

Overview of Oracle8i

In this chapter, I'll introduce you to Oracle8i. I'll cover the key facilities included with the product, its architecture, and how security works. Then I'll cover how to use Visual Basic and ADO 2.5 with Oracle8i, including how to build a connection string and how Oracle's data types map into Visual Basic.

Overview

Oracle8i is the most widely used relational database management system (RDBMS) in the world. It's well known for its high performance and stability. It runs on most computer systems, including most Unix variants such as Sun's Solaris, Hewlett-Packard's HP-UX, IBM's AIX and OS/390 (MVS/ESA/XA), and even Linux. Of course it will also run on Microsoft Windows 2000/NT Server operating systems.

Oracle8i includes these key features:

- ◆ High performance
- ◆ Integrated Internet capabilities
- ◆ High availability with automated standby database
- ◆ Extensive Java and XML support
- ◆ Support for OLE DB and ODBC
- ◆ Support for data warehousing and Very Large Databases (VLDB)
- ◆ Integration with Windows 2000 services and tools



In This Chapter

Introducing Oracle8i editions and utilities

Understanding the Database architecture

Explaining the Database data types

Connecting to Oracle8i with ADO

Reviewing the Security model



Oracle8i editions

Oracle8i comes in four editions: Lite, Personal, Standard, and Enterprise. Since the same core database engine architecture is used for all of these editions, compatibility is assured throughout the product line.

The Lite Edition

Oracle8i Lite is targeted at environments where the smallest possible database system is needed, like mobile laptop computers, handheld computers, and information appliances. It includes built-in replication to Oracle8i Standard Edition and Oracle8i Enterprise Edition to facilitate easy exchange of information. While some features of the larger versions of Oracle8i are missing, you can run Oracle8i Lite Edition with as little as one megabyte of memory and five megabytes of disk space.

The Personal Edition

Oracle8i Personal Edition supplies a single user solution that is targeted at developers who want to develop and test their programs in a standalone situation and then deploy them against a Standard Edition server or an Enterprise Edition server. You can also use this Edition in situations where your application supports only a single user on a single computer.

You can run Oracle8i Personal Edition on Windows 98, Windows 2000, and Windows NT. It includes all of the same features and options found on Oracle8i Enterprise Edition, including Advanced Queuing and Oracle Partitioning. Note that some features may rely on facilities that aren't present in Windows 98.

 Tip

Freebie copy of Oracle8i (at least for 30 days): You can download a demo version of the Personal Edition from Oracle's Web site at technet.oracle.com. You will be required to register before you can access this site, but there are no charges for registering. This Web site also has a wealth of information that you will find useful when working with Oracle databases.

The Standard Edition

Oracle8i Standard Edition includes all of the tools necessary to build robust multi-user applications. It represents an affordable alternative to the Enterprise Edition when you don't require all of the advanced features. Some of the key features of the Standard Edition include:

- ♦ **Extensive Java support.** While not important to most Visual Basic programmers, this feature is useful when developing multi-platform applications.
- ♦ **Windows specific support.** Includes native ODBC and OLE DB drivers, support for Microsoft Transaction Server, App Wizard for Visual Studio, and OLE Objects for Oracle.

The Enterprise Edition

Oracle8i Enterprise Edition is Oracle's high-end database management system. It is targeted at applications that experience high volumes of transactions and that require very high levels of availability. Some of the key features in the Enterprise Edition include:

- ♦ **Transparent Application Failover/Automated Standby Database** — a key tool for providing high availability. It allows you to have standby database server that is ready to take over in case of a database or application failure.
- ♦ **Oracle Parallel Server** — a high performance feature that allows you to divide your database into multiple chunks, where each chunk exists on a separate physical computer. This allows for higher performance, since multiple computers are actively processing data. The database servers are clustered together to form a single view of the system to make it easy to use.
- ♦ **Oracle Partitioning** — a feature that allows you to break large tables into smaller, individually-managed tables, which can be spread across multiple servers using Oracle Parallel Server. This feature is extremely useful when managing data warehouses.
- ♦ **Internet File System** — a feature that makes an Oracle8i database appear as a shared network drive. It supports a wide variety of standard protocols, including HTML, FTP, and IMAP4, which allows a large number of users to store and retrieve data using a platform that offers better reliability than a normal networked disk drive.
- ♦ **Oracle Time Series and Oracle Spatial** — tools that are useful when building a data warehouse. Oracle Time Series allows you to stored time stamped data efficiently in an Oracle database, while Oracle Spatial allows you to analyze data such as the proximity of a customer to a store location.

Note

The best things in database aren't free: Some of the features available in the Enterprise Edition are extra cost items. Check the Oracle Web site for more information about which items aren't standard and how much they cost.

Oracle8i utilities

Oracle8i has the following utilities that you will use while developing your applications: SQL*Plus, Enterprise Manager, SVRMGR, PL/SQL, Net8, and Oracle Management Server. The key utility is called SQL*Plus and is used for a number of different tasks, from creating tables to testing stored procedures. Enterprise Manager is a tool similar to SQL Server's Enterprise Manager and is used to help you design databases and perform other system management-related functions.

SQL*Plus

SQL*Plus is a general-purpose query tool like Microsoft's Query Analyzer. This tool allows you to enter SQL statements and execute them. There are two versions of this tool. One operates in a DOS window, while the other runs in its own window (see Figure 26-1). Both versions are command-driven (i.e., you type commands rather than selecting menu items and/or pressing buttons).

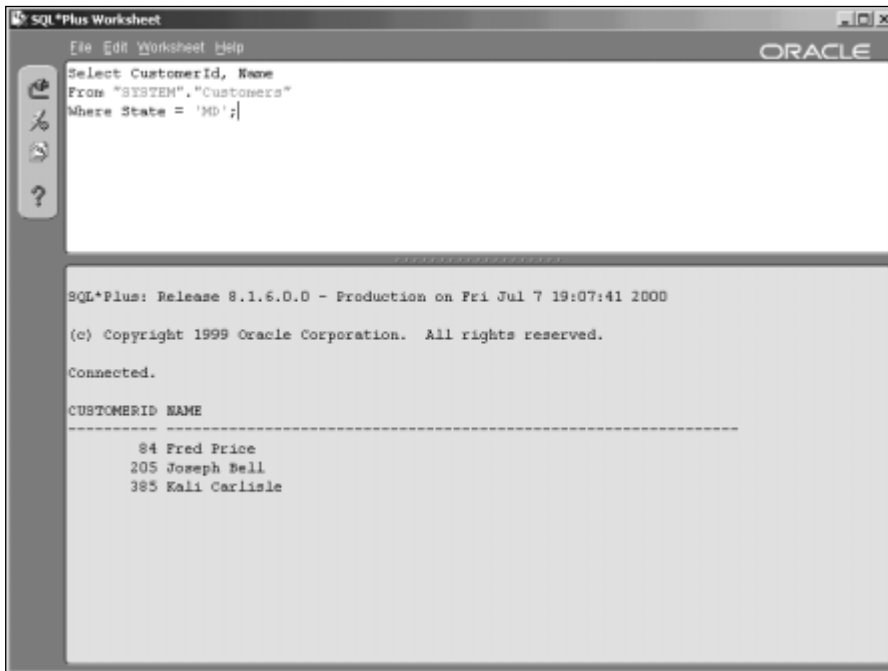


Figure 26-1: Running SQL*Plus Worksheet in a window.

Enterprise Manager

The Enterprise Manager utility assists you with functions typically performed by a database administrator, such as creating databases, creating users, and monitoring the activity of your databases (see Figure 26-2). It also includes a number of tools, such as SQL*Plus Worksheet, that allows you to enter, edit, and run SQL commands in a graphical environment, as well as DBA Studio, which can help you design your databases.

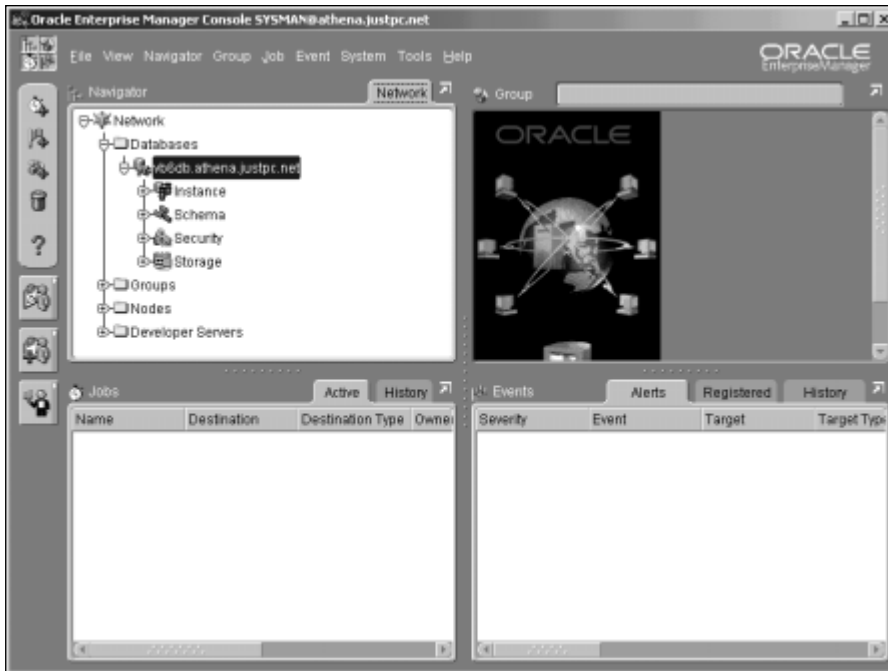


Figure 26-2: Running Enterprise Manager.

SVRMGR

The SVRMGR utility allows you to control the databases running on your server (see Figure 26-3). You can use this tool to start and stop your database, as well as run other SQL commands. Like SQL*Plus, it has both windowed and DOS-based versions.

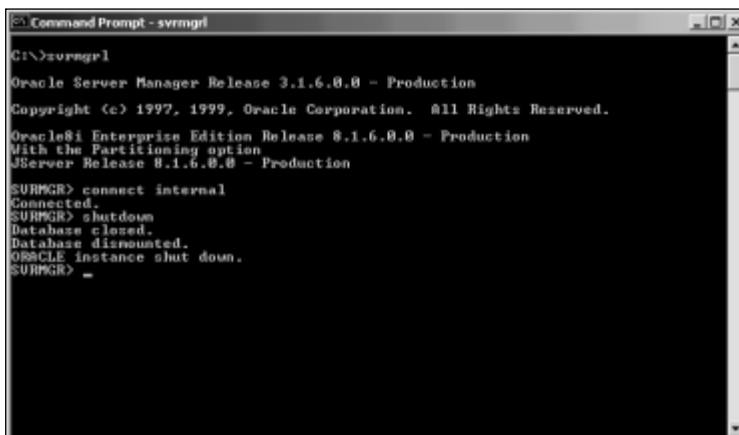


Figure 26-3: Running SVRMGR in a command prompt.



Tip

Bigger is better: The normal DOS window displays only 25 lines. You can increase the number of lines displayed to 43 or 50 by right clicking on the icon in the upper-left corner of the window and choosing Properties and then the Screen tab. You can also enable the scrollbars to display even more lines. Both options make it easier to see your results before they scroll off the top of the screen.

PL/SQL

PL/SQL is Oracle's implementation of the SQL standard. It includes a number of extensions that make it a very powerful language for creating stored procedures. As you might expect, PL/SQL is not compatible with T-SQL in many areas, such as procedural statements and data definition statements. For the most part, statements like **Select** and **Update** will work the same as they do in other database systems.

Net8

Net8 isn't really a utility as much as it is a critical part of the network communications required to operate Oracle8i. It permits you to define various external communications protocols for different types of clients and translate them into an internal database protocol that is independent of the clients. On your database server, Net8 is also used to handle database-to-database communications for things like remote database access and replication. A copy of Net8 also runs on your local computer. The OLE DB provider passes your database requests to Net8, which in turn routes them to the appropriate database server.



Note

By any other name: Net8 was known as SQL*Net in previous versions of the Oracle database server.

Oracle Management Server

Oracle Management Server is a utility process that allows you to manage the various components in an Oracle8i database system. It runs as its own process and maintains its own separate security system. It is tightly integrated with Enterprise Manager and provides facilities that allow Enterprise Manager to manage multiple database instances.

Understanding the Database Architecture

Like SQL Server, Oracle8i uses a true client/server architecture. This means that the database server runs in a different address space from the client program, and the client and server can be run on different computers. Unlike SQL Server, Oracle8i can run on a variety of hardware platforms. This means that some of the functions implemented on Windows will look and act a little differently than applications designed for non-Windows operating systems. However, the core Oracle8i database architecture is the same no matter which operating system is hosting the database server.

Key database objects

Oracle8i consists of one or more databases on the database server. Each database is known as an *instance*. Unlike SQL Server, which maintains a single database server with multiple databases beneath it, each Oracle8i database server is a unique instance and requires an independent software installation.

Tablespaces

Each database instance contains a series of objects known as *tablespaces*, which hold the data for your database. Each tablespace contains one or more physical files that are used to hold a collection of tables and indexes. One tablespace is reserved for system information, while other tablespaces are used to hold application data.

A table (or table partition if you partition the tables) can't span more than one tablespace. This means that there must be sufficient space in the tablespace for all of the tables in the tablespace. If you run out of disk space, you can either increase the space available in the existing physical files or add additional physical files to the tablespace.

Tables

By now, you are familiar with tables, so I won't go into all of the details about rows and columns, but there are a few unique aspects to Oracle's implementation. Each row in the database has a unique value known as a *rowid*. This value is used to locate the row in the database.

Tables that are often accessed together can be grouped together to share the same physical storage. This process physically places the rows close together for tables that are joined together using **Select** statements. This arrangement can make a big difference in performance, because when the data to be joined is in close proximity, fewer I/Os are required. This technique is known as *clustering*.

Schemas

The *schema* represents a view of the information in your database. It represents a set of tables and other database objects that can be accessed by a user. As such, a schema represents a good way to secure database objects. Only the database administrator is permitted to see the database objects in every schema in the database.

It is important to note that schemas are logical views of the database and are in no way coupled to tablespaces, unless of course the database administrator chooses to assign all of the tables for a particular schema to a particular tablespace.

Sequences

A *sequence* is an object that generates numbers according to a specified order. This allows you to generate unique values for things like counters and timestamps to guarantee that each row added to a table has a unique value.

Stored Procedures, functions, and packages

Recall that *stored procedures* are pre-compiled programs written in PL/SQL that are compiled and saved in the database server for quick execution. You have your choice of writing procedures that don't return a value or *functions* that return a single value. Stored procedures can also return a recordset containing a collection of rows selected from a table or a view.

 Cross-Reference

See Chapter 25, "Stored Procedures," for a detailed discussion of stored procedures.

A *package* is a way to group similar procedures, functions, and variables into a single Oracle8i object. This has two primary advantages: first, that because all of the objects are compiled into a single entity, calls between functions and procedures are faster, since there is no need to search for and load external routines; and second, that managing one package is easier than managing each of the individual routines.

Triggers

Triggers are one way you can enforce referential integrity. Each time you execute a **Delete**, **Insert**, or **Update** statement, a stored procedure will be called that can check the changes to ensure that they are being made properly. For instance, assume that you have two tables with a one-to-many relationship. You can create a trigger that will delete all of the associated rows in one table when you delete a record in the other table.

 Note

In the key of SQL: You can also use foreign keys to enforce referential integrity.

Synonyms

Synonyms are basically just another name for a database object. Suppose you want to change the name of a table, but you're not certain that you've updated all of the applications that use the old name. You can create a synonym for the table using the old table's name.

Memory architecture

One of the keys to Oracle's performance is its memory architecture. The key to database performance is being able to avoid disk I/O by proper buffering of information in memory.

Tip

A gig here and a gig there makes for good performance: One thing I can't stress enough is that you need lots of memory to run a database server. A perfect relational database would have enough memory to hold everything it stores on disk. In most cases, this isn't possible because you have too much data on disk — but it's nice to dream.

System Global Area

Each instance of an Oracle database has a block of memory called the System Global Area (SGA), which is used to hold shared resources. All users have access to this area. It is used for maintaining various types of buffers, including the library cache where the SQL cursors are maintained, and the data dictionary cache, where the structures that describe the objects in the database are kept. This information is extracted from system tables, including:

- ♦ **USER_OBJECTS** — contains information about the various objects in the database, plus a reference to other tables that contain more detailed information about a specific type of object.
- ♦ **USER_TABLES** — contains the definitions of the tables in the database.
- ♦ **USER_TAB_COLUMNS** — contains the definitions for the columns in the table.
- ♦ **USER_SOURCE** — contains the source code for stored procedures.
- ♦ **USER_SEQUENCES** — contains the definitions of the sequences in your database.
- ♦ **USER_INDEXES** — contains information about the indexes associated with each table.

Program Global Area

The Program Global Area (PGA) maintains information on the active processes running in the database. These processes either perform work on behalf of a client application or perform background processing and monitoring of the database server. For instance, when a client application requests a recordset, a server process will retrieve the necessary rows (hopefully from the SGA, otherwise from disk) and return them to the client. A background process, on the other hand, might collect performance statistics, which can be analyzed to determine bottlenecks.

Network architecture

While SQL Server is managed through a single server process, Oracle8i relies on a number of different server applications. Each database instance has its own server process, while there is another server application devoted to networking. The networking process uses a piece of software known as Net8 (described earlier in this chapter). Net8's function is to receive requests from client applications and pass them on to the appropriate database server for processing. It is also responsible for handling communications between the various database instances and for facilitating communications between various database instances on different computer systems.

Oracle8i data types

Oracle8i supports a wide variety of data types, as shown in Table 26-1.

Table 26-1
Oracle8i Data Types

<i>Oracle8i Data Type</i>	<i>Visual Basic Data Type</i>	<i>Comments</i>
Bfile	Byte Array	Contains a binary large object up to 4 gigabytes in length.
Blob	Byte Array	Contains a binary large object that can be up to 4 gigabytes in length.
Char	String	Since Char fields always have a fixed length, the length of the String value will be the same as the length of the field. A Char field may contain up to 2,000 characters.
Clob	String	A character-oriented large object up to 4 gigabytes in length.
Date	Date	A date and time value that is more accurate than the Date data type. The value will be rounded as needed.
Decimal	Currency	Contains a packed decimal number with up to 38 digits of precision.
Float	Single, Double	Uses the same storage format as Number.

<i>Oracle8i Data Type</i>	<i>Visual Basic Data Type</i>	<i>Comments</i>
Long	String	A character string whose maximum length can be up to 2 gigabytes.
Long Raw	Byte Array	A binary string whose maximum length can be up to 2 gigabytes.
Nchar	String	Contains a fixed-length Unicode string with up to 2,000 characters.
Nclob	String	A character-oriented large object containing Unicode characters.
Number	Currency	Contains a packed decimal number with up to 38 digits of precision.
Nvarchar2	String	Contains a variable length Unicode string with up to 4,000 characters.
Raw	String	A binary string up to 2,000 bytes long.
Rowid	String	The internal format is automatically converted to a String value.
Varchar2	String	Contains a variable length String with a maximum length of 4,000 characters.

Connecting to Oracle8i

Building an ADO connection string for an Oracle8i database involves specifying values for the provider and the database, as shown below:

```
Provider=MSDAORA.1; Data source=vb6db.Athena.justpc.net
```

You can also include the `User ID=` and `Password=` keywords if you want to add that information as part of the connection string, or you can supply them as arguments to the `Open` method.

You can also configure the Oracle OLE DB provider via the Data Link Properties window. The first step is to select the Microsoft OLE DB Provider for Oracle (see Figure 26-4). After pressing the Next button, you can enter the name of your Oracle database server and optionally specify your user name and password (see Figure 26-5). You can verify that the information you entered was correct by pressing the Test Connection button.

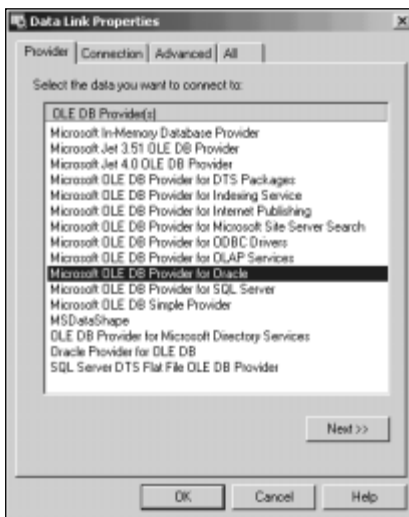


Figure 26-4: Choosing the Oracle provider.



Figure 26-5: Specifying values for the key connection properties.

Tip

There's always a choice: Oracle also has a provider for OLE DB that you may wish to try, and there are third party OLE DB providers that may outperform both Microsoft's and Oracle's, though at a considerable cost in time and money.

Note

A provider is only provider: The Oracle provider supplied by Microsoft only translates the calls from your Visual Basic program into something that the Oracle8i client software on your computer will understand. You must install a copy of Net8 on your client computer for any program to communicate with an Oracle8i database.

Oracle8i Security

In order to access an Oracle8i database, you must be properly authenticated and authorized. Authentication ensures that the user name and password associated with an individual is valid before access is granted to the database server. Once the users have been granted access to the server, everything that the users can perform is based on the permissions they have been authorized.

Authentication

In order to access an Oracle database, every user must have a valid user ID and a valid password. This information is usually kept in the database, though Oracle does include hooks to use an external directory, such as the Windows 2000 Active Directory or Kerberos.

Authorization

Inside Oracle8, you must be granted the privilege to access a particular database object or command. In fact, without the proper privileges, you can't even connect to the database, even if you had a valid user ID and password. Privileges are managed using the **Grant** and **Revoke** SQL statements, or by using the Enterprise Manager utility's GUI interface. The two types of privileges, object privileges and system privileges, are discussed next. Object privileges dictate the user's access to a specific database object, while system privileges determine the user's ability to execute specific Oracle and SQL statements.

Object privileges

Object privileges specify which commands can be used against specific database objects. Depending on the object, you apply the privileges shown in Table 26-3. The syntax for the **Grant** statement to apply object privileges looks like this:

```
Grant <privileges> On <object> To <users> [With Grant Option]
```

where <privileges> contains a list of one or more privileges separated by commas selected from Table 26-2; <object> specifies the database object that the privileges are associated with; and <users> is a list of one or more user names separated by commas that will receive the privileges or the keyword **Public** indicating that any database user may have the privilege. The **With Grant Option** clause allows the user receiving the privilege to pass it along to other users by using the **Grant** statement.



Caution

With Grant Option security can really be compromised: Because you passed on the ability to control security to another user, they can pass it on to others. Thus, people that you may not want to have access to the object may get access from someone else. While using the **With Grant Option** is necessary to implement a distributed security management system, you must use it with care to ensure that only authorized users can access your database.

Table 26-2
Object Privileges

<i>Statement</i>	<i>Associated Privileges</i>
Alter	Tables, Sequences, Snapshots
Delete	Tables, Views
Execute	Procedures, Functions, Packages, Libraries, Types
Index	Tables
Insert	Tables, Views
Read	Directories
References	Tables
Select	Tables, Views, Sequences, Snapshots
Update	Tables, Views, Snapshots

For example, if you wanted to allow Jill to retrieve and update information in the Customers table, but not insert or delete rows, you would use this **Grant** statement:

```
Grant Select, Update On Customers To Jill
```

System privileges

System privileges control the ability to execute specific Oracle and SQL statements. Like object privileges, system privileges are based on the concept of allowing a user to use a specific command. Unlike object privileges, system privileges allow you to use that command on any object or against the database itself (see Table 26-4). Note that some of the privileges describe general capabilities within the database server, and not just the actual commands used to perform a task. These are also listed in Table 26-4.

```
Grant <privileges> To <user> [With Admin Option]
```

where **<privileges>** is a list of one or more system privileges, separated by commas, chosen from Table 26-3, and **<user>** is a list of one or more user names separated by commas.

The **With Admin Option** clause allows the user to pass the privilege to another user.

Table 26-3
System Privileges

<i>Command</i>	<i>Associated Privileges</i>
Alter	Any Cluster, Any Index, Any Procedure, Any Role, Any Sequence, Any Snapshot, Any Table, Any Trigger, Any Type, Database, Profile, Resource Cost, Rollback Segment, Session, System, Tablespace, User
Analyze	Any
Audit	Any, System
Backup	Any Table
Become	User
Comment	Any Table
Create	Any Cluster, Any Directory, Any Index, Any Library, Any Procedure, Any Sequence, Any Snapshot, Any Synonym, Any Table, Any Trigger, Any Type, Any View, Cluster, Database Link, Directory, Library, Procedure, Profile, Public Database Link, Public Synonym, Role, Rollback Segment, Session, Sequence, Snapshot, Synonym, Table, Tablespace, Trigger, Type, User, View
Delete	Any Table
Drop	Any Cluster, Any Directory, Any Index, Any Library, Any Procedure, Any Role, Any Sequence, Any Snapshot, Any Synonym, Any Table, Any Trigger, Any Type, Any View, Library, Profile, Public Database Link, Public Synonym, Rollback Segment, Tablespace, User
Execute	Any Procedure, Any Type
Force	Any Transaction, Transaction
Grant	Any Privilege, Any Role
Insert	Any Table
Lock	Any Table
Select	Any Sequence, Any Table
Update	Any Table
Non-command based privileges	Manage Tablespace, Readup, Restricted Session, Unlimited Tablespace, Writedown, Writeup

Roles

Managing permissions for individual users can be overwhelming on a database system with a large number of users. Thus, Oracle8i also uses the concept of roles to simplify database administration. A *role* represents a collection of privileges that can be assigned to a user. If you assign a role to a user, the user inherits all of the privileges in the role. Likewise, if you add a privilege to an existing role, all of the users that are assigned that role will automatically receive the new privilege.

When granting privileges to a role, you use the same **Grant** statement that you would use for an object privilege. You simply specify the name of the role in place of the user. For instance, you would use the following statement to grant the **Select** privilege to the Clerks role:

```
Grant Select To Clerks
```

When granting a role to a user, you use the form of the **Grant** statement used to associate system privileges to a user. In this case, the role is substituted for a system privilege. The following statement shows how to grant the Clerks role to the user Samantha. When executed after the previous **Grant** statement, Samantha will be able to use the **Select** statement.

```
Grant Clerks To Samantha
```

Thoughts on Oracle8i

Oracle8i is a high-performance database that runs on many different hardware and software platforms. It is also more difficult to install and maintain than SQL Server. If you need the biggest, fastest database management system in existence today, Oracle8i should be at the top of your list, since it runs on the biggest computer systems available. Companies like Amazon.com and eBay rely on Oracle database servers.

However, this power comes at a cost. You really need an experienced Oracle DBA to maintain your database. Unlike SQL Server, which contains a large number of wizards to perform routine tasks, Oracle8i doesn't. If you're not very familiar with Oracle databases in general, you need someone to help you set up the various processes and ensure that they work properly.

The other downside to Oracle8i is its cost. Oracle knows that Oracle8i is the highest performance database on the market today and charges accordingly. While it is hard to argue about the value of Oracle8i, it is important to note that solutions from other database vendors may be less expensive. Of course, if your application is large enough, then your only option may be Oracle, no matter how much the competition charges for their software.

I sometimes look at this issue from a mainframer's perspective. In the mainframe world, operating system software, including database management systems, are very complex products that are highly adjustable. You can't simply install a package and begin using it. It usually takes anywhere from a few days to a few months to install a piece of software. Some of this time is spent selecting options and setting various parameters in order to make the software work, while the rest of the time is spent ensuring that you have the right options set.

A person that is experienced in the software package can set the options much more quickly than a person learning the software. But in the long run, having lots of options to select and parameters to set allows an organization to make the software really fly by, eliminating the options not needed and tweaking the other settings so that each function performed by the database gets the resources it needs.

Summary

In this chapter you learned:

- ♦ about the Oracle8i database server and the various editions that are available.
- ♦ about the major utilities and key components.
- ♦ about the Oracle8i database architecture.
- ♦ about the data types available in Oracle8i and how they map into Visual Basic data types.
- ♦ how to connect to an Oracle8i database with ADO.
- ♦ about Oracle8i security.



