

*What You Need to Know About  
Oracle Database Architecture and Features*

**3rd Edition**  
Includes Oracle9i & Oracle8i



# Oracle Essentials

*Oracle Database 10g*



**O'REILLY®**

*Rick Greenwald,  
Robert Stackowiak & Jonathan Stern*

# Introducing Oracle

Where do we start? One of the problems in comprehending a massive product such as the Oracle database is the difficulty of getting a good sense of how the product works without getting lost in the details of implementing specific solutions. This book aims to solve this problem by giving you a thorough grounding in the concepts and technologies that form the foundation of the Oracle Database Server. Oracle also provides an Application Server and business applications, including the E-Business Suite and the Oracle Collaboration Suite,\* which are outside the scope of the main body of this book.

We've tried to write a book for a wide range of Oracle users, from the novice to the experienced user. To address this range of users, we've focused on the concepts and technology behind the Oracle database. Once you fully understand these facets of the product, you'll be able to handle the particulars of virtually any type of Oracle database. Without this understanding, you may feel overburdened as you try to connect the dots of Oracle's voluminous feature set and documentation.

This first chapter lays the groundwork for the rest of the discussions in this book. Of all the chapters, it covers the broadest range of topics; most of these are discussed further later in the book, but some of the basics—for example, the brief history of Oracle and the contents of the different “flavors” of the Oracle database products—are unique to this chapter.

Oracle has grown from its humble beginnings as one of a number of databases available in the 1970s to the market leader of today. In its early days, Oracle Corporation was known more as an aggressive sales and promotion organization than a technology supplier. Over the years, the Oracle database has grown in depth and quality,

---

\* Previous versions of this book did cover *iFS*. In Oracle Database 10g, however, this product is now known as Oracle Files and is now a part of the Oracle Collaboration Suite. For this reason, *iFS* is not covered in this edition.

and its technical capabilities now are generally recognized as the most advanced. With each release, Oracle has added more power and features to its already solid base while improving the manageability.

Several recent Oracle database releases are the focus of this book:

#### *Oracle8i*

Oracle8i, released in 1999, added a new twist to the Oracle database—a combination of enhancements that made the Oracle8i database the focal point of the world of Internet (the *i* in 8i) computing.

#### *Oracle9i*

Oracle9i, released in 2001, introduced Real Application Clusters as a replacement for Oracle Parallel Server, and added many management and data warehousing features.

#### *Oracle Database 10g*

Oracle Database 10g, released in 2003 and the current release, enables grid (the *g* in 10g) computing. A *grid* is simply a pool of computers that provides needed resources for applications on an as-needed basis. The goal is to provide computing resources that transparently scale to the user community, much as an electrical utility company can deliver power to meet peak demand by accessing energy from other power providers' plants via a power grid. Oracle Database 10g further reduces the time, cost, and complexity of database management through the introduction of self-managing features such as the Automated Database Diagnostic Monitor, Automated Shared Memory Tuning, Automated Storage Management, and Automated Disk Based Backup and Recovery. One important key to Oracle Database 10g's usefulness in grid computing is the ability to provision CPUs and data.

Before we dive into the specific foundations of these releases, we must spend a little time describing some Oracle basics—how databases evolved to arrive at the relational model, a brief history of Oracle Corporation, and an introduction to the basic features and configurations of the database.

## **The Evolution of the Relational Database**

The relational database concept was described first by Dr. Edgar F. Codd in an IBM research publication entitled “System R4 Relational” appearing in 1970. Initially, it was unclear whether any system based on this concept could achieve commercial success. Nevertheless, Relational Software, Incorporated (RSI) began in 1977 and released Oracle V.2 as the world's first relational database within a couple of years. By 1985, Oracle could claim more than 1,000 relational database customer sites. By comparison, IBM would not embrace relational technology in a commercial product until the Query Management Facility in 1983.

Why has relational database technology grown to become the de facto database technology since that time? A look back at previous database technology may help to explain this phenomenon.

Database management systems were first defined in the 1960s to provide a common organizational framework for what had been data stored in independent files. In 1964, Charles Bachman of General Electric proposed a network model with data records linked together, forming intersecting sets of data, as shown on the left in Figure 1-1. This work formed the basis of the CODASYL Data Base Task Group. Meanwhile, the North American Aviation's Space Division and IBM developed a second approach based on a hierarchical model in 1965. In this model, data is represented as tree structures in a hierarchy of records, as shown on the right in Figure 1-1. IBM's product based on this model was brought to market in 1969 as the Information Management System (IMS). As recently as 1980, almost all database implementations used either the network or hierarchical approach. Although several competitors utilized these technologies, only IMS remains.

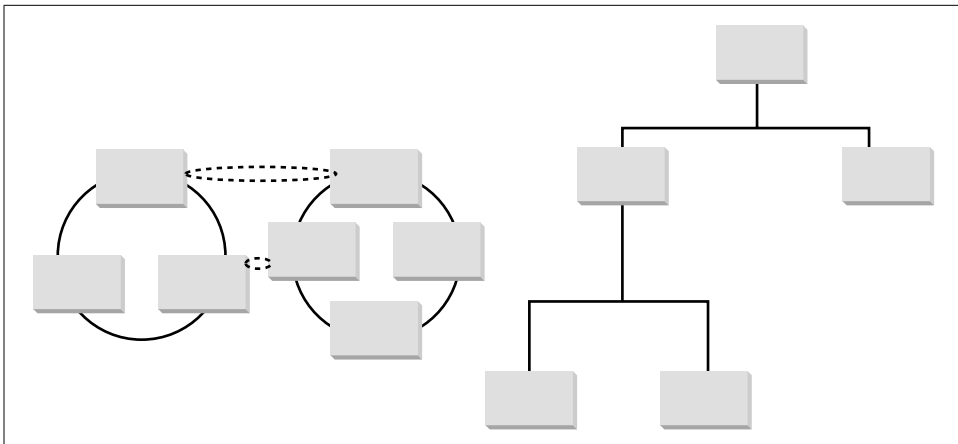


Figure 1-1. Network model (left) and hierarchical model (right)

## Relational Basics

The relational database uses the concept of linked two-dimensional tables consisting of rows and columns, as shown in Figure 1-2. Unlike the hierarchical approach, no predetermined relationship exists between distinct tables. This means that the data needed to link together the different areas of the network or hierarchical model need not be defined. Because relational users don't need to understand the representation of data in storage to retrieve it (many such users created ad hoc queries against the data), ease of use helped popularize the relational model.

Relational programming is nonprocedural and operates on a set of rows at a time. In a master-detail relationship between tables, there can be one or many detail rows for

<b>DEPTNO</b>	<b>DEPTNAME</b>	<b>LOCATION</b>
10	Accounting	San Francisco
20	Research	San Francisco
30	Sales	Chicago
40	Operations	Dallas

<b>EMPNO</b>	<b>EMPNAME</b>	<b>TITLE</b>	<b>DEPTNO</b>
71712	Johnson	Clerk	10
83321	Smith	Mgr	20
85332	Stern	SC Mgr	30
88888	Carter	Mgr	10

Figure 1-2. Relational model with two tables

each individual master row, yet the statements used to access, insert, or modify the data would simply describe the set of results. In many early relational databases, data access required the use of procedural languages that worked one record at a time. Because of this set orientation, programs can access more than one record in a relational database more easily. Relational databases can be used more productively to extract value from large groups of data.

The contents of the rows in Figure 1-2 are sometimes referred to as *records*. A column within a row is referred to as a *field*. Tables are stored in a database *schema*, which is a logical organizational unit within the database. Other logical structures in the schema often include the following:

#### *Views*

Provide a single view of data derived from one or more tables or views. The view is an alternative interface to the data, which is stored in the underlying table(s) that make up the view.

#### *Sequences*

Provide unique numbers for column values.

#### *Stored procedures*

Contain logical modules that can be called from programs.

#### *Synonyms*

Provide alternative names for database objects.

#### *Indexes*

Provide faster access to table rows.

#### *Database links*

Provide links between distributed databases.

The relationships between columns in different tables are typically described through the use of *keys*, which are implemented through referential integrity constraints and their supporting indexes. For example, in Figure 1-2, you can establish a link between the DEPTNO column in the second table, which is called a *foreign key*, to the DEPTNO column in the first table, which is referred to as the *primary key* of that table.

Finally, even if you define many different indexes for a table, you don't have to understand them or manage the data they contain. Oracle includes a *query optimizer* (described in Chapter 4) that chooses the best way to use your indexes to access the data for any particular query.

The relational approach lent itself to the Structured Query Language (SQL). SQL was initially defined over a period of years by IBM Research, but it was Oracle Corporation that first introduced it to the market in 1979. SQL was noteworthy at the time for being the only language needed to use relational databases, because you could use SQL:

- For queries (using a SELECT statement)
- As a Data Manipulation Language or DML (using INSERT, UPDATE, and DELETE statements)
- As a Data Definition Language or DDL (using CREATE or DROP statements when adding or deleting tables)
- To set privileges for users or groups (using GRANT or REVOKE statements)

Today, SQL contains many extensions with ANSI/ISO standards that define its basic syntax.

## How Oracle Grew

In 1983, RSI was renamed Oracle Corporation to avoid confusion with a competitor named RTI. At this time, the developers made a critical decision to create a portable version of Oracle (Version 3) that ran not only on Digital VAX/VMS systems, but also on Unix and other platforms. By 1985, Oracle claimed the ability to run on more than 30 platforms. Some of these platforms are historical curiosities today, but others remain in use. (In addition to VMS, early operating systems supported by Oracle included IBM MVS, DEC Ultrix, HP/UX, IBM AIX, and Sun's Solaris version of Unix.) Oracle was able to leverage and accelerate the growth of minicomputers and Unix servers in the 1980s. Today, Oracle is portable to both Microsoft Windows and Linux, which are the leading operating systems on popular commodity servers.

In addition to multiple platform support, other core Oracle messages from the mid-1980s still ring true today, including complementary software development and decision support tools, ANSI standard SQL across platforms, and connectivity over

standard networks. Since the mid-1980s, the database deployment model has evolved from dedicated database application servers to client/server to Internet computing implemented with PCs and thin clients accessing database applications via browsers—and, to the grid with Oracle Database 10g.

Oracle introduced many innovative technical features to the database as computing and deployment models changed (from offering the first distributed database to the first Java Virtual Machine in the core database engine). Oracle also continues to support emerging standards such as XML and .NET. Table 1-1 presents a short list of Oracle’s major feature introductions.

*Table 1-1. History of Oracle technology introductions*

Year	Feature
1979	Oracle Release 2—the first commercially available relational database to use SQL
1983	Single code base for Oracle across multiple platforms
1984	Portable toolset
1986	Client/server Oracle relational database
1987	CASE and 4GL toolset
1988	Oracle Financial Applications built on relational database
1989	Oracle6
1991	Oracle Parallel Server on massively parallel platforms
1993	Oracle7 with cost-based optimizer
1994	Oracle Version 7.1 generally available: parallel operations including query, load, and create index
1996	Universal database with extended SQL via cartridges, thin client, and application server
1997	Oracle8 generally available: including object-relational and Very Large Database (VLDB) features
1999	Oracle8i generally available: Java Virtual Machine (JVM) in the database
2000	Oracle9i Application Server generally available: Oracle tools integrated in middle tier
2001	Oracle9i Database Server generally available: Real Application Clusters; OLAP and data mining API in the database
2003	Oracle Database 10g enables grid computing and simplifies and automates key management tasks

# The Oracle Family

Oracle Database 10g Database Server describes the most recent major version of the Oracle Relational Database Management System (RDBMS) family of products that share common source code. This family includes:

- Personal Oracle, a database for single users that’s often used to develop code for implementation on other Oracle multiuser databases
- Oracle Standard Edition, which was named Workgroup Server in its first iteration as part of the Oracle7 family and is sometimes simply referred to as Oracle Server

- Oracle Enterprise Edition, which includes all Standard Edition functionality and additional functionality
- Oracle Lite, used primarily for mobile applications

Oracle8 was introduced in 1997 with larger size limitations and management features, such as partitioning, aimed at very large database implementations. In 1998, Oracle announced Oracle8*i*, which is sometimes referred to as Version 8.1 of the Oracle8 database. The *i* was added to denote added functionality supporting Internet deployment in the new version. Oracle9*i* followed, with Application Server available in 2000 and Database Server in 2001. Oracle Database 10g was introduced in 2003; the *g* denotes Oracle's focus on emerging grid deployment models. The terms Oracle, Oracle8, Oracle8*i*, Oracle9*i* and Oracle Database 10g (or Oracle10g) might appear to be used somewhat interchangeably in this book, because Oracle Database 10g includes all the features of previous versions. When we describe a new feature that was first made available specifically in certain releases, we've tried to note that fact to avoid confusion, recognizing that many of you may have old releases of Oracle. We typically use the simple term Oracle when describing features that are common to all these releases.

Oracle has focused development around a single source code model since 1983. While each database implementation includes some operating system-specific source code at very low levels in order to better leverage specific platforms, the interfaces that users, developers, and administrators deal with for each version are consistent. Because features are consistent across platforms for implementations of Oracle Standard Edition and Oracle Enterprise Edition, companies can migrate Oracle applications easily to various hardware vendors and operating systems while leveraging their investments in Oracle technology. This development strategy also enables Oracle to focus on implementing new features only once in its product set, instead of having to add functionality at different times to different implementations.

## Oracle Standard Edition

Oracle Standard Edition refers to a specific database offering, once known as Workgroup Server. From a functionality and pricing standpoint, this product intends to compete in the entry-level multiuser and small database category, supporting smaller numbers of users. These releases are available today on Windows and Unix platforms such as HP Compaq, HP/UX, IBM AIX, Linux, and Sun Solaris.

## Oracle Enterprise Edition

Oracle Enterprise Edition is aimed at larger-scale implementations that require additional features. Enterprise Edition is available on far more platforms than the Oracle release for workgroups and includes advanced management, networking, programming, and data warehousing features, as well as a variety of special-purpose options, such as clustering, which are available at extra cost.



# Oracle Personal Edition

Oracle Personal Edition is the single-user version of Oracle Enterprise Edition. Personal Edition is most frequently used for development on a single machine. Because the features match those of Enterprise Edition, a developer can write applications using the Personal Edition and deploy them to multi-user servers. Some companies deploy single-user applications using this product. However, Oracle Lite offers a much more lightweight means of deploying the same applications.

# Oracle Lite

Oracle Lite, once known as Oracle Mobile, is intended for single users who are using wireless/mobile devices. It differs from other members of the Oracle database family in that it doesn't use the same database engine. Instead, Oracle developed a lightweight engine compatible with the limited memory and storage capacity of hand-held devices. Oracle Lite is described in more detail at the end of this chapter.

Because the SQL supported by Oracle Lite is largely the same as the SQL for other Oracle databases, you can run applications developed for those database engines using Oracle Lite. Replication of data between Oracle Lite and other Oracle versions is a key part of most implementations.

Table 1-2 summarizes the situations in which you would typically use each database product. We've used the Oracle product names to refer to the different members of the Oracle database family.

Table 1-2. Oracle family of database products

Database name	When appropriate
Oracle Standard Edition	Version of Oracle server for a small number of users and a smaller database
Oracle Enterprise Edition	Version of Oracle for a large number of users or a large database with advanced features for extensibility, performance, and management
Oracle Personal Edition	Single-user version of Oracle typically used for development of applications for deployment on other Oracle versions
Oracle Lite	Lightweight database engine for mobile computing on notebooks and handheld devices

# Summary of Oracle Features

The Oracle database is a broad and powerful product. The remainder of this book examines different aspects of Oracle such as data structures, performance, and parallel processing. But before you can understand each of the different areas of Oracle in depth, you must familiarize yourself with the range of features in the Oracle database.

The rest of this chapter gives you a high-level overview of the basic areas of functionality in the Oracle product family. By the end of this chapter, you will at least have some orientation points to guide you in exploring the topics in the rest of this book.

To give some structure to the broad spectrum of the Oracle database, we've organized the features into the following sections:

- Database application development features
- Database connection features
- Distributed database features
- Data movement features
- Performance features
- Database management features

At the end of each of the following sections describing database features we've included a subsection called "Availability," which indicates the availability of each feature in specific Oracle products. You should be aware that as this feature list grows and Oracle implements packaging changes in new versions, the availability of these features in the version you implement may vary slightly.



In this chapter, we've included a lot of terminology and rather abbreviated descriptions of features. Oracle is a huge system. Our goal here is to quickly familiarize you with the full range of features in the system. Subsequent chapters will provide additional details. Obviously, though, whole books can be (and have been!) written about each of the feature areas summarized here.

## Database Application Development Features

The main use of the Oracle database system is to store and retrieve data for applications. The features of the Oracle database and related products described in this section are used to create applications. We've divided the discussion in this section into two categories: database programming and database extensibility options. Later in this chapter, we describe the Oracle Developer Suite, a set of optional tools used in Oracle Database Server and Oracle Application Server development.

### Database Programming

All flavors of the Oracle database include different languages and interfaces that allow programmers to access and manipulate the data in the database. Database programming features usually interest two groups: developers building Oracle-based applications that will be sold commercially, and IT organizations within companies that custom-develop applications unique to their businesses. The following sections describe the languages and interfaces supported by Oracle.

## SQL

The ANSI standard Structured Query Language (SQL) provides basic functions for data manipulation, transaction control, and record retrieval from the database. However, most end users interact with Oracle through applications that provide an interface that hides the underlying SQL and its complexity.

## PL/SQL

Oracle's PL/SQL, a procedural language extension to SQL, is commonly used to implement program logic modules for applications. PL/SQL can be used to build stored procedures and triggers, looping controls, conditional statements, and error handling. You can compile and store PL/SQL procedures in the database. You can also execute PL/SQL blocks via SQL\*Plus, an interactive tool provided with all versions of Oracle. Oracle Database 10g includes a more optimized version of the core PL/SQL engine, as Oracle9i allowed creation and storage of precompiled PL/SQL program units.

## Java features and options

Oracle8i introduced the use of Java as a procedural language with a Java Virtual Machine (JVM) in the database (originally called JServer). JVM includes support for Java stored procedures, methods, triggers, Enterprise JavaBeans™ (EJBs), CORBA, and HTTP. The Accelerator is used for project generation, translation, and compilation, and can also be used to deploy/install shared libraries.

The inclusion of Java within the Oracle database allows Java developers to leverage their skills as Oracle application developers. Java applications can be deployed in the client, Application Server, or database, depending on what is most appropriate. We discuss Java development in Chapters 13 and 14.

Oracle data warehousing options for OLAP and data mining provide a Java API. These applications are typically custom built using Oracle's JDeveloper.

## Large objects

Interest in the use of large objects (LOBs) continues to grow, particularly for storing nontraditional datatypes such as images. The Oracle database has been able to store large objects for some time. Oracle8 added the capability to store multiple LOB columns in each table. Oracle Database 10g essentially removes the space limitation on large objects.

## Object-oriented programming

Support of object structures has been included since Oracle8i to allow an object-oriented approach to programming. For example, programmers can create user-defined datatypes, complete with their own methods and attributes. Oracle's object support includes a feature called Object Views through which object-oriented programs can

make use of relational data already stored in the database. You can also store objects in the database as varying arrays (VARARRAYs), nested tables, or index organized tables (IOTs). We discuss the object-oriented features of Oracle further in Chapter 13.

### **Third-generation languages (3GLs)**

Programmers can interact with the Oracle database from C, C++, Java, COBOL, or FORTRAN applications by embedding SQL in those applications. Prior to compiling the applications using a platform's native compilers, you must run the embedded SQL code through a precompiler. The precompiler replaces SQL statements with library calls the native compiler can accept. Oracle provides support for this capability through optional "programmer" precompilers for languages such as C and C++ (Pro\*C) and COBOL (Pro\*COBOL). More recently, Oracle added SQLJ, a precompiler for Java that replaces SQL statements embedded in Java with calls to a SQLJ runtime library, also written in Java.

### **Database drivers**

All versions of Oracle include database drivers that allow applications to access Oracle via ODBC (the Open DataBase Connectivity standard) or JDBC (the Java DataBase Connectivity open standard). Also available are data providers for OLE DB and for .NET.

### **The Oracle Call Interface**

If you're an experienced programmer seeking optimum performance, you may choose to define SQL statements within host-language character strings and then explicitly parse the statements, bind variables for them, and execute them using the Oracle Call Interface (OCI).

OCI is a much more detailed interface that requires more programmer time and effort to create and debug. Developing an application that uses OCI can be time-consuming, but the added functionality and incremental performance gains often make spending the extra time worthwhile.

#### **Why Use OCI?**

Why would someone want to use OCI instead of the higher-level interfaces? In certain programming scenarios, OCI improves application performance or adds functionality. For instance, in high-availability implementations in which multiple systems share disks and implement Real Application Clusters/Oracle Parallel Server, you may want users to reattach to a second server transparently if the first fails. You can write programs that do this using OCI.

## National Language Support

National Language Support (NLS) provides character sets and associated functionality, such as date and numeric formats, for a variety of languages. Oracle9i featured full Unicode 3.0 support. All data may be stored as Unicode, or select columns may be incrementally stored as Unicode. UTF-8 encoding and UTF-16 encoding provide support for more than 57 languages and 200 character sets. Oracle Database 10g adds support for Unicode 3.2. Extensive localization is provided (for example, for data formats) and customized localization can be added through the Oracle Locale Builder. Oracle Database 10g includes a Globalization Toolkit for creating applications that will be used in multiple languages.

## Availability

All of these database programming features are included in both Oracle Standard Edition and Oracle Enterprise Edition.

## Database Extensibility

The Internet and corporate intranets have created a growing demand for storage and manipulation of nontraditional datatypes within the database. There is a need for extensions to the standard functionality of a database for storing and manipulating image, audio, video, spatial, and time series information. These capabilities are enabled through extensions to standard SQL.

For more details regarding these features of Oracle, see Chapter 13.

## Oracle Text and interMedia

Oracle Text includes what was previously referred to as the “ConText cartridge” with Ultrasearch capabilities. It can identify the gist of a document by searching for themes and key phrases in the document.

Oracle *interMedia* bundles additional image, audio, video, and locator functions and is included in the database license. Oracle *interMedia* offers the following capabilities:

- The image portion of *interMedia* can store and retrieve images.
- The audio and video portions of *interMedia* can store and retrieve audio and video clips, respectively.
- The locator portion of *interMedia* can retrieve data that includes spatial coordinate information.

## Oracle Spatial Option

The Spatial option is available for Oracle Enterprise Edition. It can optimize the display and retrieval of data linked to coordinates and is used in the development of

spatial information systems. Several vendors of Geographic Information Systems (GIS) products now bundle this option and leverage it as their search and retrieval engine.

## **XML**

Oracle added native XML datatype support to the Oracle9i database and XML and SQL interchangeability for searching. The structured XML object is held natively in object relational storage meeting the W3C DOM specification. The XPath syntax for searching in SQL is based on the SQLX group specifications.

# **Database Connection Features**

The connection between the client and the database server is a key component of the overall architecture of a computing system. The database connection is responsible for supporting all communications between an application and the data it uses. Oracle includes a number of features that establish and tune your database connections.

The following features relate to the way the Oracle database handles the connection between the client and server machines in a database interaction. We've divided the discussion in this section into two categories: database networking and Oracle Application Server.

## **Database Networking**

Database users connect to the database by establishing a network connection. You can also link database servers via network connections. Oracle provides a number of features to establish connections between users and the database and/or between database servers, as described in the following sections.

### **Oracle Net**

Oracle's network interface, Oracle Net, was formerly known as Net8 when used in Oracle8, and SQL\*Net when used with Oracle7 and previous versions of Oracle. You can use Oracle Net over a wide variety of network protocols, although TCP/IP is by far the most common protocol today.

Features associated with Oracle Net, such as shared servers, are referred to as Oracle Net Services.

### **Oracle Names**

Oracle Names allows clients to connect to an Oracle server without requiring a configuration file on each client. Using Oracle Names can reduce maintenance efforts, because a change in the topology of your network will not require a corresponding change in configuration files on every client machine.

## Oracle Internet Directory

The Oracle Internet Directory (OID) was introduced with Oracle8i. OID serves the same function as Oracle Names in that it gives users a way to connect to an Oracle Server without having a client-side configuration file. However, OID differs from Oracle Names in that it is an LDAP (Lightweight Directory Access Protocol) directory; it does not merely support the Oracle-only Oracle Net protocol.

## Oracle Connection Manager

Each connection to the database takes up valuable network resources, which can impact the overall performance of a database application. Oracle's Connection Manager, illustrated in Figure 1-3, reduces the number of Oracle Net client network connections to the database through the use of *concentrators*, which provide connection multiplexing to implement multiple connections over a single network connection. Connection multiplexing provides the greatest benefit when there are a large number of active users.

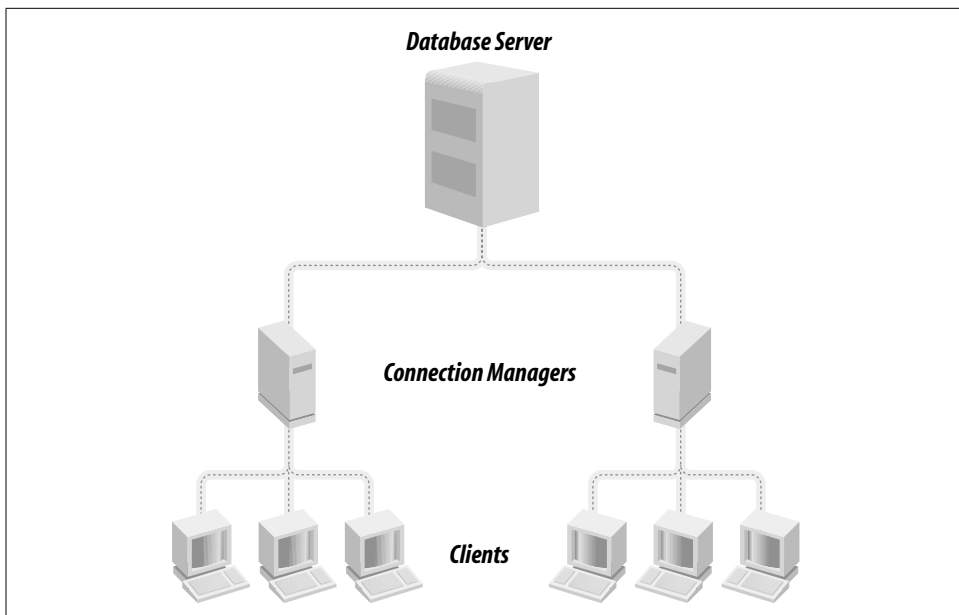


Figure 1-3. Concentrators with Connection Managers for a large number of users

You can also use the Connection Manager to provide multiprotocol connectivity when clients and servers run different network protocols. This capability replaces the multiprotocol interchange formerly offered by Oracle, but it is less important today because most companies use TCP/IP as their standard protocol.

Oracle Database 10g introduces dynamic Connection Manager configuration, enabling changing of CMAN parameters without shutting down the CMAN process.

## **Advanced Security Option**

The Advanced Security Option was once known as the Advanced Networking Option (ANO). Key features include network encryption services using RSA Data Security's RC4 or DES algorithm, network data integrity checking, enhanced authentication integration, single sign-on, and DCE (Distributed Computing Environment) integration.

## **Label Security Option**

Oracle Label Security controls access to data by comparing labels assigned to data with label authorizations granted to users. Multiple authorization levels are possible within a single database. Label security authorizations are managed through the Policy Manager. Policies are enforced in the database instead of through views, thus greatly simplifying management of data accessibility and providing a more secure implementation.

## **Availability**

Advanced networking features such as the Oracle Connection Manager, Advanced Security Option, and Label Security Option have typically been available for the Enterprise Edition of the database, but not for the Standard Edition.

## **Oracle Application Server**

The popularity of Internet and intranet applications has led to a change in deployment from client/server (with fat clients running a significant piece of the application) to a three-tier architecture (with a browser supplying everything needed on a thin client). The Oracle Application Server provides a means of implementing the middle tier of a three-tier solution for web-based applications, component-based applications, and enterprise application integration. Oracle9iAS and Oracle Application Server 10g (referenced here together as Application Server) replaced a previous Oracle Application Server (OAS) and Oracle Web Application Server. Oracle's Application Server can be scaled across multiple middle-tier servers.

This product includes a web listener based on the popular Apache listener, servlets and JavaServer Pages (JSPs), business logic, and/or data access components. Business logic might include JavaBeans, Business Components for Java (BC4J), and Enterprise JavaBeans (EJBs). Data access components can include JDBC, SQLJ, BC4J, and EJBs. TopLink provides a mapping tool that links Java objects to the database via JDBC such that the Java developer need not build SQL calls and or face broken Java applications due to database schema changes.

Oracle's Application Server offers additional solutions in the cache, portal, intelligence, and wireless areas:



### *Cache*

Oracle Application Server Web Cache introduced a middle tier for the caching of web pages or portions of pages. An earlier cache, Oracle Application Server Database Cache, was used for caching PL/SQL procedures and anonymous PL/SQL blocks but is no longer supported with Oracle Application Server 10g.

### *Portal*

The Oracle Application Server Portal is also a part of the Oracle Developer Suite (discussed later in this chapter) and is used for building easy-to-use browser interfaces to applications through servlets and HTTP links. The developed portal is deployed to the Application Server.

### *Business Intelligence*

Application Server Business Intelligence components include the Portal, but also:

- Oracle Reports, which provides a scalable middle tier for the reporting of prebuilt query results
- Oracle Discoverer, for ad hoc query and relational online analytical processing (ROLAP)
- A deployment platform for OLAP and data mining applications custom-built with JDeveloper

These capabilities are discussed in Chapter 9.

### *Wireless*

Oracle Wireless Edition (formerly known as Oracle Portal-to-Go) includes:

- Content adapters for transforming content to XML
- Device transformers for transforming XML to device-specific markup languages
- Personalization portals for service personalization of alerts, alert addresses, location marks, and profiles; the wireless personalization portal is also used for the creation, servicing, testing, and publishing of URL service and for user management

## **Availability**

Because Oracle Application Server is a separate product, it can be used with various versions of the Oracle database and with either edition.

The Application Server packaging includes a Standard Edition, an Enterprise Edition, and a Java Edition (including key components for Java developers). Portal and TopLink are included in the Standard Edition. The Enterprise Edition adds the following capabilities: Forms Services, Reports Services, Discoverer Viewer, Oracle Internet Directory, Oracle Workflow, and Oracle Application Interconnect. The Java Edition bundle includes a HTTP Server, OC4J, TopLink, BC4J, and JDeveloper. For more details about Application Server, see Chapter 14.

# Distributed Database Features

One of the strongest features of the Oracle database is its ability to scale up to handle extremely large volumes of data and users. Oracle scales not only by running on more and more powerful platforms, but also by running in a distributed configuration. Oracle databases on separate platforms can be combined to act as a single logical distributed database.

This section describes some of the basic ways that Oracle handles database interactions in a distributed database system.

## Distributed Queries and Transactions

Data within an organization is often spread among multiple databases for reasons of both capacity and organizational responsibility. Users may want to query this distributed data or update it as if it existed within a single database.

Oracle first introduced distributed databases in response to the requirements for accessing data on multiple platforms in the early 1980s. *Distributed queries* can retrieve data from multiple databases. *Distributed transactions* can insert, update, or delete data on distributed databases. Oracle's two-phase commit mechanism, which is described in detail in Chapter 12, guarantees that all the database servers that are part of a transaction will either commit or roll back the transaction. Background recovery processes can ensure database consistency in the event of system interruption during distributed transactions. Once the failed system comes back online, the same process will complete the distributed transactions.

Distributed transactions can also be implemented using popular transaction monitors (TPs) that interact with Oracle via XA, an industry standard (X/Open) interface. Oracle8i added native transaction coordination with the Microsoft Transaction Server (MTS), so you can implement a distributed transaction initiated under the control of MTS through an Oracle database.

## Heterogeneous Services

Heterogeneous Services allow non-Oracle data and services to be accessed from an Oracle database through generic connectivity via ODBC and OLE-DB included with the database.

Optional Transparent Gateways use agents specifically tailored for a variety of target systems. Transparent Gateways allow users to submit Oracle SQL statements to a non-Oracle distributed database source and have them automatically translated into the SQL dialect of the non-Oracle source system, which remains transparent to the user. In addition to providing underlying SQL services, Heterogeneous Services provide transaction services utilizing Oracle's two-phase commit with non-Oracle databases and procedural services that call third-generation language routines on

non-Oracle systems. Users interact with the Oracle database as if all objects are stored in the Oracle database, and Heterogeneous Services handle the transparent interaction with the foreign database on the user's behalf.

## Availability

All the technologies discussed in this section are included in both editions of the database, although Transparent Gateways are an add-on available at additional cost.

## Data Movement Features

Moving data from one Oracle database to another is often a requirement when using distributed databases, or when a user wants to implement multiple copies of the same database in multiple locations to reduce network traffic or increase data availability. You can export data and data dictionaries (metadata) from one database and import them into another. Oracle Database 10g introduces a new high speed data pump for the import and export of data. Oracle also offers many other advanced features in this category, including replication, transportable tablespaces, and Advanced Queuing.

This section describes the technology used to move data from one Oracle database to another automatically.

## Basic Replication

You can use basic replication to move recently added and updated data from an Oracle “master” database to databases on which duplicate sets of data reside. In basic replication, only the single master is updated. You can manage replication through the Oracle Enterprise Manager (OEM or EM). While replication has been a part of all recent Oracle releases, replication based on logs is a more recent addition, first appearing in Oracle9i Release 2.

## Advanced Replication

You can use advanced replication in multimaster systems in which any of the databases involved can be updated and conflict-resolution features are needed to resolve inconsistencies in the data. Because there is more than one master database, the same data may be updated on multiple systems at the same time. Conflict resolution is necessary to determine the “true” version of the data. Oracle's advanced replication includes a number of conflict-resolution scenarios and also allows programmers to write their own. We cover replication in more detail in Chapter 12.

## Transportable Tablespaces

Transportable tablespaces were introduced in Oracle8i. Instead of using the export/import process, which dumps data and the structures that contain it into an intermediate file for loading, you simply put the tablespaces in read-only mode, move or copy them from one database to another, and mount them. You must export the data dictionary (metadata) for the tablespace from the source and import it at the target. This feature can save a lot of time during maintenance, because it simplifies the process. Oracle Database 10g allows you to move data with transportable tablespaces between different platforms or operating systems.

## Advanced Queuing and Streams

Advanced Queuing (AQ), first introduced in Oracle8, provides the means to asynchronously send messages from one Oracle database to another. Because messages are stored in a queue in the database and sent asynchronously when a connection is made, the amount of overhead and network traffic is much lower than it would be using traditional guaranteed delivery through the two-phase commit protocol between source and target. By storing the messages in the database, AQ provides a solution with greater recoverability than other queuing solutions that store messages in filesystems.

Oracle messaging adds the capability to develop and deploy a *content-based publish and subscribe solution* using a rules engine to determine relevant subscribing applications. As new content is published to a subscriber list, the rules on the list determine which subscribers should receive the content. This approach means that a single list can efficiently serve the needs of different subscriber communities.

In the first release of Oracle9i, AQ added XML support and Oracle Internet Directory (OID) integration. This technology is leveraged in Oracle Application Interconnect (OAI), which includes adapters to non-Oracle applications, messaging products, and databases.

The second release of Oracle9i introduced *Streams*. Streams have three major components: log-based replication for data capture, queuing for data staging, and user-defined rules for data consumption. Oracle Database 10g includes support for change data capture and file transfer solutions via Streams.

## Extraction, Transformation, Loading

Oracle Warehouse Builder is a tool for the design of target data stores including data warehouses and a metadata repository, but it also provides a frontend to building source-to-target maps and for generating extraction, transformation, and loading (ETL) scripts. OWB leverages key embedded ETL features first made available in the Oracle9i database.

## Availability

Although basic replication has been included with both Oracle Standard Edition and Enterprise Edition, advanced features such as advanced replication, transportable tablespaces, and Advanced Queuing have typically required Enterprise Edition.

## Performance Features

Oracle includes several features specifically designed to boost performance in certain situations. We've divided the discussion in this section into two categories: database parallelization and data warehousing.

### Database Parallelization

Database tasks implemented in parallel speed up querying, tuning, and maintenance of the database. By breaking up a single task into smaller tasks and assigning each subtask to an independent process, you can dramatically improve the performance of certain types of database operations.

Parallel query features became a standard part of Enterprise Edition beginning with Oracle 7.3. Parallel query became supported in Virtual Private Databases (VPD) with Oracle Database 10g. Examples of query features implemented in parallel include:

- Table scans
- Nested loops
- Sort merge joins
- GROUP BYs
- NOT IN subqueries (anti-joins)
- User-defined functions
- Index scans
- Select distinct UNION and UNION ALL
- Hash joins
- ORDER BY and aggregation
- Bitmap star joins
- Partition-wise joins
- Stored procedures (PL/SQL, Java, external routines)

When you're using Oracle, by default the degree of parallelism for any operation is set to twice the number of CPUs. You can adjust this degree automatically for each subsequent query based on the system load. You can also generate statistics for the cost-based optimizer in parallel. Parallel operations are described in more detail in Chapter 6.

## Availability

You can perform maintenance functions such as loading (via SQL\*Loader), backups, and index builds in parallel in Oracle Enterprise Edition. Oracle Partitioning for the Enterprise Edition enables additional parallel Data Manipulation Language (DML) inserts, updates, and deletes as well as index scans.

## Data Warehousing and Business Intelligence

The parallel features discussed in the previous section improve the overall performance of the Oracle database. Oracle has also added some performance enhancements that specifically apply to data warehousing applications. For detailed explanations of these and complementary products and features related to data warehousing and business intelligence, see Chapter 9.

### Bitmap indexes

Oracle added support for stored bitmap indexes to Oracle 7.3 to provide a fast way of selecting and retrieving certain types of data. Bitmap indexes typically work best for columns that have few different values relative to the overall number of rows in a table.

Rather than storing the actual value, a bitmap index uses an individual bit for each potential value with the bit either “on” (set to 1) to indicate that the row contains the value or “off” (set to 0) to indicate that the row does not contain the value. This storage mechanism can also provide performance improvements for the types of joins typically used in data warehousing. Bitmap indexes are described in more detail in Chapter 4.

### Star query optimization

Typical data warehousing queries occur against a large *fact table* with foreign keys to much smaller *dimension tables*. Oracle added an optimization for this type of *star query* to Oracle 7.3. (See Figure 9-2 for an illustration of a typical star schema.) Performance gains are realized through the use of Cartesian product joins of dimension tables with a single join back to the large fact table. Oracle8 introduced a further mechanism called a *parallel bitmap star join*, which uses bitmap indexes on the foreign keys to the dimension tables to speed star joins involving a large number of dimension tables.

### Materialized views

Since Oracle8i, materialized views have provided another means of achieving a significant speed-up of query performance. Summary-level information derived from a fact table and grouped along dimension values is stored as a materialized view. Queries that can use this view are directed to the view, transparently to the user and the SQL they submit.

## **Analytic functions**

A growing trend in Oracle and other systems is the movement of some functions from decision-support user tools into the database. Oracle8i and Oracle9i releases featured the addition of ANSI standard OLAP SQL analytic functions for windowing, statistics, CUBE and ROLLUP, and much more. Oracle Database 10g further adds to this SQL library of analytic functions and statistics in the database.

## **OLAP Option**

Introduced in Oracle9i, OLAP services in the OLAP Option provide a Java OLAP API and are typically leveraged to build custom OLAP applications through the use of Oracle's JDeveloper product. The OLAP Option may also be accessed via SQL.

## **Data Mining Option**

Since Oracle9i, popular data-mining algorithms have been embedded in the database through the Data Mining Option and are exposed through a Java data-mining API. Data mining applications are typically custom built using Oracle's JDeveloper with DM4J.

## **Availability**

Oracle Standard Edition lacks important data warehousing features available in the Enterprise Edition, such as bitmap indexes and many parallelization features. Enterprise Edition is recommended for data warehousing projects.

# **Database Management Features**

Oracle includes many features that make the database easier to manage. We've divided the discussion in this section into four categories: Oracle Enterprise Manager, add-on packs, backup and recovery, and database availability.

## **Oracle Enterprise Manager**

As part of every Database Server, Oracle provides the Oracle Enterprise Manager (EM), a database management tool framework with a graphical interface used to manage database users, instances, and features (such as replication) that can provide additional information about the Oracle environment. EM can also manage Oracle's Application Server, Collaboration Suite, and E-Business Suite.

Prior to the Oracle8i database, the EM software was installed on Windows-based systems, each repository accessible by only a single database manager at a time. EM evolved to a Java release providing access from a browser or Windows-based system. Multiple database administrators could then access the EM repository at the same time.

More recently, an EM HTML console was released with Oracle9iAS with important new application performance management and configuration management features. The HTML version supplemented the Java-based Enterprise Manager earlier available. Enterprise Manager 10g, released with Oracle Database 10g, also comes in Java and HTML versions. EM can be deployed in several ways: as a central console for monitoring multiple databases leveraging agents, as a “product console” (easily installed with each individual database), or through remote access, also known as “studio mode.” The HTML-based console includes advanced management capabilities for rapid installation, deployment across grids of computers, provisioning, upgrades, and automated patching.

Oracle Enterprise Manager 10g has several additional options (sometimes called *packs*) for managing the Oracle Enterprise Edition database. These options, which are available for the HTML-based console, the Java-based console, or both, include:

- Database Diagnostics option
- Application Server Diagnostics option
- Database Tuning option
- Database Change Management option
- Database Configuration Management option
- Application Server Configuration Management option

Standard management pack functionality for managing the Standard Edition is now also available for the HTML-based console.

## **EM2Go**

EM2Go is a mobile version of Enterprise Manager that provides a subset of the functionality in EM. EM2Go is accessed through a Pocket PC browser on a PDA device. Database management capabilities include alert notification and viewing, job creation and scheduling, database performance monitoring and status (including Top Sessions, locks, and SQL assessment), basic storage administration, tablespace operations management, and system configuration status. EM2Go also monitors availability and performance of Oracle Application Servers.

## **Availability**

While both the Java- and HTML-based consoles of Enterprise Manager 10g can be used for managing Oracle, functionality is different in the two consoles. We discuss this further in Chapter 5. Oracle Application Server, Oracle E-Business Suite, and Oracle Collaboration Suite are managed through the HTML-based console.



## Backup and Recovery

As every database administrator knows, backing up a database is a rather mundane but necessary task. An improper backup makes recovery difficult, if not impossible. Unfortunately, people often realize the extreme importance of this everyday task only when it is too late—usually after losing business-critical data due to a failure of a related system.

The following sections describe some products and techniques for performing database backup operations. We discuss backup and recovery strategies and options in much greater detail in Chapter 10.

### Recovery Manager

Typical backups include complete database backups (the most common type), tablespace backups, datafile backups, control file backups, and archived redo log backups. Oracle8 introduced the Recovery Manager (RMAN) for the server-managed backup and recovery of the database. Previously, Oracle's Enterprise Backup Utility (EBU) provided a similar solution on some platforms. However, RMAN, with its Recovery Catalog stored in an Oracle database, provides a much more complete solution. RMAN can automatically locate, back up, restore, and recover datafiles, control files, and archived redo logs. RMAN, since Oracle9i, can restart backups and restores and implement recovery window policies when backups expire. The Oracle Enterprise Manager Backup Manager provides a GUI-based interface to RMAN. Oracle Enterprise Manager 10g introduces a new improved job scheduler that can be used with RMAN and other scripts, and that can manage automatic backups to disk.

### Incremental backup and recovery

RMAN can perform incremental backups of Enterprise Edition databases. Incremental backups back up only the blocks modified since the last backup of a datafile, tablespace, or database; thus, they're smaller and faster than complete backups. RMAN can also perform point-in-time recovery, which allows the recovery of data until just prior to an undesirable event (such as the mistaken dropping of a table).

### Oracle Storage Manager and Automated Disk Based Backup and Recovery

Various media-management software vendors support RMAN. Since Oracle8i, a Storage Manager has come bundled with Oracle to provide media-management services, including the tracking of tape volumes, for up to four devices. RMAN interfaces automatically with the media-management software to request the mounting of tapes as needed for backup and recovery operations.

Oracle Database 10g introduces Automated Disk Based Backup and Recovery. The disk acts as a cache, and archives and backups can then be copied to tape. The disk "cache" can also serve as a staging area for recovery.

## Availability

Backup and recovery facilities are available for both Oracle Standard Edition and Enterprise Edition.

## Database Availability

Database availability depends upon the reliability and management of the database, the underlying operating system, and the specific hardware components of the system. Oracle has improved availability by reducing backup and recovery times. It has done this through:

- Providing online and parallel backup and recovery
- Improving the management of online data through range partitioning
- Leveraging hardware capabilities for improved monitoring and failover

The relevant features are described in the following sections.

### Partitioning option

Oracle introduced partitioning as an option with Oracle8 to provide a higher degree of manageability and availability. You can take individual partitions offline for maintenance while other partitions remain available for user access. In data warehousing implementations, partitioning is frequently used to implement rolling windows based on date ranges. Hash partitioning, in which the data partitions are divided up as a result of a hashing function, was added to Oracle8i to enable an even distribution of data. You can also use composite partitioning to enable hash subpartitioning within specific range partitions. Oracle9i added list partitioning, which enables the partitioning of data based on discrete values such as geography, and composite range-list.

### Data Guard

Oracle first introduced a standby database feature in Oracle 7.3. The standby database provides a copy of the production database to be used if the primary database is lost—for example, in the event of primary site failure, or during routine maintenance. Primary and standby databases may be geographically separated. The standby database is created from a copy of the production database and updated through the application of archived redo logs generated by the production database. Data Guard, first introduced in Oracle9i, fully automates this process; previously, you had to manually copy and apply the logs. Agents are deployed on both the production and standby database, and a Data Guard Broker coordinates commands. A single Data Guard command is used to run the eight steps required for failover.

In addition to providing physical standby database support, Data Guard can create a logical standby database. In this scenario, Oracle archive redo logs are transformed into SQL transactions and applied to an open standby database.

Oracle Database 10g introduces several new features, including support for real-time application of redo data, integration with the Flashback database feature, and archive log compression. From Oracle Database 10g forward, rolling upgrades are supported.

## **Fail Safe**

The Fail Safe feature provides a higher level of reliability for an Oracle database. Failover is implemented through a second system or node that provides access to data residing on a shared disk when the first system or node fails. Oracle Fail Safe for Windows, in combination with Microsoft Cluster Services, provides a failover solution in the event of a system failure.

Fail Safe is primarily a disaster recovery tool, so some downtime does occur as part of a failover operation. The recommended solution for server availability has been Real Application Clusters since Oracle9i.

## **Oracle Real Application Clusters**

The Oracle Parallel Server (OPS) option was replaced by Real Application Clusters (RAC) beginning with Oracle9i. RAC can provide failover support as well as increased scalability on Unix, Linux, and Windows clusters. Oracle8i OPS greatly improved scalability for read/write applications through the introduction of Cache Fusion. Oracle9i RAC provided Cache Fusion for write/write applications by greatly minimizing much of the disk write activity used to control data locking. Oracle Database 10g introduces a new level of RAC portability and Oracle support by providing integrated “clusterware” for the supported RAC platforms.

With Real Application Clusters, you can run multiple Oracle instances on systems in a shared disk cluster configuration or on multiple nodes of a Massively Parallel Processor (MPP) configuration. RAC is also a key enabler for grid computing. RAC coordinates traffic among the systems or nodes, allowing the instances to function as a single database. As a result, the database can scale across dozens of nodes. Because the cluster provides a means by which multiple instances can access the same data, the failure of a single instance will not cause extensive delays while the system recovers; you can simply redirect users to another instance that’s still operating.

## **Data Guard and RAC**

Data Guard and RAC in combination replaced Parallel Fail Safe beginning with Oracle9i. Data Guard provides automated failover with bounded recovery time in conjunction with Oracle Real Application Clusters. In addition, it provides client

rerouting from the failed instance to the instance that is available with fast reconnect, and automatically captures diagnostic data. With Oracle Database 10g and Enterprise Manager Version 4, it is possible to manage Data Guard configurations containing RAC primary and secondary databases implemented with Oracle Database 10g's integrated clusterware.

### **Automated Storage Management**

Oracle Database 10g introduces Automated Storage Management (ASM), which provides optimum striping and mirroring of data for performance and availability. Because disks can now be optimally tuned via the database, ASM in Oracle Database 10g greatly simplifies this critical management task.

### **Availability**

Advanced high-availability features such as the Partitioning Option and Real Application Clusters have typically been available for Oracle Enterprise Edition but not for Standard Edition. Oracle Data Guard is bundled with the Enterprise Edition.

## **Oracle Developer Suite**

Many Oracle tools are available to developers to help them present data and build more sophisticated Oracle database applications. Although this book focuses on the Oracle database, this section briefly describes the main Oracle tools for application development: Oracle Forms Developer, Oracle Reports Developer, Oracle Designer, Oracle JDeveloper, Oracle Discoverer Administrative Edition and Oracle Portal. Oracle Developer Suite was known as Oracle Internet Developer Suite with Oracle9i.

### **Oracle Forms Developer**

Oracle Forms Developer provides a powerful tool for building forms-based applications and charts for deployment as traditional client/server applications or as three-tier browser-based applications via Oracle Application Server. Developer is a fourth-generation language (4GL). With a 4GL, you define applications by defining values for properties, rather than by writing procedural code. Developer supports a wide variety of clients, including traditional client/server PCs and Java-based clients. The Forms Builder includes a built-in JVM for previewing web applications.

### **Oracle Reports Developer**

Oracle Reports Developer provides a development and deployment environment for rapidly building and publishing web-based reports via Reports for Oracle's Application Server. Data can be formatted in tables, matrices, group reports, graphs, and combinations. High-quality presentation is possible using the HTML extension Cascading Style Sheets (CSS).

## Oracle JDeveloper

Oracle JDeveloper was introduced by Oracle in 1998 to develop basic Java applications without writing code. JDeveloper includes a Data Form wizard, a Beans Express wizard for creating JavaBeans and BeanInfo classes, and a Deployment wizard. JDeveloper includes database development features such as various Oracle drivers, a Connection Editor to hide the JDBC API complexity, database components to bind visual controls, and a SQLJ precompiler for embedding SQL in Java code, which you can then use with Oracle. You can also deploy applications developed with JDeveloper using the Oracle Application Server. Although JDeveloper uses wizards to allow programmers to create Java objects without writing code, the end result is generated Java code. This Java implementation makes the code highly flexible, but it is typically a less productive development environment than a true 4GL.

## Oracle Designer

Oracle Designer provides a graphical interface for Rapid Application Development (RAD) for the entire database development process—from building the business model to schema design, generation, and deployment. Designs and changes are stored in a multiuser repository. The tool can reverse-engineer existing tables and database schemas for reuse and redesign from Oracle and non-Oracle relational databases.

Designer also includes generators for creating applications for Oracle Developer, HTML clients using Oracle's Application Server, and C++. Designer can generate applications and reverse-engineer existing applications or applications that have been modified by developers. This capability enables a process called *round-trip engineering*, in which a developer uses Designer to generate an application, modifies the generated application, and reverse-engineers the changes back into the Designer repository.

## Oracle Discoverer Administration Edition

Oracle Discoverer Administration Edition enables administrators to set up and maintain the Discoverer End User Layer (EUL). The purpose of this layer is to shield business analysts using Discoverer as an ad hoc query or ROLAP tool from SQL complexity. Wizards guide the administrator through the process of building the EUL. In addition, administrators can put limits on resources available to analysts monitored by the Discoverer query governor.

## Oracle Portal

Oracle Portal, introduced as WebDB in 1999, provides an HTML-based tool for developing web-enabled applications and content-driven web sites. Portal application systems are developed and deployed in a simple browser environment. Portal includes wizards for developing application components incorporating “servlets” and

access to other HTTP web sites. For example, Oracle Reports and Discoverer may be accessed as servlets. Portals can be designed to be user-customizable. They are deployed to the middle-tier Oracle Application Server.

Oracle Portal brought a key enhancement to WebDB, the ability to create and use *portlets*, which allow a single web page to be divided up into different areas that can independently display information and interact with the user.

### **Availability**

All of these pieces are bundled in the Oracle Developer Suite. The Oracle Developer Suite is an additional cost option.

## **Oracle Lite**

Oracle Lite is Oracle's suite of products for enabling mobile use of database-centric applications. Key components of Oracle Lite include the Oracle Lite Database, Mobile Development Kit, and Mobile Server (an extension of the Oracle Application Server).

Although the Oracle Lite Database engine runs on a much smaller platform than other Oracle implementations (it requires a 50K to 1 MB footprint depending on the platform), Mobile SQL, C++, and Java-based applications can run against the database. ODBC is supported. Java support includes Java stored procedures and JDBC. The database is self-tuning and self-administering. In addition to Windows-based laptops, Oracle Lite is also supported for handheld devices running WindowsCE, Palm's Computing Platform, and Symbian EPOC.

In typical usage of Oracle Lite, the user will link her handheld or mobile device running the Oracle Lite Database to a large footprint Oracle Database Server. Data will be synchronized between the two systems. The user will then remove the link and work in disconnected mode. After she has performed her tasks, she'll relink and resynchronize the data with the Oracle Database Server.

The variety of synchronization capabilities include the following:

- Bidirectional synchronization between the mobile device and Oracle's larger footprint databases
- Publish-and-subscribe-based models
- Support for protocols such as TCP/IP, HTTP, CDPD, 802.1, and HotSync

You can define priority-based replication of subsets of data. Because data distributed to multiple locations can lead to conflicts—such as which location now has the “true” version of the data—automated conflict and resolution is provided. You can also customize the conflict resolution.

The Mobile Server provides a single platform for publishing, deploying, synchronizing, and managing mobile applications. The web-based control center can be used for controlling access to mobile applications. Oracle’s former “Web-to-Go” product is also part of the Mobile Server and provides centralized wizard-based application development and deployment.