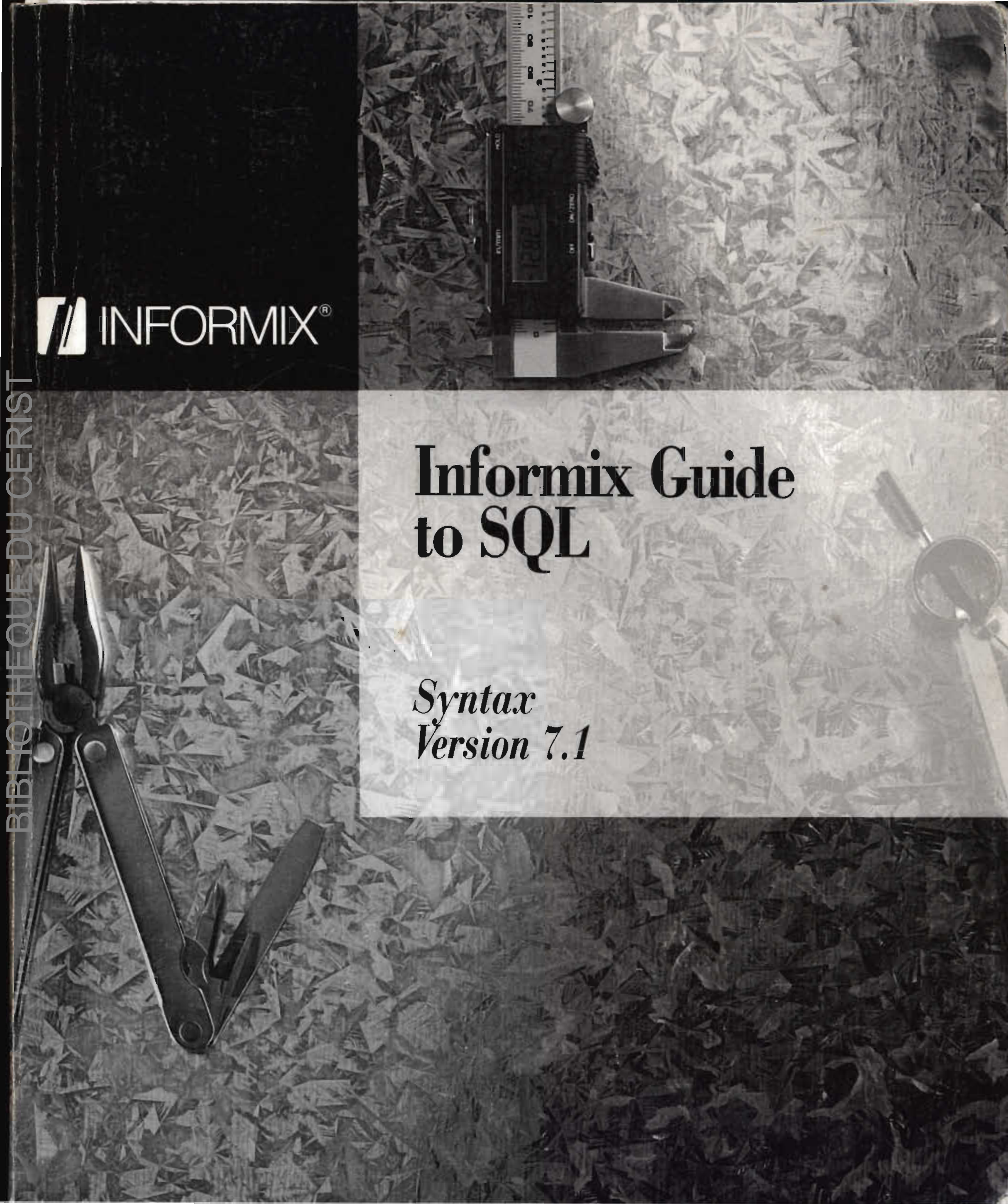


INFORMIX®

Informix Guide to SQL

Syntax
Version 7.1

Published by INFORMIX® Press

Informix Software, Inc.
4100 Bohannon Drive
Menlo Park, CA 94025

Documentation Team: Sally Cox, Geeta Karmarkar, Steve Klitzing, Mary Kraemer, Catherine Lyman, Tom Noronha, Eileen Wollam

Preface

The *Informix Guide to SQL: Syntax* is intended to be used as a companion volume to the *Informix Guide to SQL: Tutorial* and the *Informix Guide to SQL: Reference.*

This manual contains all the syntax descriptions for Structured Query Language (SQL) and Stored Procedure Language (SPL) statements. The *Informix Guide to SQL: Tutorial* explains the philosophy and concepts behind relational databases, and the *Informix Guide to SQL: Reference* provides reference information for aspects of SQL other than the language statements.

Required Software

You must have the following Informix software to enter and execute SQL and SPL statements:

- An INFORMIX-OnLine Dynamic Server or an INFORMIX-SE database server

  The database server must be installed either on your computer or on another computer to which your computer is connected over a network.
- Either an Informix application development tool, such as INFORMIX-4GL; or an SQL application programming interface (API),

such as INFORMIX-ESQL/C; or the DB-Access database access utility, which is shipped as part of your database server.

The application development tool, SQL API, or DB-Access enables you to compose queries, send them to the database server, and view the results that the database server returns. You can use DB-Access to try out many of the SQL statements described in this guide. See the *DB-Access User Manual* for a list of all the SQL statements that you can run from DB-Access.

Summary of Chapters

The *Informix Guide to SQL: Syntax* includes the following chapters:

- The Introduction tells how SQL fits into the Informix family of products and books, explains how to use this book, introduces the demonstration database from which the product examples in this manual are drawn, and lists the new features for Version 7.1 of Informix database server products.
- Chapter 1, "SQL Statements," explains the workings of all the SQL statements supported by Informix products. Detailed syntax diagrams walk you through every clause of each SQL statement, and syntax tables explain the input parameters for each clause. Thorough usage instructions, pertinent examples, and references to related material complete the description of each SQL statement.

  Following the SQL statements in Chapter 1 are descriptions of SQL segments. SQL segments are language elements, such as table names and expressions, that occur in many SQL statements. Instead of describing each segment in each statement where it occurs, this manual provides a comprehensive stand-alone description of each segment. Whenever a segment occurs within a given syntax diagram, the diagram points to the stand-alone description of the segment for further information.
- Chapter 2, "SPL Statements," presents all the detailed syntax diagrams and explanations for SPL statements. You can use stored procedures to perform any function you can perform in SQL as well as to expand what you can accomplish with SQL alone. You write a stored procedure using SPL and SQL statements. For task-oriented

information about using stored procedures, see the *Informix Guide to SQL: Tutorial.*

Informix Welcomes Your Comments

A reader-response card is provided with this manual. Please use this card to tell us what you like or dislike about this manual. To help us with future versions of the manual, please tell us about any corrections or clarifications that you would find useful. Return this card to:

Informix Software, Inc.
Technical Publications Department
4100 Bohannon Drive
Menlo Park, CA 94025

If you prefer to share your comments on-line, address your e-mail to:

doc@informix.com

Related Reading

Like the other books in the SQL manual series, this book is written for people who already know how to use computers and who rely on them in their daily work.

If you want additional technical information on database management, consult the following books. The first book is an introductory text for readers who are new to database management, and the second book is a more complex technical work for SQL programmers and database administrators.

- *Database: A Primer* by C. J. Date (Addison-Wesley Publishing, 1983)
- *An Introduction to Database Systems* by C. J. Date (Addison-Wesley Publishing, 1994).

If you are interested in learning more about SQL, consider the following books:

- *A Guide to the SQL Standard* by C.J. Date with H. Darwen (Addison-Wesley Publishing, 1993)
- *Understanding the New SQL: A Complete Guide* by J. Melton and A. Simon (Morgan Kaufmann Publishers, 1993)
- *Using SQL* by J. Groff and P. Weinberg (Osborne McGraw-Hill, 1990)

This manual assumes that you are familiar with your computer operating system. If you have limited UNIX system experience, you might want to look at your operating system manual or a good introductory text before you read this manual. Some texts about UNIX systems are suggested in the following list:

- *Introducing the UNIX System* by H. McGilton and R. Morgan (McGraw-Hill Book Company, 1983)
- *Learning the UNIX Operating System*, by G. Todino, J. Strang, and J. Peek (O'Reilly & Associates, 1993)
- *A Practical Guide to the UNIX System*, by M. Sobell (Benjamin/Cummings Publishing, 1989)
- *UNIX for People* by P. Birns, P. Brown, and J. Muster (Prentice-Hall, 1985)
- *UNIX System V: A Practical Guide* by M. Sobell (Benjamin/Cummings Publishing, 1995)

Table of Contents

Introduction

Chapter 1 SQL Statements

Chapter 2 SPL Statements

Index

Introduction

Structured Query Language (SQL) is an English-like language that you can use when creating, managing, and using relational databases. The SQL provided with Informix products is an enhanced version of the industry-standard query language developed by International Business Machines Corporation (IBM).

In addition to SQL, Informix provides the Stored Procedure Language (SPL) with which you can write stored procedures. Stored procedures are programs that are stored as database objects.

Informix Products That Use SQL

Informix produces many application development tools and SQL application programming interfaces (API). Application development tools currently available include products such as INFORMIX-SQL, INFORMIX-4GL, and INFORMIX-NewEra. SQL APIs currently available include INFORMIX-ESQL/C and INFORMIX-ESQL/COBOL.

The INFORMIX-NewEra development environment can access Informix database servers directly through embedded SQL statements as well as through the call-level interface of the CCL/INFORMIX and CCL/ODBC function libraries.

Informix products work with a database server, either INFORMIX-OnLine Dynamic Server, INFORMIX-SE, or an INFORMIX-Gateway product. The DB-Access database access utility is shipped as a part of each database server.

If you are running client applications developed with Version 4.1 and 5.0 application development tools, you use INFORMIX-NET to connect the client to the network.

Products Covered in This Manual

All the information presented in this manual is valid for the following products. Differences in their use of SQL are indicated where appropriate.

- INFORMIX-ESQL/C, Version 7.1
- INFORMIX-ESQL/COBOL, Version 7.1
- INFORMIX-OnLine Dynamic Server, Version 7.1
- INFORMIX-SE, Version 7.1
- INFORMIX-Enterprise Gateway, Version 7.1

Other Useful Documentation

You can refer to the following related Informix documents that complement this manual:

- A companion volume to the Syntax, the *Informix Guide to SQL: Tutorial*, provides a tutorial on SQL as it is implemented by Informix products. It describes the fundamental ideas and terminology that are used when planning, using, and implementing a relational database.
- An additional companion volume to the Syntax, the *Informix Guide to SQL: Reference*, provides reference information on the types of Informix databases you can create, the data types supported in Informix products, system catalog tables associated with the database, environment variables, and the SQL utilities. This guide also provides a detailed description of the **stores7** demonstration database and contains a glossary.
- The *SQL Quick Syntax Guide* contains syntax diagrams for all statements and segments described in this manual.
- You, or whoever installs your Informix products, should refer to the *UNIX Products Installation Guide* for your particular release to ensure that your Informix product is properly set up before you begin to work with it. A matrix depicting possible client/server configurations is included in the *UNIX Products Installation Guide.*

- Depending on the database server you are using, you or your system administrator need either the *INFORMIX-SE Administrator's Guide* or the *INFORMIX-OnLine Dynamic Server Administrator's Guide.*
- The *DB-Access User Manual* describes how to invoke the utility to access, modify, and retrieve information from Informix database servers.
- When errors occur, you can look them up by number and learn their cause and solution in the *Informix Error Messages* manual. If you prefer, you can look up the error messages in the on-line message file described in the section "ASCII and PostScript Error Message Files" later in this Introduction and in the Introduction to the *Informix Error Messages* manual.

How to Use This Manual

This manual assumes that you are using INFORMIX-OnLine Dynamic Server as your database server. Features and behavior specific to INFORMIX-SE are noted throughout the manual.

The following sections describe the conventions used in this manual for typographical format, syntax, and example of code.

Typographical Conventions

Informix product manuals use a standard set of conventions to introduce new terms, illustrate screen displays, describe command syntax, and so forth. The following typographical conventions are used throughout this manual:

italics	New terms, emphasized words, and variables are printed in italics.
boldface	Database names, table names, column names, file names, utilities, and other similar terms are printed in boldface.

`computer` Information that OnLine displays and information that you enter is printed in a computer typeface.

KEYWORD All keywords appear in uppercase letters.

✦ The diamond symbol appears at the beginning and the end of product-specific information.

🔑 This symbol indicates a unique identifier (primary key) for each table.

⚠ This symbol indicates a *warning*. Warnings provide critical information that, if ignored, could cause harm to your database.

➡ This symbol indicates *important* information that you should consider when working with the product.

💡 This symbol indicates a *tip*. It alerts you to useful information that, for instance, might indicate a shortcut or make it easier to navigate in the product or manual.

Additionally, when you are instructed to "enter" or "execute" text, immediately press RETURN after the entry. When you are instructed to "type" the text or "press" a key, no RETURN is required.

Syntax Conventions

Syntax diagrams describe the format of SQL and SPL statements or commands, including alternative forms of a statement, required and optional parts of the statement, and so forth. These diagrams have conventions, which are described and illustrated in this section.

Each diagram begins at the upper left with a keyword and ends with a vertical line. Between these points, you can trace any path that does not stop or back up. Each path describes a valid form of the statement.

The path always approaches elements from the left and continues to the right. The only exception is the separators in loops (elements that indicate a path you can repeat). The path approaches these separators counterclockwise from the right.

Along a path, you might encounter the following elements:

KEYWORD
: You must spell a word in uppercase letters exactly as shown; however, you can use either uppercase or lowercase letters when you enter it.

(.,;+*-/)
: Punctuation and mathematical notations are literal symbols that you must enter exactly as shown.

' '
: Single quotes are literal symbols that you must enter as shown.

variable
: A word in italics represents a value that you must supply. The nature of the value is explained immediately following the diagram unless the variable appears in a box. In that case, the page number of the detailed explanation follows the variable name.

ADD Clause p. 1-14
: A reference in a box represents a subdiagram on the same page (if no page number is supplied) or on a specified page. Imagine that the subdiagram is spliced into the main diagram at this point.

ESQL
: A code in an icon is a signal warning you that this path is valid only for some products or under certain conditions. The codes indicate the products or conditions that support the path. The following codes are used:

OL	This path is valid only for INFORMIX-OnLine Dynamic Server.
SE	This path is valid only for INFORMIX-SE.
D/B	This path is valid only for DB-Access.
ESQL	This path is valid for SQL statements in INFORMIX-ESQL/C and INFORMIX-ESQL/COBOL.

E/C — This path is valid only for INFORMIX-ESQL/C.

E/CO — This path is valid only for INFORMIX-ESQL/COBOL.

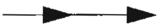
SPL — This path is valid only if you are using Informix Stored Procedure Language (SPL).

NLS — This path is valid only if you have created your database as an Native Language Support (NLS) database.

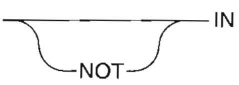
OP — This path is valid only for INFORMIX-OnLine/Optical.

+ — This path is an Informix extension to ANSI SQL-92 entry level standard SQL. If you initiate Informix extension checking and include this syntax branch, you receive a warning. If you have set the DBANSIWARN environment variable at compile time, or have used the **-ansi** compile flag, you receive warnings at compile time. If you have DBANSIWARN set at run time, or if you compiled with the **-ansi** flag, warning flags are set in the **sqlwarn** structure.

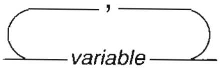
— ALL — A shaded option is the default. Even if you do not explicitly type the option, it will be in effect unless you choose another option.


Syntax enclosed in a pair of arrows indicates that this is a subdiagram.

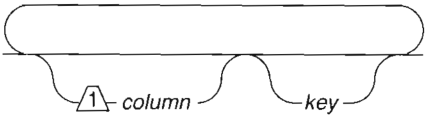
The vertical line is a terminator and indicates that the statement is complete.



A branch below the main line indicates an option



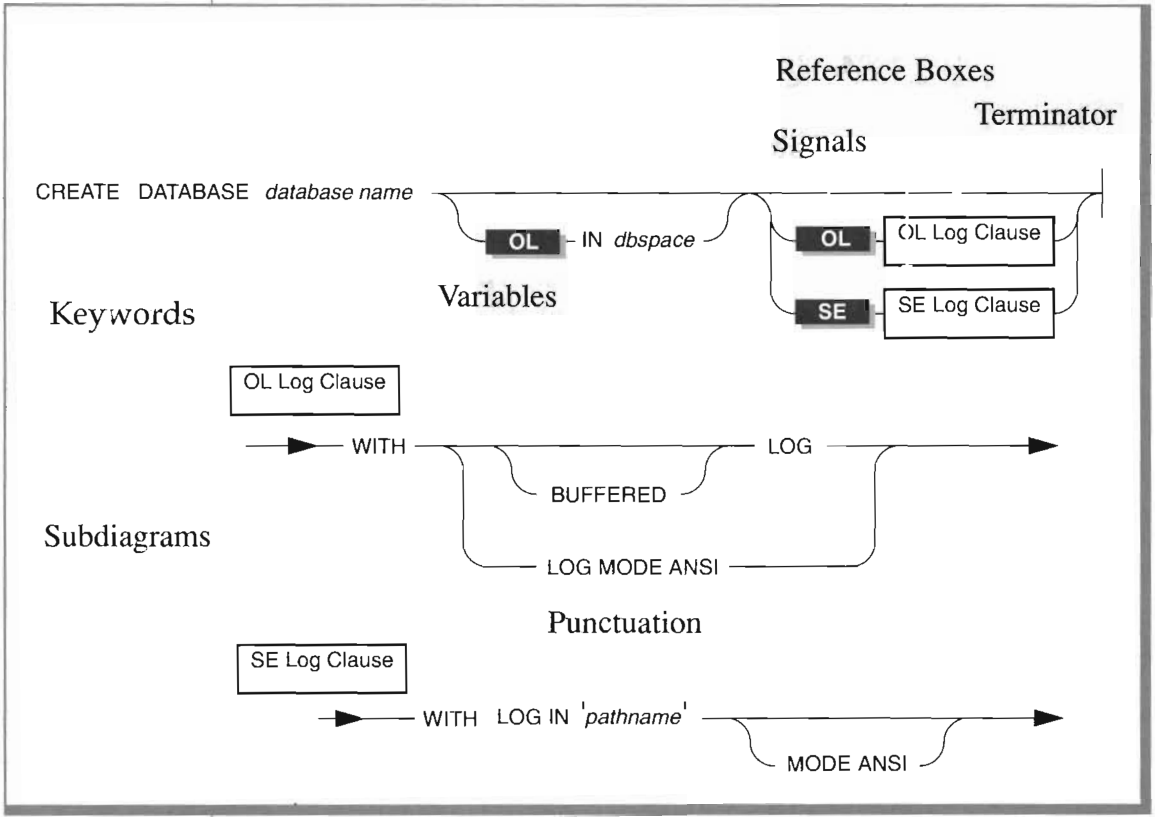
A loop indicates a path that can be repeated. Punctuation along the top of the loop indicates the separator symbol for list items.



A gate (/1\ ) in an option indicates that you can only use that option once, even if it is within a larger loop.

Figure 1 shows the elements of a syntax diagram for the CREATE DATABASE statement.Many syntax diagram conventions are illustrated.

Figure 1
Elements of a syntax diagram



To construct a statement using Figure 1, start at the top left with the keywords CREATE DATABASE. Then follow the diagram to the right, proceeding through the options that you want. Figure 1 conveys the following information:

1. You must type the words CREATE DATABASE.
2. You must supply a *database name*.
3. You can stop, taking the direct route to the terminator, or you can take one or more of the optional paths.
4. If desired, you can designate a dbspace by typing the word IN and a dbspace name.
5. If desired, you can specify logging. Here, you are constrained by the database server with which you are working.
   - ❑ If you are using INFORMIX-OnLine Dynamic Server, go to the subdiagram named *OL Log Clause*. Follow the subdiagram by typing the keyword WITH, then choosing and typing either LOG, BUFFERED LOG, or LOG MODE ANSI. Then, follow the arrow back to the main diagram.
   - ❑ If you are using INFORMIX-SE, go to the subdiagram named *SE Log Clause*. Follow the subdiagram by typing the keywords WITH LOG IN, typing a quote, supplying a pathname, and closing the quotes. You can then choose the MODE ANSI option below the line or continue to follow the line across.
6. Once you are back at the main diagram, you come to the terminator. Your CREATE DATABASE statement is complete.

Icons in the Text

In the statement descriptions in this manual, icons that appear in the left margin indicate that the text located between the diamond symbols (✦) is valid only for a specific product or products, or under certain conditions. In addition to the icons described on page 7 of this Introduction, the following icons can appear in the left margin. These icons indicate material that is relevant under specific conditions:

ANSI

This icon indicates that the functionality described in the text that is located between the diamond symbols (♦) is valid only if your database is ANSI-compliant.

X/O

This icon indicates that the functionality described in the text that is located between the diamond symbols (♦) conforms to X/Open specifications for dynamic SQL. This functionality is available when you compile your SQL API with the **-xopen** flag.

Example Code Conventions

Examples of SQL code occur throughout this manual. Except where noted, the code is not specific to any single Informix application development tool. If only SQL statements are listed in the example, they are not delineated by semicolons. To use this SQL code for a specific product, you must apply the syntax rules for that product. For example, if you are using the Query-language option of DB-Access, you must delineate multiple statements with semicolons. If you are using an SQL API, you must use EXEC SQL and a semicolon (or other appropriate delimiters) at the start and end of each statement, respectively.

For instance, you might see the following example code:

```
CONNECT TO stores7
.
.
.
DELETE FROM customer
    WHERE customer_num = 121
.
.
.
COMMIT WORK
DISCONNECT CURRENT
```

For detailed directions on using SQL statements for a particular application development tool or SQL API, see the manual for your product.

Also note that dots in the example indicate that more code would be added in a full application, but it is not necessary to show it to describe the concept being discussed.

Useful On-Line Files

In addition to the Informix set of manuals, the following on-line files, located in the **$INFORMIXDIR/release** directory, might supplement the information in this manual:

Documentation Notes describe features not covered in the manual or that have been modified since publication. The file containing the Documentation Notes for this product is called **SQLSDOC_7.1**.

Release Notes describe feature differences from earlier versions of Informix products and how these differences might affect current products. The file containing the Release Notes for Version 7.1 of Informix database server products is called **SERVERS_7.1**.

Machine Notes describe any special actions required to configure and use Informix products on your computer. Machine Notes are named for the product described, for example, the Machine Notes file for INFORMIX-OnLine Dynamic Server is **ONLINE_7.1**.

Please examine these files because they contain vital information about application and performance issues.

ASCII and PostScript Error Message Files

Informix software products provide ASCII files that contain all the Informix error messages and their corrective actions. To access the error messages in the ASCII file, Informix provides scripts that let you display error messages on the screen (**finderr**) or print formatted error messages (**rofferr**). See the Introduction to the *Informix Error Messages* manual for a detailed description of these scripts.

The optional Informix Messages and Corrections product provides PostScript files that contain the error messages and their corrective actions. If you have installed this product, you can print the PostScript files on a PostScript printer. The PostScript error messages are distributed in a number of files of the format **errmsg1.ps**, **errmsg2.ps**, and so on. These files are located in the **$INFORMIXDIR/msg** directory.

The Demonstration Database

The DB-Access utility, which is provided with your Informix database server products, includes a demonstration database called **stores7** that contains information about a fictitious wholesale sporting-goods distributor. The sample command files that make up a demonstration application are also included.

Most of the examples in this manual are based on the **stores7** demonstration database. The **stores7** database is described in detail and its contents are listed in Appendix A of the *Informix Guide to SQL: Reference.*

The script that you use to install the demonstration database is called **dbaccessdemo7** and is located in the **$INFORMIXDIR/bin** directory. The database name that you supply is the name given to the demonstration database. If you do not supply a database name, the name defaults to **stores7**. Follow these rules for naming your database:

- Names can be up to 18 characters long for INFORMIX-OnLine Dynamic Server databases and up to 10 characters long for INFORMIX-SE databases.
- The first character of a name must be a letter or an underscore (_).
- You can use letters, characters, and underscores (_) for the rest of the name.
- DB-Access makes no distinction between uppercase and lowercase letters.
- The database name should be unique.

When you run **dbaccessdemo7**, you are, as the creator of the database, the owner and Database Administrator (DBA) of that database.

If you installed your Informix database server product according to the installation instructions, the files that make up the demonstration database are protected so you cannot make any changes to the original database.

You can run the **dbaccessdemo7** script again whenever you want to work with a fresh demonstration database. The script prompts you when the creation of the database is complete, and asks if you would like to copy the sample command files to the current directory. Enter `N` if you have made changes to the sample files and do not want them replaced with the original versions. Enter `Y` if you want to copy over the sample command files.

Creating the Demonstration Database

Use the following steps to create and populate the demonstration database:

1. Set the **INFORMIXDIR** environment variable so that it contains the name of the directory in which your Informix products are installed. Set **INFORMIXSERVER** to the name of the default database server. The name of the default database server must exist in the **$INFORMIXDIR/etc/sqlhosts** file. (For a full description of environment variables, see Chapter 4 of the *Informix Guide to SQL: Reference*.) For information about **sqlhosts**, see the *INFORMIX-OnLine Dynamic Server Administrator's Guide* or the *INFORMIX-SE Administrator's Guide*.
2. Create a new directory for the SQL command files. Create the directory by entering the following command:

   ```
   mkdir dirname
   ```
3. Make the new directory the current directory by entering the following command:

   ```
   cd dirname
   ```

4. Create the demonstration database and copy over the sample command files by entering the **dbaccessdemo7** command:

   To create the database without logging, enter the following command:

   ```
   dbaccessdemo7 dbname
   ```

   To create the demonstration database with logging enter the following command:

   ```
   dbaccessdemo7 -log dbname
   ```

   If you are using INFORMIX-OnLine Dynamic Server, by default the data for the database is put into the root dbspace. If you wish, you can specify a dbspace for the demonstration database.

   To create a demonstration database in a particular dbspace enter the following command:

   ```
   dbaccessdemo7 dbname -dbspace dbspacename
   ```

   You can specify all the options in one command as shown in the following command:

   ```
   dbaccessdemo7 -log dbname -dbspace dbspacename
   ```

   If you are using INFORMIX-SE, a subdirectory called ***dbname*.dbs** is created in your current directory and the database files associated with **stores7** are placed there. You will see both data (**.dat**) and index (**.idx**) files in the ***dbname*.dbs** directory. (If you specify a dbspace name, it will be ignored.)

   To use the database and the command files that have been copied to your directory, you must have UNIX read and execute permissions for each directory in the pathname of the directory from which you ran the **dbaccessdemo7** script. Check with your system administrator for more information about operating-system file and directory permissions. UNIX permissions are discussed in the *INFORMIX-OnLine Dynamic Server Administrator's Guide* and the *INFORMIX-SE Administrator's Guide*

5. To give someone else the permissions to access the command files in your directory, use the UNIX `chmod` command.

6. To give someone else the access to the database that you have created, grant them the appropriate privileges using the GRANT statement. To remove privileges, use the REVOKE statement. The GRANT and REVOKE statements are described in Chapter 1 of this manual.

Compliance with Industry Standards

The American National Standards Institute (ANSI) has established a set of industry standards for SQL. Informix SQL-based products are fully compliant with SQL-92 Entry Level (published as ANSI X3.135-1992), which is identical to ISO 9075:1992 on INFORMIX-OnLine Dynamic Server. In addition, many features of OnLine comply with the SQL-92 Intermediate and Full Level and X/Open CAE (common applications environment) standards.

Informix SQL-based products are compliant with ANSI SQL-92 Entry Level (published as ANSI X3.135-1992) on INFORMIX-SE with the following exceptions:

- Effective checking of constraints
- Serializable transactions

New Features in Informix Version 7.1 Products That Use SQL

The Introduction to each Version 7.1 product manual contains a list of new features for that product. The Introduction to each manual in the Version 7.1 *Informix Guide to SQL* series contains a list of new SQL features.

A comprehensive listing of all the new features for Version 7.1 Informix products is found in the Release Notes file called **SERVERS_7.1**.

This section highlights the major new features implemented in Version 7.1 of Informix products that use SQL.

- New FRAGMENT BY clause in the CREATE TABLE statement

  The CREATE TABLE statement contains a new FRAGMENT BY clause that allows you to create fragmented tables. Fragmentation enables you to define groups of rows within a table based on a rule and to specify a separate dbspace for each group.

- New FRAGMENT BY EXPRESSION clause in the CREATE INDEX statement

  The CREATE INDEX statement contains a new FRAGMENT BY EXPRESSION clause that allows you to create fragmented indexes.

- New ALTER FRAGMENT statement

  You can use the ALTER FRAGMENT statement to dynamically alter an existing table or index-fragmentation strategy as well as to initially create fragments.

- New ADD ROWIDS and DROP ROWIDS clauses in the ALTER TABLE statement

  You can use the ADD ROWIDS clause to add the rowid column to fragmented tables. You can use the DROP ROWIDS clause to drop the rowid column from a fragmented table that has a rowid column.

- New FRAGMENTS keyword in INFO statement

  The FRAGMENTS keyword allows you to display the dbspace names where fragments are located for a specified table.

- New SET DATASKIP statement

  The SET DATASKIP statement allows you to control whether INFORMIX-OnLine Dynamic Server skips any dbspaces in the fragmentation list that are unavailable while a transaction is being processed.

- New **sysfragments** system catalog table

  The **sysfragments** table stores fragmentation information for tables and indexes.

- New SET PDQPRIORITY statement

  The SET PDQPRIORITY statement supports the parallel-data-query (PDQ) functionality. PDQ allows queries to be processed in parallel. The SET PDQPRIORITY statement allows an application to set the query priority level dynamically within the application. The query priority level refers to the degree of parallelism to be used for queries.

- New output in the SET EXPLAIN statement

  The output of the SET EXPLAIN statement has been enhanced to support the new fragmentation and PDQ features.

- Changes to the **dbexport** and **dbschema** utilities

  The **dbexport** and **dbschema** utilities have been enhanced to support the new fragmentation and PDQ features.
- New SET TRANSACTION statement

  You can now use the ANSI isolation levels for transactions. In addition, the SET TRANSACTION statement permits you to use read-only and read-write access modes.
- New TRIM function

  The TRIM function allows you to strip leading characters or blanks, trailing characters or blanks, or both from a character string.
- Support for delimited identifiers

  Delimited identifiers allow you to specify as identifiers character strings that are otherwise identical to SQL reserved keywords. They also allow you to specify nonalphanumeric characters in database object names and to use lowercase identifiers. This feature is only activated when the **DELIMIDENT** environment variable is set.
- WITH REOPTIMIZATION clause for the OPEN CURSOR statement

  The WITH REOPTIMIZATION clause allows you to reoptimize a query plan if that query plan becomes deoptimized.
- INTO clause for the EXECUTE statement

  The INTO clause allows you to execute a prepared singleton select statement or execute a prepared EXECUTE PROCEDURE statement. In addition, you can specify output variables using the INTO clause.
- New environment variables

  The following environment variables, described in Chapter 4, "Environment Variables," of the *Informix Guide to SQL: Reference* are new in Version 7.1:

  - **DBBLOBBUF**
  - **DELIMIDENT**
  - **BUF_FET_SIZE**
  - **OPTCOMPIND**
  - **PDQPRIORITY**

- XPG4 SQL CAE compliance
  - X/Open information schemas

    The information schema consists of a set of read-only views from which you can retrieve information about any tables, views, and columns to which you have access.
  - Support for CHARACTER VARYING syntax

    The CHARACTER VARYING data type syntax, which is compliant with XPG4 and ANSI, performs like the Informix data type VARCHAR. The data type supports both a maximum and minimum length.
  - Specification of cursors as READ ONLY

    You can specify cursors as READ ONLY with the DECLARE statement.
  - Optional specification of RESTRICT/CASCADE on the DROP TABLE, DROP VIEW, and REVOKE statements

    You can now restrict drop and revoke actions with the RESTRICT keyword. Previously, all drop requests to tables and views and revoke requests cascaded to drop all related tables, views, and permissions. Now you can specify the RESTRICT keyword on these statements to disallow the drop or revoke.
- Hexadecimal load/unload format

  This feature allows the LOAD/UNLOAD statements, as well as the **dbexport**, **dbimport**, and **dbload** utilities, to format the nonprintable ASCII characters in hexadecimal format. It affects the formats of unload files if the CHAR/VARCHAR NCHAR/NVARCHAR data types contain nonprintable ASCII characters. It also allows you to load the nonprintable characters from the file into the database after they are prepared in hex format.