating systems. See your *Sybase IQ Installation and Configuration Guide* for a list of components on your platform.

These utilities are:

- *Sybase Central*

An application for managing Sybase databases. It helps you manage database objects and perform common administrative tasks such as creating databases, creating multiplexes, adding users, and adding tables. Tutorials and procedures in this book use Sybase Central.

- DBISQL

Interactive SQL (DBISQL) is an application that lets you type a SQL statement and send it to a database. Because interactions with databases use SQL statements, you can carry out any database operation from DBISQL. For more information about Interactive SQL, see the *Sybase IQ Utility Guide*.

- *Command-line utilities*

A set of command-line utilities is provided for carrying out administrative tasks, such as backing up a database. Command-line utilities can be included in batch files for repeated use. For more information about command-line utilities, see the *Sybase IQ Utility Guide*.

Using the documentation

You can see a complete set of documentation for Sybase IQ and all its associated tools in the Sybooks CD.

The Sybase IQ plug-in for Sybase Central™ has its own online help. Choose Help from the main menu after starting Sybase Central, then specify Sybase IQ Help.

Using accessibility features

Sybase Central lets you tab to each field or activate a desired menu command or text box by typing Alt plus the underscored letter in the text box or menu item name.

Table 1-2 lists main menu shortcuts.

Note Shortcut keys may change from release to release.

Table 1-2: Sybase Central main menu shortcuts

Shortcut	Menu item
Alt+A	Task
Alt+E	Edit
Alt+F	File
Alt+H	Help
Alt+T	Tools
Alt+V	View

Table 1-3 lists function keys to run common Sybase Central commands:

Table 1-3: Sybase Central function keys

Function key	Command
F2	Initiate editing in a dialog cell.
F9	Open Connection Profiles dialog
F11	Connect
F12	Disconnect

You can use combinations Ctrl+C (Cut) and Ctrl+V (Paste) anywhere in the Sybase Central wizards. In all wizards, F1 maps to online help and the escape key maps to the Cancel button.

Each wizard, in turn, has its own shortcut. Table 1-4 lists shortcuts for the Start Server wizard.

Table 1-4: Start Server shortcut keys

Key combination	Parameter
Alt+a	Password
Alt+C	Connection Profile
Alt+D	Database Name
Alt+H	Host Machine
Alt+P	Port Number
Alt+S	Server Name
Alt+U	Username

Using screen readers

Table 1-5 lists terms appearing in Sybase IQ help and user documentation that may be initially ambiguous to screen reader users unfamiliar with the product. The first column lists the term, abbreviation or acronym as it appears on screen or in print. The second column lists how the term, abbreviation or acronym is typically pronounced.

Note In order to use the JAWS screen reader with Sybase IQ, you must install the Java Access Bridge package, which you can download for free from Sun at <http://java.sun.com/products/accessbridge/>.

Table 1-5: Pronunciation of common IQ terms

What the screen reader says	How it is pronounced
.db	dot D B
.iqmsg	dot I Q M S G
.odbc.ini	dot O D B C dot in E
.srvlog	dot S R V log
asiqdemo	A S I Q demmo
asiqdemo.cfg	A S I Q demmo dot C F G
asiqdemo.db	A S I Q demmo dot D B
asiqdemo.iq	A S I Q demmo dot I Q
asiqdemo.iqmsg	A S I Q demmo dot I Q M S G
asiqdemo.iqtmp	A S I Q demmo dot I Q T M P
asiqdemo.log	A S I Q demmo dot log
boot.ini	boot dot in E
CLOB	C lob
dba	D B A
dbisql	D B I sequel
dbisqlc	D B I sequel C
DBLib	D B libe
dbname	D B name
dbname.db	D B name dot D B
dbname.iq	D B name dot I Q
dbname.iqmsg	D B name dot I Q M S G
dbname.iqtmp	D B name dot I Q temp
dbremote	D B remote
dbrsa9.dll	D B R S A 9 dot D L L
dbspace	D B space
dbstop	D B stop
dbtls9.dll	D B T L S 9 dot D L L
default.cfg	default dot C F G
demo	demmo
envvar	E N V V A R
ERRORLEVEL	error level
iAnywhere	I anywhere
jConnect	J connect
mydatabase.db	my database dot D B
mydatabase.log	my database dot log
mydb.db	my D B dot D B

What the screen reader says	How it is pronounced
rsaserver.crt	R S A server dot C R T
sample.crt	sample dot C R T
sp_iqcheckdb	S P underscore I Q check D B
SQL	sequel
SQLCODE	sequel code
SQLCONNECT	sequel connect
sqlpp	sequel P P
SQLSTATE	sequel state
start_asiq	start underscore A S I Q
start_asiq.exe	start underscore A S I Q dot E X E
stop_asiq	stop underscore A S I Q
svrname	S V R name
Sybase	sigh base
SybaseIQAgent	sigh base I Q agent
util_db.ini	U till underscore D B dot in E
utility_db	utility underscore D B

The Sybase IQ program group

This section is for Windows users only. It does not apply to users of UNIX operating systems. Installing Sybase IQ under UNIX does not provide a program group, although all of the following items except the Sybase IQ Service Manager are available for UNIX users.

You can reach the program group by opening the Start menu and choosing Programs→Sybase→Adaptive Server IQ 12.6.

The program group contains some or all of the following items. The items you see depend on the choices you made when installing the software.

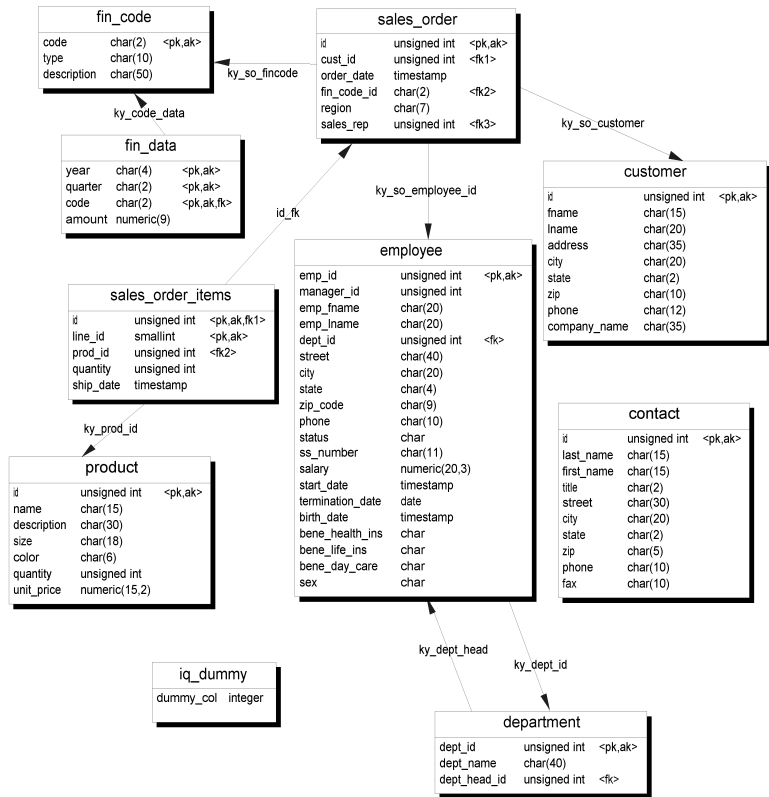
- *Interactive SQL Classic*
Starts the Interactive SQL utility for sending SQL statements to a database.
- *Interactive SQL Java*
Starts the Java-based Interactive SQL utility for sending SQL statements to a database.

- *Raw Device Access*
Adds or removes users and raw devices from the registry.
- *Sample Applications and Projects*
Displays a list of sample applications and projects. For information on running these samples, see *Adaptive Server Anywhere Programming Guide*.
- *Start Sybase IQ Demo Database*
Starts the database server running the sample database.
- *Sybase Central Java Edition*
Starts the database administration utility.
- *Sybase IQ ReadMe File*
Lists last-minute changes to documentation and software and special material that needs emphasis.
- *Sybase IQ Service Manager*
Lets you configure, modify, or delete IQ as a Windows service.
- *Sybase on the Web*
Starts a browser and opens a page with an introduction to Sybase online resources.

The sample database

The sample database included with Sybase IQ contains the following tables.

Figure 1-1: Tables in the sample database



The database is composed of several files named *asIQdemo*, which is stored in the demo subdirectory in your installation directory. Many of the examples throughout the documentation use this database.

The sample database contains internal information about a fictional company (employees, departments, and financial data), as well as product information (products) and sales information (sales orders, customers, and contacts).

Tables in the database

The tables in the sample database include the following:

Table name	Contents
contact	Contacts are held separately from customers.
customer	Information about customers includes names and addresses.
department	Information about departments, such as manager and name.
employee	Information about employees, such as name, salary, and location.

Table name	Contents
fin_code	Each expense and revenue item has a financial code.
fin_data	Quarter-by-quarter financial information about the company.
iq_dummy	A table with exactly one row, useful for extracting information from the database.
IQ_SYSTEM_LOG IN_INFO_TABLE	Used by database administrator. See “Managing IQ user accounts and connections,” Chapter 12, “Managing User IDs and Permissions,” <i>Sybase IQ System Administration Guide</i> .
IQ_USER_LOGIN_ INFO_TABLE	Used by database administrator. See “Managing IQ user accounts and connections,” Chapter 12, “Managing User IDs and Permissions,” <i>Sybase IQ System Administration Guide</i> .
product	Information about products, such as price and quantity available.
sales_order	Information about sales orders, such as date and the sales representative who made the order.
sales_order_items	Each order consists of one or more items. Information about sales order items is held in a separate table.
SYSOPTIONDEFA ULTS	Used by database administrator. See “Managing IQ user accounts and connections,” Chapter 2, “Database Options,” <i>Sybase IQ Reference Manual</i> .

Case sensitivity

The sample database is case insensitive. This means that case is not considered in comparison and string operations. For example, you can type user IDs and passwords in either uppercase or lowercase when using the sample database. Note that, unlike the sample database, any IQ databases you create are case sensitive by default.

Preparing for the tutorial

To prepare for tutorial tasks in subsequent chapters, make sure that Sybase IQ has been installed on your system according to the instructions in the *Sybase IQ Installation and Configuration Guide*.

About this chapter

In the previous chapter, you learned about Sybase IQ and the Sybase data warehouse product family. This chapter describes some basic terms and concepts that will help you work with Sybase IQ.

Sybase IQ and data warehousing terms

Before you delve into Sybase IQ and Sybase IQ documentation, you should familiarize yourself with these terms.

Adaptive Server Anywhere

Adaptive Server® Anywhere is Sybase's transaction-processing relational database management system which can be used standalone or as a network server in a multiuser client/server or three-tier environment. Adaptive Server Anywhere is a component of SQL Anywhere® Studio. It is specifically designed to have a small “footprint” (to use fewer memory and disk resources than the average database management system). Sybase IQ is an extension of Adaptive Server Anywhere, and supports many of the same features.

Anywhere database

In this book, each Adaptive Server Anywhere database is referred to as an **Anywhere database**. Every IQ database uses an Anywhere database for the Catalog Store.

Catalog Store

The portion of each IQ database that contains its metadata. (Metadata describes the layout of the IQ tables, columns, and indexes.) The default name for this file is `dbname.db`.

Component Integration Services (CIS)

Component Integration Services are embedded within Sybase IQ and all Adaptive Server configurations to provide IQ users with direct access to relational or nonrelational databases on the mainframe, UNIX, or Windows servers.

Connection Profile

A **connection profile** stores the information needed to establish a connection to a running IQ server. The profile is primarily used to simplify user connections to a server. IQ extends connection profiles to facilitate starting servers and creating databases. You can create connection profiles using the Connection manager via the Tools menu.

DbSPACE

A **dbSPACE** is a file system file or raw partition used to store data for an Sybase IQ database. If your database runs out of room, you can expand it by adding additional dbSPACES. Users can move IQ data off of disks and take the disks offline without any downtime.

Enterprise database

In this book, each Adaptive Server Enterprise database is referred to as an **Enterprise database**. Adaptive Server Enterprise is Sybase's integrated set of software products for relational database applications.

An Enterprise database is also known as a SQL Server store. You can use Sybase IQ to query data in Enterprise databases.

Freelist

A **freelist** is a structure used by Sybase IQ to track which blocks are in use by a dbspace.

IQ database

Databases that you create using Sybase IQ servers are called **IQ databases**. IQ databases are specially indexed to take advantage of the query speed of Sybase IQ.

Each IQ database that you create includes at least three DBspaces: an IQ Store (for data), a Catalog Store (for metadata), and an IQ Temporary Store (for temporary data). It also generates an IQ Message Log file.

IQ Local Store

The **IQ Local Store** is a portion of a query server that stores data local to that server. Changes to the data are local to the server, allowing users to continue development operations on a query server without affecting the shared main storage.

For details, see “Multiplex capability” in Chapter 2, “Terms and Concepts,” in *Sybase IQ System Administration Guide*

IQ Message Log

A Message Log file created when the first user connects to an IQ database. The default name for this file is dbname.iqmsg. Sybase IQ logs error messages, status messages and insert notification messages in this file.

IQ Store

The portion of each IQ database that contains its data.

The IQ Store is sometimes called the IQ Main Store. All multiplex databases have a shared IQ Store. They may also have IQ Local Store(s) on one or more query servers, but changes to these stores are local to those servers. See “IQ Local Store” on page 15.

IQ Temporary Store

The portion of each IQ database where temporary tables are stored. The database server uses temporary tables for sorting and temporary processing. Data in them persists only as long as you are connected to the database. The default name for this file is `dbname.iqtmp`.

IQ Transaction log

The part of an IQ database that records changes to the database for use in recovery from a system failure. It includes information about versions and free space. The Transaction Log is created in the same directory as the Catalog Store. The default name for this file is `dbname.log`.

Join index

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.

Join indexes are a special type of index used in Sybase IQ. Join indexes may improve response time for queries that join two or more tables. In certain situations, ad hoc queries may be preferred instead of join indexes.

Metadata

Metadata describes the data in your database – for example, the size and datatype of each column in a table. Metadata for each IQ database is stored in the Catalog Store. (See “IQ database” on page 15.)

Multiplex

A Sybase IQ multiplex is an IQ database spread across a shared disk array so that multiple servers run Sybase IQ (and each has its own Catalog Store and IQ Temporary Store, and optional local IQ Store) and all the servers share a common IQ Store. For more information, including all related definitions, see Multiplex capability in Chapter 1, “Overview of Sybase IQ System Administration,” *Sybase IQ System Administration Guide*.

Object

A user-created table, index, or join index. Objects are divided into **persistent objects**, which remain in the database over user disconnects and server restarts, and **temporary objects**, tables and views that only remain in the database during the current session. Permanent tables are also called **base tables**.

Proxy database

By creating a proxy database, you can search data in multiple Adaptive Server Anywhere servers, in Adaptive Server Enterprise databases, and in non-Sybase databases. Conversely, from an Enterprise database, you can create proxy databases that enable you to query your IQ and Anywhere databases.

Synchronize command

The process of updating join index data. You use the SYNCHRONIZE JOIN INDEX statement to do this. The system administrator needs to synchronize Sybase IQ join indexes periodically if multiple users update tables that may be involved in join indexes.

Synchronization

Bringing an outdated query server up to date.

Database computing concepts

This section describes how database applications and the database server work together to manage databases.

Any information system contains the following pieces:

- **A database** Data is stored in a database.
- **A database server** The database server manages the database. No other applications address the database file directly; they all communicate with the database server.
- **A language interface** Applications communicate with the database server using an interface. You can use JDBC, Sybase Open Client, or Embedded SQL.

The language interface provides a set of function calls available to client applications for communicating with the database. For JDBC, the library is commonly called a driver. The interface is typically provided as a shared library on UNIX operating systems or a dynamic link library (DLL) on PC operating systems. The JDBC interface uses the Sybase jConnect driver, which is a zip file of compiled Java classes.

If you are working with a Sybase IQ network server, the language interface resides on the client computer.

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

If you develop an application using a rapid application development (RAD) tool such as one of the Sybase Enterprise Application Studio™ tools, 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 do use one of the supported language interfaces.

Sybase IQ database files

When you create a database, Sybase IQ creates a number of files.

Table 2-1: Database files

Physical file name	Internal name	Description	Number created
dbname.db	Not applicable	Anywhere dbspace file. Part of the Catalog Store.	One or more per server
dbname.log	Not applicable	Anywhere transaction log. Part of the Catalog Store.	Only one per server
dbname.iq	IQ_SYSTEM_MAIN	IQ dbspace file for the main IQ Store. (User specifies the names for additional files).	One or more per database
dbname.iqtmp	IQ_SYSTEM_TEMP	IQ temporary dbspace file.	One or more per database
dbname.iqloc	IQ_LOCAL	IQ dbspace file for main IQ store(s) local to a particular query server. This file is not created automatically.	Optionally, one or more per query server. Multiplex databases only.
dbname.iqmsg	IQ_SYSTEM_MSG	IQ human-readable trace file of debugging output.	One per database

Note: The database administrator may override the default file names and extensions shown.

You can create a database using a relative or fully qualified pathname for each of the files for the database. If you omit the directory path, Sybase IQ creates the files as follows:

- The Catalog Store is created relative to the working directory of the server.

- The IQ Store is created relative to the working directory of the server.
- The Temporary Store is created in the same directory as the Catalog Store. (This also occurs if you do not specify any file name.)
- The Message Log is created in the same directory as the Catalog Store. (This also occurs if you do not specify any file name.)
- The Transaction Log is created in the same directory as the Catalog Store. (This also occurs if you do not specify any file name.)

Note For best performance, you should place the Transaction Log on a different physical device from the Catalog Store and IQ Store, because they are randomly accessed files and the transaction log is a sequentially accessed file.

The Main and Temp stores contain most of your tabular data. Each database has its own unique files for temporary data. If you don't specify a file, IQ creates one automatically for use as a temporary storage space. For details about creating a temporary storage area, and what happens if the storage area defined is insufficient, see *Sybase IQ Installation and Configuration Guide*.

You can manage the contents of these files using Sybase Central, as described in Chapter 4, “Managing Databases.”

Sybase IQ includes a sample database named *asiqdemo* for you to work with. The files that comprise the database are located in the *demo* subdirectory of your installation directory.

How the pieces fit together

A complete application in a client/server environment includes a database server with one or more client applications.

The database server may be a network server, which supports network communications. No changes are needed to the client application for it to work in a client/server environment.

SQL and database computing

When a client application wants to carry out a database task, such as retrieving information or inserting a new row into a table, it does so using Structured Query Language (SQL) statements. SQL is pronounced “sequel”.

Depending on how a client application is developed, SQL statements can be supplied in function calls from the programming language, or you may build them graphically in a special window provided by the application development tool.

The programming interface delivers the SQL statement to the database server. The database server receives the statement and executes it, returning any required information (such as query results) back to the application.

Client/server communication protocols carry information between the client application and the database server. Programming interfaces define how an application sends the information. No matter what interface and network protocol you use, it is SQL statements that are sent to a server, and the results of SQL statements that are returned to the client application.

Relational database concepts

A relational database management system (RDBMS) is a system that stores and retrieves data in table format. A relational database consists of a collection of tables that store interrelated data.

This section introduces some of the terms and concepts that are important when talking about relational databases.

Database tables

In a relational database, all data is contained 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 number, a sequence of characters (for text), or a date. Each row in the table has at most one value for each column. Where there is no value for a particular row and column, we sometimes say that the value is NULL, which may be interpreted as not currently known, or not applicable, or not yet available.

An example of a table containing employee information may appear as follows:

emp_ID	emp_lname	emp_fname	emp_phone
10057	Huong	Zhang	1096
10693	Donaldson	Anne	7821

Characteristics of relational tables

The tables of a relational database have some important characteristics:

- There is no logical significance to the order of the columns or rows. However, the column order does affect a few special queries. Likewise, the row order does affect the order in which some queries return the row, and can affect the performance of queries. If you care about the order of the rows, then the query should specify the order in which the rows are to be returned. If a particular order is very common and all other orders are uncommon, it may be best to keep the rows in that particular order.
- Each row contains either no value (a NULL column) or contains one and only one value for each column.
- Each value for a given column is of the same type.

The following table lists some of the formal and informal relational database terms describing tables and their contents, together with their equivalent term in other nonrelational databases. This manual uses the informal terms.

Formal relational term	Informal relational term	Equivalent nonrelational term
Relation	Table	File
Attribute	Column	Field
Tuple	Row	Record

What do you keep in each table?

When you are designing your database, you should make sure that each table in the database holds information about a specific thing, such as employees, products, or customers.

A relational database is not a set of unrelated tables. You can use primary and foreign keys to describe relationships between the information in different tables.

Primary and foreign keys

Primary and foreign keys enable each row in the database tables to be identified, and enable relationships between the tables to be defined. These keys define the relational structure of a database.

Each table has a primary key

Each table in a relational database may or may not have a primary key. The **primary key** is a column, or set of columns, that allows each row in the table to be uniquely identified. No two rows may have the same value of a primary key.

You can select a primary key from all of the available columns. It is good practice to keep your primary key for each table as compact as possible. If possible, the primary key should be an unsigned integer data type, rather than char or varchar.

Examples

In a table holding information about employees, the primary key may be an ID number assigned to each employee.

In the sample database, the table of sales order items has the following columns:

- An ID, identifying the order to which the item belongs.
- A line ID, identifying each item on any order.
- A product ID, identifying the product being ordered.
- A quantity, showing how many items were ordered.
- A ship date, showing when the order was shipped.

In order to identify a particular item, both the ID and the line ID are required. The primary key is made up of both these columns.

Tables are related by foreign keys

The information in one table is related to that in other tables by foreign keys.

Example

The sample database has one table holding employee information and one table holding department information. The department table has the following columns:

- *dept_id*

An ID number for the department. This is the primary key for the table.

- *dept_name*
A column holding the name of the department.
- *dept_head_id*
The employee ID for the department manager.

To find the name of a particular employee's department, there is no need to put the name of the employee's department into the employee table. Instead, the employee table contains a column holding the department ID of the employee's department. This is called a foreign key to the department table. A **foreign key** references a particular row in the table containing the corresponding primary key or unique constraint. The primary key and unique constraint so referenced are known as the **candidate key**.

In this example, the employee table (which contains the foreign key in the relationship) is called the foreign table or referencing table. The department table (which contains the referenced primary key) is called the primary table or the referenced table.

Other database objects

A relational database holds more than a set of related tables. Among the objects that make up a relational database are:

Object	Description
Indexes	<p>Indexes allow quick lookup of information. 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.</p> <p>Indexes are an important design element for high performance.</p>
Java Objects	<p>You can install Java classes into the Catalog Store of your database. Java classes provide a way of building logic into your database.</p> <p>For details about Java data type support in IQ, see <i>Sybase IQ System Administration Guide</i>.</p>

Object	Description
Procedures & Functions	<p>These are routines held in the database itself that act on the information in the database.</p> <p>You can create and name your own stored procedures to execute specific database queries and to perform other database tasks. Stored procedures can take parameters. For example, you might create a stored procedure that returns the names of all customers who have spent more than an amount you specify as a parameter in the call to the procedure.</p>
Users & Groups	<p>Each user of a database has a user ID and password. You can set permissions for each user, so that confidential information is kept private. Users can be assigned to groups, in order to make the administration of permissions easier.</p>
Views	<p>Views are computed tables, or virtual 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.</p> <p>The tables that actually hold the information are sometimes called base tables to distinguish them from views.</p>

For a complete list, see “Viewing other database objects” on page 39.

Queries

You query or retrieve data from a database with the SQL `SELECT` statement. The basic query operations in a relational system are selection, projection, and join. The `SELECT` statement implements all of them.

Projections and restrictions

A **projection** is a subset of the columns in a table. A **restriction** (also called **selection**) is a subset of the rows in a table, based on some conditions.

For example, the following `SELECT` statement retrieves the names and prices of all products that cost more than fifteen dollars:

```
SELECT name, unit_price
FROM product
WHERE unit_price > 15
```

This query uses both a restriction (`WHERE unit_price > 15`) and a projection (`SELECT name, unit_price`)

Joins

A join links the rows in two or more tables by comparing the values in key columns and returning rows that have matching values. For example, you might want to select the order item identification numbers and product identification numbers for all order items for which more than a dozen items were shipped. If you created a join index on `sales_order_items.id` and `product.id`, Sybase IQ would use the join to speed up queries like the following:

```
SELECT sales_order_items.id, product.id
FROM product, sales_order_items
WHERE sales_order_items.quantity > 12
```

Other SQL statements

You can do more with SQL than just query. SQL includes statements that create tables, views, and other database objects. It also includes statements that modify tables (the insert and delete statements), and commands that perform many other database tasks discussed in this manual.

The system tables

Every database contains a set of **system tables**, which are special tables used by the system to manage data and the system. These tables are also sometimes called the **data dictionary** or the system catalog. In IQ they are stored in the Catalog Store.

System tables contain information about the database. You never alter the system tables directly in the way you can alter other tables. The system tables hold information about the tables in a database, the users of a database, the columns in each table, and so on. This information is data about data, or **metadata**. You can use the system tables to check the size and data type for various columns, prior to inserting data.

Running and Connecting to Servers

About this chapter

This chapter includes tutorials on starting, stopping, and connecting to servers.

Understanding database servers

This section describes the basic components of a Sybase IQ system as a basis for using the product and performing tutorial tasks.

Access to your database must be made through a **server**, which provides a communications channel and a manipulation device for the database. An IQ server can dynamically start and stop a database, and accept connections from applications or users running on the same machine or on other computers by means of the network that links the two machines. Users can have connection rights to a database, not to the server. Sybase strongly recommends that you manage only one database per server.

You can use multiple Sybase IQ servers to combine the use of IQ, Anywhere, and Enterprise databases. You can also use multiple Adaptive Server Enterprise servers to combine the use of IQ and Enterprise databases.

A Sybase IQ multiplex is an IQ database spread across a shared disk array. Each server in a multiplex runs Sybase IQ. Each set of an IQ Temporary Store and Catalog Store make up one server, and the servers share a common IQ Store. (Servers may also, if desired, have local IQ Stores of privately managed data.) Users with large query loads and shared disk arrays may create an IQ multiplex to reduce I/O requirements and gain additional CPU power and memory space. A Sybase IQ multiplex supports many users, each executing complex queries against the shared database.

You can manage all Sybase IQ servers using one tool — Sybase Central.

Sybase Central and database management

Sybase Central is a database management tool installed with Sybase IQ. You can use it to start Sybase IQ servers and create databases. It exposes database settings, properties, and utilities in a graphical user interface.

Note You must use Sybase Central version 4.3, which ships with Sybase IQ 12.6, to administer Sybase IQ 12.6 servers and databases. You must use Sybase Central version 3.2, which ships with earlier versions of Sybase IQ, to administer earlier versions.

Database administration tasks typically fall into two categories:

- Tasks carried out by sending SQL statements to the database engine.
- Tasks carried out by Sybase IQ utilities.

Sybase Central provides an interface for both kinds of tasks.

Data definition tasks that you can carry out with Sybase Central can also be carried out using DBISQL to send SQL statements to the database engine or server. (See the *Sybase IQ Reference Manual* for SQL statement syntax.) However, since administrating a multiplex requires managing multiple servers or message agents on multiple systems sharing a single IQ Store, you must coordinate some activities. *Therefore Sybase recommends that certain functions, such as creating multiplex databases and query servers, be performed using Sybase Central.* See Chapter 5, “Working with Database Objects,” in the *Sybase IQ System Administration Guide* for details.

For more information about Sybase IQ utilities, see Chapter 3, “Database Administration Utilities,” in the *Sybase IQ Utility Guide*.

Note This version of Sybase IQ supports only Sybase Central Java Edition, which is shortened to “Sybase Central” throughout this book. Although Sybase Central Java Edition can also be used to manage other Sybase databases, this chapter describes only how to use Sybase Central Java Edition with Sybase IQ databases. Please see your product documentation for information about Sybase Central support.

Starting and stopping Sybase Central

This tutorial introduces the Sybase Central user interface. It describes how to start Sybase Central, how to start a Sybase IQ server, how to connect to a database, and how to view a database schema in Sybase Central.

After completing the tutorial you should feel comfortable exploring Sybase Central's capabilities by yourself.

❖ Starting Sybase Central on UNIX systems

1 Source the ASIQ-12_6.csh (or .sh) script. (Always source this file before invoking utilities like Sybase Central or the IQ Agent.)

2 If you plan to administer multiplex database(s), type:

```
% $ASDIR/bin/S99SybaseIQAgent1260
```

3 Type this command:

```
% scjview
```

Note On UNIX systems, for proper display in Sybase Central Java or dbisql, run with the default colors of the Common Desktop Environment. If you run under Open Windows or change the default colors, you may experience display problems.

❖ Starting Sybase Central on Windows systems

- From the Start menu, choose Programs > Sybase > Adaptive Server IQ 12.6 > Sybase Central Java Edition.

The Sybase Central interface

Starting Sybase Central opens the main Sybase Central window, which is similar on UNIX and Windows systems.

The main window is split into two vertically-aligned panels. The left panel displays a hierarchical view of database objects or containers in a treelike structure. A **container** is a database object that can hold other database objects, including other containers. Sybase Central is at the top, with contained objects indented beneath it.

Plug-ins for Sybase Central, such as the Sybase IQ database management system, occupy the first level after the root level. A **plug-in** is a graphical tool for managing a particular product. When you install the product, the plug-in automatically installs. When you next start Sybase Central, the new product automatically “plugs in” to Sybase Central and appears in the main window.

Once you connect to a server, the right panel displays the contents of any container that you select. To select a Sybase IQ object, click the Sybase IQ object in the left panel, or double-click the object in the right panel.

Registering a plug-in

If you do not see the plug-in for Sybase IQ in the main Sybase Central window, you can load it manually.

❖ Loading a plug-in

- 1 Select Tools > Plug-ins.
- 2 If Sybase IQ is listed, select it, then click Register. If not, select Load.
Use the Browse button to find and select the file *\$ASDIR/java/iq.jpr*.
- 3 Click OK.

Stopping Sybase Central

To stop Sybase Central, select File > Exit. Do not stop Sybase Central at this time. You will need it running to continue the tutorial in the following section.

Running the IQ Agent

The IQ Agent must be running in order to do many multiplex operations (such as creating query servers) on Sybase Central. The IQ Agent generally is started whenever you reboot a Windows machine where IQ is installed. If you are planning to administer a multiplex, you may want to verify that the IQ Agent is running. For more about the IQ Agent, see “Specifying the Agent Port” on page 35.

❖ Checking the IQ Agent on Windows

- 1 Choose Control Panel > Administrative Tools > Services.

- 2 Select “Sybase IQ Agent.”
- 3 If Status does not say “Started,” right-click the agent and choose Start. If Status says “Started,” click the menu bar and choose Close.

❖ **Checking the IQ Agent on UNIX**

- 1 Run the `stop_asiq` utility with the `-agent` option:

```
% stop_asiq -agent
```

- 2 If no running agent is owned by your username, change directory to `$ASDIR/bin` and run `S99Sybase \IQAgent1260`.

Copying the sample database

Make a copy of the sample database so that you can restore it after making changes. Do this at the operating system level. It cannot be done within Sybase Central.

❖ **Copying the sample database**

- 1 From the operating system, navigate to the directory that contains the sample database file, `asiqdemo.db`.

Its default directory is `$ASDIR/demo` on UNIX, `%ASDIR%/demo` on Windows.

- 2 Create a subdirectory, for example, `mydemo`, to hold the copy.
- 3 Use UNIX `cp` or Windows copy and paste (Ctrl+C, Ctrl+V) to create a copy of the database in your backup directory.
- 4 Work with the copy.

Starting the database server

This section describes how to start a database server running the sample database. For complete information on starting servers, see *Sybase IQ System Administration Guide*.

Note If your system has multiple users or is one of several machines in a subnet, edit the file `$ASDIR/demo/asiqdemo.cfg` to specify a unique port and server name. Change the arguments to the `-n servername` switch and the `-x 'tcpip{port=#####}` to specify a server name and port number that are not in use. Avoid the port number 2638, which is also the default port for Adaptive Server Anywhere, if that product is installed on your system.

❖ Starting a server running the sample database from Windows

- 1 Log in to your host machine as the user DBA, or using an account with DBA privileges.
- 2 From the Start menu, choose Programs > Sybase > Adaptive Server IQ 12.6 > Start Sybase IQ Demo Database.

This starts a server running the sample database. The server window appears briefly, then disappears, after which the server appears as an icon in the system tray. The server name is your system name followed by an underscore and `asiqdemo`.

❖ Starting a server running the sample database from the command prompt (UNIX or Windows)

- 1 Log in as the user DBA, or using an account with DBA privileges
- 2 Change to a directory where you have write permission and make a copy of the `$ASDIR/demo` directory there (`%ASDIR%demo` on Windows).
- 3 At the command prompt, type the following command:

```
start_asiq @asiqdemo.cfg asiqdemo.db
```

This starts a server running the sample database.

❖ Starting a server running the sample database with Sybase Central

The following steps assume that you have started Sybase Central and are running the IQ plug-in.

- 1 Select the Sybase IQ plug-in in the left pane.

- 2 Click the Utilities tab in the right pane, or press Alt+T, I, S.
- 3 Double-click Start Server or tab into the container, then press S.
- 4 Follow the instructions in the Start Server Wizard.

The following sections describe the Start Server Wizard's screens and your options.

Start Server Wizard: Options

The wizard can start a single server or all the servers in an IQ multiplex. For this tutorial, leave the Single server option (the default) checked, because the sample server is a single one. For details about **multiplex** capability, see "Multiplex capability" in Chapter 1, "Overview of Sybase IQ System Administration," of the *Sybase IQ System Administration Guide*.

Click Next or tab to Next and press the spacebar to move to the next panel and supply connection parameters.

Start Server Wizard: Connection Parameters

On the Connection Parameters screen, you may tab through the parameter fields or type the key combination from the following table to select a particular field. On forms such as wizard screens, each field is identified by an underlined letter. This letter is the shortcut key (fast key). The shortcut keys are different on each screen, and may change in a new release.

Table 3-1: Connection parameter shortcut keys

Key combination	Parameter
Alt+a	Password
Alt+C	Connection Profile
Alt+D	Database Name
Alt+H	Host Machine
Alt+P	Port Number
Alt+S	Server Name
Alt+U	Username

IQ optionally saves name parameters in a Connection Profile to simplify future connections to a database. Once you create a Connection Profile, you can specify the profile name and it fills in all the parameter fields for you. For this tutorial, leave Other selected in the Connection Profile box.

For this tutorial, type `asiqdemo_` followed by your machine name for the Server Name.

For the Host Machine (Alt+H), type your host name or specify localhost.

The Port Number (Alt+P) defaults to 2638, the default Sybase port number. In general, always change the Port Number to one that is not in use.

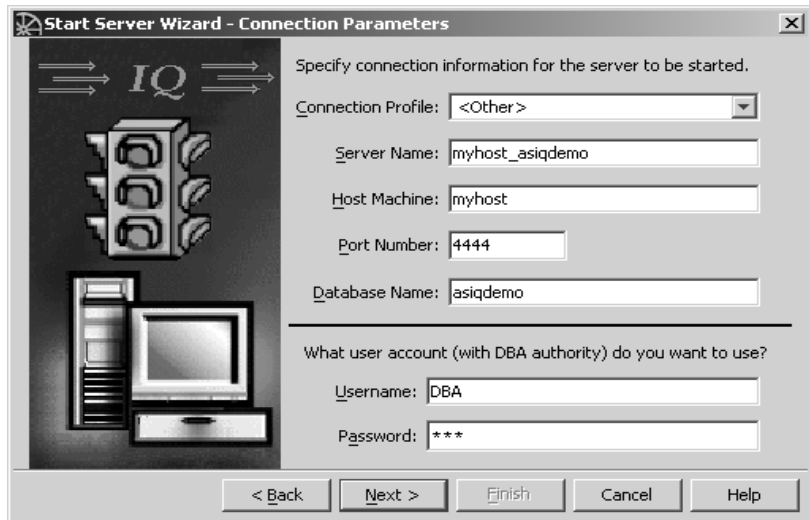
For the Database Name (Alt+D), type `asiqdemo`.

The default Username is sufficient; changing the Username from DBA to another user limits functionality.

For the Username (Alt+U), type DBA.

For the Password (Alt+P), type SQL.

Figure 3-1: Start Server Wizard - Connection Parameters Screen



Click Next or Alt+N when finished, to move to the next panel and supply the database path.

When you press Next, Sybase Central checks the host:port combination and verifies that an agent is running on the given host. If the agent is not found, but you know it is running, you may need to specify its port. If the host:port combination is already in use, Sybase Central returns an error.

Specifying the Agent Port

The Agent Port is used for communications between the IQ Plug-in and the IQ Agent, which must be running on the given host. The Agent Port number should be different from the server's Port Number. To view port numbers:

- On Windows, use the Event Viewer in Control Panel > Administrative Tools.
- On UNIX, use the `stop_asiq -agent` command.

The Start Server Wizard looks for the agent running on the default port (1099) first, then the value of the environment variable `ASIQPORT`, if set, and then checks to see if the user has registered a port with the server name. If none of these supplies a usable port number for the Agent, the Start Server Wizard displays the Agent Port field on the Connection Parameters form for you to fill in.

You can start Sybase Central Java Edition with the runtime setting, `-DASIQPORT=####`, and this value is checked after the default 1099. If you don't specify a value, Sybase Central defaults to `ASIQPORT`, if set.

Start Server Wizard: Database Path

To specify the database path, tab to the field, click the text box, tab, or press `Alt+D` and type the path, or use the Browse button or `Alt+r` to locate the database if needed. Browse is only enabled if the host specified in the previous screen is the local host. The Browse button can locate either a directory or a file.

Be sure to include the `.db` suffix when typing the database path.

For example, on Windows:

```
C:\Program Files\Sybase\ASIQ-12_6\demo\asiqdemo.db
```

On UNIX:

```
/s1/dept40/SybaseIQ/demo/asiqdemo.db
```

A UNIX path cannot contain blanks. Using "Sybase IQ" instead of "SybaseIQ" in the path would invalidate it.

Note You may find it convenient to arrange drive letter assignments (on Windows systems) or symbolic links (on UNIX systems) to match paths and names across the hosts in the multiplex.

If the database is strongly encrypted, type the encryption key.

Sybase Central uses the `params.cfg` file each time it starts the database, to specify options for the session. The options specify run time values for environment settings. For details about startup options, see the *Sybase IQ Reference Manual*. If there is no `params.cfg` file in the path you provided, the wizard creates one.

Click Next or Alt+N when finished to view the Start Server Wizard Summary.

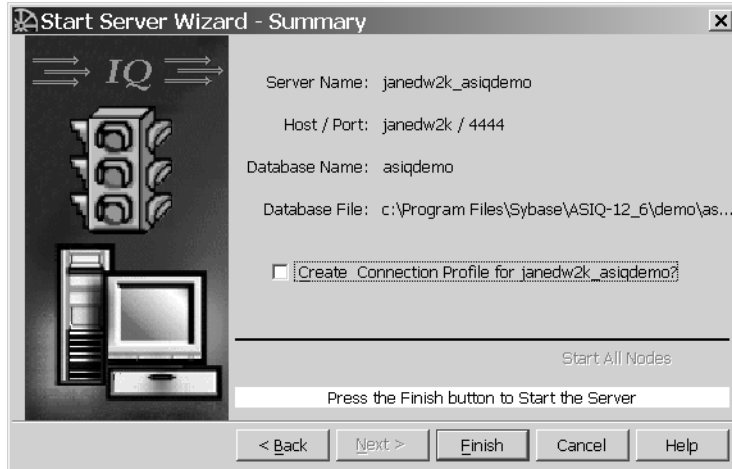
Start Server Wizard: Summary

The Summary screen displays the start parameters you specified.

If you did not specify a Connection Profile, the wizard prompts you to create one. Connection Profiles make subsequent connections easier.

Click the radio button at the bottom of the screen, or type Alt+C to create a profile with the provided information.

To change information, click Back to return to a previous screen.



To start the server, click Finish, or tab to the Finish button and press the spacebar. The wizard updates the status line while starting your server.

Navigating the main Sybase Central window

Follow the steps in this tutorial to view tables, columns, indexes, joined tables and users in the IQ database.

Connecting to a database

Once you have started a server, you can connect to databases on it.

Connect dialog

The Connect dialog lets you connect to servers and databases using Sybase Central and Interactive SQL.

❖ **Opening the Connect dialog**

- 1 Press F11 or select Tools > Connect from the Sybase Central menu or type Alt+T, C.
- 2 Supply connection parameters at the dialog prompts.

Connection Profiles

The Connection Profiles dialog lets you name a set of connection parameters and specify them by name to connect easily to databases you use frequently.

❖ **Creating a Connection Profile**

- 1 Press F9 or select Tools > Connection Profiles from the Sybase Central menu or type Alt+T, P.
- 2 Click New and supply parameters.

Viewing a database schema

A database schema is the collection of all objects in the database. Sybase Central displays a database schema as a hierarchy of containers and their contents. This section describes how to view the schema of a database.

Expanding a database container

There are a variety of methods for viewing the objects in a database, including the following:

- Click a container in the left panel to select that container. The right panel then shows the contents of the selected container. Among the contents of the server container are all the databases you have attached to, including those you have not yet connected to by entering a user ID and password.

- Click the left and right arrow keys to expand or collapse the hierarchical tree of objects. This allows you to view database objects at levels below the level of the currently selected container. If no plus or minus sign appears next to a container, it contains no objects extending beyond its own level.
- Double-clicking on the container in the left panel expands or contracts the object's hierarchy in addition to selecting it and displaying its contents in the right panel. You can use the View/Choose Columns menu option (Alt+V, C) to customize the columns displayed.

Note If you cannot display certain fields, use the facilities your windowing system provides to change the Sybase Central display to use dark text on a white or light background.

After you have opened an object deep in the hierarchy, for example, a column, the left panel reflects all the items above it.

The left panel displays container objects only. The right panel displays the contents of the container object selected in the left panel. For example, if you select the Sybase IQ plug-in in the left panel, the right panel displays servers, utilities, and services for IQ.

Viewing the tables in a database

The following steps show how to examine the contents of a table folder in a database container.

❖ Viewing tables in a database

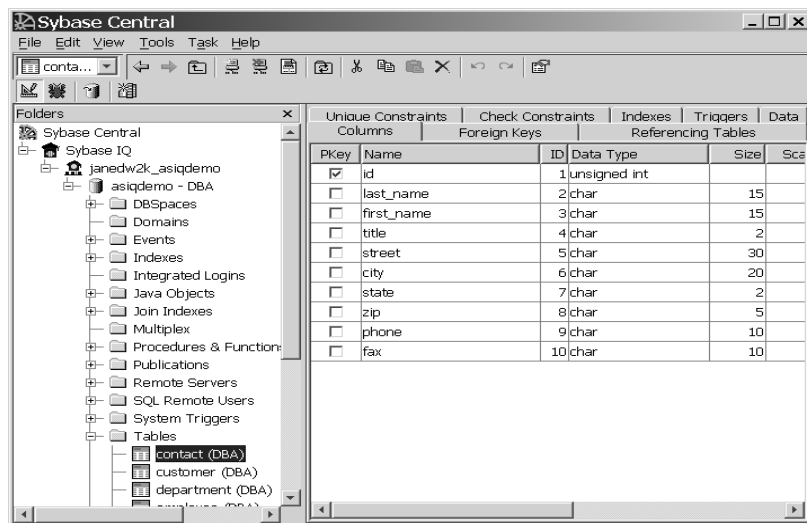
- 1 Select the Tables folder. You may have to expand the server and database objects in the hierarchical tree in the left panel in order to view the Tables folder.

If the database container is open, pressing a “T” selects the tables container. Pressing the first letter or any item in a container opens that item in the list of items.

- 2 Each table in the Tables folder is itself a container. Select a table in the left panel to reveal the contents of that table in the right panel. Each table object contains folders for columns, foreign keys, relations, indexes, and more.

- 3 Expand the table object in the left panel to reveal its contents in the hierarchical tree. Select each object in the table container by clicking on it once in the left panel. The right panel displays the contents of the table object selected in the left panel.

Figure 3-2: Sybase Central object hierarchy for a Sybase IQ database



Viewing other database objects

You can navigate a database by clicking or double-clicking in either panel or use the arrow keys or letter keys (such as ‘T’ for table). Explore the contents of the other folders in the database.

Every Sybase IQ database contains individual folders for the following objects:


Object	Description
Dbspaces	Additional .db, .iq, or .tmp files for the database.
Domains	User-defined data types.
Events	An action, such as a stored procedure, scheduled to be carried out at a certain time.
Indexes	Structures that store and represent data so that it can be used for query processing. Indexes allow quick lookup of information. 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.

Object	Description
Integrated Logins	Login feature that allows use of the same single user ID and password for operating system logins, network logins, and database connections.
Java Objects	Java programming objects including packages, JAR files, and Java classes. You can install Java classes into the Catalog Store of your database. Java classes provide a way of building logic into your database.
Join Indexes	Tables with join indexes evaluated when accessed.
Multiplexes	IQ databases spread across a shared disk array, to optimize support for many users with large query loads.
Procedures & Functions	SQL procedures for a module-based language.
Publications	Identifies replicated data in SQL Remote.
Remote Servers	Servers connected to the local server by means of a network.
SQL Remote Users	Query or write servers that receive changes from other servers in a multiplex.
System Triggers	Triggers that modify the system tables.
Tables	Base tables stored in the database
Triggers	Special forms of stored procedure that execute only when a user runs a query that modifies the data.
Users & Groups	Categories for administering who is permitted to use the database.
Views	Computed tables, stored in the database as a query and evaluated when accessed.

Explore the sample database until you are comfortable locating database objects in the Sybase Central main window.

Disconnecting databases and users

❖ Disconnecting users from a database

- 1 Select the database and click the Connected Users tab. For this tutorial, use the asiqdemo database.
- 2 Right-click the username.
- 3 Choose Tools > Disconnect, click the disconnect toolbar button  , or right-click the database container and choose Disconnect from the submenu.

If multiple databases are connected, the Disconnect dialog box lists the database name and the server (*host name_database name*) for all connected databases. If only one database is connected, it is automatically disconnected.

- 4 Select the database from which to disconnect.
- 5 Choose Disconnect.

Connecting to databases from DBISQL

Once a database server is running, you can connect to it from an application. In this example, we'll connect to the sample database, *asiqdemo*.

❖ Connecting from the command prompt

- 1 Ensure that the sample database is loaded on a running server, as described in “Starting the database server”.
- 2 Open a command prompt on your UNIX or Windows system.
- 3 Start DBISQL by typing the following:

```
dbisql -c "uid=DBA;pwd=SQL" -host fiona -port 1870
```

The `-c` parameter specifies connection parameters. For a complete list of connection parameters, see the *Sybase IQ System Administration Guide*, Chapter 4, “Connection and Communication Parameters.”

Note The default user ID and password for the *asiqdemo* database and other IQ databases are “DBA” and “SQL.” If you have trouble connecting to a database, try changing the case of the user ID and password that you supply; the case sensitivity for that database may be set ON.

❖ Connecting from a Windows program group

- 1 Start the server, as described in “Starting the database server”
Select Start > Programs > Sybase > Adaptive Server IQ 12.6 > Interactive SQL Java

The DBISQL logon window appears.

- 2 On the Identification tab, type the user ID

```
DBA
```

and the password

SQL

This is the default user ID and password for Sybase IQ databases when they are created. (The password does not appear when you type it. This prevents someone else from seeing your password.)

On the Identification tab, you may also specify connection parameters stored in an **ODBC data source** to connect a client application. For more information about creating an ODBC data source and connecting using ODBC data sources, see “Working with ODBC data sources,” *Sybase IQ System Administration Guide*.

- 3 Click the Database tab and type the server name (for example, “*hostname_asiqdemo*” for the demo database). If the server is remote, type *host:port* instead of the server name.
- 4 Click the Advanced tab and choose a JDBC driver, for example, jConnect5.

On this tab, you can also type connection parameters, for example:

```
"links=tcip{host=name,port=####}"
```

This information could be substituted for the server name on the Database tab.

- 5 Click OK to connect to the database.

If Interactive SQL cannot connect to your database, click the Database tab and type “*host=servername:nnnn*” where *servername* is the name of your system and *nnnn* is your port number. (The default port number is 2638, but if the host was started with a different number, use that instead.)

Other applications connect to the database server in much the same way. To make the connection, they must submit both user ID and password.

The *Sybase IQ Utility Guide* gives more details about how to use DBISQL.

Sending commands to the database

Once you are connected to the database, you can send commands to it. For example, type the following line in the SQL Statements window.

```
SELECT * FROM employee;
```

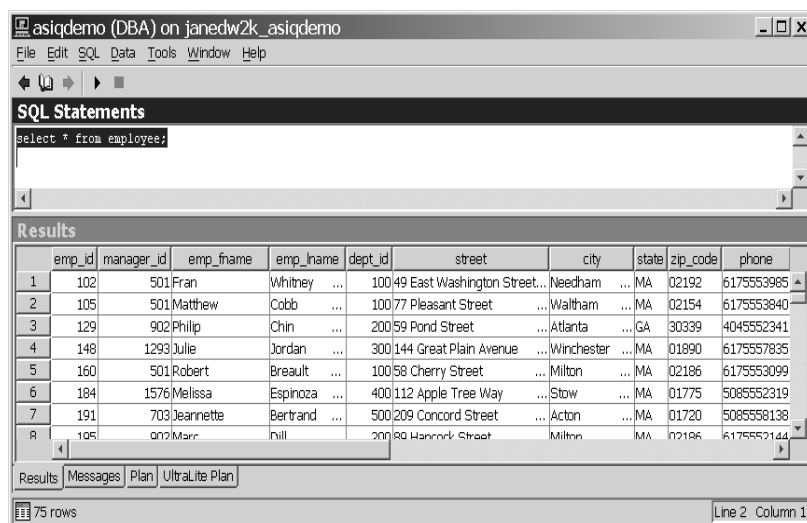

Execute your command using F9 or Alt+S, X.

For more details about executing SQL statements, see Chapter 2, “Using Interactive SQL (dbisql),” in the *Sybase IQ Utility Guide*.

The example SQL statement that you just executed displays all (*) the columns of the table named `employee`. This table contains information about the employees of a fictitious merchandising company.

DBISQL returns output from your statement in the Results area at the bottom of the screen.

Figure 3-3: Interactive SQL window



DBISQL retrieved the information by sending a request to your database server. The server, in turn, looked up the information in the `employee` table and returned it to DBISQL.

Stopping a database server

You can now shut down your database server.

❖ Stopping the database server

- 1 In Sybase Central, right-click the server name.

2 Choose Stop Server from the submenu.

By default, DBA permission is required to shut down the network server, and all users can shut down a personal server.

Normally, you should not shut down a server while it is still connected to one or more clients.

Shutting down
multiplex servers

To shut down servers in an IQ multiplex, open the Multiplex folder, select the server to stop and choose File > Server > Stop (Alt+F, v, o). Alternately, right-click the write server and choose Server > Stop. You may then choose which server(s) to stop.

In the following chapter, you will learn how to create and manage databases on your server.

About this chapter

In the previous chapter, you learned how to start and connect to servers. This chapter walks you through database administration tasks — creating, viewing, and modifying your database and its contents.

Managing databases

To see what actions you can perform on an object in Sybase Central, right-click the object. You can use the resulting submenu, for example, to view properties or launch creation wizards. Selecting an object within a database also displays a creation wizard toolbar button.

In this tutorial, we will use Sybase Central to create a new database, view database properties, and delete databases.

Note You can use Ctrl+C (Cut) and Ctrl+V (Paste) anywhere in the Sybase Central wizards.

Creating databases

To create a database, select the IQ plug-in, click the Utilities tab, and double-click Create Database in the right panel or choose Tools > Sybase IQ > Create New Database.

Multiplex databases

To create a multiplex database, create a database, then enable it for multiplex by creating a query server. For details, see *Sybase IQ System Administration Guide*.

Specifying server options

Create SQL Script

In the Create Database - Options dialog, selecting the option, Create SQL script (Alt+C), Sybase Central generates a script of the CREATE DATABASE statement that creates your database once you complete the wizard. For example:

```
SET TEMPORARY OPTION escape_character="Off";
CREATE DATABASE 'd:\mpx\fiona.db'
TRANSACTION LOG ON
COLLATION 'ISO_BINENG'
CASE RESPECT
PAGE SIZE 4096
BLANK PADDING ON
IQ PATH 'fiona.iq'
IQ SIZE 10
IQ PAGE SIZE 131072
TEMPORARY PATH 'fiona.iqtmp'
TEMPORARY SIZE 10
;
```

Sybase Central names this script *<dbname>.create.sql* and places it in the database directory.

Start the new database automatically

An IQ server is required in order to create an IQ database, although you do not need to *start* an IQ server explicitly before creating a database. If you have no active server, Sybase Central will start the Utility Server automatically.

Select the box to choose the server to create the database, or click Alt+A for the active IQ server or Alt+U for the Utility Server. The Utility Server is a special server for performing administration tasks when there is no active server.

To start the new database automatically, click the checkbox or press Alt+S. The IQ Agent must be running.

If you choose to create the database but not start it, Sybase Central starts the Utility Server just long enough to create the database, then shuts it down.

What engine will create the database?

Note On Windows systems, the IQ Agent starts automatically when the machine is rebooted right after installation and at each subsequent reboot. On UNIX systems, you can configure it to start automatically also. For more information, see Chapter 2, “Running Sybase IQ” in the *Sybase IQ System Administration Guide*.

Click Next.

Setting connection parameters

To connect to Sybase IQ databases, you need to specify certain parameters:

- Host and port number for the server
- Username and password for the user account

Starting the database automatically

Note You may specify these parameters in a Connection Profile. For details, see “Creating connection profiles” in *Sybase IQ System Administration Guide*.

On the Connection Parameters screen, the Username defaults to:

DBA

This identifies a user with full administration and resource creation rights.

The password for this user defaults to:

SQL

By default, all newly-created Sybase IQ databases contain this user ID and password. The database administrator is responsible for providing the desired level of security by changing passwords and creating other user IDs. *For security reasons, Sybase strongly recommends that you change the DBA password from the default.*

To change the DBA password, you must have DBA authority.

To change the DBA password using dbisql.

```
GRANT CONNECT TO DBA IDENTIFIED BY new_password
```

Unless there is only one server per system, *always change the default port number for the host machine*. You can specify any port number that is not already in use. To list port numbers in use, use the *netstat* or *dblocate* command. Port numbers are listed after the host name in the Local Address column.

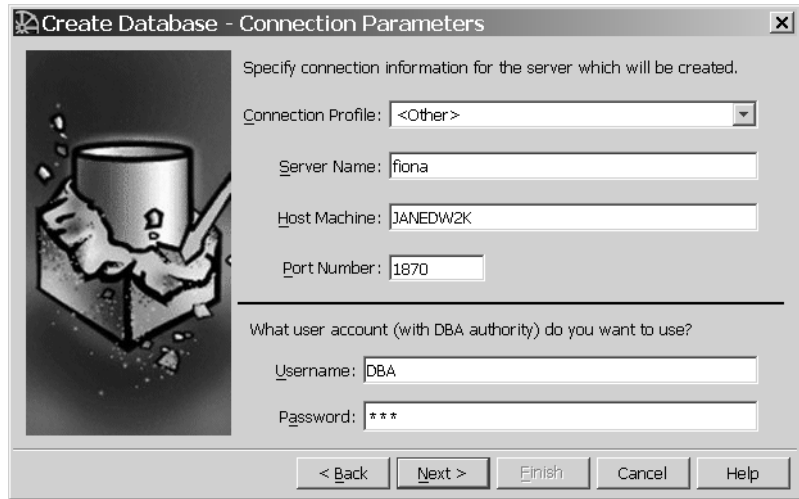
If a connection profile exists for the server, you may instead choose the server from the Server Lookup dropdown menu. Click Next to continue.

Manual startup with Utility Server

If you chose not to start the database automatically, type the password for the DBA user and check that the host machine and port number are correct, then click Next.

Manual startup with active server

If you chose not to start the database automatically, and use an active server instead of the Utility Server, the screen looks like this:



Type the server name and port number for the active IQ server. If a connection profile exists for the server, you may instead choose the server from the Connection Profile dropdown menu.

Click Next when you have specified all connection information.

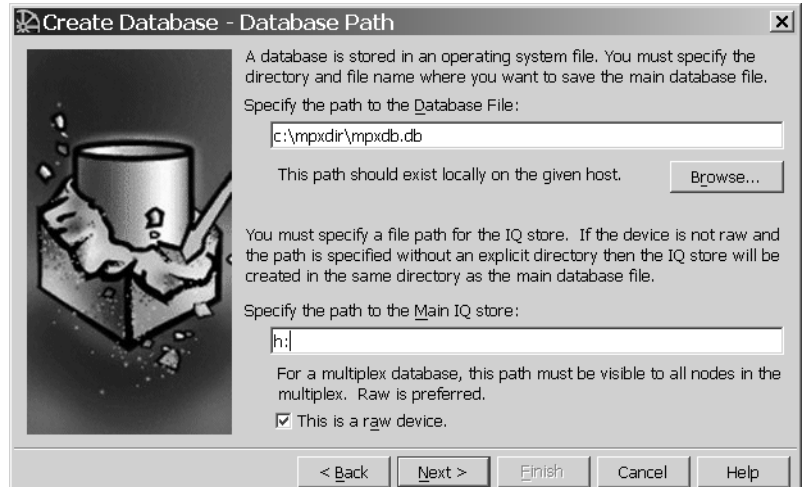
Database and IQ Store Path Information

Type the full path (directory specification) for the database file. This path should be on a local file system. Sybase does not recommend placing database files on shared file systems. If the host where the new database is to be created is the local host, then you may use the Browse buttons to locate directories for your path.

Note Folder and file names are case sensitive on UNIX systems.

Type the full path to the IQ file. This is the main segment file containing the IQ data. Make certain that the database path is on a device large enough to accommodate some growth. If no path is given, IQ uses the path to the database file.

If you plan to use multiplex functionality, this path must be visible to all database hosts for multiplex databases, and a raw device is required. For more information, see “Before creating a multiplex database” in Chapter 5, “Working with Database Objects,” *Sybase IQ System Administration Guide*.



Raw device naming
on UNIX systems

On UNIX devices, raw device names have various formats specific to a particular operating system. Check your operating system documentation for the correct format. For example:

```
/dev/rdisk/c1t3d0s6
```

Raw device naming
for Windows systems

On Windows systems, raw devices may be named by specifying the drive letter. For example:

```
h:
```

Do not supply the “\\.” in the device name when using Sybase Central.

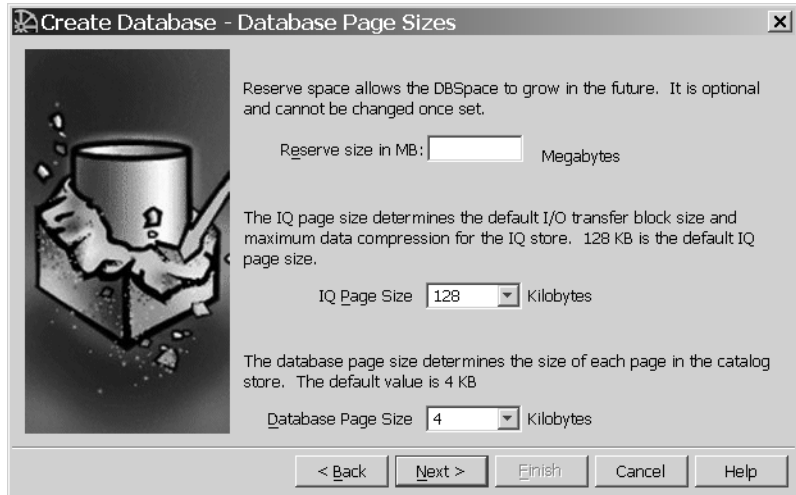
For more information about naming raw devices, see, Chapter 5, “Working with Database Objects” in *Sybase IQ System Administration Guide*.

If you have no raw device, deselect the raw device option.

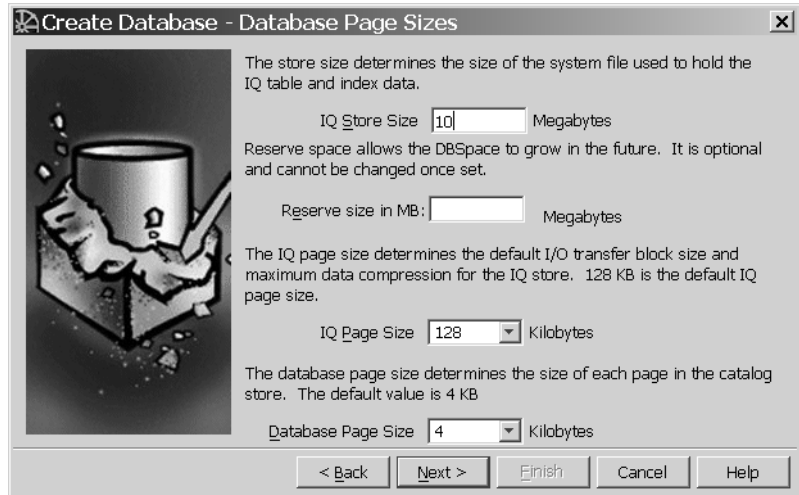
Click Next.

Setting Database Page Size Values

When the shared IQ Store is on a raw partition, the IQ Store Size box disappears because Sybase IQ determines the size of the raw device and sets IQ SIZE to that value. The Database Page Sizes screen would appear as follows:



When the IQ Store is a file, a screen like the following displays. Type the IQ Store size in MB in the space provided. The default is 1000, but for this tutorial, use 10MB.



For this tutorial, leave the Reserve size blank. In general, the amount of Reserve space you allow depends on how much space your hardware supports. If the dbspace partition is a single disk, you should never need more reserve than the largest disk currently available. If the partition is on a disk array, the size of a single partition could be much large, since a disk array can have a single partition that uses many disks.

Accept the IQ page size supplied, or choose one from the dropdown box. The Create Database wizard supplies an appropriate default page size in the IQ Page Size box. In general, the IQ page size determines the I/O transfer block size.

See the *Sybase IQ System Administration Guide* for more information. For this example, click Next to accept the default IQ page size (128KB).

Table 4-1: Default and minimum sizes of IQ and Temporary Stores

IQ page size	Default size of IQ store	Default size of Temporary Store	Minimum IQ Store size when specified explicitly	Minimum Temporary Store size when specified explicitly
65536	4096000	2048000	4MB	2MB
131072	8192000	4096000	8MB	4MB
262144	16384000	8192000	16MB	8MB
524288	32768000	16384000	32MB	16MB

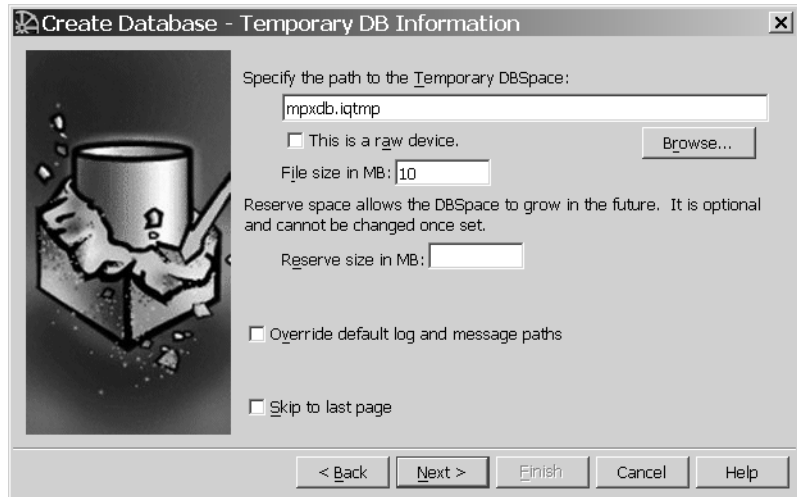
Select a Database Page Size for the Catalog Store. (The Catalog Store holds the metadata for Sybase IQ.) The default is 4096 bytes (4KB). Sybase recommends that you use this default.

Click Next.

Creating the temporary store

The Temporary Store (dbspace) is the main temporary segment file containing the temporary tables generated by certain queries. Depending on your queries, this file may be very large. If you do not specify a path, it defaults to *dbname.iqtmp* in the same directory as your database file.

Accept the default or type the path for the temporary dbspace. Make sure that the temporary dbspace is local to the given host, not on a network drive. You can also add other dbspaces later.



If the temporary dbspace is not a raw device, type its size in the second box. (The default size is a function of the IQ Size.) For this tutorial, use 10MB.

If the temporary dbspace is a raw device, click the checkbox. The File Size box disappears in this case.

For this tutorial, leave the Reserve size blank. In general, the amount of Reserve space you allow depends on how much space your hardware supports. If the dbspace partition is a single disk, you should never need more reserve than the largest disk currently available. If the partition is on a disk array, the size of a single partition could be much large, since a disk array can have a single partition that uses many disks.

By default the message and log files are placed in the directory with the database file.

- To accept the default log and message paths, leave the override box unchecked.
- To specify a new location for the log and message files, check “Override default log and message paths.”

Type a new path for the log file, message file, or both. If the host is local, you may use the Browse button to choose the location if you prefer.

If you don't plan to mirror the transaction log or disable Java classes and jConnect support, and plan to accept the default IQ database attributes and collation, click Skip to last page. Otherwise, click Next.

If you want to change any of the default database properties, do not click Skip.

Mirroring the transaction log

You can mirror the transaction log for extra security. For higher availability, Sybase recommends that you create the mirrored transaction log on a different disk from the one where the database resides. To mirror the log, click the Maintain a mirror log file checkbox, then type the log file specification in the space provided and click Next.

To avoid mirroring the log (the default), simply click Next.

Using Java in the database

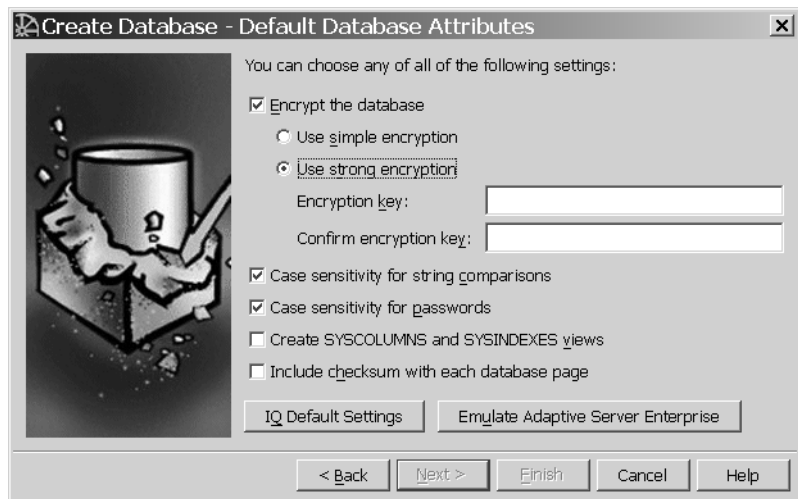
Java support for stored procedures and table definitions is installed by default.

Note For the current release, Java support is *required*. Do not deselect it.

If you wish to access system catalog information by means of the Sybase jConnect JDBC driver, which is installed by default, click Next. If you do not, deselect the check box, then click Next.

Choosing database attributes

Click the checkbox(es) to select any desired attributes. The options at the bottom of the screen let you use the defaults for IQ (Alt+Q) or ASE (Alt+U), if preferred. Choosing some attributes displays additional choices, for example:



Encrypt the database

Encryption makes it harder for someone to decipher the data in your database by using a disk utility to look at the file. File compaction utilities cannot compress encrypted database files as much as unencrypted ones. This attribute is deselected by default.

Simple encryption makes data unreadable, but it could be deciphered by someone with cryptographic expertise. Strong encryption, also called Transport Layer Security (TLS), protects the confidentiality and integrity of network packets as they pass between client and server. Strong encryption uses a 128-bit algorithm and a security key, which is a password you supply.

Case Sensitivity for string comparisons

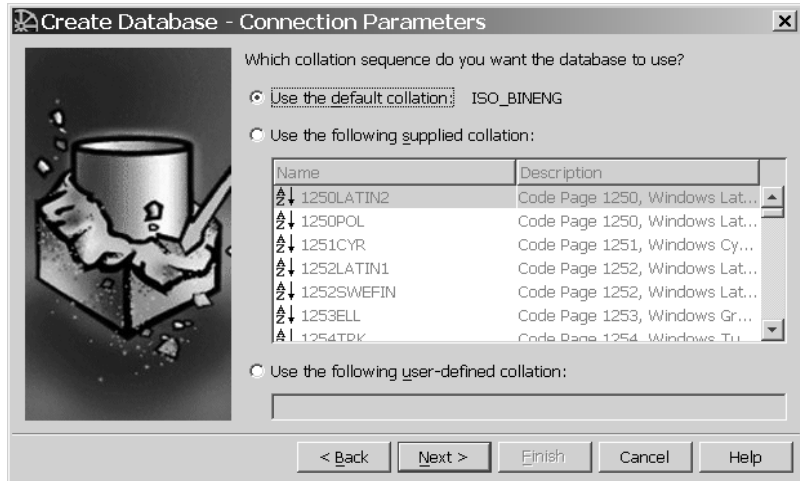
Case Sensitivity causes all values to be considered case sensitive in comparisons and string operations. Case Sensitivity defaults to ON for IQ databases.

To support the ISO/ANSI SQL standard semantics, you would select Case sensitivity for both string comparisons and passwords ON (the default).

Case sensitivity for passwords	When you connect to your database, the password that you type need not match the case sensitivity of your database. If you create the database with Case sensitivity for string comparisons OFF, or with Case sensitivity for passwords OFF, the password “SQL” or “sql” will work.
Create SYSCOLUMNS and SYSINDEXES views	<p>The SYSCOLUMNS view presents the information from the SYS.SYSCOLUMN table in a more readable format. This table includes default settings and primary key information for columns.</p> <p>The SYSINDEXES view presents the information from the SYSINDEXES and SYSIXCOL tables in a more readable format. These tables include column index information extracted from the host data file or remote server table.</p>
Include checksum	<p>Database page checksums are used to detect whether a database page has been modified on disk. When a database is created with checksums enabled, a checksum is calculated for each Catalog Store page before it is written to disk. When a Catalog Store page is read from disk, its checksum is recalculated and compared to the stored checksum. If the values differ, the page has been modified or otherwise corrupted while on disk. Checksums apply to the Catalog Store only, not the IQ Store.</p> <p>When you have finished selecting database attributes, click Next.</p>

Selecting collating sequence

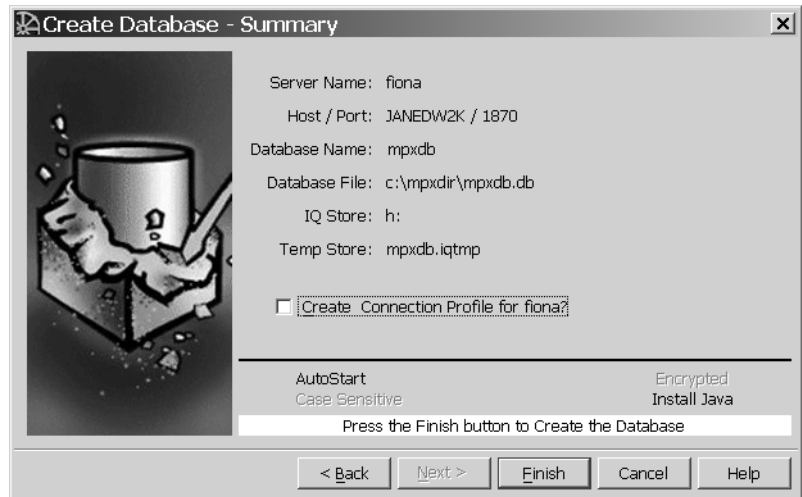
A collation sequence is a combination of a **character set** and a **sort order** for characters in the database. The default collation sequence, ISO_BINENG, provides optimal performance for IQ databases, but not necessarily the most natural sort order. For more information about collation sequences, see the *Sybase IQ System Administration Guide*.



You may define your own Collation Label by clicking the option “Use the following user-defined collation” and then typing the label name in the space provided.

Click Next.

The Summary screen confirms the choices you selected. You can click Back to return to any screen and change attributes.



Database attributes that you chose are listed below the line. Click Finish to create the database with the properties shown on this screen.

The status line displays updates as Sybase Central creates the database.

If asked to start the new database, Sybase Central connects to the new database and displays the server in the hierarchy view. (See “Specifying server options” on page 46 for the automatic startup option.)


To enable your database for multiplex capability, add query servers. For instructions, see *Creating databases with multiplex functionality* in Chapter 5, “Working with Database Objects,” *Sybase IQ System Administration Guide*.

To set up data sources for your servers, see *Sybase IQ Installation and Configuration Guide*. You may also add dbspaces to IQ servers as needed. For details, see *Sybase IQ System Administration Guide*.

Once you create a database, you can create tables and indexes, which are stored in the IQ Store. You can also create procedures and views, which are stored in the system tables. For syntax, see the *Sybase IQ Reference Manual*. For details on creating database objects, see *Sybase IQ System Administration Guide*.

Viewing database properties

Property sheets are tabbed dialog boxes that reveal all editable and non-editable properties. Sybase Central Java displays the database in the left pane when you expand the Server item. To display database property sheets, do either of the following:

- Select the database and choose File > Properties (Alt+F, R) or click the Properties button .
- Right-click the database name. From the submenu choose Properties.

Viewing database options

The database options control database behavior. For a complete list of the database options, see the *Sybase IQ Reference Manual*. To view the current database options, right-click the database name. Choose Options from the submenu.

In the Database Options dialog box, you can alter permanent options by changing the setting in the Value box and clicking Set Permanent Now.

Deleting databases

You can use the DROP command in DBISQL to remove the database files (temporary files, IQ database files, and catalog files) for a database.

❖ Deleting a database

- 1 Start the server without a database (the preferred method) or with a different database than the one to be dropped.
- 2 Start DBISQL, if it is not already running. (For instructions, see “Connecting to other databases from DBISQL” on page 87 in *Sybase IQ System Administration Guide*.) If no database was specified, the utility database is used, and the Database Name must be set to *utility_db*.
- 3 Type the following command:

```
DROP DATABASE <DBname>
```

For more details about this command, see the *Sybase IQ Reference Manual*.

Managing tables


To add a table to your database, you define the name and properties of your table and its columns using the appropriate wizard. Three wizards are available:

- Table Creation wizard — for persistent tables
- Global Temporary Table Creation wizard — for table data that persists only during your connection. Each user has a separate instance of the data in a global temporary table. Those rows are visible only to the connection that inserts them.
- Proxy Table Creation wizard — for tables on remote servers

Creating a table and adding columns with Sybase Central

In this tutorial we create a table in the sample database. The new table will describe different offices for the fictitious company.

❖ Creating a table

- 1 Expand the server and database containers, then click the Tables folder to open it.
- 2 Click the New Table toolbar button  or choose File > New > Table (Alt+F, N, T). This starts the Table Creation wizard.
- 3 Type the new table name in the dialog box. In this case, use the name *office*.
- 4 From the dropdown menu, select the owner for the table. By default, the *asiqdemo* database has only one user, *DBA*. Click Next.
- 5 For this tutorial, click or arrow to the *dbspace IQ_SYSTEM_MAIN* to store the new table there. (On a query server in a multiplex database, this would not be possible.) Click Next.
- 6 Click in the Comment field and type:


Company offices

 in the Comment text box. Comments are optional. Click Finish.
- 7 The *office* table appears in the left pane, and the column tab in the right pane. Follow the steps in Creating columns for a table to add columns to the new table.

❖ **Creating columns for a table**

- 1 The Create Table wizard adds each new table to the hierarchy and displays a columns tab for you to complete.
- 2 To use the new column as a primary key, leave the PKey box checked.

Note If a check box appears under Allow Nulls column, or if duplicate values are stored in the office_id column, the column cannot be changed to a primary key.

- 3 In the Name box in the first row of columns, type the name of the first column in the table. For this tutorial, use the name *office_id*.
- 4 Select the Data Type integer. Leave the ID, Size, Scale, Nulls, Unique and Comments columns blank for the *office_id* column.
- 5 Tab to a new row or use Alt+F, N for a new column. Add rows for the following columns, following steps 3 and 4 and tabbing to the Size/Precision column to add the sizes below. Leave the Allow Nulls column checked for each:
 - region char(7)
 - street char(40)
 - city char(20)
 - state char(2)
 - zip char(9)
- 6 Finish creating the table by clicking the Save Table toolbar button . The office table is now visible on the tables tab in the right pane of Sybase Central.

The table is now present in the database, although it has no data. To enter data, use the Interactive SQL utility or the Data tab.

The following section describes how to edit an existing table.

Editing existing tables

To see what actions you can perform on a table, right-click the table. In this manner, you can display submenus for most objects throughout the Sybase Central interface. Database objects also have creation wizards, launched by toolbar buttons.

❖ Editing column properties

- 1 In the left panel, click the *customer* table. You may have to expand the database container and the Tables folder to do this.
- 2 Select the Columns tab in the right panel. The tab displays the columns in the *customer* table. The newly-added *address* column should be visible.
- 3 Change the name of the column to *office_address* by typing this in the Name box. This modification applies to the *address* column of the *office* table only. Once a column has been changed there is no longer any connection between the original column and the new column. Click OK.


Deleting tables with Sybase Central

Tables can be deleted, or dropped, from a database. The office table created in a previous section can be dropped from the sample database, restoring the database to its original state.

❖ Deleting the office table from the sample database

- 1 Right-click the *office* table.
- 2 Select Delete from the submenu.

or

Left-click the table and press the Delete key or the Delete toolbar button. 

You can delete proxy tables in the same way. If you try to delete a proxy table that was created with the new local remote option and the remote server is inaccessible, you get an error message indicating that the remote server is inaccessible.

Warning! Deleting a proxy table that was created with the new local and remote table option also deletes the remote table. You will lose all data stored in the remote database.

Managing procedures

Stored procedures are kept in a folder within the database container object. This tutorial shows how to view and alter the contents of a procedure, how to execute a procedure, and how to create new procedures using the Sybase Central code editor.

The Sybase Central code editor is a separate window for displaying and editing the code of triggers, procedures, and functions.

Beyond text-editing functions, it provides:

- Automatic syntax highlighting.
- Automatic formulation of DDL statements.
- Ability to find and replace text, and to jump to specific line numbers.
- Ability to execute the DDL code against the database.


Viewing stored procedure code with Sybase Central

There are several stored procedures included in the sample database. The following steps explain how to view and edit stored procedures.

❖ Viewing and editing procedures

- 1 Select the Procedures & Functions folder from the sample database in the left panel. You may have to expand the database container to do this.
- 2 Right-click the procedure in the right panel; for example `sp_customer_list`.
- 3 In the popup, click Edit in New Window (Ctrl+E). The Sybase Central code editor appears, displaying the text of the procedure.

```
ALTER procedure
"DBA"."sp_customer_list"()
result(id integer,company_name char(35))
begin
select id,company_name from customer
end
```

- 4 Menu options and toolbar buttons let you cut, paste, copy, delete.
- 5 When finished editing, save the procedure using the Save Procedure toolbar button  or Alt+F, S. If you made no changes, choose File, Close (Alt+F, C).

❖ **Running a stored procedure**


- 1 Select the Procedures & Functions folder from the sample database in the left panel. You may have to expand the database container to do this.
- 2 Right-click the procedure in the right panel, for example `sp_customer_list`.
- 3 From the popup, choose Execute from Interactive SQL (Alt+F,u). Click OK.
- 4 Sybase Central starts Interactive SQL, connects to the database, and executes the procedure.

A call to the `sp_customer_list` procedure returns a set of customer IDs and company names from the Customer table.

Setting permissions
on procedures

Stored procedures have permissions associated with them. In order to execute a procedure, you either need to be granted permission to execute a procedure, or you need to be a member of a user group that has permission to execute the procedure.

❖ **Viewing and altering procedure permissions**

- 1 Right-click a procedure and click the Properties toolbar button  or select Properties (Alt+R) from the submenu. A properties sheet displays. (For example, right-click `sp_customer_list`.)
- 2 Click the Permissions tab to see which user IDs have been granted permissions on this procedure. For `sp_customer_list`, none do, as the only user for the sample database is DBA, who automatically has execute permissions as owner of the procedure.
- 3 Click Grant to grant users or groups permission to execute this procedure. Grant permission to execute this procedure to the Public user group by selecting the Public user group icon and clicking OK.
- 4 Click Cancel or Apply, then OK.

Managing users and groups

In Sybase IQ, both users and groups are objects within the database. However, groups are also containers, capable of containing users (making users a member of that group) and other groups.

In this structure, permissions granted to a group are inherited by those users and groups which have membership in the group. Sybase IQ allows you to create users and groups permitted to use a database and grant membership to groups.

This tutorial demonstrates how to create a group for the database, create an individual user, and make the user a member of the group.

Adding a group to the database

In this section we add a sales group to the sample database.

❖ Adding a group to a database


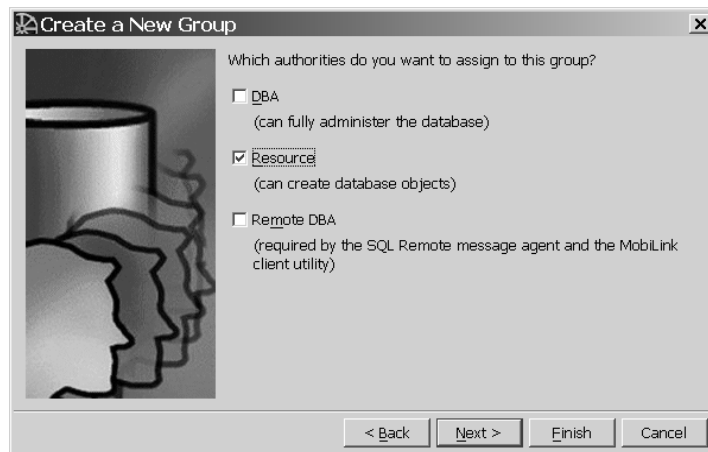
- 1 Select the Users & Groups folder. You may need to expand the sample database container in the left panel to do this.
- 2 Click New Group  toolbar button, or choose Alt+F, N, G.
- 3 Type the name Sales in the text box. This is the name of the user group. Click Next.
- 4 Click the checkbox “Allow this group to connect.”
- 5 Type a password, for example, *Sales*; confirm it by tabbing and typing it again. Click Next.
- 6 Check the Resource authorities option and click Next.

Figure 4-1: Creating a new group



Clicking Next gives you the option of typing a comment for this group.

7 Click Finish to create the new group.

The container automatically refreshes and displays the new group, Sales.

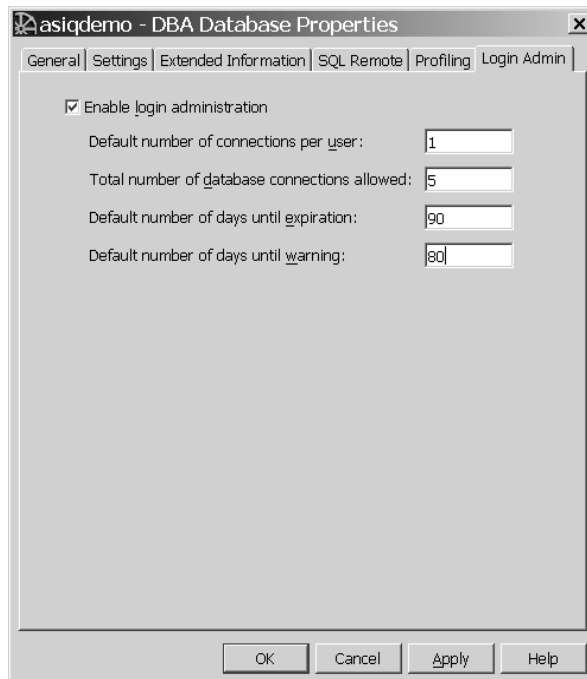
Adding a user to the database


This section describes how to add a user to a database.

❖ Adding a user to the sample database

Create users in order to add permissions to database objects.

- 1 For Sybase IQ User Administration functionality, right-click the database and choose Properties from the submenu. If you do not wish to manage logins for this user, skip to Step 4.
- 2 Choose the Login Admin tab. Click “Enable login administration.”
- 3 Specify the numbers for administration settings. For this tutorial, set 1 connection per user, 5 total database connections, 90 days until expiration, and 80 days until warning



- 4 Select the Users & Groups folder in the left panel. You may need to expand the sample database container in the left panel to do this.
- 5 Click New User  toolbar button, or choose Alt+F, N, U.
- 6 Type the name Sandy in the text box and click Next. This is the user ID for the new user.
- 7 Check “Allow this user to connect” and type a password. Alt+C or tab to the Confirm box, and retype the password. For example, you could use the password “beach”.

If you have enabled Sybase IQ User Administration, the following screen appears.

You can select Login locked to lock this user out of the database, if desired. You can also modify the number of active connections and days the password is valid. The remaining limits are set on the database (see Step 3).



- 8 Click Next. The authorities screen displays. To select the authority desired for the new user, check the option or apply the shortcut key combination from the following list:
 - DBA (Alt+D) — Can fully administer the database
 - Resource (Alt+R) — Can create database objects
 - Remote (Alt+m) — Required by SQL Remote

For this tutorial, grant DBA authority.

- 9 Click Next. Type “Sales dept. intern” in the Comment box.
- 10 Click Finish to create the new user.

Adding a user to a group

In this section we add a user to a group.

❖ Adding users to a group

- 1 Double-click the user in the right container. The user container opens.
- 2 Right-click the Membership tab and select New Memberships, or press Alt+F, N, M.



- 3 Select the Sales group.
- 4 Click OK.

Restoring the sample database to its original state

To restore the database to its original state, you can delete the Sales group and the user Sandy. For each of the two objects:

- 1 Right-click the object in the Sybase Central hierarchy.
- 2 Select Delete from the submenu, or click Alt+E,D.
- 3 Click Yes to confirm the deletion. The container is automatically updated.

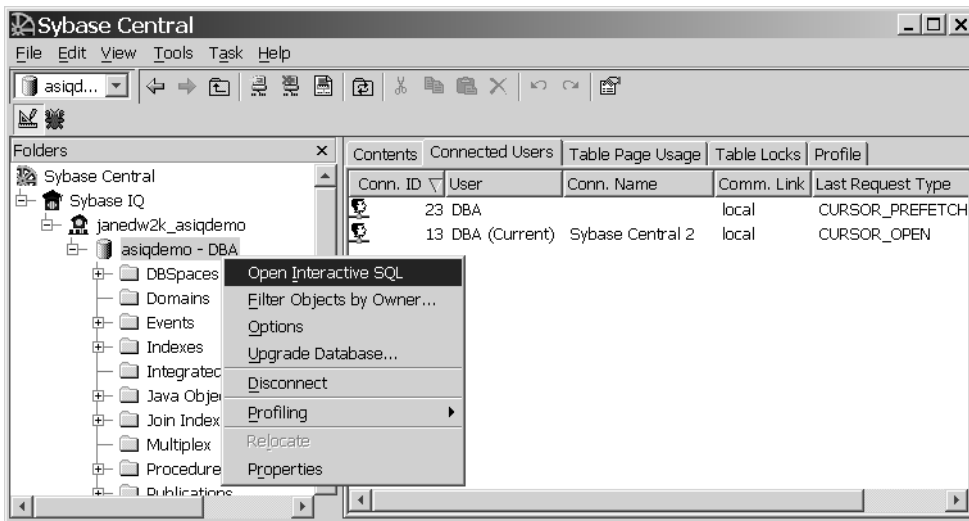
Managing connected users

The Connected Users tab displays all users connected to the current database.

The next example shows how to create new connections to the sample database and view them using Sybase Central.

❖ Creating and viewing connections

- 1 Select the database in the left pane and click the Connected Users tab in the right pane.
- 2 Right-click the server name in Sybase Central's left panel. From the submenu, choose Interactive SQL.




- 3 Repeat the previous step. The screen automatically refreshes.
- 4 The right pane should now shows two more connected users. Each user is identified by a Connection ID assigned by the server.

❖ Disconnecting users from a database

- 1 Select the desired user to disconnect by clicking the Conn ID.
- 2 Right-click the Conn ID or Alt+F, D and choose Disconnect.

❖ Disconnecting your current connection

- 1 Click the Disconnect toolbar button  or choose Tools > Disconnect.

- 2 If your connection is the only active one, you will be disconnected immediately with no prompts. If there are multiple active connections to multiple servers, a disconnect dialog box appears and you must verify the connection desired.

About this chapter

In the previous chapter, you learned how to manage your database's schema and objects. This chapter tells how to index columns to take advantage of the query optimization features of Sybase IQ and how to load data into your tables.

You will complete tutorial tasks on creating column and join indexes and loading data.

Creating column indexes

When you create a table and specify its columns, Sybase IQ automatically creates certain default storage structures to optimize query processing. Knowing what kinds of queries you plan to run, you can add multiple indexes to any column.

Note Some additional tasks are required for multiplex databases. Please see the *Sybase IQ System Administration Guide* instead of this chapter if you need to index data in a multiplex.

It is faster to create all the indexes needed *before* you insert any data into your database. You can drop any of the optional indexes later if you decide you don't need it.

Warning! Be sure to verify that the index is not important before you drop it. Different queries use different indexes, even when they appear to be similar. Dropping the wrong indexes may adversely affect performance. Capture queries, run times, and query plans to determine which indexes are required.

Each type of column index is designed to speed processing of a certain kind of data.

Table 5-1: Sybase IQ column index types

Index type	Purpose
Compare (CMP)	Stores the binary comparison (<, >, =, <=, >=, or NE) of any two columns with identical data types, precision, and scale.
DATE	An index on columns of data type DATE used to process queries involving date quantities.
Datetime (DTTM)	An index on columns of data type DATETIME or TIMESTAMP used to process queries involving datetime quantities.
High_Group (HG)	An enhanced b-tree index to process equality and group by operations on high-cardinality data. (Recommended for more than 1,000 distinct values or for a table with less than 25,000 rows.)
High_Non_Group (HNG)	A non value-based bitmap index ideal for most high-cardinality DSS operations involving ranges or aggregates.
Low_Fast (LF)	A value-based bitmap for processing queries on low-cardinality data. (Recommended for up to 1,000 distinct values and more than 25,000 rows in the table. Can support up to 10,000 distinct values.)
TIME	An index on columns of data type TIME used to process queries involving time quantities.
WD	Used to index keywords by treating the contents of a CHAR or VARCHAR column as a delimited list.

These index types are unique to IQ data and cannot be applied to Anywhere tables. For more information, see the *Sybase IQ System Administration Guide*.


Sybase IQ assumes you will add either a LF or a HG index to every column mentioned in a WHERE clause and in a GROUP BY clause.

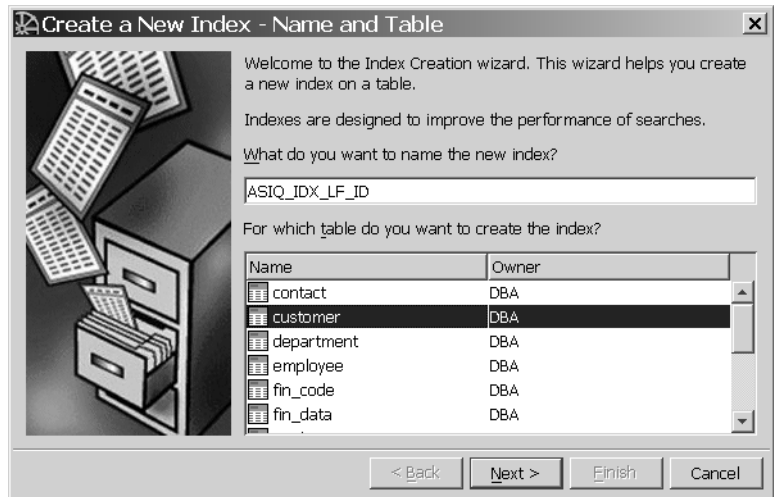
When you designate a column or set of columns as either a PRIMARY KEY or UNIQUE, IQ creates a High_Group index for it automatically. Choose one PRIMARY KEY from all UNIQUE constraints for the table. Both PRIMARY KEY and UNIQUE constraints do not allow nulls; however, a unique index would allow them. UNIQUE constraints also provide hints on column constraints to the query optimizer. The *Sybase IQ System Administration Guide* describes when to use each type of index and the space and time trade-offs of each.

Sybase IQ always uses the fastest index available for the current query or join predicate. If you did not create the index types the query optimizer would ideally like to use for a column, Sybase IQ can still resolve queries involving the column, but response time may be slower than it would be with the correct index type(s).

For this tutorial, appropriate index types have been determined for you. Simply follow the steps to index columns in your table.

❖ **Adding a Low Fast index to a column**

- 1 Connect to the asiqdemo database, if not connected.
- 2 Select the Indexes container and click the New Index toolbar button  or choose Alt+F,N,I.
- 3 Type the name ASIQ_IDX_LF_ID. You may choose any index name unique in the database. Descriptive names are best.
- 4 Select the customer table from the list and click Next.



- 5 On the Type screen, select the Low Fast option.

The default number of records to add before notification is sufficient for this tutorial. For details about the notify count, see the *Sybase IQ System Administration Guide*.

- 6 Click Next. On the Columns screen, select the id column and click Add, then Next.

7 Type the comment “Low Fast index for id column” in the Comment box and click Finish to create the index. The Indexes container is updated with the new index.

8 Now add a Low Fast index to the State column, using steps 3 through 7.

The remaining columns in the table do not require any of the optional indexes. The default storage structures are sufficient.

You can use the CREATE INDEX command instead of Sybase Central to create column indexes if you prefer. For syntax, see the *Sybase IQ Reference Manual*.

Creating join indexes

In addition to the column indexes, Sybase IQ allows you to define join indexes. **Join indexes** are IQ internal structures that optimize joins of related tables. In *Sybase IQ Performance and Tuning Guide* you will learn more about join relationships between tables. In this chapter, tutorial instructions will supply join information for you to use in creating join indexes.

Join indexes are created on sets of columns rather than individual columns. A join index represents a full outer join of two or more tables. The query engine may use this full outer join as a starting point for queries that include left outer, right outer, and inner joins. You can create a join index for any set of columns that your users commonly join to resolve queries. For guidelines about creating join indexes, see the *Sybase IQ System Administration Guide* and “Ad hoc joins vs. using join indexes” in “Joining Tables,” *Sybase IQ Performance and Tuning Guide*.

You should note three important rules about creating join indexes:


- Create join indexes *after* indexing columns, because the join index will rely on the column indexes of related tables.
- The creator of the join index must also be the owner of the tables used in the index, or the join index will fail.
- Join index columns must have identical data type and scale.

The following tasks show how to create join indexes. Refer to “The sample database” on page 10 to see which columns are in each table in this example.

Note Some additional tasks are required for multiplex databases. Please see Chapter 5, “Working with Database Objects,” in *Sybase IQ System Administration Guide* instead of this chapter if you need to create join indexes in a multiplex.

The first join index, `so_soi_idx`, will join the `sales_order` and `sales_order_items` tables. To create it, follow these steps:

❖ **Creating a join index for `sales_order` and `sales_order_items`**


- 1 Select the Join Indexes container and click the New Join Index toolbar button  or choose Alt+F,N,J to open the Create Join Index Wizard.
- 2 In the Name and Table dialog, type “`so_soi_idx`” for the name.
- 3 In the Join Type dropdown list, select Natural, because you are joining equivalent columns with the same name (`id`) from two tables. For more details, see *Sybase IQ Performance and Tuning Guide*.

Note When joining with the NATURAL keyword, take care that *only* the columns to be joined have the same name. (For example `id = id`.) Watch out for common column names like `date` and `description`.

- 4 Choose `sales_order` for the left table.
- 5 Choose `sales_order_items` for the right table.
- 6 Click Next. Sybase IQ locates the join columns.
- 7 In the Comment space, type the following to describe the join index, `Join index for sales_order and sales_order_items`.
- 8 Click Finish.
- 9 The `so_soi_idx` join index appears in the Join Indexes folder.

❖ **Creating a join index for `department` and `employee`**

Now create a join between `department` and `employee`, using equivalent values in columns from two tables. Follow these steps to create this joined index, which we will call `dept_emp_idx`:

- 1 Select the Join Indexes container and click the New Join Index toolbar button  or choose Alt+F,N,J to open the Create Join Index Wizard

- 2 Type the name of your new join index, “dept_emp_jdx,” in the Name box.
- 3 In the Join Type dropdown box, select ON. The ON clause is required in order to join equivalent columns with the same name (dept_id) from two tables.
- 4 Choose department for the left table.
- 5 Choose employee for the right table.
- 6 Click Next.
- 7 In the Choose Index Columns dialog, select the desired Left Table column, dept_id and the Right Table column dept_id.
- 8 Click Add. After you add the two columns, they appear in the Joined Columns box as DBA.department.dept_id=DBA.employee.dept_id. (You may have to scroll to read the entire join.)
- 9 Click Next.
- 10 On the Add Comment screen, type the following to describes the join index, `Join index for department and employee.` Click Finish.
- 11 The dept_emp_jdx join index appears in the Join Indexes folder.

For detailed information about planning and creating join indexes, see *Sybase IQ System Administration Guide*.

Updating join index data

Join indexes are unavailable to other writers while tables referenced by the index are being updated. If you load your tables before you create a join index using those tables, you must synchronize your join indexes after loading the tables. Sybase Central does not currently support the synchronize operation; you must use the SYNCHRONIZE JOIN INDEX command. See “Synchronizing join indexes” on page 90 for details about how to update your join indexes.

Loading data

After creating tables and indexing their columns, you can fill them with data. You can load your IQ database using any of the following:

- The ODBC programming interface

- Embedded SQL programming interface
- The Open Client programming interface
- DBISQL
- DBISQLC
- Sybase Central

For details about the programming interfaces listed above, see the *Sybase IQ Utility Guide*. If a client application has been written to perform the load, use the appropriate programming interface.

You may also insert data manually. Interactive SQL (DBISQL) is an interface available on all supported platforms. You can use the DBISQL command line interface at the UNIX or Windows command prompt. Windows users can also invoke Interactive SQL from Sybase Central or the Sybase IQ Program Group. In *Sybase IQ Utility Guide* you will learn how to run DBISQL and will issue some basic commands.

There are three ways to load a table using SQL commands:

- INSERT FROM SELECT can load information from either databases on your system or an attached database.
- INSERT VALUES inserts data “manually” into specified table columns
- LOAD inserts data from files. It can be used to insert binary data from backups and upgrades. To insert data from files, you must use the LOAD command instead of INSERT. Sybase IQ supports loading from both ASCII and binary files and supports both fixed-length and variable-length formats.

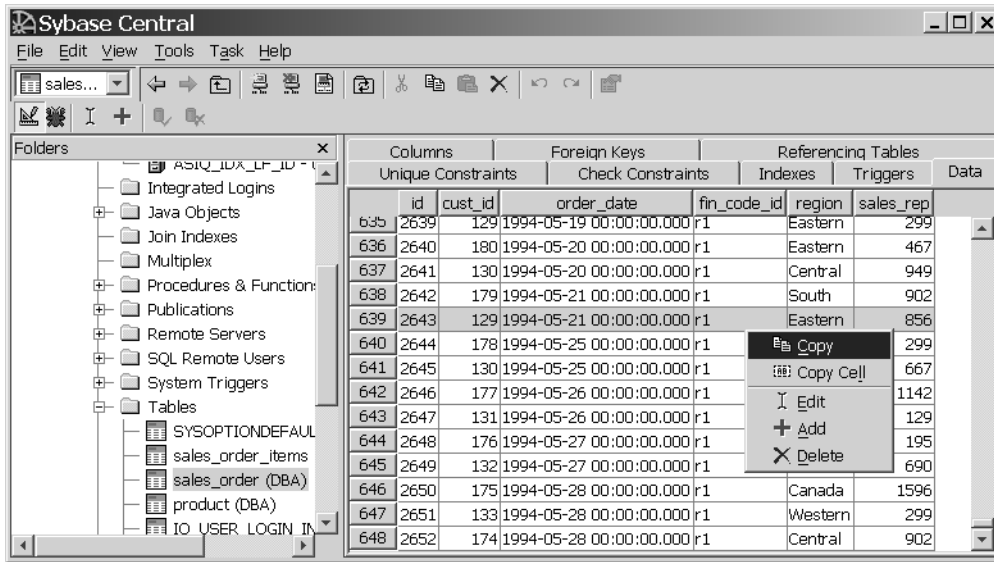
Note Some additional tasks are required for multiplex databases. Please see the *Sybase IQ System Administration Guide* instead of this chapter if you need to load data into a multiplex database.

To load large amounts of data, most users create command files. To create a command file, follow the instructions in Chapter 2, “Using Interactive SQL (dbisql)” in the *Sybase IQ Utility Guide*.

Syntax and examples for the INSERT and LOAD commands are in the *Sybase IQ Reference Manual*.

To add data using Sybase Central, open the Tables container and choose the Data tab in the right pane. Right-click the row you want to update and choose Add from the submenu shown in Figure 5-1.

Figure 5-1: Adding data through Sybase Central



About this chapter

In the previous chapter, you learned how to index columns and load user data. This chapter walks you through managing the way user data is distributed across devices in your system.

Overview of dbspaces

Sybase IQ distributes user data across multiple disks at the application level by representing each device as a dspace. A dspace can be an operating system file or a raw disk partition. Dbspaces can contain both user data and internal database structures used for startup, recovery, backup, and transaction management.

Creating a database creates several database files by default:

- *db-name.db* is the catalog dspace containing the system tables and stored procedures describing the database and any standard Adaptive Server Anywhere database objects you add. It is known as the Catalog Store, and has the dspace-name SYSTEM.
- *db-name.iq* is the main data dspace containing the IQ table data and indexes. It is known as the IQ Store, and has the dspace-name IQ_SYSTEM_MAIN.
- *db-name.iqtmp* is the initial temporary dspace containing the temporary tables generated by certain queries. It is known as the IQ Temporary Store and has the dspace-name IQ_SYSTEM_TEMP.

Multiplex users may also create an IQ Local Store on a query server. It behaves like the IQ Store with respect to only that query server, and shares buffer cache with the IQ Main Store. This store has no default extension, although many users choose *.iqloc*.

For more about dbspaces, see “Adding dbspaces” in *Sybase IQ System Administration Guide*.

Although it is best to create all required dbspaces at database creation, Sybase IQ allows database administrators to add new dbspaces at any time to increase space available to the database.

Creating dbspaces

To create a dspace, open the Dspaces folder, chose File > New > Dspace (or Alt+F, N, D), and run the Create Dspace wizard.

Specifying dspace name and store

The dspace name, an internal name for the dspace, must be unique in a database and across a multiplex. Dspace names are case sensitive for databases created with CASE RESPECT. (To view the CaseSensitive setting, right-click the database, choose Properties, and view the Extended Information tab.) Type the name for your new dspace in the first text box.

Now specify a store for the dspace:

- On a write server, the wizard creates a dspace in the Main IQ store (Alt+M) by default. If you prefer the Temporary IQ store, click the Temporary IQ store option (Alt+T).
- On a query server, the wizard creates a dspace in the Temporary IQ store (Alt+T) by default. To create a Local IQ store, click the Local IQ store option (Alt+L). Changes to the data in a local store are local to the query server, allowing users to continue development operations on that query server without affecting the shared main storage. For details, see “Multiplex capability” in Chapter 2, “Terms and Concepts,” in *Sybase IQ System Administration Guide*

Click Next.

Specifying dbspace path

The filename is the actual filename of the dbspace, with a path where necessary. A filename without an explicit directory is created in the same directory as the initial dbspace of that store. the *.db* file. Any relative directory is relative to that initial dbspace. Be sure to specify the filename suffix — *.iq* for an IQ Main Store, *.iqtmp* for IQ Temporary Store.

If the dbspace is an operating system file, deselect the option “This is a raw device.” Then specify the file size in MB.

If desired, specify the size in megabytes of free list space to reserve, so that the dbspace can be increased in size in the future. The size can be any number greater than 0. The reserve cannot be changed after the dbspace is created.

When you specify reserve size, the database uses more space for internal (free list) structures. If reserve size is too large, the space needed for the internal structures can be larger than the specified size, which results in an error.

Note Reserve size and mode (read write, read only, relocate) for temporary dbspaces are lost if the database is restored from a backup.

A database can have up to 2047 dbspaces, including the initial dbspaces created when you create the database. However, your operating system may limit the number of files per database.

Click Finish to create the new dbspace or Back to make changes.

Altering dbspaces

To alter a dbspace, right-click a dbspace and choose Properties, Usage. If you have DBA privileges, you can change the size and/or mode of the dbspace.

To change the size, type a new size (in MB) in the Size text box. The size to which a dbspace can grow depends on the reserve size set when the database was created. Once set, the reserve size cannot be changed. For more information, see “Setting Database Page Size Values” on page 50.

To change the mode, choose a mode from the Mode submenu. Modes are:

- Read Write — Default for new dbspaces, writes to the dbspace are allowed

- Read Only — Transactions active at the time of the status change may write to the dbspace, but future ones cannot.
- Relocate — Space will not be allocated from the dbspace and objects on the dbspace may be relocated. Relocating moves all data on main or local dbspaces with Relocate status to main dbspaces with Read Write status.

❖ **Relocating a dbspace**

- 1 Open the DBspaces container.
- 2 Right-click IQ_SYSTEM_MAIN and choose Properties.
- 3 Click the Usage tab and select Relocate from the Mode submenu. Click OK.
- 4 Right-click the desired table or index and choose Relocate from the Mode submenu.

Deleting dbspaces

To drop a dbspace, right-click a dbspace and choose Delete from the submenu, or choose Alt+E, D.

The Confirm Delete question box requests verification. Click Y (Alt+Y) to confirm, or N (Alt+N) to cancel the delete.

Copying dbspaces

To copy a dbspace, right-click on a dbspace and choose Copy.

About this chapter

In the previous chapter, you learned how to distribute user data across system devices. This chapter explains how to modify data. You will complete tutorial tasks on inserting table rows, canceling and confirming changes, and deleting data.

Migrating data from databases from earlier versions

If you have a Sybase IQ database created with an earlier version, you can easily upgrade it to the current version from Sybase Central.

❖ Upgrading the database

In order to upgrade a database using Sybase Central, you must be connected to the database.

- 1 Start Sybase Central and connect to the database that you wish to upgrade.
- 2 Right-click the database container.
- 3 Choose Upgrade Database from the popup.
- 4 Highlight the database you want to upgrade and click Next.
- 5 If jConnect meta-information support was not installed, and you want to install it, check the option. Sybase highly recommends installing jConnect support.
- 6 Choose an option to install or remove Java support, and click Finish.

Adding rows to a table

In Chapter 3, you learned to use the Sybase Central table editor to update a table. Now let's use the INSERT statement to add data about a new department.

Suppose that a new eastern sales department is created, with the same manager as the current Sales department. You can add this information to the database using the following INSERT statement in Interactive SQL:

```
INSERT
INTO department ( dept_id, dept_name, dept_head_id )
VALUES ( 220, 'Eastern Sales', 902 )
```

If you make a mistake and forget to specify one of the columns, Sybase IQ reports the following error:

```
Error at line 1
The insert VALUES do not match the column list.
```

Note Case sensitivity may matter when inserting values.

The NULL value is a special value used to indicate that something is either not known or not applicable. Some columns are allowed to contain the NULL value, and others are not.

To check whether a column in a table allows the NULL value, use the sp_iqcolumn stored procedure in Interactive SQL. For example, type the following:

```
sp_iqcolumn employee
```

	table_name	table_owner	column_name	domain_name	width	scale	nulls	default	cardinality	est_car...	location	remarks	check
1	employee	DBA	emp_id	unsigned int	4	0	N	(NULL)	75	75	(NULL)	(NULL)	
2	employee	DBA	manager_id	unsigned int	4	0	Y	(NULL)	0	5	(NULL)	(NULL)	
3	employee	DBA	emp_fname	char	20	0	N	(NULL)	0	66	(NULL)	(NULL)	
4	employee	DBA	emp_lname	char	20	0	N	(NULL)	0	75	(NULL)	(NULL)	
5	employee	DBA	dept_id	unsigned int	4	0	N	(NULL)	5	5	(NULL)	(NULL)	
6	employee	DBA	street	char	40	0	N	(NULL)	0	75	(NULL)	(NULL)	
7	employee	DBA	city	char	20	0	N	(NULL)	0	25	(NULL)	(NULL)	
8	employee	DBA	state	char	4	0	N	(NULL)	0	4	(NULL)	(NULL)	

In the nulls column, a “Y” value indicates that nulls are allowed.

A short form for
INSERT

You can use a short form to enter values for all the columns in a table in the order they appear when you `SELECT *` from the table (the order in which they were created). The following is equivalent to the previous `INSERT` command:

```
INSERT  
INTO department  
VALUES ( 220, 'Eastern Sales', 902 )
```

Note You should use this form of `INSERT` with caution; it will not work as expected if you ever change the order of the columns in the table or if you add or remove a column from the table.

Canceling changes

You may be a little concerned about all of the changes you have made to the employee table. However, SQL allows you to undo all of these changes with one command:

```
ROLLBACK
```

The `ROLLBACK`
statement

The `ROLLBACK` statement undoes all changes you have made to the database since the last time you made changes permanent (see `COMMIT` in the next section).

Making changes permanent

The SQL statement

```
COMMIT
```

makes all changes permanent.

Note Use COMMIT with care. When trying the examples in this tutorial, be careful not to COMMIT any changes until you are sure that you want to change the database permanently.

While Sybase IQ excels at bulk changes in a single transaction, its per transaction overhead is higher than a traditional OLTP database. (Also true at the statement level.) This means that COMMIT statements may take longer to complete than in an OLTP system, but in a data warehousing environment, there tend to be fewer commits than in OLTP systems.

Making changes permanent in DBISQL

The default action in DBISQL is to do a COMMIT on exit. This can be controlled with the DBISQL option COMMIT_ON_EXIT.

For more information on DBISQL options, see *Sybase IQ Utility Guide*.

DBISQL has another option named AUTO_COMMIT. If this option is on, DBISQL does a COMMIT operation after every command. The default for this option is OFF. Usually you will want it OFF, giving you the opportunity to ROLLBACK the changes if, for example, a delete operation doesn't produce the intended results. Keep in mind that the increased overhead from more frequent COMMIT operations will affect your system performance.

Ensuring data consistency

Sybase IQ has special features that ensure data consistency in a data warehouse environment.

Typically, in a data warehouse environment, many users need to read from the database, but only the DBA needs to update it. However, there is often a need to make updates while other users continue to request and receive query results. Sybase IQ keeps track of database changes using **table-level snapshot versioning**. It keeps a record of what the table looks like when a user begins a write transaction.

Sybase IQ allows multiple readers, but only one writer to a table. In the next procedure, you will connect as two different users and try to write to a table from both connections at the same time. One statement will be rolled back and will receive an error message while the other commits.

❖ **Testing data consistency**

- 1 Connect to the database using a connection name in Interactive SQL:

```
CONNECT DATABASE asiqdemo
AS sales
```

- 2 Start a read/write transaction.

```
INSERT INTO fin_code
(code, type, description)
VALUES ('e6', 'expense', 'Services')
```

- 3 Connect to the same database with a different connection name and run a query.

```
CONNECT DATABASE asiqdemo
AS marketing;
SELECT cust_id, order_date
FROM sales_order
```

Sybase IQ returns the requested information; this is a read-only transaction.

- 4 Connect to the same database using a different connection name:

```
CONNECT DATABASE asiqdemo
AS accounting
```

- 5 Start a read/write transaction:

```
INSERT INTO fin_code
(code, type, description)
VALUES ('r3', 'revenue', 'Sales & Marketing')
```

This command will fail and be rolled back.

- 6 As user *sales*, commit your transaction.
- 7 Retry your transaction as user *accounting*. You may now commit or roll back this transaction.

After each transaction commits, Sybase IQ writes updated data pages to disk. This approach is ideal for the data warehouse, where a single application may write millions of rows of data. A **checkpoint** is a point in a transaction when the database writes information to disk. Most OLTP databases write data to disk at checkpoints. Sybase IQ does not wait for a checkpoint to write physical data. Sybase IQ uses checkpoints to write certain information to disk for internal tracking, which is used if you need to recover your database. While you may set explicit checkpoints, most Sybase IQ checkpoints occur automatically.

To limit the amount of data that Sybase IQ writes to disk, you may set a savepoint. A **savepoint** defines a point in a transaction after which all changes can be undone by a `ROLLBACK TO SAVEPOINT` statement. This allows you to commit data before the entire transaction finishes.

Now let's try setting some savepoints to return to as we run two similar transactions. This time, we can put the second transaction on hold until the first one completes.

❖ **Setting savepoints**

- 1 Connect to the database:

```
CONNECT DATABASE asiqdemo
```

- 2 Start a read/write transaction.

```
INSERT INTO fin_code  
(code, type, description)  
VALUES ('e8', 'expense', 'Services')
```

- 3 Set a savepoint.

```
SAVEPOINT TUES9_45
```

Naming your savepoint (as shown) is optional. You must follow the rules for object names described in *Sybase IQ Reference Manual*

- 4 Start a read/write transaction:

```
INSERT INTO fin_code  
(code, type, description)  
VALUES ('r3', 'fees', 'Administration')  
COMMIT
```

If this insert command were to fail, the transaction would roll back to Savepoint TUES9_45. The data inserted in Step #2 would not be lost.

If you are working on several tables within one database, Sybase IQ allows multiple readers and writers in a database, as long as the writers write to different tables.

❖ **Updating multiple tables from multiple accounts**

- 1 Connect to the demo database using connection name *marketing*:

```
CONNECT DATABASE asiqdemo  
AS marketing
```

- 2 Start a read/write transaction.

```
INSERT INTO fin_code
```

```
(code, type, description)
VALUES ('e9', 'expense', 'R & D')
```

- 3 Connect to the same database using a different connection name:

```
CONNECT DATABASE asiqdemo
AS sales
```

- 4 Start a read/write transaction using a different table:

```
INSERT INTO sales_order
(id, cust_id, order_date, sales_rep)
VALUES ('2088', '140', '05-29-98', '195')
COMMIT
```

This transaction will commit immediately because, although both are in *asiqdemo* database, the insert operations affect different tables.

For more details about transaction support, see the *Sybase IQ System Administration Guide*.

Deleting rows

Sometimes you will want to remove rows from a table. Suppose Rodrigo Guevara (employee ID 249) leaves the company. The following statement deletes Rodrigo Guevara from the employee table.

```
DELETE
FROM employee
WHERE emp_id = 249
```

Example: using the DELETE command

You can delete more than one row with one command. For example, the following statement would delete all employees who had a termination date that is not NULL from the employee table.

```
DELETE
FROM employee
WHERE termination_date IS NOT NULL
```

This example would not remove any employees from the database as the `termination_date` column is NULL for all employees.

With DELETE, the search condition can be as complicated as necessary. For example, if the employee table is being reorganized, the following statement removes from the employee table all employees in the 617 area code with employee ID 902 as manager. This WHERE clause is a compound search condition including a function (LEFT).

```
DELETE
FROM employee
WHERE LEFT( phone, 3 ) = '617'
AND manager_id = 902
```

Since you have made changes to the database that you do not want to keep, you should undo the changes as follows:

```
ROLLBACK
```

Updating rows

Other times, you may want to change certain rows based on the value of a particular column. The SET clause specifies the columns to update and their new values, and the WHERE clause specifies the rows to update.

For example, if the telephone area code changed from '508' to '978', you could change every row with the old area code to the new area code with the following command:

```
UPDATE customer
SET LEFT(phone,3) = '978'
WHERE LEFT (phone, 3) = '508'
```

Synchronizing join indexes

While you update a table that is involved in a join index, that join index is unavailable to other writers. Sybase IQ lets you control when it updates join indexes, because such updates may take a lot of time depending on the order of updates to various tables referenced by the indexes.

The `SYNCHRONIZE JOIN INDEX` command updates one or all of the join indexes in the current database belonging to the connected user. If you do not specify one or more indexes as argument(s), it synchronizes all join indexes in the current database.

To update a join index, Sybase IQ must update all the other join indexes that reference a table in the join, as well as all the join indexes that reference tables in those join indexes.

You may choose to issue this command before you finish a group of `INSERT` or `DELETE` statements, so that the data will become available to users as soon as the command finishes. You may alternatively choose to run a batch job to issue the command at a time when you know the load on the machine is light to avoid system slowdowns. After each `SYNCHRONIZE JOIN INDEX` statement, you need to issue a `COMMIT` statement.

Note For best performance, avoid issuing the `SYNCHRONIZE JOIN INDEX` command after every insert or delete.

In this chapter, we've made several changes to the employee table. To synchronize the employee/sales_order join index (`EMP_SO_JDX`), issue the following commands:

```
SYNCHRONIZE JOIN INDEX
EMP_SO_JDX
COMMIT
```


Deleting dbspaces, tables, and indexes

There are several ways to delete databases, dbspaces, tables, and indexes (including join indexes) in Sybase Central.

❖ Deleting dbspaces, tables and indexes

- 1 Right-click the instance of the object (for example, on the product table in the asiqdemo database)
- 2 Select Delete.

Other methods:

- Left-click the object and then click the Delete toolbar button 

- Left-click the object and press Alt+E,d.
- Use the DROP command in DBISQL. For complete syntax, see the *Sybase IQ Reference Manual*

About this chapter

In the previous chapter, you learned how to update data in your database. This chapter introduces data retrieval, in the form of views.

Using views

It is often impractical to type complicated queries repeatedly. SQL provides views that allow you to give names to frequently executed `SELECT` commands. A view is a computed table which is useful for security purposes, and for tailoring the appearance of database information to make data access straightforward.

Defining a view

Suppose that you frequently need to list a summary of employees and their departments. The following query produces the information you need.

Creating a view

To list employees and the departments to which they belong, type:

```
SELECT emp_fname, emp_lname, dept_name
FROM employee JOIN department
ON department.dept_id = employee.dept_id
```

emp_fname	emp_lname	dept_name
Fran	Whitney	R&D
Matthew	Cobb	R&D
Philip	Chin	Sales
Julie	Jordan	Finance
Robert	Breault	R&D
...

You can create a view that produces the results of this command as follows:

```
CREATE VIEW emp_dept AS
SELECT emp_fname, emp_lname, dept_name
FROM employee JOIN department
ON department.dept_id = employee.dept_id
```

This command creates a view called `emp_dept` that looks in many respects just like any other table in the database.

You can list everything in this view just as you do from a table:

Displaying view data

To list employees and the departments to which they belong, type the following:

```
SELECT *
FROM emp_dept
```

emp_fname	emp_lname	dept_name
Fran	Whitney	R&D
Matthew	Cobb	R&D
Philip	Chin	Sales
Julie	Jordan	Finance
Robert	Driscoll	R&D
...

It is important to remember that the information in a view is not stored separately in the database. Each time you refer to the view, SQL executes the associated `SELECT` statement to find the appropriate data.

On one hand, this is good; it means that if someone modifies the employee table or the department table, the information in the `emp_dept` view will be automatically up to date. On the other hand, if the `SELECT` command is complicated it may take a long time for SQL to find the correct information every time you use the view.

Providing names for the view columns

You can provide names for the view columns explicitly. First you must get rid of the original view definition as follows:


```
DROP VIEW emp_dept
```

Next, you can redefine the view with the new column names. Let's use Sybase Central to do this.

Managing views with Sybase Central

The Add View Wizard includes an optional template for view creation. When you open the Views container, the template functionality becomes available.

❖ Defining a view

- 1 Expand the server and database containers, then click the Views folder.
- 2 Click the New View toolbar button  or Alt+F, N, V.
- 3 Type the view name in the text box. For this tutorial, use EMP_DEPT.
- 4 Select the view's owner from the dropdown menu. For this tutorial, choose DBA.
- 5 Choose the option to use the View Creation wizard or a template. Choose the wizard. Click Next.
- 6 Type the SELECT statement to create the view. For example:

```
CREATE VIEW "DBA"."emp_dept" (/* view column
name,... */)
AS SELECT "department"."dept_id",
"department"."dept_name", "employee"."emp_id",
"employee"."emp_fname", "employee"."emp_lname",
"street", "employee"."city", "employee.state",
"employee.phone",
FROM "DBA"."department", "DBA"."employee"
```

- 7 Select File > Save View.

❖ Browsing view data

- 1 Run Interactive SQL.
- 2 Type a select statement in the format:

```
select * from <owner>.<view>
```

For example

```
SELECT * FROM "DBA"."emp_dept"
```

You can access the results from Sybase Central Java Edition using File > View data in Interactive SQL.

View restrictions and advanced uses

Views can be thought of as computed tables. Any `SELECT` command can be used in a view definition except commands containing `ORDER BY`. Views can use `GROUP BY` clauses, subqueries, and joins. Disallowing `ORDER BY` is consistent with the fact that rows of a table in a relational database are not stored in any particular order. When you use the view, you can specify an `ORDER BY`.

You can also use views in more complicated queries. Here is an example using a join:

```
SELECT LastName, dept_head_id
FROM emp_dept, department
WHERE emp_dept.Department = department.dept_name
```

Note Views cannot be used in a join index.

Using views for security

Views can be used to restrict access to information in the database. For example, suppose you wanted to create a user ID for the sales department head, Moira Kelly, and restrict her user ID so that it can only examine information about employees in the sales department.

Creating the new user ID

First you need to create the new user ID for Moira Kelly using the `GRANT` statement. From `DBISQL`, connected to the sample database as `dba`, type the following:

```
GRANT CONNECT TO M_Kelly
IDENTIFIED BY SalesHead
```

Granting permissions

Next you need to grant user `M_Kelly` the right to look at employees of the sales department.

```
CREATE VIEW SalesEmployee AS
SELECT emp_id, emp_lname, emp_fname
FROM "dba".employee
WHERE dept_id = 200
```

The table should be identified as `"dba".employee` for the `M_Kelly` user ID to be able to use the view.

Now you must give `M_Kelly` permission to look at the new view by entering:

```
GRANT SELECT ON SalesEmployee TO M_Kelly
```

Looking at the view

Connect to the database as M_Kelly and now try looking at the view:

```
CONNECT USER M_Kelly IDENTIFIED BY SalesHead ;
SELECT * FROM "dba".SalesEmployee
```

emp_id	emp_lname	emp_fname
129	Chin	Philip
195	Dill	Marc
299	Overbey	Rollin
467	Klobucher	James
641	Powell	Thomas
...

However, you do not have permission to look directly at the employee and department tables. If you execute the following commands, you will get permission errors.

```
SELECT * FROM "dba".employee ;
SELECT * FROM "dba".department
```

Using built-in
functions for secure
views

You can also take advantage of built-in Sybase IQ functions when creating secure views. In the following example, the view `secure_view` is intended to restrict access to records in the `secure_table` to specific users.

The view definition uses the `suser_name()` function, a built-in Sybase IQ function that stores a server user name for each user. The following view allows a user to see records if he or she has all of the following:

- A specific server user name (`suser_name`)
- A grade equal to or higher than the viewed record(s) (as defined by records in the `security_auth_table`)

```
CREATE VIEW secure_view AS
SELECT a.secure_name, a.security_code
FROM secure_table a, security_auth_table b
WHERE SUSER_NAME() = b.opr_name
      AND a.department_id = b.department_id
      AND a.grade <= b.grade
```

You may also create a view that restricts access by using the built-in function `suser_id()`, which stores a unique identification number for each user.

Restrictions on views

When defining a view, you should include only the columns which will be used in queries in the SELECT list of the view.

Do not use the CONTAINS predicate in a view that has a user-defined function, because the CONTAINS criteria will be ignored. Use the LIKE predicate with wildcards instead, or issue the query outside of a view.

The following may degrade the performance of views in queries (when compared to similar queries that do not use views).

- Nested views having any items appear on the select-list of both the inner and outer views that are not used by the query referencing the outer view
- Cases where there are joins both inside and outside the view, and the view does not contain an explicit or implicit GROUP BY clause
- Cases where a user breaks up a query by placing part of it into a view and now multiple predicates reference the same column inside and outside the view.
- In cases where a predicate outside the view contains a reference to a view select-list item plus either an outer reference or an uncorrelated subquery.

When a view specification contains an outer join, a predicate in a query that references the view cannot be “pushed down” into the indexes (since it would change the semantics at the outer join). So, you may notice that such queries run slower than a similar query that does not use a view.

Predicates are described in the *Sybase IQ Reference Manual*.

Index

A

- accessibility 6
- ad hoc queries 3
- Adaptive Server Anywhere 1, 13
 - database 13
- adding users to databases 65
- adding users to groups 67
- administering logins 65
- Agent
 - port 35
 - starting 29
- asiqdemo.db* file
 - about 10

B

- base tables 24

C

- case sensitivity
 - passwords 55
 - string comparisons 54
- catalog
 - system 26
- Catalog Store 14
- checksums 55
- collation sequence 56
- column indexes 71
 - types 71
- columns
 - editing properties 61
- COMMIT statement
 - about 85
- Compare indexes 72
- Component Integration Services 1
- CONNECT statement

- example 96
- connection parameters 47
- connection profile 14
- connectivity sample applications 10
- containers 29
 - expanding 38
 - selecting 37
- control keys
 - Sybase Central wizards 45
- conventions
 - documentation xi, xii
 - syntax xi
 - typographic xii
- CREATE VIEW statement
 - about 93
- creating databases 45
- cutting and pasting
 - Sybase Central 45

D

- data
 - case sensitivity 54
 - loading 76
 - tabular 20
- data dictionary 26
- data marts 3
- data types
 - about 21
- data warehouses 2
- database
 - sample 10
- databases
 - adding users 65
 - administering 28
 - Anywhere 13
 - collation label 56
 - connecting to 41
 - creating 45

Index

- creating with Sybase Central 45
- deleting 58
- Enterprise 14
- files 19
- IQ 15
- managing 28
- proxy 17
- relational 21
- reserve space 51, 52
- sample xii
- specifying IQ Page Size 51
- DBISQL
 - command line parameters 41
 - logon window 41
 - See* Interactive SQL
- DBISQLC 77
- dbspaces 14
 - deleting 82, 91
 - growth 51, 52
 - relocating 82
 - viewing with Sybase Central 39
- decision support systems 2
- deleting
 - databases 58
 - objects 91
 - tables 61
- deleting dbspaces 82
- disconnecting users from databases 68
- documentation 5
 - accessibility features xii
 - Adaptive Server Anywhere ix
 - conventions xi, xii
 - on CD ix
 - online ix
- dummy IQ table 10

E

- editing
 - column properties 61
 - tables 60
- enterprise transaction environment 1
- equijoins 75

F

- Federal Rehabilitation Act
 - section 508 xii
- fixed length format 77
- foreign keys
 - about 23
 - defined 23
- freelists 15
- function keys 6
- functions
 - viewing with Sybase Central 40

G

- GRANT statement
 - example 96
- groups
 - adding 64
 - viewing with Sybase Central 40

H

- High_Group indexes 72
- High_Non_Group indexes 72

I

- indexes
 - creating 71
 - definition 24, 39
 - deleting 91
 - join 16
 - types 71
- INSERT statement
 - examples 84
- inserting binary data 77
- inserting data
 - from backups 77
- Interactive SQL 5
- Interactive SQL Classic 77
- IQ Local Store 15
- IQ Message Log 15
- IQ Page Size

- default 51
 - specifying 51
- IQ Store 15
- iq_dummy table 10
- ISO_BINENG collation 56

J

- Java Access Bridge package 7
- java sample applications 10
- JAWS screen reader 7
- join indexes 16, 74
 - creating 75
 - synchronizing 17
- joins
 - definition 26

K

- keys
 - foreign 23
 - primary 23

L

- language interface 18
- loading data 76
 - from databases 77
 - from files 77
 - inserting values 77
- Local Stores
 - IQ 15
- logins
 - managing 65
- logs
 - message 20
 - transaction 20
- Low_Fast indexes 72

M

- make files 10

- manuals 5
- Message Log
 - IQ 15
- message log 20
- metadata 16
- mobile computing 1
- multiplex 17
 - stopping servers 44
- multiplex startup mode 33

N

- network server 4
- network server executable 4
- New Group toolbar button 64
- NULL value
 - allowed in columns 84

O

- objects 17
- odbc sample applications 10
- OLTP
 - See* On-Line Transaction Processing
- On-Line Transaction Processing 2

P

- passwords
 - and permissions 96
 - case sensitivity 55
- performance 20
- permissions
 - setting 63
- plug-ins 30
 - connecting 30
 - registering 30
- port number
 - specifying on Windows 42
- port numbers
 - in use 47
- primary keys
 - defined 23

Index

- procedures
 - setting permissions 63
 - viewing 62
 - viewing with Sybase Central 40
- program group 9
- programming interfaces 77
- projections 25
- proxy database 17

R

- Raw Device Access 10
- raw partitions
 - for temporary file pathname 51
- readme file 10
- relational database concepts 21
- relocating dbspaces 82
- reserve space 51, 52
- ROLLBACK statement
 - about 85
 - examples 90
- rows
 - updating 90

S

- sample applications 10
- sample database xii, 10
 - connecting to 41
 - iq_dummy table 10
 - starting 10
- screen readers 7
- section 508
 - compliance xii
- servers
 - definition 27
 - shutting down 43
 - starting 32
 - stopping 43
- Service Manager 10
- shortcut keys 6
- SQL 21
- standards
 - section 508 compliance xii

- standards and compatibility
 - section 508 compliance xii
- start_asiq* 4
- start_asiq.exe* 4
- starting multiplexes 32
- stopping servers 43
- Store
 - Catalog 14
 - IQ 15
 - Local 15
 - Temporary 20
- stored procedures
 - definition 25
- strings
 - case sensitivity 54
- Structured Query Language
 - See SQL
- Sybase Central 5
 - control keys 45
 - description 28
 - in program group 10
 - main window 29
 - starting 29
 - stopping 30
- Sybase IQ
 - benefits 3
 - components 4
 - definition 1
 - documentation 5
 - network server 4
 - sample database 10
 - Service Manager 10
 - User Administration 65
 - utilities 28
- system tables 26

T

- tables
 - characteristics 22
 - creating 59
 - deleting 61, 91
 - editing 60
 - iq_dummy 10
 - persistent 17

- temporary 17
 - viewing with Sybase Central 38
- tabular data 20
- TCP/IP protocol
 - for DBISQL connection 68
- temporary file
 - specifying 51
- temporary segment file 51
- Temporary Store 20
- transaction log 20
- Transact-SQL
 - viewing procedures in 63
- triggers
 - viewing 62
- tuples 22
- tutorials
 - Sybase Central 28

U

- updating rows 90
- User Administration 65
- user manuals 5
- users
 - adding to databases 65
 - viewing with Sybase Central 40
- utilities 28
 - database administration 28
- utility database
 - database name 58
 - starting with 58

V

- variable length format 77
- views
 - browsing data 95
 - defining 95
 - definition 25
 - deleting 94
 - permissions 96

W

- WD indexes 72
- WHERE clause 89, 90

