

SYBASE®

Introduction to Sybase IQ

Sybase IQ

15.2

DOCUMENT ID: DC38159-01-1520-01

LAST REVISED: April 2010

Copyright © 2010 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 trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

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	2
The Sybase IQ approach	2
Why Sybase IQ?	3
Sybase IQ features	4
Integration with other Sybase products	5
Sybase IQ components	6
The Sybase IQ database server	6
Administration utilities	6
Using the documentation	7
The Sybase IQ program group	7
The demo database	8
Preparing for the tutorial	11
CHAPTER 2 Terms and Concepts	13
Sybase IQ and data warehousing terms	14
Adaptive Server	14
Anywhere database	14
Catalog store	14
Component Integration Services (CIS)	14
Connection Profile	14
Dbfile	15
Dbpace	15
Free list	15
IQ database	15
IQ main store	15
IQ message log	16
IQ temporary store	16
IQ transaction log	16
Join index	17
Metadata	17

- Multiplex 17
- Object 17
- Partition key 18
- Proxy table 18
- Range partition 18
- SQL Anywhere 18
- Store 18
- Synchronize command 19
- Synchronization 19
- Table partition 19
- Tablespace 19
- Database computing concepts 20
 - Sybase IQ database files 21
 - How the pieces fit together 22
 - SQL and database computing 22
- Relational database concepts 23
 - Database tables 23
 - Primary and foreign keys 24
 - Other database objects 26
 - Queries 27
 - Other SQL statements 28
 - System tables 28

- CHAPTER 3 Running and Connecting to Servers 29**
 - Understanding database servers 30
 - Sybase Central and database management 30
 - Starting and stopping Sybase Central 31
 - Starting the database server 35
 - Connecting to a database 36
 - Navigating the main Sybase Central window 37
 - Viewing a database schema 37
 - Viewing the tables in a database 38
 - Viewing other database objects 39
 - Disconnecting databases and users 41
 - Connecting to databases with Interactive SQL 41
 - Sending commands to the database 43
 - Shutting down the database server 44

- CHAPTER 4 Managing Databases 45**
 - Managing databases 46
 - Creating databases 46
 - Deleting databases 61
 - Managing tables 61

	Creating a table and adding columns with Sybase Central.....	61
	Editing existing tables.....	63
	Deleting tables with Sybase Central.....	64
	Managing procedures	64
	Viewing stored procedure code with Sybase Central.....	65
	Managing users and groups.....	66
	Creating a login policy	66
	Adding a group to the database	69
	Adding a user to the database	72
	Managing connected users	75
CHAPTER 5	Indexing and Loading Data	77
	Creating column indexes.....	78
	Creating join indexes.....	81
	Updating join index data	83
	Loading data	83
CHAPTER 6	Managing Dbspaces	85
	Overview of dbspaces.....	86
	Creating dbspaces	86
	Specifying dbspace name and store	87
	Specifying dbspace files	88
	Altering dbspaces.....	89
CHAPTER 7	Updating Databases	91
	Adding rows to a table.....	92
	Canceling changes.....	93
	Making changes permanent.....	93
	Ensuring data consistency	94
	Deleting rows	96
	Updating rows	97
	Synchronizing join indexes.....	98
	Deleting dbspaces, tables, and indexes.....	99
CHAPTER 8	Using views	101
	Using views	102
	Defining a view.....	102
	View restrictions and advanced uses	104
	Using views for security	105
	Restricting access to information in the database.....	107
Index		109

About This Book

Audience

Read this book to get a hands-on introduction to Sybase® IQ. This book assumes no technical knowledge of Sybase IQ or any other Sybase product. For detailed information, refer to other books in the Sybase IQ documentation set.

Related Sybase IQ documents

The Sybase IQ 15.2 documentation set includes:

- *Release Bulletin* provides information about last-minute changes to the product and documentation.
- *Installation and Configuration Guide* provides platform-specific instructions on installing, migrating to a new version, and configuring Sybase IQ for a particular platform.
- *Advanced Security in Sybase IQ* covers the use of user encrypted columns within the Sybase IQ data repository. You need a separate license to install this product option.
- *Error Messages* lists Sybase IQ error messages referenced by Sybase error code, SQLCode, and SQLState, and SQL preprocessor errors and warnings.
- *IMSL Numerical Library User's Guide: Volume 2 of 2 C Stat Library* contains a concise description of the IMSL C Stat Library time series C functions. This book is only available to RAP – The Trading Edition™ Enterprise users.
- *Introduction to Sybase IQ* includes hands-on exercises for those unfamiliar with Sybase IQ or with the Sybase Central™ database management tool.
- *New Features Summary Sybase IQ 15.2* summarizes new features and behavior changes for the current version.
- *Performance and Tuning Guide* describes query optimization, design, and tuning issues for very large databases.
- *Quick Start* lists steps to build and query the demo database provided with Sybase IQ for validating the Sybase IQ software installation. Includes information on converting the demo database to multiplex.

-
- *Reference Manual* – Includes two reference guides to Sybase IQ:
 - *Reference: Building Blocks, Tables, and Procedures* describes SQL, stored procedures, data types, and system tables that Sybase IQ supports.
 - *Reference: Statements and Options* describes the SQL statements and options that Sybase IQ supports.
 - *System Administration Guide* – Includes two volumes:
 - *System Administration Guide: Volume 1* describes startup, connections, database creation, population and indexing, versioning, collations, system backup and recovery, troubleshooting, and database repair.
 - *System Administration Guide: Volume 2* describes writing and running procedures and batches, programming with OLAP, accessing remote data, setting up IQ as an Open Server, scheduling and event handling, programming with XML, and debugging.
 - *Time Series Guide* describes SQL functions used for time series forecasting and analysis. You need RAP – The Trading Edition™ Enterprise to use this product option.
 - *Unstructured Data Analytics in Sybase IQ* explains storage and retrieval of unstructured data within the Sybase IQ data repository. You need a separate license to install this product option.
 - *User-Defined Functions Guide* provides information about the user-defined functions, their parameters, and possible usage scenarios.
 - *Using Sybase IQ Multiplex* tells how to use multiplex capability, designed to manage large query loads across multiple nodes.
 - *Utility Guide* provides Sybase IQ utility program reference material, such as available syntax, parameters, and options.

**Related SQL
Anywhere
documentation**

Because Sybase IQ shares many components with SQL Anywhere Server, a component of the SQL Anywhere® package, Sybase IQ supports many of the same features as SQL Anywhere Server. The IQ documentation set refers you to SQL Anywhere documentation, where appropriate.

Documentation for SQL Anywhere includes:

- *SQL Anywhere Server – Database Administration* describes how to run, manage, and configure SQL Anywhere databases. It describes database connections, the database server, database files, backup procedures, security, high availability, and replication with Replication Server®, as well as administration utilities and options.
- *SQL Anywhere Server – Programming* describes how to build and deploy database applications using the C, C++, Java, PHP, Perl, Python, and .NET programming languages such as Visual Basic and Visual C#. This book also describes a variety of programming interfaces such as ADO.NET and ODBC.
- *SQL Anywhere Server – SQL Reference* provides reference information for system procedures, and the catalog (system tables and views). It also provides an explanation of the SQL Anywhere implementation of the SQL language (search conditions, syntax, data types, and functions).
- *SQL Anywhere Server – SQL Usage* describes how to design and create databases; how to import, export, and modify data; how to retrieve data; and how to build stored procedures and triggers.

You can also refer to the SQL Anywhere documentation in the SQL Anywhere 11.0.1 collection at Product Manuals at <http://sybooks.sybase.com> and in DocCommentXchange at http://dcx.sybase.com/dcx_home.php.



Introducing Sybase IQ

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

Topic	Page
Welcome to Sybase IQ	2
The Sybase IQ approach	2
Why Sybase IQ?	3
Sybase IQ features	4
Integration with other Sybase products	6
Sybase IQ components	6
The demo database	8
Preparing for the tutorial	11

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 SQL Anywhere, a small footprint version of Adaptive Server Enterprise, often used for mobile and occasionally connected computing. Component Integration Services are embedded within Sybase IQ and all Adaptive Server configurations to provide Sybase IQ users with direct access to relational or nonrelational databases on the mainframe, UNIX, or Windows servers.

There are slight differences in the SQL (Structured Query Language) syntax supported by Sybase IQ, SQL Anywhere, and Adaptive Server Enterprise. *Reference: Building Blocks, Tables, and Procedures* provides an overview of compatibility of the SQL used by Sybase IQ, SQL Anywhere, and Adaptive Server Enterprise. See Appendix A, “Compatibility with Other Sybase Databases.”

The Sybase IQ approach

A Sybase 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 let many users update the database instantly and accurately, without interfering with one another. By contrast, in a Sybase IQ database, fast query response for many users is most important.

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

- Data is stored in columns, not rows.
- Placing indexes on all columns provides a performance advantage.
- A large page size provides a performance advantage.
- A large temporary cache provides 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 Sybase IQ database, keep these fundamental differences in mind. For details on setting up Sybase IQ for optimal performance, see *System Administration Guide: Volume 1*, *System Administration Guide: Volume 2*, and *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, specifies 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 Table 1-1:

Table 1-1: Comparison of conventional relational databases and decision support databases

Conventional databases are designed for...	Decision support databases are designed for...
Clerical users	Managerial, clerical and analytical users
Current data	Historical data
Detailed data	Summarized and detailed data
Highly changeable data	Stable data
Day-to-day operations	Strategic decision support
Transactions	Analysis
Structured queries	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 features

Sybase IQ supports an interactive approach to decision support that includes:

- Intelligent query processing that use index-only access plans to process any type of query.
- Ad hoc query performance on uniprocessor and parallel systems.
- Multiplex capability for managing large query loads in a multi-server configuration.
- Fully-flexible schema support.

- Efficient query execution without query-specific tuning 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 input/output (I/O).

Integration with other Sybase products

You can integrate Sybase IQ with other Sybase products to build and deploy a complete analytics infrastructure:

Sybase WorkSpace An integrated development environment enabling developers and analysts to collaborate when modeling and developing database applications. Sybase WorkSpace presents user-interface tools for Enterprise Modeling and Database Development. Database Development provides graphical tools for SQL development, and advanced editing and debugging of stored procedures, triggers, user-defined functions, and database events. For more information, see the documentation collection installed with Sybase WorkSpace.

Sybase PowerDesigner® Delivered as the Enterprise Modeling component of Sybase WorkSpace, Sybase PowerDesigner® is a graphical enterprise modeling environment which provides integrated modeling through standard methodologies and notations; automatic code generation through customizable templates; reverse engineering capabilities to document and update existing systems; a scalable enterprise repository solution with strong security; automated, customizable reporting capabilities; and an extensible environment. For information on using Sybase PowerDesigner® for database design and modeling, see the documentation collection installed with Sybase PowerDesigner®.

Sybase ETL Provides a scalable grid architecture that enables parallel transformation processing across operating system boundaries and computers. Sybase ETL lets you extract data from multiple heterogeneous data sources and load it into one or more data targets using a comprehensive set of transformation functions. Sybase ETL capabilities include: data extraction, data transformation, and data loading. For information on using Sybase ETL for extract, transform, and load operations, see the Sybase ETL 4.8 documentation collection at <http://sybooks.sybase.com>.

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 SQL 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 or Linux operating systems, use *start_iq*.
- On Windows, use *start_iq.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 Sybase 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 *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 *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 *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.

The Sybase IQ program group

On Windows, click Start, point to Programs, choose Sybase | Sybase IQ 15.2 to reach the Sybase IQ program group. 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 – Starts the Java-based Interactive SQL utility for sending SQL statements to a database.
- ODBC Administrator for 32-bit or 64-bit platforms – Manages ODBC connections to your databases.

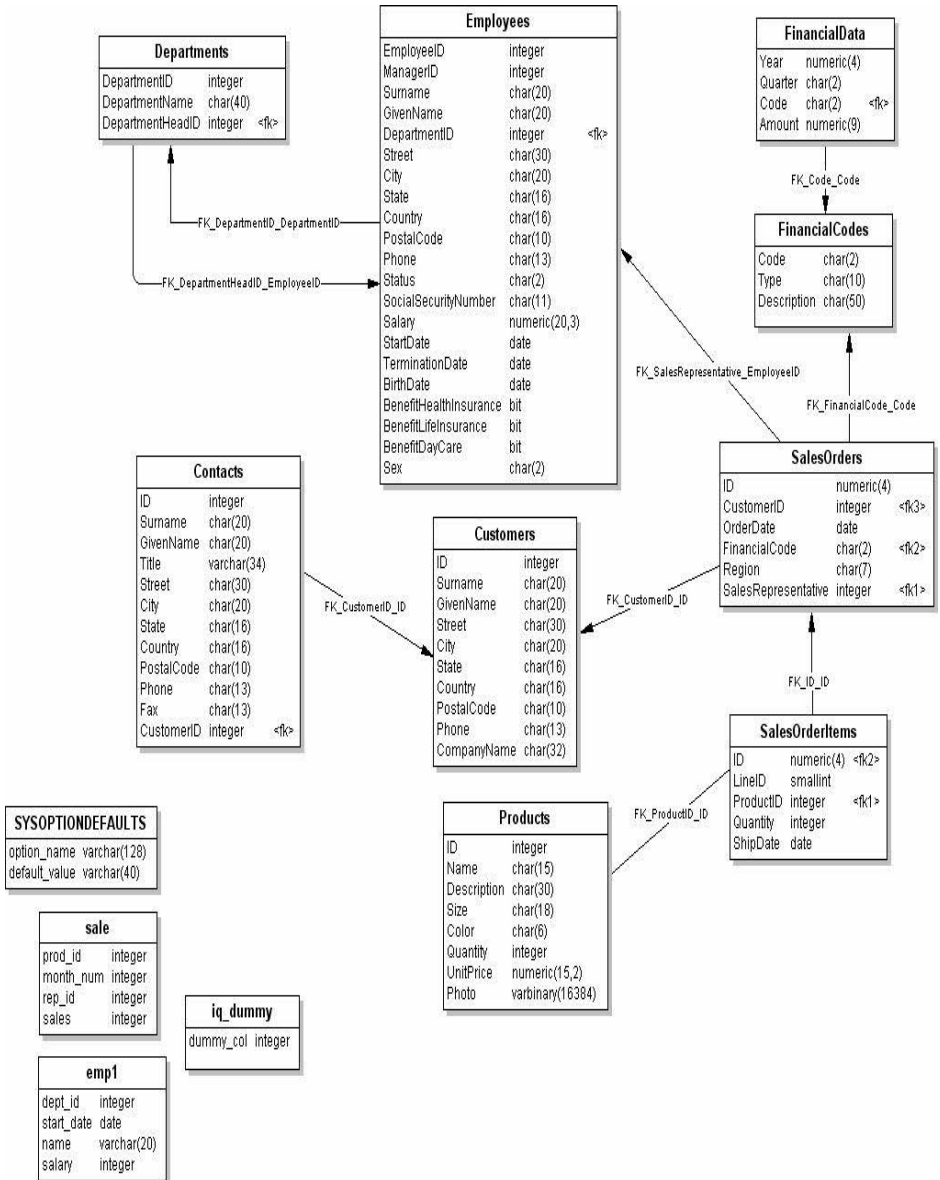
- Raw Device Access – Adds or removes users and raw devices from the registry.
- Sample Applications and Projects – Displays a list of sample applications and projects. For information on running these samples, see *SQL Anywhere Server – Programming*.
- Start Sybase IQ Demo Database – Starts the database server running the demo database.
- Sybase Central – 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 Sybase IQ as a Windows service.
- Sybase on the Web – Starts a browser and opens a page with an introduction to Sybase online resources.

The demo database

Many of the queries and code samples in this document use the demo database included with Sybase IQ as a data source. The demo database (*iqdemo.db*) is stored in *\$IQDIR15/demo* on UNIX or *%ALLUSERSPROFILE%\SybaseIQ\demo* on Windows.

Tables in the demo database are delineated by ownership. Tables owned by GROUPO contain information about a fictional company; tables owned by the DBA include utility and sample data tables.

Figure 1-1: Tables in the demo database



GROUPO tables in the database

GROUPO tables (see below) contain 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).

Table name	Contents
Contacts	Customer contacts and sales leads.
Customers	Customer names and addresses.
Departments	Company departments, such as manager and name.
Employees	Employee information, such as name, salary, and location.
FinancialCodes	Each expense and revenue item has a financial code.
FinancialData	Quarter-by-quarter financial information about the company.
Products	Product information, such as price and quantity available.
SalesOrderItems	Sales order items. Each order consists of one or more items. Information about sales order items is held in a separate table.
SalesOrders	Individual sale orders, including customer ID, OrderDate, FinancialCode, Region, and SalesRepresentative.

DBA tables in the database

Tables owned by the DBA in *iqdemo.db* include utility and sample data tables.

Table	Description
iq_dummy	<p><i>iq_dummy</i> is a one-row, one-column dummy table that you can use to extract information from the database. For example, running the NOW() function against <i>iq_dummy</i> returns the current date and time:</p> <pre>SELECT NOW() FROM iq_dummy</pre> <p>Use of the DUMMY system table in Sybase IQ is implied for all queries that do not have a FROM clause. For more information, see DUMMY system table in Reference.</p>
empl	Sample employee table that includes dept_id, start_date, name, and salary columns.
sale	Sample sales table that includes prod_id, month_num, rep_id, and sales columns.

SYSOPTION-DEFAULTS table

SYSOPTIONDEFAULTS is a utility table owned by DBO that contains all Sybase IQ option names and values. You can query this table, if you need to see all option default values.

Case sensitivity

The demo 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 demo database. Note that, unlike the demo database, any Sybase IQ databases you create are case sensitive by default.

Preparing for the tutorial

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

Terms and Concepts

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.

Topic	Page
Sybase IQ and data warehousing terms	14
Database computing concepts	20
Relational database concepts	23

Sybase IQ and data warehousing terms

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

Adaptive Server

Adaptive Server is an integrated set of Sybase software products for relational database applications. You can use Sybase IQ to query data in ASE databases.

Anywhere database

Every Sybase IQ database uses an SQL Anywhere database for the catalog store. This book refers to a SQL Anywhere database as an Anywhere database.

Catalog store

The portion of each Sybase IQ database that contains its metadata. (Metadata describes the layout of the Sybase IQ tables, columns, and indexes.) The catalog store contains the SYSTEM dbspace and up to 12 additional other catalog dbspaces. The default name is *dbname.db*.

Component Integration Services (CIS)

Embedded within Sybase IQ and all Adaptive Server configurations, CIS provides Sybase IQ users with direct access to relational or nonrelational databases on the mainframe, UNIX, or Windows servers.

Connection Profile

Stores information needed to establish a connection to a running Sybase IQ server. The profile is primarily used to simplify user connections to a server. Sybase IQ extends connection profiles to facilitate starting servers and creating databases.

Dbfile

An operating system file used to store data for a Sybase IQ database. Each dbfile has a corresponding logical filename and physical file path. Each dbspace name, dbfile name, and physical file path must be unique. The dbfile name can be the same as the dbspace name.

The *SYSDBFILE* view shows all the dbfiles in your database, including the catalog dbspace file, the IQ message file, dbfiles in the IQ main and temporary dbspaces, the transaction log file, and the SA temporary file.

Dbspace

A logical collection of dbfiles. If a database runs out of room, you can expand it by adding additional dbspaces. Users can move Sybase IQ data off of disks and take the disks offline without any downtime.

Free list

A structure that Sybase IQ uses to track which blocks are in use by a dbspace.

IQ database

A database that you create using a Sybase IQ server. IQ databases are specially indexed to take advantage of the query speed of Sybase IQ.

Each IQ database that you create includes three stores: an IQ main 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 main store

The IQ main store is the portion of each Sybase IQ database that contains the *IQ_SYSTEM_MAIN* dbspace and other user dbspaces. The IQ main store is contains persistent database structures, such as backup metadata and rollback data for committed transactions.

Sybase recommends that you avoid storing user tables and indexes in IQ_SYSTEM_MAIN and instead create additional dbspaces, called user main dbspaces, to store user tables and indexes. The IQ main store is sometimes called the IQ store.

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.

IQ_SYSTEM_MSG is a system dbspace that points to the file path of the database IQ message log file. IQ_SYSTEM_MSG is not considered a store because it doesn't store any data. Sybase IQ logs error messages, status messages and insert notification messages in this file.

IQ temporary store

The IQ temporary store contains the IQ_SYSTEM_TEMP dbspace. The IQ temporary store is the portion of each IQ database that stores temporary tables and temporary scratch space data structures.

The database server uses temporary data structures to sort and process data. Data in these tables persists only as long as you are connected to the database.

IQ transaction log

The IQ transaction log records changes to the database. The transaction log includes version information, free space, and other information you can use to recover from a system failure. By default, the transaction log is created in the same directory as the catalog store. The default name for this dbfile 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.

A **join index** is 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 is data that describes the data in your database – for example, the size and data type of each column in a table. Metadata for each Sybase IQ database is stored in the catalog store. (See “IQ database” on page 15.)

Multiplex

A powerful feature in Sybase IQ that provides application scalability through a clustered server configuration. Sybase IQ **multiplex** allows concurrent data loads and queries via independent data processing nodes connected to a shared data source. Each multiplex server has its own catalog store and IQ temporary store and all the servers share a common IQ store. For more information, see *Using Sybase IQ Multiplex*.

Object

An **object** can be 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.

Partition key

A **partition key** is a table column defined by the table creator that determines how a table should be partitioned.

Proxy table

A **proxy table** is an table object that maps to a table on a remote server, and whose column attributes and index information are derived from the object at the remote location.

You can use proxy tables to search data in multiple SQL Anywhere servers, ASE databases, and non-Sybase databases. Conversely, you can also create proxy tables that enable you to query your Sybase IQ and Anywhere databases.

Range partition

A **range partition** is logical subset of table rows based on the values of a single table column.

SQL Anywhere

SQL 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.

SQL Anywhere is specifically designed to use fewer memory and disk resources than the average database management system. Sybase IQ is an extension of SQL Anywhere, and supports many of the same features.

Store

A **store** is one or more dbspaces that store persistent or temporary data for a special purpose. Sybase IQ has three stores: the catalog store, the IQ main store, and the IQ temporary store.

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

Synchronization brings an outdated multiplex secondary node server up to date.

Table partition

A **table partition** is a collection of rows that is a subset of a user-created table. A given row cannot be placed in two different partitions. Each partition can be placed in its own dbspace and managed individually.

Tablespace

A **tablespace** unit of storage within the database that may be administered as a logical subset of total storage. You may allocate individual objects and subobjects to individual tablespaces. A tablespace in Sybase IQ is referred to as a **dbspace**.

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.
- **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 ODBC, 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 ODBC and 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

Default physical file name	Internal name	Description	Number created
<i>dbname.db</i>	Not applicable	SQL Anywhere dbspace file. Part of the catalog store.	One or more per server
<i>dbname.log</i>	Not applicable	SQL Anywhere transaction log. Part of the catalog store.	One for each server
<i>dbname.iq</i>	IQ_SYSTEM_MAIN	The initial dbfile file for the IQ main store. (User specifies the names for additional files).	One or more for each database
<i>dbname.iqtmp</i>	IQ_SYSTEM_TEMP	The initial IQ temporary dbfile file.	One or more for each database
<i>dbname.iqmsg</i>	IQ_SYSTEM_MSG	IQ human-readable trace file of debugging output.	One for each database
<i>dbname.lmp</i>	Not applicable	IQ human-readable license file.	One for each database

Note The DBA can 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, 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 temporary stores contain most of your tabular data. Each database has its own unique files for temporary data. If you don't specify a file, Sybase IQ creates one automatically for use as a temporary storage space.

You can manage the contents of these files using Sybase Central, see “Managing Databases” on page 45

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

Client applications use Structured Query Language (SQL) statements to carry out database tasks, such as retrieving information or inserting rows into tables.

Depending how a client application is developed, SQL statements could 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 SQL statements to the database server. The database server executes them and returns results 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 which interface and network protocol you use, SQL statements are sent to a server, and the results of SQL statements return 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 Sybase IQ shares with typical 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 only 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.

Primary keys

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. 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 the demo database, the SalesOrderItems table includes these columns:

- An ID column that identifies the customer's order.
- A LineID column that provides an identification number for each item of the sales order.
- A ProductID column that identifies the product that the customer ordered.
- A Quantity column that shows how many items were ordered.
- A ShipDate column that identifies the date the order shipped.

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

Foreign keys

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

Example

The demo database has one table holding employee information and one table holding department information. The Departments table has these columns:

- DepartmentID – An ID number that identifies the department. This is the primary key for the table.
- DepartmentName – A column holding the name of the department.
- DepartmentHeadID – 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 Employees table. Instead, the Employees table contains a column holding the employee's department ID. This is called a foreign key to the Departments 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 Sybase IQ, see <i>System Administration Guide: Volume 2</i>.</p>
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, UniPrice
FROM Products
WHERE UnitPrice > 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, this query joins the `SalesOrderItems`, `Products`, `Employees`, and `SalesOrders` tables to calculate sales totals and revenue:

```
SELECT Products.ID AS 'Product Code',
       Products.Name AS Item,
       Products.Description AS Style,
       SUM(SalesOrderItems.Quantity) AS Sales,
       Products.UnitPrice,
       SUM(SalesOrderItems.Quantity * Products.UnitPrice)
       AS Revenue FROM Employees
JOIN SalesOrders ON SalesOrders.SalesRepresentative =
  Employees.EmployeeID
JOIN SalesOrderItems ON SalesOrderItems.ID =
  SalesOrders.ID
JOIN Products ON Products.ID =
  SalesOrderItems.ProductID
GROUP BY Products.ID, Products.Description,
  Products.Name, Products.UnitPrice
ORDER BY Sales
```

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.

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 Sybase 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 before inserting data.

Running and Connecting to Servers

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

Topic	Page
Understanding database servers	30
Sybase Central and database management	30
Starting and stopping Sybase Central	31
Starting the database server	35
Navigating the main Sybase Central window	37
Connecting to databases with Interactive SQL	41
Sending commands to the database	43

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. A Sybase 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 for each server.

You can use multiple Sybase IQ servers to combine the use of Sybase IQ, Anywhere, and Enterprise databases. You can also use multiple Adaptive Server Enterprise servers to combine the use of Sybase 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. Users with large query loads and shared disk arrays may create a Sybase IQ multiplex to 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 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 6.0, which ships with Sybase IQ 15.2, to administer Sybase IQ 15.2 servers and databases.

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 *Reference: Statements and Options* 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. 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 *System Administration Guide: Volume 1* for details.

For more information about Sybase IQ utilities, see Chapter 4, “Database Administration Utilities” in the *Utility Guide*.

Note Although Sybase Central can also be used to manage other Sybase databases, this chapter describes only how to use Sybase Central 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 be ready to use Sybase Central.

Note Most of the examples in this tutorial uses the demo database. You may want to make a copy of the demo database and store it in a separate directory so that you can restore it after making changes.

❖ Starting Sybase Central on UNIX systems

1 Do one of the following to set the environmental variables:

```
% source $SYBASE/IQ-15_2/IQ-15_2.csh
```

or

```
% . $SYBASE/IQ-15_2/IQ-15_2.sh
```

Always source this file before invoking utilities like Sybase Central or the Sybase IQ Agent.

- 2 If you plan to administer multiplex database(s), start the Sybase IQ Agent before running Sybase Central:

```
% $IQDIR15/bin<platform>/S99SybaseIQAgent15
```

Where *<platform>* is the name of your operating system: */bin64* or */bin32*.

- 3 Type this command to start Sybase Central:

```
% scjview
```

Note On UNIX systems, for proper display in Sybase Central 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.

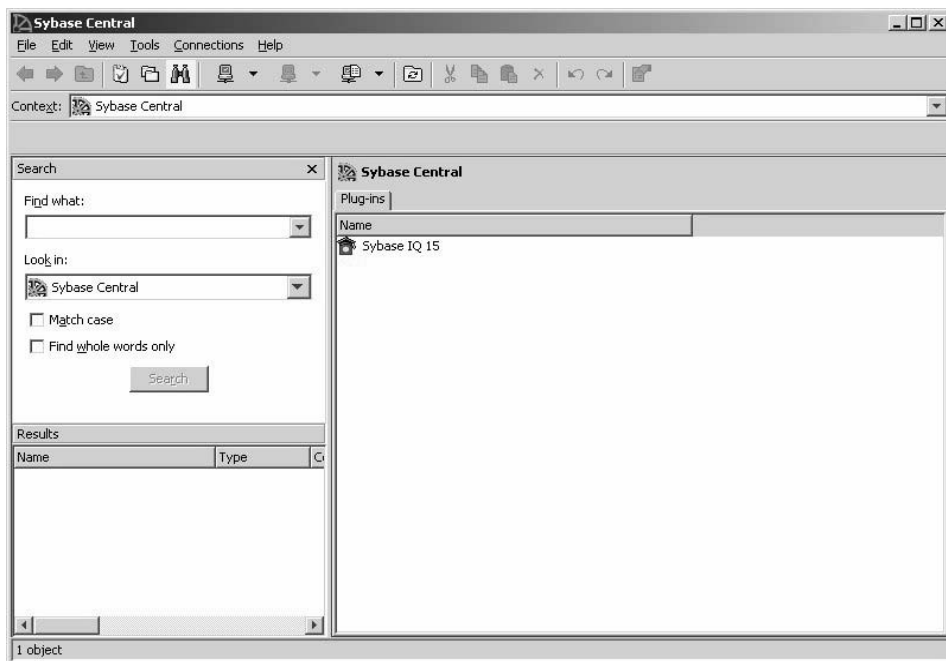
❖ Starting Sybase Central on Windows

- Click Start, point to Programs, choose Sybase | Sybase IQ 15.2 | Sybase Central.

Note The Fast Launcher option, which starts Sybase Central faster, is only available on Windows. To enable it, use Tools > Options. The Fast Launcher option requires a significant amount of memory and the impact on your system depends on the system configuration.

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 pane displays tasks, folders, or search options. The right pane displays additional options, actions, or views associated with the object or task you select. You can use the options on the Toolbar or View menu to change the perspective of the left pane.

Plug-ins for Sybase Central, such as the Sybase IQ database management system, are graphical tools for managing a particular product. Installing a Sybase product also installs its plug-in. The next time you start Sybase Central, the new product automatically “plugs in” to Sybase Central and appears in the main window.

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 Click Tools, choose 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 `$IQDIR15/java/iq.jpr`.
- 3 Click OK.

Stopping Sybase Central

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

Running the Sybase IQ Agent

The Sybase IQ Agent must be running to do many multiplex operations (such as creating query servers) on Sybase Central. The Sybase IQ Agent generally is started whenever you reboot a Windows machine where Sybase IQ is installed. Before you administer a multiplex, verify that the Sybase IQ Agent is running.

❖ Checking the Sybase IQ Agent on Windows

- 1 On the Control Panel, choose Administrative Tools | Services.
- 2 On the Services console, check the status of the Sybase IQ Agent 15.2.
- 3 If Status does not say Started, right-click the agent, choose Start.

❖ Checking the Sybase IQ Agent on UNIX

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

```
% stop_iq -agent
```

- 2 If no running agent is owned by your username, change directory to `$IQDIR15/bin<platform>` and run `S99SybaseIQAgent15`.

To start the agent using the host name returned by `uname -n` use the optional `-host` parameter, as follows:

```
S99SybaseIQAgent15 -host
```

To start the agent using the host's alias enter:

```
S99SybaseIQAgent15 -host <foo>
```

where *foo* is an alias present in the `/etc/hosts` file.

Starting the database server

This section describes how to start a database server running the demo database. See Chapter 2, “Running Sybase IQ” in *System Administration Guide: Volume 1* for more information.

Note If your system has multiple users or is one of several machines in a subnet, edit the file `$IQDIR15/demo/iqdemo.cfg` to specify a unique port and server name. Change the arguments to the `-n servername` switch and `-x tcpip{port=####}` to specify a server name and port number that are not in use. If SQL Anywhere is installed on your system, do not use port number 2638.

❖ Starting a server running the demo database from Windows

- 1 Log in to the host machine.
- 2 Click Start, point to Programs, choose Sybase | Sybase IQ 15.2 | Start Sybase IQ Demo Database.

When you start a server successfully, the server icon appears in the system tray. The server name is your system name followed by an underscore and `iqdemo`.

❖ Starting a server running the demo database from the command line (UNIX or Windows)

- 1 Log in to the host machine.
- 2 Change to a directory where you have write permission and make a copy of the `$IQDIR15/demo` directory there (`%IQDIR15%\demo` on Windows).
- 3 At the command prompt, type the following command:

```
start_iq @iqdemo.cfg iqdemo.db
```

This starts a server running the demo database.

❖ Starting a server running the demo database with Sybase Central

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

- 1 Click Tools, point to Sybase IQ 15.2, choose Start Server.
- 2 Follow the instructions on the screen.

See Start Server Wizard in Sybase IQ 15.2 online help for specific connection parameters.

Connecting to a database

After you start a server, you can connect to the databases on that server. You can provide the connection parameters on a connection dialog or use a connection profile to connect to the database.

❖ **Opening the Connect dialog**

- 1 Click Connections, choose Connect with Sybase IQ 15.2.
- 2 On the connection dialog, add the appropriate connection parameters.

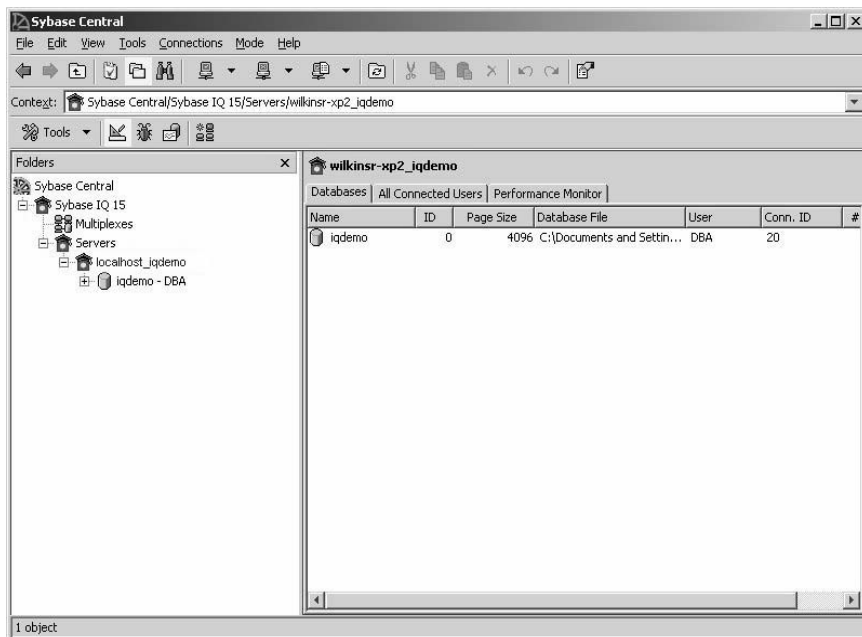
❖ **Creating a Connection Profile**

Connection profiles store connection parameters to databases you use frequently.

- 1 Click Connections, choose Connection Profiles.
- 2 On the Connection Profiles dialog, click New, and create the profile.

Navigating the main Sybase Central window

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



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:

- In Tasks view, the Contents tab in the right pane displays a folder view of database objects. To perform a particular task, double-click the object in the right pane and a set of related tasks appear in the left pane. Click the task you want to perform from the task list.

- In Folders view, database objects appear in a hierarchical tree in the left pane. Clicking a branch of the tree expands or contracts the view; details about the object appear in the right panel. For example, if you select the Sybase IQ plug-in from the left panel, the right panel displays servers, utilities, and services for Sybase IQ.
- The Search pane lets you search for items in the object hierarchy. To search for an item, type the name of the object in the Find what box, choose an object class from the Look in dropdown, then click Search. Click on an object in the Results panel to display the object in the right pane.

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.

Viewing the tables in a database

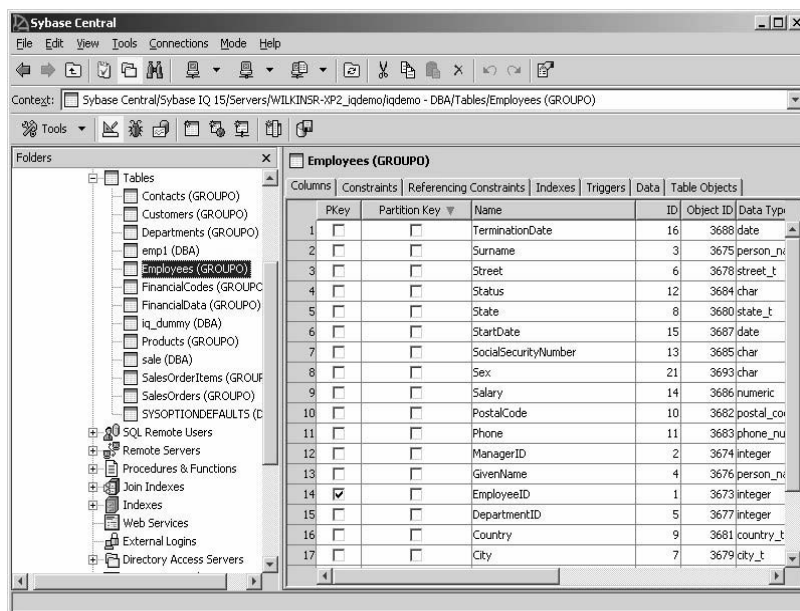
The following steps show how to examine the contents of a table folder in a database container.

❖ Viewing tables in a database

- 1 Connect to a database, then do one of the following:
 - In Tasks view, double-click the Tables folder in the right panel, then double-click a table to see the column definitions.
 - In Folders view, click the Tables folder, then click the table to see the column definitions in the right panel.

The right panel includes tabs that expose tables's columns, Constraints, Referencing Constraints, Indexes, Each table object contains folders for columns, foreign keys, relations, indexes, and more.

- Click on a tab in the right panel to expose the table's Columns, Constraints, Referencing Constraints, Indexes, Triggers, Data, and Table Objects.



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	A logical collection of dbfiles within a database.
Directory Access Servers	Gives you access to the local file system of the computer running the database server. You can create proxy tables for the directory access server to access the files and directories on the server computer.
Domains	Sometimes called user-defined data types, domains are aliases for built-in data types.
Events	Automate database administration, such as a stored procedure scheduled to execute out at a certain time.

Object	Description
External Environments	Refer to programming languages, utilities, and objects external to the database server.
External Logins	Provide access to a directory on a remote or directory access server.
Indexes	Structures that store and represent data 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.
Join Indexes	Joins that are created on sets of columns rather than individual columns, and represents a full outer join of two or more tables.
Login Mappings	Map a Windows user profile or Kerberos principal to an existing database user.
Login Policies	Set of options that define rules to be applied when a user connects to a database.
Procedures & Functions	SQL procedures for a module-based language.
Publications	Identify replicated data in SQL Remote. In a Sybase IQ database, publications can only be created on SQL Anywhere tables.
Remote Servers	Give you access to data located on separate database servers.
SQL Remote Subscriptions	Creates links from a publisher to a remote database (subscriber).
SQL Remote Users	Define remote databases in a SQL remote setup.
System Triggers	Events that modify the system tables.
Tables	Base tables stored in the database.
Triggers	Execute a SQL statement or procedure automatically when someone modifies the data.
Users & Groups	Mechanisms that identify database users and privileges.
Views	Computed tables, stored in the database as a query and evaluated when accessed.
Web Services	Describe how to process HTML and XML requests made to the web server.

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

Disconnecting databases and users

❖ Disconnecting from a database

- Do one of the following:
 - Click Connections, and choose Disconnect.
 - Click the Disconnect button on the Toolbar.
 - Right-click the database container, and choose Disconnect.

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.

Connecting to databases with Interactive SQL

When a database server is running, you can connect to it from an application. In this example, we'll connect to the demo database, *iqdemo*.

❖ Connecting from the command line

Ensure that the demo database is loaded on a running server, as described in “Starting the database server.”

- 1 Open a command prompt on your UNIX or Windows system.
- 2 To start dbisql, enter:

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

The `-c` parameter specifies connection parameters. For a complete list of connection parameters, see Chapter 4, “Connection and Communication Parameters,” in the *System Administration Guide: Volume 1*.

Note The default user ID and password for the *iqdemo* database and other Sybase 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 the Windows program group

- 1 Start the server, as described in “Starting the database server”

Select Start | Programs | Sybase | Sybase IQ 15.2 | Interactive SQL.

- 2 On the Identification tab, enter DBA in the User box, and sql in the Password box.

This is the default user ID and password for Sybase IQ databases. DBA is not case-sensitive; sql is case-sensitive, and must appear in all lowercase letters.

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” and “Sybase IQ Connections” in the *System Administration Guide: Volume 1*.

- 3 On the Database tab, choose a server from the Server name box. Click Find to search for running servers.

Server names appear in this format *<hostname>_dbname*. If the host name is *localhost-xp* and you are running the sample *iqdemo* database, the server name would be *localhost-xp_iqdemo*. If the server is remote, type *host:port* instead of the server name.

- 4 Choose a database from the Database name box. When there is only one database running, you can leave this field blank.
- 5 What you do next depends on the server location:
 - If you are running dbisql against a server installed on the same machine, you must also choose an appropriate Database file.
 - If you are running dbisql against a network server, click the Network tab, click the TCP/IP box, then specify the Host name and Port number.
- 6 Click OK to open Interactive SQL.

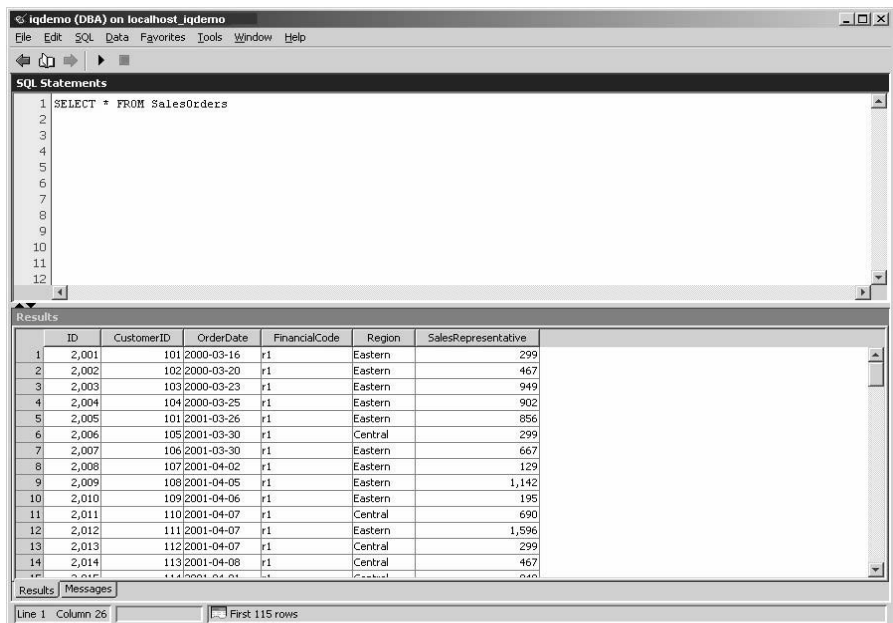
Other applications connect to the database server in much the same way. To make the connection, they must submit both user ID and password. The *Utility Guide* gives more details about how to use dbisql.

Sending commands to the database

After you connect, you can send commands to the database. For example, type the following line in the SQL Statements window:

```
SELECT * FROM SalesOrders
```

Click the right-facing triangle button on the Toolbar or press F9 to execute the query. The SQL statement displays all (*) the columns from the SalesOrders table.



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

Shutting down the database server

DBA permission is required to shut down the network server; 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.

Sybase Central

In Sybase Central, do one of the following to shut down your database server.

- In Folder view, right-click the server, and choose Stop Server.
- In Tasks view, click Work with server... from the list of Other Tasks. Click Other Tasks, double-click the Servers icon, right-click the server, and choose Stop Server.

Interactive SQL

In Interactive SQL, use the STOP ENGINE statement to stop a database server:

```
STOP ENGINE [database-server-name] [UNCONDITIONALLY]
```

The STOP ENGINE statement stops the specified database server. If you supply the UNCONDITIONALLY keyword, the database server is stopped even if there are other connections to the database server. By default, the database server will not be stopped if there are other connections to it.

You can only use a STOP ENGINE statement in Interactive SQL; you cannot use a STOP ENGINE statement in stored procedures, triggers, events, or batches.

Note In “Managing Databases” on page 45, you will learn how to create and manage databases on your server.

Managing Databases

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

Topic	Page
Managing databases	46
Managing tables	61
Managing procedures	64
Managing users and groups	66

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 to view properties or launch creation wizards. Selecting an object within a database also displays a creation wizard toolbar button.

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

Creating databases

Sybase Central includes a wizard to help you create a new IQ database. The wizard displays a series of dialog boxes with options that help you define characteristics of the database. To start the Create Database Wizard, click Tools, point to Sybase IQ 15.2, and choose Create Database.

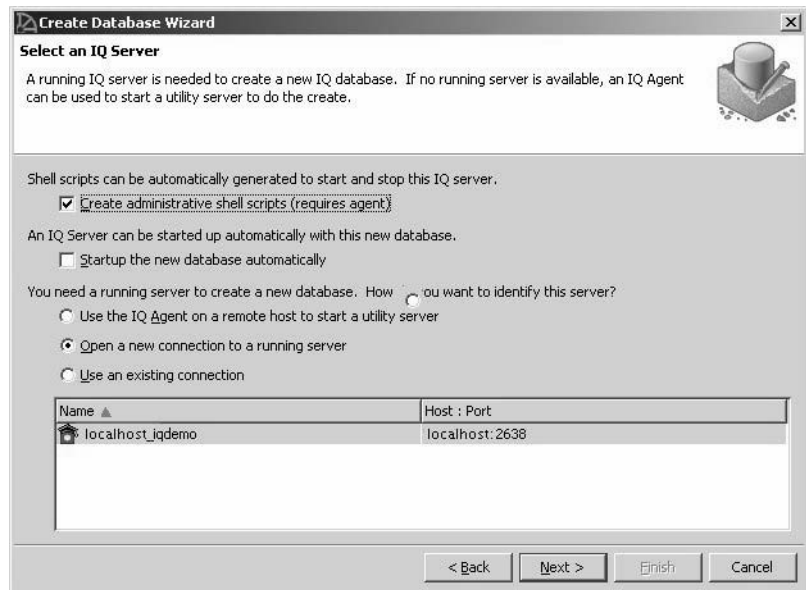
Multiplex databases

To create a multiplex database, see *Using Sybase IQ Multiplex*.

Choosing an IQ server

When you start the database creation wizard, the first dialog prompts you to select and define the server options you want to use. A running Sybase IQ server is needed to create a new Sybase IQ database. If no running server is available, you can use an IQ Agent to start a utility server to create the database.

After you select the server you want to use, click Next to move to the next dialog.



Create Administrative Shell Script

This option generates a Create Database statement based on the information that you provide. Sybase Central names this script *CreateDb.sql* and stores the file in the database directory.

Starting the new database automatically

After you create a new Sybase IQ database, Sybase Central can start a Sybase IQ server automatically and connect to the database.

Click the “Startup the New Database Automatically” check box to enable this option. Starting an IQ server requires that a running IQ Agent process be on the destination host.

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

Identifying a running server

Creating a database requires an IQ server. To identify the server you want to use, do one of the following:

- If there is already an IQ server running on the destination host, click the Active radio button so the plug-in uses that server to execute the create statement. You specify connection information for this server on the next page.

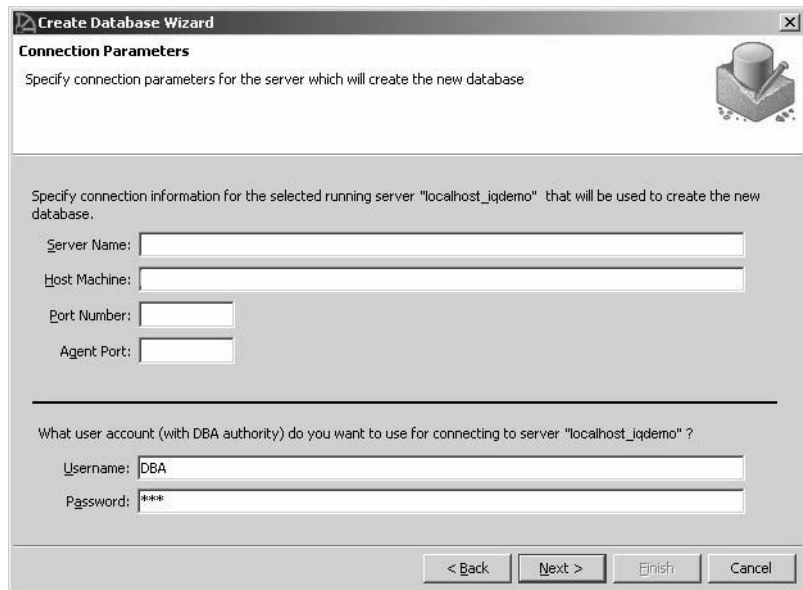
- If there is no active IQ server running on the desired host, the plug-in can startup a Utility server automatically if you click the Utility radio button for this option. As with startup, an IQ Agent is required to automatically startup the utility server. After the database is created, the plug-in shuts down the utility server automatically.

Note On Windows, 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.

Setting connection parameters

To create the database, Sybase Central must connect to a Sybase IQ server. This dialog prompts you for the server connection parameters, including the server name, host name, server port number, agent port number, user ID, and password.

After you define the connection parameters, click Next to move to the next dialog.



The screenshot shows a Windows-style dialog box titled "Create Database Wizard" with a close button (X) in the top right corner. The main title is "Connection Parameters". Below the title, there is a subtitle: "Specify connection parameters for the server which will create the new database." To the right of this text is a small icon of a server rack. Below this is a separator line. The next section is titled "Specify connection information for the selected running server 'localhost_jqdemo' that will be used to create the new database." and contains four input fields: "Server Name:", "Host Machine:", "Port Number:", and "Agent Port:". Below these is another separator line. The final section is titled "What user account (with DBA authority) do you want to use for connecting to server 'localhost_jqdemo' ?" and contains two input fields: "Username:" with the text "DBA" entered, and "Password:" with "****" entered. At the bottom of the dialog are four buttons: "< Back", "Next >", "Finish", and "Cancel".

Starting the database automatically or using an existing IQ server

If you chose to start your database automatically or you indicated that an existing IQ server should be used to create the database, then the wizard provides a list of existing Connection Profiles. Alternately, you can supply a specific server name.

Note The best way to specify connection information is to use a connection profile. For details, see “Creating connection profiles,” Chapter 3, “Sybase IQ Connections,” in *System Administration Guide: Volume 1*.

If there is a Connection Profile that you want to use, choose a profile from the drop-down. If there is no appropriate connection profile available, identify the server that you want to use in the Server Name field.

No matter what server creates the database, you need to specify the parameters in Table 4-1.

Table 4-1: Create Database connection parameters

Parameter name	Description
Host Machine	Name of the host system where the new database is created.
Port Number	Port number on which the server is listening.
Agent Port	Port number where the IQ Agent on the given host listens. The default Agent port number is 1099. If you do not know the port, leave it blank: when you move to the next page, the wizard attempts to find the IQ Agent for you.
Username	Name of any user with DBA authority. The Username defaults to DBA. This identifies a user with full administration and resource creation rights.
Password	Password for the given user name. The password for this user defaults to <code>sql</code> .

Note By default, all newly-created Sybase IQ databases use DBA and `sql` as user ID and password combination. For security reasons, Sybase strongly recommends that you change the DBA password from the default. Changing the DBA password requires DBA authority.

Manual startup with Utility Server

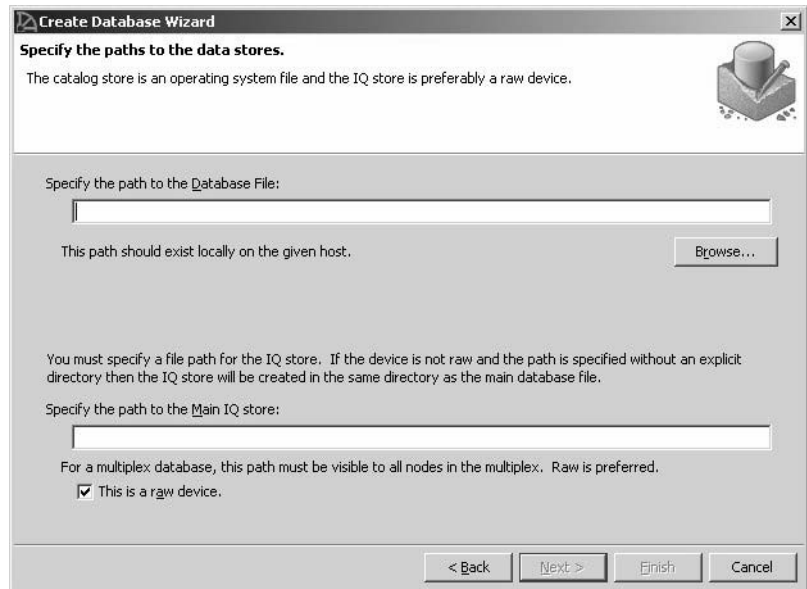
If you chose not to start the database automatically, type the password for the DBA user and check that the host machine and port number are correct, then click Next.

Manual startup with active server

If you start the database manually, and use an active server instead of the Utility Server, type the server name and port number for the active IQ server. If a connection profile exists for the server, you can choose the server from the Connection Profile dropdown menu.

Database and IQ store path information

The next dialog prompts you for the location of the data stores. You must specify the full path to the database file and the main IQ or catalog store. After you define the path(s) to the data stores, click Next to move to the next dialog.



Path to the database file

Type the full path to 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, use the Browse button to choose the path.

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

Path to the IQ store

Type the full path to the IQ store. The IQ store is the main segment file that contains the Sybase IQ data. Make certain that the database path is on a device large enough to accommodate some growth. If you do not provide a path, Sybase 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.

Naming raw devices

Raw device naming conventions differ on UNIX and Windows.

- On UNIX, raw device names vary by operating system. Check your operating system documentation for the correct format.
- On Windows, you can specify the drive letter to name a raw devices. *Do not* include path separator characters (“\\.\”) in the device name when you name a raw device in Sybase Central.

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

Note For more information about naming raw devices, see, Chapter 5, “Working with Database Objects” in *System Administration Guide: Volume 1*.

Setting database page size values

The next dialog prompts you for the database page size values. The page size you select affects the performance of the database. For this tutorial, use the page size recommendations for each category listed below, then click Next.

Create Database Wizard

Specify the Page Size

The page size you select will affect the performance of the database.

The store size determines the size of the system file used to hold the IQ table and index data.

IQ Store Size Megabytes

Reserve space allows the DBSpace to grow in the future. It is optional and cannot be changed once set.

Reserve size in MB: Megabytes

The IQ page size determines the default I/O transfer block size and maximum data compression for the IQ store. 128 KB is the default IQ page size.

IQ Page Size Kilobytes

The database page size determines the size of each page in the catalog store. The default value is 4 KB

Database Page Size Kilobytes

< Back Next > Finish Cancel

IQ store Size 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.

When the IQ store is a file, the default store size is 1000MB, but for this tutorial, use 10MB.

Reserve space For this tutorial, leave the Reserve size blank. Reserve space allows the size of the dbfile to be increased later using the ALTER DBSPACE... ALTER FILE ADD|SIZE command. Generally, 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.

IQ page size 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.

See the *System Administration Guide: Volume 1* and *System Administration Guide: Volume 2* for more information. For this example, click Next to accept the default IQ page size (128KB).

Table 4-2: Default and minimum sizes of Sybase IQ and temporary stores

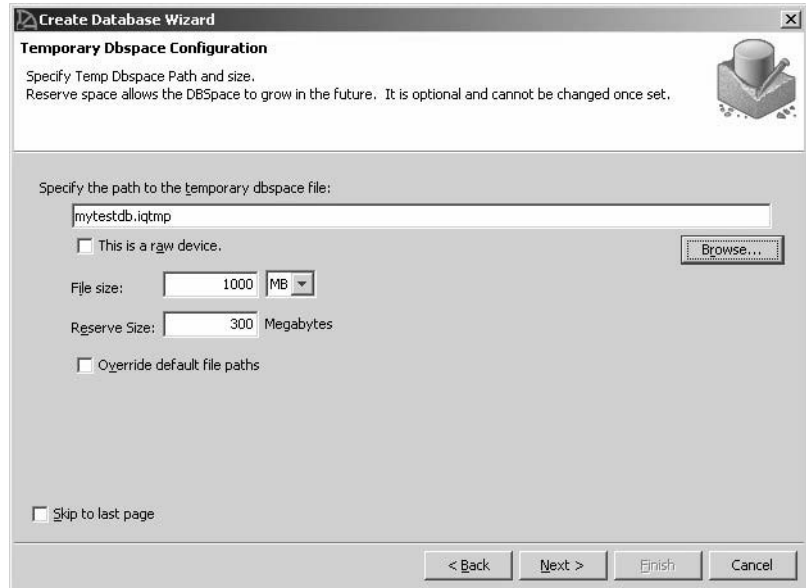
IQ page size	Default size of IQ store	Default size of temporary store	Minimum IQ store size when specified explicitly	Minimum temporary store size when specified explicitly
65536	4096000	2048000	4MB	2MB
131072	8192000	4096000	8MB	4MB
262144	16384000	8192000	16MB	8MB
524288	32768000	16384000	32MB	16MB

Database page size 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.

Creating the temporary store

The temporary store (dbspace) is a set of temporary files that contain the temporary tables and temporary data objects generated by some queries. Depending on your queries, this file may be very large.

For this tutorial follow the recommendations for each category listed below, then click Next to advance to the next screen.



dbspace file path

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 temp dbfiles later. If you do not specify a path, it defaults to *dbname.iqtmp* in the same directory as your database file.

Raw device

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

File size

If the temporary dbfile is not a raw device, type a value of at least 100MB in the File size box. (The default size is a function of the IQ Size.).

Reserve size

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 larger, since a disk array can have a single partition that uses many disks.

Default file paths

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.

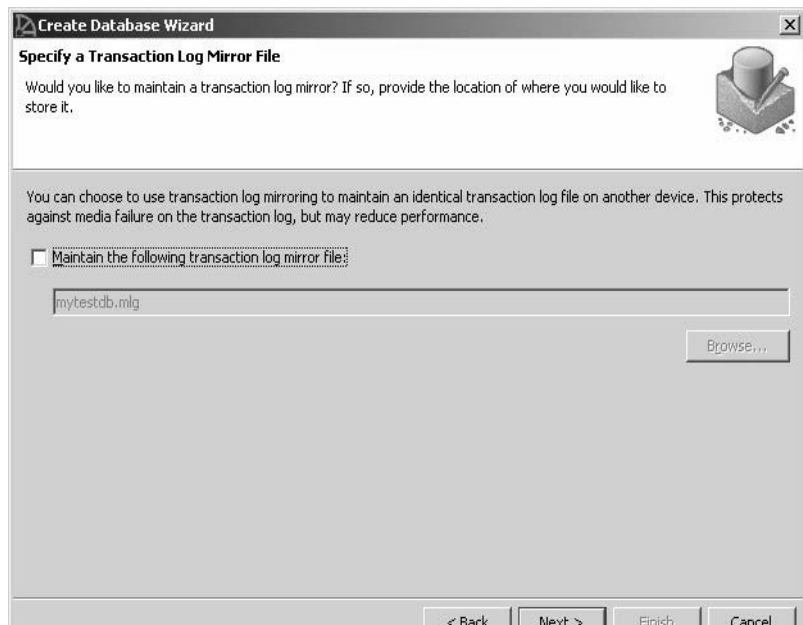
Skip to last page?

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

Mirroring the transaction log

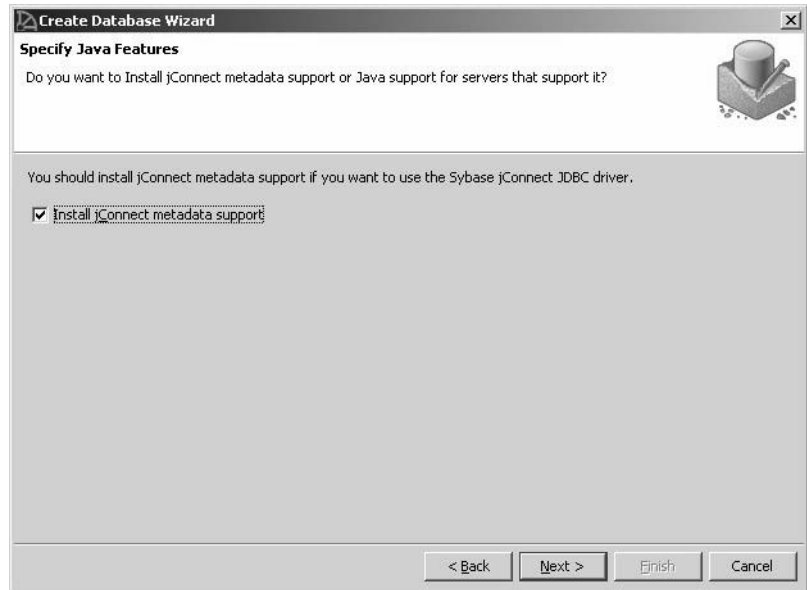
You can create a mirrored transaction log for extra security. A mirrored transaction log is a copy of the default transaction log stored on another device that protects against media failure.

Sybase recommends that you create the log on a different disk from the one where the database resides. For this tutorial, leave the checkbox blank, and click Next.



Specify Java features

The next dialog prompts you to decide whether you want to install metadata that supports the jConnect JDBC driver. jConnect is a high-performance JDBC driver that provides optimal connectivity to the complete family of Sybase products.

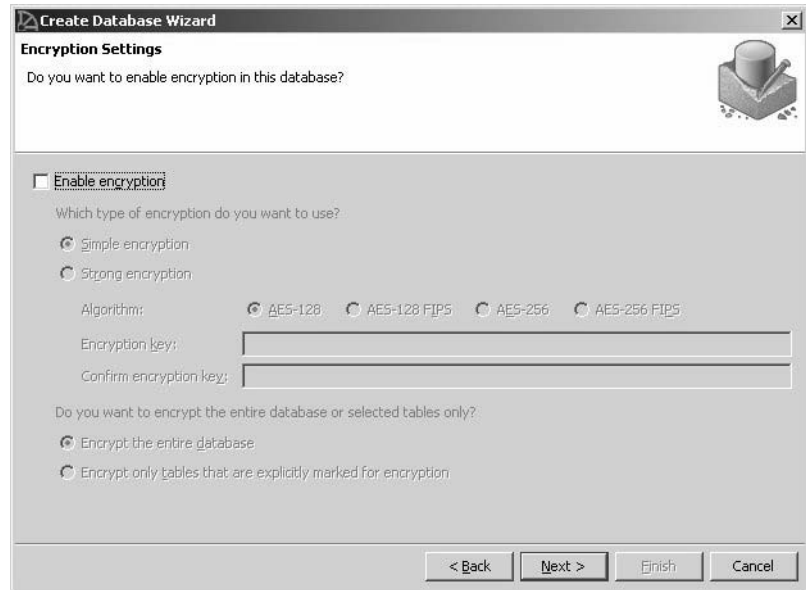


The jConnect driver is installed as part of Sybase IQ. If you do not want to install the driver, clear the check box before you click Next to advance to the next dialog.

Choosing encryption options

The next dialog lets you enable encryption. If you decide to encrypt your database, you must decide on the kind of encryption you want to use, create an encryption key, and decide whether you want to encrypt the entire database or only those tables you specifically mark for encryption.

After you determine the encryption options, click Next.



Enabling encryption

Click the Enable encryption check box to enable encryption.

Simple or strong encryption

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.

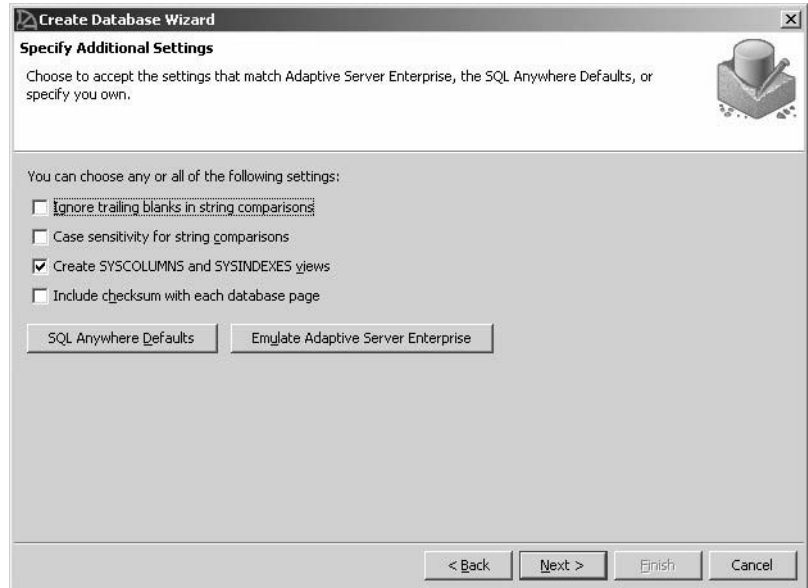
If you choose Strong encryption, you can either:

- Select an algorithm—AES-128, AES-256, or AES-256 FIPS.
- Enter an encryption key and then enter it again to confirm. If the two key fields do not match exactly, an error window displays when you press the Next button.

Whichever encryption option you choose, you must select either Encrypt the Entire Database or Encrypt only Tables Explicitly Marked for Encryption.

Choosing additional settings

Sybase IQ includes some default settings that match Adaptive Server Enterprise and SQL Anywhere. You can use the options on this screen to change those defaults. After you specify the settings you want to use, click Next.



Ignore trailing blanks
in string comparisons

Choosing this option causes Sybase IQ to ignore trailing spaces in string comparisons.

Case Sensitivity for
string comparisons

Choosing this option causes all values to be considered case sensitive in comparisons and string operations. Case Sensitivity defaults to ON for Sybase IQ databases. (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” works).

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

Create
SYSCOLUMNS and
SYSINDEXES views

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.

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.

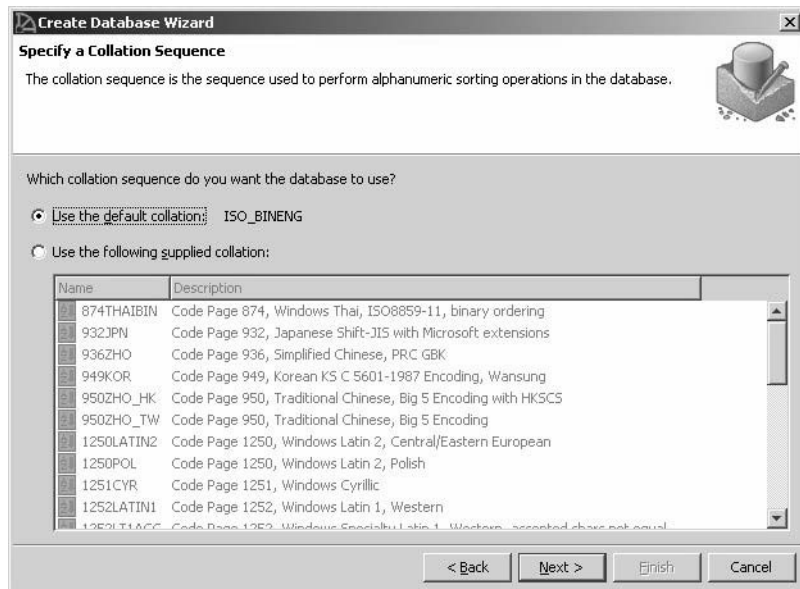
Include checksum

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.

Selecting a collating sequence

This dialog lets you choose a collation sequence. A collation sequence is a combination of a character set and sort order for characters in the database. You can use the default collation sequence or choose one from the list of collation sequences that Sybase IQ supports.

After you specify the collation sequence, click Next to review the settings you chose for your database.



Default collation sequence

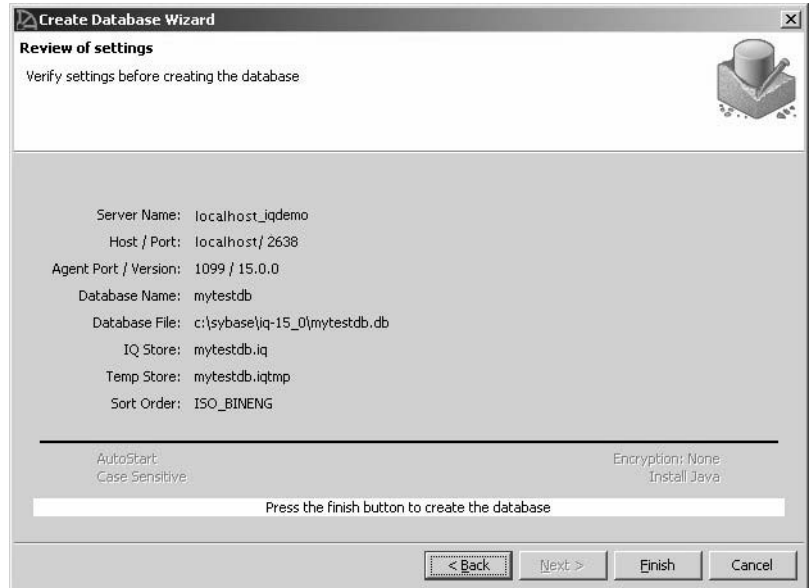
The default collation sequence is ISO_BINENG. ISO_BINENG provides optimal performance for Sybase IQ databases, but not necessarily the most natural sort order. For more information about collation sequences, see the *System Administration Guide: Volume 1*.

Alternative collation sequence

If you want to use a different collation sequence, click **Use the following supplied collation**, then choose an item from the list.

Reviewing your selections

The summary screen lets you review the setting you chose before you create the database. Click Back to return to any screen and change attributes. After you verify the settings, click Finish to create the database.



Server and database properties appear above the line; database attributes appear below the line. 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 “Choosing an IQ server” on page 46 for the automatic startup option.)

Note

- For details about multiplex capability, see *Using Sybase IQ Multiplex*.
 - For additional information about setting up data sources and creating database objects, see Chapter 5, “Working with Database Objects” in the *System Administration Guide: Volume 1*.
 - After 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 *Reference: Building Blocks, Tables, and Procedures*.
-

❖ **Viewing database properties**

- Do one of the following:
 - In Task view, choose a database from the list of Connected Databases, click File, and choose Properties.
 - In Folders view, right-click the database, and choose Properties.

❖ **Viewing database options**

- Do one of the following:
 - In Task view, choose a database from the list of Connected Databases, click File, choose Options.
 - In Folders view, right-click the database, and choose Options.

Note Database options control database behavior. For a complete list of the database options, see the *Reference: Statements and Options*.

Deleting databases

You can use the DROP command in dbisql to remove the database files (temporary files, Sybase 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,” Chapter 3, “Sybase IQ Connections,” in *System Administration Guide: Volume 1*. 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 *Reference: Statements and Options*.

Managing tables

To add a table to your database, use the appropriate wizard to define the properties of your table:

Table 4-3: Table creation wizards

For...	Use this wizard...
Persistent tables.	Table Creation wizard
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.	Global Temporary Table Creation
Tables on remote servers.	Proxy Table Creation wizard

Creating a table and adding columns with Sybase Central

In this tutorial, you create an *office* table in *iqdemo* that describes different offices for a fictitious company.

❖ **Creating an office table**

- 1 Do one of the following:
 - In Task view, choose a database from the list of Connected Database, then choose Create a table from the Table Design Tasks.
 - In Folder view, right-click the Tables folder, point to New, and choose Table.
- 2 Name the new table office, and click Next.
- 3 Click IQ_SYSTEM_MAIN as the dbspace you want to use to store the new table, and click Next.
- 4 In the Comment field, enter: *Company offices*.
- 5 Click Finish.

The Table wizard creates the office table, opens the columns tab in the right pane, and prompts you to define the columns. Follow the steps below to add columns to the new office table.

❖ **Creating columns for the office table**

On the Columns tab, each row represents a single column, and contains all of the attributes for that column.

- 1 Use the values in the following table to create the first column. Leave the PKey box checked.

Pkey	Name	Data type
✓	office_id	integer

Leave the ID, Size, Scale, Nulls, Unique and Comments attributes blank

- 2 To create a new column:
 - Tab to the end of the current row.
 - Click the New Column button on the Toolbar.
 - Click File, point to New, and choose Column.
- 3 Use these attributes to create columns for the office table.

Name	Data type	Size	Nulls
region	char	7	✓
street	char	40	✓
city	char	20	✓
state	char	2	✓

Name	Data type	Size	Nulls
zip	char	9	✓

- 4 To save the column definitions:
 - Click the Save Table button on the Toolbar.
 - Click File, and choose Save.

The office table is now visible on the tables tab in the right pane of Sybase Central. The office table is now part of the iqdemo database. To add data, click the Data tab, right-click, and choose Add Row, or you can use the Interactive SQL utility. 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 Do one of the following:
 - In Tasks view, choose Work with another table in this database from the list of Other Tasks. Double-click the office table.
 - In Folders view, click the office table.
- 2 If necessary, click the Columns tab in the right panel.
- 3 Change the name zip column to postal_code. Notice that the row that you changed is marked by an icon that indicates that the column is modified.
- 4 Do one of the following to save the new column definitions;
 - Click the Save Table button on the Toolbar.
 - Click File, and choose Save.

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 demo database, restoring the database to its original state.

❖ **Deleting the office table from the demo database**

- Right-click the office table, and choose Delete.

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 demo database. The following steps explain how to view and edit stored procedures.

❖ Viewing and editing procedures

1 Do one of the following:

- In Tasks view, double-click the Procedures & Functions folder in the Contents tab, then double-click ShowCustomers.
- In Folders view, open the Procedures & Functions folder, and choose ShowCustomers.

Sybase Central displays the procedure in a SQL editor in the right pane. Right-click anywhere in the SQL editor to display standard editing options. You can also use the editing options on the Toolbar.

2 Do one of the following to save your changes:

- Click the Save button on the Toolbar.
- Click File, and choose Save.

❖ Running a stored procedure

• Do one of the following:

- In Tasks view, double-click the Procedures & Functions folder in the Contents tab, double-click the procedure you want to run, then choose Execute from Interactive SQL from the list of Procedure & Function Design Tasks.
- In Folders view, open the Procedures & Functions folder, right-click the procedure you want to run, and choose Execute from Interactive SQL.

❖ Viewing and altering procedure permissions

Stored procedures have permissions associated with them. You need explicit permission as a user or a member of a group that has permission to execute the procedure. To review and change permissions for a procedure:

- 1 Right-click a procedure, choose Properties.
- 2 On the Procedure Properties sheet, click the Permissions tab to see which users have permissions on this procedure.
- 3 Click Grant, choose the user or group you want to grant permission to execute this procedure, click OK.

- 4 Click Apply.

Managing users and groups

In Sybase IQ, users and groups are database objects. This tutorial demonstrates how to establish a login policy for a separate class of users, create a group and user.

Note For additional information about login policies, see *Managing users and groups* in Sybase IQ online help.

Creating a login policy

Users and groups are also governed by appropriate login policies. A login policy is a set of rules that are applied when you create a database connection for a user or group. All new databases include a root login policy, which may not be appropriate for all users.

You can use the Login Policies wizard to define rules that apply when a user connects to a database. This section shows you how to create a login policy for access by members of a Sales group.

❖ Adding a new login policy

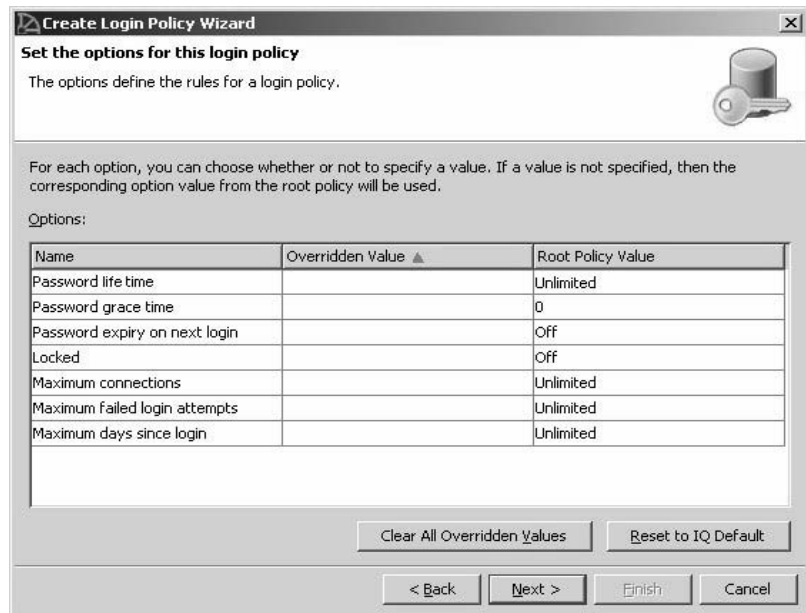
- 1 Do one of the following:
 - In Tasks view, double-click Login Policies, choose Create a login policy from the list of Login Policy Design Tasks.
 - In Folders view, right-click Login Policies, point to New, and choose Login Policy.

The login policy wizard starts.



- 2 Name the new login policy, Sales access, then click Next.

The policy options dialog appears.



- 3 Set the appropriate policy options, then click Next to display the comments dialog.



- 4 On the comments dialog, type an appropriate comment, then click Finish.

Adding a group to the database

In this section, you add a sales group to the demo database.

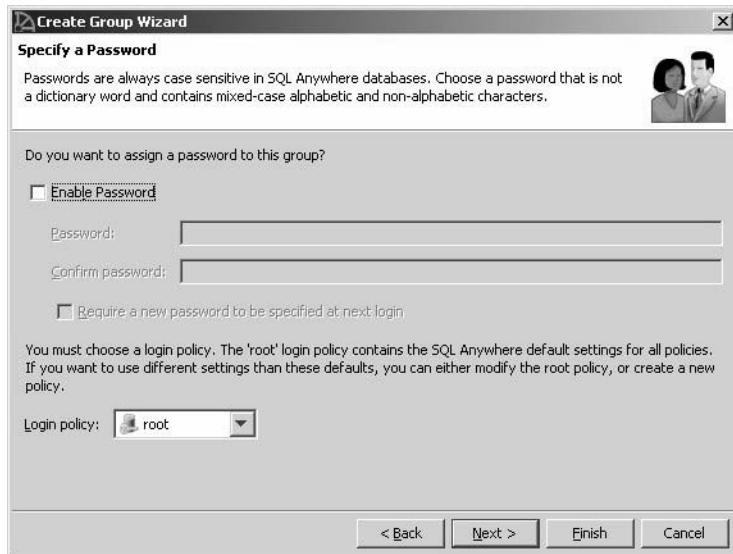
❖ Adding a group to a database

- 1 Do one of the following:
 - In Tasks view, double-click Users & Groups, choose Create a group from the list of User & Group Design Tasks.
 - In Folders view, right-click Users & Groups, point to New, and choose Group.

The new group wizard starts.



2 Name the new group Sales, then click Next.

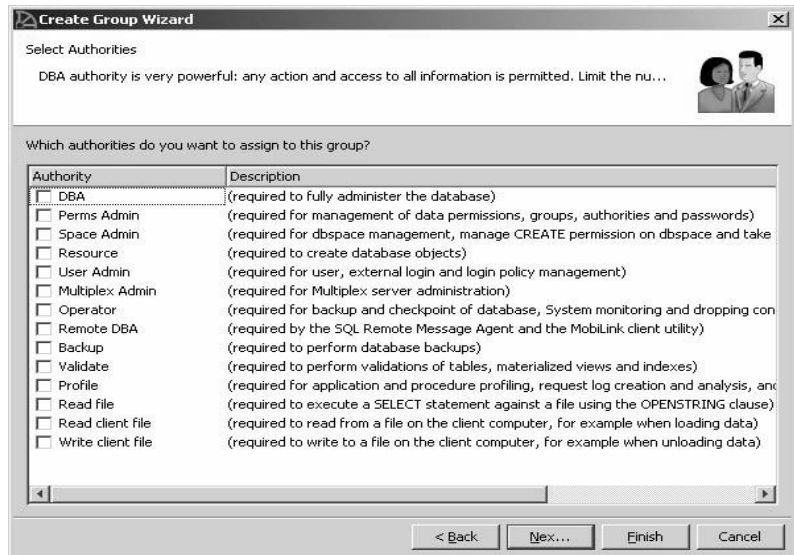


3 Click the Enable Password box.

4 In the Password box, type an appropriate password, then re-type the password in the Confirm password box, click Next.

If you click the Require a new password to be specified at next login box, Sybase Central prompts first group member to login to choose a new password at the next login.

- 5 Choose Sales access as the group login policy, click Next.



- 6 Choose the authorities you want to assign to this group, click Next.



- 7 Type an appropriate comment in the comment box and click Finish.

Adding a user to the database

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

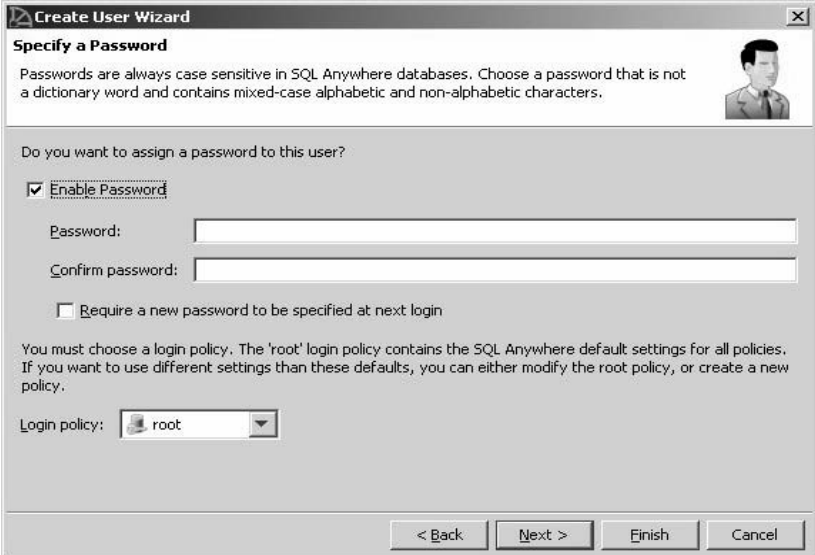
❖ Adding a user to the demo database

- 1 Do one of the following:
 - In Tasks view, double-click Users & Groups, choose Create a user from the list of User & Group Design Tasks.
 - In Folders view, right-click Users & Groups, point to New, choose User.

The new user wizard starts.



- 2 Name the new user, “Sales person” then click Next.

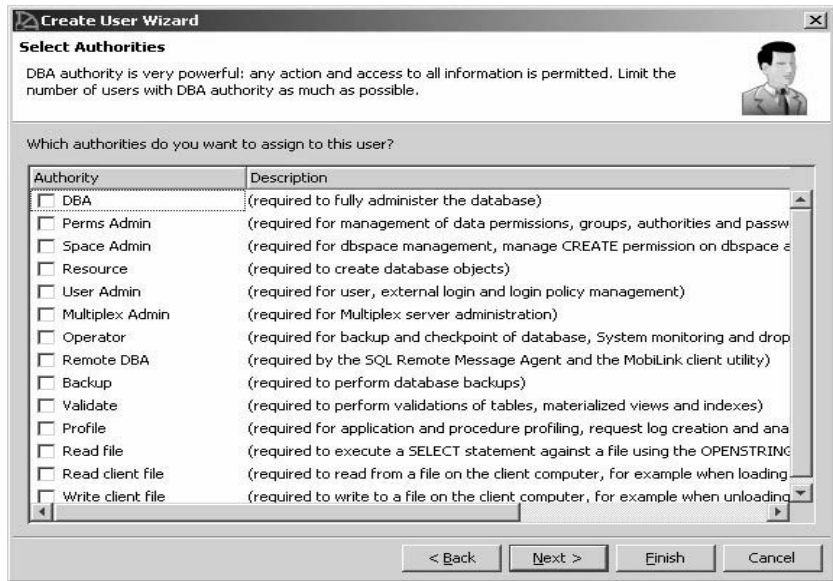


The screenshot shows a window titled "Create User Wizard" with a close button in the top right corner. The main heading is "Specify a Password". Below the heading is a note: "Passwords are always case sensitive in SQL Anywhere databases. Choose a password that is not a dictionary word and contains mixed-case alphabetic and non-alphabetic characters." To the right of this note is a small icon of a man in a suit. Below the note is the question "Do you want to assign a password to this user?". There are two checkboxes: "Enable Password" (checked) and "Require a new password to be specified at next login" (unchecked). Below these are two text input fields: "Password:" and "Confirm password:". Below the input fields is a "Login policy:" label and a dropdown menu showing "root". At the bottom of the window are four buttons: "< Back", "Next >", "Finish", and "Cancel".

- 3 Click the Enable Password box.
- 4 In the Password box, type an appropriate password, then re-type the password in the Confirm password box, click Next.

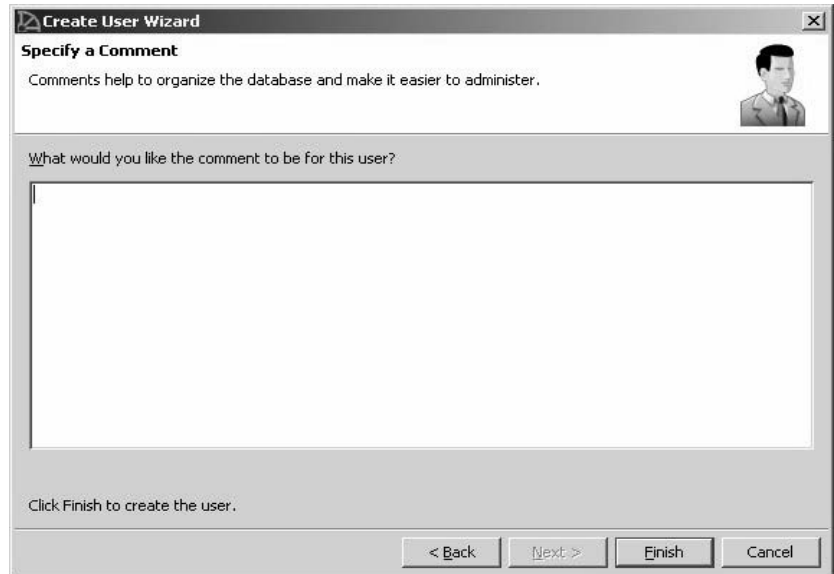
If you click the Require a new password to be specified at next login box, Sybase Central prompts the user to choose a new password the first time that user logs in.

5 Choose Sales access as the login policy, click Next.



6 Choose the authorities you want to assign, click Next.

See “Database permissions and authorities overview” in SQL Anywhere documentation in *SQL Anywhere 11.0.1 > SQL Anywhere Server - Database Administration > Configuring Your Database > Managing user IDs, authorities, and permissions*.



7 Type an appropriate comment in the comment box and click Finish.

❖ Restoring the demo database to its original state

To restore the database to its original state, you can delete the Sales group and the Sales person user.

- 1 Do one of the following:
 - In Tasks view, double-click Users & Group on the Contents tab.
 - In Folders view, click Users & Groups.
- 2 In the right-pane, click the user or group you want to delete, choose Delete.

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 demo database and view them using Sybase Central.

❖ **Creating and viewing connections**

- In the right pane, click the Connected Users tab.

The Connected Users tab lists all connected users. Each user is identified by a Connection ID assigned by the server.

❖ **Disconnecting users from a database**

- 1 In the right pane, click the Connected Users tab.
- 2 Right-click the user's Conn ID, choose Disconnect.

❖ **Disconnecting your current connection**

- Do one of the following:
 - Click the Disconnect button on the Toolbar.
 - Click Connections, choose Disconnect.

Indexing and Loading Data

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.

Topic	Page
Creating column indexes	78
Creating join indexes	81
Loading data	83

Creating column indexes

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

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, VARCHAR, or LONG VARCHAR column as a delimited list.

These index types are unique to Sybase IQ data and cannot be applied to SQL Anywhere tables. For more information, see the *System Administration Guide: Volume 1*.

Note 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, Sybase 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 *System Administration Guide: Volume 1* 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 and storage locations 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 *iqdemo* database, if not connected.
- 2 Do one of the following:
 - In Folder view, right-click Indexes, point to New, choose Index on IQ Table...
 - In Tasks view, double-click the Indexes folder, and choose Create an index from the Index Design tasks.
- 3 Select the *Customers* table from the list, name the new index `IQ_IDX_LF_ID`, then click Next.
- 4 Choose the Low Fast option as the Index type, then click Next.

The default number of records to add before notification is sufficient for this tutorial. For details about the notify count, see the *System Administration Guide: Volume 1*.
- 5 On the dbspace dialog, choose an appropriate dbspace to store the index. For this tutorial, choose *iq_main*, then click Next.
- 6 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 *Reference: Statements and Options*.

Creating join indexes

Join indexes are Sybase IQ internal structures that optimize joins of related tables.

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, see the *System Administration Guide: Volume 1* and “Joining Tables” in the *Performance and Tuning Guide*.

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 demo database” on page 8 to see which columns are in each table in this example.

Note Some additional tasks are required for multiplex databases. Please see *Using Sybase IQ Multiplex* instead of this chapter if you need to create join indexes in a multiplex.

The first join index, *so_soi_idx*, joins the *sales_order* and *sales_order_items* tables.

❖ Creating a join index for SalesOrders and SalesOrderItems

- 1 In Folders view, right-click Join Indexes, point to New, and choose Join Index.
- 2 In the Name and Table dialog, enter *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 *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 SalesOrders for the left table.
- 5 Choose SalesOrderItems 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 SalesOrders and SalesOrderItems`.
- 8 Click Finish. The `so_soi_jdx` join index appears in the Join Indexes folder.

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

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

- 1 In Folders view, right-click Join Indexes, point to New, choose Join Index.
- 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 to join equivalent columns with the same name (`dept_id`) from two tables.
- 4 Choose Departments for the left table.
- 5 Choose Employees for the right table.
- 6 Click Next.
- 7 In the Choose Index Columns dialog, select the desired Left Table column, `DepartmentID` and the Right Table column `DepartmentID`.
- 8 Click Add. After you add the two columns, they appear in the Join Index Columns box as
`GROUP0.Departments.DepartmentID=GROUP0.employees.DepartmentID`. (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 Departments and Employees`. 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 *System Administration Guide: Volume 1*.

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 98 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 Sybase 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 *Utility Guide*. If a client application has been written to perform the load, use the appropriate programming interface.

You can use Interactive SQL (dbisql) to insert data manually. Interactive SQL (DBISQL) is an interface available on all supported platforms. See the *Utility Guide* to learn how to run dbisql.

Use these SQL commands to load a table:

- INSERT FROM SELECT loads information from a database 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, 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.

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

Syntax and examples for the INSERT and LOAD commands are in *Reference: Statements and Options*.

To add a row to a table using Sybase Central:

- In Folder view, open the Tables folder, click the table you want to update, then click the Data tab in the right pane. Right-click a row, and choose Add.
- In Tasks view, double-click the Tables folder, choose the table you want to update, then click the Data tab. Right-click a row, and choose Add.

Managing Dbspaces

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.

Topic	Page
Overview of dbspaces	86
Creating dbspaces	86
Altering dbspaces	89

Overview of dbspaces

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

A **store** is one or more dbspaces that store persistent or temporary data for a special purpose. Sybase IQ has three stores:

- The catalog store contains the SYSTEM dbspace and up to twelve additional catalog dbspaces.
- The IQ main store contains the IQ_SYSTEM_MAIN dbspace and other user dbspaces.
- The IQ temporary store contains the IQ_SYSTEM_TEMP dbspace.

For more about dbspaces, see Chapter 5, “Working with Database Objects,” in *System Administration Guide: Volume 1*.

Sybase recommends that, immediately after creating a database, administrators create a second main dbspace (a user main dbspace) and set it to be the default dbspace. To prevent users from placing tables and indexes in IQ_SYSTEM_MAIN, revoke CREATE privilege in IQ_SYSTEM_MAIN from PUBLIC and grant it to the new user main dbspace. 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

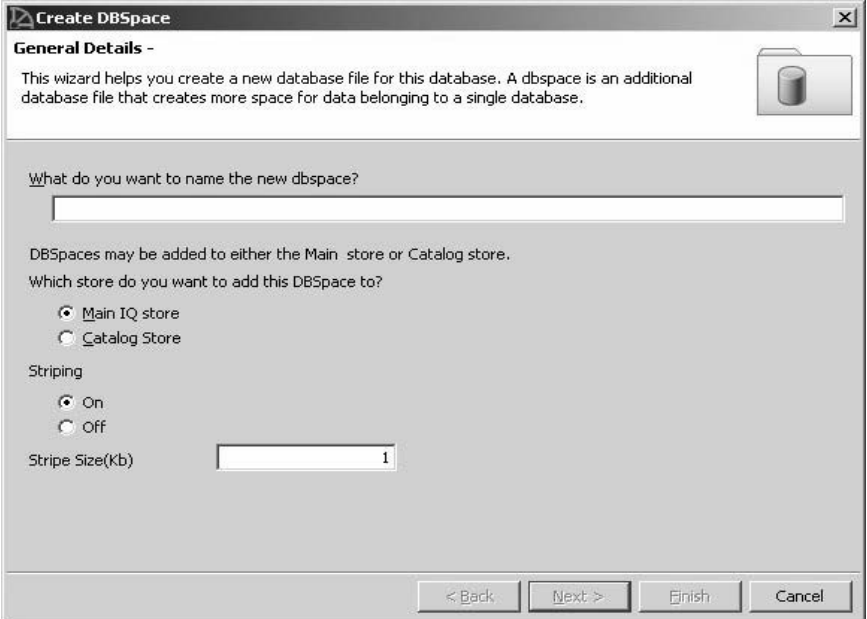
Before you create a dbspace, there must be an IQ agent running on the machine where you want to create the dbspace. To create a dbspace in Sybase Central, connect to a server, then:

- In Tasks view, double-click Dbspaces folder, choose Create a dbspace from the list of Dbspace Design Tasks
- In Folder view, right-click Dbspaces, point to New, and choose Dbspace.

Specifying dbspace name and store

When the wizard starts, the first dialog prompts you for general details about the dbspace you want to create, including the name, storage location, and striping options. For more about the striping options, see ALTER DBSPACE statement in *Reference: Statements and Options*.

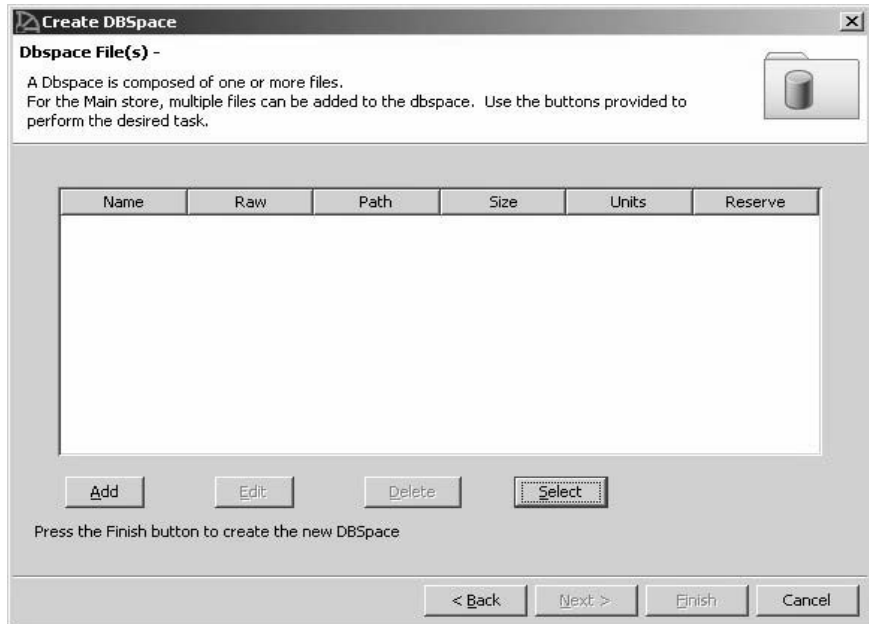
After you provide these details, click Next.



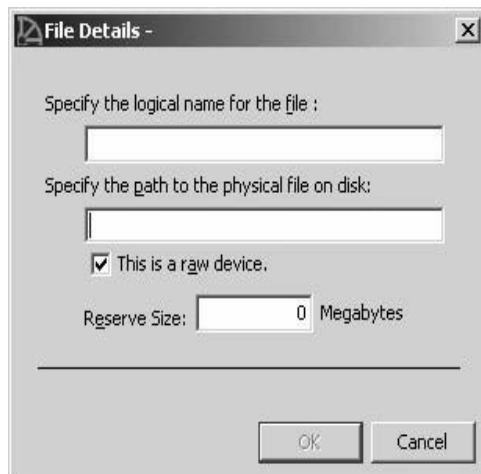
The screenshot shows a Windows-style dialog box titled "Create DBSpace". The "General Details" section contains the following text: "This wizard helps you create a new database file for this database. A dbspace is an additional database file that creates more space for data belonging to a single database." Below this is a text input field for the dbspace name. The next question is "Which store do you want to add this DBSpace to?", with two radio button options: "Main IQ store" (selected) and "Catalog Store". Under the "Striping" section, there are two radio button options: "On" (selected) and "Off". At the bottom of this section is a "Stripe Size(Kb)" input field with the value "1". The dialog box has standard navigation buttons at the bottom: "< Back", "Next >", "Finish", and "Cancel".

Specifying dbspace files

The second dialog lets you add files to your dbspace.



Click Add to open a dialog that prompts you for the file details.



The logical name for the file is the actual file name of the dbspace, with a path where necessary. A file name without an explicit path reference 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 file name suffix — *.iq* for an IQ main store, *.iqtmp* for IQ temporary store.

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

You may also specify an amount of space to reserve in megabytes, 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.

Each dbspace may contain any number of files. The only limit is that the total number of dbfiles is 16383.

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. You must have DBA privileges to change dbspace properties.

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 encounter an error and rollback.

Updating Databases

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.

Topic	Page
Adding rows to a table	92
Canceling changes	93
Making changes permanent	93
Ensuring data consistency	94
Deleting rows	96
Updating rows	97
Synchronizing join indexes	98
Deleting dbspaces, tables, and indexes	99

Adding rows to a table

This chapter show you how to 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 Departments ( DepartmentID, DepartmentName,
DepartmentHeadID )
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 number of insert VALUES does not match the column
list.
```

Note Case sensitivity may matter when inserting values.

Nulls

The NULL value is a special value used to indicate that something is either not known or not applicable. However, the NULL value is a legal value in any data type. There are two separate and distinct cases where NULL is used:

Situation	Description
Missing	The field does have a value, but that value is unknown.
Inapplicable	The field does not apply for this particular row.

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 Employees
```

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 Departments
VALUES ( 220, 'Eastern Sales', 902 )
```

Canceling changes

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 COMMIT statement 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, overhead for each transaction 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 *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.

Note Increased overhead from frequent COMMIT operations affects 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 iqdemo
AS sales
```

- 2 Start a read/write transaction.

```
INSERT INTO FinancialCodes
(Code, Type, Description)
VALUES ('e6', 'expense', 'Services')
```

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

```
CONNECT DATABASE iqdemo
AS marketing;
SELECT CustomerID, OrderDate
FROM SalesOrders
```

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 iqdemo
AS accounting
```

- 5 Start a read/write transaction:

```
INSERT INTO FinancialCodes
(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.

Set some savepoints to return to as you run two similar transactions. This time, you can put the second transaction on hold until the first one completes.

❖ Setting savepoints

- 1 Connect to the database:

```
CONNECT DATABASE iqdemo
```

- 2 Start a read/write transaction.

```
INSERT INTO FinancialCodes
(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 *Reference: Statements and Options*.

- 4 Start a read/write transaction:

```
INSERT INTO FinancialCodes
(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 iqdemo
AS marketing
```

- 2 Start a read/write transaction.

```
INSERT INTO FinancialCodes
(Code, Type, Description)
VALUES ('e9', 'expense', 'R & D')
```

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

```
CONNECT DATABASE iqdemo
AS sales
```

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

```
INSERT INTO SalesOrders
(ID, CustomerID, OrderDate, SalesRepresentative)
VALUES ('2088', '140', '05-29-98', '195')
COMMIT
```

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

For more details about transaction support, see Chapter 10, “Transactions and Versioning” in the *System Administration Guide: Volume 1*.

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 Employees
WHERE EmployeeID = 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 Employees
WHERE TerminationDate 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 Employees
WHERE LEFT(Phone, 3 ) = '617'
AND ManagerID = 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 Customers
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 *Employees* table. To synchronize the Employees/SalesOrders 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).

Sybase Central

- Right-click the object, choose Delete.
- Click the object, click the Delete button on the Toolbar.

DBISQL

- Use the DROP command in DBISQL. For complete syntax, see *Reference: Statements and Options*.

Using views

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

Topic	Page
Using views	102
Defining a view	102
Using views for security	105
Restricting access to information in the database	107

Using views

A view is a computed table defined by the result set of its view definition, which is expressed as a SQL query. You can use views to show database users exactly the information you want to present, in a format you can control.

A **regular view** describes a view that is recomputed each time you reference the view, and the result set is not stored on disk. This is the most commonly used type of view. A **materialized view** describes a view whose result set is precomputed and materialized on disk similar to the contents of a base table. Materialized views are useful in data warehousing scenarios, where frequent queries of the actual base tables can be extremely expensive. Examples in this section refer to regular views.

Note Materialized views are only supported for SQL Anywhere tables in the IQ catalog store.

Defining a view

Suppose you frequently need to list employees by department. The following query creates a list of all employees by department and state that you can run against iqdemo:

```
SELECT Departments.DepartmentID,
       Departments.DepartmentName, Employees.EmployeeID,
       Employees.GivenName, Employees.Surname,
       Employees.Phone
FROM Departments, Employees
ORDER BY Employees.State
```

Running the query creates a results set that looks like this:

DepartmentID	DepartmentName	EmployeeID	GivenName	Surname	Phone
100	R & D	148	Julie	Jordan	6175557835
200	Sales	148	Julie	Jordan	6175557835
300	Finance	148	Julie	Jordan	6175557835
400	Marketing	148	Julie	Jordan	6175557835
---	---	---	---	---	---

Creating a view

In Interactive SQL, you can use a CREATE VIEW statement to create and store a view. This example creates a view called emp_dept in iqdemo:

```
CREATE VIEW emp_dept AS
  SELECT Departments.DepartmentID,
     Departments.DepartmentName, Employees.EmployeeID,
     Employees.GivenName, Employees.Surname,
     Employees.Phone
  FROM Departments, Employees
  ORDER BY Employees.State
```

In Sybase Central, you can use a wizard to create a view. To start the wizard, connect to iqdemo, then do one of the following:

- In Tasks view, double-click the Views folder, then choose Create a view from the list of View Design Tasks.
- In Folders view, right-click Views, point to New, choose View.

When the wizard starts, follow the instructions on the screen.

Displaying view data

You can query a view just like a table:

```
SELECT *
FROM emp_dept
```

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. This means that if the sales data changes, information in the view will be automatically up to date.

To see the results in Sybase Central, right-click the View, choose View Data in Interactive SQL.

Changing a view

In Interactive SQL, you can use an ALTER VIEW statement to change a view definition with a modified version. The emp_dept view lists all employees by department. You can use the following command to change the view to display only those employees in the Sales department:

```
ALTER VIEW emp_dept AS
  SELECT Departments.DepartmentID,
     Departments.DepartmentName, Employees.EmployeeID,
     Employees.GivenName, Employees.Surname,
     Employees.Street, Employees.City,
     Employees.State, Employees.Phone
  FROM Departments, Employees
  WHERE Departments.DepartmentID = 200
  ORDER BY Employees.State
```

ALTER VIEW replaces the existing SELECT statement in the view with the SELECT you defined in the ALTER VIEW command. Existing permissions on the view are maintained.

To change a view in Sybase Central, right-click the View, and choose Edit in New Window.

Deleting views

To delete a view from the database in Interactive SQL, use the DROP statement:

```
DROP VIEW <view name>
```

To drop a view in Sybase Central, right-click the view you want to delete, and choose Delete from the shortcut menu.

View restrictions and advanced uses

Views can be thought of as virtual 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:

```
CREATE VIEW Q1_Revenue AS
SELECT Products.Name AS Item, Products.Description AS
  Style, SUM(SalesOrderItems.Quantity) AS Sales,
  Products.UnitPrice, SUM(SalesOrderItems.Quantity *
  Products.UnitPrice) AS Revenue FROM Employees
JOIN SalesOrders ON SalesOrders.SalesRepresentative =
  Employees.EmployeeID
JOIN SalesOrderItems ON SalesOrderItems.ID =
  SalesOrders.ID
JOIN Products ON Products.ID =
  SalesOrderItems.ProductID
WHERE SalesOrders.OrderDate >='2001-01-01'
AND SalesOrders.OrderDate <='2001-04-30'
GROUP BY Products.Description, Products.Name,
  Products.UnitPrice
```

Running the query creates a results set that looks like this:

Item	Style	Sales	Unit Price	Revenue
Tee Shirt	Tank Top	744	9.00	6696.00
Sweatshirt	Hooded Sweatshirt	756	24.00	18144.00
Visor	Cloth Visor	756	7.00	5292.00
Baseball Cap	Wool cap	685	10.00	6850.00
Tee Shirt	Crew Neck	672	14.00	9408.00
Shorts	Cotton Shorts	1524	15.00	22860.00
Baseball Cap	Cotton Cap	900	9.00	8100.00
Tee Shirt	V-neck	780	14.00	10920.00
Sweatshirt	Zippered Sweatshirt	564	24.00	13536.00
Visor	Plastic Visor	684	7.00	4788.00

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, connect to the demo 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 EmployeeID, Surname, GivenName
FROM Employees
WHERE DepartmentID=200
```

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 Employees;
SELECT * FROM Departments
```

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.

Restricting access to information in the database

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 *Reference: Statements and Options*.

Index

A

- Adaptive Server Enterprise 14
- adding users to databases 72
- Agent
 - starting 31

B

- base tables 26

C

- case sensitivity
 - string comparisons 57
- catalog
 - system 28
- Catalog Store 14
- checksums 58
- collation sequence 58
- column indexes 78
- columns
 - editing properties 63
- COMMIT statement
 - about 93
- compare indexes 79
- Component Integration Services 2
- CONNECT statement
 - example 105
- connection
 - parameters 48
 - profile 14
- connectivity sample applications 8
- containers 33
 - expanding 38
 - selecting 37
- CREATE VIEW statement
 - about 102

- creating databases 46

D

- data
 - case sensitivity 57
 - loading 83
 - tabular 22
- data dictionary 28
- data marts 4
- data types
 - about 23
- data warehouses 3
- database
 - demo 8
 - sample 8
- databases
 - Adaptive Server Enterprise 14
 - adding users 72
 - administering 30
 - Anywhere 14
 - connecting to 41
 - creating 46
 - creating with Sybase Central 46
 - deleting 61
 - files 21
 - IQ 15
 - managing 30
 - proxy 18
 - relational 23
 - reserve space 52, 53
 - selecting collation 58
 - specifying IQ Page Size 52
- dbfiles 15
- dbisql
 - command line parameters 41
 - See Interactive SQL
- dbisqlc 83
- dbspaces 15

Index

- deleting 99
- growth 52, 53
- viewing with Sybase Central 39

decision support systems 3

deleting

- databases 61
- objects 99
- tables 64

demo database 8

- connecting to 41
- iq_dummy table 8
- starting 8

disconnecting users from databases 76

documentation 7

- SQL Anywhere viii
- Sybase IQ vii

dummy IQ table 8

E

editing

- column properties 63
- tables 63

enterprise transaction environment 2

equijoins 82

F

fast launcher option 32

fixed length format 84

foreign keys

- about 25
- defined 24

free lists 15

functions

- viewing with Sybase Central 40

G

GRANT statement

- example 105

groups

- adding 69

H

High_Group indexes 79

High_Non_Group indexes 79

I

indexes

- creating 78
- definition 26, 40
- deleting 99
- join 17
- types 78

INSERT statement

- examples 92

inserting

- binary data 84
- data from backups 84

Interactive SQL 7

Interactive SQL Classic 83

IQ Message Log 16

IQ Page Size

- default 52
- specifying 52

IQ Store 15

iq_dummy table 8

iqdemo.db file

- about 8

ISO_BINENG collation 58

J

java sample applications 8

join indexes 17

- creating 81
- synchronizing 19

joins, defined 27

K

keys

- foreign 24
- partition 18
- primary 24

L

language interface 20
 loading data 83
 from databases 83
 from files 84
 inserting values 84
 logs
 message 21
 transaction 22
 Low_Fast indexes 79

M

make files 8
 manuals 7
 Message Log
 IQ 21
 message log 16
 metadata 17
 mobile computing 2
 multiplex 17

N

network server 6
 network server executable 6
 New Group toolbar button 69
 NULL value
 allowed in columns 92

O

objects 17
 ODBC sample applications 8
 OLTP
 See On-Line Transaction Processing
 On-Line Transaction Processing 3

P

partition keys 18

partitions
 range 18
 table 19
 passwords
 and permissions 105
 performance 22
 permissions
 setting 65
 plug-ins 33
 connecting 33
 registering 33
 primary keys
 defined 24
 procedures
 setting permissions 65
 viewing 65
 viewing with Sybase Central 40
 program group 7
 programming interfaces 83
 projections 27
 proxy database 18

R

range partitions 18
 Raw Device Access 8
 raw partitions
 for temporary file pathname 53
 readme file 8
 relational database concepts 23
 reserve space 52, 53
 ROLLBACK statement
 examples 97
 rows
 updating 97

S

sample applications 8
 servers
 definition 30
 shutting down 44
 starting 35
 stopping 44

Index

- Service Manager 8
 - SQL 22
 - SQL Anywhere 2, 18
 - database 14
 - start_iq* 6
 - start_iq.exe* 6
 - starting multiplexes 35
 - stopping servers 44
 - Store
 - Catalog 14
 - IQ 15
 - Temporary 21
 - stored procedures
 - definition 26
 - stores
 - definition 18
 - strings
 - case sensitivity 57
 - Structured Query Language
 - See* SQL
 - Sybase Central 6
 - description 30
 - fast launcher option 32
 - in program group 8
 - main window 33
 - starting 31, 32
 - stopping 34
 - Sybase IQ
 - benefits 4
 - components 6
 - definition 2
 - documentation 7
 - network server 6
 - sample database 8
 - Service Manager 8
 - utilities 31
 - SYSOPTIONDEFAULTS table 10
 - system tables 28
- T**
- tables
 - characteristics 24
 - creating 62
 - deleting 64, 99
 - editing 63
 - iq_dummy* 8
 - logical partitions 18
 - partitions 19
 - persistent 17
 - temporary 17
 - viewing with Sybase Central 38
 - tablespaces 19
 - tabular data 22
 - temporary file
 - specifying 53
 - temporary segment file 53
 - Temporary Store 21
 - transaction log 22
 - triggers
 - viewing 65
 - tuples 24
 - tutorials
 - Sybase Central 30
- U**
- updating rows 97
 - user manuals 7
 - users
 - adding to databases 72
 - viewing with Sybase Central 40
 - utilities 31
 - database administration 31
 - utility database
 - database name 61
 - starting with 61
- V**
- variable length format 84
 - views
 - definition 26
 - permissions 105
- W**
- WD indexes 79

WHERE clause 96, 97

