



# CHAPTER 1

## Database Fundamentals

### CRITICAL SKILLS

- 1.1 Define a Database
- 1.2 Learn the Oracle Database 11g Architecture
- 1.3 Learn the Basic Oracle Database 11g Data Types
- 1.4 Work with Tables
- 1.5 Work with Stored Programmed Objects
- 1.6 Become Familiar with Other Important Items in Oracle Database 11g
- 1.7 Work with Object and System Privileges
- 1.8 Introduce Yourself to the Grid
- 1.9 Tie It All Together



This chapter is the start of your Oracle Database 11g journey. The Oracle database is a complex product and you will need to learn the basics first. From this point forward, we will walk you through the skills that you'll need to begin working with Oracle Database 11g. We'll begin at the core of this product, with the fundamentals of a database. This chapter will also give you an understanding of the contents of your database and prepare you to move into the more complex areas of Oracle Database 11g technology.

### CRITICAL SKILL 1.1

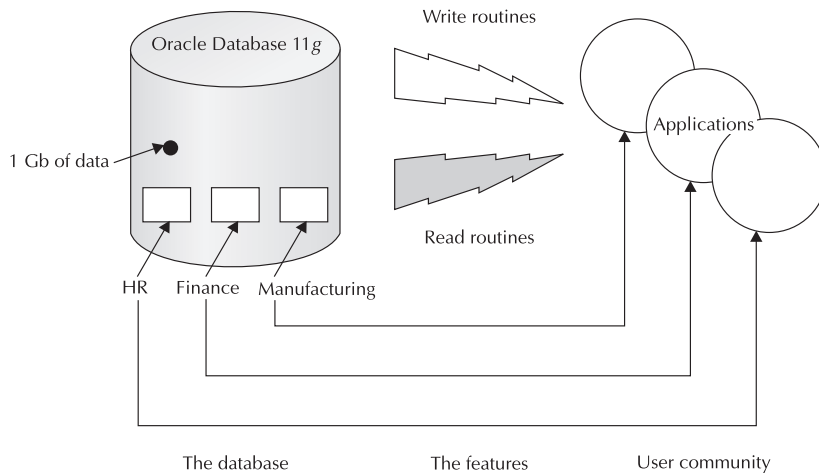
## Define a Database

Oracle Database 11g is the latest offering from Oracle. Perhaps you have heard a lot of hype about Oracle Database 11g, and perhaps not. Regardless of your experience, 11g is a rich, full-featured software intended to revolutionize the way many companies do their database business. Think of a database as the Fort Knox for your information. A database is an electronic collection of information designed to meet a handful of needs:

- 1. What is a database?** Databases provide one-stop shopping for all your data storage requirements, no matter whether the information has to do with human resources, finance, inventory, sales, or something else. The database can contain any amount of data, from very little to very big. Data volumes in excess of many hundreds of gigabytes are commonplace in this day and age, where a gigabyte is 1,073,741,824 bytes.
- 2. What must it be able to do?** Databases must provide mechanisms for retrieving data quickly as applications interact with their contents. It is one thing to store tax information for the 300 million citizens of a country, but it's another kettle of fish to retrieve that data, as required, in a short time period.
- 3. How is it suitable for corporate data?** Databases allow the sharing of corporate data such that personnel data is shared amongst one's payroll, benefits, and pension systems. A familiar adage in the database industry is "write once, read many." Databases are a manifestation of that saying—one's name, address, and other basic personnel information are stored in one place and read by as many systems requiring these details.

Figure 1-1 shows, in a nutshell, the components that come together to deliver the corporate database management solution affectionately called Oracle Database 11g.

There is a great deal of academic interest in the database industry, because the theory of the relational database is founded in relational algebra. As data is entered into and stored in Oracle Database 11g, the relationships it has to other data are



**FIGURE 1-1.** *The players in the Oracle Database 11g solution*

defined as well. This allows the assembling of required data as applications run. These relationships can be described in plain English for a fictitious computer parts store in the following example:

- Each geographical location that the store does business in is uniquely identified by a `quad_id`.
- Each manufacturer that supplies parts is uniquely identified by a ten-character `manufacturer_id`. When a new manufacturer is registered with the system, it is assigned a `quad_id` based on its location.
- Each item in the store's inventory is uniquely identified by a ten-character `part_id` and must be associated with a valid `manufacturer_id`.

Based on these three points, practitioners commonly develop statements similar to the following to describe the relationships between locations, manufacturers, and parts:

- **A one-to-many relationship** Locations and manufacturers—more than one manufacturer can reside in a specified location.
- **A many-to-many relationship** Manufacturers and computer parts—the store purchases many different parts from each manufacturer.

These two relationships are established as data is captured in the store's database and other relationships can be deduced as a result—for example, one can safely say

## 4 Oracle Database 11g: A Beginner's Guide

“parts are manufactured in one or more locations based on the fact that there are many manufacturers supplying many different products.” Oracle has always been a relational database product, commanding a significant percentage of market share compared to its major competition. Let's get started and look at the Oracle Database 11g architecture.

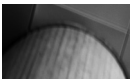
### CRITICAL SKILL 1.2

## Learn the Oracle Database 11g Architecture

As with many new software experiences, there is some jargon that we should get out of the way before starting this section.

- **Startup** This is the act of issuing the appropriate commands to make an Oracle Database 11g accessible to applications. After a startup activity completes, the database is referred to as *opened*. Once opened, the database moves to the next step where it is started. At this point, the database is ready to use.
- **Shutdown** This is the act of stopping Oracle Database 11g. When Oracle Database 11g is shut down, nobody can access the data in its files.
- **Instance** This is a set of processes that run in a computer's memory and provide access to the many files that come together to define themselves as Oracle Database 11g.
- **Background processes** These are processes that support access to an Oracle Database 11g that has been started, playing a vital role in Oracle's database implementation. Various background processes are spawned when the database is started and each performs a handful of tasks until a database is shut down.

Let's look at the assortment of files and background processes that support Oracle Database 11g.



### NOTE

*In order to work with the code snippets and the sample schemas we discuss throughout this book, you will need to have the Oracle Database 11g software installed and the first database successfully created. The Database Configuration Assistant (dbca) is the fastest way to set up your first database. Most of the time you simply accept the defaults suggested on the dbca screens. If you have any problems with either the software installation or the dbca, please either consult a more senior colleague or surf MetaLink (<http://metalink.oracle.com>) to get assistance (after supplying appropriate login credentials).*

## The Control Files

Oracle's *control files* are binary files containing information about the assortment of files that come together to support Oracle Database 11g. They contain information that describes the names, locations, and sizes of the database files. Oracle insists there is only one control file, but knowledgeable technicians have two or three and sometimes more. As Oracle Database 11g is started, the control files are read and the files described therein are opened to support the running database.

## The Online Redo Logs

As sessions interact with Oracle Database 11g, the details of their activities are recorded in the online redo logs. Redo logs may be thought of as transaction logs; these logs collect transactions. A *transaction* is a unit of work, passed to the database for processing. The following listing shows a few activities that can be referred to as two transactions:

```
-- Begin of transaction #1
create some new information
update some existing information
create some more new information
delete some information
save all the work that has been accomplished
-- End of transaction #1
-- Begin transaction #2
update some information
back out the update by not saving the changed data
-- End transaction #2
```

Oracle Database 11g insists that there are at least two online redo logs to support the instance. In fact, most databases have two or more redo log groups with each group having the same number of equally sized members.

## The System Tablespace

*Tablespace* is a fancy Oracle Database 11g name for a database file. Think of it as a *space* where a *table* resides. As an Oracle Database 11g is created, a system tablespace is built that contains Oracle's data dictionary. As Oracle Database 11g operates, it continually gets operational information out of its data dictionary. As records are created, this system tablespace defines attributes of the data it stores, such as

- **Data types** These are the characteristics of data stored in the database. Are they numeric, alphanumeric, or perhaps binary or some video or audio format?
- **Field size** This is the maximum allowable size for fields as they are populated by the applications. This is where, for example, a country description is defined as from 1 to 30 characters long, containing only letters.

## 6 Oracle Database 11g: A Beginner's Guide

- **Ownership** Who owns the information as the database data files are populated?
- **Viewing and manipulation rights** Who is allowed to look at the data and what are the types of activities that each database user can perform on that data?

The system tablespace is a very close cousin of the sysaux tablespace discussed next.

### The Sysaux Tablespace

Many of the tools and options that support the Oracle Database 11g activities store their objects in this sysaux tablespace. This is mandatory as a database is created. The Oracle Enterprise Manager (OEM) Grid Control repository used to go in its own oem\_repository tablespace, but with Oracle Database 11g (and its predecessors), its objects now reside in sysaux.

### Default Temporary Tablespace

As the dbca does its thing, a tablespace is created that serves as the default location for intermediary objects Oracle Database 11g builds as it processes SQL statements. *SQL* stands for structured query language, an industry standard in the database arena, which is used to retrieve, create, change, and update data. Most of the work Oracle does to assemble a result set for a query operation is done in memory. A *result set* is a collection of data that qualifies for inclusion in a query passed to Oracle. If the amount of memory allocated for query processing is insufficient to accommodate all the activities required to assemble data, Oracle uses this default temporary tablespace as its secondary work area for many activities, including sorting.

### Undo Tablespace

As sessions interact with Oracle Database 11g, they create, change, and delete data. *Undo* is the act of restoring data to a previous state. Suppose one's address is changed from 123 Any Street to 456 New Street via a screen in the personnel application. The user who is making the change has not yet saved the transaction. Until that transaction is saved (referred to as *committed* in the world of Oracle Database 11g) or abandoned (referred to as *rolled back* in the same world), Oracle maintains a copy of the changed data in its *undo tablespace*.

### The Server Parameter File

Oracle Database 11g sometimes calls the server parameter file its *spfile*. This is where its startup parameters are defined and the values in this file determine the

environment that database operates in. As one starts an Oracle instance, the spfile is read and various memory structures are allocated based on its contents.

## Background Processes

Essentially, background processes facilitate access to Oracle Database 11g and support the instance while it is running. These are the main background processes; many of their names haven't changed over the past few releases prior to Oracle Database 11g.

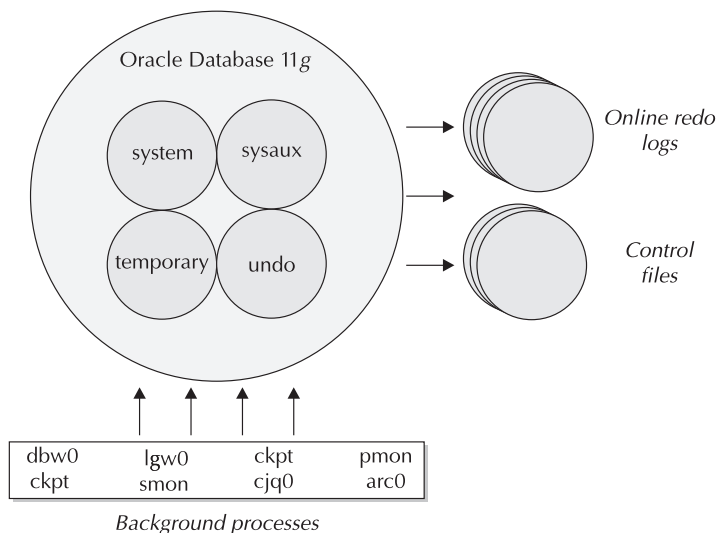
- **The database writer (dbw0) process** This process (named dbwr in earlier versions of Oracle Database) is responsible for writing the contents of database buffers to disk. As sessions interact with Oracle Database 11g, all the information they use passes through Oracle's database buffers, a segment of memory allocated for this activity.
- **The log writer (lgw0) process** This process (named lgwr in previous versions of Oracle Database) manages the writing of information to the online redo logs. A log buffer area is set aside in memory where information destined for the online redo logs is staged. The transfer of this information from memory to disk is handled by this process.
- **The checkpoint process (ckpt)** This is responsible for updating information in Oracle Database 11g's files during a checkpoint activity. A *checkpoint* is the activity of writing information from memory to the appropriate locations in Oracle Database 11g. Think of a checkpoint as a stake in the ground allowing the restoration of a system to a specific point in time. The checkpoint process may trigger lgw0 and dbw0 to do their specialized tasks.
- **The system monitor (smon) process** This is the gatekeeper of consistency as Oracle Database 11g runs. *Consistency* defines the interrelatedness of the database components with one another. A consistent instance must be established every time Oracle Database 11g starts, and it is smon's job to continually enforce and reestablish this consistency. Plainly put: an inconsistent database is trouble!
- **The process monitor (pmon)** This is responsible for cleaning up any resources that may have been tied up by aborted sessions interacting with the database. The famous CTRL-ALT-DEL that people tend to use to reboot a personal computer can leave resources tied up in Oracle Database 11g. It is pmon's job to free up these resources.
- **The job queue coordination (cjq0) process** This is responsible for spawning job processes from Oracle Database 11g's internal job queue. Oracle Database 11g does some self-management using its job queue, and

## 8 Oracle Database 11g: A Beginner's Guide

users of the database can create jobs and have them submitted to this cjq0 coordinator.

- **The archiver (arc0) process** This is responsible for copying online redo logs to a secondary storage location before they are reused by the next set of transactions. In the “Online Redo Logs” section of this chapter, we discuss how Oracle Database 11g insists there are at least two online redo logs. Suppose we call these groups A and B. Oracle Database 11g uses these two groups in a cyclical fashion, moving back and forth from A to B to A to B and so on. The arc0 process, when and if instructed, will make a copy of a file from log group A before allowing it to be reused.

Figure 1-2 illustrates the way the architecture components we have described come together to support Oracle Database 11g. Oracle Database 11g is opened and then started, and the control files are read to get its bearings. Then the online redo logs and the assortment of tablespaces listed in the control files are acquired. As the instance comes to life, the background processes take over and manage the operations of the database from there.



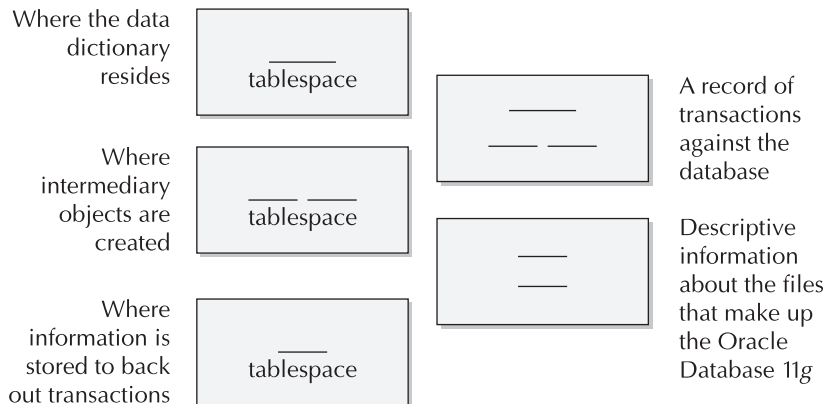
**FIGURE 1-2.** Tablespaces, support processes, and infrastructure files

## Project 1-1 Review the Oracle Database 11g Architecture

There are many types of files that come together to support Oracle Database 11g. In this section, we have discussed control files, online redo logs, the system tablespace, and an assortment of datafiles and tablespaces that support the database. As well, we have looked at the series of background processes that allow users to interact with Oracle Database 11g. In this brief project, you will apply what you have learned about the processes that support Oracle Database 11g. As you descend into the land of Oracle Database 11g, you'll find that this information is crucial to your understanding of this remarkable software solution.

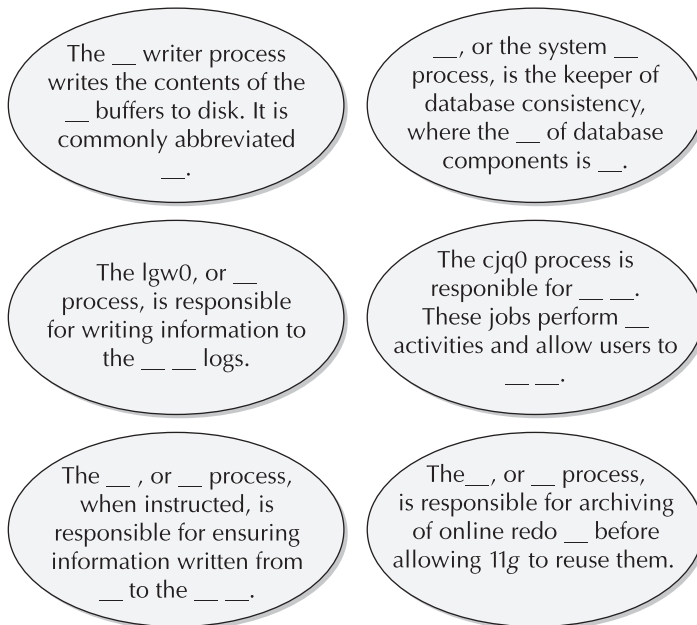
### Step by Step

- There are a few pieces missing in the following diagram of the infrastructure of files that support Oracle Database 11g. Fill in the missing text where required.



- The second diagram shows a partial makeup of the background processes with Oracle Database 11g. Complete the missing text where indicated by broken lines.

(continued)



## Project Summary

You don't need to master Oracle Database 11g architecture to become fluent with the software. Just as an electrician needs the assistance of a good set of blueprints, the Oracle Database 11g technical person should understand some of the inner workings of the software. A peek under the covers, as brief as it may have been in this section, is a good path to follow while becoming familiar with what Oracle Database 11g is all about.

Before moving on to discuss Oracle Database 11g data types, let's spend a minute looking at the database administrator, the ultimate director of the operations of the database.

## The Database Administrator

This privileged user of Oracle Database 11g is commonly the most experienced technician in the shop, with some exceptions. Often, recent adopters of the Oracle technology have little or no in-house experience, and one or more employees may find themselves targets of the familiar directive "So, you're the new Oracle Database 11g DBA!" One scrambles to find sources for technical knowledge when thrust into this role. What better place to be than reading *Oracle Database 11g: A Beginner's Guide*? The following list outlines common responsibilities of the Oracle Database 11g DBA:

- **Installation and configuration** The DBA must install and customize the Oracle Database 11g software and any assorted programs that will run alongside and access the database.

- **Create datafiles and tablespaces** The DBA decides which application the data will reside in.
- **Create and manage accounts** The DBA sets up the accounts with the usernames and passwords that interact with the database.
- **Tuning** The DBA tweaks the environment that Oracle Database 11g operates in by adjusting initialization parameters using the system parameter file.
- **Configure backups** Alongside recovery testing, the DBA performs this activity to ensure the usability and integrity of system backups.
- **Work with developers** This is an ongoing process for DBAs, to ensure that the code they write is optimal and that they use the server's resources as efficiently as possible.
- **Stay current** DBAs keep abreast of the emerging technology and are involved in scoping out future directions based on the enhancements delivered with new software releases.
- **Work with Oracle Support Services** DBAs initiate service requests (SRs) to engage support engineers in problem-solving endeavors. The front-end of the SR creation process is called MetaLink (described earlier in the chapter).
- **Maximize resource efficiency** The DBA must tune Oracle Database 11g so that applications can coexist with one another on the same server and share that machine's resources efficiently.
- **Liaise with the system administrators** DBAs must ensure that the appropriate disk space and processor power are available and properly utilized.

As with most lists, after reading the preceding bullet points, you may wonder what else DBAs do with their time. As you work with Oracle Database 11g, you'll experience other activities that will plug any loopholes that may exist in the previous list.

### CRITICAL SKILL 1.3

## Learn the Basic Oracle Database 11g Data Types

Very early in one's journey through the world of Oracle Database 11g, it becomes time to learn its common data types. Regardless of your past experience in information technology, data types are nothing new. Let's look at the most common type of data that can be stored in Oracle Database 11g; keep in mind that the list is much longer than the one presented here.

## varchar2

By far the most common data type, `varchar2` allows storage of just about any character that can be entered from a computer keyboard. In earlier software solutions, this was commonly referred to as *alphanumeric* data. The maximum length of `varchar2` is 4000 bytes or characters, and it possible to store numeric data in this data type. This is a variable length character string, with no storing of trailing insignificant white space:

```
create table ... (
  name          varchar2(30),
  city          varchar2(30),
  ...
  ...
  state         varchar2(2) );
```

If a program or SQL statement tries to store a set of characters in a `varchar2` field that is longer than the field's specification, an error is returned, and the statement stops running and returns control back to you.

## number

The number data type allows the storing of integer as well as integer/decimal digits. When non-integer data is stored, the total number of significant digits of the number is referred to as *precision*, while the portion to the right is called *scale* or *decimal places*. For example, the number 29.1963 has a precision of 6 and a scale of 4. The maximum precision is 38 and the maximum scale is 127. The confusing part of the specification of a number data type comes into play when storing non-integer information. Table 1-1 illustrates this concept.

When defining a number data type with decimal places, it's important to know that the maximum integer portion of the number data type is the difference between the two numbers specified. The specification (6,3) allows for two, not six, integer

---

Number Specification	Column Length (Precision)	Decimal Digits (Scale)
(3,2)	3	2
(6,3)	6	3
(17,12)	17	12

---

**TABLE 1-1.** *Number Data Type Specification*

digits. If more decimal digits are received than the column definition permits, it rounds the value before storage.

## date

The *date* data type stores time and date information, with the time component rounded to the nearest full second.

There are many, many functions available to be performed on date fields as they are extracted from an Oracle Database 11g.

When date columns are selected from Oracle Database 11g, it is common to perform a function on their values to make them more readable. By default, the time component of a date column is not displayed without manipulating its contents using a `to_char` function, described in Chapter 4. By default the general display format for a date is `DD-MON-YY` (day, month, and year). This format may be changed via the `NLS_DATE_FORMAT` parameter or by using a display format function.

## timestamp

The *timestamp* data type is a close relative of date. The major advantage is that the timestamp stores information about the second to a much higher accuracy. In this time when every subsecond counts, the timestamp can be a valuable asset. There is a time component in this data type, displayed with the data without the need for the `to_char` function. This listing illustrates this concept:

```
SQL> create table timestamp_test (ts timestamp);
Table created.
SQL> insert into timestamp_test values (sysdate);
1 row created.
SQL> select * from timestamp_test;
TS
-----
14-DEC-09 05.25.07.000000 PM
SQL> create table date_test (d date);
Table created.
SQL> insert into date_test values (sysdate);
1 row created.
SQL> select * from date_test;
TS
-----
14-DEC-06
```

## clob

The *clob* data type allows storage of very large objects in excess of four gigabytes in size. Since this is a true character data type, it is very similar to the `varchar2` data type except for its much larger maximum size.

### blob

The *blob* data type permits storage of large unstructured binary objects. Sound and video are examples of blob data.

It's now time to have a look at the most common object in Oracle Database 11g: the table. After that, we will have a look at a few types of programming units written using SQL, which a person can store in Oracle Database 11g.

#### CRITICAL SKILL 1.4

## Work with Tables

The best way to think of a table in a relational database such as Oracle Database 11g is to see it as a spreadsheet with rows and columns. With this in mind, note the following:

- Rows are often referred to as *records*.
- Each column has a name unique to the table that it resides in.
- The intersection of each row and column, referred to as a cell in a spreadsheet, is called a *field* in Oracle Database 11g.

Picture the following SQL statement, which creates a table (the line numbers are not part of the code):

```
1- create table part_master (  
2-   id                number(8) not null,  
3-   manufacturer_code number(4) not null,  
4-   inception         date not null,  
5-   description       varchar2(60) not null,  
6-   unit_price        number(6,2) not null,  
7-   in_stock         varchar2(1));
```

Let's pick apart the code and highlight the main points in Table 1-2.

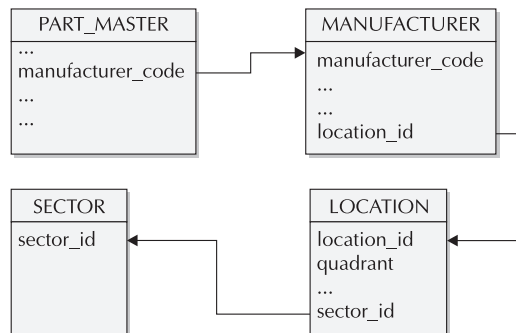
Table 1-2 mentions the concept of a relational database. Let's inspect a few other tables and see how they are related to one another.

## Tables Related to part\_master

The `manufacturer_code` column in `part_master` points to a record in `manufacturer`. Also, some columns in `manufacturer` may end up being related to column values in other tables. Figure 1-3 illustrates these relationship concepts, the heart of the Oracle Database 11g implementation.

Line	Important Points
1	The table has a unique name, from 1 to 30 characters. It is stored in Oracle Database's data dictionary in uppercase.
2	The ID column is numeric with anywhere from one to eight digits. The application that creates and keeps track of parts may insist that the first character of the ID be a digit between 1 and 9. Since the field is defined as numeric, if the leading digit were a 0, the part ID would only be seven digits long.
3	The <code>manufacturer_code</code> is the only manufacturer information stored in <code>part_master</code> . Further information about who made the product is in a related table—hence, the terminology <i>relational</i> database.
4	<code>inception</code> , as a date field, contains a date and time specification, though it will display a default month abbreviation and a two-character year unless some manual manipulation is performed (for example, 12-NOV-05).
5	<code>description</code> is a free-form field with a variable length of up to 30 characters.
6	<code>unit_price</code> can accommodate up to four integer and two decimal digits.
7	<code>in_stock</code> is a one-character flag of sorts; thus, the system designers can decide to use an indicator like a "1" or "X" to represent items that are in stock. Notice how this is the only one of seven fields in the <code>PART_MASTER</code> table that can be left blank.

**TABLE 1-2.** *part\_master* Table Definitions



**FIGURE 1-3.** *Relationships to part\_master*

---

Table	Part Number	Column Value	Related Column Value
part_master	33499909	manufacturer_code	3490
manufacturer	3490	location_id	5
location	5	quadrant	Pacific Northwest

---

**TABLE 1-3.** *Following Relationships Between Tables*

Suppose someone wanted to know where in the country a certain part was manufactured. By looking at Figure 1-3, that information is not readily available in `part_master`. However, `part_master` has a `manufacturer_code`. So, a person would traverse to `manufacturer` using `manufacturer_code` to get a `location_id`. Armed with that value, one then proceeds to `location` to get a `quadrant` column value. After this navigation is complete, a person would know where a specific part is built. Table 1-3 maps out this journey.

As illustrated in Table 1-3, you can deduce that part 33499909 comes from the Pacific Northwest—a deduction that is made by following the relationships between matching columns in the three tables in question.

### CRITICAL SKILL 1.5

## Work with Stored Programmed Objects

Oracle Database 11g offers the ability to store user-defined programming units in the data dictionary, called *stored objects*. These programming units are written in PL/SQL, the topic of Chapter 5. Without worrying about what goes inside these objects, let's do an overview of what they are all about.

### Views

Views are predefined subsets of data from an Oracle Database 11g table. The SQL query that builds the view is stored in the data dictionary and need not be reassembled every time the view is used. Suppose a personnel application stores the location of all employees in its `EMPLOYEE_MASTER` table in the `loc_id` column. With Oracle Database 11g, you can define a view called `emp_hq` as follows:

```
create or replace view emp_hq
as select * from employee_master
where loc_id = '2';
```

`EMP_HQ` becomes a valid object of the select statement just as if it were a table of its own.

## Ask the Expert

**Q: What is the major difference between the clob and blob data types in Oracle Database 11g?**

**A:** The clob stores only alphanumeric data, whereas the blob can accommodate any type of data, including sound and video.

**Q: When specifying the number data type, how is the total length of the field determined?**

**A:** The total length of a numeric field is determined by the digit(s) to the left of the comma if the specification includes an integer and decimal component. For example, *number(4,1)* denotes a maximum of four digits, of which one digit can be stored to the right of the decimal point.

**Q: Which of the Oracle Database 11g background processes is responsible for writing information from memory into the database files?**

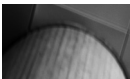
**A:** This is the job of the database writer, or dbw0, process.

**Q: Where does Oracle Database 11g read its environment from as it is started?**

**A:** The startup parameters are read from the system parameter file, which can be a binary file stored in Oracle Database 11g.

**Q: As sessions interact with the data in Oracle Database 11g, what role does the undo tablespace play in the architecture of the software?**

**A:** When transactions change the contents of information in Oracle Database 11g's tables, this special tablespace keeps a "before image" of the changes in case the operator decides to back out before saving newly entered information.



### NOTE

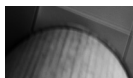
*At this point, you should realize that views are generally built based on more than one table. A view provides the perfect environment to predefine join conditions between tables to ensure that they adhere to business rules and perform at an optimal level.*

### Triggers

Just as their name implies, *triggers* are stored objects that fire based on the execution of certain events. Suppose a payroll application wants to audit salary increases: a trigger is created that fires when the salary column in `hr_master` is updated. The trigger could do the following:

1. Create a record in `sal_audit`.
2. Trap the time and date of the transaction.
3. Place the user's login ID in the `doer` column.
4. Place the old salary value in the `old_sal` column.
5. Place the new salary value in the `new_sal` column.

Code in the trigger traps the event by specifying *on update*. While triggers are commonly used for auditing, the types of activities they can initiate are endless.



#### NOTE

*Triggers cannot exist independently of an Oracle Database 11g table. They are associated with one and only one table and, if a table is dropped, so is the trigger.*

Triggers, as well as procedures, packages, and functions described next, are most commonly written using PL/SQL. The PL/SQL programming language is the topic of Chapter 5.

### Procedures

*Procedures* perform specific tasks as applications interact with Oracle Database 11g. If there are a number of interrelated activities to carry out in one place, a procedure is an ideal way to do this. Procedures can accept parameters when invoked and can interact with objects in Oracle Database 11g. They encapsulate related activities into single programming units that simplify logic and share data values as they perform various activities. They offer extremely flexible features, many of which are not available with triggers.

### Functions

*Functions* are very close relatives of procedures, except that they return a value to the code that called them. Oracle Database 11g delivers many functions out of the

box and developers can create their own functions to augment what is delivered with the software. Suppose you want to strip all the vowels out of a name with a function. You can pass in a name (for instance, Bellissimo) and gets back the text “Bllssm” when the function completes its work. Let’s look at the `get_age` function, which operates based on the following logic:

```
given a date of birth (format DD-MON-YYYY)
using an SQL function
get the months between today's date and the date passed in
divide the number of months by 12
truncate the results (giving the span in years between the 2 dates)
pass integer back
```

## Packages

*Packages* roll functions and procedures together into a cohesive programming unit. Often, developers prefer to bundle like functionality together since it makes logical sense to call one larger unit and have it perform a series of tasks. Let’s look at the `CREATE_EMPLOYEE` package in Table 1-4.

---

<b>Component Name</b>	<b>Type</b>	<b>Work Accomplished</b>
<code>give_holidays</code>	Procedure	Creates the default holiday quota based on the new person’s rank in the company.
<code>notify_benefits</code>	Procedure	Creates a record in the <code>BEN_QUEUE</code> table to alert the benefits people of the new employee.
<code>is_under_25</code>	Function	Returns a “1” if the new employee is under 25 years old as of December 31 of the year they were hired.
<code>is_over_59</code>	Function	Returns a “1” if the new employee is 60 years old or older as of the calendar date of hire.

---

**TABLE 1-4.** *Members of the CREATE\_EMPLOYEE Package*

## Progress Check

---

1. Oracle Database 11g is referred to as a relational database. Why is the word *relational* used?
  2. What is the maximum length of a `varchar2` data type that can be stored in Oracle Database 11g?
  3. What is data consistency when referred to as a feature of Oracle Database 11g? Give an example.
  4. What types of stored objects can be encapsulated into an Oracle Database 11g package?
  5. What is the fundamental difference between a procedure and a function in Oracle Database 11g?
  6. Data in the system tablespace is often referred to as *metadata*—or data about data. Name at least two types of metadata in the system tablespace.
  7. What is the difference between the timestamp and date data types?
  8. What type of information and data ends up being stored in the `sysaux` tablespace?
- 

### Progress Check Answers

1. The word *relational* is used because Oracle Database 11g defines the relationships between tables. It is these relationships that allow applications to navigate an assortment of tables and assemble results from more than one table.
2. The `varchar2` data type can accommodate up to 4000 characters.
3. Data consistency refers to the ability to ensure that related items of information are manipulated in a similar fashion. Suppose an application assigns a department to a new employee as a two-digit number field. Sometime down the road, due to company growth, the department identifier is changed to three digits. All the data where this used-to-be two-character identifier is stored must be changed to reflect the expansion of the department codes.
4. Packages can contain a mixture of one or more functions and procedures.
5. A procedure receives from zero to many parameters as it is invoked and then goes about its business until the end of its code segment. A function, on the other hand, accepts one or more parameters as it is called and returns a value to the code from where it was invoked. The procedure passes nothing back to its caller.
6. Metadata defines items such as the names of tables in the database, the owners of the tables, the data types of the columns in each table, and who is allowed to look at what data.
7. When columns are displayed, they use the date data type, containing a day/month/year component, whereas, by default, the timestamp data type columns contain a time-of-day component as well.
8. The `sysaux` tablespace contains tables required to manage Oracle Database, such as the items required to support OEM Grid Control.

**CRITICAL SKILL 1.6**

## Become Familiar with Other Important Items in Oracle Database 11g

So far, you've had a brief look at tables, views, tablespaces, and a handful of stored objects, such as views, triggers, procedures, packages, and functions. Let's round out this introduction to Oracle Database 11g architecture by covering a few other items commonly encountered. The following discussion is a hodgepodge of things that are necessary for a person's understanding of the Oracle Database 11g architecture and operations. Keep in mind that you must also spend a bit of time looking at the role of the database administrator, affectionately called the *DBA*, who is the gatekeeper of the database and the person responsible for its smooth operation.

**NOTE**

*You'll get a more detailed look at the DBA in Chapter 6, with more information on how DBAs go about carrying out their administrative chores.*

### Indexes

Tables are made up of rows and columns, which are the basis of all objects in Oracle Database 11g. As applications interact with the database, they often retrieve vast amounts of data. Suppose MyYP, a fictitious Internet company, provided Yellow Pages listings for North America, and the data was stored primarily in a table called YP\_MASTER. Each row in the YP\_MASTER table is uniquely identified by a combination of company name, municipality, and geographic location (state or province). As words are retrieved from the database to satisfy online queries, indexes would provide a quick access path to the qualifying data. Specific index characteristics are relevant to the power they deliver in Oracle Database 11g. For instance:

- They are built on one or more columns in a table using simple SQL statements.
- They are separate from the tables that they are built on and can be dropped without affecting the data in the table itself. On the contrary, when a table is dropped, any indexes it has disappear with the table.
- The function they perform can be likened to the index in a book. If one were looking for a specific topic in a textbook, the best place to start would be the index—it provides a shortcut to the information being sought. If one imagined that YP\_MASTER were a book rather than a table, finding Y&M Plumbing in Pensacola, Florida would be faster using the index than reading the book from the start to the 25<sup>th</sup> letter of the alphabet. The names on the corner of the pages in a phone book are like an index.

## 22 Oracle Database 11g: A Beginner's Guide

- Indexes occupy space in the database; even though there are ways to keep their space to a minimum, extra space is required and must be pre-allocated.

### Users

Most of us are familiar with usernames and passwords from our experience logging into corporate networks and other secure systems. Oracle Database 11g implements the same mechanism with login credentials and privileges given out by the database administrator. Once accounts are created, people initiate connections to Oracle Database 11g and work with their own data and other users' data where the appropriate privileges have been given out. We discuss object privileges in the "Work with Object and System Privileges" section immediately following this one.



#### NOTE

*With Oracle Database 11g, the terminology user, account, and schema are used synonymously.*

Once an account is created, it is often given the rights to occupy space in one or more Oracle Database 11g tablespaces. This is discussed in the next section.

### Tablespace Quotas

As additional nonsystem tablespaces are created, the database administrator gives out quotas that allow users to occupy space therein. *Tablespace quotas* are given out using an SQL statement with three parts:

- The username to whom the quota is being given.
- The name of the tablespace within which the username is being permitted to create tables.
- The amount of that quota—whether it's mentioned in absolute bytes (for example, 500,000) or more commonly in quantities of megabytes (500MB, for instance). Unlimited quotas can be allowed using the keyword *unlimited*.

Regardless of how a quota is given out, the SQL statement passed to Oracle Database 11g resembles the following:

```
SQL*Plus: Release 11.1.0.1.0 - Production on Sun Mar 11 10:29:42 2009
Copyright (c) 1982, 2007, Oracle. All rights reserved.
Connected to:
Oracle11g Enterprise Edition Release 11.1.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options
```

```
SQL> alter user hr quota 500m on hr_data;
User altered.
SQL> alter user ap quota unlimited on ap_idx;
User altered.
```

## Synonyms

You'll remember that in the "Work with Tables" section we discussed that the key was passing Oracle Database 11g the create table keywords. In a nutshell, table creation is undertaken after establishing a successful connection to the database, and then, with appropriate privileges in place, defining a table. One of the key concepts with all database management systems is sharing data. Since it is key to only have one copy of a table and to have its contents shared amongst applications, *synonyms* are a way to reference other people's data.

Suppose you wanted to use the PART\_MASTER table in an application owned by a user other than the owner. That owner would permit us to work with the table's data, and then we would create a synonym to reference its contents. The code would resemble the following:

```
SQL*Plus: Release 11.1.0.1.0 - Production on Sun Mar 11 10:29:42 2009
Copyright (c) 1982, 2007, Oracle. All rights reserved.
Connected to:
Oracle11g Enterprise Edition Release 11.1.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options
SQL> create synonym part_master for inv.part_master;
Synonym created.
SQL> select count(*)
  2  from part_master
  3  where in_stock is not null;
      COUNT(*)
-----
      13442
```

The preceding SQL statement references an object called part\_master. Depending on how your access is defined, the way that you reach the physical table may be different. When you are the owner of the table and you use the table's name in the from clause, Oracle understands that you would like to use your own table. If you do not own a table by that name, Oracle then looks in a list of table synonyms or pointers to a table with that name owned by someone else. This process of using synonyms is a transparent operation. If you do not own the table by the name or no synonym exists, you will receive an error. There are actually two kinds of synonyms:

- *Private synonyms* are created in one account and are only usable by the creator.
- *Public synonyms* are created by a central privileged user and are available to anyone able to connect to Oracle Database 11g.



### NOTE

*One needs the appropriate object privileges to be able to work with someone else's data using a private or public synonym. The synonym itself does not imply that the appropriate privileges can be circumvented.*

## Roles

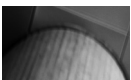
Often it makes sense to group similar users together to streamline the organization of people who use Oracle Database 11g. Using roles, the DBA can logically lump personnel together and give out object privileges to roles rather than individual users. Roles can be password protected, though in most implementations they do not have this level of complexity.

## Default User Environments

As accounts are created by the DBA, users are given a default environment to use unless some specifics are coded as they interact with Oracle Database 11g. Users are commonly set up with the following default environment settings:

- The *default tablespace* is where tables are placed unless the *create table* statement explicitly points at a nondefault tablespace that the user has a quota for.
- *Temporary tablespaces* are the tablespaces where users perform sort and merge operations while the Oracle Database 11g engine is processing queries.

Users can be given membership in one or more roles and have their default profile changed as well. As users are created, they do not automatically inherit a default tablespace; one must be manually given out during or following the user creation statement. Users do automatically point at a temporary tablespace, as discussed in the “Default Temporary Tablespace” section of this chapter, unless manually pointed elsewhere.



### NOTE

*With Oracle Database 11g one is now able to set the default tablespace for the entire database instance. This is done via an “ALTER DATABASE DEFAULT TABLESPACE tablespace-name;” command. You will need to have the privileges to be able to perform this operation.*

## Progress Check

1. Name at least four tasks handled by the Oracle Database 11g administrator.
2. What is the difference between *public* and *private* synonyms?
3. What is meant by a user's default tablespace?
4. What two units of measurement are commonly used to specify a tablespace quota?
5. Where do DBAs go to create *TARs*, where assistance is requested from Oracle's support organization?
6. Which of the following—*procedures*, *packages*, or *triggers*—cannot exist independent of a table to which they belong?

### CRITICAL SKILL 1.7

## Work with Object and System Privileges

It's next to impossible to work with data in Oracle Database 11g without looking at object privileges. In this section, we are going to look at these privileges as well as a suite of system privileges closely related to managing Oracle Database 11g. The four main object privileges are *select*, *insert*, *update*, and *delete*, all discussed in the next four sections. Oracle Database 11g uses the term *grant* when referring to giving out both object and system privileges.

### Select

This is the primary and most commonly used privilege, permitting other users to view your data. There are three parts to grant statements:

- The keywords grant select on.
- The name of the object upon which the privileges are being given out.
- The recipient of the grant.

#### Progress Check Answers

1. Installation, upgrades, tuning, and environment setup are four of many tasks performed by the DBA.
2. A private synonym can only be referenced in a SQL statement by the user who created and owns the synonym. A public synonym, created by a centralized user such as a DBA, is available to all users.
3. The default tablespace is the one within which users occupy space by default, unless another tablespace is mentioned as a table is created.
4. Quota on tablespaces is usually given out using bytes or megabytes as units of measurement.
5. The DBA goes to MetaLink to request assistance from Oracle's support organization.
6. Triggers cannot exist on their own without association with an Oracle Database 11g table.

Once the select privilege has been given out, the recipients, using a private or public synonym as described earlier in the “Synonyms” section of this chapter, can reference your objects in their SQL statements.

### Insert

This privilege allows users to create rows in tables belonging to other users. The creator of new rows in other users' objects is bound by the same rules used if they owned the objects themselves. They must adhere to the boundaries defined by the data types of the columns in the rows they create. For example, when rows are inserted into a table that has a column defined as type DATE, they must ensure that valid date type data is placed in the column so defined. As rows are created in an Oracle Database 11g table, the transaction must be committed to the database before the row becomes part of the information available to other users. With Oracle Database 11g, we use the term *commit* the same way the word *save* is used with other types of software.

### Update

This privilege allows a person to change the contents of columns in rows belonging to other tables. The SQL update statement can change the value of data in one or more columns. As with insert activity, the update transactions need to commit their work to make it permanent in the Oracle Database 11g files.

### Delete

Delete operations interact with one or more rows in Oracle Database 11g tables and must be followed by a commit as well to write the results of the transaction to the database files.

You will see more in Chapter 4 about how SQL statements are constructed using the four keywords in the previous sections. SQL statements are subject to rigorous syntax requirements which, if not followed, return an assortment of Oracle errors. Just as with other programming languages you may be familiar with, the SQL statement processing engine is very strict with reserved words and the placement of the pieces that come together to form an SQL transaction. Let's briefly discuss system privileges that allow certain users of Oracle Database 11g to perform secure activities.

### System Privileges

We have mentioned the database administrator in a number of places in this introductory chapter. Classically, secure operations are performed by the DBAs; however, one can grant system privileges to specified users so that they can perform

selected activities themselves. The following list illustrates a few examples of these secure operations:

- **Alter system** There are a number of modes that Oracle Database 11g can operate from. The modes are toggled using alter system. For example, this privilege can be given out to Jane by issuing the command grant alter system to jane.
- **Create user/alter user** Often, the DBA wants to partition some of the user creation activities between a handful of users of Oracle Database 11g. This is done by giving out the create user system privilege. Once new users are created, you often don't want to tweak their environment; this can be accomplished by issuing the grant alter user statement to one or more users of the database.
- **Create session/table/trigger** Sometimes when new users are created, they are given the create session system privilege which allows them to connect to Oracle Database 11g. In many cases, depending on how new users are created, they are not allowed to build any objects until they receive the create table system privilege. As well, many users are not capable of defining triggers until they receive the create trigger system privilege.

System privileges were introduced with early releases of Oracle7 (circa 1993) and have played a useful role in the division of labor in the database since their inception. Now it's time to get into the meat of the seventh letter of the alphabet, *g*, that throughout this chapter has followed the two-digit version number of this software release—11.

### CRITICAL SKILL 1.8

## Introduce Yourself to the Grid

As many have heard, the “*g*” in Oracle Database 11g stands for *grid*. Grid computing is a technology that allows for seamless and massively scalable access to a distributed network of diverse yet homogenous computer types. Oracle Database 11g is the glue permitting different vendors' computers to work together providing a seemingly endless supply of shared computer resources. Oracle sees the grid as revolutionizing the way companies go about doing their business. Grid computing targets the delivery of information as a utility, similar to the way electrical and telephone services are currently delivered to the public—hence the term *grid*. The industry as a whole, but Oracle in particular, sees a delivery method from the grid such that consumers will only pay for what they use. Interlaced computers will allow idle

## Ask the Expert

**Q: Name the four main object privileges used in Oracle Database 11g.**

**A:** The four most common privileges are select, insert, update, and delete.

**Q: Placing an Oracle Database 11g in a state where it can be accessed by applications is referred to as what activity?**

**A:** Putting an Oracle Database 11g in a normal operating mode for day-to-day access by a company's applications is referred to as *startup*.

**Q: How many integer and decimal digits can a field defined in the data dictionary as *number(10,2)* accommodate?**

**A:** The field would be able to store up to eight integer digits and two decimal digits.

**Q: When Oracle Database 11g is passed the value "Beginner " for storage in a varchar2 column, how does it deal with trailing insignificant spaces?**

**A:** The trailing spaces are trimmed before the information is stored in the database. Though not as common as varchar2, the char data type can be used to store trailing spaces.

**Q: What would Oracle Database 11g store as a value in a number(6,2) field when passed the value 9.8882?**

**A:** It would store 9.89 in a number(6,2) field when passed 9.8882.

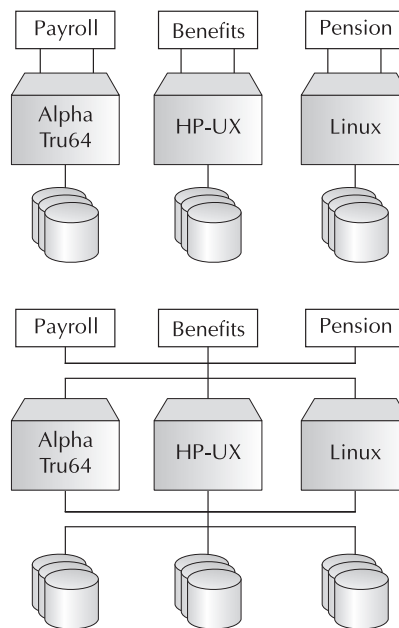
capacity to be leveraged by the grid to provide for a form of parallel processing on steroids. The following are the major players that enable the Oracle grid technology:

- **Real Application Clusters (RAC)** Involves a suite of networked computers sharing a common Oracle Database 11g and running platform-independent clusterware, the glue that makes the interconnect between the clustered nodes so transparent.
- **Automatic Storage Management (ASM)** A front-end management system that can group disks from an assortment of manufacturers together to form a

suite of disks that is available to all computers on the grid. ASM encapsulates the complete life cycle of disk management and allocation into a centralized GUI interface.

- **Oracle Resource Manager** Provides a framework within which administrators can control the computing resources of nodes on the grid.
- **Oracle Scheduler** Allows the handing out of jobs to members of the grid to facilitate the execution of business tasks anywhere and everywhere where idle resources exist.
- **Oracle Streams** Assists the processing requirements whereby copies of data need to be streamed between nodes in the grid, providing the mechanisms to keep data in sync on one database with the database from which the data originated. Oracle Streams' tight integration with the Oracle Database 11g engine facilitates this synchronization and delivers a preferred method of replication.

Figure 1-4 illustrates the primary differences between grid computing and traditional approaches to providing computer services.

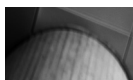


**FIGURE 1-4.** Differences between traditional and grid computing

The following points reinforce the details of the two scenarios depicted in Figure 1-4:

- The three applications at the top of the figure each have a dedicated server, each with its own dedicated disk. If the Linux server were to go out of service, the pension application would grind to a halt. There is no built-in mechanism for pension system processing to carry on another server.
- The three applications at the bottom are interlaced with one another. The benefits application can be hosted on from one to three of the available servers. As well, the database files that support these three applications can reside upon (and be read from) any of the nine disks in the grid's disk farm.

The browser-based OEM Grid Control holds the whole thing together. With the implementation of the ASM component of Oracle Database 11g, disks are managed by OEM, database instances are managed by OEM, clusterware is managed through OEM; the list is endless. Figure 1-5 shows the first OEM screen that appears after entering appropriate login credentials.



**NOTE**

*There is an OEM configuration program (called emca) that must be successfully run before you can access the browser-based OEM. The screen shown in your version of Oracle Database 11g may be somewhat different than the one shown in Figure 1-4. The look and feel of the OEM Grid Control screens can change significantly between minor releases of the software.*

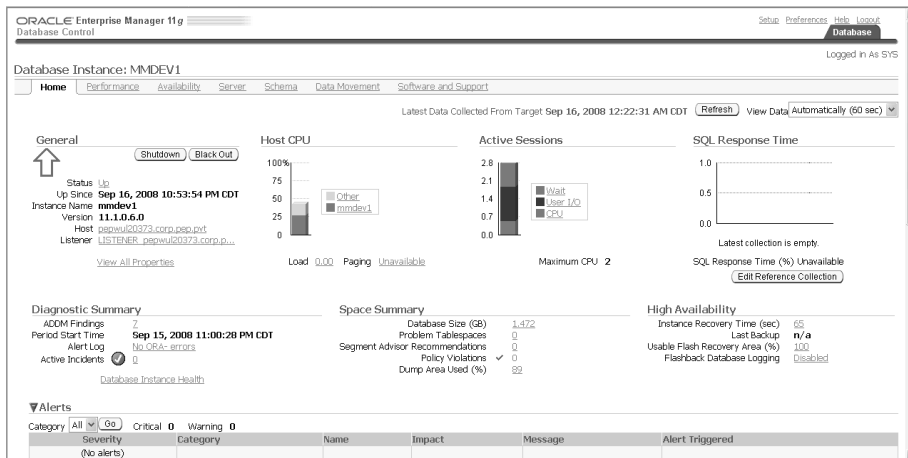


FIGURE 1-5. OEM startup

**CRITICAL SKILL 1.9**

## Tie It All Together

Now that was quite a journey! We have covered database fundamentals, with an Oracle Database 11g flavor. Relational database management systems have been around for a few decades, and the release of Oracle Database 11g is a landmark in the industry. There have been many academic discussions about the grid technology— some claim Oracle Database 11g is a grid implementation, while others don't. Regardless of which side of the fence you're on, Oracle Database 11g is a big step. Let's pull it all together and spend a bit of time on the big picture.

Oracle Database 11g is a collection of special files created using its database configuration assistant and then completing the work using OEM Grid Control. Access to these database files is facilitated by a set of shared memory processes referred to as an *instance*. Many technicians use this term synonymously with *database*. There is nothing wrong with that because, even though they are technically different pieces, they cannot survive without each other.

Relationships between objects in the database are defined in the data dictionary— hence the familiar term *relational database*. It is these relationships that provide the power and allow Oracle Database 11g to store vast amounts of data. Storing that data is one thing—retrieving it for applications in a quick and complete fashion is another story. Data retrieval is one of the strengths of the Oracle Database 11g engine.

Over the next eight chapters, we will be delving into more details of the Oracle Database 11g offering, paying specific attention to the following:

- **Installing Oracle** In order to become productive with Oracle, it is valuable to learn the steps that are necessary to install and connect to the database. This will help you get started as well as prepare you for the samples within the book.
- **Networking** The glue that holds many systems together and allows computers to communicate with one another in widely diverse and separated locations.
- **SQL—Structured Query Language** This is the way we communicate with Oracle Database 11g. Whatever the programming language (from C to Java), SQL is all the database engine understands.
- **PL/SQL** A programming language native to the Oracle Database 11g engine, providing more procedural capabilities that can amplify and enhance the functionality of SQL.
- **The Database Administrator** The person who is the gatekeeper of Oracle Database 11g and the one responsible for its smooth operation and optimal performance.

- **Backup and Recovery** Two areas critical to the smooth operation of Oracle Database 11g. Oracle Database 11g's Recovery Manager (referred to as RMAN) is the fundamental building block in its backup and recovery implementation. In addition, your backup and recovery strategy is one that will need to be considered as input to your Disaster Recovery strategy.
- **High Availability** The need to support databases and applications for extended and uninterrupted times requires Oracle to be able to support these demands. This chapter focuses on Rapid Application Clusters (RAC) and Automatic Storage Management (ASM) to serve as the foundation to your high availability solutions.
- **Large Database Features** Oracle Database 11g expands on an already solid offering in this area. With Oracle9i, they boasted the ability to support a database of up to 500 petabytes. Oracle Database 11g expands that upper limit to many exabytes, a staggering number, to say the least—where an exabyte is 1,152,921,504,606,846,976 bytes—or about one trillion million!

## Chapter 1 Mastery Check

1. The \_\_\_\_\_ background process is primarily responsible for writing information to the Oracle Database 11g files.
2. How many online redo log groups are required to start an Oracle Database 11g?
  - A. 3
  - B. 2
  - C. 4
  - D. 1
3. Of the following four items of information, which one is not stored in Oracle Database 11g's control files?
  - A. The name of the database files
  - B. The creator of the database
  - C. The location of the database files
  - D. The sizes of the database files
4. What is the function of a default temporary tablespace in the support of Oracle Database 11g?

5. Differentiate between an Oracle Database 11g and an instance.
6. Activities such as allocating space in the database and user management are commonly performed by the DBA. What feature in Oracle Database 11g allows some of these secure operations to be carried out by non-DBA users? How are these rights given out?
7. As a user of Oracle Database 11g is created, you often specify a default tablespace. In this context, what does *default tablespace* mean?
  - A. The system tablespace
  - B. A tablespace the user can occupy space in without a private or public synonym
  - C. The tablespace within which objects are created if a location (tablespace) is not explicitly mentioned as a table is created
8. The \_\_\_\_\_ GUI interface is used to create a new database.
9. What happens when one tries to store the text “Madagascar” in a field with a specification of `varchar2(8)`?
10. What is the most common way one uses triggers in Oracle Database 11g? Give an example of this activity.
11. What programming language, native to Oracle Database 11g, is used to create stored objects such as triggers and functions?
  - A. SQL\*Plus
  - B. OEM Grid Control
  - C. Basic
  - D. PL/SQL
12. What is the role of the `sysaux` tablespace in Oracle Database 11g?
13. The `clob` and `blob` data types differ in all but one of the following three ways. Which one does not apply to the differences between the two data types?
  - A. The `clob` holds standard alphanumeric data, whereas the `blob` may store binary information.
  - B. The `blob` contains a time (hour/minute) component, but the `clob` does not.
  - C. The `blob` contains unstructured free-form data, whereas the rules governing the type of information that can be stored in the `clob` are more stringent.

## 34 Oracle Database 11g: A Beginner's Guide

14. There are many ways to replicate data from one node to another. What main feature does Oracle Streams provide that is missing from many other methods?
15. What does the acronym SQL stand for?
  - A. Structured Query Language
  - B. Simple Query Language
  - C. Straightforward Question-based Learning