

Creating Database Objects with Oracle8i

In this chapter, I'm going to show you how to use the two most common utilities in Oracle8i, SQL*Plus and Enterprise Manager, to access your database. Then I'll talk about how to create tables and indexes using Enterprise Manager. Finally, I'll cover how to manage your security using Enterprise Manager.

Introducing SQL*Plus

SQL*Plus comes in two flavors: a command-line driven utility that runs under DOS and a window-oriented utility that provides a pane where you can enter SQL statements and a pane to show the results. Both tools accept the same set of SQL statements and will return the same results. The only difference between the two is how the data is presented to the user. I'll use the command-line variety in this section.

Command-Line SQL*Plus

The command-line version of SQL*Plus is started from a command prompt by entering `SQLPLUS`. You can enter commands directly from the keyboard or enter them into a file and execute the entire file. The results of the commands will be displayed immediately following the command you enter.

Note

DOS and commands: In trying to eradicate the concept of DOS from the user's mind, Microsoft engineers cleverly renamed the old DOS Window to Command Prompt in Windows 2000; however, it's the same old command interpreter you've come to love and hate.



In This Chapter

Introducing Oracle8i SQL*Plus

Introducing Oracle8i Enterprise Manager

Constructing tablespaces, tables, and indexes

Managing security

Creating roles and users



Connecting to your database with SQL*Plus

These parameters include the user name, the password, and the database you plan to use. If you don't supply all of the information, you will be prompted for it. The format you must use is shown below:

```
<username>[/<password>][@<connectstring>]
```

where <username> is a valid user name for the database you wish to access, <password> is the password associated with the user name, and <connectstring> contains the name of the database you wish to access.

For example, if you want to have a user name of `toy` and `vb6db.athena.justpc.net` as the name of the database you wish to access, you can start SQL*Plus using the following command:

```
SQLPLUS toy@vb6db.athena.justpc.net
```

SQL*Plus will prompt you for the password, and assuming that you entered it correctly, you could be prompted with an `SQLPLUS>` prompt (see Figure 27-1).



```
E:\>sqlplus -
Usage: SQLPLUS [ <option> ] [ <login> ] [ <start> ] [
where <option> ::= - | -? | [ -M <n> ] [ -R <n> ] [ -S ]
      <login> ::= <username> [ / <password> ] [ @ <connect_string> ] | / | /NOLOG
      <start> ::= @ <filename> [ <text> ] [ <parameter> ... ]
      "-" displays the usage syntax
      "-?" displays the SQL*Plus version banner
      "-M <n>" uses HTML markup options <n>
      "-R <n>" uses restricted mode <n>
      "-S" uses silent mode

E:\>sqlplus toy@vb6db.athena.justpc.net
SQL*Plus: Release 8.1.6.0.0 - Production on Wed May 17 23:04:44 2000
(c) Copyright 1999 Oracle Corporation. All rights reserved.
Enter password:
Connected to:
Oracle8i Enterprise Edition Release 8.1.6.0.0 - Production
With the Partitioning option
JServer Release 8.1.6.0.0 - Production
SQL>
```

Figure 27-1: Logging onto a database with SQL*Plus.

Note

There are other options also: SQL*Plus supports several other options on the command line. See *Oracle8i DBA Bible* by Jonathan Gennick, and published by IDG Books, for more information.

Caution

It's not very hidden: While you can enter your password on the command line, I don't recommend it. In Windows 2000, someone can simply press the up arrow until they see the command you executed to start SQL*Plus, even inside the SQL*Plus program. After quitting SQL*Plus, this information will remain in the command history of both Windows 2000/NT and Windows 98/95 systems with DOSKEY running. If it was the last command entered, it will be visible by simply hitting the F3 key.

Entering commands

Once SQL*Plus has been started, an `SQL>` prompt will be displayed. You can enter any command or SQL statement at the prompt. If the syntax is correct, the command or SQL statement will be immediately executed and the results displayed below what you typed.

If the display is too wide, it will automatically be wrapped to the next line. Note that the lines displayed in the DOS window will automatically scroll so that the most recent information is displayed. This can cause a problem if you are executing a **Select** statement, which returns a large number of rows. In this case, you should either use SQL*Plus Window or increase the screen buffer size under the DOS window's property settings.

Note

Confusing, isn't it?: Commands and SQL statements are treated differently in SQL*Plus. Commands occupy only a single line, while SQL statements may occupy multiple lines. To indicate the end of an SQL statement, you must type a semicolon (;). Typing a semicolon at the end of a command will generate an error message. Also, if you don't supply sufficient information for a command, you may be prompted for the additional information, or you may receive a message describing how to use the command.

Useful commands

Table 27-1 lists some of the more useful commands you will find in SQL*Plus. While this is not a complete list, it should give you a feel for the types of commands available. You can use the **Help Index** command to find a complete list of the commands that are available.

Table 27-1
Useful Commands in SQL*Plus

<i>Command</i>	<i>Description</i>
Clear SQL	Clears the contents of the SQL buffer.
Connect	Logons to a different database server or the same database server as a different user.
Define _Editor	Specifies the name of your favorite editor program that will be run when you use the Edit command.
Describe	Provides a detailed description of a database object, such as a table or stored procedure.
Edit	Edits the contents of the SQL buffer in your favorite editor.
Get	Loads the contents of a file into the SQL buffer.
Help	Provides basic help on SQL*Plus commands.
Host	Runs the specified DOS command and returns to SQL*Plus.
List	Lists the statements in the SQL buffer.
Quit	Exits SQL*Plus and returns to DOS.
Run	Lists the statements in the SQL buffer and executes them.
Save	Saves the SQL buffer to a disk file.
Set Linesize	Determines the width of the line.
Set Newpage	Determines the number of lines displayed between pages.
Set Pagesize	Determines the number of lines in a page.
Set Pause	When On, means that the output will be displayed in pages.
Set Serveroutput	When On, means that output will be directed to the console when using the DBMS_OUTPUT package.
Spool	Directs output to an external file or to the system printer.
Start	Runs the SQL statements in a disk file.

One of the most useful commands in SQL*Plus is the **Describe** command. This allows you to quickly see the structure of a table or the statements in a stored procedure. I find this helpful if I don't have a piece of paper handy containing columns and tables in a database.

After running an SQL statement, it is stored in the SQL buffer. You can list the contents of this buffer, edit the statement, or run the statement over again using the commands shown in Table 27-1. The contents remain in the SQL buffer until another SQL statement is entered or the buffer is explicitly cleared using the **Clear SQL** command. You can list the contents of the SQL buffer with the **List** command. The **Save** and **Get** commands will write a copy of the buffer to a disk file and load it back again, respectively. Use the **Run** command to execute the statements in the SQL buffer.

The **Edit** command copies the SQL buffer to a file called `afiedt.buf` and loads it into uses whatever editor you specify in the **Define _Editor** command. You then edit your statements, save them and close the editor to return to SQL*Plus. By default on a Windows platform, you'll use `Notepad`. You could also use the **Save** command to save the SQL buffer into a file and then use the **Host** command to perform the same function. Note that your SQL*Plus session is suspended until the editor or host command is complete.

SQL*Plus contains commands that can make working with your SQL statements easier. The first thing you should do if you are testing **Select** statements is to use the **Set Pause On** command. This command will automatically break your output into pages and pause the output at the end of each page. The number of lines in each page is controlled by the **Set Pagesize** command. You can also control the number of lines displayed between page breaks by using the **Set Newpage** command.

If you choose, you can direct the output of the SQL statements to a disk file or a printer by using the **Spool** command. To redirect your output to a file, specify **Spool** followed by the name of the disk file. To redirect your output to the printer, issue the **Spool Out** command. Then you can issue any collection of SQL statements or commands that generate output. When you're finished, issue the **Spool Off** statement to close the output file or send your output to the printer.

Tip

Make it fit: Use the **Set Pagesize**, **Set Newpage**, and **Set Linesize** commands to adjust the characteristics of your printer.

Introducing Enterprise Manager

Enterprise Manager is a comprehensive tool designed to help you manage your database using graphical tools. It provides a central point from which you can manage all of your Oracle8i database servers, even if they are on multiple physical computers. It communicates with the Oracle Management Server, which in turn communicates with the database servers that you wish to access. While not as easy to use as SQL Server's Enterprise Manager, Oracle8i Enterprise Manager is a big improvement over issuing SQL statements in SQL*Plus to perform common tasks such as creating tasks and users.

The Enterprise Manager console

Rather than combine all of the tools into a single application, Enterprise Manager consists of the Enterprise Manager console, plus a number of other utilities that can be launched from the console or in a stand-alone fashion. Some of the more important tools include:

- ♦ **SQL*Plus Worksheet** — a graphical version of SQL*Plus.
- ♦ **DBA Studio** — provides a graphical way to create and maintain your database structures.
- ♦ **Enterprise Security Manager** — simplifies the processes needed to create users and map them onto the proper roles.
- ♦ **Net8 Assistant** — makes it easy to configure your communications network.

Tip

There's no substitute for page count: If you are working with a database like Oracle8i with lots of tools and utilities, you should invest in a good reference. I suggest the *Oracle8i DBA Bible* by Jonathan Gennick published by IDG Books.

Logging onto the Enterprise Manager console

When you initially start the Enterprise Manager console (choose Start ⇨ Oracle – OraHome81 ⇨ Enterprise Manager ⇨ Console), you will see the Enterprise Manager Login window (see Figure 27-2). In this window, you need to supply a valid administrator ID and password, as well as choose the location of the Oracle Management Server.

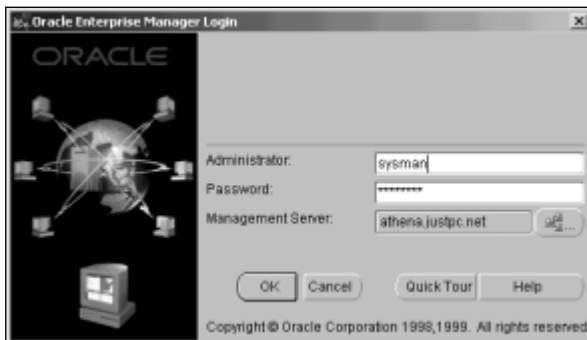


Figure 27-2: Logging onto the Enterprise Manager console.

Note

Yet another ID: The administrative ID used to log in to the Oracle Management Server is not the same thing as the user name you specify when you log in to a database. The administrative ID is used to manage network resources, not database resources.

Using the Enterprise Console

The Enterprise Console presents a packed view of the resources and tools that are available for you to monitor and manage your collection of database servers (see Figure 27-3).

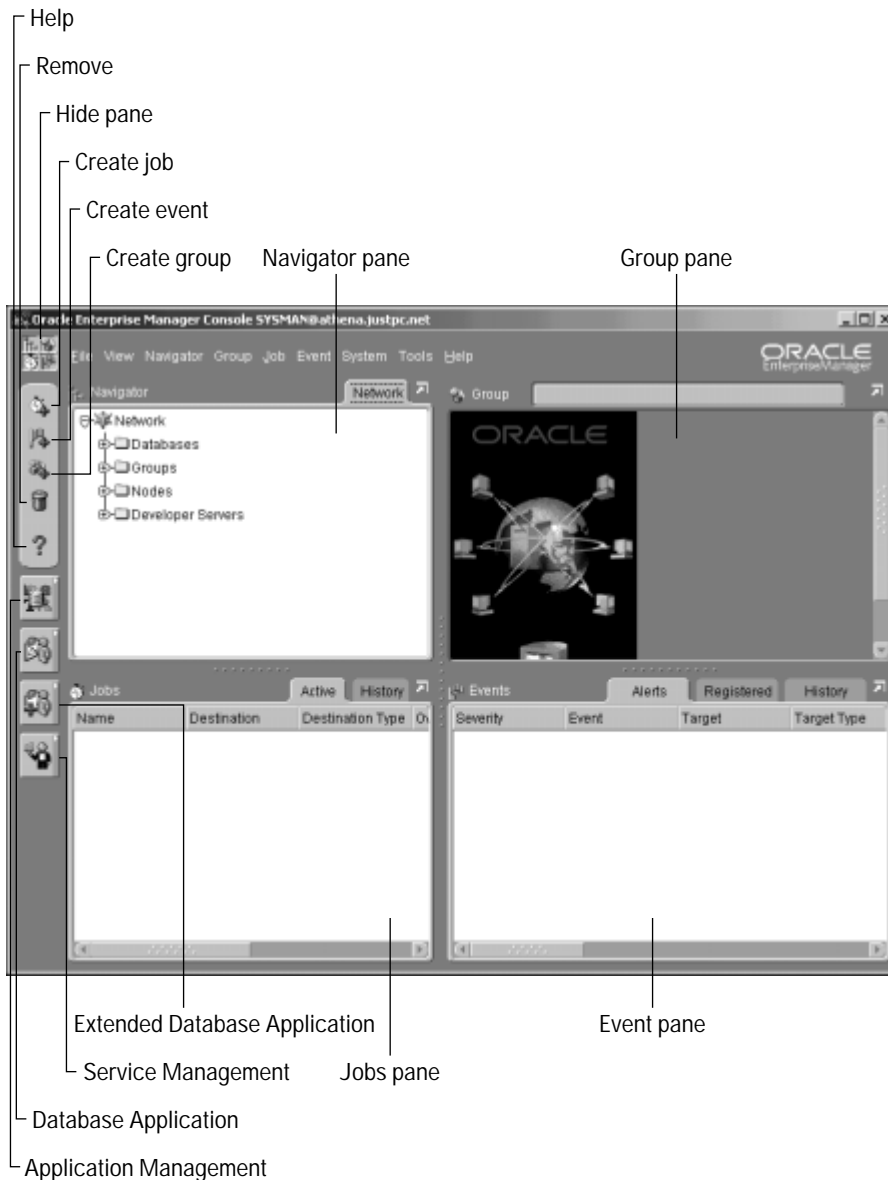


Figure 27-3: Running the Enterprise Console.

- ♦ **Navigator pane** contains information about the objects that can be managed through the Enterprise Manager console.
- ♦ **Group pane** is customized to display the information you wish from the Navigator pane.
- ♦ **Jobs pane** lists the jobs defined in the database that performs tasks like database backups and batch processing.
- ♦ **Event pane** contains information about the events you define in your database.
- ♦ **Hide pane** provides a quick way to hide any of the four main panes (Navigator, Group, Jobs and Events).
- ♦ **Create job** walks you through the process to add a job to the system.
- ♦ **Create event** displays a window that you use to define an event.
- ♦ **Create group** defines a group that can be displayed in the group pane.
- ♦ **Remove** allows you to delete an object in the console.
- ♦ **Help** invokes the help subsystem.
- ♦ **Application Management** allows you to launch the Oracle Applications Manager.
- ♦ **Database Applications** includes options to start the DBA Studio and the SQL*Plus Worksheet utilities.
- ♦ **Extended Database Applications** provides buttons that bring up the Oracle interMedia Text Manager, the Oracle Replication Manager, and the Oracle Spatial Index Advisor.
- ♦ **Service Management** launches the Net8 Assistant, the OSA Manager, or the Oracle Internet Directory Manager.

Connecting to databases

In the Navigator window, you can explicitly connect to a database to view the information about it. Just expand the Databases icon to show the databases that are available. Then right click on the database and choose Connect from the pop-up menu. This will display a connection form, as shown in Figure 27-4.

Fill in the user name and password that you wish to use to access that particular database instance. Then check the Save as Preferred Credential if you want to make the user name and password the default when connecting to this database. This means that whenever you expand the database icon, you won't be prompted for connection information. Since the Enterprise Manager knows which database you want to access, it supplies the information in the Service field and doesn't allow you to change it.

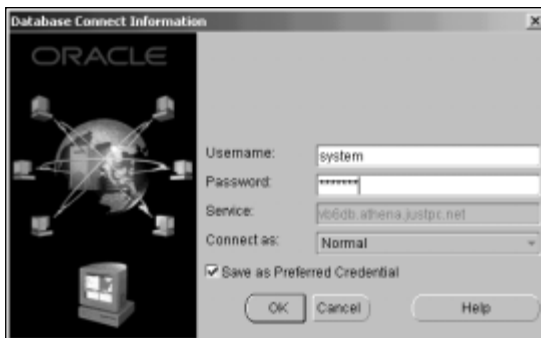


Figure 27-4: Connecting the Enterprise Manager to a database.

Application developers typically need many of the functions and resources available in the Oracle8i Enterprise Manager. However, the SQL*Plus Worksheet utility and the DBA Studio are two tools that are worth discussing in more detail.

SQL*Plus Worksheet

Running functions inside a DOS window isn't for everyone. Oracle8i includes an alternative to SQL*Plus, called SQL*Plus Worksheet. This utility is basically a graphical version of SQL*Plus. The main improvement is that the SQL buffer is displayed in one pane, while the results of its execution are displayed in another pane. You can start this utility directly from the Enterprise Manager or by choosing Start ⇨ Oracle – OraHome81 ⇨ Database Administration ⇨ SQLPlus Worksheet.

Note

SQL*Plus Window ain't SQL*Plus Worksheet: There's a third version of SQL*Plus, called SQL*Plus Window (Start ⇨ Oracle – OraHome81 ⇨ Application Development ⇨ SQL Plus). This utility is merely SQL*Plus for DOS running in a window. You still enter commands at a prompt. Its primary advantage is the scrollbars that allow you to look back at the commands and SQL statements you execute and their results.

Connecting to your database

You start SQL*Plus Worksheet from the Enterprise Manager by right clicking on the database you wish to use and choosing Database Application ⇨ SQL*Plus Worksheet from the pop-up menu. If you were already connected to the database, or have a preferred credential for this database, you will automatically be connected when SQL*Plus Worksheet starts. Otherwise, SQL*Plus Worksheet will be started without an active connection, and you will need to execute a **Connect** statement to connect to your database.

If you start SQL*Plus Worksheet using the Start button, you'll be prompted with the Enterprise Manager Login form. You have a choice of two ways to log in. You can log in through the Oracle Management Server using your administrator ID and password, or you can log in directly to the database using your normal database password (see Figure 27-5). If you wish, you can click on the Login to the Oracle Management Server and log in to the Oracle Management Server first. Then the preferred credentials will be automatically used for this particular database instance.

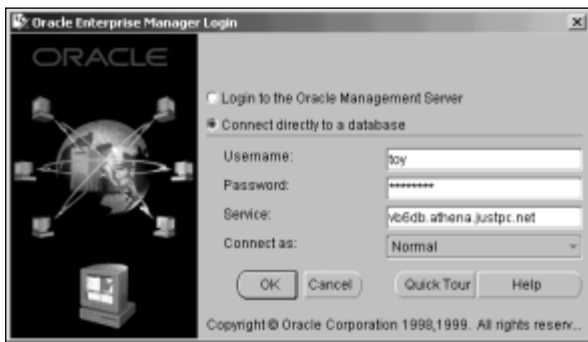


Figure 27-5: Login to SQL*Plus Worksheet using a normal database user name.

Running SQL*Plus Worksheet

After logging onto SQL*Plus Worksheet, you'll see a two-paneled window with a set of icons along the left side, as shown in Figure 27-6. The top pane holds the SQL statements you wish to execute, while the bottom pane contains their results. When you initially log in, you'll see the results of the **Connect** command that was used to access the database.

- ♦ **Command pane** is the pane where you enter you SQL statements and/or commands for execution.
- ♦ **Results pane** is the pane where the results from executing a command are displayed.
- ♦ **Connection** allows you to log out from the database and back on as a different user.
- ♦ **Execute** starts processing the SQL statements listed in the command area.
- ♦ **History** maintains a list of the commands and SQL statements you have executed during this session.
- ♦ **Help** displays the Enterprise Manager Help System

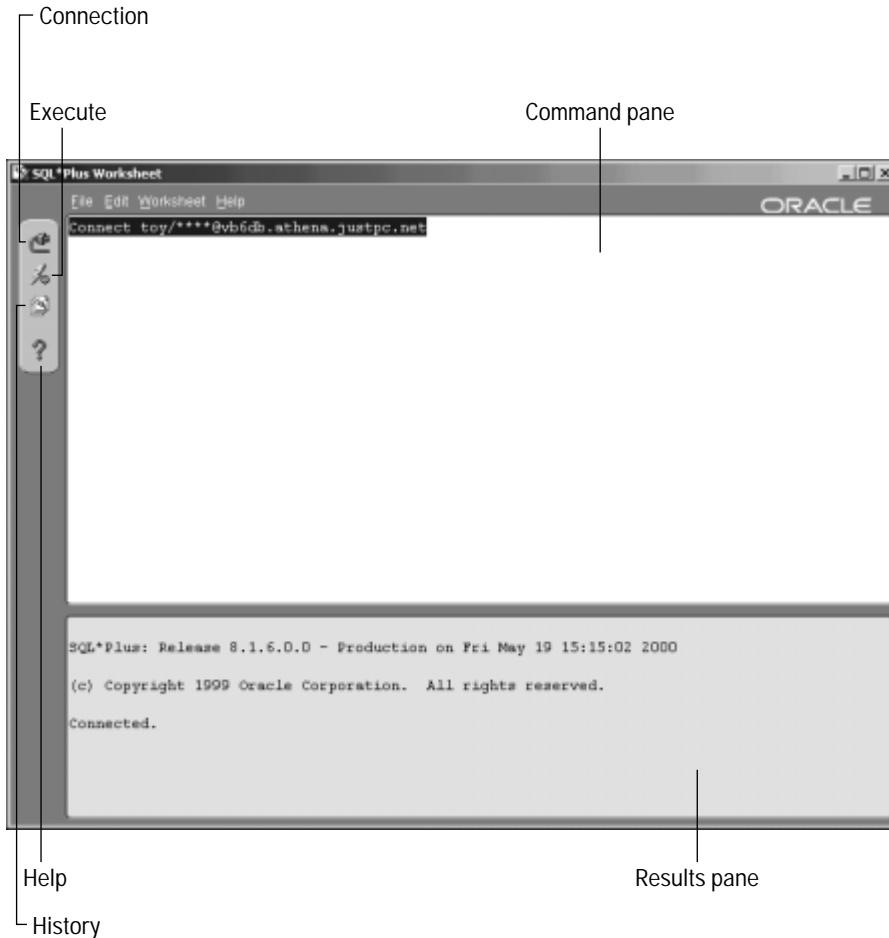


Figure 27-6: Running SQL*Plus Worksheet.

As you might expect, all you need to do to execute a command or SQL statement is to enter it into the Command pane and press the Execute button. One word of caution — once you start executing a command, you can't stop it. While this isn't a problem for most commands, consider the problem that might occur if you try to **Select** several thousand rows from a table.

DBA Studio

The DBA Studio utility is designed to make your life easier when you're designing an Oracle8i database. Many functions that previously required you to specify very complex SQL statements, like **Create Table** or **Create Schema**, are now implemented in a much easier to use GUI.

Tip

A real Oracle DBA uses ERWin: While DBA Studio is a powerful tool, you might want to consider using ERWin by Computer Associates. While this product is somewhat expensive, it can be worth its weight in gold when designing complex Oracle databases, and unlike database designer solutions from the database vendor, this tool allows you to design databases that can be implemented for nearly any database vendor.

Starting DBA Studio

You can start this utility directly from the Enterprise Manager, or by choosing Start ⇨ Oracle – OraHome81 ⇨ Database Administration ⇨ DBA Studio. When you start the DBA Studio directly from Windows, you will be prompted to launch DBA Studio standalone or to log in to the Oracle Management Server. In either case, you will want to connect to the database as SYSDBA (see Figure 27-7). This will ensure that you have all of the appropriate capabilities you'll need while designing your database.

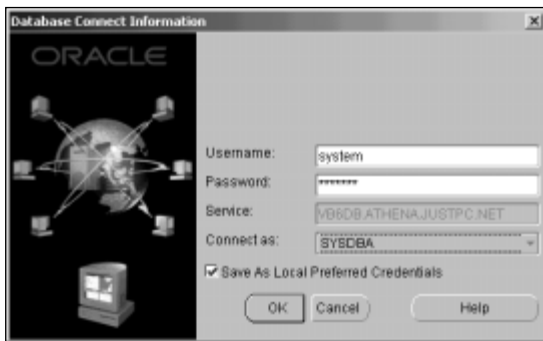


Figure 27-7: Logging onto DBA Studio in standalone mode.

Caution

But it's encrypted: If you choose to save your login information as the preferred credential when logging into DBA Studio in standalone mode, you should know that the password is stored on your local hard disk. Even though your password is encrypted, anyone who has physical access to your machine can access DBA Studio by using your default login information.

Running DBA Studio

DBA Studio presents a two-paned view similar to that used by SQL Server Enterprise Manager (see Figure 27-8). It also includes a column of icons down the left side of the window that you can use to perform commonly used functions.

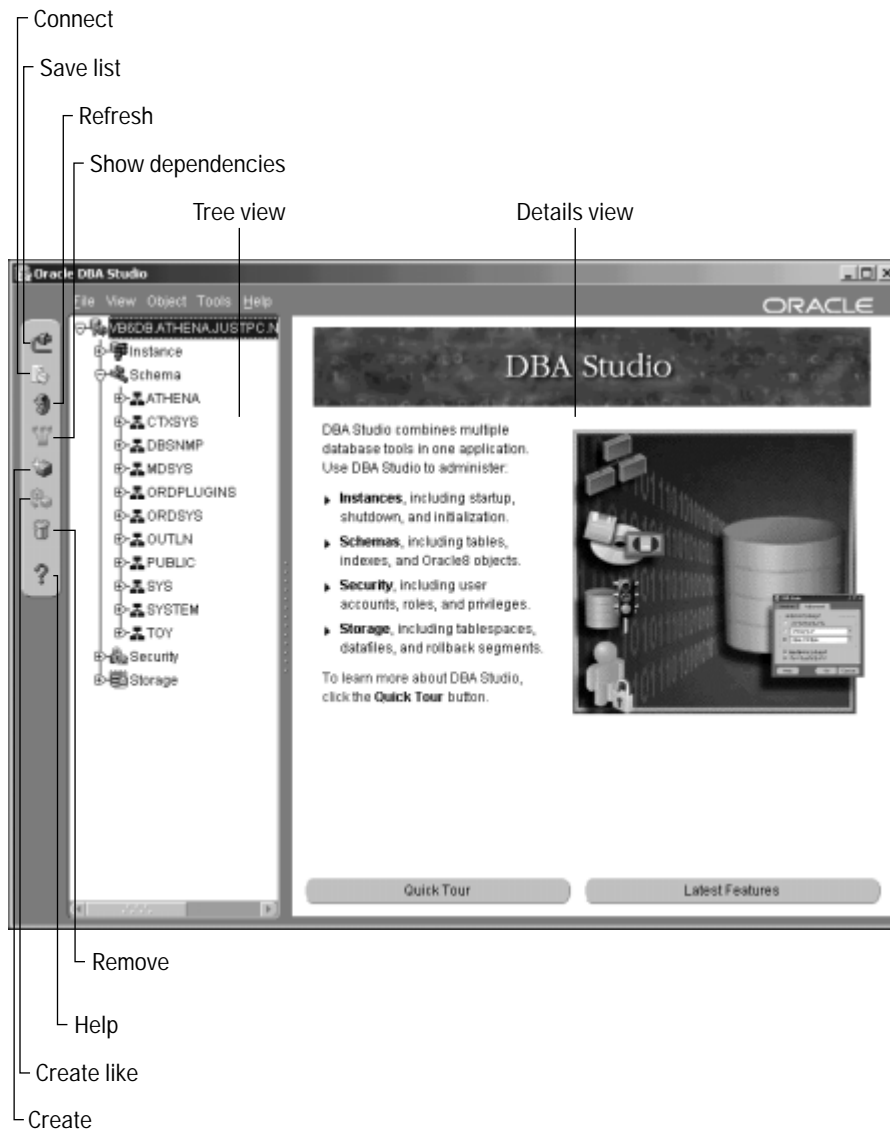


Figure 27-8: Running DBA Studio.

- ♦ **Tree view** contains a hierarchical list of icons that reference the database objects that you can manipulate with this utility.
- ♦ **Details view** provides additional information about the currently selected icon in the tree view.
- ♦ **Connect** allows you to establish a database connection to another database.
- ♦ **Save List** creates a file that contains the list of objects, such as the tables that are contained in a particular schema.
- ♦ **Refresh** gets a fresh copy of the information displayed in the details view and the icon view.
- ♦ **Show Dependencies** displays the objects that the selected object depends on, as well as those objects that depend on the selected object.
- ♦ **Create** creates a brand new object in the currently selected category in the icon view.
- ♦ **Create Like** creates a new object based on the currently selected object.
- ♦ **Remove** deletes the currently selected object.
- ♦ **Help** invokes the help subsystem.

Creating Tablespaces

Since creating databases is a task best suited to a database administrator, I want to begin by walking you through the process to create a tablespace. A *tablespace* represents a pool of disk storage that can be used to hold tables and indexes.

Before you create a tablespace

Before you create a tablespace, you should consider the following questions:

- ♦ Which database instance should hold the new tablespace? Most servers that run Oracle8i run multiple database instances. This allows the database administrator to allocate tablespace and tables to different database instances to optimize performance.
- ♦ What name do you want to give the new tablespace? Tablespace names can be from 1 to 30 characters in length and can't duplicate the name of another tablespace or an Oracle8i reserved word.
- ♦ Which disk location should hold the files for the tablespace? This information will vary depending on the operating system. On Windows-based systems, you'll have to provide the drive, directory path, and file name for each file that will hold the data in the tablespace.

- ♦ What should the initial size of each file be? If you need to grow your tablespace, you can add additional files or use the **Alter Tablespace** command to modify how the existing data files can grow.

Creating your tablespace

Start Oracle DBA Studio and follow these steps:

1. Verify that you are connected to the database where you wish to add a tablespace and click on the Create icon. This will display the Create On window, as shown in Figure 27-9.



Figure 27-9: Select the tablespace icon to create a new tablespace.

2. Choose Tablespace and press the Create button. This will display the Create Tablespace property window, as shown in Figure 27-10. Then you just need to fill in the blanks. Enter a name for the tablespace. This will automatically fill in a value for the file name. If you wish, you can change the name of the file. If you plan to add additional files to this tablespace, you should include a `_01` to indicate that this is the first file. The File Directory will default to the one currently used by the database. You can also change it, if you wish to place the file on another disk drive. Finally, you need to fill in the size of the tablespace. If you click on the area next to the numeric value under Size, you can change the units from megabytes (MB) to kilobytes (KB).
3. Press the Create button to allocate the space for the tablespace. A message box will be displayed, letting you know when the tablespace has successfully been completed.

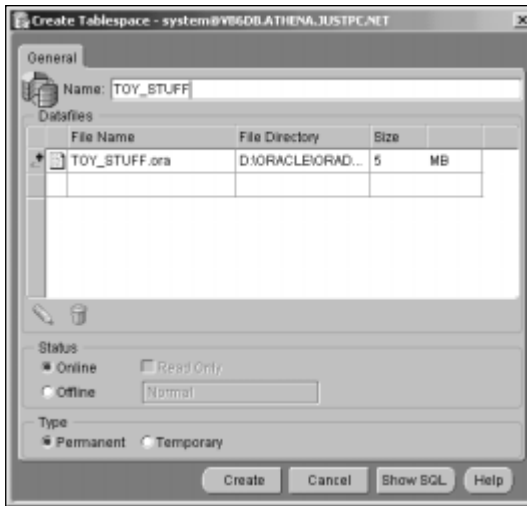


Figure 27-10: Provide the information for the tablespace.

Creating Tables and Indexes

Before you create the tables for your application, you should decide on a name for your database schema. A schema name will help you group all of the database objects you create into a single entity. While you don't explicitly create your schema in the DBA Studio, you reference its name each time you create a new database object.

Creating a table isn't much more than specifying a name for your table and the list of columns you want included. However, the more time you spend preparing to build your table, the better off you will be. Here are a few questions you may want to consider.

- ♦ What tablespace will hold your table?
- ♦ What is the name of your table? Your table name should be prefixed by the schema name that you want to use. If you don't specify a schema name, your user name will be used. Both identifiers can be up to 30 characters in length. While you can use spaces and other special characters as part of the table name, you will have to enclose the table name in quotation marks each time you use it. Also, this will make the table name case-sensitive. Also, remember that the table name must be unique within the schema you specify.
- ♦ What is the name of each column? The rules for naming a column are the same as naming the table.

- ♦ What is the data type for each column? The data type should be one of the data types listed in Chapter 26. You may have to choose the size of the field and the scale depending on the data type you select.
- ♦ Should the column accept **Null** values? Note that **Null** values should not be used in any column that is used as part of a primary or foreign key.
- ♦ Is there a default value for the column? This is an optional field that allows you to insert a specific value into the table instead of marking the column as **Null**.
- ♦ Should you perform a referential integrity check? This implies that each value entered in this field must exist in another table in this database.

Creating your table

Oracle DBA Studio allows you to create a table through a multi-step wizard or by displaying a worksheet-like form to fill out. I'm going to step you through the wizard since it simplifies much of the work. Then I'll show you how to modify the table using the form.

1. Start DBA Studio, select the appropriate database, and press the Create button to display the object list, as shown in Figure 27-11. Select Table from the list, make sure the Use Wizard check box is checked, and then press the Create button.



Figure 27-11: Starting the Create Table wizard.

2. In the first step of the Table Wizard (see Figure 27-12), you are asked to specify the name of the table, the schema, and the tablespace that should be used.

Table Wizard, step 1 of 13: Introduction

Introduction

What do you want the Name of the new table to be?
STATES

Which Schema do you want the table to be part of?
TOY

Which Tablespace do you want to create the table in?
TOY_STUFF

Cancel Help < Back Next >

Figure 27-12: Entering basic information about the table.

3. In the next step of the wizard (see Figure 27-13), you will enter the columns you want to include and their data type. You can also specify a default value for each column as well. Simply enter the information in the Properties of Column section of the form and press the Add button. You can also remove a column that you have added by selecting it in the Columns defined area of the form and pressing Remove.

Table Wizard, step 2 of 13: Columns Definition

Columns Definition

Columns defined:

- STATE
- STATENAME
- SALESTAXRATE
- Untitled-

Add Remove

Properties of column: <Untitled>

Column Name: -UNTITLED>

Column Datatype: VARCHAR2

Size: 10

Scale:

Does this column have a default value? If so, please enter.

Cancel Help < Back Next > Finish

Figure 27-13: Entering column information.

- In step 3 of the Table Wizard, you select the primary key (see Figure 27-14). While Oracle permits you to create the table without a primary key, I believe that every table you create should have a primary key. Choose the Yes, I Want To Create a Primary Key radio button to display the list of columns for the table, and then specify the order of the columns that comprise the primary key. If a column is not part of the primary key, leave the order field blank.

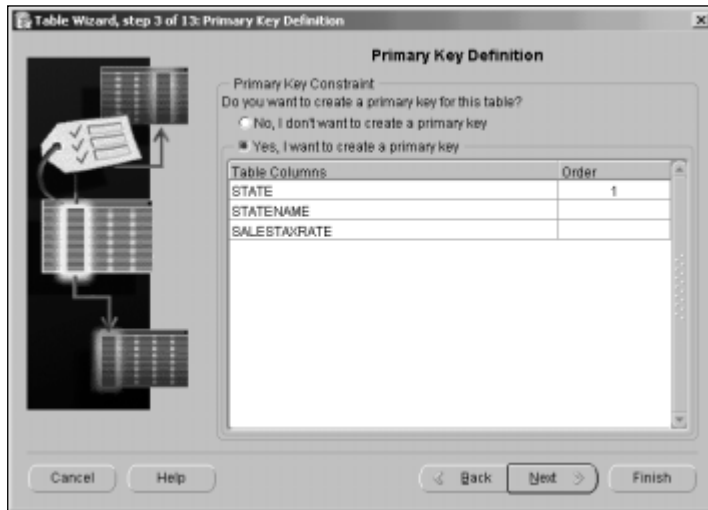


Figure 27-14: Selecting the primary key for your table.

- Pressing Next allows you to specify **Null** and **Unique** constraints on each field in your new table (see Figure 27-15). To modify a column, select the column in the Columns Defined area of the form. You should always ensure that both constraints are selected for the primary key. However, you should only use **Unique** for the primary key if you have a very good reason for doing so because each **Unique** clause you add after the primary key imposes a lot of extra work each time you insert a new row.
- Foreign key constraints are selected in step 5 of the Table Wizard (see Figure 27-16). For each column that references the primary key of another table, choose Yes, The Column is a Foreign Key and then specify the name of the Schema, Table, and Column that comprise the foreign key. If you have no foreign keys, then choose No, The Column is not a Foreign Key.

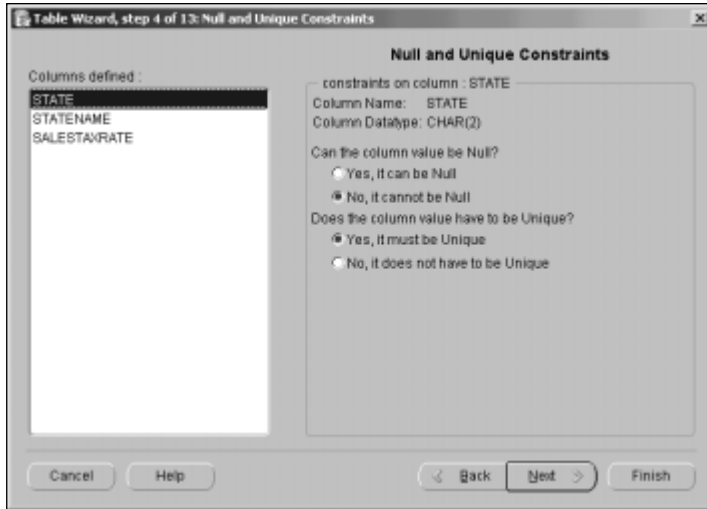


Figure 27-15: Choosing Null and Unique constraints for each column in your table.

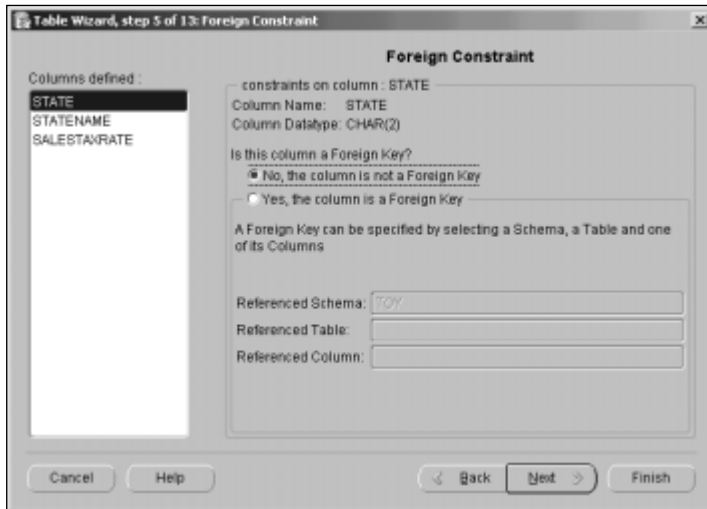


Figure 27-16: Defining foreign key relationships.

7. Step 6 allows you to define check constraints for each column (see Figure 27-17). These are tests that the value you insert into the column must meet before the row can be inserted or updated. Select the column in the Columns defined section of the form, and then add the constraint if desired.

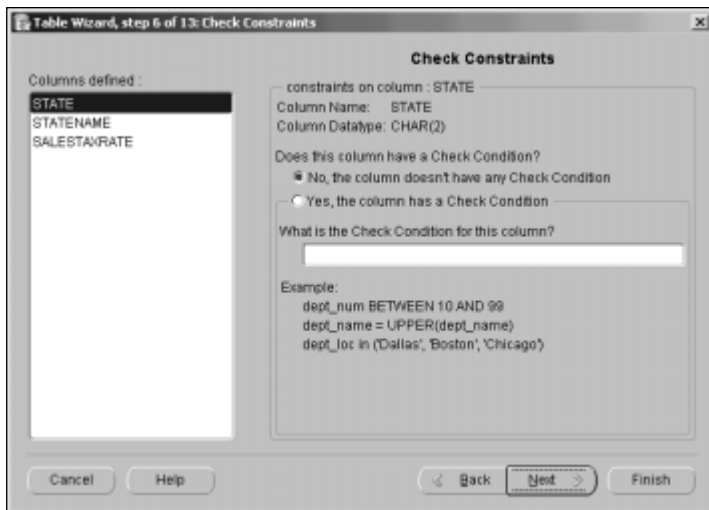


Figure 27-17: Creating check constraints.

8. The remaining steps of the wizard allow you to optimize how the table is stored. Unless you are a knowledgeable Oracle database administrator, I suggest that you skip these steps. Pressing the Finish button will create your table. When the wizard has finished, a message box will be displayed indicating that your table was successfully created.

Tip

Like Create Like, man: If you have to create a table that is similar to another table that is already in your database, consider clicking on the Create Like button, instead of the Create button. Select an existing table in tree view and press the Create Like button. Your new table will include all of the definitions that are in the table you previously selected. You may then make any modifications you want—add columns, delete columns, change names, data types, and so on. This can be much easier than you think.

Modifying a table

You can modify a table by right clicking on the table name from the tree view and choosing the Edit option from the pop-up menu. This will display the Edit Table property window, as shown in Figure 27-18. You may change any of the column's characteristics, including Datatype, Size, Scale, Foreign Key References (Ref), Nulls, and Default Value, all on this single form. Notice that you can't change any of the information at the top part of the form such as the name of the table, its schema or table space.

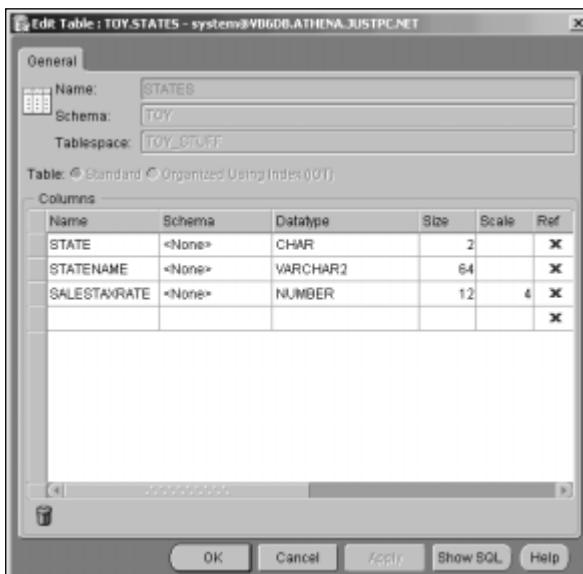


Figure 27-18: Modifying a table's characteristics.

You can make any changes you wish to a table as long as it doesn't have data in it. If your table does contain data, you can make changes as long as you follow these rules:

- ♦ A new column must accept **Null** values. If you wish to add a column that doesn't permit **Null** values, you add the column permitting **Null** values, then use an **Update** statement to assign a value to this column for every row in the table. Then you can modify the column to prohibit **Null** values.
- ♦ You always increase the size of a character-based column (**Char**, **Varchar2**, etc.). If you want to decrease the size, you must first change the value for each row in the table to **Null**, then make the change.
- ♦ You can always increase the number of digits or increase or decrease the number of decimal places in a **Number** column. To decrease the number of digits in the column, you must first ensure that each row contains a **Null** value for this column before making the change.
- ♦ You can change the data type assigned to a column only if the column's value for each row in the table is **Null**.

Tip

There's always a way if you really want to do something: One way to change the characteristics of a column is to add a temporary column to your table that accepts **Null** values, with all of the characteristics you want the changed column to have. Then use an **Update** statement and assign the value from the original column to the temporary column using whatever functions are necessary to convert the value properly. Then delete the old column from your table and give the original column's name to the temporary column.

Creating an index

You can always add an index to your table by clicking the Create button and choosing Index from the list of available database objects. This will display the Create Index property window (see Figure 27-19). Simply fill in the values for the name of the index, the schema it will be associated with, and the tablespace it will use for storage. Then choose the schema and table name for the table where the index will be applied. Next, choose the columns and the order they will appear in the index and press Create to build the index.



Figure 27-19: Creating an index.

Managing Security

Managing security is an important part of every database management system (DBMS). As you would expect, it involves three main processes: creating roles, creating users, and assigning permissions to both users and roles.

Creating roles

As you've seen before, a role is a way to bundle together permissions and other roles to form a single entity that can be associated with a user. Creating a role isn't very difficult. Simply press the Create button, and select Role from the list of

database objects. This will display the Create Role property window (see Figure 27-20). You must enter a name for the role and choose whether you want additional authentication from the user in order to enable the role.



Figure 27-20: Creating a new role.

On the Role tab, you can choose to add any existing roles to this role. This means you can create a single role that encompasses several other roles, thus simplifying security management.

Creating users

Using DBA Studio, you can create a user very easily by pressing the Create button and selecting Users from the list of database objects. This will display the Create User property window. This property window has three tabs: General, Role, and System Privileges, each of which holds part of the information available about a user. After filling out the information in each tab, simply press the Create button to create the new user.



Tip

Create Like strikes again: Creating a new user is one of the most painful processes any computer center has to manage. One way to avoid some of the pain is to create a standard user and then use the Create Like process to duplicate it, including all of its security roles and privileges. While you should still review everything to make sure that it is appropriate for the user you are creating, Create Like will help you avoid making simple mistakes.

Entering general information for a user

Figure 27-21 shows the General tab of the Create User property window. To create a new user, simply fill out the user's name and specify the password information. Then associate the user with the appropriate Default and Temporary tablespaces. If you check the Expire Password Now check box, the user will be required to change their password the first time they log on to the system. Also, selecting Locked means that the user is prohibited from using the user name until you select the Unlocked radio button.



The screenshot shows the 'Create User' dialog box with the following fields and options:

- Name:** SAMANTHA
- Profile:** DEFAULT
- Authentication:** Password
- Enter Password:** [masked]
- Confirm Password:** [masked]
- Expire Password Now
- Tablespaces:**
 - Default: USERS
 - Temporary: TEMP
- Status:** Locked Unlocked

Buttons at the bottom: Create, Cancel, Show SQL, Help.

Figure 27-21: Entering general information about a user.

Note

Three strikes and you're locked: You can enable a feature in your database that will lock a user out of the database if they specify an invalid password three times in a row.

Granting roles

In order for the user to perform any useful work, you must assign them to one or more roles (see Figure 27-22). At a minimum, they should be granted the `CONNECT` role, which will permit them to **Connect** to the database server. Simply select the roles you wish to grant to the user and press the arrow that is pointing down. The selected roles are copied to the Granted pane. When the user is created, the user will inherit the security permissions of each of the roles in the Granted pane.

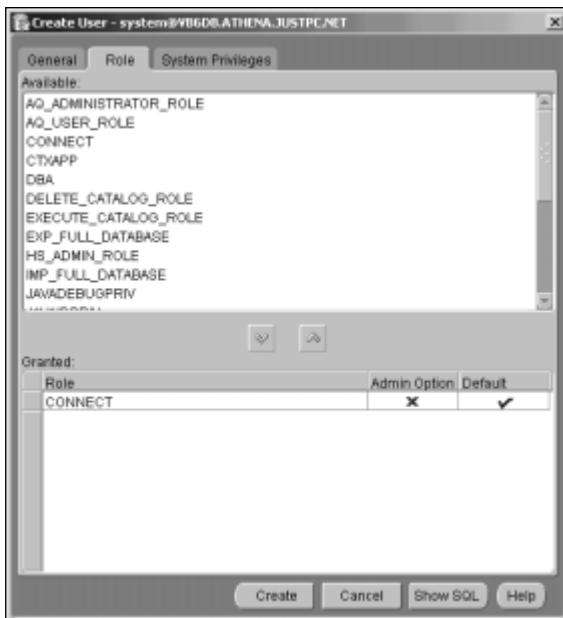


Figure 27-22: Selecting roles for a user.

Selecting system privileges

For the most part, a user shouldn't be assigned any of the privileges listed on the System Privileges tab (see Figure 27-23). These privileges should be reserved for database administrators and/or database operators. (See Chapter 26, "Overview of Oracle8i," for a more detailed discussion about the system privileges and what they permit the user to do).

Granting permissions

In order to secure your database, you must use one of the forms of SQL*Plus and enter the appropriate **Grant** and **Revoke** statements. You can't do this from DBA Studio. To simplify the permissions process, I suggest that you create separate roles for each type of access you want to provide for a table and grant the resources to the role. This means you might have a role for read access, one for update access, and another for administrative use. I would then repeat this process for each database object (view, stored procedure, etc.) that is part of the application.

I would then create another role for each access level for your application and assign each of the lower-level roles created earlier that are appropriate for the application. This simplifies the process of administration, since you would only have to associate one role for each application that the user has access to. Then if you change the application slightly, you need only manage the permissions associated with the lower-level roles.

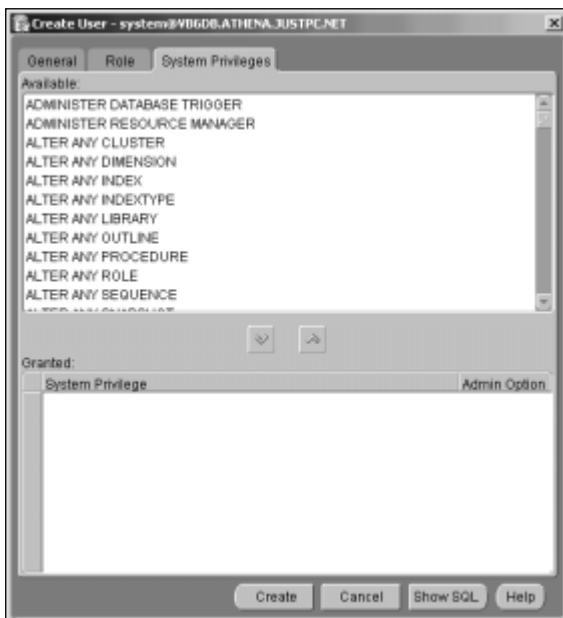


Figure 27-23: Selecting system privileges.

Thoughts on Creating Oracle8i Database Objects

Creating database objects in Oracle8i can be very simple or incredibly complex—it's your choice. Oracle8i includes many features that you can exploit when you build your database that I haven't discussed here. These features can make your database much more efficient. However, using these features can be difficult. This is another reason that an experienced database administrator is worth their weight in memory chips.

Not all of the options available to you via SQL statements are available through the DBA Studio. If you need to be able to manage the details of how the data files are allocated in a table space or define more complex relationships between tables, you will need to use one of the SQL*Plus variants to execute the specific SQL commands that perform the desired function.

Summary

In this chapter you learned:

- ♦ about SQL*Plus and some useful commands you may want to use.
- ♦ about Enterprise Manager and how to use it to access multiple database instances.
- ♦ about SQL*Plus Worksheet and how it differs from SQL*Plus.
- ♦ how to create a tablespace.
- ♦ how to create tables and indexes.
- ♦ how to manage security by creating users and granting roles.

