

IBM Informix DB-Access

User's Guide

Version 5.2
November 2002
Part No. 000-9124

Note:

Before using this information and the product it supports, read the information in the appendix entitled "Notices."

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1996, 2002. All rights reserved.

US Government User Restricted Rights—Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of Contents

Introduction

In This Introduction	3
Summary of Chapters	4
DB-Access and Other IBM Products	5
Other Useful Documentation	6
How to Use This Manual	7
Typographical Conventions	7
Command-Line Conventions	8
Useful Online Files	11
ASCII Error Message Files	12
Using the ASCII Error Message File.	12
The Demonstration Database	14
Creating the Demonstration Database on IBM Informix OnLine	16
Creating the Demonstration Database on IBM Informix SE	17
Compliance with Industry Standards	18
New Features in IBM Informix Server Products, Version 5.x	18
IBM Welcomes Your Comments	20

Chapter 1

Working with DB-Access

In This Chapter	1-3
What Is DB-Access?	1-4
Differences Between DB-Access and IBM Informix SQL.	1-4
Differences Between IBM Informix OnLine and IBM Informix SE	1-5
Preparing to Use DB-Access	1-6
Setting Environment Variables for DB-Access	1-6
Creating the Demonstration Database	1-7
Finding Your Way Around DB-Access	1-7
How to Use Your Terminal	1-8

Invoking DB-Access	1-9
Displaying the Software Version Number	1-10
Calling up the DB-Access Main Menu	1-11
Invoking DB-Access and Selecting a Database.	1-12
Going to a Submenu	1-13
Going to a Submenu and Submenu Option.	1-14
Executing a Command File	1-16
Echoing File Input to the Screen.	1-17
Checking for ANSI Compliance.	1-18
The DB-Access Main Menu	1-19
The DB-Access Menu Structure	1-20
Using DB-Access Menus and Screens	1-21
Using Menu Screens.	1-21
Using Text-Entry Screens	1-23
Using the HELP Screen	1-25
Using a Text Editor	1-26
Using the SQL Editor	1-26
Using a System Editor	1-29

Chapter 2 Creating Databases and Tables

In This Chapter	2-3
Working with a Database	2-4
Selecting a Database.	2-4
Creating a Database	2-5
Dropping a Database	2-5
Working with a Table	2-5
Creating a Table	2-6
Modifying a Table	2-6
Dropping a Table.	2-6
Getting Table Information.	2-6
The DB-Access Facilities	2-7
Using the Interactive Schema Editor	2-7
Using SQL Statements	2-9
Method 1: Using the Menu Options	2-12
How To Create a Database	2-12
Leaving the Menu	2-14
Where System Files Are Stored	2-14

How To Create a Table in the Database	2-15
Calling up the Table Schema	2-15
Exiting the Schema Editor	2-23
Where Table Information Is Stored	2-24
How To Modify a Table in the Database	2-24
Using the Alter Option	2-25
How To Get Table Information	2-28
How to Drop a Table	2-28
How to Drop a Database	2-30
Method 2: Using SQL	2-31
How To Create a Database	2-32
How to Create a Table	2-33
Assigning Column Constraints	2-34
Creating an Index on a Table	2-34
How to Modify a Table	2-34
Adding a Column to a Table	2-35
Modifying a Column in a Table	2-35
Dropping a Column from a Table	2-36
Renaming a Column in a Table	2-36
Adding and Dropping Constraints	2-37
How To Get Table Information	2-37
How to Drop a Table	2-38
How to Drop a Database	2-39

Chapter 3 The Query-Language Menu Option

In This Chapter	3-3
Choosing the Query-Language Option	3-4
Entering a New SQL Statement	3-6
Running an SQL Statement	3-7
Running Error-Free Statements	3-8
What Happens When There Are Errors	3-8
Viewing the Next Page of Data	3-9
Modifying an SQL Statement	3-10
Editing an SQL Statement	3-11
Redirecting Query Results	3-12
Sending Output to a Printer	3-13
Sending Output to a File.	3-14
Sending Output to a Pipe	3-15
Choosing an Existing SQL Statement	3-16
Sample SQL Command Files	3-17

Saving the Current SQL Statement	3-19
Displaying Table Information	3-20
Dropping an SQL Statement	3-23

Chapter 4 The Database Menu Option

In This Chapter	4-3
Choosing the Database Option	4-4
The SELECT DATABASE Screen	4-5
The CREATE DATABASE Screen	4-7
The DROP DATABASE Screen	4-9

Chapter 5 The Table Menu Option

In This Chapter	5-3
Selecting the TABLE Menu	5-4
Creating a Table	5-5
The CREATE TABLE Screen	5-6
Naming the Table	5-6
Building the Table Schema	5-7
Differences Between IBM Informix OnLine and IBM Informix SE	5-8
Exiting the CREATE TABLE Menu.	5-8
Adding a Column to a Table	5-9
Adding to the Table Schema	5-17
Building the Table Schema	5-17
Modifying a Column in a Table	5-18
Dropping a Column.	5-25
Rebuilding the Table Schema.	5-26
Displaying Subsequent Table Screens	5-27
Setting Table Options	5-27
Altering a Table	5-31
The ALTER TABLE Screen	5-31
Differences Between IBM Informix OnLine and IBM Informix SE.	5-33
Exiting the ALTER TABLE Screen	5-33
Adding a Column with the ALTER TABLE Menu	5-34
Modifying a Column with the ALTER TABLE Menu	5-35
Dropping a Column with the ALTER TABLE Menu	5-37
Using the Screen Option on the ALTER TABLE Menu	5-38
Changing Table Options with the ALTER TABLE Menu	5-38
Displaying Table Information	5-39

Dropping a Table	5-42
Confirming Your Decision	5-43
Exiting the DROP TABLE Screen	5-43

Appendix A Sample Command Files

Appendix B How to Read Online Help

Appendix C Notices

Index

Introduction

In This Introduction	3
Summary of Chapters	4
DB-Access and Other IBM Products	5
Other Useful Documentation	6
How to Use This Manual	7
Typographical Conventions	7
Command-Line Conventions	8
Useful Online Files	11
ASCII Error Message Files	12
Using the ASCII Error Message File.	12
The finderr Script.	13
The rofferr Script	13
The Demonstration Database	14
Creating the Demonstration Database on IBM Informix OnLine	16
Creating the Demonstration Database on IBM Informix SE	17
Compliance with Industry Standards	18
New Features in IBM Informix Server Products, Version 5.x.	18
IBM Welcomes Your Comments	20

In This Introduction

DB-Access is a utility that allows you to access, modify, and retrieve the information in a relational database. It provides much of the same functionality as the IBM Informix SQL database management product. However, DB-Access does not allow you to create and compile custom reports or forms or to create and run custom menus. These features are available only in IBM Informix SQL.

With DB-Access, you can perform these database management tasks:

- Create, modify, and drop databases and tables
- Load data from operating system files
- Run queries using an interactive query language
- Insert, delete, update, and query on data in the database
- Create and drop privileges and indexes

Summary of Chapters

The *IBM Informix DB-Access User's Guide* is a complete guide to the features that make up the DB-Access relational database access utility, and includes the following chapters:

- The Introduction tells how DB-Access fits into the IBM Informix family of products and manuals, explains how to use the manual, introduces the demonstration database from which the product examples are drawn, describes ASCII error message files, and lists the new features for Version 5.0 of IBM Informix server products.
- [Chapter 1, "Working with DB-Access,"](#) provides an overview of DB-Access with IBM Informix OnLine and IBM Informix SE. It explains how to use your terminal and the system editor and how to invoke the utility and use the various screens and menus. It also illustrates and describes the DB-Access Main Menu structure and tells how to get online help.
- [Chapter 2, "Creating Databases and Tables,"](#) illustrates how to create, modify, drop, and select databases and tables using either an SQL statement or a Main Menu option. It also introduces the Interactive Schema Editor (ISED) and the Informix implementation of SQL.
- [Chapter 3, "The Query-Language Menu Option,"](#) illustrates and describes how to use the various menu options when you select the Query-Language option from the DB-Access Main Menu.
- [Chapter 4, "The Database Menu Option,"](#) illustrates and describes how to use the various menu options when you select the Database option from the DB-Access Main Menu.
- [Chapter 5, "The Table Menu Option,"](#) illustrates and describes how to use the various menu options when you select the Table option from the DB-Access Main Menu.
- [Appendix A, "Sample Command Files,"](#) illustrates and briefly describes all the command files provided with DB-Access that you can use when you run the script to install the demonstration database.
- [Appendix B, "How to Read Online Help,"](#) illustrates how to read the syntax diagrams for SQL statements described in online help.

- A Notices appendix describes IBM products, features, and services.
- An index directs you to areas of particular interest.

DB-Access and Other IBM Products

DB-Access is one of many IBM Informix application development tools, CASE tools, and utilities currently available. Other IBM Informix application development tools include products like IBM Informix 4GL and the IBM Informix 4GL Interactive Debugger, and the IBM Informix embedded-language products, such as IBM Informix ESQL/C.

DB-Access works with a database server, either IBM Informix OnLine or IBM Informix SE. If you are running applications on a network, you will use an IBM Informix client/server product such as IBM Informix NET or IBM Informix STAR. IBM Informix NET is the communication facility for multiple IBM Informix SE database servers. IBM Informix STAR allows distributed database access to multiple IBM Informix OnLine database servers.

Other Useful Documentation

You may want to refer to a number of related IBM Informix product documents that complement the *IBM Informix DB-Access User's Guide*.

- If you have never used SQL (Structured Query Language) or an IBM Informix application development tool before, you might want to read *IBM Informix Guide to SQL: Tutorial* to learn basic database design and implementation concepts.
- A companion volume to the Tutorial, *IBM Informix Guide to SQL: Reference*, provides full information on the structure and contents of the demonstration database that is provided with DB-Access. It includes details of the Informix system catalog, describes Informix and common UNIX environment variables that should be set, and defines column data types supported by IBM Informix products. Further, it provides a detailed description of all the SQL statements supported by IBM Informix products. It also contains a glossary of useful terms.
- You, or whoever installs your IBM Informix OnLine or IBM Informix SE database server, should refer to the *UNIX Products Installation Guide* for your particular release to ensure that DB-Access is properly set up before you begin to work with it.
- If you are using DB-Access across a network, you may also want to refer to the appropriate *IBM Informix NET and IBM Informix STAR Installation and Configuration Guide*.
- Depending on the database server you are using, you or your system administrator will need either the *IBM Informix OnLine Administrator's Guide* or the *IBM Informix SE Administrator's Guide*.
- When errors occur, you can look them up, by number, and learn their cause and solution in the *IBM Informix Error Messages* manual. If you prefer, you can look up the error messages in the online message file described in the section "[ASCII Error Message Files](#)" later in this introduction.

How to Use This Manual

This manual assumes that you are using IBM Informix OnLine as your database server. Features and behavior specific to IBM Informix SE are noted throughout the manual.

Typographical Conventions

The *IBM Informix DB-Access User's Guide* uses 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 the manual:

Convention	Meaning
KEYWORD	All primary elements in a programming language statement (keywords) appear in uppercase letters in a serif font.
<i>italics</i>	Within text, new terms and emphasized words appear in italics.
<i>italics</i>	Within syntax and code examples, variable values that you are to specify appear in italics.
boldface	Names of program entities (such as classes, events, and tables), environment variables, file and pathnames, and interface elements (such as icons, menu items, and buttons) appear in boldface.
<i>boldface</i>	
monospace <i>monospace</i>	Information that the product displays and information that you enter appear in a monospace typeface.
KEYSTROKE	Keys that you are to press appear in uppercase letters in a sans serif font.



Tip: When you are instructed to “enter” characters or to “execute” a command, immediately press RETURN after the entry. When you are instructed to “type” the text or to “press” other keys, no RETURN is required.

Command-Line Conventions

DB-Access supports a variety of command-line options. These are commands that you enter at the operating system prompt to perform certain functions or go to specified menus in DB-Access. Each valid command-line option is illustrated in a diagram in [Chapter 1, “Working with DB-Access.”](#)


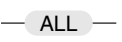


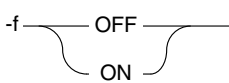
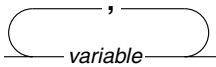
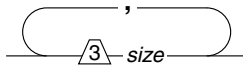
This section defines and illustrates the format of the commands available in DB-Access and other IBM Informix products. These commands have their own conventions, which may include alternative forms of a command, required and optional parts of the command, and so forth.

Each diagram displays the sequences of required and optional elements that are valid in a command. A diagram begins at the upper left with a command. It ends at the upper right 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 command. You must supply a value for words that are in italics.

Along a command-line path, you may encounter the following elements:

Element	Description
command	This required element is usually the product name or other short word that invokes the product or calls the compiler or preprocessor script for a compiled IBM Informix product. It might appear alone or precede one or more options. You must spell a command exactly as shown and use lowercase letters.
<i>variable</i>	A word in italics represents a value that you must supply, such as a database, file, or program name. A table following the diagram explains the value.
-flag	A flag is usually an abbreviation for a function, menu, or option name, or for a compiler or preprocessor argument. You must enter a flag exactly as shown, including the preceding hyphen.
.ext	A filename extension, such as .sql or .cob , might follow a variable that represents a filename. Type this extension exactly as shown, immediately after the name of the file. The extension might be optional in certain products.

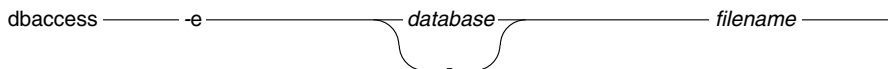
(1 of 2)

Element	Description
(. , ; + * - /)	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.
	A reference in a box represents a subdiagram. Imagine that the subdiagram is spliced into the main diagram at this point. When a page number is not specified, the subdiagram appears on the same page.
	A shaded option is the default action.
	Syntax within a pair of arrows indicates a subdiagram.
	The vertical line terminates the command.
	A branch below the main path indicates an optional path. (Any term on the main path is required, unless a branch can circumvent it.)
	A loop indicates a path that you can repeat. Punctuation along the top of the loop indicates the separator symbol for list items.
	A gate ($\sqrt{3}$) on a path indicates that you can only use that path the indicated number of times, even if it is part of a larger loop. You can specify <i>size</i> no more than three times within this statement segment.

(2 of 2)

The following diagram shows the elements of a DB-Access command used to echo file input to the screen:

Figure 1
Elements of a command-line diagram



To construct a similar command, start at the top left with the command `dbaccess`. Then follow the diagram to the right, including the elements that you want. This diagram conveys the following information:

1. You must type the word `dbaccess`.
2. You can echo the SQL statements in a command file to the screen by typing the flag `-e` before the database name.
3. You must supply a *database* name or use a hyphen (-) to indicate that a database name is specified in the command file that you want to run.
4. You must specify the *filename* of a command file whose SQL statements you want to echo to the screen.

On some command-line diagrams, you can take the direct route to the terminator, or you can take an optional path indicated by a branch below the main line.

Once you are back at the main diagram, you come to the terminator. Your `dbaccess` command is complete. Press RETURN to execute the command.

Useful Online Files

In addition to the IBM Informix set of manuals, the following online files, located in the `$INFORMIXDIR/release` directory, may supplement the information in the *IBM Informix DB-Access User's Guide*.

Online File	Purpose
DBACCESSDOC_5.txt	The documentation notes file describes features that are not covered in the manual or that were modified since publication.
ENGREL_5.txt	The release notes file describes feature differences from earlier versions of IBM Informix products and how these differences might affect current products. This file also contains information about any known problems and their workarounds.
ONLINE_5.txt	The machine notes file describes any special actions that you must take to configure and use IBM Informix products on your computer. Machine notes are named for the product described.

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

DB-Access and a number of other IBM Informix products also provide online help files that walk you through each menu option. To invoke the help feature, simply press CTRL-W wherever you are in DB-Access. (The notational conventions and rules governing SQL statement syntax shown in online help are illustrated in [Appendix B, "How to Read Online Help."](#))

ASCII Error Message Files

IBM Informix software products provide ASCII files that contain all the error messages and their corrective actions. You can access the error messages in the ASCII file with scripts that let you display error messages on the terminal or print formatted error messages.

Using the ASCII Error Message File

You can use the file that contains the ASCII text version of the error messages and their corrective actions in two ways:

- Use the **finderr** script to display one or more error messages on the terminal screen.
- Use the **rofferr** script to print one error message or a range of error messages.

The scripts are in the `$INFORMIXDIR/bin` directory. The ASCII file has the following path:

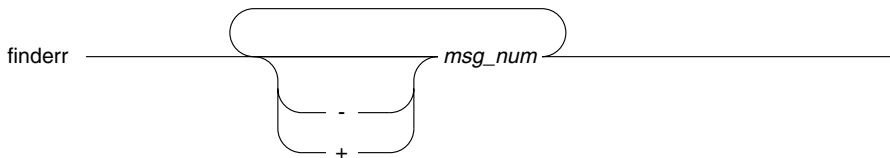
`$INFORMIXDIR/msg/errmsg.txt`

The error message numbers range from -1 to -33000. When you specify these numbers for the **finderr** or **rofferr** scripts, you can omit the minus sign. A few messages have positive numbers; these messages are used solely within the application development tools. In the unlikely event that you want to display them, you must precede the message number with a + sign.

The messages numbered -1 to -100 can be platform-dependent. If the message text for a message in this range does not apply to your platform, check the operating system documentation for the precise meaning of the message number.

The finderr Script

Use the **finderr** script to display one or more error messages and their corrective actions on the terminal screen. The **finderr** script has the following syntax:



msg_num is the number of the error message to display.

You can specify any number of error messages per **finderr** command. The **finderr** command copies all the specified messages, and their corrective actions, to standard output.

For example, to display the -359 error message, you can enter the following command:

```
finderr -359
```

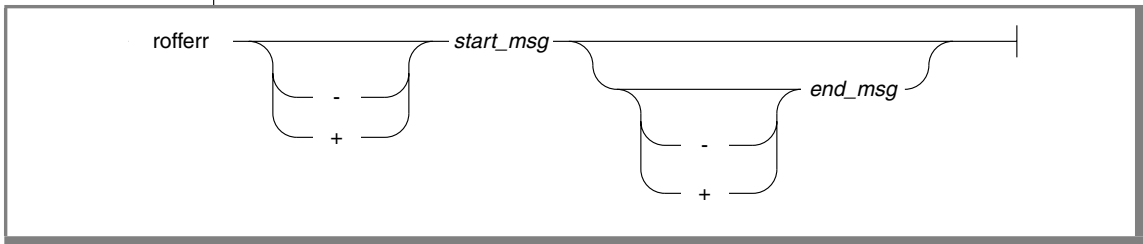
The following example demonstrates how to specify a list of error messages. This example also pipes the output to the UNIX **more** command to control the display. You can also redirect the output to another file so that you can save or print the error messages:

```
finderr 233 107 113 134 143 144 154 | more
```

The rofferr Script

Use the **rofferr** script to format one error message, or a range of error messages, for printing. By default, **rofferr** displays output on the screen. You need to send the output to **nroff** to interpret the formatting commands and then to a printer, or to a file where the **nroff** output is stored until you are ready to print. You can then print the file. For information on using **nroff** and on printing files, see your UNIX documentation.

The **rofferr** script has the following syntax:



`start_msg` is the number of the first error message to format. This error message number is required.

`end_msg` is the number of the last error message to format. This error message number is optional. If you omit `end_msg`, only `start_msg` is formatted.

The following example formats error message -359. It pipes the formatted error message into **nroff** and sends the output of **nroff** to the default printer:

```
rofferr 359 | nroff -man | lpr
```

The following example formats and then prints all the error messages between -1300 and -4999:

```
rofferr -1300 -4999 | nroff -man | lpr
```

The Demonstration Database

Your server software includes a demonstration database called **stores5** that contains information about a fictitious wholesale sporting-goods distributor. The sample command files that make up a demonstration application are included as well. (These files are illustrated in [Appendix A, "Sample Command Files."](#))

Most of the examples in this manual are based on the **stores5** demonstration database. The **stores5** database is described in detail and its contents are listed in the *IBM Informix Guide to SQL: Reference*.

The script you use to install the demonstration database is called **dbaccessdemo5** 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 **stores5**. Follow these rules for naming your database:

- Names for databases can be up to 10 characters long.
- The first character of a name must be a letter.
- 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 **dbaccessdemo5**, you are, as the creator of the database, the owner and Database Administrator (DBA) of that database.

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

You can run the **dbaccessdemo5** 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. Answer "N" to the prompt if you have made changes to the sample files and do not want them replaced with the original versions. Answer "Y" to the prompt if you want to copy over the sample command files.

Creating the Demonstration Database on IBM Informix OnLine

To create and populate the demonstration database in the IBM Informix OnLine environment

1. Set the INFORMIXDIR environment so that it contains the name of the directory in which your IBM Informix products are installed. Set SQLEXEC to `$INFORMIXDIR/lib/sqlturbo`. (For a full description of environment variables, see the *IBM Informix Guide to SQL: Reference*.)
2. Create a new directory for the SQL command files. Create the directory by entering

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

```
cd dirname
```
4. Create the demonstration database and copy over the sample command files by entering

```
dbaccessdemo5 dbname
```

The data for the database is put into the root dbspace.

To give someone else the SQL privileges to access the data, use the GRANT and REVOKE statements. The GRANT and REVOKE statements are described in the *IBM Informix Guide to SQL: Reference*.

To use 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 `dbaccessdemo5` script. To give someone else the permissions to access the command files in your directory, use the UNIX `chmod` command.

Creating the Demonstration Database on IBM Informix SE

To create and populate the demonstration database in the IBM Informix SE environment

1. Set the INFORMIXDIR environment so that it contains the name of the directory in which your IBM Informix products are installed. Set SQLEXEC to `$INFORMIXDIR/lib/sqlexec`. (For a full description of environment variables, see the *IBM Informix Guide to SQL: Reference*.)
2. Create a new directory for the demonstration database. This directory will contain the example command files included with the demonstration database. Create the directory by entering

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

```
cd dirname
```
4. Create the demonstration database and copy over the sample command files by entering

```
dbaccessdemo5 dbname
```

When you run the **dbaccessdemo5** script, it creates a subdirectory called *dbname.dbs* in your current directory and places the **stores5** database files there. You will see both data and index files in the *dbname.dbs* directory.

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 **dbaccessdemo5** script. To give someone else the permissions to access the command files in your directory, use the UNIX **chmod** command. Check with your system administrator for more information about operating system file and directory permissions. UNIX permissions are discussed in the *IBM Informix SE Administrator's Guide*.

To give someone else access to the database that you have created, grant them the appropriate privileges using the GRANT statement in DB-Access. To remove privileges, use the REVOKE statement. The GRANT and REVOKE statements are described in the *IBM Informix Guide to SQL: Reference*.

Compliance with Industry Standards

The American National Standards Institute (ANSI) has established a set of industry standards for SQL. IBM 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. In addition, many features of Informix database servers comply with the SQL-92 Intermediate and Full Level and X/Open SQL CAE (common applications environment) standards.

New Features in IBM Informix Server Products, Version 5.x

This section highlights the major new features implemented in Version 5.0 of IBM Informix server products:

- **Enhanced Connectivity (IBM Informix OnLine only)**

The version 5.2 IBM Informix OnLine database server enables you to connect to Version 7.x client application tools when both server and client are installed in the same machine.

- **Enhanced support for chunk offsets (IBM Informix OnLine only)**

The version 5.2 IBM Informix OnLine database server supports chunk offset values up to 2 terabytes.

- **Referential and Entity Integrity**

New data integrity constraints allow you to specify a column or columns as representing a *primary* or *foreign key* of a table upon creation, and to establish dependencies between tables. Once specified, a parent-child relationship between two tables is enforced by the database server. Other constraints allow you to specify a default value for a column, or to specify a condition for a column that an inserted value must meet.

- **Stored Procedures**

A stored procedure is a function written by a user using a combination of SQL statements and Stored Procedure Language (SPL). Once created, a procedure is stored as an object in the database in a compiled, optimized form, and is available to other users with the appropriate privileges. In a client/server environment, the use of stored procedures can significantly reduce network traffic.

- **Dynamic SQL**

Support is provided for the X/Open implementation of dynamic SQL using a system descriptor area. This support involves the new SQL statements `ALLOCATE DESCRIPTOR`, `DEALLOCATE DESCRIPTOR`, `GET DESCRIPTOR`, and `SET DESCRIPTOR`, as well as changes in the syntax of existing dynamic management statements.

- **Optimizer Enhancement**

You can use the new `SET OPTIMIZATION` statement to instruct the database server to select a high or low level of query optimization. The default level of `HIGH` causes the database server to examine and select the best of all possible optimization strategies. Since this level of optimization may result in a longer-than-desired optimization time for some queries, you have the option of setting an optimization level of `LOW`.

- **Relay Module (IBM Informix NET only)**

The new Relay Module component of IBM Informix NET resides on the client machine in a distributed data processing environment and *relays* messages between the application development tool and an IBM Informix OnLine or IBM Informix SE database server through a network interface. The Relay Module allows version 5.0 application development tools to connect to a remote database server without the need to run an Informix database server process on the client.

- **Fast Indexing (IBM Informix OnLine only)**

The version 5.0 IBM Informix OnLine database server uses a new method of creating large indexes when you execute the `CREATE INDEX` statement. In this method, index entries are sorted prior to their insertion into the B+ tree structure, resulting in faster index creation.

- Two-Phase Commit (IBM Informix STAR only)

The new two-phase commit protocol allows you to manipulate data in multiple databases on multiple OnLine database servers within a single transaction. It ensures that transactions that span more than one OnLine database server are committed on an all-or-nothing basis.

- Support for Transaction Processing in the XA Environment (IBM Informix TP/XA only)

A new product, IBM Informix TP/XA, allows you to use the IBM Informix OnLine database server as a Resource Manager in conformance with the *X/Open Preliminary Specification for Distributed Transaction Processing*. The *IBM Informix TP/XA User Manual* describes the changes in the behavior of existing SQL statements that manage transactions in an XA environment.

IBM Welcomes Your Comments

To help us with future versions of our manuals, let us know about any corrections or clarifications that you would find useful. Include the following information:

- The name and version of your manual
- Any comments that you have about the manual
- Your name, address, and phone number

Send electronic mail to us at the following address:

`docinf@us.ibm.com`

This address is reserved for reporting errors and omissions in our documentation. For immediate help with a technical problem, contact Customer Services.

Working with DB-Access

In This Chapter	1-3
What Is DB-Access?	1-4
Differences Between DB-Access and IBM Informix SQL	1-4
Differences Between IBM Informix OnLine and IBM Informix SE	1-5
Preparing to Use DB-Access.	1-6
Setting Environment Variables for DB-Access	1-6
Creating the Demonstration Database	1-7
Finding Your Way Around DB-Access	1-7
How to Use Your Terminal	1-8
Invoking DB-Access	1-9
Displaying the Software Version Number	1-10
Calling up the DB-Access Main Menu	1-11
Invoking DB-Access and Selecting a Database	1-12
Going to a Submenu	1-13
Going to a Submenu and Submenu Option	1-14
DATABASE Menu Options	1-14
SQL Menu Options	1-15
TABLE Menu Options:	1-15
Executing a Command File.	1-16
Echoing File Input to the Screen	1-17
Checking for ANSI Compliance	1-18
The DB-Access Main Menu	1-19
The DB-Access Menu Structure	1-20

Using DB-Access Menus and Screens.	1-21
Using Menu Screens	1-21
Selecting an Option	1-22
Leaving a Menu Screen	1-22
Getting Help	1-23
Using Text-Entry Screens	1-23
Entering Text on the Screen	1-24
Leaving a Text-Entry Screen	1-24
Getting Help	1-24
Using the HELP Screen	1-25
Using a Text Editor	1-26
Using the SQL Editor	1-26
Using a System Editor	1-29

In This Chapter

This chapter introduces DB-Access. It tells you how to get started with DB-Access and acquaints you with the various types of menus, screens, functions, and editors that you use when working with the utility.

The following topics are discussed in this chapter:

- Preparing to use DB-Access
- Setting environment variables for DB-Access
- How to use special keys on your terminal
- How to invoke DB-Access from the command line
- The DB-Access menu structure
- How to use menus and text-entry screens
- Getting online help
- Using the SQL and system text editors

To make the best use of DB-Access, you should understand the basic concepts in this chapter. This manual also refers to other IBM Informix publications as the need arises. These references can be particularly helpful if you have had limited experience with computers or database management systems.

What Is DB-Access?

DB-Access is a utility that lets you access and manipulate the data in a *relational database* and perform a variety of data management tasks such as organizing, storing, retrieving, and viewing data.

A database is a collection of information or data and is made up of *tables*. In turn, the tables are made up of *rows* and *columns*. The **stores5** database provided with your database server is described in detail in the *IBM Informix Guide to SQL: Reference*. Database fundamentals are introduced in the *IBM Informix Guide to SQL: Tutorial*.

You can use DB-Access menus and screens to create and drop databases; create, modify, and drop tables; and enter, modify, and retrieve information in a database.

Differences Between DB-Access and IBM Informix SQL

This section lists the differences between IBM Informix SQL Version 4.0 or 4.1 and DB-Access Version 5.0.

DB-Access does not provide the following Main Menu options that are available with IBM Informix SQL:

- Form for creating and compiling custom forms
- Report for creating and compiling custom reports
- User-Menu for creating and running custom menus

Also, unlike IBM Informix SQL, DB-Access does not stop running when it encounters an error within an SQL script. Instead, it returns an error message and continues processing the SQL statement(s).

On the other hand, DB-Access provides three features not found in IBM Informix SQL:

- DB-Access offers an additional command-line option, an **-e** flag, that lets you echo the SQL statement(s) in a command file to the screen before you execute the statement(s).
- In addition, unlike IBM Informix SQL, DB-Access lets you specify the initial and next extent sizes of a table through the menu.
- The References option on the INFO Menu provides information on the referential integrity of database tables.

Differences Between IBM Informix OnLine and IBM Informix SE

DB-Access is packaged with both the IBM Informix OnLine and IBM Informix SE database servers. DB-Access differs in the following ways, depending on the database server you are using:

- With IBM Informix OnLine, the TABLE OPTIONS Menu within the Interactive Schema Editor (ISED) has additional options. These options let you specify table extent sizes and the locking mode.
- With IBM Informix OnLine, the Variable-length option within the ISED ADD TYPE Menu lets you specify the VARCHAR, TEXT, and BYTE data types for columns.
- With IBM Informix OnLine, when you display table information using the Status option on the INFO Menu of the SQL or TABLE Menu or use the keywords INFO STATUS in an SQL statement, you do not see audit trail file information because the IBM Informix OnLine logging facility replaces audit trails.
- With IBM Informix SE, you can issue the CHECK TABLE and REPAIR TABLE statements to check and repair the indexes of a table following an abnormal stoppage. These SQL statements call the **bcheck** utility, which is not available with IBM Informix OnLine. (The **tbcheck** utility serves a similar function for OnLine.)

In addition, other SQL statements or keywords may be specific to OnLine or SE. The syntax and usage for all SQL statements that you can use with DB-Access are described in the *IBM Informix Guide to SQL: Reference*.

Preparing to Use DB-Access

This section describes the steps you must follow before you can begin to work with DB-Access. It explains how to set environment variables and use the demonstration database that is provided with DB-Access and acquaints you with conventions for using your terminal. It assumes that either the OnLine or SE database server is installed on your computer according to the installation instructions that come with the database server software.

Setting Environment Variables for DB-Access

You must set the following environment variables before you can properly use DB-Access.

- The `INFORMIXDIR` environment variable specifies the directory where your database server files are installed.
- The `INFORMIXTERM` environment variable specifies whether DB-Access should use the `termcap` file or `terminfo` directory to locate terminal capability information.
- The UNIX environment variable `TERMCAP` or `TERMINFO` must be set so that DB-Access can communicate with your terminal.
- `TERM` is a UNIX environment variable that lets DB-Access recognize and communicate with the kind of terminal you are using.
- `PATH` is a UNIX environment variable that determines the search path so that the shell searches the correct directory for executable DB-Access files.
- If you are using IBM Informix NET or IBM Informix STAR, you might need to set one or more additional environment variables: `DBNETTYPE`, `DBPATH`, or `SQLRM` and `SQLRMDIR`.

You can set these environment variables at the system prompt or in your `.profile` (Bourne shell) or your `.login` or `.cshrc` (C shell) file.

- If you set these variables at the system prompt, you must reassign them every time you log onto the system.
- If you set these variables in your `.profile`, `.login`, or `.cshrc` file, they are assigned automatically every time you log onto the system.



Tip: If you set or modify the environment variables in a file rather than on the command line, log out and then log back in or “source” the file before you begin to work with DB-Access in order to allow the shell to read your entries.

Complete details of how to set and use the most common Informix and UNIX environment variables are provided in the *IBM Informix Guide to SQL: Reference*.

Creating the Demonstration Database

The examples in this manual are based on the **stores5** demonstration database. The structure, contents, and relationship of the nine tables that make up the **stores5** database are described in the *IBM Informix Guide to SQL: Reference*. The system tables for the **stores5** database are also described in that book.

If you have not already done so, create the demonstration database as described in the section [“The Demonstration Database” on page 14](#) of the Introduction to this manual.

Finding Your Way Around DB-Access

Before you can start working with DB-Access, be sure that

1. Your computer is up and running.
2. Your database server product has been correctly installed on your system.
3. IBM Informix OnLine is online, if you are using that server.
4. The operating system prompt appears on your screen.
5. Your environment variables are properly set for DB-Access.

This section explains how to use your terminal and how to invoke and begin working with DB-Access.

How to Use Your Terminal

Your terminal keyboard has some special keys that you use to give instructions to DB-Access. Before you begin using DB-Access, locate the following keys:

- ARROWS** The arrow keys are usually found at the lower right of your keyboard.
- The [↑] key moves the cursor up one line. If your terminal does not have a [↑] key, use the CTRL-K key.
- The [↓] key moves the cursor down one line. If your terminal does not have a [↓] key, use the CTRL-J key.
- The [←] key moves the cursor back one position on the screen. If your terminal does not have a [←] key, use the CTRL-H key.
- The [→] key moves the cursor forward one position on the screen. If your terminal does not have a [→] key, use the CTRL-L key.
- BACKSPACE** The BACKSPACE key may be marked with a left-pointing arrow. It is usually located at the top right of the keyboard.
- CONTROL** The CONTROL key is often labeled CTRL or CNTRL and is usually located at the left of the keyboard. This manual refers to it as CTRL.
- On some systems, you use CTRL-C to cancel or abort a program or to leave a current menu and return to the menu one level above. This manual refers to CTRL-C as the Interrupt key.
- DELETE** The DELETE key is sometimes labeled RUBOUT, CANCEL, or DEL.
- On most systems, you use the DELETE key to cancel or abort a program or to leave a current menu and return to the menu one level above. This manual refers to the DELETE key as the DEL key or the Interrupt key.

ESCAPE	The ESCAPE key may be labeled ESC. It is most often found in the upper left corner of your keyboard.
RETURN	The RETURN key may be labeled RETURN or NEWLINE or it might be indicated with just a right-angle arrow. It is located on the right side of the keyboard.
SPACEBAR	The SPACEBAR is generally unlabeled.

Invoking DB-Access

You invoke DB-Access by entering various command-line options at the operating system prompt. These options, illustrated in this section, let you perform the following actions:

- Display just the software version number.
- Invoke DB-Access and display the Main Menu.
- Invoke DB-Access and select a database.
- Go directly to a submenu.
- Go directly to a submenu and submenu option.
- Go directly to and execute a command file.
- Echo to the screen all input that is read from a file.
- Automatically check your SQL statements for ANSI compatibility.

Unlike using the menu system, when you exit from a submenu or submenu and option that you have specified from the command line, you do not go back to the next higher menu level. Instead, you exit directly to the operating system command line.

The following diagrams show the syntax for the various DB-Access command-line options. You can enter an option preceded by a hyphen in uppercase or lowercase letters. For information on how to read the diagrams, see the section [“Command-Line Conventions”](#) on page 8 of the Introduction.

Displaying the Software Version Number

You can display the version number for DB-Access without invoking the utility.

```
dbaccess -v
```

-v or -V displays the name and version number of the DB-Access software that has been installed.

DB-Access takes no further action after displaying the version number on the screen. Enter other commands described in this section to invoke and work with the utility.

You cannot use any other `dbaccess` options with **-v** or **-V**.

This is an example of what you see when you use the **-v** command-line option:

```
DB-Access Version 5.00.UC1
```

Calling up the DB-Access Main Menu

You can invoke DB-Access and display the Main Menu.

```
dbaccess
```

To begin working with DB-Access, simply enter `dbaccess` at the operating system prompt. This invokes DB-Access and displays the Main Menu.

```
DB-Access: [Query-Language] Database Table Exit
Use IBM Informix Structured Query Language.

-----Press CTRL-W for Help -----
```

You can select various options from the Main Menu and work with DB-Access as described in this manual. See the section [“The DB-Access Main Menu” on page 1-19](#).

If you enter `dbaccess` and DB-Access does not display the Main Menu, you might see the following message instead:

```
Unknown terminal type.
```

This message means that the DB-Access program does not recognize the type of terminal you are using. DB-Access might not be installed correctly or your `TERM`, `TERMCAP`, or `TERMINFO` environment variable could be improperly set. Ask your system administrator for assistance or review the section [“Preparing to Use DB-Access” on page 1-6](#).

If you do not include an option in the `dbaccess` command line, you must use the menu options to select a current database and move around the DB-Access menu system. The rest of this section describes command-line options that you can use to perform certain operations without going through the Main Menu.

Invoking DB-Access and Selecting a Database

You can invoke DB-Access, display the Main Menu, and select a database as the current database.

```
dbaccess _____ database _____|
```

database specifies the name of the database you want to work with.

You can use the **database** option with other `dbaccess` command-line options.

When you invoke DB-Access from the command line, you can include the name of an existing database and make it the current database. For example, if you want to make the **skink** database the current database, enter the following command:

```
dbaccess skink
```

This option invokes DB-Access and displays the Main Menu with the database name included on the fourth line:

```
DB-Access: [Query-Language] Database Table Exit
Use IBM Informix Structured Query Language.

----- skink -----Press CTRL-W for Help -----
```

You can now begin working with the DB-Access menus and screens.

If you have IBM Informix STAR installed and running, you can specify a database on another IBM Informix OnLine database server by including the server name with the database name on the command line. Either of the following commands selects the **kiwi** database on the **pavlova** database server as the current database:

```
dbaccess kiwi@pavlova
dbaccess //pavlova/kiwi
```

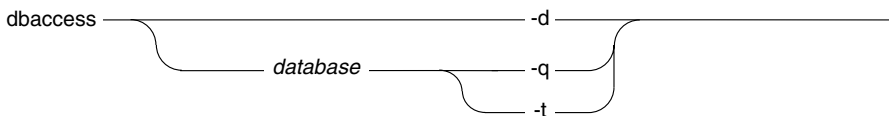
If you have IBM Informix NET installed and running, you can specify a database on another IBM Informix SE database server by including the server name and directory path with the database name on the command line. Enter the following command to select the **fiji** database in the **islands** directory on the **southsea** database server:

```
dbaccess //southsea/islands/fiji
```

For more information on how to select or name a database, refer to the sections “Database Name” and “Identifier” in the *IBM Informix Guide to SQL: Reference*.

Going to a Submenu

You can include a command-line option to invoke DB-Access and select an option from the Main Menu.

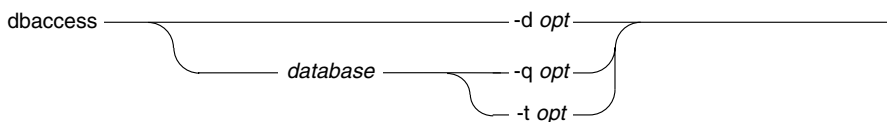


- | | |
|-----------------|---|
| <i>database</i> | specifies the name of the current database. |
| -d | goes directly to the DATABASE Menu. |
| -q | goes directly to the SQL (query-language) Menu. |
| -t | goes directly to the TABLE Menu. |

If you do not include a database name before a **-q** or **-t** menu option, you first have to choose a current database from the SELECT DATABASE screen. Then the requested menu appears.

Going to a Submenu and Submenu Option

For the **-q**, **-d**, and **-t** options, you can follow the menu option with a pattern of keyboard commands. This lets you not only invoke DB-Access and select an option from the Main Menu but also go to a screen or menu from the specified submenu.



database specifies the name of the current database.

-d goes directly to the DATABASE Menu.

-q goes directly to the SQL (query-language) Menu.

-t goes directly to the TABLE Menu.

opt selects an option on the specified menu.

DB-Access accepts the following command-line options (listed in alphabetical order).

DATABASE Menu Options

-dc chooses the Create option on the DATABASE Menu.

-dd chooses the Drop option on the DATABASE Menu.

-ds chooses the Select option on the DATABASE Menu.

SQL Menu Options

- qc chooses the Choose option on the SQL Menu.
- qd chooses the Drop option on the SQL Menu.
- qi chooses the Info option on the SQL Menu.
- qm chooses the Modify option on the SQL Menu.
- qn chooses the New option on the SQL Menu.
- qs chooses the Save option on the SQL Menu.
- qu chooses the Use-editor option on the SQL Menu.

TABLE Menu Options:

- ta chooses the Alter option on the TABLE Menu.
- tc chooses the Create option on the TABLE Menu.
- td chooses the Drop option on the TABLE Menu.
- ti chooses the Info option on the TABLE Menu.

You cannot go directly to the Run or Output option on the SQL Menu. If you try, you see the following error message:

```
823: There are no statements to run.
```

When you select the Modify option on the SQL Menu, you must first select a command file to modify from the CHOOSE Menu. Then the MODIFY screen appears with the text displayed.

If you do not include a database name before a **-q opt** or **-t opt** menu option, you first have to choose a current database on the SELECT DATABASE screen. The menu and/or screen you selected then appears.

The following command takes you to the Drop option on the DB-Access DATABASE Menu where you can drop a database:

```
dbaccess -dd
```

The following command takes you to the Info option on the DB-Access SQL Menu where you can get information on the structure of tables in the **skink** database:

```
dbaccess skink -qi
```

The following command takes you to the Create option on the DB-Access TABLE Menu where you can use the schema editor to create a table in the **skink** database:

```
dbaccess skink -tc
```

Executing a Command File

When you invoke DB-Access from the command line, you can specify the current database and execute a file containing one or more SQL statements.

```
dbaccess database filename
```

database specifies the name of the current database.

- indicates that you are not specifying a database in the command line because one is specified in a DATABASE statement in the command file.

filename specifies the name of the command file or SQL script file that you want to execute. (You can include the extension **.sql**, which is implied if omitted).

Enter the following command to execute the SQL statements in the **scales.sql** file on the **skink** database:

```
dbaccess skink scales
```

Enter the following command to execute the SQL statements in the **scales.sql** file on the database specified in that file:

```
dbaccess - scales.sql
```

Echoing File Input to the Screen

You can use the *echo* option to display on the screen all the commands that are executed in a command file as well as the output from the SQL statement(s) in the command file.

```
dbaccess -e - database filename
```

-e echoes to the screen all input that is read from a specified file.

database specifies the name of the database you want to select as the current database.

- indicates that you are not specifying a database in the command line because one is specified in an SQL statement in the command file.

filename specifies the name of the command file whose lines are echoed to the screen. You can include the extension **.sql**.

The **-e** option must precede all other options in the command line. You must specify a filename in the command.

Each line that is read from the specified file (with or without a specified database) is echoed to the screen. For example, say the file **froggie.sql** contains the following SQL statement:

```
CREATE DATABASE newt
```

The following command:

```
dbaccess - froggie.sql
```

prints the following message on the screen:

```
Database created.
```

However, the following command, with the **-e** option:

```
dbaccess -e - froggie.sql
```



prints the following lines on the screen:

```
CREATE DATABASE newt
Database created.
```

Tip: If DB-Access is invoked in interactive mode, echoing does not take place.

Checking for ANSI Compliance

You can check your SQL statements for compliance with ANSI standards by including the **-ansi** parameter or setting the DBANSIWARN environment variable.

```
dbaccess _____ -ansi _____ |
```

-ansi causes DB-Access to generate a warning whenever it encounters an Informix extension to ANSI syntax.

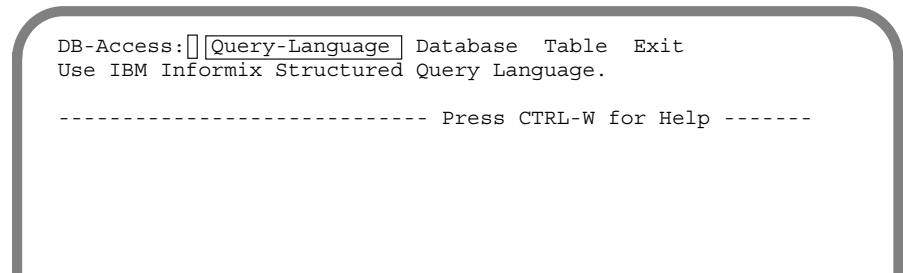
You can use the **-ansi** option with other `dbaccess` options. If DBANSIWARN is set, you do not need to specify **-ansi** on the DB-Access command line.

For more information on ANSI-compliant databases, refer to the *IBM Informix Guide to SQL: Reference* and the *IBM Informix Guide to SQL: Tutorial*.

The DB-Access Main Menu

When you start up DB-Access, the DB-Access Main Menu appears, as shown in [Figure 1-1](#).

Figure 1-1
The DB-Access Main Menu



The DB-Access Main Menu displays the following four options:

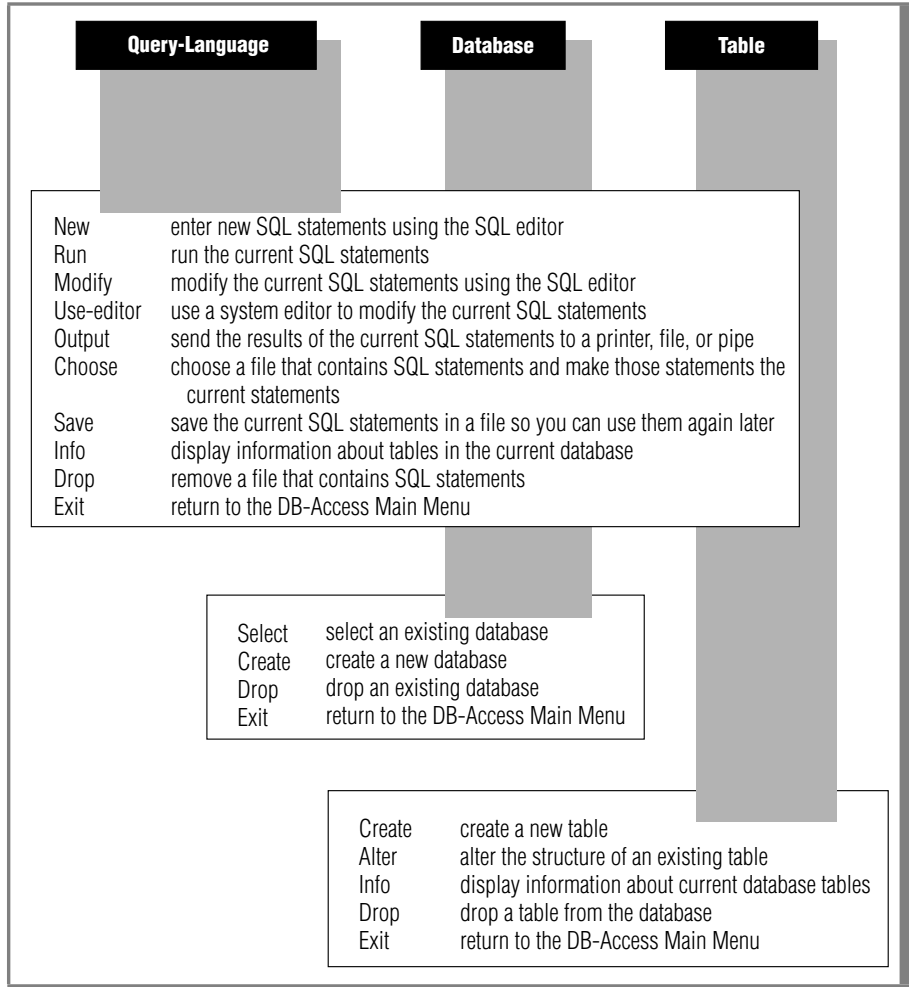
- | | |
|--------------------|---|
| Query-
Language | displays the SQL Menu that lets you work with the SQL query language. Select this option to enter and run SQL statements. |
| Database | displays the DATABASE Menu. Choose this option to select, create, or drop a database. |
| Table | displays the TABLE Menu. Select this option to create, modify, or drop a table in the current database. |
| Exit | exits DB-Access and returns you to the operating system. |

The first three options are described in detail in separate chapters of this manual.

The DB-Access Menu Structure

The Query-Language, Database, and Table options on the DB-Access Main Menu each call a submenu that displays additional options. Figure 1-2 illustrates the DB-Access menu structure or *hierarchy*.

Figure 1-2
The DB-Access menu hierarchy



Using DB-Access Menus and Screens

The DB-Access menu system uses these kinds of screens:

- A menu screen (like the DB-Access Main Menu)
- A text-entry screen
- A series of HELP screens

The following sections summarize the behavior of these screens. Detailed information about the uses of the various DB-Access menus and screens is provided in [Chapter 2](#) through [Chapter 5](#).

Using Menu Screens

The DB-Access menu screen is a ring menu that provides you with a set of choices. The menu screen header usually shows the following kinds of information:

- The top line of a menu screen lists your options. One option is always highlighted and is the *current* option. Screen examples in this manual illustrate a highlighted option by enclosing it in a box.

The current option appears on the screen in reverse video (dark letters on a light background) if your terminal can display reverse video, or else it is surrounded by angle brackets < > . When a menu screen is displayed, the cursor is somewhere on this first line.

- The second line, known as the *message line*, briefly tells you what you will be able to do if you select the highlighted option.

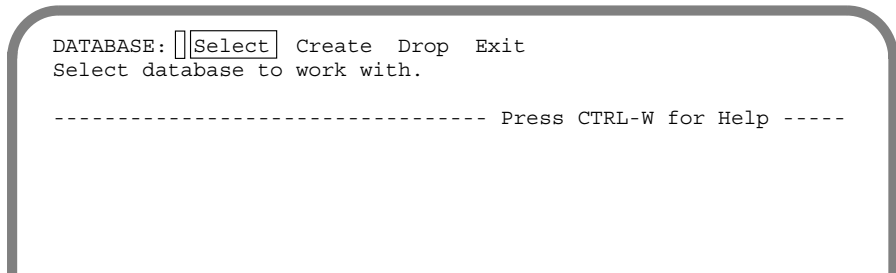
Each time you press the SPACEBAR or the left and right ARROW keys, the highlight moves to the next option, and the description changes. If you cannot remember what an option does, just highlight it and read the description on this line.

- The third line is blank.
- The fourth line displays the name of the current database, if one has been selected, and also reminds you that you can always:

Press CTRL-W for Help

The DATABASE Menu, shown in [Figure 1-3](#), is an example of a menu screen.

Figure 1-3
A sample menu screen



Selecting an Option

In [Figure 1-3](#), the Select option is highlighted, indicating that it is the current option. Press RETURN to choose that option. You can select other menu options in either of two ways:

- Use the SPACEBAR or the left and right ARROW keys to highlight the option you want and then press RETURN.
- Simply type the first letter of the option you want to select; case is not important. For instance, you can type c or C to select the Create option on the DATABASE Menu.

On some menus, two options might start with the same letter. To distinguish between the options, you select one by typing the first *capitalized* letter in the option name. For example, the TABLE OPTIONS Menu that appears if you are using DB-Access with IBM Informix OnLine has both an eXtent option and an Exit option. To select the eXtent option, type x or X; type e or E to exit.

DB-Access displays the screen for the menu option you have selected.

Leaving a Menu Screen

Every menu has an Exit option. When you want to leave a menu screen, type e or E for Exit. DB-Access then displays the previous menu or screen. (If you exit from the Main Menu, you return to the operating system.)

Getting Help

Press CTRL-W to display a HELP screen that provides information about the highlighted menu option. When you indicate that you have finished viewing the HELP screen text, DB-Access redisplay the menu you were working with before you asked for help. For details, see [“Using the HELP Screen” on page 1-25](#).

Using Text-Entry Screens

On a DB-Access text-entry screen, you enter or select text instead of choosing a menu option. The screen header usually shows the following kinds of information:

- The top line of a text-entry screen displays the screen name followed by double angle brackets (>>) and the cursor.
- The second line briefly tells you what to do on the text-entry screen.
- The third line is blank.
- The fourth line displays the name of the current database (if one has been selected) and the message:

Press CTRL-W for Help.

The SELECT DATABASE screen, shown in [Figure 1-4](#), is an example of a text-entry screen. It appears when you choose the Select option on the DATABASE Menu.

Figure 1-4

A sample text-entry screen

```

SELECT DATABASE >> 
Select a database with the Arrow keys, or enter a name, then
press Return.

----- Press CTRL-W for Help -----


stores5

```

Entering Text on the Screen

You might be required to enter text in the first line of the screen header or below the broken line, depending on the operation.

Whatever you type in the header of a text-entry screen appears after the double angle brackets at the top of the screen. Press RETURN when you have finished typing, and DB-Access displays the next screen or takes other appropriate action.

The SELECT DATABASE screen and some others give you the option of selecting an item from a list on the lower part of the screen instead of typing your selection. Use the ARROW keys to highlight the item you want and then press RETURN. DB-Access displays the next screen or takes other appropriate action.

Other text-entry screens, such as some of those selected from the SQL Menu, might require you to enter text below the screen header. These types of screens involve the use of some kind of text editor. See [“Using a Text Editor” on page 1-26](#).

Leaving a Text-Entry Screen

Text-entry screens do not have an Exit option. When you want to leave a text-entry screen, press the Interrupt key (usually DEL or CTRL-C) and DB-Access redisplay the previous menu or screen.

Getting Help

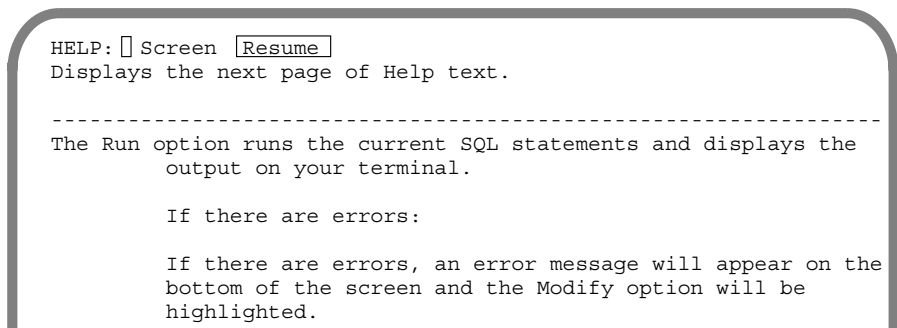
Press CTRL-W to display a HELP screen that provides information about the operation of the current screen. When you indicate that you have finished reading the HELP screen text, DB-Access redisplay the screen you were working with before you asked for help. See the next section, [“Using the HELP Screen” on page 1-25](#), for details.

Using the HELP Screen

Every DB-Access menu and screen has a HELP screen associated with it. The HELP screen contains information about your current options and suggests appropriate actions. Whenever you want information about what to do next, press CTRL-W, and DB-Access displays a HELP screen.

Figure 1-5 shows some of the text that appears when you press CTRL-W for information about the Run option on the SQL Menu.

Figure 1-5
A partial HELP screen with text



The header of a HELP screen provides the following information:

- The first line lists your options.
- The second line briefly tells you about the highlighted option.

Two options let you page through the help text or resume your previous actions.

- Select the Screen option to display the next page of the help text.
- Select the Resume option to return to the menu or screen you were working on.

If there is more than one page of help text, the Screen option is highlighted. Press RETURN to view the next screen. To select the Resume option, highlight it with the SPACEBAR or the right arrow key and then press RETURN, or type r or R.

If there is only one page of help text, the Resume option is highlighted, and you need only press RETURN to resume your work.

See [Appendix B, “How to Read Online Help,”](#) for an illustration of how to read the syntax diagrams that appear when you request online help for creating, modifying, or editing an SQL statement.

Using a Text Editor

You need a text editor to prepare and modify SQL command files when you select the Query-Language option on the Main Menu. DB-Access provides two different ways to edit the text of SQL statements before you run them: an *SQL editor* and the *system editor*.

Using the SQL Editor

The DB-Access utility includes an internal text editor called the SQL editor. When you select the New or Modify option on the SQL Menu and enter text on the screen, you are using the SQL editor.

The screen that appears when you select the New option (shown in [Figure 1-6](#)) or the Modify option indicates the various editing keys you can use.

Figure 1-6
SQL statement text-entry screen and SQL editor keys

```
NEW:  ESC      = Done editing          CTRL-A = Typeover/Insert
      CTRL-R = Redraw                  CTRL-X = Delete character
      CTRL-D = Delete rest of line

----- stores5 ----- Press CTRL-W for Help -----
□
```

Note the position of the cursor, indicating that you enter text on the screen below the header. If you make a mistake while using the SQL editor, move the cursor with the RETURN and arrow keys and retype the text as needed.

Use the editing keys listed at the top of a NEW or MODIFY text-entry screen to perform special functions when you enter SQL statements.

CTRL-A	Switches between <i>insert</i> mode and <i>typeover</i> mode. You are automatically in typeover mode when you start to use the SQL editor. In insert mode, the text beneath the cursor shifts to the right as you enter new characters. In typeover mode, characters you enter replace the text beneath the cursor.
CTRL-D	Deletes characters from the current cursor position through the end of the line.
CTRL-R	Redraws the current screen. Use it when you receive an electronic message or some other interference that makes it difficult to read the SQL statement text that you have entered.
CTRL-X	Deletes a character that appears beneath the cursor.
ESC	Returns you to the SQL Menu when you have finished entering or editing the SQL statement. You can then run or modify the statement or select the system editor for more extensive editing.

The SQL editor does not display more than 80 characters on a line and does not wrap lines.

- If you choose an existing command file in which the characters in a line extend beyond the 80th column, DB-Access displays a percent sign (%) in the 80th column to indicate an overflow. You will not see all the characters beyond the percent sign, but the statement should run correctly.
- If you type in characters in a new command file so that a line extends beyond the 80th column, DB-Access overwrites all the characters in the 80th column. You will not see the overflow, and the statement will not run correctly.

Although DB-Access might read and compile characters beyond the percent sign, it is difficult to work with text that is not visible. Therefore, you are advised to press RETURN at a logical place in the first 80 characters of each line so that the full text appears on the screen. If you need to type in a quoted character string that exceeds 80 characters, such as an INSERT into a long CHAR column, use a system editor instead of the SQL editor.

When you use the SQL editor, you can enter as many lines of text as you need. You are not limited by the size of the screen, although you may be limited by the memory constraints of your system or the maximum SQL statement size of 64 kilobytes.

If you insert more lines than one screen can hold, the SQL editor scrolls down the page with the additional text. The beginning and ending line numbers of the current page are displayed on the fourth line of the text-entry screen, as shown in [Figure 1-7](#)

Figure 1-7
SQL statement text-entry screen with scrolling

```
NEW:  ESC      = Done editing          CTRL-A = Typeover/Insert
      CTRL-R = Redraw                  CTRL-X = Delete character
      CTRL-D = Delete rest of line

-- 3 to 20 of 20 --- stores5 ----- Press CTRL-W for Help -----
□
```

When you finish entering text with the SQL editor and return to the SQL Menu, you can select the Save option to save your command file for future editing or execution. To retrieve a saved command file that you want to modify, select the Choose option. These SQL Menu options are illustrated and described in [Chapter 3](#).

If you decide you would rather work with the system editor, press ESC to return to the SQL Menu and then select the Use-editor option.

Using a System Editor

When you know that you will be entering or modifying a long SQL statement or series of statements, you might prefer the flexibility and familiarity of a system editor to the SQL editor. Select the Use-editor option from the SQL Menu to use the system editor.

The first time in a session that you select the Use-editor option, DB-Access might display a USE-EDITOR screen like the one shown in [Figure 1-8](#).

Figure 1-8

Sample system editor screen for entering and modifying SQL statements

```

USE-EDITOR >>vi █
Enter editor name. (RETURN only for default editor)

----- stores5 ----- Press CTRL-W for Help -----

```

If you have already specified an editor in this session, or if the DBEDIT environment variable has been set, DB-Access calls up that editor immediately and does not display the USE-EDITOR screen.

DB-Access assumes a default operating system editor. Common UNIX system editors are **vi**, as shown in [Figure 1-8](#), and **ex**.

You can specify a different default text editor by setting the DBEDIT environment variable. See the description of DBEDIT in the *IBM Informix Guide to SQL: Reference* for information on how to specify a system editor.

Press RETURN to select the default editor displayed in the first line of the USE-EDITOR screen. Or, type in the name of a different editor that has been set for your system and then press RETURN.

DB-Access calls the editor you specify and creates a temporary file for your text. You can then modify or enter new SQL statements and exit the text file, following the standard rules for the system editor.

Select the Save option on the SQL Menu to save your text file for future editing or execution. Select the Choose option to retrieve a saved text file that you want to edit or run. The options on the SQL Menu are illustrated and described in [Chapter 3, "The Query-Language Menu Option."](#)

Creating Databases and Tables

In This Chapter	2-3
Working with a Database.	2-4
Selecting a Database	2-4
Creating a Database	2-5
Dropping a Database	2-5
Working with a Table	2-5
Creating a Table	2-6
Modifying a Table	2-6
Dropping a Table	2-6
Getting Table Information	2-6
The DB-Access Facilities	2-7
Using the Interactive Schema Editor	2-7
Using SQL Statements	2-9
Using SQL with DB-Access	2-9
List of Available SQL Statements	2-10
Method 1: Using the Menu Options	2-12
How To Create a Database	2-12
Leaving the Menu	2-14
Where System Files Are Stored	2-14
How To Create a Table in the Database	2-15
Calling up the Table Schema	2-15
Naming the New Table.	2-16
Building the Table Schema	2-16
Naming the Columns	2-17
Defining the Data Type of a Column	2-17
Indexing a Column	2-19

Allowing Null Values	2-20
Defining the Next Column.	2-21
Correcting Mistakes in the Schema	2-21
Exiting the Schema Editor	2-23
Where Table Information Is Stored	2-24
How To Modify a Table in the Database	2-24
Using the Alter Option	2-25
Adding a Column to a Table	2-25
Modifying a Column in a Table	2-26
Dropping a Column from a Table	2-27
How To Get Table Information	2-28
How to Drop a Table	2-28
How to Drop a Database	2-30
Method 2: Using SQL	2-31
How To Create a Database	2-32
How to Create a Table	2-33
Assigning Column Constraints	2-34
Creating an Index on a Table	2-34
How to Modify a Table	2-34
Adding a Column to a Table	2-35
Modifying a Column in a Table	2-35
Dropping a Column from a Table.	2-36
Renaming a Column in a Table	2-36
Adding and Dropping Constraints	2-37
How To Get Table Information	2-37
How to Drop a Table	2-38
How to Drop a Database	2-39

In This Chapter

This chapter describes two different methods for creating and dropping a database and for creating, altering, and dropping the tables in a database. It also shows three ways to get information about the structure of the tables in a database and how to query on the contents of a table.

The following two DB-Access facilities are introduced in this chapter:

- The Interactive Schema Editor (ISED)
- The Informix implementation of Structured Query Language (SQL)

You can perform relational database management tasks with these facilities. Follow the instructions in this chapter to learn both ways to perform the following actions:

- Create a database.
- Create a table in that database.
- Modify the new table.
- Get information on the structure of your new table.
- Drop the new table.
- Drop the new database.

Tip: You should first become familiar with the screen and terminal concepts and menu hierarchy described in [Chapter 1, “Working with DB-Access,”](#) and make sure you are properly set up to use DB-Access before you continue with this chapter.



Working with a Database

This section provides an overview of the two methods for working with a database on DB-Access. Full details are provided in other chapters of this manual and in other publications, as follows:

- [Chapter 3, “The Query-Language Menu Option,”](#) provides information on how to use the options on the SQL Menu that appears when you select the Query-Language option on the DB-Access Main Menu. Use the Query-Language option to issue SQL statements that select, create, and drop a database.
- [Chapter 4, “The Database Menu Option,”](#) describes how to use the options on the DATABASE Menu that appears when you select the Database option on the DB-Access Main Menu.
- The *IBM Informix Guide to SQL: Reference* contains the complete syntax and usage of all SQL statements (such as DATABASE, CREATE DATABASE, DROP DATABASE, and others) that operate on a database. It also describes the structure and contents of the **stores5** demonstration database, and the details of the Informix *system catalog*, which describes the structure of the **stores5** database.
- See the *IBM Informix Guide to SQL: Tutorial* for an overview of database fundamentals.

Selecting a Database

The database you are going to work with is called the *current* database. With DB-Access, you can select a current database in two ways:

- Use the Select option on the DATABASE Menu.
- Run the SQL statement DATABASE from the SQL Menu.

Creating a Database

You can select an existing database or create your own. With DB-Access, you can create a database in two ways:

- Use the Create option on the DATABASE Menu.
- Run the SQL statement `CREATE DATABASE` from the SQL Menu.

Dropping a Database

With DB-Access, you can drop a database in two ways:

- Use the Drop option on the DATABASE Menu.
- Run the SQL statement `DROP DATABASE` from the SQL Menu.

Working with a Table

This section provides an overview of the ways you can build and modify a table in a relational database. Full details are provided in other chapters of this manual and in other publications, as follows:

- [Chapter 3, “The Query-Language Menu Option,”](#) provides information on how to use the options on the SQL Menu that appears when you select the Query-Language option on the DB-Access Main Menu. Use this option to issue SQL statements that create, alter, and drop tables.
- [Chapter 5, “The Table Menu Option,”](#) describes how to use the options on the TABLE Menu that appears when you select the Table option on the DB-Access Main Menu.
- The *IBM Informix Guide to SQL: Reference* contains the complete syntax and usage of all SQL statements (such as `CREATE TABLE`, `ALTER TABLE`, `DROP TABLE`, and others) that operate on a table. It also provides a complete discussion of data types that can be assigned to the columns in a table.

Creating a Table

With DB-Access, you can create a table in either of two ways:

- Use the DB-Access schema editor from the TABLE Menu.
- Run the SQL statement CREATE TABLE from the SQL Menu.

Modifying a Table

You might need to make changes to the structure of an existing table or one you are creating. You can do this in three ways:

- Use the Alter option on the TABLE Menu to change an existing table.
- Use the Modify option on the CREATE TABLE Menu to change a table that you are creating.
- Run the SQL statement ALTER TABLE from the SQL Menu to change an existing table.

Dropping a Table

With DB-Access, you can drop a table from a database in either of two ways:

- Use the Drop option on the TABLE Menu.
- Run the SQL statement DROP TABLE from the SQL Menu.

Getting Table Information

You can get information about the *structure* and *characteristics* of a table in three ways:

- Use the Info option on the TABLE Menu.
- Use the Info option on the SQL Menu.
- Run the SQL statement INFO from the SQL Menu.

To query on the *data* that is in the tables in a relational database, you must issue a SELECT statement from the SQL editor. For detailed information on how to create and use the SELECT statement, see the *IBM Informix Guide to SQL: Tutorial*. In addition, a number of command files included with DB-Access contain sample SELECT statements with which you can experiment. These command files are discussed later in this chapter and illustrated in [Appendix A, “Sample Command Files.”](#)

The DB-Access Facilities

Special DB-Access facilities let you perform the following actions:

- Create and modify tables in the database using the menus provided with the Interactive Schema Editor.
- Enter, modify, and retrieve database information and modify the structure of the database using Structured Query Language.

This section introduces these features. Later sections of this chapter walk you through the steps for both methods of creating and modifying databases and tables.

Using the Interactive Schema Editor

A database is made up of one or more tables. A table *schema* is essentially a blueprint that defines the structure (columns) of a table in a database. You use the Interactive Schema Editor to create the schema for each table in a database.

Select the Table option on the DB-Access Main Menu. From this menu, you can use the Interactive Schema Editor with the CREATE TABLE Menu, as shown in [Figure 2-1](#). The various menu options and screens involved in creating and modifying a table schema are described in detail in [Chapter 5, “The Table Menu Option.”](#)

Figure 2-1

The CREATE TABLE Menu in the Interactive Schema Editor (IBM Informix OnLine)

```
CREATE TABLE customer:  Add Modify Drop Screen Table_options Exit
Adds columns to the table above the line with the highlight.

-- Page 1 of 1 -- mydata ----- Press CTRL-W for Help -----

Column Name                Type                Length  Index  Nulls
-----

```

You build the schema for a table by adding columns to it. For each column, specify the name, data type, length, type of index, and whether the column allows null values. [Figure 2-2](#) shows the finished table schema for the **customer** table in the **stores5** database.

Figure 2-2

The schema for the customer table (IBM Informix OnLine)

```
CREATE TABLE customer:  Add Modify Drop Screen Table_options Exit
Adds columns to the table above the line with the highlight.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help -----

Column Name                Type                Length  Index  Nulls
-----
customer_num              Serial              101     Unique No
fname                     Char                15
lname                     Char                15
company                   Char                20
address1                   Char                20
address2                   Char                20
city                       Char                15
state                     Char                2
zipcode                   Char                5       Dups   Yes
phone                     Char                18

```

The **dbschema** utility provided with your database server enables you to quickly produce an SQL command file that contains the statements required to replicate an entire database or selected table. See either the *IBM Informix SE Administrator's Guide* or the *IBM Informix OnLine Administrator's Guide* for a detailed description of the **dbschema** utility. Also, see the *IBM Informix Guide to SQL: Reference* for information on the structure and contents of the **stores5** database that is provided with DB-Access.

Using SQL Statements

SQL is an English-like, interactive query language that you can use when working with relational databases. The SQL supplied with DB-Access is an enhanced version of the industry-standard SQL developed by IBM. With SQL in DB-Access, you can perform a variety of database management tasks, including:

- Creating and dropping tables
- Entering and deleting data
- Querying a database
- Renaming tables and columns

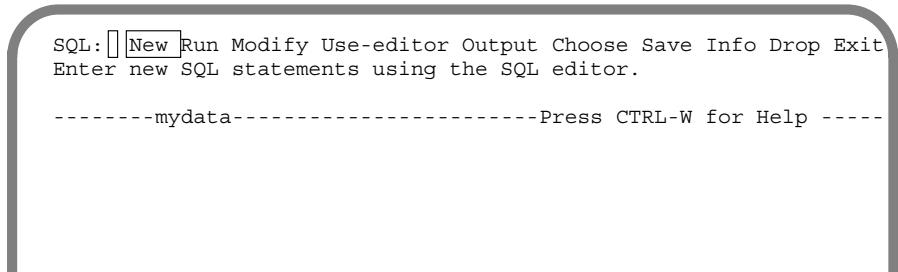
Informix additions to SQL let you load and unload database tables into ASCII text files, change databases, and change the names of tables and columns. Other extensions increase the functionality of standard SQL statements. The uses of SQL and statement syntax conventions are described in detail in the *IBM Informix Guide to SQL: Reference*.

Using SQL with DB-Access

Use SQL with DB-Access to enter one or more SQL statements. A statement is simply an instruction that tells DB-Access what you want to do. For example, to create a table, you use the CREATE TABLE statement; to query a database, you use the SELECT statement.

Select the Query-Language option on the DB-Access Main Menu. From this menu you can use the SQL Menu, shown in [Figure 2-3](#), to enter and run SQL statements. The various menu options and screens used for this purpose are described in detail in [Chapter 3, "The Query-Language Menu Option."](#)

Figure 2-3
The SQL Menu for entering SQL statements



While you are working with DB-Access, SQL remembers the last sequence of statements, if any, with which you worked. These statements are called the current statements. Through the SQL Menu, you can run these statements, save them to a file for later use, send their results to a printer or file, or modify them with an editor.

You can check your SQL statements for ANSI compatibility in two ways:

- When the `DBANSIWARN` environment variable is set, DB-Access generates a warning whenever it encounters an Informix extension to ANSI standard syntax.
- You can use the `-ansi` flag to check SQL statements for ANSI compatibility when you invoke DB-Access on the command line.

List of Available SQL Statements

Figure 2-4 lists all the SQL statements that you can run from the SQL Menu of the DB-Access utility. For the complete syntax and full description of how to use most of these SQL statements, see the *IBM Informix Guide to SQL: Reference*.

Figure 2-4
SQL statements you can use with DB-Access

- | | |
|---------------|--------------|
| ■ ALTER INDEX | ■ INFO |
| ■ ALTER TABLE | ■ LOAD |
| ■ BEGIN WORK | ■ LOCK TABLE |
| ■ CHECK TABLE | ■ OUTPUT |

- CLOSE DATABASE
- COMMIT WORK
- CREATE AUDIT
- CREATE DATABASE
- CREATE INDEX
- CREATE PROCEDURE
- CREATE PROCEDURE FROM
- CREATE SCHEMA
- CREATE SYNONYM
- CREATE TABLE
- CREATE VIEW
- DATABASE
- DELETE
- DROP AUDIT
- DROP DATABASE
- DROP INDEX
- DROP SYNONYM
- DROP TABLE
- DROP VIEW
- EXECUTE PROCEDURE
- GRANT
- RECOVER TABLE
- RELEASE
- RENAME COLUMN
- RENAME TABLE
- REPAIR TABLE
- REVOKE
- ROLLBACK WORK
- ROLLFORWARD DATABASE
- SELECT
- SET CONSTRAINTS
- SET DEBUG FILE TO
- SET EXPLAIN
- SET ISOLATION
- SET LOCK MODE
- SET LOG
- SET MOUNTING TIMEOUT
- START DATABASE
- UNLOAD
- UNLOCK TABLE
- UPDATE
- UPDATE STATISTICS

Some of these SQL statements are supported only on IBM Informix SE or on IBM Informix OnLine. In addition, certain SQL statements function differently, depending on which server you are using.

For complete information on SQL, see *IBM Informix Guide to SQL: Reference* and *IBM Informix Guide to SQL: Tutorial*.

Method 1: Using the Menu Options

The easiest way to create a database or table is to select the Database or Table option, respectively, on the DB-Access Main Menu. Menus are provided at each step, and you are prompted to enter each piece of necessary information. HELP menus also are available for added assistance.

This chapter walks you through the steps needed to create and drop a database and to create, modify, and drop a table in that database using the DB-Access menu system. Full details of these menu options are provided in [Chapter 4, “The Database Menu Option,”](#) and [Chapter 5, “The Table Menu Option.”](#)

How To Create a Database

To create a database, select the Database option on the Main Menu. DB-Access displays the DATABASE Menu.

```
DATABASE:  Create Drop Exit  
Select a database to work with.
```

```
----- Press CTRL-W for Help -----
```

Type `c` or highlight the Create option with an arrow key, as shown in the following screen, and press RETURN.

```

DATABASE:  Select  Create Drop Exit
Create a new database.

----- Press CTRL-W for Help -----

```

The CREATE DATABASE screen appears and prompts you to name the new database.

```

CREATE DATABASE >> 
Enter the name you want to assign to the new database, then press
Return.

----- Press CTRL-W for Help -----

```

Type the database name on the top line (this chapter uses **mydata**) and press RETURN. DB-Access creates a database with that name and automatically makes it the *current* database.

The DATABASE Menu reappears and displays the name of the current database (**mydata**) on the fourth line.

```

DATABASE:  Select Create Drop Exit
Select a database to work with.

----- mydata ----- Press CTRL-W for Help -----

```

See [Chapter 4](#) for guidelines on naming a database, if you decide not to use **mydata**, and for information on naming a database on another server.

Leaving the Menu

Exit the DATABASE Menu by typing `e` or highlighting Exit with an arrow key. The DB-Access Main Menu reappears.

```
DB-Access: [Query-Language] Database Table Exit
Use IBM Informix Structured Query Language.

-----mydata----- Press CTRL-W for Help -----
```

You always have the option of leaving the current menu or screen. On most menus, you do this by selecting the Exit option on the current menu.

In the processes described in this chapter, certain menus or screens do not have an Exit option. You can always leave the current screen or menu and move up a level in the menu hierarchy, without making any choices or changes, by pressing the Interrupt key (usually DEL or CTRL-C).

Where System Files Are Stored

When you create a database, a number of system tables that describe the structure of the database are generated automatically. These system tables make up the system catalog.

IBM Informix OnLine creates the system catalog that contains the data dictionary describing the structure of the new database in a common area of the disk called a *dbspace*. If you do not specify the *dbspace*, IBM Informix OnLine creates the system catalog in the **root** *dbspace*.

With IBM Informix SE, the system tables and data for the new database are placed in a subdirectory of your current directory named for the database. The subdirectory has the extension **.dbs** (for example, **mydata.dbs**).

DB-Access uses the system catalog to keep track of the tables, columns, indexes, views, synonyms, and privileges in each database. The system catalog is described in detail in the *IBM Informix Guide to SQL: Reference*.

How To Create a Table in the Database

Once you have created a database, the next step is to create the tables you want included in the database. DB-Access does not limit the number of tables in a database; the limit is determined by the amount of disk space available on your computer.

The tables you create with DB-Access are placed in the current database. This chapter uses **mydata** as the current database. If **mydata** is not your current database, use the options available with the DATABASE Menu to select or create **mydata** or another current database of your choice. The name of the current database is displayed on the fourth line from the top of the screen.

Calling up the Table Schema

To begin creating a table, select the Table option on the DB-Access Main Menu. The TABLE Menu appears with the Create option highlighted.

```
TABLE: [Create] Alter  Info  Drop  Exit  
Create a new table.
```

```
----- mydata ----- Press CTRL-W for Help -----
```

Naming the New Table

Press RETURN and the CREATE TABLE screen appears, prompting you to name the new table.

```
CREATE TABLE >>   
Enter the table name you wish to create with the schema editor.  
  
----- mydata ----- Press CTRL-W for Help -----
```

Enter a table name (this chapter uses the name **clients**) and press RETURN. (See [Chapter 5](#) for guidelines on naming a table, if you decide not to use **clients**.) DB-Access next displays the CREATE TABLE Menu.

```
CREATE TABLE clients:   Add Modify Drop Screen Table_options Exit  
Adds columns to the table above the line with the highlight.  
  
---- Page 1 of 1 --- mydata ----- Press CTRL-W for Help -----  
  
Column Name                Type                Length  Index Nulls  
  

```

The CREATE TABLE Menu displays the name of the current database and the table that you want to create using the schema editor.

Building the Table Schema

The CREATE TABLE Menu works with the *schema editor* to design the table schema. Whenever you select the Create or Alter option on the TABLE Menu to create or alter a table schema, you use the schema editor.

The CREATE TABLE Menu displays the schema for the table on the bottom portion of the screen. Each horizontal line in the schema represents one column in the table, with the name of the column at the left.

To create a table schema, you define each column in the table, one column at a time. As you define each column, DB-Access prompts you for the information it needs. Most of the prompts are in the form of menus, so you can select the appropriate response quickly and easily. Some of the screens require you to enter something other than a menu selection, such as the name of the column.

As you respond to these prompts, DB-Access puts your information in the schema. You move from left to right across the screen as you define each column, and from top to bottom as you define additional columns.

Naming the Columns

Select the Add option on the CREATE TABLE Menu, and DB-Access displays the ADD NAME screen and prompts you for the column name.

```

ADD NAME >> 
Enter column name. RETURN adds it. INTERRUPT returns to
CREATE/ALTER menu.

---- Page 1 of 1 ---- mydata ----- Press CTRL-W for Help ----

Column Name                Type                Length  Index Nulls
-----

    
```

Enter **customer_num** for the column name and press RETURN.

In selecting column names, you must follow the same guidelines that apply to table names. You must use a different name for each column within a single table so that DB-Access can identify each column.

Defining the Data Type of a Column

You can store many different kinds of data in a table, such as dates, zip codes, names, part numbers, part descriptions, salaries, costs. You need to choose the appropriate *data type* for the kind of data that you intend to store in each column.

The available data types that you can use to name columns in tables are thoroughly described in the *IBM Informix Guide to SQL: Reference*.

After you name a column, the ADD TYPE Menu appears.

```

ADD TYPE clients : Char Number Serial Date Money date-Time Interval...
Permits any combination of letters, numbers, and symbols.

--- Page 1 of 1 ---mydata ----- Press CTRL-W for Help ----

Column Name          Type          Length  Index Nulls
customer_num         
    
```

You are now ready to assign a data type to the **customer_num** column. The **customer_num** column contains the customer number for each store. The SERIAL data type is appropriate for this column because you want to ensure that a unique number is assigned to each customer.

Type *s* to select *Serial* from the menu. DB-Access then displays the ADD STARTING NUMBER screen.

```

ADD STARTING NUMBER >> 
Enter the starting number. RETURN adds it.

---- Page 1 of 1 ---- mydata ----- Press CTRL-W for Help ----

Column Name          Type          Length  Index Nulls
customer_num         Serial         101     Unique No
    
```

DB-Access displays the SERIAL data type under the *Type* heading of the schema and, because you are describing a SERIAL column, asks you for the number DB-Access should start with. If you do not specify a starting point, SERIAL values begin with 1 and new numbers are assigned automatically.

Type *101* to begin the numbering sequence at 101 and then press RETURN. (You would press RETURN to select the default starting number, 1).

Indexing a Column

Creating indexes for the columns in a table helps DB-Access find information more quickly. However, DB-Access can find information in a database even if you do not index any columns. If there is only one table in your database, you probably will not need any indexes until you have entered several hundred rows of data into the table. Too many indexes might have the undesired effect of slowing down changes to the data. As the size of your database increases and your database needs change, you can always add and delete indexes.

A detailed discussion of indexes and indexing strategies appears in the *IBM Informix Guide to SQL: Tutorial*.

After you define the data type of the column, you can define an index for the column. DB-Access displays the ADD INDEX Menu.

```

ADD INDEX      clients :  Yes  No
Specifies that this column will NOT have an index.

---- Page 1 of 1 ---- mydata ----- Press CTRL-W for Help ----

Column Name           Type           Length   Index Nulls
customer_num          Serial         101      

```

The ADD INDEX Menu displays the following two options:

Yes creates an index on the column.

No does not create an index on the column.

If you select the Yes option, DB-Access prompts you to indicate whether the index allows duplicates.



Tip: This option creates a nonclustered, ascending index for the values in the associated database column. To create a different type of index, you must use the SQL statement CREATE INDEX. To place a constraint on the column, use the CREATE TABLE or ALTER TABLE statement.

For the **customer_num** column definition, DB-Access automatically supplies a value for the **Index** column. The word **Unique** appears under the **Index** heading because DB-Access assumes you want to index a column with a SERIAL data type. Also, because it is a SERIAL column, duplicate values should not exist.

Allowing Null Values

After you have defined the index for the column, you can determine whether the column allows null values. DB-Access displays the ADD NULLS Menu.

```
ADD NULLS      clients :  Yes No
Permits null values in this column.

---- Page 1 of 1 ----mydata ----- Press CTRL-W for Help ----

Column Name          Type          Length  Index  Nulls
customer_num         Serial        101     Unique 
```

The ADD NULLS Menu displays the following two options:

Yes allows null values in the column.

No does not allow null values in the column.

For the **customer_num** column definition, DB-Access automatically supplies a value for the **Nulls** column. The word **No** appears under the **Nulls** heading because null values should not be allowed in a SERIAL column.

Defining the Next Column

The ADD NAME Menu appears again, with the first line of the schema completed and the highlight at the left of the next line. DB-Access is ready for you to define the next column.

```

ADD NAME >> 
Enter column name. RETURN adds it. INTERRUPT returns to CREATE ALTER menu.

----- Page 1 of 1 ----- mydata ----- Press CTRL-W for Help -----

Column Name                Type           Length  Index Nulls
customer_num                Serial         101     Unique No


```

Name another column and enter a different data type (only one column in a table can have a SERIAL data type). Continue building the schema.

Correcting Mistakes in the Schema

If you make a mistake when entering information about a column, you can back up and correct it as long as you are still defining the same column (still working on the same line). Use the [←] and [→] keys to move the highlight back and forth under the headings in that line and make your corrections. If you notice a mistake in a column definition after you have moved on to another column, use the Alter option, as described in the section [“How To Modify a Table in the Database”](#) on page 2-24.

After you finish building a schema for the **clients** table, the screen looks something like this:

```
CREATE TABLE clients :  Modify Drop Screen Table_options Exit
Adds columns to the table above the line with the highlight.

---- Page 1 of 1 ---- mydata ----- Press CTRL-W for Help ----

Column Name                Type                Length  Index Nulls
customer_num               Serial              101     Unique No
fname                      Char                15      Yes
lname                      Char                15      Yes
company                    Char                20      Yes
address1                   Char                20      Yes
address2                   Char                20      Yes
city                       Char                15      Yes
state                      Char                2       Yes
zipcode                    Char                5       Dups   Yes
phone                      Char                18      Yes

```

The **clients** table created in this chapter is similar to the **customer** table included in the **stores5** demonstration database.

Exiting the Schema Editor

After you have entered the final piece of information defining the last column (**phone**) in the **clients** table, the schema is complete. Press the Interrupt key and the CREATE TABLE Menu reappears.

Select the Exit option. The EXIT Menu appears with the table schema displayed.

```
EXIT clients:  Discard-new-table
Builds a new table and returns to the Table Menu.

---- Page 1 of 1 ---- mydata ----- Press CTRL-W for Help ----

Column Name          Type          Length  Index Nulls
customer_num         Serial        101     Unique No
fname                Char          15      Yes
lname               Char          15      Yes
company              Char          20      Yes
address1             Char          20      Yes
address2             Char          20      Yes
city                 Char          15      Yes
state                Char          2       Yes
zipcode              Char          5       Dups  Yes
phone                Char          18      Yes
```

The EXIT Menu has the following two options:

Build-new-table builds the table.

Discard-new-table discards the table instructions.

Select the Build-new-table option and **DB-Access** creates the table and returns you to the TABLE Menu.

Where Table Information Is Stored

With IBM Informix OnLine, information and associated indexes for tables you create are stored in a *tblspace*. A *tblspace* is a logical grouping of extents. The creator of a table can specify the initial extent size and next extent size and modify them (using ALTER TABLE) as a table grows. The section “[Setting Table Options](#)” on page 5-27 discusses how to set the extent sizes for a table using the Table-options option on the CREATE TABLE or ALTER TABLE Menu.

With IBM Informix SE, DB-Access builds two system files for each table you create and stores them in the same directory as the system catalog tables. These files have the same (full or partial) name as the table, a unique number starting at 100, and an extension. For example, the two files representing the data and indexes for the **clients** table you created in the **mydata.dbs** database directory would be named **clients110.dat** and **clients110.idx**. (On certain platforms, there is an additional file, **clients110.lok**.)

How To Modify a Table in the Database

Once you have created a table with the schema editor, you might need to modify its structure. This section describes how to use the Alter option from the TABLE Menu to alter a database table. For information on using the ALTER TABLE statement, refer to the *IBM Informix Guide to SQL: Reference*.

Once you set up the schema for a table (even after you have put data into the table), DB-Access allows you to change the way you store your data. Occasionally, making a change of this kind might also result in truncated data. DB-Access always warns you of this and gives you the option of not making the change. (An example of a change that could cause a truncation of data is changing a CHAR column from a length of 20 to 10. DB-Access would truncate the last 10 characters of this column in each row in the table.)

Using the Alter Option

You can use the Alter option on the TABLE Menu to make changes to the structure of a table. Select the Alter option and DB-Access prompts you for the name of the table you wish to modify. Enter the table name and DB-Access displays the ALTER TABLE Menu and accesses the schema editor.

Use the following options on the ALTER TABLE Menu to make your changes to the table schema:

- | | |
|--------|--|
| Add | lets you add one or more columns to the table schema. |
| Modify | lets you modify a column definition by changing one or more of the five schema headings (Column Name, Type, Length, Index, and Nulls). |
| Drop | lets you remove an entire column from the table schema. |

If the table schema is long, it may not all appear on one screen. Use the Screen option on this menu to scroll to the next screen of the schema and locate the column you want to modify.

To alter a table, you must be the owner of the table, have DBA privileges in the current database, or be granted the ALTER privilege on the table.

Adding a Column to a Table

The procedure for adding a column to a table schema is the same as that used to create a table with the schema editor.

1. Select the Alter option on the TABLE Menu. DB-Access asks for the name of the table you want to change. Select the table, and the ALTER TABLE Menu is displayed. Use the arrow keys to move the highlight to the location of the new column.
2. Select the Add option. As before, enter the information needed to create one or more columns. When you finish, press the Interrupt key. DB-Access again displays the ALTER TABLE Menu.
3. Select the Exit option. Then select Build-new-table to make the table addition permanent. DB-Access makes the changes to the database table and returns you to the TABLE Menu.

Modifying a Column in a Table

Use the Alter option to change an existing column definition in the table schema. You can change one or more parts of a column definition, as follows:

- 1.** Select the Alter option on the TABLE Menu. DB-Access asks for the name of the table you want to change. Select the table, and the ALTER TABLE Menu is displayed. Use the arrow keys to move the highlight to the part of the schema you want to change.
- 2.** Select the Modify option. DB-Access displays the appropriate screen for the part of the schema you highlighted. Make the desired changes by entering information on the menu that appears for each part of the column definition.
- 3.** If you want to modify another part of the schema, use the arrow keys to place the highlight. Make menu selections by using the SPACEBAR to highlight the option you want and then press RETURN. The screen changes as you move the highlight. When you finish your changes to the table schema, press RETURN. DB-Access displays the ALTER TABLE Menu.

If your changes decrease the length of a column, the MODIFY ANY-WAY Menu appears, with a warning that you may lose data. Select Yes if you want to continue with the change or No if you do not.

- 4.** Select the Exit option on the ALTER TABLE Menu. Then select Build-new-table on the EXIT Menu to make the changes permanent. DB-Access makes the changes to the database table and returns you to the TABLE Menu.

Dropping a Column from a Table

You can remove any column from the table schema by using the Drop option on the ALTER TABLE Menu.

1. Select the Alter option from the TABLE Menu. DB-Access asks for the name of the table you wish to change. Select the table, and the ALTER TABLE Menu is displayed. Use the arrow keys to move the highlight to the column you want to delete.
2. Select the Drop option and DB-Access displays the REMOVE Menu, with a message reminding you that this change will delete data when you select Build-new-table.

```
REMOVE clients :  Yes  No
Deletes the highlighted column from the table.

---- Page 1 of 1 ---- mydata ----- Press CONTROL-W for Help ----

Column Name          Type          Length  Index Nulls
-----
customer_num        Serial        101     Unique No
fname                Char          15
lname                Char          15
company              Char          20
address1             Char          20
address2             Char          20
city                 Char          15
state                Char          2
zipcode              Char          5       Dups
phone                Char          18

Any data in this column will be lost when you select Exit, Build
```

3. Select the Yes option from the REMOVE Menu if you want to delete the column; select No and the table is untouched. If you select the Yes option, DB-Access removes the column definition from the screen and returns to the ALTER TABLE Menu.
4. Select Exit, then select Build-new-table, and DB-Access drops the column from the table schema. If you decide you do not want to remove the column, select the Discard-new-table option instead, and DB-Access restores the table schema to its former state.

How To Get Table Information

You can get information about the structure and characteristics of your new table through two different menus, as follows:

- Select the Info option on the TABLE Menu that appears when you select the Table option on the DB-Access Main Menu. This menu is described in [Chapter 5, “The Table Menu Option.”](#)
- Select the Info option on the SQL Menu that appears when you select the Query-Language option on the DB-Access Main Menu. This menu is described in [Chapter 3, “The Query-Language Menu Option.”](#)

Both menus let you request and display information about the columns, indexes, privileges, and status of any table in a database for which you have CONNECT privilege.

How to Drop a Table

This section outlines how to drop a table using the menu options available with DB-Access. These menu options are detailed in [Chapter 5](#).

Select the Table option on the DB-Access Main Menu to call up the TABLE Menu.

```
TABLE: [Create] Alter  Info  Drop  Exit
Create a new table.

-----mydata-----Press CTRL-W for Help -----
```

Select the Drop option on the TABLE Menu to drop an existing table schema. DB-Access displays the DROP TABLE screen.

```
DROP TABLE >> 
Enter the table name you wish to drop from the database.

----- mydata ----- Press CTRL-W for Help -----
clients
customer
orders
```

This screen lists the names of tables that exist in the current database. Type the name of the table you want to drop and press RETURN, or highlight the name of the table with the arrow keys and then press RETURN.

You will then be asked to confirm that you want to drop the table. Select Yes if you do and No if you do not.

- If you select Yes, DB-Access drops the table and all its data and returns you to the TABLE Menu.
- If you select No, DB-Access returns you to the TABLE Menu and leaves the specified table and its data intact.

To drop a table, you must be the owner of the table or have DBA privileges in the current database.

How to Drop a Database

This section tells how to drop a database using the menu options available with DB-Access. These menu options are described in detail in [Chapter 4, “The Database Menu Option.”](#)

On the DB-Access Main Menu, select the Database option and press RETURN to call up the DATABASE Menu.

```
DATABASE:  Create Drop Exit
Select database to work with.

----- mydata ----- Press CTRL-W for Help -----
```

Select the Drop option from the DATABASE Menu and the DROP DATABASE screen appears.

```
DROP DATABASE >> 
Enter the name of the database you want to drop.

----- Press CTRL-W for Help -----


stores5
```

Type in the name of the database you want to drop or highlight it with the arrow keys; then press RETURN.

DB-Access displays a special screen that asks for confirmation before it drops the database. The default is No, so you cannot drop a database by mistake. If you want to drop the highlighted database, press Y or use the right arrow key to highlight Yes and press RETURN. DB-Access drops the database and all its tables. It also closes the current database, if one was open.

Method 2: Using SQL

SQL provides a flexible and efficient method for performing the following functions:

- Creating and dropping databases, tables, and indexes
- Selecting a different current database
- Entering and deleting data
- Querying a database
- Sending query results to a file or program
- Displaying information about one or more tables
- Renaming tables and columns
- Checking and repairing tables (IBM Informix SE)
- Loading and unloading data from and to ASCII files
- Granting and revoking database and table privileges

This chapter outlines the steps needed to create and drop a database and to create, modify, and drop a table in that database using SQL statements. It also describes how to use SQL to get information on the structure and contents of a table.

With DB-Access, you enter and run SQL statements through the SQL Menu that appears when you select the Query-Language option on the DB-Access Main Menu.

```
SQL: [New] Run Modify Use-editor Output Choose Save Info Drop Exit
Enter new SQL statements using the SQL editor.

-----mydata-----Press CTRL-W for Help -----
```

Full details of how to create, modify, and run SQL statements are provided in [Chapter 3](#). See the *IBM Informix Guide to SQL: Reference* for information on the syntax and usage of all the SQL statements named in this chapter.

How To Create a Database

Issue the CREATE DATABASE statement to create a database. You must run the CREATE DATABASE statement instead of using the menu options to create any of the following types of databases:

- An ANSI-compliant database (IBM Informix OnLine or IBM Informix SE).
- A database with logging (IBM Informix OnLine or IBM Informix SE).
- An IBM Informix OnLine database stored in a dbspace other than the **root** dbspace.

For example, the following statement creates an ANSI-compliant database with logging on IBM Informix OnLine:

```
CREATE DATABASE lutefisk WITH LOG MODE ANSI
```

The following example creates an ANSI-compliant database with a log file on IBM Informix SE:

```
CREATE DATABASE lutefisk WITH LOG IN "/ole/ufda/lfile" MODE ANSI
```

If you are the OnLine administrator (user **informix**), you can make an IBM Informix OnLine database ANSI-compliant and start logging with the Databases option of the LOGICAL-LOGS Menu in DB-Monitor. (See the *IBM Informix OnLine Administrator's Guide* for details.)

If you are using the IBM Informix SE database server, you can turn on transaction logging and put a database in ANSI-compliant mode with the SQL statement START DATABASE. You must have DBA privilege for the database and there must be no current database.

The START DATABASE statement gives you exclusive access to the database. To make the database available to other authorized users, you must issue the CLOSE DATABASE statement or exit DB-Access.

See the complete syntax and usage for the CREATE DATABASE and START DATABASE statements in the *IBM Informix Guide to SQL: Reference*.

How to Create a Table

Issue the CREATE TABLE statement to create a table in a database. You must use the CREATE TABLE (or ALTER TABLE) statement, not the schema editor, to create a table with the following features:

- Specified storage location (dbspace name) for IBM Informix OnLine databases
- Specified foreign or primary key

Also, when you add a column to a table, you must use the CREATE TABLE (or ALTER TABLE) statement, not the schema editor, to create a column with the following features:

- Unique constraints
- Referential constraints
- Check constraints
- Default values

Just as you do when you use the schema editor to create a table, you can define in your SQL statement the following characteristics for each column in the table: column name, data type, length or starting number, indexes, and whether it allows null values.

Each table in the **stores5** demonstration database was created with a different CREATE TABLE statement. The following statement (found in the **c_calls.sql** command file) creates the **cust_calls** table:

```
CREATE TABLE cust_calls
(
  customer_num      INTEGER,
  call_dtime        DATETIME YEAR TO MINUTE,
  user_id           CHAR(18) DEFAULT USER,
  call_code         CHAR(1),
  call_descr        CHAR(240),
  res_dtime         DATETIME YEAR TO MINUTE,
  res_descr         CHAR(240),
  PRIMARY KEY (customer_num, call_dtime),
  FOREIGN KEY (customer_num) REFERENCES customer (customer_num),
  FOREIGN KEY (call_code) REFERENCES call_type (call_code)
);
```

See the complete syntax and usage for the CREATE TABLE statement in the *IBM Informix Guide to SQL: Reference*.

Assigning Column Constraints

You must use the CREATE TABLE statement, not the schema editor, to place constraints on one or more columns in a table. The following example creates a table called **accounts** and puts a constraint on the first column:

```
CREATE TABLE accounts
(a_name CHAR(12) UNIQUE CONSTRAINT acc_name, a_code SERIAL)
```

See the CREATE TABLE statement in the of *IBM Informix Guide to SQL: Reference* for the exact syntax.

Creating an Index on a Table

The schema editor creates a nonclustered, ascending index for the values in the associated database column. To create a different type of index, use the CREATE INDEX statement. (To modify a different type of index, use the ALTER INDEX statement.) The following example creates a clustered index on the **customer** table that orders the table by **zipcode**:

```
CREATE CLUSTER INDEX c_clust_ix ON customer (zipcode)
```

The CREATE INDEX and ALTER INDEX statements are described in detail in the *IBM Informix Guide to SQL: Reference*. Also, see the section “Index Name” in the *IBM Informix Guide to SQL: Reference* and the discussion of indexing in the *IBM Informix Guide to SQL: Tutorial*.

How to Modify a Table

Issue the ALTER TABLE statement to modify the structure of a table in a database. You can include various clauses in that SQL statement to add a column, delete a column, modify the data type of a column, and add or drop a unique constraint associated with one or more columns in the table.

You must own the table, have DBA privilege, or be granted the ALTER privilege on the table to use ALTER TABLE.

See the complete syntax and usage for the ALTER TABLE statement in the *IBM Informix Guide to SQL: Reference*.

Adding a Column to a Table

Include an ADD clause in an ALTER TABLE statement to add one or more new columns to a table. The following example adds another code column at the end of the **cust_calls** table, assigning a CHAR data type and a length of 3:

```
ALTER TABLE cust_calls
  ADD (user_code CHAR(3))
```

When you run this statement, DB-Access appends the **user_code** column to the list of columns. You can add as many columns to the end of the table as you want, as long as each column name is unique.

You can include a BEFORE clause with the ADD keyword of the ALTER TABLE statement to insert a new column before an existing column. The following statement inserts the **county** column before the **state** column of the **customer** table:

```
ALTER TABLE customer
  ADD (county CHAR(25) BEFORE state)
```

You can insert as many columns in the table as you want. When you add a column to a table, DB-Access assigns a null value to each row for each new column. For this reason, you cannot add a column that disallows null values to an existing table. You can, however, add a column and modify it to disallow nulls in a later ALTER TABLE statement.

Modifying a Column in a Table

Include a MODIFY clause in an ALTER TABLE statement to

- Change the data type of a column
- Change the length of a character column
- Allow or disallow null values in a column
- Add or change the default value for a column

If you create a SMALLINT column and find later that you need to store integers larger than 32,767, you need to change the data type for the column to INTEGER. The following example changes the data type of the **item_num** column in the **items** table from SMALLINT to INTEGER:

```
ALTER TABLE items
  MODIFY (item_num INTEGER)
```

Similarly, you can increase the length of a CHAR column. The next example changes the length of the **company** and **city** columns in the **customer** table to 30 characters and 20 characters, respectively:

```
ALTER TABLE customer
  MODIFY (company CHAR(30),
         city CHAR(20))
```

The following example modifies the existing **company** column to not allow null values:

```
ALTER TABLE customer
  MODIFY (company CHAR(30) NOT NULL)
```

Dropping a Column from a Table

A table cannot contain two columns with the same name. If you accidentally add a column in the wrong location, you must delete it first and then add the column again in the correct location.

Use the ALTER TABLE statement with the DROP keyword to drop a column and any data it contains. The following statement drops the **manage_phone** column from the **customer** table:

```
ALTER TABLE customer
  DROP (manage_phone)
```

Renaming a Column in a Table

Use the RENAME COLUMN statement (not ALTER TABLE) to change the name of a column in a table. For example, to rename the **customer_num** column in the **customer** table to **cust_number**, run the following statement:

```
RENAME COLUMN customer.customer_num
  TO cust_number
```

When you rename a column, indexes and privileges are modified automatically to include the new name.

(RENAME COLUMN is an extension to ANSI syntax. If you set DBANSIWARN or specify the **-ansi** parameter, a warning message is displayed.)

Adding and Dropping Constraints

You must use the ALTER TABLE statement, not the schema editor, to add or drop a constraint for a column in a table. The following example adds the unique constraint **u_name** to the **manu_name** column of the **manufact** table:

```
ALTER TABLE manufact
  ADD CONSTRAINT UNIQUE (manu_name)
  CONSTRAINT u_name
```

Existing values in the column or columns you select must already be unique, and the column or columns you select cannot have an ascending index.

The following example drops the existing constraint **u_name** from the **manufact** table:

```
ALTER TABLE manufact
  DROP CONSTRAINT (u_name)
```

You must specify the name of the constraint that you want to drop. If you do not remember the name of a constraint, you can query the **sysconstraints** table in the system catalog.

See the ALTER TABLE statement in the *IBM Informix Guide to SQL: Reference* for the exact syntax.

How To Get Table Information

Issue the INFO statement to get information about the *structure* and *characteristics* of your new table. Use various clauses in the INFO statement to display information about the columns, indexes, privileges, and status of a table.

The following example requests column information for the **cust_calls** table:

```
INFO COLUMNS FOR cust_calls
```

The following example requests column information for the table **bunyip.customer** in the **stores5** database on the **topend** database server, which you can access if you have IBM Informix STAR:

```
INFO COLUMNS FOR stores5@topend:bunyip.customer
```

See the syntax and usage for the INFO statement in the *IBM Informix Guide to SQL: Reference*.

Use the SELECT statement in DB-Access to query on the *data* in one or more tables in a database.

The following example selects four columns from the **cust_calls** table and displays data for the specified **user_id**:

```
SELECT customer_num, call_code, call_dtime, res_dtime
FROM cust_calls
WHERE user_id = "harald"
```

You cannot find out the initial and next extent sizes or the lock mode for a table through a menu or the Info option. You can, however, issue a SELECT statement within the SQL Menu to access the **systables** system catalog table.

The following example displays the initial and next extent sizes for all the tables in the database:

```
SELECT tabname, fextsize, nextsize FROM systables
```

The following example displays the lock mode for all the tables in the database:

```
SELECT tabname, locklevel FROM systables
```

The SELECT statement is described in detail in the *IBM Informix Guide to SQL: Reference*. In addition, examples of simple and advanced SELECT statements appear in the *IBM Informix Guide to SQL: Tutorial*.

How to Drop a Table

Issue the DROP TABLE statement to remove a table from a database.

You must be the owner of a table or have DBA privilege to drop the table. Before you remove a table from the database, be sure that you have chosen the database that contains the table you want to drop.

Remember, when you drop a table, you also lose all the data in that table. Unlike the Drop option on the TABLE Menu, this method does not give you a chance to reconsider once you run the statement. If you accidentally delete the wrong table, you must recreate it and restore all the data from a backup copy. If you do not have a backup copy, you must reenter all the data.

The following example drops a table that is in the current database and owned by the current user:

```
DROP TABLE goanna
```

See the complete syntax and usage for the DROP TABLE statement in the *IBM Informix Guide to SQL: Reference*.

How to Drop a Database

Issue the DROP DATABASE statement to drop a database. If you want to drop the current database, you must first run the CLOSE DATABASE statement. You cannot drop a database that is in use, even if you are the only one using it.

You must have the appropriate database privileges to drop a database. Before you can drop a database, be sure that you own all the tables in the database or have DBA status.

Remember, when you drop a database, you also lose all the tables, indexes, and data in that database. Unlike the Drop option on the DATABASE Menu, this method does not give you a chance to reconsider once you run the statement.

The following example drops an IBM Informix SE database by specifying the full pathname:

```
DROP DATABASE "/riley/snowy/cooma"
```

See the complete syntax and usage for the DROP DATABASE and CLOSE DATABASE statements in the *IBM Informix Guide to SQL: Reference*.

The Query-Language Menu Option

In This Chapter	3-3
Choosing the Query-Language Option	3-4
Entering a New SQL Statement	3-6
Running an SQL Statement	3-7
Running Error-Free Statements	3-8
What Happens When There Are Errors	3-8
Viewing the Next Page of Data	3-9
Modifying an SQL Statement	3-10
Editing an SQL Statement	3-11
Redirecting Query Results	3-12
Sending Output to a Printer	3-13
Sending Output to a File	3-14
The New-file Option	3-14
The Append-file Option	3-14
Sending Output to a Pipe	3-15
Choosing an Existing SQL Statement	3-16
Sample SQL Command Files	3-17
Saving the Current SQL Statement	3-19
Displaying Table Information	3-20
Dropping an SQL Statement	3-23

In This Chapter

This chapter describes how to use the features of the Query-Language option on the DB-Access Main Menu. Select this option if you want to work with the Informix implementation of Structured Query Language (SQL).

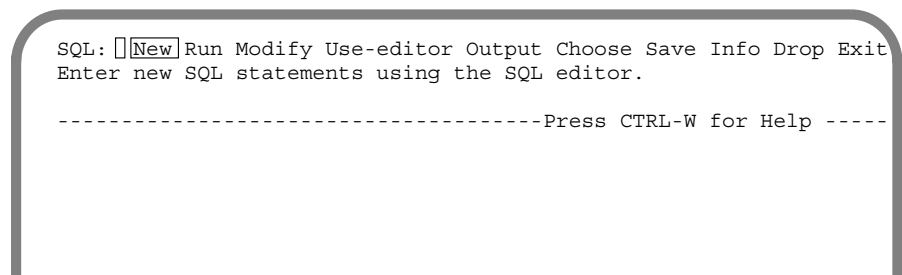
To work with databases and tables, you can either enter an SQL statement or select the Database or Table option from the DB-Access Main Menu. This chapter discusses the first method, which involves using a text editor. For information on using a text editor, a map of the Main Menu, and an overview of how to work with menu, text-entry, and HELP screens, refer to [Chapter 1, “Working with DB-Access.”](#)

For complete details on how to construct and use SQL statements, see *IBM Informix Guide to SQL: Tutorial* and *IBM Informix Guide to SQL: Reference*. The latter book shows the syntax for the SQL statements that you can use with DB-Access.

Choosing the Query-Language Option

Select the Query-Language option from the DB-Access Main Menu to use SQL, the Structured Query Language. Type `q` or `Q`, or highlight the Query-Language option and press `RETURN`, and the SQL Menu appears, as shown in [Figure 3-1](#).

Figure 3-1
The SQL Menu



The SQL Menu displays the following 10 options:

- | | |
|------------|--|
| New | lets you enter new SQL statements using the SQL editor. (See “Entering a New SQL Statement” on page 3-6.) |
| Run | executes the current SQL statement or statements. (See “Running an SQL Statement” on page 3-7.) |
| Modify | lets you use the SQL editor to modify the current SQL statement or statements. (See “Modifying an SQL Statement” on page 3-10.) |
| Use-editor | lets you enter or edit SQL statements with a system editor. (See “Editing an SQL Statement” on page 3-11.) |
| Output | routes the output from executing the current SQL statements to a system file, printer, or system pipe. (See “Redirecting Query Results” on page 3-12.) |

Choose	lets you select an existing command file containing SQL statements and make them your current statements. You can run or edit the current statements. (See “Choosing an Existing SQL Statement” on page 3-16.)
Save	saves the current SQL statements in a command file for later use. You can retrieve this command file by selecting the Choose option on the SQL Menu. (See “Saving the Current SQL Statement” on page 3-19.)
Info	lets you retrieve information about the columns, indexes, access privileges, reference privileges, and status of a table. (See “Displaying Table Information” on page 3-20.)
Drop	drops a command file from the database. (See “Dropping an SQL Statement” on page 3-23.)
Exit	exits the SQL Menu and returns you to the DB-Access Main Menu.

Within the Query-Language option, you can enter any SQL statement supported by the database server. The SQL statements available in Version 5.0 of IBM Informix products are listed in [Chapter 2, “Creating Databases and Tables.”](#) For complete details of SQL statements and their syntax and usage, see the *IBM Informix Guide to SQL: Reference*.

If you select the Query-Language option on the DB-Access Main Menu and you have not yet specified a database with which to work, the SELECT DATABASE screen appears. You can specify a current database or press the Interrupt key and go to the SQL Menu to choose a command file containing an SQL statement where the database is specified. (See [Chapter 4, “The Database Menu Option,”](#) for information on selecting and creating databases, or [Chapter 1, “Working with DB-Access,”](#) for how to select a database from the command line when you invoke DB-Access.)

Entering a New SQL Statement

When you are ready to enter an SQL statement, select the New option on the SQL Menu. The NEW screen appears and indicates that you are using the SQL editor, as shown in [Figure 3-2](#).

Figure 3-2

The NEW screen for entering new SQL statements

```
NEW:      ESC      = Done editing          CTRL-A = Typeover/Insert
          CTRL-R = Redraw                CTRL-X = Delete character
          CTRL-D = Delete rest of line

----- stores5 ----- Press CTRL-W for Help -----
█
```

When the NEW screen appears, you can type in an SQL statement using the SQL editor. The editor enables you to enter statements and edit them before you run them.

You can use almost any format you like when you enter an SQL statement. You can string several SQL statements together by using a semicolon to separate them. See the *IBM Informix Guide to SQL: Reference* for rules regarding syntax and usage of SQL statements.

Generally, if you make a mistake while typing a new SQL statement, you can move the cursor with the arrow keys and reenter parts of the statement. You also can use the editing keys to change between insert and typeover mode, redraw the screen, delete a character, and delete the rest of a line. The uses of the editing keys, listed at the top of the screen, are described in the section [“Using the SQL Editor” on page 1-26](#).

When you use the SQL editor, you can enter as many lines of text as you wish. You are limited only by the memory constraints of your system, not by the size of the screen. See the discussion in [Chapter 1, “Working with DB-Access,”](#) for additional restrictions.

If you prefer working with the system editor, you can always press ESC and then select the Use-editor option on the SQL Menu. This option is described in the section “Editing an SQL Statement” on page 3-11.

When you have finished entering a new SQL statement, press ESC. The SQL Menu reappears and you can choose to run the statement.

Tip: *If there are current statements on the screen and you do not save them in a command file with the Save option, they will be lost when you select the New option.*



Running an SQL Statement

The SQL Menu reappears with the Run option highlighted and the statement text displayed in the bottom part of the screen, as shown in [Figure 3-3](#).

Figure 3-3

The SQL Menu with SQL statement text ready to run

```
SQL: [ ] New [Run] Modify Use-editor Output Choose Save Info Drop Exit
Run the current SQL statements.
```

```
----- stores5 ----- Press CTRL-W for Help -----
```

```
CREATE TABLE mystock
(
  stock_num      SMALLINT,
  manu_code      CHAR(3),
  description     CHAR(15),
  unit_price     MONEY(6),
  unit           CHAR(4),
  unit_descr     CHAR(15),
  PRIMARY KEY (stock_num, manu_code) CONSTRAINT stock_man_primary
  FOREIGN KEY (manu_code) REFERENCES manufact
)
```

Press RETURN or type R or r, and DB-Access will do one of two things:

- Run the statement
- Display an error message on the screen, indicating a syntax or other error

When you select the Run option, DB-Access first checks each statement to make sure it conforms to the SQL usage and syntax rules. If your statements contain no mistakes, DB-Access processes them.

Running Error-Free Statements

If a statement runs correctly, DB-Access displays the requested results (for a SELECT statement) or displays a message at the bottom of the screen indicating that the action you requested has been completed. For the statement shown in [Figure 3-3](#), you would see the following message:

```
Table created.
```

What Happens When There Are Errors

If an error occurs, the Modify option on the SQL Menu is highlighted. You can press RETURN to choose this option, or select the Use-editor option to correct the statement, and then try to run it again.

If you make any syntax or typing mistakes in an SQL statement, DB-Access does not process the statement. Instead, it continues to display the text of the statement along with a message describing the error. For example, if there is a syntax error, you see the following message at the bottom of the screen:

```
201: A syntax error has occurred.
```

If there is an execution or run-time error, DB-Access continues to process the statement and also returns an error message. For example, if you try to create a table that has already been created, you see the following kind of message at the bottom of the screen:

```
310: Table (mavis.mystock) already exists in database.
```

If you try to execute a command that contains more than one SQL statement, you might not see the error message immediately. If, for example, the first statement is a SELECT that runs correctly, whereas the next statement contains a typing error, the data retrieved by the first statement appears on the screen before the error message appears for the second statement.

Viewing the Next Page of Data

When you run a statement such as a SELECT, the query might retrieve more rows than can fit on a single screen. In this case, you see a screen such as the one shown in [Figure 3-4](#).

Figure 3-4

The Next Page screen for displaying additional query results

```

RUN: Next Restart Exit
Display the next page of query results.

----- stores5 ----- Press CTRL-W for Help -----

customer_num      106
call_dtime        1990-06-12 08:20
user_id           maryj
call_code         D
call_descr        Order was received, but two of the cans of ANZ
                  tennis balls within the case were empty
res_dtime         1990-06-12 08:25
res_descr         Authorized credit for two cans to customer, issued
                  apology. Called ANZ buyer to report the QA problem
customer_num      110
call_dtime        1990-07-07 10:24
user_id           richc
call_code         L
call_descr        Order placed one month ago (6/7) not received.
res_dtime         1990-07-07 10:30
res_descr         Checked with shipping (Ed Smith). Order sent
                  yesterday- we were waiting for goods from ANZ.
                  Next time will call with delay if necessary.

```

The Next option is highlighted. Keep pressing RETURN until you have seen all the rows selected. A message at the bottom of the screen indicates the number of rows returned when you are on the last screen of information.

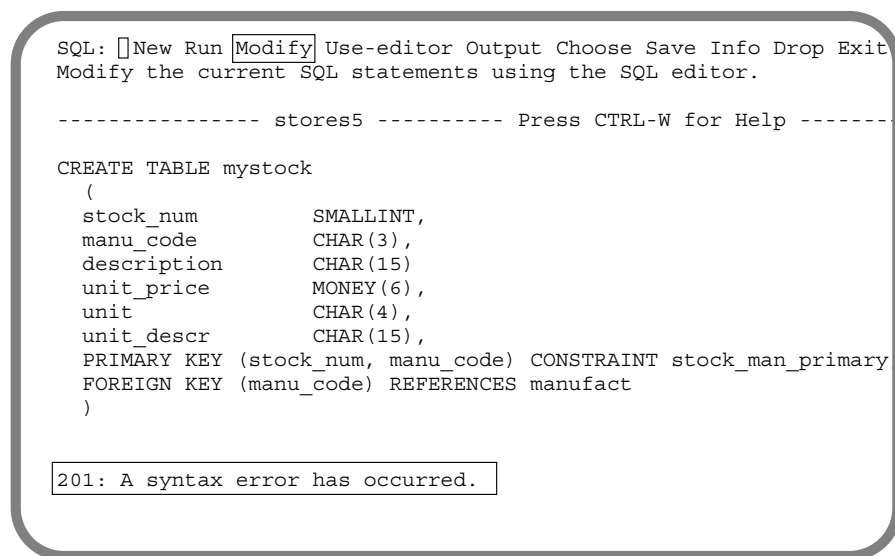
Select Restart at any time to display the results from the beginning. Then press RETURN to continue viewing the results or Exit to return to the SQL Menu. If you exit before all the results have been displayed, a message at the bottom of the screen indicates that the query has been interrupted.

Modifying an SQL Statement

When DB-Access finds an error in an SQL statement you are trying to run, the Modify option is highlighted on the SQL Menu, and the current statement text and error message are displayed as shown in [Figure 3-5](#).

Figure 3-5

The SQL Menu with SQL statement text to be modified

The screenshot shows a terminal window with a menu at the top: 'SQL: [] New Run [Modify] Use-editor Output Choose Save Info Drop Exit'. The 'Modify' option is highlighted with a box. Below the menu is a separator line: '----- stores5 ----- Press CTRL-W for Help -----'. The main content is an SQL statement: 'CREATE TABLE mystock (stock_num SMALLINT, manu_code CHAR(3), description CHAR(15), unit_price MONEY(6), unit CHAR(4), unit_descr CHAR(15), PRIMARY KEY (stock_num, manu_code) CONSTRAINT stock_man_primary FOREIGN KEY (manu_code) REFERENCES manufact)'. At the bottom, a message box contains the text: '201: A syntax error has occurred.'

```
SQL: [ ] New Run [Modify] Use-editor Output Choose Save Info Drop Exit
Modify the current SQL statements using the SQL editor.

----- stores5 ----- Press CTRL-W for Help -----

CREATE TABLE mystock
(
  stock_num      SMALLINT,
  manu_code      CHAR(3),
  description     CHAR(15),
  unit_price     MONEY(6),
  unit           CHAR(4),
  unit_descr     CHAR(15),
  PRIMARY KEY (stock_num, manu_code) CONSTRAINT stock_man_primary
  FOREIGN KEY (manu_code) REFERENCES manufact
)

201: A syntax error has occurred.
```

Press RETURN, and DB-Access calls the SQL editor and positions the cursor on the line with the first error. You can correct the error by using the SQL editor, or you can press ESC to exit to the SQL Menu and select the Use-editor option to edit the statement using your system editor.

- If you use the SQL editor to make your corrections, press ESC when you have finished editing the statement.
- If you use the system editor, exit the file according to the convention for that editor.

You then return to the SQL Menu, where you can press RETURN to run the statement again.



Tip: You can find the text of all error messages, along with suggestions for correcting the errors, in the “IBM Informix Error Messages” manual. Error messages and their corrective actions also are available in an online file called **errmsg.txt**, located in the **\$INFORMIXDIR/msg** directory. For details on how to view online error messages, see the Introduction to this manual.

Editing an SQL Statement

If you want to use a system editor to enter a new SQL statement or to modify an existing SQL statement or command file, select the Use-editor option on the SQL Menu. A text-entry screen appears and puts you in a temporary edit file.

If this is the first time in a session that you are calling a system editor, or if the DBEDIT environment variable is not set, you might be asked to select an editor before you can continue. The screen displays the default editor; press RETURN to use that editor or enter another editor that is accepted by your system.

If this is a new SQL statement, the screen is blank; if this is a statement that you have elected to change or have called up with the Choose option, the text of the SQL statement appears on the screen.

Figure 3-6 shows the text of an incorrect SQL statement with an error message positioned below the error and displays the name of the temporary file that contains the statement.

Figure 3-6

A temporary text-editing file with SQL statement text to be corrected

```
CREATE TABLE mystock
(
  stock_num      SMALLINT,
  manu_code      CHAR(3),
  description    CHAR(15)
  unit_price     MONEY(6),
  ^
#
#201:A syntax error has occurred
#
unit            CHAR(4),
unit_descr     CHAR(15),
PRIMARY KEY (stock_num, manu_code) CONSTRAINT stock_man_primary,
FOREIGN KEY (manu_code) REFERENCES manufact
)

"tmp/rsqa02775.err" 11 lines, 132 characters
```

Make your corrections to the text. When you finish entering or modifying your SQL statement(s), exit as you normally would with that particular editor. The SQL Menu reappears with the Run option highlighted and the statement text displayed in the bottom half of the screen.

Press RETURN to run the statement. Or, select another menu option to save the statement in a command file or direct its output.

Redirecting Query Results

The output from an SQL statement such as SELECT normally appears on the screen. The Output option on the SQL Menu lets you send query results to the printer, store them in a system file, or pipe them to a program.

You can use the SQL statement OUTPUT for the same purpose. It is described in the *IBM Informix Guide to SQL: Reference*.

Select the Output option on the SQL Menu. The OUTPUT Menu appears, as shown in [Figure 3-7](#).

Figure 3-7
The OUTPUT Menu for redirecting query results

```

OUTPUT: [Printer] New-file Append-file To-pipe Exit
Send query results to a printer.

----- stores5 ----- Press CTRL-W for Help -----

SELECT * FROM customer

```

Before you can send query results to a printer, file, or pipe, you must have the SELECT statement on the screen as the current statement. Write a new statement, modify or edit an existing one, or choose a command file (as discussed in the section “[Choosing an Existing SQL Statement](#)”) and return to the SQL Menu.

Select an option on the OUTPUT Menu, and the current query is printed, stored, or piped to a program just as it would have appeared on the screen.

If you want to return to the SQL Menu without redirecting query results from an OUTPUT screen, press the Interrupt key.

Sending Output to a Printer

If you want to send your query results directly to a printer, select the Printer option on the OUTPUT Menu. DB-Access sends the results to your default printer and displays a message on the bottom of the screen indicating how many rows were retrieved. The query results do not appear on the screen.

Sending Output to a File

You can write query results to a new file or append the results to an existing file. You can then use your operating system programs to edit, copy, rename, or delete the file.

The New-file Option

If you want to create a new file to store query results, select the New-file option on the OUTPUT Menu. You then see the OUTPUT NEW-FILE screen, as shown in [Figure 3-8](#).

Figure 3-8
The OUTPUT NEW-FILE screen

```
OUTPUT NEW-FILE >> 
Enter the name you want to assign to the new file, then press Return.

----- stores5 ----- Press CTRL-W for Help -----

SELECT * FROM customer
```

Type a name for the file and then press RETURN. DB-Access forwards the results of the query to that file and displays a message telling how many rows were retrieved. The query results do not appear on the screen.

Tip: If you enter the name of an existing file, this procedure overwrites the existing file with the query results.

The Append-file Option

If you want to add your query results to the end of an existing file without replacing what is already stored there, select the Append-file option on the OUTPUT Menu. You then see the OUTPUT APPEND-FILE screen, as shown in [Figure 3-9](#).



Figure 3-9
The OUTPUT APPEND- FILE screen

```
OUTPUT APPEND-FILE >>   
Enter the name of the file you want to append results to, then press Return  
.  
  
----- stores5 ----- Press CTRL-W for Help -----  
  
SELECT * FROM customer
```

Type the name of an existing file where you want to append the query results and press RETURN. DB-Access appends the query results to the end of that file and displays a message telling how many rows were retrieved. The query results do not appear on the screen.

Sending Output to a Pipe

If you want to send your query results to a pipe, select the To-pipe option on the OUTPUT Menu. You then see the OUTPUT TO-PIPE screen, as shown in [Figure 3-10](#).

Figure 3-10
The OUTPUT TO-PIPE screen

```
OUTPUT TO-PIPE >>   
Enter the name of the pipe you want to send results to, then press Return.  
  
----- stores5 ----- Press CTRL-W for Help -----  
  
SELECT * FROM customer
```

Specify a pipe program, and DB-Access sends the results to that pipe, if you have permission to run it. The query results do not appear on the screen.

Choosing an Existing SQL Statement

When you save SQL statements in a command file, as described in the section “Saving the Current SQL Statement” on page 3-19, you can retrieve the command file and run or edit the SQL statements at any time.

Select the Choose option on the SQL Menu. The CHOOSE screen appears and displays a list of the command files to which you have access. (The `.sql` extension is not shown.) For example, the screen shown in Figure 3-11 lists the command files that are included with the demonstration database; they are illustrated and described in Appendix A, “Sample Command Files.”

Figure 3-11

The CHOOSE screen listing current command files

```
CHOOSE >> 
Choose a command file with the Arrow Keys, or enter a name,
then press Return.

----- stores5 ----- Press CTRL-W for Help -----
c_calls      c_stock      opt_disk      sel_order
c_cat        c_stores     sel_agg       sel_sub
c_custom     c_table      sel-all      sel_union
c_index      c_type       sel_group     upd_table
c_items      c_view1      sel_join
c_manuf      c_view2      sel_ojoin1
c_orders     d_proc       sel_ojoin2
c_proc       d_view       sel_ojoin3
c_state      ins_table    sel_ojoin4
```

If there is no current database, the list includes all the command files located in the current directory and in any directories specified by the DBPATH environment variable.

To select a command file, use the arrow keys to highlight its name or type the name of the file at the prompt and then press RETURN.

The SQL Menu reappears with the statements contained in the command file displayed on the screen as the current statements. You can then modify, run, edit, or output these statements by choosing the appropriate menu option.

To leave the CHOOSE screen without selecting a command file, press the Interrupt key. You then return to the SQL Menu.

Sample SQL Command Files

In addition to the command files you create, if **dbaccessdemo5** has been run in the current directory, the screen lists SQL command files that were included with DB-Access to provide practice with the **stores5** demonstration database.

Some files contain the SQL statements used to create the **stores5** database and its tables. Other files let you create an index, create and drop a view and a stored procedure, insert rows into and update a table, query on the data in tables in various ways, and otherwise familiarize yourself with SQL and relational database concepts. An alphabetical list of these command files follows. They are shown and described in [Appendix A](#).

- c_calls.sql** creates the **cust_calls** table.
- c_cat.sql** creates the **catalog** table (IBM Informix OnLine only).
- c_custom.sql** creates the **customer** table.
- c_index.sql** creates an index on the **zipcode** column of the **customer** table.
- c_items.sql** creates the **items** table.
- c_manuf.sql** creates the **manufact** table.
- c_orders.sql** creates the **orders** table.
- c_proc.sql** creates a stored procedure.
- c_state.sql** creates the **state** table.
- c_stock.sql** creates the **stock** table.

c_stores.sql	creates the stores5 database.
c_table.sql	creates a new database and a custom table named sports .
c_type.sql	creates the call_type table.
c_view1.sql	creates a single-table view.
c_view2.sql	creates a multiple-table view.
d_proc.sql	drops a stored procedure.
d_view.sql	drops a view.
ins_table.sql	inserts rows into the sports table.
sel_agg.sql	queries on table data: sample SELECT with aggregate function.
sel_all.sql	queries on table data: contains all seven SELECT statement clauses.
sel_group.sql	queries on table data: sample SELECT statement with GROUP BY and HAVING clauses.
sel_join.sql	queries on table data: sample SELECT statement with simple join on two tables.
sel_ojoin1.sql	queries on table data: sample SELECT statement with simple outer join on two tables.
sel_ojoin2.sql	queries on table data: sample SELECT statement with outer join for simple join to a third table.
sel_ojoin3.sql	queries on table data: sample SELECT statement with outer join for outer join to a third table.
sel_ojoin4.sql	queries on table data: sample SELECT statement with outer join of each of two tables to a third table.
sel_order.sql	queries on table data: sample SELECT statement with ORDER BY and WHERE clauses.

- sel_sub.sql** queries on table data: sample SELECT statement with a sub-query.
- sel_union.sql** queries on table data: sample SELECT statement with UNION on two tables.
- upd_table.sql** updates the **sports** table.

See the *IBM Informix Guide to SQL: Reference* for the names and structure of all the tables in the **stores5** database, plus a map of the database.

Saving the Current SQL Statement

When you have entered and successfully run an SQL statement, you can save it in a *command file* for later use. A command file is a system file that contains one or more SQL statements.

To save the current SQL statement(s) in a command file, select the Save option on the SQL Menu. DB-Access displays the SAVE screen and prompts you to enter a name for the command file as shown in [Figure 3-12](#).

Figure 3-12

The SAVE screen for saving results in a command file

```
SAVE >> 
Enter the name you want to assign to the command file.

----- stores5 ----- Press CTRL-W for Help -----

CREATE TABLE sports
(
  catalog_no          SERIAL(3001),
  stock_no           SMALLINT,
  mfg_code            CHAR(5),
  mfg_name            CHAR(20),
  phone               CHAR(18),
  descript            VARCHAR(255)
);
```

Enter a name for the command file and press RETURN. (The example shown is in the file **c_table**. That name will thus appear on the SAVE screen.)

Command file names can be up to 10 characters long. The first character must be a letter, but you can use letters, numbers, and underscores (_) for the rest of the name.

You can use uppercase and lowercase letters in the name. However, remember that UNIX systems are case-sensitive. The file **ords1** is not the same as **Ords1** or **ORDS1**.

DB-Access stores the statements in a file, using the name you gave and the extension **.sql**. For example, a statement you name **cust1** is stored in a command file named **cust1.sql**. It will appear as **cust1** on the CHOOSE screen, but as **cust1.sql** if you list the directory files from the command line. You can retrieve the saved statements at any time with the Choose option on the SQL Menu.

To leave the SAVE screen without assigning a name to a command file, press the Interrupt key. You return to the SQL Menu.

Displaying Table Information

Use the Info option on the SQL Menu to display information about the columns, indexes, access privileges, reference privileges, and status of a table.

The Info option is also available from the TABLE Menu, as described in [Chapter 5](#). In addition, you can use the SQL statement INFO for the same purpose. It is described in the *IBM Informix Guide to SQL: Reference*.

Select the Info option on the SQL Menu. DB-Access displays the INFO FOR TABLE screen, as shown in [Figure 3-13](#).

Figure 3-13
The INFO FOR TABLE screen

```

INFO FOR TABLE >> 
Choose a table with the Arrow Keys, or enter a name, then press
Return.

----- mydata ----- Press CTRL-W for Help -----

clients
customer
orders
    
```

This screen lists the names of tables that exist in the current database. If you are not the owner of a table, the table name is prefixed by the owner name, as in "june".clients.

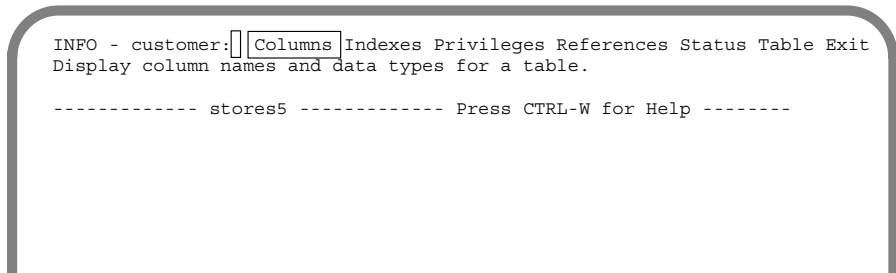
To leave the INFO FOR TABLE screen without requesting table information, press the Interrupt key. You return to the TABLE Menu.

You can select a table in one of two ways:

- Type its name and press RETURN.
- Use the arrow keys to highlight the name of the table you want and then press RETURN.

For example, for the **customer** table, type in **customer** or highlight it with an arrow key, and then press RETURN. The INFO Menu appears, with **customer** in the top line, as shown in [Figure 3-14](#).

Figure 3-14
The INFO Menu for displaying table information



The INFO Menu displays the following seven options:

- | | |
|------------|--|
| Columns | lists all the columns in the specified table and displays their data type and whether they allow null values. |
| Indexes | lists the name, owner, and type of each index for a specified table, whether the index is clustered, and the names of the columns that are indexed. |
| Privileges | lists the users who have access privileges for the specified table and whether they have the table-level privileges Select, Update, Insert, Delete, Index, and Alter.

Unless your login is listed separately, you have the privileges given for public (a general category for all users).

If you want information about database-level privileges, you must use a SELECT statement within the SQL Menu to access the sysusers system catalog table. |
| References | lists the users who have the table-level References privilege for the specified table and which columns they can reference. |

Status	lists the table name, the table owner, the size of the row (in number of bytes), the number of rows in the table (as of the last UPDATE STATISTICS statement), the number of columns in a row, and the date the table was created. (On IBM Informix SE, it also lists the name of the audit trail file, if there is one.)
Table	redispays the INFO FOR TABLE Menu so you can select a new table for which to request information on the INFO Menu.
Exit	leaves the INFO Menu and returns you to the SQL Menu.

You cannot find out the initial and next extent sizes or the lock mode for a table through a menu or through the Info option. You can, however, issue a SELECT statement within the SQL Menu to access the **systables** system catalog table.

If you are using IBM Informix STAR, you can use the Info option to request information about external tables on the same or another OnLine server by entering the expanded table name at the prompt. The following example requests information on the **bunyip.customer** table in the **stores5** database on the IBM Informix OnLine **topend** server:

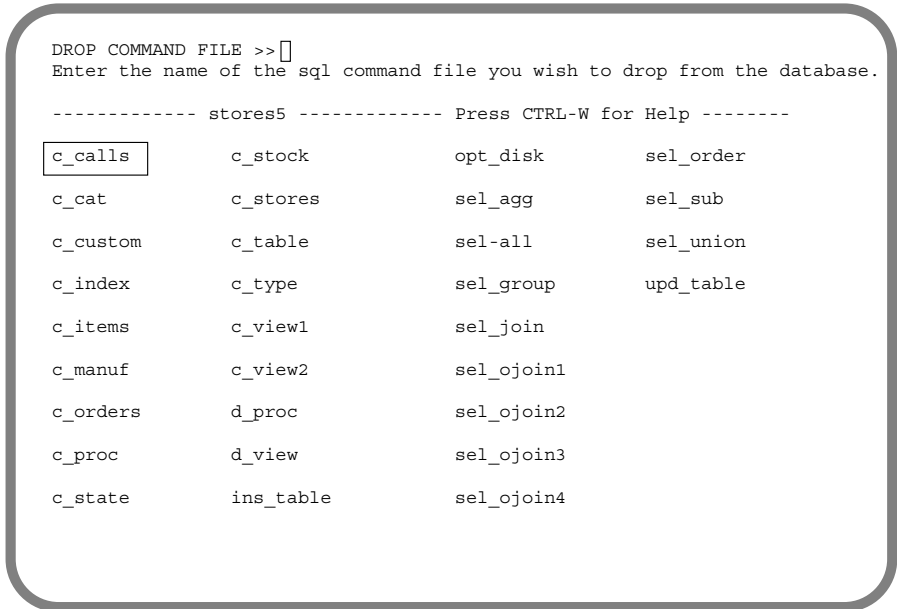
```
INFO FOR TABLE >> stores5@topend:bunyip.customer
```

Dropping an SQL Statement

If **dbaccessdemo5** has been run, your current database directory includes a number of command files that contain SQL statements. Some of these files are included with the demonstration database when the database server is installed; others may have been created by you and put in the directory with the Save option on the SQL Menu.

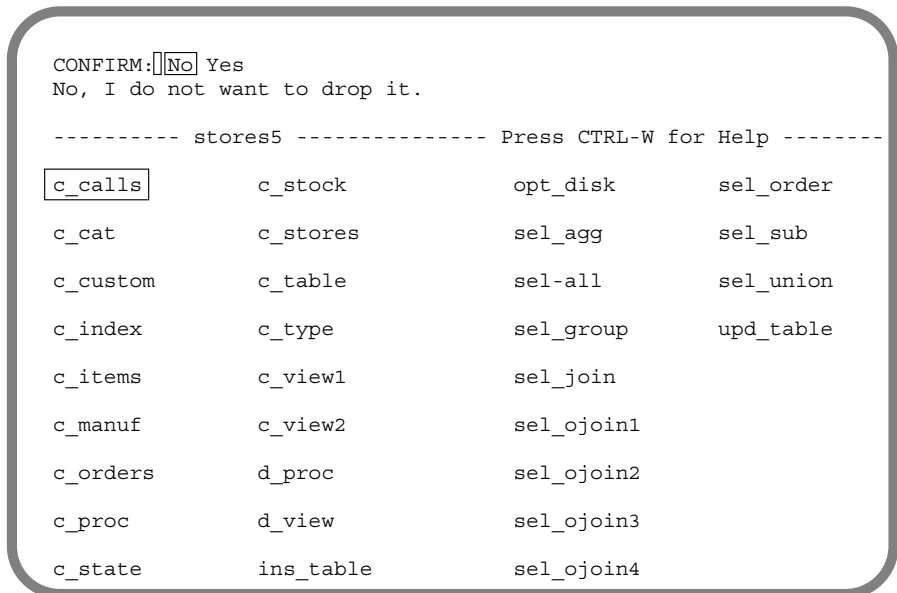
You can remove command files from the current database directory by selecting the Drop option on the SQL Menu. The DROP COMMAND FILE screen appears with an alphabetical list of command files in the current database, as shown in [Figure 3-15](#).

Figure 3-15
The DROP COMMAND FILE screen



Type in the name of the command file you want to drop or highlight it with the arrow keys and press RETURN. DB-Access then displays a special screen header that asks for confirmation before it drops the command file. (See [Figure 3-16](#)).

Figure 3-16
The DROP COMMAND FILE confirmation screen



The default is No, to help prevent you from deleting a command file by mistake. To drop the highlighted command file, press Y or use the right arrow key to highlight Yes and press RETURN. DB-Access drops the command file and returns you to the SQL Menu.

To leave the DROP COMMAND FILE screen without dropping a command file, press N or RETURN, or press the Interrupt key. You then return to the SQL Menu.

The Database Menu Option

In This Chapter	4-3
Choosing the Database Option	4-4
The SELECT DATABASE Screen	4-5
Selecting a Database	4-5
Differences Between IBM Informix OnLine and IBM Informix SE	4-6
Exiting the Screen	4-6
The CREATE DATABASE Screen.	4-7
Creating a Database	4-7
Differences Between IBM Informix OnLine and IBM Informix SE	4-8
Exiting the Screen	4-9
The DROP DATABASE Screen	4-9
Dropping a Database	4-10
Confirming Your Decision.	4-11
Exiting the Screen	4-11

In This Chapter

This chapter describes how to use the features of the Database option on the DB-Access Main Menu. Select this option if you want to create a database, select a database to work with, or drop an existing database.

To select, create, or drop a database, you can either enter an SQL statement, as described in [Chapter 3, “The Query-Language Menu Option,”](#) or select an option on the DATABASE Menu. This chapter discusses the second method.

You must specify a database before you can work with DB-Access. You can either select an existing database or create one. The database you are going to work with is called the *current* database.

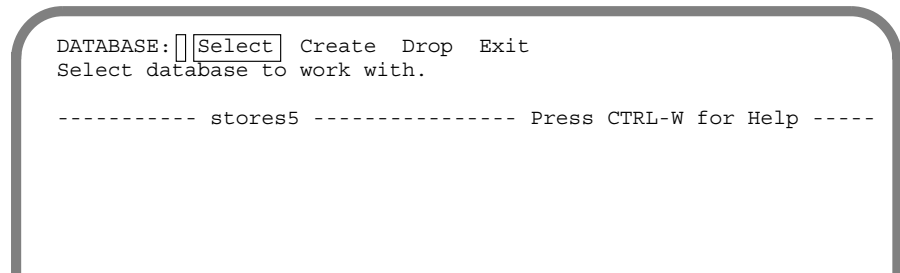
When you first select a database that was built with a pre-5.0 IBM Informix product, the database server automatically converts the database to 5.0.

For a map of the Main Menu plus an overview of how to work with menu, text-entry, and HELP screens, refer to [Chapter 1, “Working with DB-Access.”](#)

Choosing the Database Option

On the DB-Access Main Menu, type `d` or `D`, or highlight the Database option and press RETURN, to call up the DATABASE Menu, as shown in [Figure 4-1](#).

Figure 4-1
The DATABASE Menu



The DATABASE Menu displays the following four options:

- | | |
|--------|---|
| Select | makes a database the current database. |
| Create | builds a new database and makes that database the current database. |
| Drop | removes a database from the system. |
| Exit | exits the DATABASE Menu and returns you to the DB-Access Main Menu. |

The SELECT DATABASE Screen

When you choose the Select option from the DATABASE Menu, the SELECT DATABASE screen appears, as shown in [Figure 4-2](#).

Figure 4-2
The SELECT DATABASE screen

```

SELECT DATABASE >> 
Select a database with the Arrow Keys, or enter a name, then
press Return.

----- mydata ----- Press CTRL-W for Help -----


stores5
  
```

This screen also appears if you choose the Table or Query-Language option, and you did not specify a database when you entered the `dbaccess` command at the system prompt.

Selecting a Database

As indicated on the screen, you can select a database in one of the following two ways:

- Type its name and press RETURN.
With an IBM Informix SE database, a pathname can precede this name. With IBM Informix OnLine, you can specify a server name when you enter the database name.
- Use the arrow keys to highlight the name of a database and then press RETURN.

For example, to select the **stores5** database, type in or highlight **stores5** and then press RETURN.

If you enter the name of a nonexistent database or a database that DB-Access cannot locate, DB-Access displays the following message:

```
329: Database not found or no system permission
```

You also can issue the SQL statement DATABASE to select a database. For details, see the DATABASE statement in the *IBM Informix Guide to SQL: Reference*.

Differences Between IBM Informix OnLine and IBM Informix SE

The list of available databases displayed on the SELECT DATABASE screen might vary, depending on whether you are using IBM Informix OnLine or IBM Informix SE.

- With IBM Informix OnLine, DB-Access displays the names of all databases on the current OnLine database server. (If multiple OnLine servers are running on your machine, the TBCONFIG environment variable determines the current database server.)
- With IBM Informix SE, DB-Access displays the names of databases located in the current directory and any directories specified in your DBPATH environment variable.

If you have IBM Informix STAR, you can select a database on another OnLine server by specifying the server name with the database name at the prompt on the SELECT DATABASE screen. Either of the following commands select the **cuttle** database on the **squid** database server:

```
cuttle@squid  
"/squid/cuttle"
```

The first method is preferred. Remember to include quotation marks if you use the second method, and do not include any spaces.

If you have IBM Informix NET, you can select an IBM Informix SE database on another UNIX database server by including the full pathname at the prompt on the SELECT DATABASE screen. The following command selects the **cuttle** database in the **octo** directory on the **squid** database server:

```
"/squid/octo/cuttle"
```

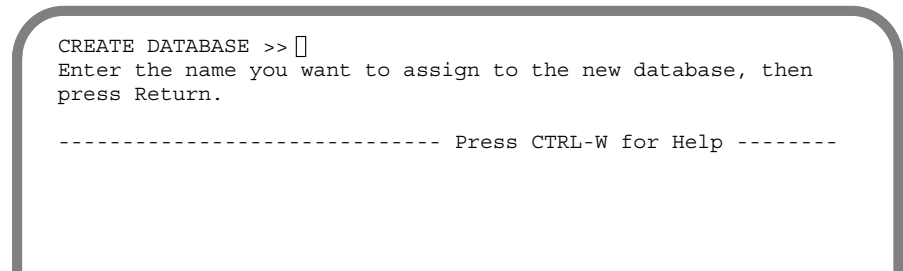
Exiting the Screen

To leave the SELECT DATABASE screen without selecting a database, press the Interrupt key. You return to the DATABASE Menu.

The CREATE DATABASE Screen

To create a new database instead of selecting an existing one, select the Create option from the DATABASE Menu. The CREATE DATABASE screen appears, as shown in [Figure 4-3](#).

Figure 4-3
The CREATE DATABASE screen



Creating a Database

Type the name of the new database and press RETURN. You can assign any name you want, as long as you follow these guidelines:

- The name you assign to the database can be from 1 to 10 characters long, inclusive.
- The database name must begin with a letter. The rest of the name can consist of any combination of letters, numbers, and underscores (_).
The database server does not distinguish between uppercase and lowercase letters in database names. Therefore, **tahiti**, **Tahiti**, and **TAHITI** all refer to the same database.
- Certain keywords may cause potential ambiguities or syntax errors when used to name a database, table, or column. See the “Identifier” segment in the *IBM Informix Guide to SQL: Reference* for a discussion of these keywords and available workarounds.
- See the discussion of IBM Informix OnLine and IBM Informix SE differences in the next section for additional restrictions regarding uniqueness.

When you create a database, it automatically becomes the current database. DB-Access displays the name of the current database in the middle of the broken line that separates the top and bottom of the screen.

If you enter the name of an existing database, DB-Access displays the following message:

```
330: Cannot create database.
```

Differences Between IBM Informix OnLine and IBM Informix SE

The database name must be unique within an IBM Informix OnLine database server. With IBM Informix SE, each database name within a given directory must be unique.

Each machine can have multiple OnLine servers, and separate OnLine servers can contain the same database name. However, on the same database server, the database name must be unique.

With IBM Informix OnLine, the new database is stored in the **root** dbspace unless you specify another dbspace. With IBM Informix SE, the new database is stored in the current directory or in the specified directory if you include the full pathname in the database name on the CREATE DATABASE screen.

If you have IBM Informix STAR, you can create a database on another OnLine server by specifying the server name with the database name at the prompt on the CREATE DATABASE screen. Either of the following examples specifies the **troppo** database on the **tonga** database server:

```
troppo@tonga  
"//tonga/troppo"
```

The first method is preferred. Remember to include quotation marks if you use the second method, and do not include any spaces.

If you have IBM Informix NET, you can create an IBM Informix SE database on another UNIX database server by including the full pathname at the prompt on the CREATE DATABASE screen. The following example specifies the **troppo** database in the **yasawa** directory on the **tonga** database server:

```
"//tonga/yasawa/troppo"
```

A database that you create with the Create option is not ANSI-compliant and has no logging. Because logging is not used, you cannot use transactions with this database.

However, with IBM Informix SE, you can turn on logging from within the Query-Language option with the SQL statement START DATABASE. See [Chapter 3, “The Query-Language Menu Option,”](#) for details on issuing SQL statements.

Also, if you are the IBM Informix OnLine administrator (user **informix**), you can make an IBM Informix OnLine database ANSI-compliant and start logging with the Databases option of the LOGICAL-LOGS Menu in DB-Monitor. (See the *IBM Informix OnLine Administrator’s Guide* for details.)

You must run the CREATE DATABASE statement (from within the Query-Language option) to create one of the following types of databases:

- An ANSI-compliant database (OnLine or SE)
- A database with logging (OnLine or SE)
- An OnLine database stored in a non-**root** dbspace

See the CREATE DATABASE statement in the *IBM Informix Guide to SQL: Reference* for information about logging, ANSI compliance, and other usage guidelines.

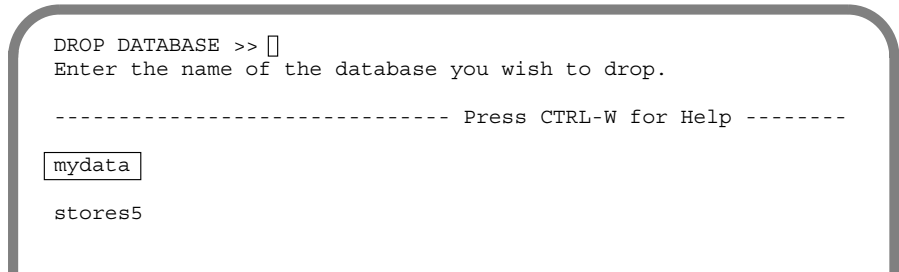
Exiting the Screen

To leave the CREATE DATABASE screen without creating a new database, press the Interrupt key. You return to the DATABASE Menu.

The DROP DATABASE Screen

When you select the Drop option from the DATABASE Menu, the DROP DATABASE screen appears, as shown in [Figure 4-4](#).

Figure 4-4
The DROP DATABASE screen



Dropping a Database

You can drop a database in one of the following two ways:

- Type its name and press RETURN.
- Use the arrow keys to highlight the name of the database you want and then press RETURN.

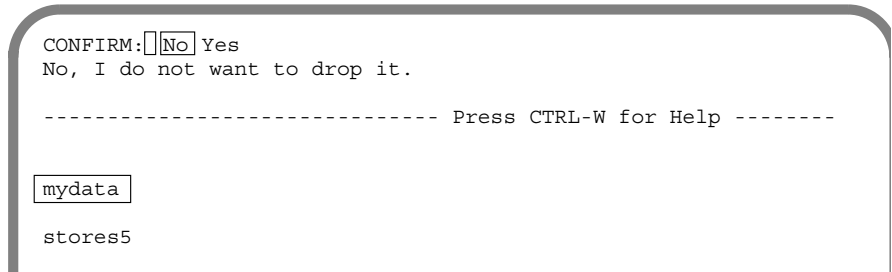
For example, to drop the **stores5** database, type in **stores5** or highlight it with an arrow key, and then press RETURN.

You also can issue the SQL statement DROP DATABASE. For more information on dropping a database, see the DROP DATABASE and CLOSE DATABASE statements in the *IBM Informix Guide to SQL: Reference*.

Confirming Your Decision

DB-Access displays a special screen header that asks for confirmation before it drops the database, as shown in [Figure 4-5](#).

Figure 4-5
The DROP DATABASE confirmation screen



The default is No, to help prevent you from dropping a database by mistake. If you truly want to drop the highlighted database, press `Y` or use the right arrow key to highlight Yes and press `RETURN`. DB-Access drops the database.

When you drop a database, all data in the database is gone for good. Be absolutely sure that you choose the correct database to drop.

Exiting the Screen

To leave the DROP DATABASE screen without dropping a database, press the Interrupt key. You return to the DATABASE Menu.

The Table Menu Option

In This Chapter	5-3
Selecting the TABLE Menu	5-4
Creating a Table	5-5
The CREATE TABLE Screen	5-6
Naming the Table	5-6
Building the Table Schema	5-7
Differences Between IBM Informix OnLine and IBM Informix SE	5-8
Exiting the CREATE TABLE Menu	5-8
Adding a Column to a Table	5-9
The ADD NAME Screen	5-9
Naming the Column	5-11
Defining the Column Data Type.	5-12
Specifying the Column Length or Other Parameters.	5-14
Defining an Index for the Column	5-15
Allowing Null Values in the Column	5-16
Adding to the Table Schema	5-17
Building the Table Schema	5-17
Modifying a Column in a Table	5-18
Modifying the Column Name	5-19
Modifying the Column Data Type	5-21
Modifying the Column Length	5-22
Modifying the Column Index	5-23
Modifying the Nulls Entry	5-24
Dropping a Column	5-25
Rebuilding the Table Schema	5-26
Displaying Subsequent Table Screens	5-27

Setting Table Options	5-27
Setting the Initial Extent Size	5-28
Setting the Next Extent Size	5-29
Determining the Lock Mode	5-30
Altering a Table	5-31
The ALTER TABLE Screen	5-31
Differences Between IBM Informix OnLine and IBM Informix SE	5-33
Exiting the ALTER TABLE Screen	5-33
Adding a Column with the ALTER TABLE Menu	5-34
Modifying a Column with the ALTER TABLE Menu	5-35
Dropping a Column with the ALTER TABLE Menu	5-37
Using the Screen Option on the ALTER TABLE Menu	5-38
Changing Table Options with the ALTER TABLE Menu	5-38
Displaying Table Information	5-39
Dropping a Table	5-42
Confirming Your Decision	5-43
Exiting the DROP TABLE Screen	5-43

In This Chapter

This chapter describes how to use the features of the Table option on the DB-Access Main Menu. Select this option if you want to create a table, change or drop an existing table, or get information about a table.

To perform table-management tasks, you can either enter an SQL statement, as described in [Chapter 3, “The Query-Language Menu Option,”](#) or select an option from the TABLE Menu. This chapter discusses the second method, which employs the DB-Access Interactive Schema Editor. (See [“Using the Interactive Schema Editor”](#) on page 2-7.)

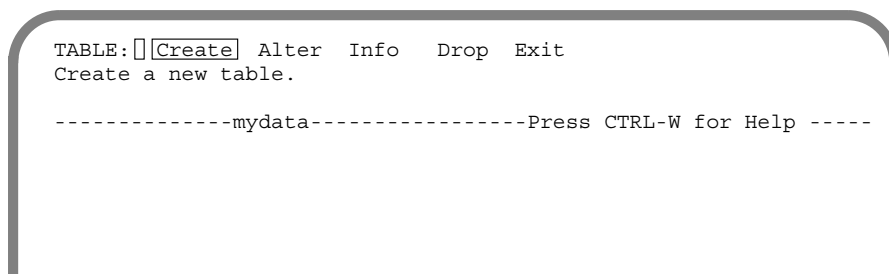
You must have a current database before you can work with database tables. See [Chapter 3](#) for details of how to create a database using SQL, or [Chapter 4, “The Database Menu Option,”](#) for a discussion of the Database option on the DB-Access Main Menu.

For a map of the Main Menu plus an overview of how to work with menu, text-entry, and HELP screens, refer to [Chapter 1, “Working with DB-Access.”](#)

Selecting the TABLE Menu

On the DB-Access Main Menu, type `t` or `T`, or highlight the Table option and press RETURN, to call up the TABLE Menu, as shown in [Figure 5-1](#).

Figure 5-1
The TABLE Menu



The TABLE Menu displays the following five options:

- | | |
|--------|--|
| Create | lets you use the Interactive Schema Editor to create a new table. (See “Creating a Table” on page 5-5.) |
| Alter | lets you use the Interactive Schema Editor to modify a table. (See “Altering a Table” on page 5-31.) |
| Info | retrieves and displays information about the structure of a table. (See “Displaying Table Information” on page 5-39.) |
| Drop | deletes a table from the database. (See “Dropping a Table” on page 5-42.) |
| Exit | exits the TABLE Menu and returns you to the DB-Access Main Menu. |

If there is no current database when you select the Table option, the SELECT DATABASE screen appears. You can select a listed database as the current database or go to the SQL Menu with the Interrupt key and create a database. (See [Chapter 4](#) for information on selecting and creating databases, or [Chapter 1](#) for how to select a database from the command line when you invoke DB-Access.)

Creating a Table

You use the *schema editor* to create a table when you use the Create option on the TABLE Menu. The schema editor appears in the lower portion of the screen. You design the schema for a table by defining each column in the table, one at a time. As you define each column, the schema editor prompts you for the necessary information.

As you type in information, the schema editor fills in the table schema. You move from left to right across the screen as you define each column, and from top to bottom as you define additional columns.

Each horizontal line in the editor represents one column, with the name of the column at the left. Each of the five headings in the editor presents some information about the column, such as the length of the column and the type of data it will store.

The schema editor allows you to perform the following functions:

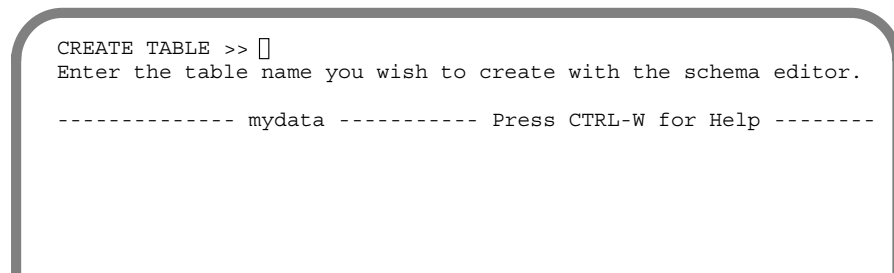
- Add a new column to the table
- Modify the currently highlighted column of the table
- Delete the currently highlighted column of the table
- Scroll the next screen of schema information into the schema editor (if the number of columns exceeds the capacity of the schema editor screen)
- Move the cursor through the lines and fields of the schema editor and highlight them with the arrow keys

You also can use the CREATE TABLE statement within SQL to create a table. See [Chapter 2, “Creating Databases and Tables,”](#) in this manual and the *IBM Informix Guide to SQL: Reference* for details.

The CREATE TABLE Screen

When you select the Create option on the TABLE Menu, DB-Access displays the CREATE TABLE screen, as shown in [Figure 5-2](#).

Figure 5-2
The CREATE TABLE screen



Naming the Table

At the prompt, type the name of the new table and press RETURN. You can assign any name you want, as long as you follow these guidelines:

- The name you assign to the table can be from 1 to 18 characters long, inclusive.
- The table name must begin with a letter. The rest of the name can consist of any combination of letters, numbers, and underscores (_).
DB-Access and the database servers make no distinction between uppercase and lowercase letters in table names. Therefore, **moorea**, **Moorea**, and **MOOREA** all refer to the same table.
- Certain keywords may cause potential ambiguities or syntax errors when used to name a database, table, or column. See the “Identifier” segment in the *IBM Informix Guide to SQL: Reference* for a discussion of these keywords and available workarounds.
- See the discussion of IBM Informix OnLine and IBM Informix SE differences in the next section for additional restrictions.

Building the Table Schema

Once you name the new table, DB-Access displays the CREATE TABLE Menu, as shown in [Figure 5-3](#).

Figure 5-3
The CREATE TABLE Menu with IBM Informix OnLine

```
CREATE TABLE clients:  Add Modify Drop Screen Table_options Exit
Adds columns to the table above the line with the highlight.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help -----

Column Name                Type                Length  Index Nulls

```

The CREATE TABLE Menu displays the following six options (with IBM Informix OnLine):

Add	adds a new column to the table.
Modify	modifies the structure of an existing column.
Drop	drops an existing column from the table.
Screen	scrolls down the screen and displays new text.
Table_options	sets the extent sizes and lock mode for the table.
Exit	exits the CREATE TABLE Menu and returns you to the TABLE Menu.



Tip: The CREATE TABLE Menu and its submenus behave differently than most DB-Access menus. You cannot use the arrow keys to move the cursor between menu options because the arrow keys control cursor movement in the schema editor. You must use the SPACEBAR to move between menu options.

From within the CREATE TABLE Menu, you cannot create a table with the following features:

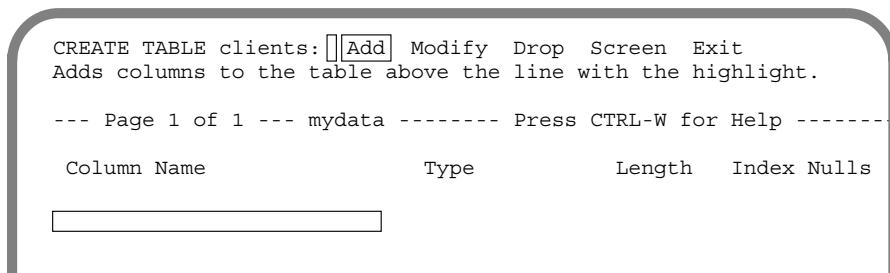
- Specified storage location (dbspace) for IBM Informix OnLine databases
- Specified pathname for IBM Informix SE databases
- Specified foreign or primary key

To create tables with these features, you must use the SQL statement CREATE TABLE or ALTER TABLE. the *IBM Informix Guide to SQL: Reference* for a discussion of ALTER TABLE and CREATE TABLE.

Differences Between IBM Informix OnLine and IBM Informix SE

Figure 5-4 shows how the CREATE TABLE Menu looks if you are using IBM Informix SE.

Figure 5-4
The CREATE TABLE Menu with IBM Informix SE



With IBM Informix SE, the CREATE TABLE Menu displays all the same options as it does with IBM Informix OnLine, except for Table_options.

Exiting the CREATE TABLE Menu

To leave the CREATE TABLE Menu without creating a table, press the Interrupt key. You return to the TABLE Menu.

Adding a Column to a Table

To add a new column to a table, you must define a new line in the schema editor. DB-Access prompts you to enter the following information about each column in a table:

- The column name
- The column data type
- The column length (if required)
- An index for the column
- Whether or not the column allows null values

You define all these column attributes through the ADD NAME screen.

From within the DB-Access schema editor, you cannot add a column with the following features:

- Unique constraints
- Referential constraints
- Check constraints
- Default values

To create columns with these features, you must use the SQL statement CREATE TABLE or ALTER TABLE, as described in [Chapter 2](#) of this manual and the *IBM Informix Guide to SQL: Reference*.

The ADD NAME Screen

Select the Add option on the CREATE TABLE (or ALTER TABLE) Menu, and DB-Access displays the ADD NAME screen with the highlight under the Column Name, as shown in [Figure 5-5](#).

Figure 5-5

The ADD NAME screen for defining columns in a table

```
ADD NAME >> 
Enter column name. RETURN adds it.
INTERRUPT returns to CREATE/ALTER menu.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help ----

Column Name                Type                Length    Index Nulls

```

Use the ADD NAME screen to define characteristics of a table, as follows:

- Column Name** names the column you want to add to the table.
- Type** defines the column data type.
- Length** sets the column length.
- Index** defines an index for the column.
- Nulls** allows or disallows null values in the column.

You must move the highlight through each field of the screen for each column you are adding to the table. Press RETURN or an arrow key to accept the default entry for each field.

As you enter information in each field, the highlight moves from left to right across the screen. As the table schema is created, the data is displayed on the screen and the next ADD screen appears.

At any time, you can press the Interrupt key to return to the previous ADD screen without adding anything on the current screen.

You can change any of the entries in a line (before you move to the next line) by using the left and right arrow keys to move through the line. The appropriate ADD screen appears and you can change your entry. Once you move the highlight to the next line, however, you must use the Modify option on the CREATE TABLE Menu to change your entry, as described later in this chapter.

Naming the Column

Type the name of the column after the prompt at the top of the ADD NAME screen and press RETURN. The column name is filled in, as shown in [Figure 5-6](#).

Figure 5-6

The ADD NAME screen with column name entered

```

ADD NAME >> 
Enter column name. RETURN adds it.
INTERRUPT returns to CREATE/ALTER menu.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help ----

Column Name                Type                Length  Index Nulls
-----
customer_num

```

You can assign any name you want, as long as you follow these guidelines:

- The name you assign to the column can be from 1 to 18 characters long, inclusive.
- The column name must begin with a letter. The rest of the name can consist of any combination of letters, numbers, and underscores (_). DB-Access and the database servers make no distinction between uppercase and lowercase letters in column names. Therefore, **kava**, **Kava**, and **KAVA** all refer to the same column.
- Certain keywords may cause potential ambiguities or syntax errors when used to name a database, table, or column. See the “Identifier” segment in the *IBM Informix Guide to SQL: Reference* for a discussion of these keywords and available workarounds.

DB-Access enters this name under Column Name, displays the ADD TYPE screen, and moves the highlight to the Type field.

Defining the Column Data Type

Once you have entered the column name, DB-Access displays the ADD TYPE screen. Here you can specify the data type for the new column, as shown in [Figure 5-7](#).

Figure 5-7

The ADD TYPE screen for defining column data types with IBM Informix OnLine

```
ADD TYPE clients :  Number Serial Date Money date-Time Interval
Permits any combination of letters, numbers, and symbols.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help ----

Column Name          Type          Length  Index  Nulls
customer_num         
```

The ADD TYPE screen displays various data types, depending on the database server. Select the data type for the column by typing in the first capitalized letter of the data type, using either uppercase or lowercase letters, or use the SPACEBAR to highlight it and then press RETURN. *Do not try to use the ARROW keys to move to your choice.*

An overview of all the data types that you can assign from the ADD TYPE screen follows. Some of these data types require that you enter additional information; this is discussed in the next section, [“Specifying the Column Length or Other Parameters” on page 5-14](#).

Database data types are described in detail in the *IBM Informix Guide to SQL: Reference*. The use of data types in the CREATE TABLE and ALTER TABLE statements is described there.

Char	CHAR data type (also known as CHARACTER): stores any string of letters, numbers, and other printable characters.
Number	Any of these five data types: INTEGER data type (also known as INT): stores whole numbers that range from -2,147,483,647 to +2,147,483,647. SMALLINT data type: stores small whole numbers that range from -32,767 to +32,767. DECIMAL data type (also known as DEC or NUMERIC): stores decimal floating-point numbers up to a maximum of 32 significant digits, with definable precision and scale. SMALLFLOAT data type (also known as REAL): stores single-precision, floating-point numbers with approximately 8 significant digits. FLOAT data type (also known as DOUBLE PRECISION): stores double-precision, floating-point numbers with up to 16 significant digits.
Serial	SERIAL data type: stores a positive sequential integer maintained by the database.
Date	DATE data type: stores a calendar date.
Money	MONEY data type: stores a currency amount.
Datetime	DATETIME data type: stores an instant in time expressed as a calendar date and time of day.
Interval	INTERVAL data type: stores a value that represents a span of time.

Variable-length

There are three variable-length data types, available only on IBM Informix OnLine:

VARCHAR data type: stores a character string of varying length ranging in size from 0 to 255 bytes.

TEXT data type: stores any kind of text data, with the maximum determined by available disk storage.

BYTE data type: stores any kind of binary data in an undifferentiated byte stream.

For **TEXT** or **BYTE**: You must specify a table and/or a blob-space name. If you choose **Table**, the column data is stored in the same **tblspace** as the other columns. If you choose **BLOB-Space-name**, you can enter the name of any valid blob-space at the prompt.

For complete information on assigning and changing database data types with the **CREATE TABLE** and **ALTER TABLE** statements, see the *IBM Informix Guide to SQL: Reference*, and the *IBM Informix Guide to SQL: Tutorial*.

If a data type requires that you enter additional information, DB-Access displays a screen that prompts you for this information. For example, defining a column as type **CHARACTER** causes DB-Access to display the **ADD LENGTH** screen so you can specify the length of the **CHARACTER** field. Defining a **TEXT** or **BYTE** type causes DB-Access to display the **ADD BLOBSPACE** screen.

Specifying the Column Length or Other Parameters

When you specify various data types on the **ADD TYPE** screen, a new **ADD** screen appears, asking you to specify the column length or other defining characteristics. You must enter the following information in the **Length** field for these data types:

Char Specify a length (the default is 20).

Number For **Decimal** type only, enter the length and scale (the default is 16,2).

Serial Enter the starting number (the default is 1).

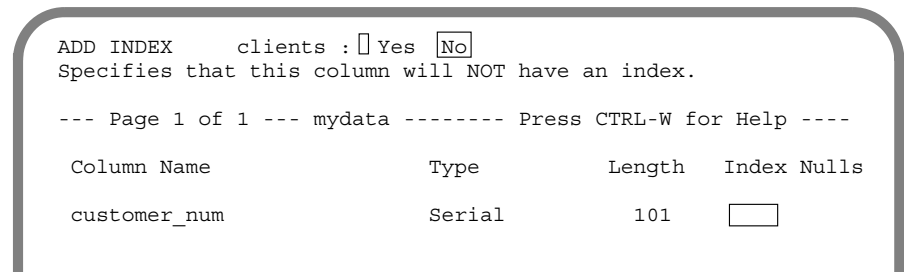
Money	Specify a length (the default is 16, 2).
Datetime	Specify first to last datetime qualifiers.
Interval	Specify first to last interval qualifiers.
Varchar	Specify a maximum length (from 1 to 255 bytes) and a minimum space (from 0 to 255 bytes).

Others data types, such as Date, do not require a length entry. The highlight skips to the Index field, leaving the Length field blank.

Defining an Index for the Column

After you define the column name and data type, you can indicate whether you want an index on the column. DB-Access displays the ADD INDEX screen, shown in [Figure 5-8](#), for this purpose.

Figure 5-8
The ADD INDEX screen



The ADD INDEX screen displays the following two options:

- Yes creates an index on the column.
- No does not create an index on the column.

The default is No. Press RETURN, or type n or N, and the highlight moves to the Nulls field, leaving the Index field blank.

Type y or Y to select the Yes option. DB-Access then displays an ADD DUPLICATES screen and prompts you to indicate whether the index can allow duplicate values. The default on this screen is Yes.

- Press RETURN, or enter y or Y and the word Dups appears in the Index field.
- Enter n or N and the word Unique appears in the Index field.

The highlight moves to the next field.

This option creates a nonclustered, ascending index for the values in the associated database column. To create a different type of index, use the SQL statement CREATE INDEX. See the discussion of indexing in the *IBM Informix Guide to SQL: Tutorial*.

Allowing Null Values in the Column

After you define the column name, data type, and index, you can specify whether the column allows null values. DB-Access displays the ADD NULLS screen, shown in [Figure 5-9](#), for this purpose.

Figure 5-9
The ADD NULLS screen

```
ADD NULLS      clients :  No
Permits null values in this column.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help ----

Column Name           Type           Length  Index Nulls
customer_num          Serial         101     Unique 
```

The ADD NULLS screen displays the following two options:

Yes allows null values in the column.

No does not allow null values in the column.

The default is Yes. Press RETURN, or type y or Y, to allow null values in that column and move the highlight to the next column-definition line. Type n or N to specify that null values are not allowed and move the highlight to the next column-definition line.

Adding to the Table Schema

Once you specify the name, data type, length, index, and null values for a column, you have completed the column definition. Press RETURN to move the highlight down to the next line in the schema editor. DB-Access redisplay the ADD NAME screen. From this screen, you can perform the following functions:

- Add another column definition to the table.
- Return to the CREATE TABLE Menu by pressing Interrupt or an arrow key.

Building the Table Schema

When you complete the schema for the new (or modified) table, select the Exit option on the CREATE TABLE (or ALTER TABLE) Menu. DB-Access displays the EXIT Menu, as shown in [Figure 5-10](#).

Figure 5-10

The EXIT screen used for building a new table

```
EXIT clients: [Build-new-table] Discard-new-table
Builds a new table and returns to the Table Menu.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help ----

Column Name           Type           Length  Index Nulls
customer_num          Serial         101     Unique No
fname                 Char           15      Yes
lname                 Char           15      Yes
company               Char           20      Yes
address1              Char           20      Yes
address2              Char           20      Yes
city                  Char           15      Yes
state                 Char           2       Yes
zipcode               Char           5       Dups  Yes
phone                 Char           18      Yes
```

The EXIT Menu lists the following two options:

Build-new-table builds the table with the new column definitions.

Discard-new-table discards the new table instructions.

The default is Build-new-table. Press RETURN to create the table and return to the TABLE Menu. Select the Discard-new-table option with an arrow key (or type d or D and press RETURN) to return to the TABLE Menu without saving the new or modified table definition.

Modifying a Column in a Table

The procedure for using the schema editor to modify a column in a table is the same as previously described for creating a column in a table. The screen names now say *modify* instead of *add*.

You can modify the following fields in the table schema:

Column Name changes the column name.

Type changes the column data type.

Length changes the column length.

Index redefines an index for the column.

Nulls allows or disallows null values in the column.

To modify an existing column, follow these steps:

1. Use the arrow keys to highlight the column definition that you want to modify.
2. Select the Modify option on the CREATE TABLE (or ALTER TABLE) Menu and press RETURN.

- DB-Access displays the appropriate screen for the field you want to modify:

Field to Modify	Screen Name
Column Name	MODIFY NAME
Column Data Type	MODIFY TYPE
Column Length	MODIFY LENGTH
Index on Column	MODIFY INDEX
Column Allows Nulls?	MODIFY NULLS

- Type your field modification on the top line of the screen and press RETURN.
- Move the highlight to the next field as needed and repeat the process. For details, see the earlier discussion of the ADD screens for each field in the table schema, beginning on [5-9](#).
- When you select Exit to leave the screen, you see the EXIT Menu. Press RETURN to select the default, Build-new-table, or type `d` or `D` or move the cursor to the Discard-new-table option with an arrow key and press RETURN. DB-Access accordingly builds or discards the table and returns you to the TABLE Menu.

You can press the Interrupt key at any time to leave a Modify screen or menu without making any changes.

The Modify screens function much the same as their Add screen counterparts discussed earlier in this chapter. The Modify screens are described briefly in the following sections. For more information, see the sections relating to the Add screens beginning on page [5-9](#).

Modifying the Column Name

Use the MODIFY NAME screen to change the name of a column in the table.

Highlight the column name that you want to change and select the Modify option on the CREATE TABLE (or ALTER TABLE) Menu. The MODIFY NAME screen appears, as shown in [Figure 5-11](#).

Figure 5-11
The MODIFY NAME screen

```

MODIFY NAME >> 
Enter column name. RETURN adds it.
INTERRUPT returns to CREATE/ALTER menu.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help ----

Column Name          Type          Length      Index Nulls
-----
customer_num         Serial        101         Unique No
fname                Char          15          Yes
lname                Char          15          Yes
company              Char          20          Yes
address1             Char          20          Yes
address2             Char          20          Yes
city                 Char          15          Yes
state                Char          2           Yes
zipcode              Char          5           Dups Yes
phone                Char          18          Yes
    
```

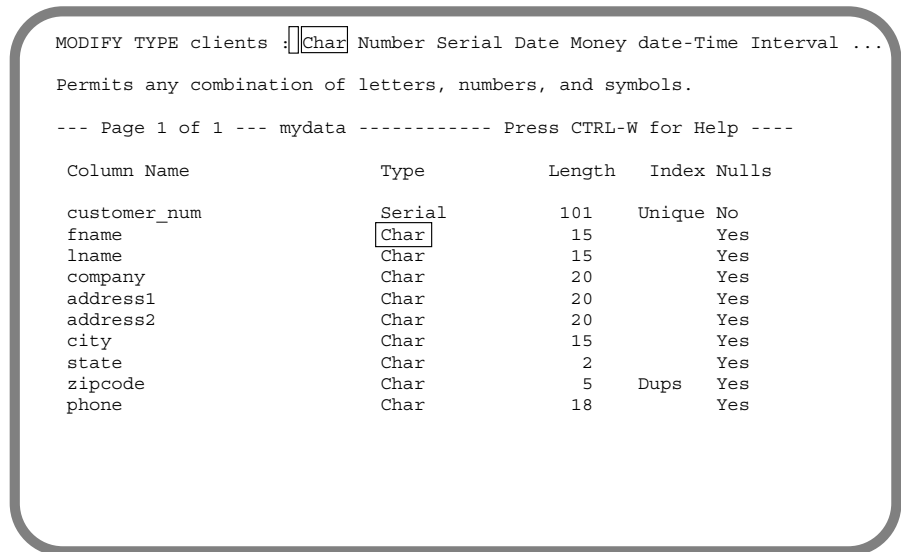
Type the new name after the prompt at the top of the MODIFY NAME screen and press RETURN. You can assign any name you want, as long as you follow the guidelines outlined in [“Naming the Column” on page 5-11](#).

DB-Access changes this name under Column Name and redisplay the CREATE TABLE screen so you can make more changes.

Modifying the Column Data Type

Move the highlight to an entry in the Type field for a column and select Modify. DB-Access displays the MODIFY TYPE screen, shown in Figure 5-12, where you can change the data type for that column.

Figure 5-12
The MODIFY TYPE screen on IBM Informix OnLine



The MODIFY TYPE screen displays various data types, depending on the database server. Select the data type for the column by typing in the first capitalized letter of the data type (using either uppercase or lowercase letters), or use the SPACEBAR. *Do not try to use the arrow keys.*

DB-Access changes this data type under Type and redisplay the CREATE TABLE screen so you can make additional changes.

Database data types are described in detail in the *IBM Informix Guide to SQL: Reference*. For information on assigning and changing database data types with the CREATE TABLE and ALTER TABLE statements, see the *IBM Informix Guide to SQL: Reference*, and the *IBM Informix Guide to SQL: Tutorial*.

Modifying the Column Length

Move the highlight to an entry in the Length field of a column and select Modify. DB-Access displays the MODIFY LENGTH screen, as shown in [Figure 5-13](#).

Figure 5-13
The MODIFY LENGTH screen

```
MODIFY LENGTH >> 
Enter column length. RETURN adds it

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help -----

Column Name          Type          Length      Index Nulls
customer_num         Serial        101         Unique No
fname                 Char          15          Yes
lname                 Char          15          Yes
company               Char          20          Yes
address1              Char          20          Yes
address2              Char          20          Yes
city                  Char          15          Yes
state                 Char          2           Yes
zipcode               Char          5           Dups Yes
phone                 Char          18          Yes
```

If you modify certain entries in the Type field, the highlight moves to this field automatically. See the discussion in [“Specifying the Column Length or Other Parameters”](#) on page 5-14.

Type in the new length and press RETURN. DB-Access changes the entry under Length and redisplay the CREATE TABLE screen so you can make further modifications.

Modifying the Column Index

Move the highlight to an entry in the Index field of a column and select Modify. DB-Access displays the MODIFY INDEX screen, as shown in [Figure 5-14](#).

Figure 5-14
The MODIFY INDEX screen

```

MODIFY INDEX      clients :  Yes  No
Specifies that this column will NOT have an index.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help ----

Column Name      Type           Length  Index Nulls
customer_num     Serial         101     Unique No
fname            Char           15       Yes
lname            Char           15       Yes
company          Char           20       Yes
address1         Char           20       Yes
address2         Char           20       Yes
city             Char           15       Yes
state            Char            2       Yes
zipcode          Char            5      Dups    Yes
phone            Char           18       Yes
    
```

The MODIFY INDEX screen has the following two options:

Yes creates an index on the column.

No does not create an index on the column.

Press RETURN and select Yes or No on the MODIFY INDEX screen. If you specify No, an existing index is dropped. If you specify Yes, the MODIFY DUPLICATES screen appears. Here, you must indicate whether to allow duplicate values.

- Enter y or Y, and the word Dups appears in the Index field.
- Enter n or N, and the word Unique appears in the Index field.

DB-Access changes the entry under Index and redisplay the CREATE TABLE screen so you can continue to modify the table schema.

This option creates a nonclustered, ascending index for the values in the associated database column. To create a different type of index, use the SQL statement CREATE INDEX; to change clustering for an index, use the ALTER INDEX statement. See the discussion of indexing in the *IBM Informix Guide to SQL: Tutorial*.

Modifying the Nulls Entry

Use the ARROW keys to highlight an entry in the Nulls field of a column and select Modify. DB-Access displays the MODIFY NULLS screen, as shown in [Figure 5-15](#).

Figure 5-15
The MODIFY NULLS screen

```
MODIFY NULLS      clients :  Yes  No
Permits null values in this column.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help ----

Column Name          Type          Length  Index Nulls
customer_num        Serial        101     Unique No
fname                Char          15       Yes
lname                Char          15      Yes
company              Char          20      Yes
address1             Char          20      Yes
address2             Char          20      Yes
city                  Char          15      Yes
state                Char           2      Yes
zipcode              Char           5      Dups  Yes
phone                Char          18      Yes
```

The MODIFY NULLS screen displays the following two options:

Yes allows null values in the column.

No does not allow null values in the column.

Type `y` or `Y` to allow null values in that column; type `n` or `N` to specify no null values. DB-Access changes the entry under Nulls and redisplay the CREATE TABLE screen.

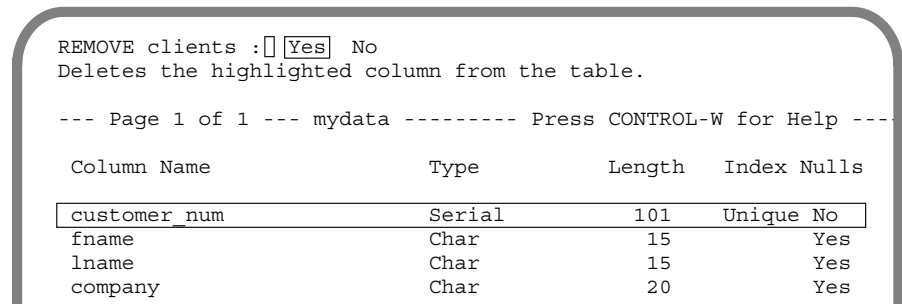
You cannot modify a column in a table to disallow nulls if any existing rows in the table have null values in that column.

Dropping a Column

To delete a column from a table schema follow these steps:

1. Position the highlight anywhere on the column that you want to drop.
2. Select the Drop option on the CREATE TABLE (or ALTER TABLE) Menu. Some or all of that column line is then highlighted on the screen.
3. DB-Access displays the REMOVE Menu, shown in [Figure 5-16](#), where you verify your decision.

Figure 5-16
The REMOVE Menu



The REMOVE Menu displays the following two options:

- Yes deletes the line currently highlighted in the schema editor.
- No does not delete the line currently highlighted in the schema editor.

The default is Yes. If you decide that you do not want to drop the highlighted column, type N or use the right arrow key to highlight No and press RETURN. DB-Access drops the column and all its data if you specify Yes and select the Build-new-table option on the EXIT Menu.

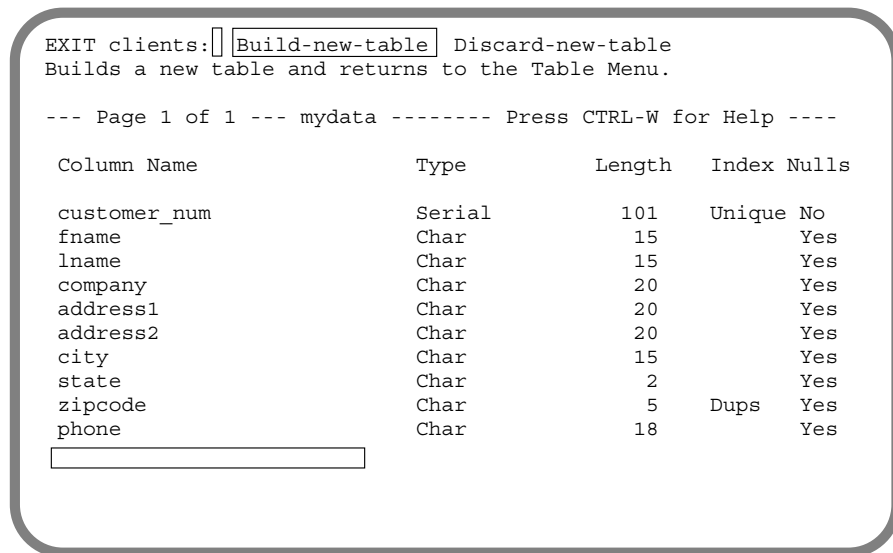
DB-Access returns you to the TABLE Menu.

Rebuilding the Table Schema

When you have completed your column modifications, you must rebuild the table. Select the Exit option on the CREATE TABLE (or ALTER TABLE) Menu. DB-Access displays the EXIT Menu, as shown in [Figure 5-17](#).

Figure 5-17

The EXIT screen used to build or rebuild a table



The EXIT Menu lists the following two options:

Build-new-table builds the table with the modified column definitions.

Discard-new-table discards the modified table instructions.

The default is Build-new-table. Press RETURN to modify the table and return to the TABLE Menu. Select the Discard-new-table option with an arrow key (or type d or D and press RETURN) to return to the TABLE Menu without saving the modified table definition.

Displaying Subsequent Table Screens

Select the Screen option on the CREATE TABLE Menu to display the next screen of column definitions in the schema editor. If you choose Select at the last screen of definitions, DB-Access redisplays the first screen.



Tip: If the table does not contain more than one screenful of columns, the Select option has no effect.

Setting Table Options

Select the Table_options option on the CREATE TABLE (or ALTER TABLE) Menu to set the extent sizes and/or lock mode for a table in an IBM Informix OnLine database. The TABLE OPTIONS Menu appears and guides you through the steps necessary to set various options for the table, as shown in [Figure 5-18](#).

Figure 5-18
The TABLE OPTIONS Menu

```

TABLE OPTIONS clients: [eXtent_size] Next_size Lock_mode Exit
Specify an initial extent size for the table.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help ----

Column Name                Type                Length  Index Nulls
[ ]
    
```

The TABLE OPTIONS Menu displays the following four options:

eXtent_size specifies the initial extent size for the table.

Next_size specifies the next extent size for the table .

Lock_mode specifies the lock mode for the table (Page or Row).

Exit exits the TABLE OPTIONS Menu and returns you to the CREATE TABLE Menu .

Setting the Initial Extent Size

When you create a table using IBM Informix OnLine, you can specify how much disk space you want to reserve for the table. IBM Informix OnLine stores data in units called *disk pages*. It manages these disk pages dynamically by reserving an *initial extent size*. When this initial extent becomes full, IBM Informix OnLine reserves a *next extent*. When that extent becomes full, IBM Informix OnLine adds another extent of *next extent* size, and so on.

You can find additional information about IBM Informix OnLine table sizing in several manuals. For information on general allocation strategies, see the *IBM Informix OnLine Administrator's Guide*. For the calculations involved in choosing extent sizes, see the *IBM Informix Guide to SQL: Tutorial*.

Specifying extent sizes is optional; the default size for both initial and additional extents is 8 one-kilobyte blocks. However, if you use the default extent sizes for a large table, you may exceed the maximum number of extents allowed per tblspace, which causes an out-of-extents error. In addition, the default value may vary, depending on your platform.

To specify an initial extent size, select the eXtent_size option on the TABLE OPTIONS Menu. (Type x or X or simply press RETURN.) DB-Access displays the Extent Size screen, as shown in [Figure 5-19](#).

Figure 5-19
The Extent Size screen for managing disk space

```

Extent Size >> 
Specify an initial extent size in kilobytes.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help ----

Column Name          Type          Length   Index Nulls


```

At this screen, you can perform either of the following actions:

- Press RETURN to specify the default extent size of 8 one-kilobyte blocks.
- Enter an initial extent size (in kilobytes) at the prompt and press RETURN.

The minimum initial extent size is 4 one-kilobyte blocks. For more information on calculating table and extent sizes, see the *IBM Informix Guide to SQL: Tutorial*.

Setting the Next Extent Size

When you select the Next_size option on the TABLE OPTIONS Menu, DB-Access displays the Next Size screen, as shown in [Figure 5-20](#):

Figure 5-20
The Next Size screen for managing disk space

```

Next Size >> 
Specify next extent size in kilobytes.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help ----

Column Name          Type          Length   Index Nulls


```

On this screen, you can specify the size for the next extent (and subsequent extents) to be allocated when the previous extent is full. You can perform either of the following actions:

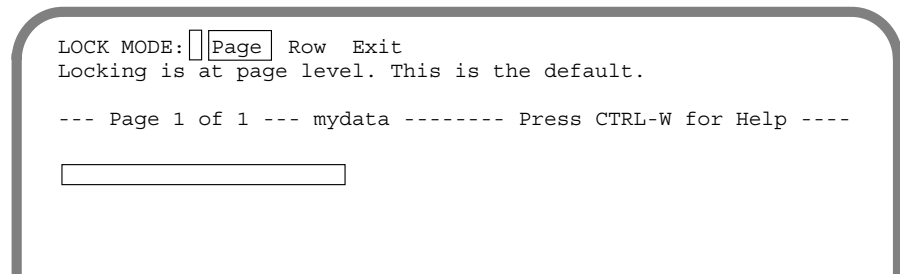
- Press RETURN to specify the default next extent size of 8 one-kilobyte blocks.
- Enter the next extent size (in kilobytes) at the prompt and press RETURN.

The minimum next extent size is 4 one-kilobyte blocks. For more information on calculating table and extent sizes, see the *IBM Informix Guide to SQL: Tutorial*.

Determining the Lock Mode

When you select the Lock_mode option on the TABLE OPTIONS Menu, DB-Access displays the LOCK MODE Menu, as shown in [Figure 5-21](#).

Figure 5-21
The LOCK MODE Menu



From this menu, you can choose the mode used when the database locks the rows in a table. The LOCK MODE Menu displays the following three options:

- | | |
|------|--|
| Page | specifies that a row will be locked by locking the entire page on which it resides. |
| Row | specifies that rows will be locked individually. |
| Exit | exits to the TABLE OPTIONS Menu. Unless you specify row-level locking before you exit, the default (Page) is used. |

One row of a table is the smallest object that you can lock. A disk page contains one or more rows of a table. In some cases, performance is enhanced by locking a disk page rather than individual rows on the page.

You choose between locking by rows or pages when you create a table. Whenever IBM Informix OnLine needs to lock a row, it locks either just the row itself or the entire page or pages that the row occupies, depending on the lock mode established for the table.

For more information on locking, see the *IBM Informix Guide to SQL: Tutorial*.

Altering a Table

This section describes how to use the Alter option on the TABLE Menu to alter an existing table schema. For information on using the ALTER TABLE statement within SQL, refer to [Chapter 2](#) in this manual and refer to the *IBM Informix Guide to SQL: Reference*.

The following sections describe one way to alter a table with the schema editor, by using the ALTER TABLE Menu. You can achieve similar results by using the Modify option on the CREATE TABLE Menu, as described in the section [“Modifying a Column in a Table”](#) on page 5-18.

The ALTER TABLE Screen

Select the Alter option on the TABLE Menu and DB-Access displays the ALTER TABLE screen, as shown in [Figure 5-22](#).

Figure 5-22
The ALTER TABLE screen

```

ALTER TABLE >> 
Enter the table name you wish to alter with the schema editor.

----- mydata ----- Press CTRL-W for Help -----

items
orders
    
```

This screen displays the names of tables defined in the current database.

At the prompt, enter the name of the table to alter (or use the arrow keys to highlight the table name in the list). DB-Access displays the ALTER TABLE Menu, as shown in [Figure 5-23](#).

Figure 5-23

The ALTER TABLE Menu with IBM Informix OnLine

```
ALTER TABLE clients:  Modify Drop Screen Table_options Exit
Adds columns to the table above the line with the highlight.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help -----

Column Name                Type           Length  Index  Nulls
-----
customer_num                Serial         101     Unique No
fname                       Char           15      Yes
lname                       Char           15      Yes
company                     Char           20      Yes
```

The ALTER TABLE Menu displays the following six options (with IBM Informix OnLine):

- Add adds a new column to the table.
- Modify modifies the structure of an existing column.
- Drop drops an existing column from the table.
- Screen scrolls down the screen and displays new text.
- Table_options sets extent sizes and the lock mode for the table.
- Exit exits the ALTER TABLE Menu and returns you to the TABLE Menu.



Tip: The ALTER TABLE Menu and its submenus behave differently from most DB-Access menus. You cannot use the arrow keys to move the cursor between menu options because the arrow keys control cursor movement in the schema editor. You must use the SPACEBAR to move between menu options.

The ALTER TABLE options work just like their counterparts in the CREATE TABLE Menu, except that you cannot change the initial extent size. This extent is set when you first create the table.

You also can use the ALTER TABLE statement within SQL to modify a table. For details, see [Chapter 2](#) in this manual and the *IBM Informix Guide to SQL: Reference*.

Differences Between IBM Informix OnLine and IBM Informix SE

[Figure 5-24](#) shows how the ALTER TABLE Menu appears if you are using IBM Informix SE.

Figure 5-24
The ALTER TABLE Menu with IBM Informix SE

```
ALTER TABLE clients:  Modify Drop Screen Exit
Adds columns to the table above the line with the highlight.

--- Page 1 of 1 --- mydata ----- Press CTRL-W for Help -----

Column Name                Type           Length  Index  Nulls
-----
customer_num                Serial         101     Unique No
fname                       Char           15      Yes
lname                       Char           15      Yes
company                     Char           20      Yes
```

With IBM Informix SE, the ALTER TABLE Menu displays all the same options as IBM Informix OnLine, except for Table_options.

Exiting the ALTER TABLE Screen

To leave the ALTER TABLE screen without altering a table, press the Interrupt key. You return to the TABLE Menu.

Adding a Column with the ALTER TABLE Menu

The procedure for using the schema editor to add a column to a table with the ALTER TABLE Menu is the same as previously described in this chapter for adding a column to a table with the CREATE TABLE Menu. Briefly, to add a column, follow these steps:

1. Select the Add option on the ALTER TABLE Menu. Use the arrow keys to move the highlight to the line where you want to add a column. Press RETURN.
2. DB-Access displays the following screens, in order, for the column you want to add:

Field to Add	Screen Name
Column Name	ADD NAME
Column Data Type	ADD TYPE
Column Length	ADD LENGTH
Index on Column	ADD INDEX
Column Allows Nulls?	ADD NULLS

3. Type your column definitions on the top line of the screen and press RETURN.
4. Move the highlight to the next column definition field as needed and repeat the process.
5. When you select Exit to leave the screen, you see the EXIT Menu. Press RETURN to select the default, Build-new-table, or type `d` or `D` or move the cursor to the Discard-new-table option with an arrow key and press RETURN. DB-Access accordingly builds or discards the table and returns you to the TABLE Menu.

You can press the Interrupt key at any time to leave an ADD screen without making any changes.

From within the DB-Access schema editor, you cannot alter a table to add a column with the following features:

- Unique constraints
- Referential constraints
- Check constraints
- Default values

To add columns with these features, you must use the SQL statement CREATE TABLE or ALTER TABLE, as described in [Chapter 2](#) of this manual and the *IBM Informix Guide to SQL: Reference*.

Modifying a Column with the ALTER TABLE Menu

The procedure for using the schema editor to modify a column in a table using the ALTER TABLE Menu is the same as previously described in this chapter for modifying a column in a table with the CREATE TABLE Menu. Briefly, to modify a column, follow these steps:

1. Use the arrow keys to move the highlight to the line and column definition that you want to modify. Select the Modify option on the ALTER TABLE Menu.
2. DB-Access displays the appropriate screen for the column you want to modify, as follows:

Field to Modify	Screen Name
Column Name	MODIFY NAME
Column Data Type	MODIFY TYPE
Column Length	MODIFY LENGTH
Index on Column	MODIFY INDEX
Column Allows Nulls?	MODIFY NULLS

3. Type each modification on the top line of the screen and press RETURN. DB-Access makes the change and redisplay the ALTER TABLE Menu. For certain column definitions, DB-Access first displays another MODIFY screen.

If you modify a data type, you might need to enter additional information. DB-Access displays a screen that prompts you for this information. For example:

- Changing a column to the CHARACTER data type causes DB-Access to display the MODIFY LENGTH screen so you can specify the length of the CHARACTER field.
- Changing a column to a DATETIME data type causes the MODIFY DATETIME QUALIFIER screens to appear.
- Changing a column to an INTERVAL data type causes the MODIFY INTERVAL QUALIFIER screens to appear.
- Changing a column to a VARCHAR data type causes the MODIFY MAXIMUM LENGTH and MODIFY MINIMUM SPACE screens to appear.
- Changing a column to a TEXT or BYTE data type causes the MODIFY BLOBSPACE screen to appear.

If you modify a unique index on a column, you see a screen that asks you to MODIFY DUPLICATES.

Data types and their requirements are described in detail in the *IBM Informix Guide to SQL: Reference*. See the discussion of indexing in the *IBM Informix Guide to SQL: Reference* and the *IBM Informix Guide to SQL: Tutorial*.

4. Move the highlight to the next field as needed and repeat the process.
5. When you select Exit to leave the screen, you see the EXIT Menu. Press RETURN to select the default, Build-new-table, or type d or D or move the cursor to the Discard-new-table option with an arrow key and press RETURN. DB-Access accordingly builds or discards the table and returns you to the TABLE Menu.

You can press the Interrupt key at any time to leave a MODIFY screen without making any changes.

Dropping a Column with the ALTER TABLE Menu

To delete a column from a table schema, follow these steps:

1. Position the highlight anywhere on the column that you want to drop.
2. Select the Drop option on the ALTER TABLE Menu. Some or all of that column line is then highlighted on the screen.

DB-Access displays the REMOVE Menu, as shown in [Figure 5-25](#).

Figure 5-25
The REMOVE Menu

```

REMOVE clients :  Yes  No
Deletes the highlighted column from the table.

--- Page 1 of 1 --- mydata ----- Press CONTROL-W for Help ---

Column Name          Type          Length  Index Nulls
-----
customer_num         Serial         101     Unique No
fname                 Char           15      Yes
lname                 Char           15      Yes
company               Char           20      Yes
    
```

The REMOVE Menu displays the following two options:

- Yes deletes the column currently highlighted in the schema editor.
- No does not delete the column currently highlighted in the schema editor.

The default is Yes. If you decide that you do not want to drop the highlighted column, type N or use the right arrow key to highlight No and press RETURN. DB-Access drops the column *and all its data* if you exit the screen and select the Build-new-table option on the EXIT Menu.

DB-Access returns you to the TABLE Menu.



Using the Screen Option on the ALTER TABLE Menu

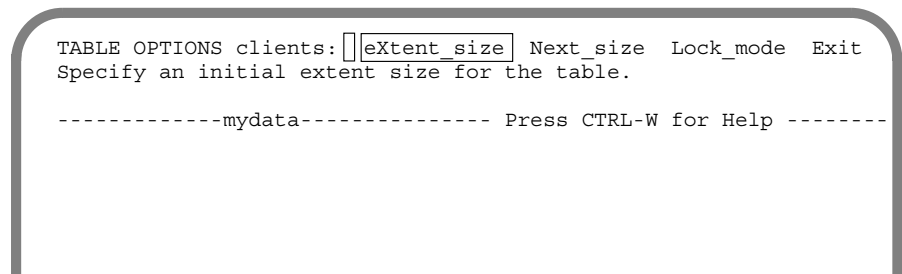
Select the Screen option on the ALTER TABLE Menu to display the next screen of column definitions in the schema editor. If you choose Screen at the last screen of definitions, DB-Access redisplay the first screen.

Tip: If the table does not contain more than one screenful of column definitions, the Select option has no effect.

Changing Table Options with the ALTER TABLE Menu

Select the Table_options option on the ALTER TABLE Menu to change the next extent size or lock mode for a table in an IBM Informix OnLine database. The TABLE OPTIONS Menu appears and guides you through the steps necessary to modify various options for the table, as shown in [Figure 5-26](#).

Figure 5-26
The TABLE OPTIONS Menu



The TABLE OPTIONS Menu displays the following four options:

- eXtent_size *You cannot change the initial extent size; it is set when you first create the table.*
- Next_size specifies the next extent size for the table .
- Lock_mode specifies the lock mode for the table (Page or Row).
- Exit exits the TABLE OPTIONS Menu and returns you to the ALTER TABLE Menu.



Tip: You can change only one option on the TABLE_OPTIONS Menu during each ALTER TABLE session. If you have changed the extent size or locking mode and want to change another table option, you must exit the ALTER TABLE Menu, build the modified table, and then start another ALTER TABLE session.

For details of how to change the next extent size and locking mode, see the section “Setting Table Options” on page 5-27. For additional information on calculating table sizes and locking, see the *IBM Informix Guide to SQL: Tutorial*.

Displaying Table Information

Use the Info option on the TABLE Menu to display information about table columns, indexes, privileges, and status.

The Info option is also available from the SQL Menu, as described in [Chapter 3](#). In addition, you can use the SQL statement INFO to display the same kinds of table information. See the *IBM Informix Guide to SQL: Reference* for details on the INFO statement.

Select the Info option on the TABLE Menu, and DB-Access displays the INFO FOR TABLE screen, as shown in [Figure 5-27](#).

Figure 5-27
The INFO FOR TABLE screen

```

INFO FOR TABLE >> 
Choose a table with the Arrow Keys, or enter a name, then press
Return.

----- mydata ----- Press CTRL-W for Help -----


customer
orders
  
```

This screen lists the names of tables that exist in the current database. If you are not the owner of a table, the table name is prefixed by the owner name, as in "june".clients.

To leave the INFO FOR TABLE screen without requesting table information, press the Interrupt key. You return to the TABLE Menu.

You can select a table in one of the following two ways:

- Type its name and press RETURN. You must use this method and include the full pathname if you want information on a table that is not in the current database.
- Use the arrow keys to highlight the name of the table you want, then press RETURN.

For example, for the **customer** table, type in **customer** or highlight it with an ARROW key, and then press RETURN. The INFO Menu appears, with **customer** in the top line, as shown in [Figure 5-28](#).

Figure 5-28
The INFO Menu for displaying table information

```
INFO - customer: [Columns] Indexes Privileges References
Status Table Exit
Display column names and data types for a table.

----- mydata ----- Press CTRL-W for Help -----
```


The INFO Menu displays the following seven options:

Columns	lists all the columns in the specified table and displays their data type and whether they allow null values.
Indexes	lists the name, owner, and type of each index for a specified table, whether the index is clustered, and the names of the columns that are indexed.
Privileges	<p>lists the users who have access privileges for the specified table and whether they have the table-level privileges Select, Update, Insert, Delete, Index, and Alter.</p> <p>Unless your login is listed separately, you have the privileges given for public (a general category for all users).</p> <p>If you want information about database-level privileges, you must use a SELECT statement within the SQL Menu to access the sysusers system catalog table.</p>
References	lists the users who have the table-level References privilege for the specified table and which columns they can reference.
Status	lists the table name, the table owner, the size of the row (in number of bytes), the number of rows in the table (as of the last UPDATE STATISTICS statement), the number of columns in a row, and the date the table was created. (On IBM Informix SE, it also lists the name of the audit trail file, if there is one.)
Table	redispays the INFO FOR TABLE Menu so you can select a different table for examination.
Exit	leaves the INFO Menu and returns you to the TABLE Menu.

You cannot find out the initial and next extent sizes or the lock mode for a table through a menu or through the Info option. You can, however, issue a SELECT statement within the SQL editor to access the **sytables** system catalog table.

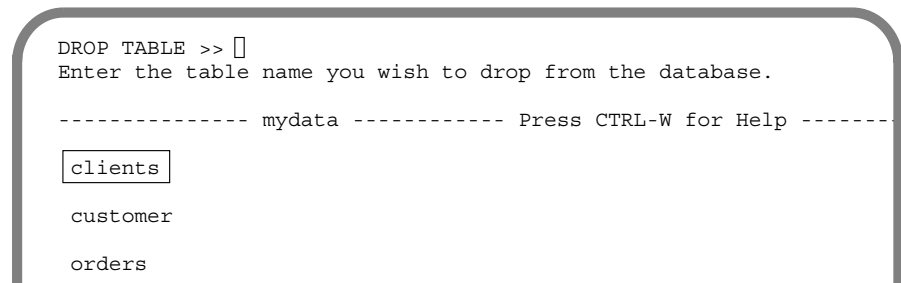
If you are using IBM Informix STAR, you can use the Info option to request information about external tables on the same or another OnLine server by entering the expanded table name at the prompt. The following example requests information on the **bunyip.customer** table in the **stores5** database on the IBM Informix OnLine **topend** server:

```
INFO FOR TABLE >> stores5@topend:bunyip.customer
```

Dropping a Table

Use the Drop option on the TABLE Menu to drop an existing table schema from the database. Type **d** or **D** or highlight Drop and press RETURN. DB-Access displays the DROP TABLE screen, as shown in [Figure 5-29](#).

Figure 5-29
The DROP TABLE screen



This screen lists the names of tables that exist in the current database. You can drop a table in one of the following two ways:

- Type its name and press RETURN. You must use this method and include the full pathname if you want to drop a table that is not in the current database.
- Use the arrow keys to highlight the name of the table you want to drop from the database and then press RETURN.

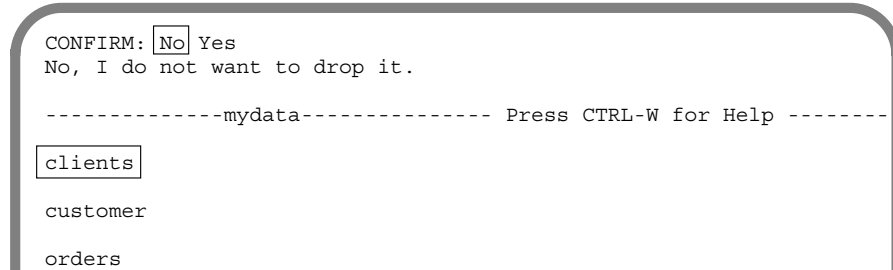
For example, to drop the **orders** table, type in **orders** or highlight it with an arrow key, and then press RETURN.

You also can use the DROP TABLE statement within SQL to drop a table. For details, see the *IBM Informix Guide to SQL: Reference*.

Confirming Your Decision

DB-Access displays a special screen that asks for confirmation before it drops the table, as shown in [Figure 5-30](#).

Figure 5-30
The DROP TABLE confirmation screen



The default is No, so you cannot drop a table by mistake. If you truly want to drop the highlighted table, press Y or use the right arrow key to highlight Yes and press RETURN. DB-Access drops the table.

Tip: When you drop a table, all data in the table is gone for good.



Exiting the DROP TABLE Screen

To leave the DROP TABLE screen without dropping a table, press the Interrupt key. You return to the TABLE Menu.

Sample Command Files

This appendix shows the contents of the various command files that are available with DB-Access. These command files all have the extension **.sql** when displayed from the command line but appear without the extension on the SQL CHOOSE Menu.

Keywords in these command files are shown in uppercase letters to make the SQL statements easier to read. Keywords in the actual command files are lowercase.

Although these command files are listed in alphabetical order, you cannot execute the command files that create tables in that order without creating errors. The order in which the tables are created is very important because of the referential constraints linking those tables. Also, all the tables in the **stores5** database are created automatically if you run the **dbaccessdemo5** script. Therefore, if you try to create a table that already exists in that database, you see an error.

For information on the contents and structure of the **stores5** database, see “The **stores5** Database Map” and the rest of the *IBM Informix Guide to SQL: Reference*.

When you select the Choose option on the SQL Menu, the CHOOSE screen appears. It displays a list of the command files to which you have access.

```
CHOOSE >> 
Choose a command file with the Arrow Keys, or enter a name,
then press Return.

----- stores5 ----- Press CTRL-W for Help -----
c_calls      c_stock     opt_disk    sel_order
c_cat        c_stores    sel_agg     sel_sub
c_custom     c_table     sel-all    sel_union
c_index      c_type     sel_group   upd_table
c_items      c_view1     sel_join
c_manuf      c_view2     sel_ojoin1
c_orders     d_proc      sel_ojoin2
c_proc       d_view      sel_ojoin3
c_state      ins_table   sel_ojoin4
```

Use these command files with DB-Access for practice with SQL and the **stores5** database. Rerun the **dbaccessdemo5** script whenever you want to refresh the **stores5** database and its tables and command files.

For more information on the **stores5** database, see the *IBM Informix Guide to SQL: Reference*.

c_calls.sql

This command file creates the **cust_calls** table.

```
CREATE TABLE cust_calls
(
  customer_num  INTEGER,
  call_dtime    DATETIME YEAR TO MINUTE,
  user_id       CHAR(18) DEFAULT USER,
  call_code     CHAR(1),
  call_descr    CHAR(240),
  res_dtime     DATETIME YEAR TO MINUTE,
  res_descr     CHAR(240),
  PRIMARY KEY (customer_num, call_dtime),
  FOREIGN KEY (customer_num) REFERENCES customer(customer_num),
  FOREIGN KEY (call_code) REFERENCES call_type(call_code)
);
```

c_cat.sql

This command file creates the **catalog** table that is available only with the IBM Informix OnLine database server.

```
CREATE TABLE catalog
(
  catalog_num   SERIAL(10001),
  stock_num     SMALLINT NOT NULL,
  manu_code     CHAR(3) NOT NULL,
  cat_descr     TEXT,
  cat_picture   BYTE,
  cat_advert    VARCHAR(255, 65),
  PRIMARY KEY (catalog_num),
  FOREIGN KEY (stock_num, manu_code) REFERENCES stock
);
```

c_custom.sql

This command file creates the **customer** table.

```
CREATE TABLE customer
(
  customer_num  SERIAL(101),
  fname         CHAR(15),
  lname         CHAR(15),
  company       CHAR(20),
  address1      CHAR(20),
  address2      CHAR(20),
```

```
city          CHAR(15),
state         CHAR(2),
zipcode       CHAR(5),
phone         CHAR(18),
PRIMARY KEY (customer_num)
);
```

c_index.sql

This command file creates an index on the **zipcode** column of the **customer** table.

```
CREATE INDEX zip_ix ON customer (zipcode);
```

c_items.sql

This command file creates the **items** table.

```
CREATE TABLE items
(
  item_num      SMALLINT,
  order_num     INTEGER,
  stock_num     SMALLINT NOT NULL,
  manu_code     CHAR(3) NOT NULL,
  quantity      SMALLINT CHECK (quantity >= 1),
  total_price   MONEY(8),
  PRIMARY KEY (item_num, order_num),
  FOREIGN KEY (order_num) REFERENCES orders (order_num),
  FOREIGN KEY (stock_num, manu_code) REFERENCES stock
    (stock_num, manu_code)
);
```

c_manuf.sql

This command file creates the **manufact** table.

```
CREATE TABLE manufact
(
  manu_code     CHAR(3),
  manu_name     CHAR(15),
  lead_time     INTERVAL DAY(3) TO DAY,
  PRIMARY KEY (manu_code)
);
```


c_orders.sql

This command file creates the **orders** table.

```

CREATE TABLE orders
(
  order_num      SERIAL(1001),
  order_date     DATE,
  customer_num   INTEGER NOT NULL,
  ship_instruct  CHAR(40),
  backlog        CHAR(1),
  po_num         CHAR(10),
  ship_date      DATE,
  ship_weight    DECIMAL(8,2),
  ship_charge    MONEY(6),
  paid_date      DATE,
  PRIMARY KEY (order_num),
  FOREIGN KEY (customer_num) REFERENCES customer (customer_num)
);

```

c_proc.sql

This command file creates a stored procedure. It reads the full name and address of a customer and takes a last name as its only argument.

```

CREATE PROCEDURE read_address (lastname CHAR(15))
RETURNING CHAR(15), CHAR(15), CHAR(20), CHAR(15), CHAR(2), CHAR(5);
DEFINE p_fname, p_city CHAR(15);
DEFINE p_add CHAR(20);
DEFINE p_state CHAR(2);
DEFINE p_zip CHAR(5);
SELECT fname, address1, city, state, zipcode
  INTO p_fname, p_add, p_city, p_state, p_zip
  FROM customer
  WHERE lname = lastname;

RETURN p_fname, lastname, p_add, p_city, p_state, p_zip;

END PROCEDURE;

```

c_state.sql

This command file creates the **state** table.

```
CREATE TABLE state
(
  code          CHAR(2),
  sname        CHAR(15),
  PRIMARY KEY (code)
);
```

c_stock.sql

This command file creates the **stock** table.

```
CREATE TABLE stock
(
  stock_num     SMALLINT,
  manu_code    CHAR(3),
  description   CHAR(15),
  unit_price    MONEY(6),
  unit         CHAR(4),
  unit_descr   CHAR(15),
  PRIMARY KEY (stock_num, manu_code),
  FOREIGN KEY (manu_code) REFERENCES manufact
);
```

c_stores.sql

This command file creates the **stores5** database.

```
CREATE DATABASE stores5;
```

c_table.sql

This command file creates a database named **restock** and then creates a custom table named **sports** (available only with the IBM Informix OnLine database server) in that database.

```
CREATE DATABASE restock;

CREATE TABLE sports
(
  catalog_no      SERIAL UNIQUE,
  stock_no        SMALLINT,
  mfg_code        CHAR(5),
  mfg_name        CHAR(20),
  phone           CHAR(18),
  descript        VARCHAR(255)
);
```

c_type.sql

This command file creates the **call_type** table.

```
CREATE TABLE call_type
(
  call_code       CHAR(1),
  code_descr      CHAR(30),
  PRIMARY KEY (call_code)
);
```

c_view1.sql

This command file creates a view called **custview** on a single table and grants privileges on the view to **public**. It includes the **WITH CHECK OPTION** keywords to verify that any changes to underlying tables made through the view do not violate the definition of the view.

```
CREATE VIEW custview (firstname, lastname, company, city) AS
  SELECT fname, lname, company, city
  FROM customer
  WHERE city = "Redwood City"
  WITH CHECK OPTION;

GRANT DELETE, INSERT, SELECT, UPDATE
ON custview
TO public;
```

c_view2.sql

This command file creates a view on two tables: **orders** and **items**.

```
CREATE VIEW someorders (custnum,ocustnum,newprice) AS
  SELECT orders.order_num,items.order_num,
         items.total_price*1.5
  FROM orders, items
  WHERE orders.order_num = items.order_num
  AND items.total_price > 100.00;
```

d_proc.sql

This command file drops the stored procedure created in **c_proc.sql**.

```
DROP PROCEDURE read_address;
```

d_view.sql

This command file drops the view named **custview** that was created in **c_view1.sql**.

```
DROP VIEW custview;
```

ins_table.sql

This command file inserts one row into the **sports** table (available only with the IBM Informix OnLine database server).

```
INSERT INTO sports
  VALUES (0,18,"PARKR", "Parker Products", "503-555-1212",
  "Heavy-weight cotton canvas gi, designed for aikido or judo
  but suitable for karate. Quilted top with side ties,
  drawstring waist on pants. White with white belt. Pre-washed
  for minimum shrinkage. Sizes 3-6.");
```

sel_agg.sql

This sample SELECT statement queries on table data using aggregate functions. It combines the aggregate functions MAX and MIN in a single statement.

```
SELECT MAX (ship_charge), MIN (ship_charge)
FROM orders;
```

sel_all.sql

This sample query contains all seven SELECT statement clauses that you can use in the Informix implementation of interactive SQL. This SELECT statement joins the **orders** and **items** tables. It also uses display labels, table aliases, and integers as column indicators; groups and orders the data; and puts the results in a temporary table.

```
SELECT o.order_num, SUM (i.total_price) price,
       paid_date - order_date span
FROM orders o, items i
WHERE o.order_date > "01/01/90"
      AND o.customer_num > 110
      AND o.order_num = i.order_num
GROUP BY 1, 3
HAVING COUNT (*) < 5
ORDER BY 3
INTO TEMP temptab1;
```

sel_group.sql

This sample SELECT statement includes the GROUP BY and HAVING clauses. The HAVING clause usually complements a GROUP BY clause by applying one or more qualifying conditions to groups after they are formed, similar to the way the WHERE clause qualifies individual rows. (One advantage to using a HAVING clause is that you can include aggregates in the search condition, whereas you cannot include aggregates in the search condition of a WHERE clause.)

Each HAVING condition compares one column or aggregate expression of the group with another aggregate expression of the group or with a constant. You can use HAVING to place conditions on both column values and aggregate values in the group list.

```
SELECT order_num, COUNT(*) number, AVG (total_price) average
FROM items
GROUP BY order_num
HAVING COUNT(*) > 2;
```

sel_join.sql

This sample SELECT statement queries on table data with a simple join on two tables: **customer** and **cust_calls**. This query returns only those rows in which the customer has made a call to customer service.

```
SELECT c.customer_num, c.lname, c.company,
       c.phone, u.call_dtime, u.call_descr
FROM customer c, cust_calls u
WHERE c.customer_num = u.customer_num;
```

sel_ojoin1.sql

This sample SELECT statement queries on table data using a simple *outer join* on two tables. The use of the keyword OUTER in front of the **cust_calls** table makes it the subservient table. An outer join causes the query to return information on *all* customers, whether or not they have made calls to customer service. All rows from the dominant **customer** table are retrieved, and null values are assigned to corresponding rows from the subservient **cust_calls** table.

```
SELECT c.customer_num, c.lname, c.company,
       c.phone, u.call_dtime, u.call_descr
FROM customer c, outer cust_calls u
WHERE c.customer_num = u.customer_num;
```

sel_ojoin2.sql

This sample SELECT statement queries on table data, creating an outer join that is the result of a simple join to a third table. This second type of outer join is known as a *nested simple join*.

This query first performs a simple join on the **orders** and **items** tables, retrieving information on all orders for items with a **manu_code** of KAR or SHM. It then performs an outer join to combine this information with data from the dominant **customer** table. (An optional ORDER BY clause reorganizes the data.)

```
SELECT c.customer_num, c.lname, o.order_num,
       i.stock_num, i.manu_code, i.quantity
FROM customer c, outer (orders o, items i)
WHERE c.customer_num = o.customer_num
      AND o.order_num = i.order_num
      AND manu_code IN ("KAR", "SHM")
ORDER BY lname;
```

sel_ojoin3.sql

This sample SELECT statement is the third type of outer join, known as a *nested outer join*. It queries on table data by creating an outer join that is the result of an outer join to a third table.

This query first performs an outer join on the **orders** and **items** tables, retrieving information on all orders for items with a **manu_code** of KAR or SHM. It then performs an outer join, which combines this information with data from the dominant **customer** table. This query preserves order numbers that the previous example eliminated, returning rows for orders that do not contain items with either manufacturer code. (An optional ORDER BY clause reorganizes the data.)

```
SELECT c.customer_num, lname, o.order_num,
       stock_num, manu_code, quantity
FROM customer c, OUTER (orders o, OUTER items i)
WHERE c.customer_num = o.customer_num
      AND o.order_num = i.order_num
      AND manu_code IN ("KAR", "SHM")
ORDER BY lname;
```

sel_ojoin4.sql

This sample SELECT statement queries on table data using the fourth type of outer join. This query shows an outer join that is the result of an outer join of each of two tables to a third table. In this type of outer join, join relationships are possible *only* between the dominant table and the subservient tables.

This query individually joins the subservient tables **orders** and **cust_calls** to the dominant **customer** table but does not join the two subservient tables. (An INTO TEMP clause selects the results into a temporary table.)

```
SELECT c.customer_num, lname, o.order_num,
       order_date, call_dtime
FROM customer c, OUTER orders o, OUTER cust_calls x
WHERE c.customer_num = o.customer_num
      AND c.customer_num = x.customer_num
INTO temp service;
```

sel_order.sql

This sample SELECT statement queries on table data using the ORDER BY and WHERE clauses. In this SELECT statement, the comparison "bicycle%" or "bicycle*" specifies the letters bicycle followed by any sequence of zero or more characters. It narrows the search further by adding another comparison condition that excludes a **manu_code** of PRC.

```
SELECT * FROM stock
WHERE description LIKE "%bicycle%"
      AND manu_code NOT LIKE "PRC"
ORDER BY description, manu_code;
```

sel_sub.sql

This sample SELECT statement queries on table data using a subquery. This self-join uses a correlated subquery to retrieve and list the 10 highest-priced items ordered.

```
SELECT order_num, total_price
FROM items a
WHERE 10 >
      (SELECT COUNT (*)
       FROM items b
        WHERE b.total_price < a.total_price)
ORDER BY total_price;
```


sel_union.sql

This example of a SELECT statement uses the UNION clause to query on data in two tables. The compound query performs a union on the **stock_num** and **manu_code** columns in the **stock** and **items** tables. The statement selects those items that have a unit price of less than \$25.00 or that have been ordered in quantities greater than three and lists their **stock_num** and **manu_code**.

```
SELECT DISTINCT stock_num, manu_code
  FROM stock
  WHERE unit_price < 25.00

UNION

SELECT stock_num, manu_code
  FROM items
  WHERE quantity > 3;
```

upd_table.sql

Updates the **sports** table (available only with the IBM Informix OnLine database server).

```
UPDATE sports
  SET phone = "808-555-1212"
  WHERE mfg_code = "PARKR";
```


How to Read Online Help

This appendix shows the conventions used in screen displays of online help available with DB-Access. You receive online help when you press CTRL-W while you are working in the product. Online help is discussed in the section [“Using the HELP Screen” on page 1-25](#).

The form of the syntax diagrams displayed when you request online help for creating, modifying, or editing an SQL statement is different than that shown in the introduction to this manual. The notational conventions and rules governing SQL statement syntax shown in online help are described on the following page. For more detailed syntax diagrams as well as information on the naming of identifiers, databases, and command files, see the *IBM Informix Guide to SQL: Reference*.

ABC Any term in an SQL statement displayed in uppercase letters is a keyword. Type keywords exactly as shown, disregarding case. For example,

```
CREATE SYNONYM synonym-name
```

means you must enter the keywords CREATE SYNONYM or create synonym without adding or deleting spaces or letters.

abc Substitute a value for any term that appears in lowercase letters. In the previous example, you should substitute a value for *synonym-name*.

() Enter any parentheses as shown. They are part of the syntax of an SQL statement, not special symbols.

[] Do not enter brackets as part of a statement. They surround any part of a statement that is optional. For example,

```
CREATE [TEMP] TABLE
```

indicates that you should enter either `CREATE TABLE` or `CREATE TEMP TABLE`.

| The vertical bar indicates a choice among several options. For example,

```
[VANILLA | CHOCOLATE [MINT] | STRAWBERRY]
```

means that you can enter either `VANILLA` or `CHOCOLATE` or `STRAWBERRY` and that, if you enter `CHOCOLATE`, you also can enter `MINT`.

{ } When you must choose one of several options, the options are enclosed in braces and are separated by vertical bars. For example,

```
{GUAVA | MANGO | PASSIONFRUIT}
```

means that you must enter either `GUAVA` or `MANGO` or `PASSIONFRUIT` and that you cannot enter more than one choice.

... An ellipsis indicates that you can enter an indefinite number of additional items like the one immediately preceding the ellipsis. For example,

```
old-column-name
```

```
...
```

means that you can enter a series of existing column names after the first one.

Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. (enter the year or years). All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

AIX; DB2; DB2 Universal Database; Distributed Relational Database Architecture; NUMA-Q; OS/2, OS/390, and OS/400; IBM Informix[®]; C-ISAM[®]; Foundation.2000[™]; IBM Informix[®] 4GL; IBM Informix[®] DataBlade[®] Module; Client SDK[™]; Cloudscape[™]; Cloudsync[™]; IBM Informix[®] Connect; IBM Informix[®] Driver for JDBC; Dynamic Connect[™]; IBM Informix[®] Dynamic Scalable Architecture[™] (DSA); IBM Informix[®] Dynamic Server[™]; IBM Informix[®] Enterprise Gateway Manager (Enterprise Gateway Manager); IBM Informix[®] Extended Parallel Server[™]; i.Financial Services[™]; J/Foundation[™]; MaxConnect[™]; Object Translator[™]; Red Brick Decision Server[™]; IBM Informix[®] SE; IBM Informix[®] SQL; InformiXML[™]; RedBack[®]; SystemBuilder[™]; U2[™]; UniData[®]; UniVerse[®]; wintegrate[®] are trademarks or registered trademarks of International Business Machines Corporation.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Windows, Windows NT, and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names used in this publication may be trademarks or service marks of others.

Index

A

- Add option
 - ALTER TABLE Menu 5-7, 5-32
 - CREATE TABLE Menu 2-17
- ADD STARTING NUMBER screen, schema editor 2-18
- ADD TYPE Menu, schema editor 2-18
- Alter option, TABLE Menu 2-25, 5-4
- ALTER TABLE Menu
 - adding a column 5-34
 - adding and dropping constraints 2-37
 - changing table options 5-38
 - Drop option 2-27
 - dropping a column 5-37
 - modifying a column 5-35
- ALTER TABLE statement
 - ADD keyword 2-35
 - DROP keyword 2-36
- Altering a table. *See* Modifying a table.
- ANSI
 - checking SQL statements for compliance 1-18, 2-10
 - compliance when creating a database 4-9
- ANSI compliance level Intro-18
- ARROW keys 1-8

B

- BACKSPACE key 1-8
- Boldface type Intro-7
- Building the table schema 2-16

C

- CHANGE ANYWAY Menu, schema editor 2-26
- Choose option
 - available command files 3-17
 - on the SQL Menu 3-16
- Column
 - adding with the ALTER TABLE Menu 2-35, 5-34
 - adding with the schema editor 2-25
 - adding with the TABLE Menu 2-25
 - allowing null values with the schema editor 2-20
 - creating with constraints 2-33
 - defining 5-5
 - defining with the schema editor 2-17
 - displaying data type and nulls with the Info option 3-22
 - dropping from a table with the schema editor 2-27
 - dropping with SQL statements 2-36
 - dropping with the ALTER TABLE Menu 2-27, 5-37
 - dropping with the schema editor 5-25

- modifying the data type with the schema editor 5-21
 - modifying the index with the schema editor 5-23
 - modifying the length with the schema editor 5-22
 - modifying the name with the schema editor 5-19
 - modifying the nulls entry with the schema editor 5-24
 - modifying with the schema editor 2-26
 - naming conventions 2-17
 - renaming with SQL statements 2-36
 - Command files
 - choosing from the CHOOSE screen 3-16
 - echoing commands to the screen 1-17
 - executing from the command line 1-16
 - how to drop 3-23
 - how to save 3-19
 - in the stores database 3-17
 - listed 3-17
 - modifying with an editor 1-26
 - rules for naming 3-20
 - shown A-1
 - Command line
 - syntax for invoking DB-Access 1-9
 - using the -ansi flag 2-10
 - Command-line conventions
 - elements of Intro-8
 - Command-line options
 - checking SQL statements for ANSI compliance 1-18
 - displaying the Main Menu 1-11
 - displaying the version number 1-10
 - echoing commands and output to the screen 1-17
 - executing a command file 1-16
 - going to a submenu 1-13
 - going to a submenu option 1-14
 - selecting a database 1-12
 - Confirmation screen
 - for dropping a column 5-37
 - for dropping a command file 3-25
 - Constraint
 - adding with ALTER TABLE 2-37
 - creating with CREATE TABLE 2-33
 - dropping with ALTER TABLE 2-37
 - Contact information Intro-20
 - CONTROL key 1-8
 - Conventions
 - command-line Intro-8
 - online help B-1
 - typographical Intro-7
 - CREATE DATABASE screen
 - differences between OnLine and SE 4-8
 - how to exit 4-9
 - selecting from the DATABASE Menu 4-7
 - shown 2-13
 - CREATE DATABASE statement 2-32
 - CREATE INDEX statement 2-34
 - Create option, TABLE Menu 5-4
 - CREATE TABLE Menu
 - correcting mistakes 2-21
 - how to exit 2-23
 - leaving the menu 5-17, 5-26
 - CREATE TABLE screen
 - how to exit 5-8
 - selecting from the TABLE Menu 5-6
 - CREATE TABLE statement 2-33
 - Creating a database
 - ANSI compliance 4-9
 - differences between OnLine and SE 4-8
 - from a menu 4-7
 - restrictions 4-9
 - restrictions on naming 4-7
 - two ways to do it 2-5
 - using an SQL statement 2-32
 - using the menu options 2-12
 - where system files are stored 2-14
 - with IBM Informix NET 4-8
 - with IBM Informix STAR 4-8
 - Creating a table
 - adding a column 5-9
 - adding to the table schema 5-17
 - building the table schema 5-7
 - restrictions on naming 5-6
 - two ways to do it 2-6
 - using an SQL statement 2-33
 - using the menu options 2-15
 - with the schema editor 5-5
 - CTRL-A key 1-27
 - CTRL-D key 1-27
 - CTRL-R key 1-27
 - CTRL-W key 1-25
 - CTRL-X key 1-27
 - Current statement, definition of 2-10
-
- ## D
- Data
 - manipulating with DB-Access 1-4
 - viewing the next page on the screen 3-9
 - Data types, defining for a column with the schema editor 2-17
 - Database
 - confirming your decision to drop 4-11
 - creating tables in 2-15, 2-24
 - creating through the Main Menu 2-12
 - creating with an SQL statement 2-32
 - creating with IBM Informix NET 4-8
 - creating with IBM Informix STAR 4-8
 - creating with the menu options 2-12
 - dropping a table with DROP TABLE 2-38
 - dropping with a menu option 2-30
 - how to create 4-7
 - how to drop 4-9
 - how to drop with the DROP DATABASE statement 2-39
 - how to select 4-5
 - naming conventions 2-13
 - relational 1-4
 - selecting from a menu 4-5

- selecting from the command line 1-12
- selecting with IBM Informix NET 4-6
- selecting with IBM Informix STAR 4-6
- stores5 Intro-14, 1-7
- turning on logging 4-9
- two ways to create 2-5
- two ways to drop 2-5
- two ways to select 2-4

DATABASE Menu

- available options 4-4
- choosing the Create option 4-7
- choosing the Drop option 4-9
- choosing the Select option 4-5
- selecting from the command line 1-13
- selecting from the Main Menu 4-4
- selecting options from the command line 1-14
- selecting the CREATE DATABASE screen 4-7
- selecting the DROP DATABASE screen 4-9
- selecting the SELECT DATABASE screen 4-5
- shown 4-4

dbspace, defined 2-14

DELETE key 1-8

Demonstration database

- copying Intro-16
- installation script Intro-14
- overview Intro-14
- SQL command files 3-17, A-1

Differences

- between DB-Access and IBM Informix SQL 1-4
- between OnLine and SE on the ALTER TABLE Menu 5-33
- between OnLine and SE on the CREATE DATABASE screen 4-8
- between OnLine and SE on the CREATE TABLE Menu 5-8
- between OnLine and SE on the SELECT DATABASE screen 4-6

- between using DB-Access on OnLine and SE 1-5

Documentation notes Intro-11

Documentation, types of

- documentation notes Intro-11
- machine notes Intro-11
- release notes Intro-11

DROP DATABASE screen

- how to exit 4-11
- selecting from the DATABASE Menu 4-9
- two ways to drop a database 4-10

DROP DATABASE statement 2-39

Drop option

- ALTER TABLE Menu 2-27, 5-7, 5-32
- confirmation screen for dropping command files 3-25
- SQL Menu 3-24

DROP TABLE statement 2-38

Dropping a column

- confirming your decision 5-37
- with the ALTER TABLE Menu 5-37

Dropping a command file,

- confirming your decision 3-25

Dropping a database

- confirming your decision 4-11
- from a menu 4-10
- two ways to do it 2-5
- using a menu option 2-30

Dropping a table

- confirming your decision 5-43
- from a menu 5-42
- two ways to do it 2-6
- using a menu option 2-28

Dropping an SQL statement

- confirming your decision 3-25
- from a menu 3-23

E

- Editing SQL statements
 - with the SQL editor 1-26
 - with the system editor 1-29
- Editor
 - editing SQL statements with the system editor 3-11

- for new SQL statements 3-6
- modifying SQL statements 3-11
- SQL 1-26
- system 1-29
- text 1-26
- what is the interactive schema editor 2-7

Environment variables Intro-7

DBANSIWARN 2-10

- required for DB-Access 1-6
- setting for default editor 1-29
- where to set 1-6

Errors, when running SQL statements 3-8

ESCAPE key 1-9, 1-27

Exit option

- ALTER TABLE Menu 5-7, 5-32
- CREATE TABLE Menu 2-23, 5-17, 5-26
- LOCK MODE Menu 5-30
- RUN screen 3-10
- SQL Menu 3-5
- TABLE Menu 5-4
- TABLE_OPTIONS Menu 5-28, 5-38

Extent size

- how to get information on 2-38
- setting with the Table_options option 5-28

eXtent_size option,

- TABLE_OPTIONS Menu 5-28

F

- Facilities for DB-Access 2-7
- Finding your way around DB-Access 1-7

H

Help

- displaying from menu screen 1-23
- displaying from text-entry screen 1-24
- how to read syntax diagrams B-1

Help Menu

- displaying the next page of help text 1-25
- how to use 1-25
- returning to the working menu or screen 1-25

How to find your way around DB-Access 1-7

How to use your terminal 1-8

I

IBM Informix NET

- creating a database 4-8
- selecting a database 4-6

IBM Informix OnLine

- creating an ANSI-compliant database with CREATE DATABASE 2-32
- demonstration database Intro-14
- where system files are stored 2-14
- where table information is stored 2-24

IBM Informix SE

- creating an ANSI-compliant database with CREATE DATABASE 2-32
- demonstration database Intro-17
- where system files are stored 2-14
- where table information is stored 2-24

IBM Informix STAR

- creating a database 4-8
- requesting table information on another server 3-23
- selecting a database 4-6

Index

- creating with the CREATE INDEX statement 2-34
- displaying name, owner, type, clustering, and columns with the Info option 3-22

Indexing a column

- specifying unique or duplicate values with the schema editor 2-19
- with the schema editor 2-19

INFO Menu

- available options 3-22
- displaying column information 3-22
- displaying column references information 3-22, 5-41
- displaying table index information 3-22
- displaying table privileges information 3-22
- displaying table status information 3-23
- how to exit 3-23
- listing tables 3-23
- option on the SQL Menu 3-22
- options 5-41
- with SQL 5-40

Info option

- Columns option 3-22
- Exit option 3-23
- Indexes option 3-22
- Privileges option 3-22
- References option 3-22, 5-41
- SQL Menu 3-20, 5-33, 5-39
- Status option 3-23
- TABLE Menu 5-4, 5-33, 5-39
- Table option 3-23

INFO statement, getting table information 2-37

Interrupt key 1-8

Invoking DB-Access

- checking for ANSI compliance 1-18
- command-line options 1-9
- DATABASE Menu options 1-14
- displaying the Main Menu 1-11
- displaying the version number 1-10
- echoing file input to a screen 1-17
- executing a command file 1-16
- going to a submenu 1-13
- going to a submenu option 1-14
- selecting a database 1-12
- SQL Menu options 1-15
- TABLE Menu options 1-15

K

Keys

- ARROW 1-8
- BACKSPACE 1-8
- CONTROL 1-8
- CTRL-A 1-27
- CTRL-D 1-27
- CTRL-R 1-27
- CTRL-W 1-25
- CTRL-X 1-27
- DELETE 1-8
- ESCAPE 1-9, 1-27
- Interrupt 1-8
- RETURN 1-9
- SPACEBAR 1-9
- used with SQL editor 1-27

L

Lock mode, how to get information on 2-38

- Lock_mode option, TABLE_OPTIONS Menu 5-28, 5-38

M

Machine notes Intro-11

Main Menu

- available options 1-19
- Database option 1-19
- displaying from the command line 1-11
- Exit option 1-19
- map of DB-Access menu hierarchy 1-20
- Query-language option 1-19
- selecting the Database option 4-4
- selecting the Query-Language option 3-4
- selecting the Table option 5-4
- structure in DB-Access 1-20
- Table option 1-19

Menu options

- altering a table 2-25
- dropping a database 2-30
- dropping a table 2-28

getting table information 2-28
 listed for DATABASE Menu 4-4
 listed for Main Menu 1-19
 listed for SQL Menu 3-4
 listed for TABLE Menu 5-4
 using to create a database 2-12
 using to create a table 2-15
 using to modify a table 2-24

Menus
 example 1-22
 getting help 1-23
 how to exit 1-22
 how to select an option 1-22
 how to use 1-21
 shown for DATABASE
 Menu ?? to 4-11
 shown for SQL Menu ?? to 3-25
 shown for TABLE
 Menu 5-4 to 5-43
 structure in DB-Access 1-20
 structure of header 1-21
 using help 1-25

Mistakes, correcting with the
 schema editor 2-21

Modify option
 ALTER TABLE Menu 5-7, 5-32
 SQL Menu 3-10

Modifying a column, with the
 ALTER TABLE Menu 5-35

Modifying a table
 adding a column with the schema
 editor 2-25
 changing a column 5-18
 changing a column with the
 schema editor 2-26
 changing the column data
 type 5-21
 changing the column index 5-23
 changing the column length 5-22
 changing the column name 5-19
 changing the nulls entry 5-24
 dropping a column with the
 schema editor 2-27
 three ways to do it 2-6
 using an SQL statement 2-34
 using the Alter option 2-25
 using the various menu
 options 2-24

N

Naming conventions
 columns 2-17
 database 2-13
 for saving a command file 3-20
 restrictions 5-11

New option, SQL Menu 3-6

Next_size option,
 TABLE_OPTIONS Menu 5-28,
 5-38

Null values, allowing with the
 schema editor 2-20

O

On-line help, how to read B-1

Options
 how to enter on a command
 line 1-9
 how to get help for 1-25
 how to select on a menu 1-22
 listed for DATABASE Menu 4-4
 listed for Main Menu 1-19
 listed for SQL Menu 3-4
 listed for TABLE Menu 5-4

Output option
 Append-file option 3-14
 New-file option 3-14
 Printer option 3-13
 SQL Menu 3-12
 To-pipe option 3-15

P

Page option, LOCK MODE
 Menu 5-30

Printing the results of a query 3-13

Q

Query
 appending results to an existing
 file 3-15
 sending results to a file 3-14
 sending results to a pipe 3-15
 sending results to a printer 3-13

storing results in a new file 3-14

Query language
 definition of 2-9
 using with DB-Access 2-9

Query-Language option
 how to use it 3-4
 on the Main Menu 3-4

R

References, displaying for a
 table 3-22, 5-41

Release notes Intro-11

RENAME COLUMN statement,
 definition of 2-36

Reserved words, restrictions
 on 5-11

RETURN key 1-9

Row option, LOCK MODE
 Menu 5-30

Run option
 Exit option 3-10
 Next option for viewing data 3-9
 Restart option to redisplay
 results 3-10
 SQL Menu 3-7

Running SQL statements
 when there are errors 3-8
 when there are no errors 3-8

S

Save option
 rules for naming saved files 3-20
 SQL Menu 3-19

Saving command files 3-19

Schema
 building with the schema
 editor 2-16
 calling with the Table option on
 the Main Menu 2-15
 how to build 2-8

Schema editor
 ADD STARTING NUMBER
 screen 2-18
 ADD TYPE Menu 2-18
 adding a column to a table 2-25
 ALTER TABLE 2-25

- assigning data types 2-18
- CHANGE ANYWAY Menu 2-26
- correcting mistakes 2-21
- creating a table 5-5
- defining a column 2-17, 5-5
- definition of 2-16
- dropping a column from a table 2-27, 5-25
- EXIT Menu 2-23, 5-17, 5-26
- how to exit 2-23
- how to use 2-7
- modifying a column in a table 2-26
- restrictions on creating a table 5-8
- what it lets you do 5-5
- Screen option, ALTER TABLE Menu 5-7, 5-32
- Screens
 - example 1-23
 - for text entry 1-23
 - getting help 1-24
 - how to enter text 1-24
 - how to exit 1-24
 - shown for DATABASE Menu 4-4 to 4-11
 - shown for SQL Menu 3-4 to 3-25
 - shown for TABLE Menu 5-4 to 5-43
 - structure of header 1-23
- SELECT DATABASE screen
 - differences between OnLine and SE 4-6
 - how to exit 4-6
 - selecting from the DATABASE Menu 4-5
 - two ways to select a database 4-5
- Selecting a database
 - differences between OnLine and SE 4-6
 - from a menu 4-5
 - two ways to do it 2-4
 - with IBM Informix NET 4-6
 - with IBM Informix STAR 4-6
- SERIAL data type, assigning starting number 2-18
- Setting environment variables for DB-Access 1-6
- Size
 - setting initial extent 5-28
 - setting next extent 5-28
- SPACEBAR key 1-9
- SQL
 - command files for stores database 3-17
 - entering statements 3-6
 - how to read syntax in online help screens B-1
 - how to use from a menu 3-3
- SQL editor
 - and new SQL statements 3-6
 - CTRL-A editing key 1-27
 - CTRL-D editing key 1-27
 - CTRL-R editing key 1-27
 - CTRL-X editing key 1-27
 - editing keys 1-27
 - ESC key 1-27
 - how to use 1-27
 - modifying SQL statements 3-11
- SQL Menu
 - available options 3-4
 - choosing the Choose option 3-16
 - choosing the Drop option 3-23
 - choosing the Info option 3-20
 - choosing the Modify option 3-10
 - choosing the New option 3-6
 - choosing the Output option 3-12
 - choosing the Run option 3-7
 - choosing the Save option 3-19
 - choosing the Use-editor option 3-11
 - running new SQL statements 3-7
 - selecting from the command line 1-13
 - selecting options from the command line 1-15
 - selecting the CHOOSE screen 3-16
 - selecting the DROP COMMAND FILE screen 3-24
 - selecting the INFO FOR TABLE screen 3-21
 - selecting the NEW screen 3-6
 - selecting the OUTPUT Menu 3-13
 - selecting the SAVE screen 3-19
 - statements you can run 2-10
- SQL statements
 - ALTER TABLE 2-35
 - appending query results to an existing file 3-15
 - choosing a command file 3-16
 - CREATE DATABASE 2-32
 - CREATE INDEX 2-34
 - CREATE TABLE 2-33
 - creating a database 2-32
 - creating a table 2-33
 - creating an index on a table 2-34
 - current defined 2-10
 - DROP DATABASE 2-38
 - DROP TABLE 2-38
 - dropping a column 2-36
 - entering new ones with the SQL editor 3-6
 - how to drop 3-23
 - how to edit with the SQL editor 1-26
 - how to edit with the system editor 1-29, 3-11
 - how to enter 3-6
 - how to modify 3-10
 - how to run 3-8
 - how to save to a command file 3-19
 - how to use 2-9
 - INFO 2-37
 - listed for DB-Access 2-10
 - modifying a table 2-34
 - redirecting query results 3-12
 - RENAME COLUMN 2-36
 - renaming a column 2-36
 - saving to a command file 3-19
 - SELECT 2-38
 - selecting the SAVE screen 3-19
 - sending output to a file 3-14
 - sending output to a printer 3-13
 - sending query results to a pipe 3-15
 - two ways to check ANSI compliance 2-10
 - using with DB-Access 2-9
 - what happens when there are errors 3-8
 - what you can do with them in DB-Access 2-31
- Status, displaying for a table 3-23
- stores5 database
 - copying Intro-16

creating on IBM Informix OnLine Intro-16
 creating on IBM Informix SE Intro-17
 overview Intro-14
 SQL command files 3-17
 System catalog, described 2-15
 System editor, how to use 1-29
 System tables, where they are stored 2-14

T

Table

adding a column with the schema editor 2-25, 5-9
 allowing null values in a column with the schema editor 2-20
 building the schema with the schema editor 2-16
 creating an index with an SQL statement 2-34
 creating through menus 2-8, 2-16, 5-8, 5-38
 creating with an SQL statement 2-33
 creating with the menu options 2-15
 defining a column 5-5
 defining a column with the schema editor 2-17
 defining the column data types with the schema editor 2-17
 displaying column information with the Info option 3-22
 displaying index information with the Info option 3-22
 displaying information for 5-39
 displaying information on the screen 3-20
 displaying Privileges information with the Info option 3-22
 displaying References information with the Info option 3-22, 5-41
 displaying Status information with the Info option 3-23
 dropping a column 5-25

dropping a column with the schema editor 2-27
 dropping from a database with the DROP TABLE statement 2-38
 dropping using a menu option 2-28
 getting information on extent size 2-38
 getting information on lock mode 2-38
 how to build a schema 2-8
 how to call up the schema 2-15
 how to create with the schema editor 5-5
 how to drop 2-38, 5-42
 how to get information on structure and characteristics 2-37
 how to get information using menu options 2-28
 indexing a column with the schema editor 2-19
 modifying a column 5-18
 modifying a column with the schema editor 2-26
 modifying the column data type 5-21
 modifying the column index 5-23
 modifying the column length 5-22
 modifying the column name 5-19
 modifying the nulls entry 5-24
 modifying with an SQL statement 2-34
 naming a column with the schema editor 2-17
 rebuilding the schema after changes 5-26
 requesting information on a table on another server 3-23
 schema displayed 2-23
 setting the extent sizes 5-27
 setting the lock mode 5-27
 three ways to get information 2-6
 three ways to modify 2-6
 two ways to create 2-6
 two ways to drop 2-6
 where information is stored 2-24

TABLE Menu

available options 5-4
 guidelines for using 5-4
 schema editor 2-25
 selecting from the command line 1-13
 selecting options from the command line 1-15
 selecting the CREATE TABLE screen 5-6
 selecting the Drop option 5-42
 selecting the Info option 5-39
 Table_Options option, ALTER TABLE Menu 5-32
 tblspace, defined 2-24
 Terminal keys
 ARROWS 1-8
 BACKSPACE 1-8
 CONTROL 1-8
 DELETE 1-8
 ESCAPE 1-9
 Interrupt 1-8
 RETURN 1-9
 SPACEBAR 1-9
 Text
 editing with the SQL editor 1-26
 editing with the system editor 1-29
 how to enter on a screen 1-24
 Text editor
 how to use 1-26
 internal editor 1-28
 SQL editor 1-28
 Typographical conventions Intro-7

U

Use-editor option, SQL Menu 3-11

V

Variables

setting environment at system prompt or in a file 1-6
 setting for DB-access environment 1-6
 Version number, displaying from the command line 1-10

W

What is DB-Access 1-4

X

X/Open compliance level Intro-18