CHAPTER 1

Introducing Oracle8i

In This Chapter

Introducing new Oracle8i features

Installing Oracle8i

Examining the starter database contents

Installing WebDB

Installing Oracle Enterprise Manager

Taking a quick tour of Oracle8i

"The Internet changes everything." Although we're not sure who first said this, Oracle certainly believes it. The release of Oracle8i and related products, such as WebDB, interMedia, and Internet File System (iFS), represent a significant push by Oracle to leverage its database expertise and become a major player in the growth of Internet computing. No longer is Oracle just a database. Now it is the Database for the Internet. Even the name has been changed, with an "i" being added to "Oracle8," giving us "Oracle8i" instead.

Oracle has done much more than just change the name of its product — the company has also added significant new functionality. A Java engine now runs within the database. Oracle interMedia enables the database to store the rich variety of sound, video, and image content so often found on the World Wide Web. Oracle Application Server (OAS) allows you to deploy robust, three-tier applications accessible to millions of people around the world. Internet File System turns the database server into a universally accessible file server. Clearly, Oracle has made some exiting changes, and at the core of all these is the database itself — Oracle8i.

This chapter tells you about many of the new features that set Oracle8i apart from its predecessors. We'll then go over some of the details that you should remember during the installation process. We'll end with a look at the starter database, followed by a quick tour of the software that you will be using to administer your database.

Introducing New Oracle8i Features

Oracle8i represents a significant increase in functionality over Oracle8. If you are still using Oracle7, the differences are even more profound. However, we're talking about an *increase* in functionality, not *changed* functionality. Oracle's backward compatibility is excellant, and applications developed for

Oracle7 and Oracle8 should work under Oracle8i as well. Oracle8i's most significant new features fall into these categories:

- ✦ Java
- ✦ Data warehousing
- ✦ Resource management and performance
- ✦ OLTP
- ✦ Ease-of-use enhancements
- ✦ Backup and recovery
- ✦ Security
- ✦ PL/SQL

This chapter doesn't discuss every last new feature, but every effort has been made to cover the most significant ones. The next few sections describe in detail the new features for each category in the preceding list.

Java

Java is certainly the big news. Oracle8i includes an embedded Java Virtual Machine (JVM) developed by Oracle specifically for its database. It runs within the same process and memory space as the database engine itself, putting it on par with Oracle's other language, PL/SQL.

You can write stored procedures, functions, and triggers in Java. Furthermore, Oracle supports two-way interoperability between PL/SQL and Java. Your Java code can make calls to PL/SQL procedures and functions, and your PL/SQL code can make calls to Java procedures and functions.

To enable you to access relational data from Java, Oracle has embedded Java database connectivity (JDBC) and SQLJ support within the database. JDBC is an industry standard set of classes and methods that allows you to query and manipulate relational data. It's similiar in concept to open database connectivity (ODBC). SQLJ is a precompiler. It works in much the same way as Pro*C and Pro*COBOL, allowing you to embed SQL statements within a Java program. When the Java program is compiled, the SQLJ precompiler translates the SQLJ statements into JDBC method calls.

Data warehousing

Oracle continues to enhance its support for data warehouses and other large databases. Significant new features include the following:

- ✦ **Support for materialized views and automatic query rewrites.** A *materialized view* is one where the query is executed once when the view is

created, and the results are saved for future reference. The most obvious use is to store summaries of information contained in large tables. *Automatic query rewrites* cause queries that later attempt the same summarization to use the materialized view instead, resulting in a significant performance boost.

- **Transportable tablespaces.** *Transportable tablespaces* allow easy movement of data from an operational database to a data warehouse. Exporting data from one database and importing it into another can be time consuming. If you set things up correctly, you can now accomplish the same task more quickly by copying one or more tablespace files.
- **Two new aggregate operators,** `CUBE` **and** `ROLLUP`. The `ROLLUP` operator allows for the automatic generation of subtotals in the results returned by a `GROUP BY` query. The `CUBE` operator generates subtotals across multiple dimensions and may be used to generate crosstab reports.
- `SAMPLE` **function**. It allows you to write data-mining queries that randomly sample a specific percentage of rows in a table.

Resource management and performance

Enhancements under the resource management and performance catagory include the following:

- **Support for function-based indexes.** This feature allows you to create an index on the result of a SQL function applied to one or more underlying columns in the table. An index on `UPPER(employee_last_name)`, for example, would make case-insensitive name searches not only possible, but also efficient.
- **Better support for index-organized tables.** An *index-organized table* is one in which the entire table is created as an index. All the data is stored in the index, and there really is no underlying table. Oracle8i allows secondary indexes to be created on these tables, allows them to store large objects, and allows you to add or modify columns using the `ALTER TABLE` command.
- **Support for resource management.** You now have the ability to allocate CPU resources among different users, or groups of users. You could, for example, limit decision support users to 10 percent of the available CPU time during the day, but allow them to use up to 80 percent at night.

OLTP

Online transaction processing (OLTP) represents one of the more common uses for relational databases. Oracle now supports stable optimizer plans. This feature allows you to store a set of optimizer-generated execution plans for an application's SQL statements on a reference database and to export those plans to other databases. An OLTP application typically generates a predefined set of SQL statements, and once an application is tuned, you generally want the same execution plan to be used anywhere that you deploy that application.

Oracle8i supports several new trigger types. You may now create triggers for the following data definition language commands:

- CREATE
- ALTER
- DROP

In addition, you may create triggers that execute in response to the following database events:

- User logon
- User logoff
- Database startup
- Database shutdown
- Server errors

Ease-of-use enhancements

Oracle8i includes several enhancements that make tasks easier for the database administrator. These include the following:

- DROP COLUMN **command.** No longer do you need to go through the often-agonizing process of dropping and recreating a table when all you really want to do is drop one column from that table.
- **Configuration Assistant.** Configuration Assistant is a wizard-based application that automates the task of creating a new Oracle database. Just answer the questions, and Configuration Assistant will write the create script for you. It will even run it for you if you like.
- **An enhanced Enterprise Manager.** Oracle Enterprise Manager Administrator may now be deployed in either a two-tier or a three-tier environment. Deployment in a three-tier environment allows you to perform DBA functions via your Web browser.
- **Support for online index rebuilds.** Previous releases of Oracle would lock a table during an index rebuild to prevent users from making changes during the rebuild process. Oracle8i allows users to update a table during an index rebuild.

Backup and recovery

Oracle8i implements the following features related to backup and recovery:

- Support for multiple archive log destinations, making it easier to maintain multiple copies of archived log files

- ✦ Support for multiple archiver processes, reducing the likelihood of bottlenecks caused by redo log files being filled faster than they can be archived
- ✦ The ability to set an upper limit on the time that you are willing to allow for crash recovery
- ✦ LogMiner, which gives you the ability to extract and analyze information from redo log files

Security

Security-related enhancements include the following:

- ✦ **Fine-grained access control.** This allows you to implement row-level security at the database level.
- ✦ **Application contexts.** This allows you to expand on fine-grained access control by making it application-specific.
- ✦ **Support for the invoker's rights model.** This allows you to write PL/SQL procedures and functions that execute using the privileges granted to the invoking user, as opposed to the privileges granted to the definer.

PL/SQL

Oracle8i brings a number of PL/SQL enhancements to the table. The following are the most significant:

- ✦ **Support for autonomous transactions.** This allows you to code completely independent transactions that you can commit regardless of whether the calling transaction commits.
- ✦ **Embedded dynamic SQL.** This provides a simpler and more efficient way to execute dynamic SQL than was previously available using the `DBMS_SQL` package.
- ✦ **Parameter passing by reference.** This improves the performance of applications that pass large structures as arguments to PL/SQL procedures and functions.
- ✦ **PL/SQL bulk binds.** This allows you to send a SQL statement to the server, together with an array of values. The SQL statement is executed once for each element in the array. The result is the same as if you had sent each element of the array one at a time, but it's much faster to send it all in one shot.

Installing Oracle8i

The process of installing Oracle software can be divided into three phases: the preinstallation phase, the installation phase, and the postinstallation phase. Most of the work should happen in the preinstallation phase. This is where you plan out the products that you want to install, decide on the hardware that you want to use,

make sure that the hardware is configured properly, and read over the installation instructions. If you do your planning right, the installation phase consists of little more than watching the installer copy files.

Preinstallation preparation

The most important part of the installation process is the preperation phase. Resist the temptation to skip this phase. People often rush to put the CD in the machine and start the installation process without really thinking through just what they want to do, and without taking the time to prepare the environment. Such a rushed approach often results in a failed installation, which can lead to a lot of frustrating cleanup work, not to mention having to start over again. At best, you will end up doing a lot of rework in the postinstallation phase that you could have avoided by doing some preplanning.

Reading the Installation Guide

The process for installing Oracle varies from one operating system to the next, from one hardware platform to the next, and from one release of Oracle to the next. For each combination of operating system, hardware platform, and software release, Oracle produces an installation guide. Often, you will receive a printed version of the installation guide along with the media (usually a CD) containing the Oracle software. If you paid for the software, this is most likely the case. If you have a demo CD that you picked up free at a trade show, you should be able to find the installation guide on the CD in HTML format.

The installation guide explains the entire installation process, tells you about any prerequisites that need to be in place, and explains any configuration options for the product. The Oracle8i installation guide, for example, explains the difference between a minimal installation and a typical installation. You have to read the installation guide to find out what each configuration has to offer, as well as what each configuration requires.

If you're installing Oracle software in a UNIX environment as opposed to Windows NT, it's even more critical to read the installation guide carefully. You will likely find that you must set a number of UNIX system parameters in certain ways, and you will need to work closely with your system administrator to make the installation succeed.

A False Sense of Security

If you work with Oracle in a Windows NT environment, you may find yourself lulled into a false sense of security. The Windows NT installation is very easy to do and requires a minimum of preparation. Indeed, you are often able to just slam the CD into the machine, run setup, and install Oracle without any advance preparation. This is not a good habit to get into; when you move into other environments, such as UNIX, the installation process won't be nearly so forgiving.

Deciding What to Install

As you read through the installation guide, part of what you should be doing is deciding exactly which product options you want to install. The Oracle8i server software has many optional components to choose from. When you run the installer, you will be given a choice between a typical installation and a custom installation. A custom installation allows you to pick and choose the exact components that you want. The trick is to *know* what you want — there are a lot of components. Unless you are extremely familiar with all the components that make up an Oracle database installation, it's usually best to start with a typical installation. Chances are good that it will be sufficient, and you can add components later if it isn't. A typical installation will provide you with the following major items:

- ✦ The Oracle8i Server software
- ✦ Net8 Listener
- ✦ The Oracle8i utilities
- ✦ SQL*Plus
- ✦ A starter database

Reading the Release Notes

Be sure to read the release notes for any Oracle product that you install. They typically come as a stapled-together sheaf of papers. It's easy to shove them aside and lose them because they don't sit on a shelf as well as a manual. The release notes document issues and potential problems that weren't known when the manuals were printed. The issues described in the release notes may not apply to your situation, but it's possible that you could spend hours trying to make something work only to finally find the solution in the release notes.

Preparing the Server(s)

The last step before installing the software is to prepare the server. The installation guide tells you how to do this, which is another reason why you should read it. Windows NT environments are the easiest to prepare. Under Windows NT, you generally need to do the following:

- ✦ Check that you have enough system memory.
- ✦ Check that you have enough disk space.
- ✦ Make sure that the correct service pack is installed.

Configuring a UNIX environment is usually more complicated, and you will find yourself faced with a list of tasks that looks like this:

- ✦ Verify that you have enough system memory.
- ✦ Verify that you have enough swap space.
- ✦ Make sure that you have enough disk space.

- ✦ Verify that the correct operating system version is installed.
- ✦ Verify that the necessary operating system patches have been installed.
- ✦ Create mount points for the Oracle software and for database files.
- ✦ Create a dba group.
- ✦ Create a user to own the Oracle software.
- ✦ Modify various kernel parameters.

Preparing a UNIX environment involves more time and effort than preparing an NT environment. It's important to take the time to work through all the preparation steps identified in your installation guide and to work with your system administrator to ensure that your system is properly prepared for Oracle.

Installing the software

If you've done the neccessary preperation ahead of time, installing the software is a non-event. You will know what you want installed. You will know how to answer the installer prompts, and you will spend most of your time waiting and watching as the files are copied from the distribution CD to your server's disk.

Postinstallation tasks

Your installation guide contains a list of postinstallation tasks that you may need to perform. These tasks include reserving a port for the Net8 listener, starting the Net8 listener, editing configuration files for the precompilers, and so forth. Make sure that you perform the appropriate tasks for the specific product mix that you installed.

In addition to the postinstallation tasks identified in the installation guide, take the time to change the passwords for the `SYS` and `SYSTEM` users in your new database to something other than the default. You might also consider whether you really want the `SCOTT` user, with its set of example tables, to exist in your database. You should also change the internal password if you accepted the default during the installation.

Examining the Starter Database Contents

When you install Oracle from scratch and you accept the default installation options, the installer will create a starter database as part of the installation process. This starter database will include the following:

- ✦ Data dictionary tables and views
- ✦ A number of PL/SQL built-in packages

- ✦ Several default users
- ✦ A number of predefined roles

The *data dictionary* is a set of tables that Oracle uses to keep track of everything else in the database. Oracle creates a number of views on these data dictionary tables to make the information more accessible to database users. Throughout this book, you will find information about how you can use these data dictionary views to learn about the various objects contained in a database.

To support developers and DBAs who use PL/SQL, Oracle supplies a rich library of predefined functions and procedures. These are organized into packages, where each package focuses on a particular type of functionality. The `DBMS_OUTPUT` package, for example, contains procedures used to display information in SQL*Plus.

In addition to the data dictionary and the built-in packages, which are critical items that you don't want to alter, the Oracle8i starter database contains a number of predefined users and roles, some of which you may not want to keep. These users and roles are described in the next two sections, followed by a discussion of the sample tables that you can add especially for use with this book.

Users

Table 1-1 contains a list of the different users that you will find in an Oracle8i starter database. You can remove some of these users if you don't want them, but some users you should *not* remove. Users that you should never attempt to drop are marked with a "YES" in the Critical column.

Table 1-1
Preinstalled Users in Oracle8i

User	*Password*	*Critical*	*Usage*
`INTERNAL`	`ORACLE`	YES	Alias for `SYS`. Not a true user.
`SYS`	`CHANGE_ON_INSTALL`	YES	The `SYS` user owns the data dictionary tables and views, as well as the built-in PL/SQL packages.

Continued

Table 1-1 *(continued)*

User	*Password*	*Critical*	*Usage*
SYSTEM	MANAGER	YES	The SYSTEM user is created for the DBA to use. The SYSTEM user also owns a few tables and views, such as those used for advanced queuing, that are critical to the operation of the database.
SCOTT	TIGER	NO	Owns a small test schema that Oracle documentation references frequently.
DEMO	DEMO	NO	Owns a small test schema that Oracle documentation references frequently.
DBSNMP	DBSNMP	NO	Oracle Enterprise Manager Administrator.
OUTLN	OUTLN	YES	Owns stored outlines such as those created using the CREATE OUTLINE command.
MTSSYS	MTSSYS	NO	Used by Oracle Services for Microsoft Transaction Server.
AURORAORBUNAUTHENTICATED	N/A	YES	Used to allow Common Object Request Broker Architecture (CORBA) connections to log on to Oracle.

User	*Password*	*Critical*	*Usage*
ORDPLUGINS	ORDPLUGINS	NO	Used by Oracle interMedia
CTXSYS	CTXSYS	NO	Used by Oracle interMedia.
ORDSYS	ORDSYS	NO	Used by Oracle interMedia.
MDSYS	MDSYS	NO	Used by Oracle interMedia.

Except for SCOTT and DEMO, you should change the passwords for all these users soon after installation to protect the security of your database. Consider dropping the SCOTT and DEMO users. Their passwords are widely known, and even though they are nonprivileged users, they still pose a threat to security. Anyone logging on as SCOTT, for example, will be able to see information in any tables and views on which SELECT access has been granted to PUBLIC.

Roles

Roles help define database security. They give you a convenient way of grouping related privileges together so that you can easily assign them to a user. The starter database contains a number of predefined roles to help you get started. These are shown in Table 1-2.

Table 1-2
Preinstalled Roles in Oracle8i

Role	*Usage*
AQ_ADMINISTRATOR_ROLE	Allows a user to function as an advanced queuing administrator
AQ_USER_ROLE	Allows a user to make use of Oracle8i's advanced queuing features
CONNECT	Gives a user enough privileges to log on to the database and to create objects such as tables, views, synonyms, sequences, and database links

Continued

Table 1-2 *(continued)*

Role	*Usage*
RESOURCE	Gives a user enough additional privileges to create stored procedures, triggers, object types, operators, and index types and does not confer the privileges necessary to actually connect to the database
DBA	Confers all system privileges, allowing the user to do pretty much anything
EXECUTE_CATALOG_ROLE	Allows you to execute most of the built-in PL/SQL packages
HS_ADMIN_ROLE	Allows a user to use the DBMS_HS package to administer heterogeneous services
JAVADEBUGPRIV	Allows a user to invoke the Java debug agent
JAVAIDPRIV	Allows a user to change his or her dynamic identity. Only the SYS user should ever be granted this role
JAVASYSPRIV	Allows a user access to privileged Java functionality
JAVAUSERPRIV	Allows a user who is using Java to access files and sockets
CTXAPP	Identifies a user as a Context Cartridge application user
DELETE_CATALOG_ROLE	Allows a user to delete audit trail records
EXP_FULL_DATABASE	Allows a user to export an entire database
IMP_FULL_DATABASE	Allows a user to import an entire database
RECOVERY_CATALOG_OWNER	Allows a user to run a recovery
SELECT_CATALOG_ROLE	Provides access to the data dictionary views
SNMPAGENT	Provides necessary access to Oracle Enterprise Manager's Intelligent Agent
TIMESERIES_DEVELOPER	Allows a user to make use of the time series cartridge
TIMESERIES_DBA	Allows a user to function as a time series administrator

Tables

The user named SCOTT owns several sample tables that Oracle refers to throughout its manual set. Regardless of whether you decide to keep these tables in *your* database, you should familiarize yourself with them so that you will understand the

references to them. Figure 1-1 shows an entity-relationship diagram depicting these tables and their relationships.

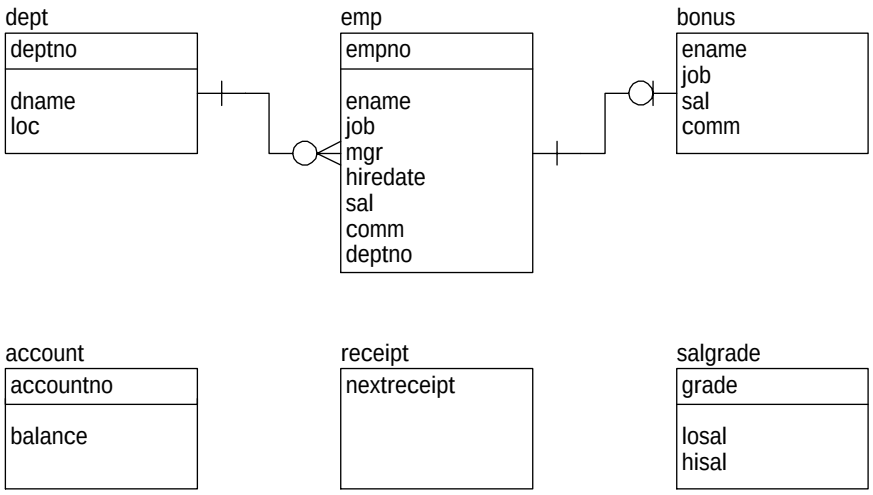


Figure 1-1: Tables in the SCOTT schema

In addition to being used for examples in the manual, the `SCOTT` tables provide a convenient set of data to use when you want to experiment with SQL.

Oracle8i Bible's sample tables

This book is filled with examples of code that provide step-by-step instructions on how to perform many tasks in the Oracle8i database. After installing your Oracle8i database, you can install all of the sample tables used here into your database.

For your convenience, the SQL scripts to create the Oracle8i Bible's sample tables are included (plus instructions for installing the tables into your database) on the CD-ROM that accompanies the book. Once you create these tables in your database, you can run most of the example code exactly as you see it in the book. Figure 1-2 shows a detailed relational database diagram of all of the sample tables on the CD-ROM.

Appendix D contains both the installation instructions and a detailed listing of the tables' contents.

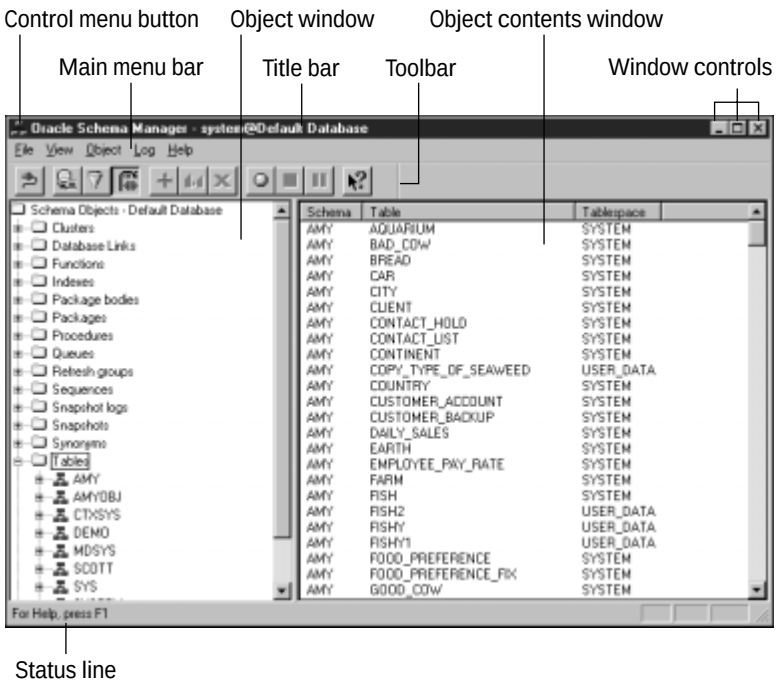


Figure 1-2: The Oracle8i Bible sample table contains many sets of relationships.

Installing WebDB

WebDB is an Oracle add-on product that allows you to develop Web sites that serve up content based on the data within your database. For example, you can easily design reports using WebDB, and then anyone using a standard Web browser can run them. WebDB is very easy to use. Besides developing reports, you can also use WebDB to design data entry forms and query forms, and you can create menus to tie together a set of forms and reports into an application. All work with WebDB is done via a standard Web browser, and any reasonably savvy user will be able to use WebDB to develop forms and reports.

Oracle includes WebDB software with an Oracle database license. It comes on a separate CD, and the WebDB installation is separate from the Oracle database installation. You don't have to install WebDB, but you should at least consider doing so. To help motivate you, WebDB includes some nifty DBA utilities that you can access through a Web browser. You'll get a glimpse of these later in this chapter, in the section titled "Taking a Quick Tour of Oracle8i."

WebDB is covered again in Chapter 26.

Preinstallation preparation

As with the database, the most important preinstallation preparation task is to read the manual. In this case, you want to read the *Getting Started — Installation and Tutorial* manual. You don't need to read the entire manual prior to installing WebDB, just be sure to read through Part 1, which talks about installing and configuring the software. The preinstallation tasks for WebDB include the following:

- ✦ Checking prerequisites
- ✦ Deciding which listener to use
- ✦ Deciding where to install the listener
- ✦ Deciding on a listener port
- ✦ Deciding between an automated installation and a manual installation

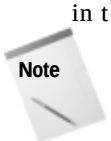
The next few sections briefly discuss each of these items.

Checking Prerequisites

WebDB requires that you run Oracle Server 7.3 or higher. This is actually fairly generous. We know of no sites running releases older than 7.3 and very few that haven't begun to migrate to either 8.0 or 8.1.

The WebDB installation process uses the SQL*Plus and SQL*Loader utilities. Make sure that these are installed on your server. Chances are that they will be, but occasionally you might run across a site where, for whatever reason, the DBA has chosen not to install these utilities on the server.

WebDB also requires that you set the `MAX_ENABLED_ROLES` parameter to a minimum value of 25. You set this parameter in your database initialization file. Remember that you need to stop and restart your database after changing a setting in the initialization file.



You will likely need to change the `MAX_ENABLED_ROLES` parameter. The default value has never been so high, and even when we installed Oracle8i, we found that the value for `MAX_ENABLED_ROLES` was less than 25.

Deciding Which Listener to Use

WebDB requires an HTTP listener to monitor the network for incoming requests to WebDB and to serve up Web pages in response. You have three choices here:

- ✦ Use the listener that ships with WebDB.
- ✦ Use the listener that comes with Oracle Application Server.
- ✦ Use some other listener.

The easiest solution is to use the listener that ships with WebDB. WebDB Listener is a lightweight listener that provides only the necessary functionality to serve up HTML pages for WebDB applications.

Deciding Where to Install WebDB Listener

You don't need to install WebDB Listener on the same physical machine as the database server. Certainly it would be easier to put everything on one machine, but for performance reasons, you may not want to do that. Another reason to put WebDB Listener on a separate machine is that it currently runs only under Windows NT and Solaris. If your database server is on an HP-UX server, for example, then you will need to install WebDB Listener on some other machine.

Deciding on a WebDB Listener Port

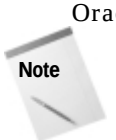
By default, WebDB Listener listens on port 80. Port 80 has become the standard port that Web servers use, and all browsers connect to that port by default. If you happen to have any other Web server software running on the same machine on which you are installing WebDB Listener, you will likely have a port conflict. In that case, you must either use some other port for WebDB or remove the conflicting software.

Deciding between an Automated Installation and a Manual Installation

WebDB is such a new product that Oracle hasn't yet worked all of the bugs out of the installer. Therefore, Oracle includes instructions for doing a manual installation on the WebDB distribution CD. You can view the manual installation instructions with any Web browser, and you'll find the file at the following location on the CD:

```
\support\maninst.htm
```

Avoid performing a manual installation if possible. It's a lot of work that requires a lot of experience. You should perform a manual installation only if you haven't been able to get the automated installation to work, and even then you should consult Oracle support first.



When recently trying to install WebDB on a Windows NT server that was running both Oracle 7.3.4 and 8.1.5, the WebDB installer didn't handle this well at all. The rules for placing the WebDB Oracle home are different for those two releases of the server software, and the installer tried to follow both sets of conflicting rules at the same time. This resulted in a catch-22 situation, which forced a manual installation.

If you do decide to perform a manual installation, you should review each of the SQL scripts prior to executing them. The scripts assume that you have created certain users and tablespaces. They also assume that the target database is your default database. If you're not an experienced DBA, you probably shouldn't attempt a manual installation without someone's help.

Installing the software

When you begin the actual installation process, keep two points in mind. First, WebDB expects the target database to be your default database. If you examine the SQL scripts that the installer runs, you will find that they all connect like this:

```
CONNECT sys/password
```

Notice that no service name has been specified on this `CONNECT` command. If you can't connect to your target database using SQL*Plus without specifying a Net8 service name, then the WebDB installer won't be able to either.

The second point to keep in mind is that if you are installing WebDB Listener, you should run the installation from the machine on which you want it to reside. The installer can run the necessary SQL scripts against a remote database, but it can't install WebDB Listener on a remote server.

The WebDB installation should be very easy as long as you've done the necessary preparation. Simply run the installer, answer the questions, and then sit back and watch while the installer does all the work.

Postinstallation tasks

The major postinstallation task for WebDB is to set up the connectivity between WebDB Listener and your database. This involves adding an entry to the tnsnames.or a file for the WebDB Oracle home. By default, WebDB expects this entry to be named WEBDB. If you use a different service name, you will have to tell that to WebDB Listener. The *Oracle WebDB Getting Started — Installation and Tutorial* manual explains how to do that.

Installing Oracle Enterprise Manager

Enterprise Manager is a set of GUI-based tools used to manage Oracle databases. Not surprisingly, Oracle has designed Enterprise Manager to fill the needs of large companies that have a lot of distributed Oracle databases. Enterprise Manager is more than just a set of GUI tools. A full-blown Enterprise Manager installation provides you with the following benefits:

- A set of GUI tools for managing the database
- A central console, from which you can monitor and manage any number of remote databases
- An Enterprise Manager server that allows you to schedule jobs to execute on remote servers

- ✦ A robust event-handling model that can alert you to impending database problems

Enterprise Manager, in a full-blown configuration, is a three-tier application. It consists of client software that sits on your PC. That client software talks to an Enterprise Manager server sitting somewhere on the network, and that server talks to the various database servers being managed. In addition, the special database known as the *repository database* contains information about database administrators, databases being monitored, jobs that are scheduled to run, and events that should be monitored. Figure 1-3 shows how all these pieces relate.

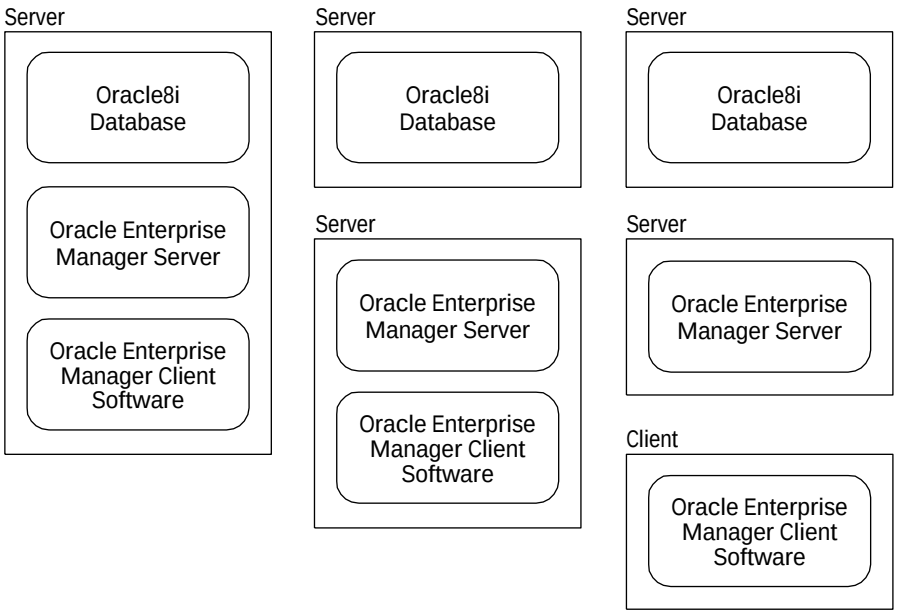


Figure 1-3: Enterprise Manager architecture

Managing Enterprise Manager in a configuration like the one shown in Figure 1-3 is almost as complex a task as managing an Oracle database. In fact, it gives you one extra database to manage — the repository. Fortunately, for small sites with only a few databases, Oracle lets you run the Enterprise Manager applications in a two-tier mode. Everything in Figure 1-3 disappears, except for the Enterprise Manager applications on the client and the databases being managed. This gives you the benefit of an easy-to-use GUI interface, without the complexity of creating a repository and configuring an Enterprise Manager server.

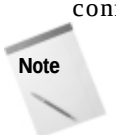
Preinstallation preparation

As with the other products, the fundamental piece of preinstallation advice is to read the manual. For Enterprise Manager, you should read through the *Oracle Enterprise Manager Configuration Guide*. This guide explains your options for installing the product and explains the hardware and software requirements for each option.

The major decision you have to make is whether to go with a two-tier or a three-tier installation. A two-tier installation involves installing the Enterprise Manager client software only on your PC, while a three-tier installation involves configuring an Enterprise Manager server and creating a repository database.

If you are going to perform a three-tier installation, your next task is to decide how many machines to use. It is possible to have all three tiers — the database being managed, the Enterprise Manager server, and the Enterprise Manager client software — on the same machine. A more distributed approach, and one that would probably perform better, is to have all three tiers on separate hardware platforms. Each DBA would get his or her own copy of the client software, the Enterprise Manager server and the repository database would reside on a server by themselves, and all the managed databases would reside on various other servers on the network.

Finally, it's best if you create the repository database prior to installing and configuring the Enterprise Manager server.



The repository database requires that you set the *processes* initialization parameter to a minimum of 200. You will get a warning during the repository creation process if you don't do this.

Installing the Enterprise Manager software

Installing the Enterprise Manager software is no harder than installing any other Oracle software. If you've done your planning up front, you need only respond to the installer prompts, sit back, and have a cup of coffee while the files are copied from the CD.

The two-tier installation is the simpler of the two possibilities. You simply run the setup program on the Enterprise Manager CD, select the option to install only the client software, and wait while the files are copied. It's that simple.

The three-tier installation is only a bit more complex. You have to install the Enterprise Manager server software on a server machine, and you need to install the client software on one or more clients. You end up performing at least two installations. Towards the end of the server installation, the installer will fire up the Enterprise Manager Configuration Assistant. This program will query you for the

name of your repository database. It will also query you for the username and password of a user with DBA privileges so that it can create the repository user, tables, views, and so forth. After you've provided the correct information, the configuration assistant connects to the repository database and creates the repository.

Postinstallation tasks

The two major postinstallation tasks are to configure Net8 for Enterprise Manager and to start the Enterprise Manager server. You need to start the server only if you installed the server, so if you just installed the client software, you have only one consideration — Net8.

Like WebDB, Enterprise Manager has it's own Oracle home directory. The setup program will have asked you to specify this directory, so you should know the correct path. This Oracle home directory has its own listener.ora file. To enable the Enterprise Manager software to connect to the databases that you want to manage, you must add entries for those databases to this file.

Finally, if you installed the Enterprise Manager server, start the server process. If the server is running on a UNIX machine, the `oemctrl` command is used to start it. If the server is running Windows NT, you will probably find it easier to start the Enterprise Manager server using the Services control panel, although the `oemctrl` command also works on NT.

Taking a Quick Tour of Oracle8i

Let's take a quick tour of Oracle8i and try out some of the tools that you'll be using to manage your Oracle database. First, you'll start by trying out two command-line utilities: SQL*Plus and Listener Control. You'll do enough to verify that your database is up and running, and you'll briefly glimpse how these tools work. Next, you'll move through several Enterprise Manager modules. Finally, the tour will end with a quick stop at the WebDB administration utility.

You may not have all of these tools installed. Oracle Enterprise Manager and WebDB are optional. If you didn't install them, you won't be able to follow along at the keyboard while reading through those sections of the tour. Read the tour anyway, though, just to see how all these tools work together.

SQL*Plus

SQL*Plus is the most venerable of the Oracle utilities. It's been around practically forever. Before Enterprise Manager came on the scene, SQL*Plus was the primary tool used to manage Oracle databases. In a nutshell, SQL*Plus is a utility that lets you enter and execute SQL statements and PL/SQL blocks. You can enter a `SELECT` statement, for example, and have the results displayed on the screen for you to see.

Oracle supports both a GUI and a command-line version of SQL*Plus. This tour will focus on the command-line version. Follow these steps to look at SQL*Plus:

1. Get to a command prompt.

   If you're running Windows NT, you'll need to open a Command Prompt window. If you're running under UNIX, log on to the operating system to get a command prompt.

If you're running Windows NT, also make sure that the correct Oracle home is in place. Installing WebDB or OEM might change your default Oracle home tree, which may make it impossible to run SQL*PLUS from the command prompt window.

2. Issue the `sqlplus` command to start SQL*Plus.

   You should see results similar to those shown in Figure 1-4.

Prior to the release of Oracle8i, the command to start SQL*Plus on Windows-based systems was `plusxxw`, where "xx" was the release number and the presence or absence of the "w" controlled whether you got the GUI version or the command-line version. The command `plus73` would start the command-line version of SQL*Plus for Oracle 7.3 under Windows NT (or 95), while `plus80w` would start the Windows version of SQL*Plus for Oracle 8.0 under Windows NT (or 95).

3. Connect to your Oracle database.

   Enter a username and password when prompted. For this example, you could log on as the `SCOTT` user, with a password of `TIGER`. Your screen should now look like the one shown in Figure 1-4.

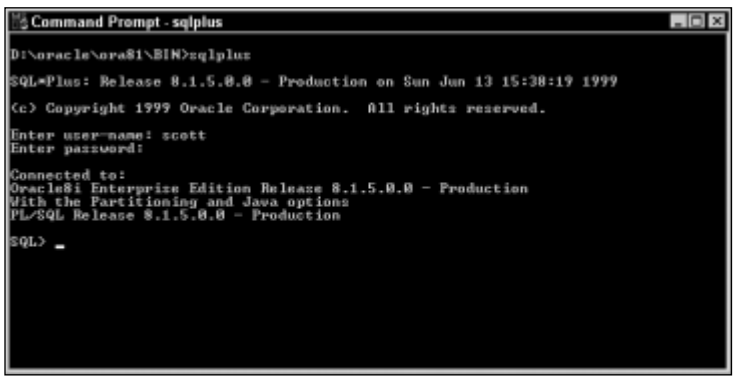


Figure 1-4: Connecting to an Oracle database

If you can connect to SQL*Plus, that's a good indication that the database is up and running.

4. Execute a SQL `SELECT` statement by typing the following SQL statement at the `SQL>` prompt:

   ```
   SELECT * FROM dual;
   ```

   You should get results like those shown in Figure 1-5. Don't forget to type the semicolon at the end of the statement. The semicolon is important because it marks the end of the statement as far as SQL*Plus is concerned. Omit the semicolon, and you'll be prompted for another line of SQL.

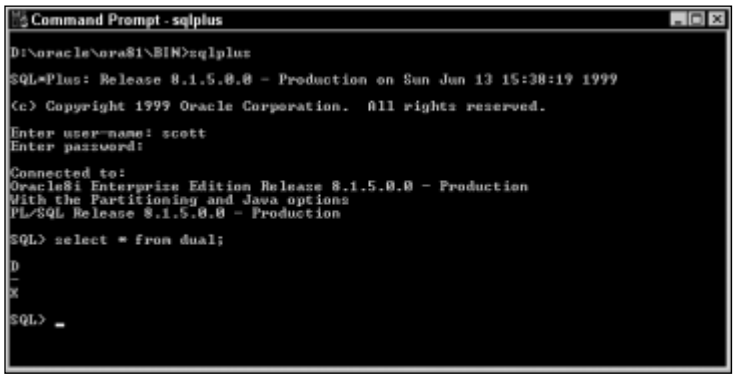


Figure 1-5: Selecting from the dual table

The dual table should be present in every Oracle database. It's a special table that has only one row and one column. It serves a variety of useful purposes. In this case, it's a known table that you can select from to prove that the database is working.

5. Exit SQL*Plus.

   We're done with SQL*Plus for now, so go ahead and exit the program and return to the command prompt. Figure 1-6 illustrates this.

Figure 1-6: Exiting from SQL*Plus

SQL*Plus can execute any valid SQL statement or PL/SQL block, and thus allows you to perform practically any database administration task. In the past, you had to use a separate program called Server Manager to start and stop an Oracle database and to run certain administrative scripts. Beginning with the release of Oracle8i, all the Server Manager functions have been merged into SQL*Plus. Server Manager is still around, but Oracle plans to desupport it someday, leaving SQL*Plus as the only command-line interface to an Oracle database.

Listener Control

Oracle's networking product is known as Net8. Net8 allows communications between two database instances over a network. Net8 also allows for communication between a database instance and a client PC on a network. On the server, Net8 implements a process known as a *listener*. It's the listener's job to monitor the network for incomming database connections. To control the listener, you use a utility known as Listener Control. Follow these steps to have a look at the Listener Control program:

1. Get to a command prompt.

   If you're running Windows NT, you'll need to open a Command Prompt window. If you're running under UNIX, log on to the operating system to get a command prompt.

2. Start the Listener Control program.

   The command to start the Listener Control utility is `lsnrctl`. Remember, if you are running Windows NT, you'll need to open a Command Prompt window first. Listing 1-1 provides an example of what you will see after entering the command.

3. Check the status of the listener by entering the `status` command. The results will appear as shown in Listing 1-2.

4. Type `exit` at the command prompt to exit the Listener Control utility.

Listing 1-1: **Entering the lsnrctl command**

```
c:\> lsnrctl

LSNRCTL for 32-bit Windows: Version 8.1.5.0.0 - Production on 13-JUN-99 15:41:28


(c) Copyright 1998 Oracle Corporation.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL>
```

Listing 1-2: **Checking the status of the listener**

```
LSNRCTL> status
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC0)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for 32-bit Windows: Version 8.1.5.0.0
Start Date                13-JUN-99 15:43:58
Uptime                    0 days 0 hr. 0 min. 7 sec
Trace Level               off
Security                  OFF
SNMP                      OFF
Listener Parameter File   d:\oracle\ora81\network\admin\listener.ora
Listener Log File         d:\oracle\ora81\network\log\listener.log
Services Summary...
  PLSExtProc            has 1 service handler(s)
  JONATHAN              has 1 service handler(s)
The command completed successfully
LSNRCTL>
```

Instance Manager

Instance Manager is an Oracle Enterprise Manager application that shows you a high-level view of your database's current activity. Instance Manager also allows you to start and stop your database instance. This section assumes you have access to a DBA user account and password. You can use the standard Oracle8i user, SYSTEM, if you wish. The default password for SYSTEM is MANAGER.

Follow these steps to look at Instance Manager:

1. Click Start, point to Programs, point to Oracle – oem_home, point to DBA Management Pack, and select Instance Manager.

   An Oracle Enterprise Manager Login window appears and asks for your input.

2. To log on as the SYSTEM user, type SYSTEM in the Username box, MANAGER (or whatever your current system password is) in the Password box, and your Net8 service name into the Service box.

   This sequence brings you to Instance Manager's main window. It will be similar to what you see in Figure 1-7, but instead of a traffic light, you will see a splash screen on the righthand side of the window.

3. Click the Database icon that is on the left side of the screen to see the current status of the database.

   You should now see a traffic light in the righthand window, as shown in Figure 1-7.

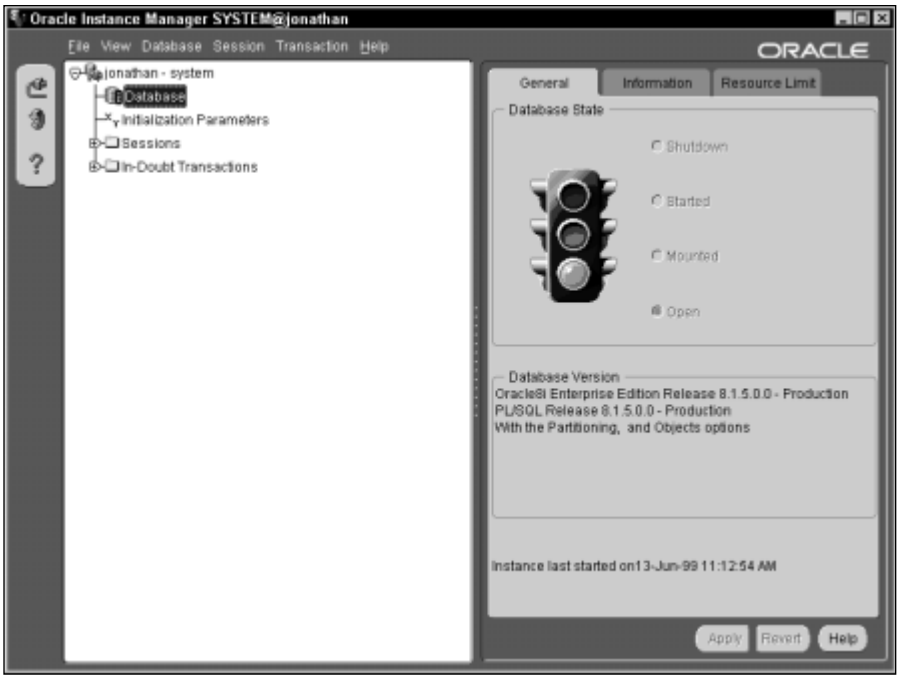


Figure 1-7: Instance Manager shows the status and current activity of the database.

4. Observe the traffic light.

If the light is green, as it likely will be, your database is open and running.

5. Double-click the Sessions folder in the left frame.

You should see a list of sessions currently active in your database. The first several sessions, the ones without usernames next to them, are the background processes for the instance. The remaining sessions are for users who are logged on to the database.

6. Click any session.

Figure 1-8 shows the details you see for each session using the database. You see the user, terminal, and the program running for the session.

7. Select the Exit command on the File menu to exit Instance Manager.

8. Close the associated Command Prompt window (Windows 95 only).

All Enterprise Manager applications are Java applications. When you run one on a Windows 95 system, in addition to the GUI window that you just saw, an associated Command Prompt window sits in the background. Figure 1-9 shows what that window looks like. In Enterprise Manager 2.0, closing one of the applications doesn't close the associated Command Prompt window. You need to do that manually by double-clicking the window's Close button. This behavior is specific to Windows 95 and 98.

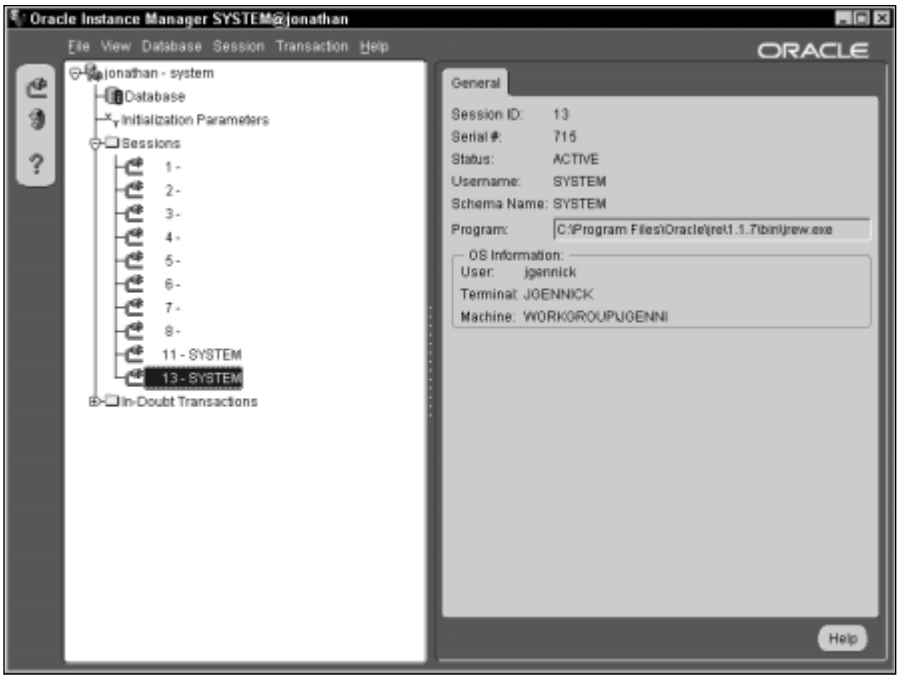


Figure 1-8: Observe your users with a click of a button.

Schema Manager

You use Schema Manager to create and modify schema objects such as tables, views, and stored procedures. Follow these steps to start Schema Manager:

1. Select Start, point to Programs, point to Oracle – oem_home, point to DBA Management Pack, and select Schema Manager to start Schema Manager.

   An Oracle Enterprise Manager Login window appears and asks for your input.

2. Log on as the `SYSTEM` user.

   You log on as the `SYSTEM` user the same way as you did previously when you ran Instance Manager. Once you log on, you will see the initial Schema Manager screen. Depending on the speed of your system, you may experience a short delay before a list of folders appears on the left side of the window.

3. Double-click the Tables folder in the left pane of the window.

   A list of tables appears in the detail pane. A list of schemas (table owners) appears under the Tables folder in the left pane of the window. Your screen should look similar to the one shown in Figure 1-9.

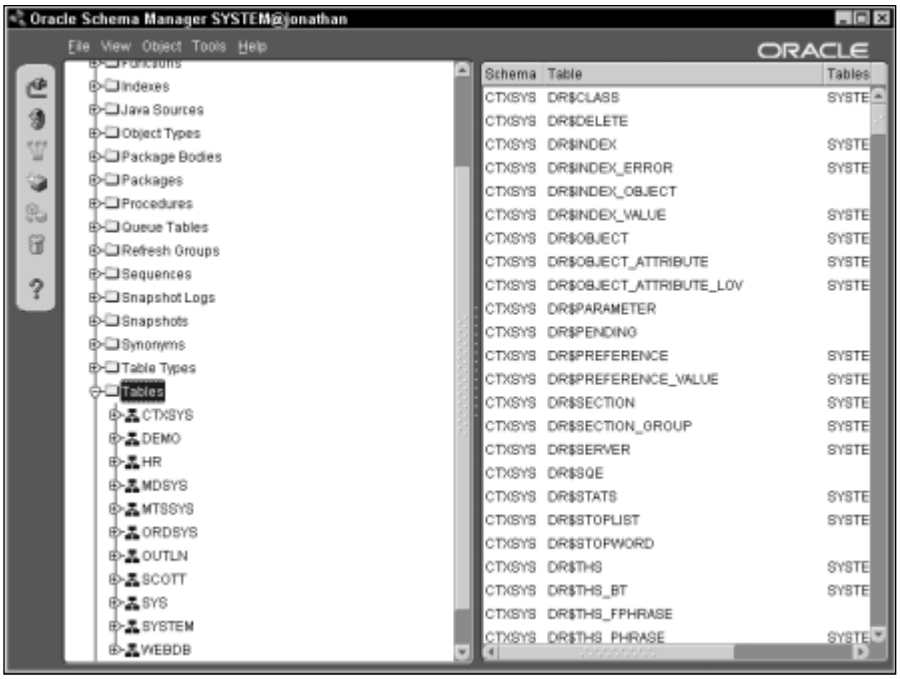


Figure 1-9: Schema Manager manages database objects such as tables.

Schema Manager arranges objects such as tables and indexes in an orderly format, similar to the way Windows NT Explorer arranges folders and files. Navigate within Schema Manager as you would in these utilities. Pull-down menus display your options. Clicking selects an item, while right-clicking displays a context-sensitive pop-up menu.

4. Double-click one of the schemas in the left pane.

   A list of tables owned by that user will appear in the right pane and will also appear underneath the schema's entry in the left pane.

5. In the left pane, click the first table in the list.

   The right pane changes to display a tabbed listing that describes the highlighted table (see Figure 1-10). The basic structure you see in the figure is common to most of the Enterprise Manager tools.

6. Select the Exit command on the File menu to exit Schema Manager.

7. Close the associated Command Prompt window (Windows 95 only).

   Schema Manager has the same type of Command Prompt window associated with it as Instance Manager does. Switch to this window — you can press Alt+Tab to do this — and double-click the Close button to close it.

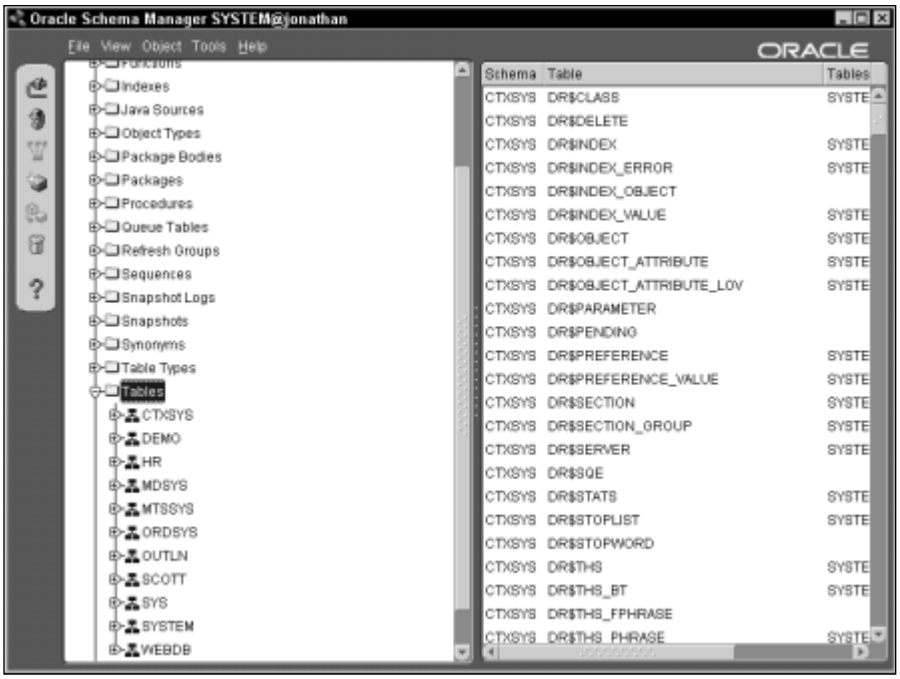


Figure 1-10: Schema Manager, showing the structure of a table

Storage Manager

Storage Manager monitors disk space and file size and also displays the mapping between tablespaces and actual files. Use Storage Manager to add new tablespaces and files to your database. Follow these steps to view Storage Manager:

1. Select Start, point to Programs, point to Oracle – oem_home, point to DBA Management Pack, and select Storage Manager to start Storage Manager.

   An Oracle Enterprise Manager Login window appears and asks for your input.

2. Log on as the `SYSTEM` user.

   Once you log on, you see the initial Storage Manager window.

3. Click the Tablespace folder in the left pane of the window.

   A list of tablespaces appears in the right pane. The list shows the status and size of each of your tablespaces. You may need to grab the vertical bar in the

center of the window and slide it to the left before you can see all the information in the right pane. Your screen should now look similar to the one shown in Figure 1-11.

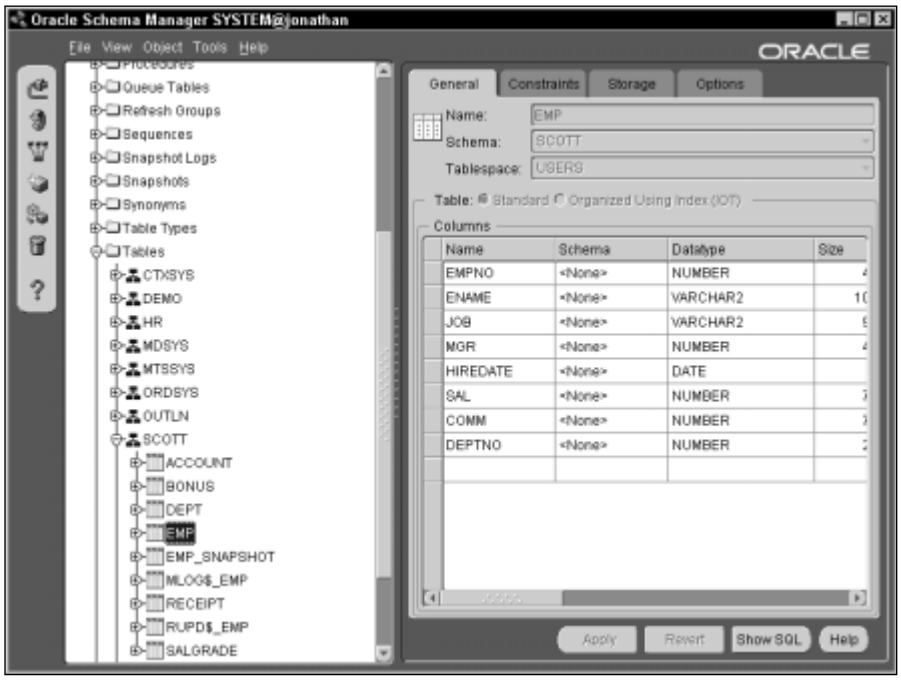


Figure 1-11: The Storage Manager window shows the status and size of each tablespace.

4. Click the plus sign next to the Datafiles folder in the left pane.

A list of files appears below the Datafiles folder in the left pane.

5. In the left pane, click the first datafile in the list.

The right pane changes to display a tabbed list describing the highlighted datafile, as shown in Figure 1-12.

6. Exit Storage Manager.

7. Close the associated Command Prompt window.

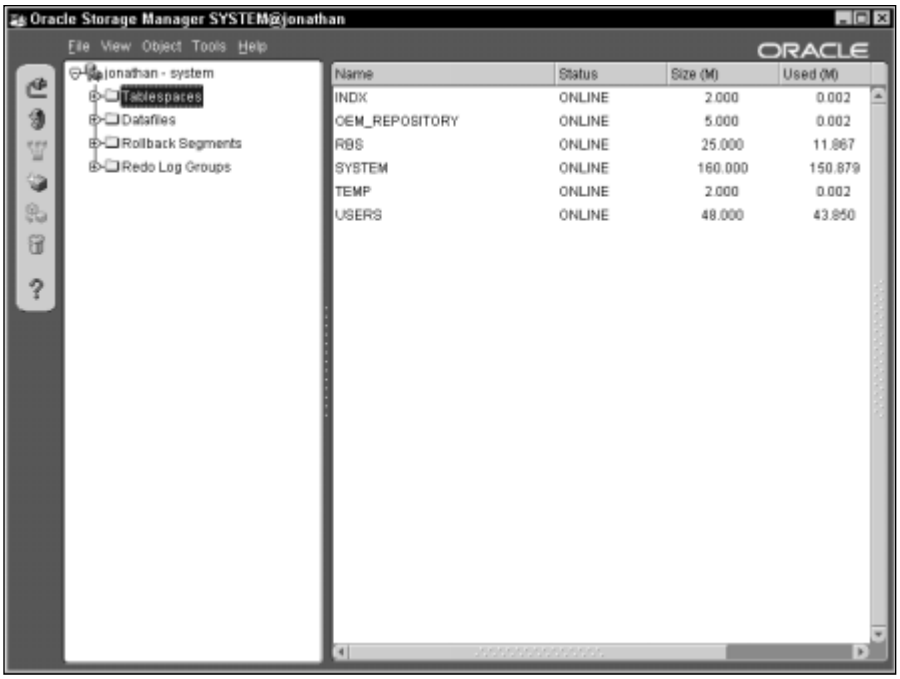


Figure 1-12: The Storage Manager window allows you to add to a datafile.

SQL*Plus Worksheet

SQL*Plus Worksheet is an Oracle Enterprise Manager tool that provides the same functionality as SQL*Plus, but it is wrapped in a GUI interface. In fact, SQL*Plus Worksheet actually calls SQL*Plus to execute whatever commands you enter. This is done to ensure absolute, 100-percent compatibility between the two products. To start SQL Worksheet, follow these steps:

1. From the Windows NT Start menu, point to Programs, point to Oracle – oem_home, point to DBA Management Pack, and select SQLP Worksheet to start the program.
2. Fill in a valid username, password, and service in the security screen that pops up.

   If you like, you can use Oracle8i's sample user account (`SCOTT`) and password (`TIGER`). The password shows up as asterisks (*) when you type it. Leave the Service box blank to connect to your local database. Otherwise, enter the Net8 service name for the Oracle database to which you will connect.

3. Accept the Connect As box as Normal. The SQL*Plus Worksheet will appear.

Figure 1-13 shows the SQLPlus Worksheet window that you can use to type your commands. The window divides horizontally into two panes. The top pane is the area in which you type SQL commands. The bottom pane displays the command and the results of the command.

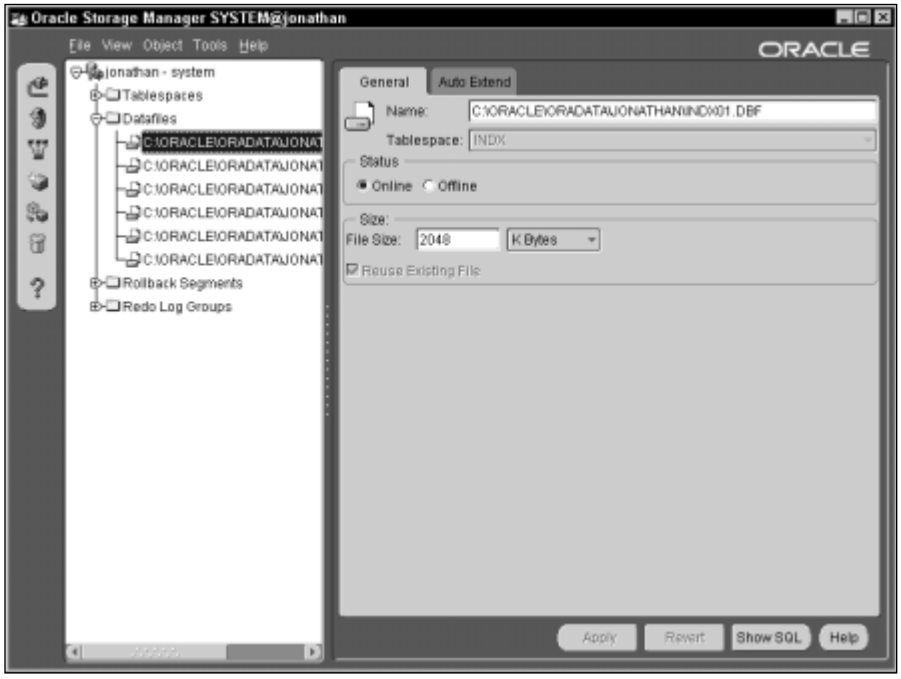


Figure 1-13: The SQL*Plus Worksheet window

4. Type the following SQL command in the top pane:

```
SELECT * FROM dual
```

5. Click the Execute button (this button has a lightning bolt on it).

You will see the results of the query appear in the bottom pane, as shown in Figure 1-14.

6. Close SQLPlus Worksheet.

7. Close the associated Command Prompt window.

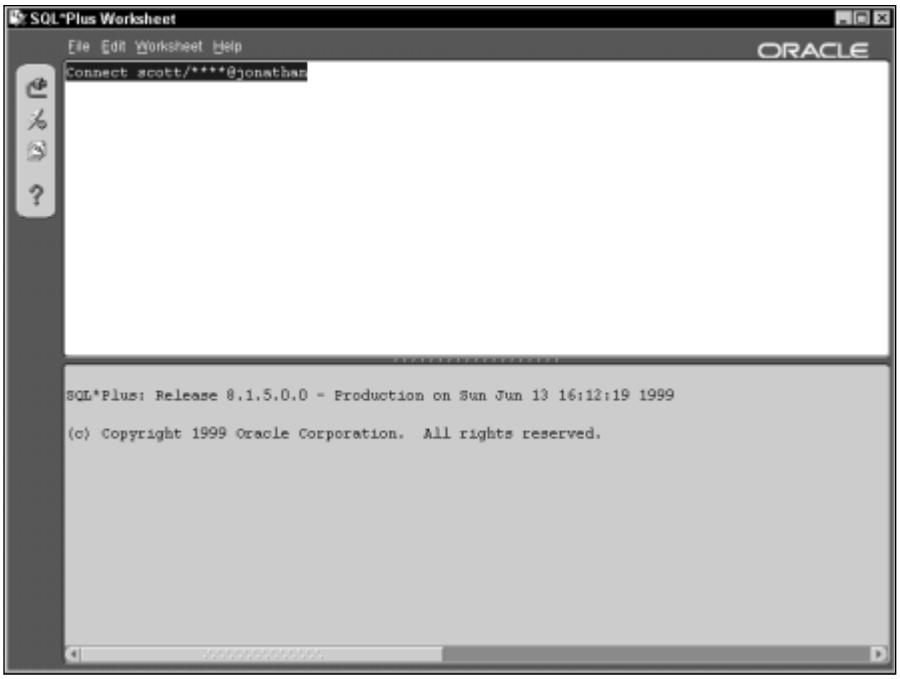


Figure 1-14: The SQLPlus Worksheet window showing the results of a query

WebDB

Do the following to start WebDB:

1. Start your Web browser.
2. In the URL box, type the name of the server where WebDB Listener is running. In the example shown in Figure 1-15, WebDB Listener is running on a node named dsat4, so "dsat4" was typed as the URL.
3. Enter your WebDB username and password, and click OK.

   A dialog box will prompt you for a username and a password. See Figure 1-15. The default WebDB username is webdb, and the default password is webdb as well.
4. Observe the WebDB opening page. It will appear similar to the one shown in Figure 1-16. From this page, you can browse the database, create new Web sites, monitor the database, and manage WebDB users.

Figure 1-15: The WebDB password prompt

Figure 1-16: The WebDB opening page

5. Check on your database's memory usage by clicking on the following links: Monitor, Database Objects, Sessions and Memory Structures, and Chart of SGA Consumption. You should see a chart showing how much memory is being used for various purposes. See Figure 1-17.

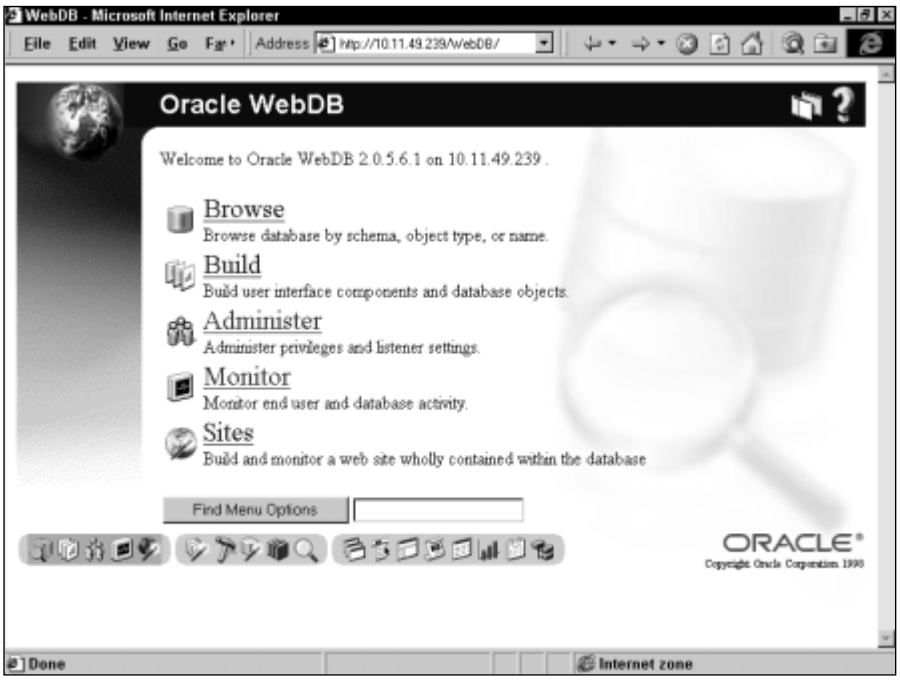


Figure 1-17: Chart of SGA consumption

6. Exit your Web browser.

Summary

In this chapter, you learned the following:

- Oracle8i includes a number of exiting new features, many of which are oriented to position Oracle8i as an Internet database. Leading the pack is the Java engine that allows you to run Java code within the database itself. WebDB is another Internet-related feature that takes an innovative approach to making Oracle data more accessible to end users.

- Oracle also added a number of features to the 8i release to make your job easier. One long-asked-for feature is the `DROP COLUMN` command that allows you to delete a column from a table without dropping and recreating that table. Oracle Enterprise Manager has been completely rewritten in Java, and you can implement it in a three-tier configuration with a Web-based interface.
- Before installing Oracle8i, you must be sure to spend the neccessary preliminary time in planning and preparation. Don't rush into it. UNIX systems, in particular, may require a number of steps to prepare the operating system and the environment for an Oracle installation.

✦ ✦ ✦