

I Introduction

ORACLE

Copyright © 2009, Oracle. All rights reserved.

PATITAPABAN PARIDA (pppparida9@gmail.com) has a
non-transferable license to use this Student Guide.

Lesson Objectives

After completing this lesson, you should be able to do the following:

- List the features of Oracle10g
- Discuss the theoretical and physical aspects of a relational database
- Describe the Oracle implementation of RDBMS and ORDBMS
- Understand the goals of the course

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, is positioned on the right side of a red horizontal bar.

Copyright © 2009, Oracle. All rights reserved.

Objectives

In this lesson, you gain an understanding of the relational database management system (RDBMS) and the object relational database management system (ORDBMS). You are also introduced to the following:

- SQL statements that are specific to Oracle
- SQL Developer, which is an environment used for executing SQL statements and for formatting and reporting purposes

Goals of the Course

After completing this course, you should be able to do the following:

- Identify the major structural components of Oracle Database 10g
- Retrieve row and column data from tables with the `SELECT` statement
- Create reports of sorted and restricted data
- Employ SQL functions to generate and retrieve customized data
- Run data manipulation language (DML) statements to update data in Oracle Database 10g
- Obtain metadata by querying the dictionary views

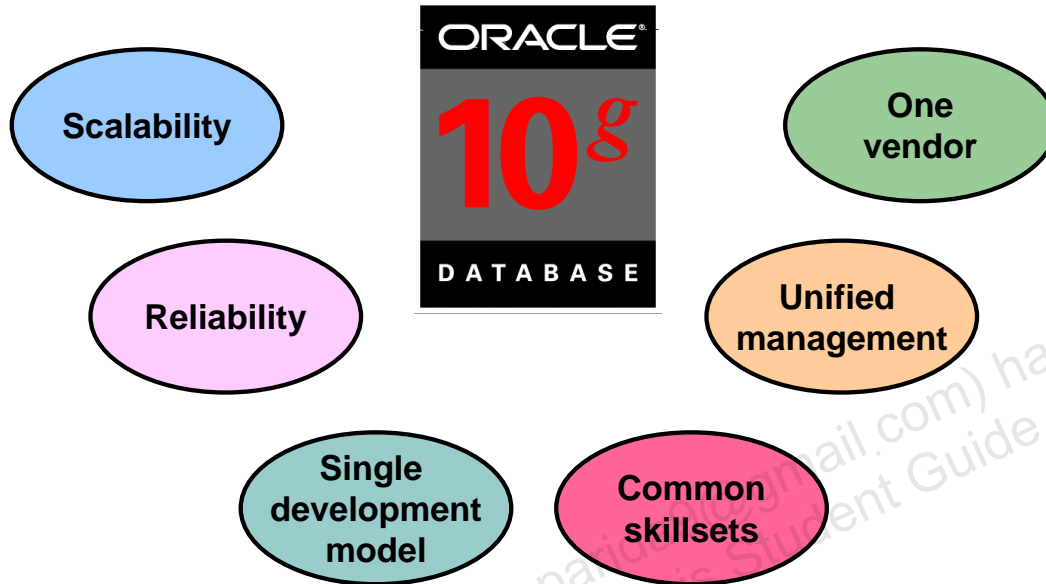
The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, is positioned on the right side of a red horizontal bar.

Copyright © 2009, Oracle. All rights reserved.

Goals of the Course

This course offers you an introduction to Oracle Database 10g database technology. In this class, you learn the basic concepts of relational databases and the powerful SQL programming language. This course provides the essential SQL skills that enable you to write queries against single and multiple tables, manipulate data in tables, create database objects, and query metadata.

Oracle10g



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Oracle10g Features

The Oracle10g release offers a comprehensive high-performance infrastructure, including:

- Scalability from departments to enterprise e-business sites
- Robust, reliable, available, and secure architecture
- One development model; easy deployment options
- Leverage an organization's current skillset throughout the Oracle platform (including SQL, PL/SQL, Java, and XML)
- One management interface for all applications
- Industry-standard technologies; no proprietary lock-in

In addition to providing the benefits listed above, the Oracle10g release contains the database for the grid. Grid computing can dramatically lower the cost of computing, extend the availability of computing resources, and deliver higher productivity and quality.

The basic idea of grid computing is the notion of computing as a utility, analogous to the electric power grid or the telephone network. As a client of the grid, you do not care where your data is or where your computation is done. You want to have your computation done and to have your information delivered to you when you want it. From the server side, grid is about virtualization and provisioning. You pool all your resources together and provision these resources dynamically based on the needs of your business, thus achieving better resource utilization at the same time.

Oracle10g



ORACLE

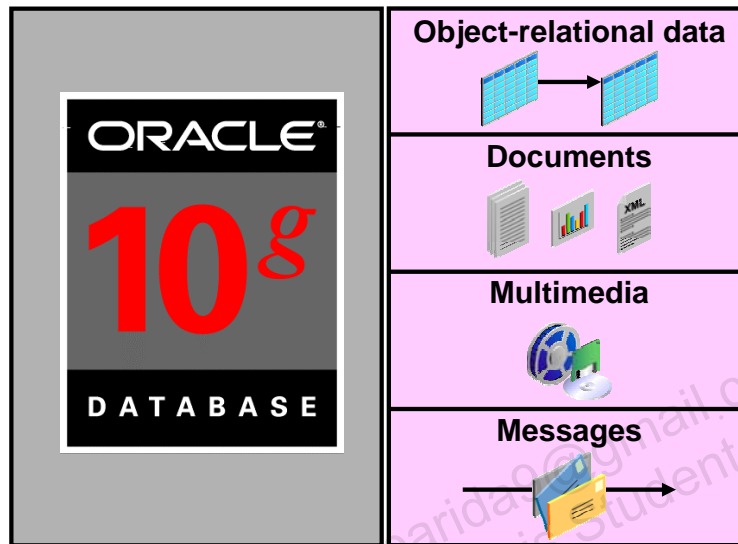
Copyright © 2009, Oracle. All rights reserved.

Oracle10g

The three grid-infrastructure products of the Oracle10g release are:

- Oracle Database 10g
- Oracle Application Server 10g
- Oracle Enterprise Manager 10g Grid Control

Oracle Database 10g



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Oracle Database 10g

Oracle Database 10g is designed to store and manage enterprise information. Oracle Database 10g cuts management costs and provides a high quality of service. Reduced configuration and management requirements and automatic SQL tuning have significantly reduced the cost of maintaining the environment.

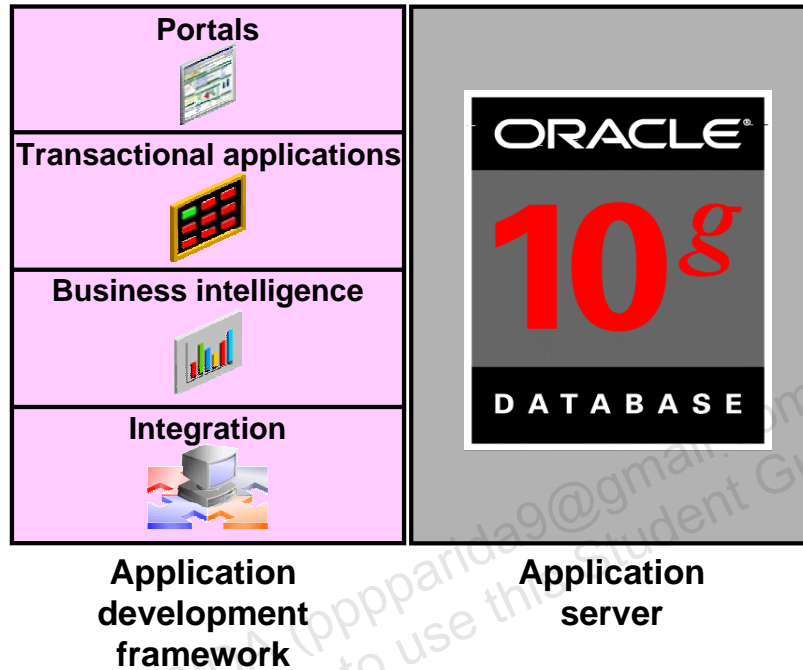
Oracle Database 10g contributes to the grid-infrastructure products of the Oracle 10g release. Grid computing is all about computing as a utility. If you are a client, you need not know where your data resides and which computer stores it. You should be able to request information or computation on your data and have it delivered to you.

Oracle Database 10g manages all your data. This is not just the object-relational data that you expect an enterprise database to manage. It can also be unstructured data such as:

- Spreadsheets
- Word documents
- PowerPoint presentations
- XML
- Multimedia data types like MP3, graphics, video, and more

The data does not even have to be in the database. Oracle Database 10g has services through which you can store metadata about information stored in file systems. You can use the database server to manage and serve information wherever it is located.

Oracle Application Server 10g



Oracle Application Server 10g

Oracle Application Server 10g provides a complete infrastructure platform for developing and deploying enterprise applications, integrating many functions including a J2EE and Web services run-time environment, an enterprise portal, an enterprise integration broker, business intelligence, Web caching, and identity management services.

Oracle Application Server 10g adds new grid computing features, building on the success of Oracle9i Application Server, which has hundreds of customers running production enterprise applications.

Oracle Application Server 10g is the only application server to include services for all the different server applications that you might want to run, including:

- Portals or Web sites
- Java transactional applications
- Business intelligence applications

It also provides integration among users, applications, and data throughout your organization.

Oracle Enterprise Manager 10g Grid Control

- Software provisioning
- Application service level monitoring



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Oracle Enterprise Manager 10g Grid Control

Oracle Enterprise Manager 10g Grid Control is the complete, integrated, central management console and underlying framework that automates administrative tasks across sets of systems in a grid environment. With Oracle Grid Control, you can group multiple hardware nodes, databases, application servers, and other targets into single logical entities. By executing jobs, enforcing standard policies, monitoring performance, and automating many other tasks across a group of targets instead of on many systems individually, Grid Control enables scaling with a growing grid.

Software Provisioning

With Grid Control, Oracle 10g automates installation, configuration, and cloning of Application Server 10g and Database 10g across multiples nodes. Oracle Enterprise Manager provides a common framework for software provisioning and management, enabling administrators to create, configure, deploy, and utilize new servers with new instances of the application server and database as they are needed.

Application Service Level Monitoring

Oracle Grid Control views the availability and performance of the grid infrastructure as a unified whole, as a user would experience it, rather than as isolated storage units, processing boxes, databases, and application servers.

Relational and Object Relational Database Management Systems

- Relational model and object relational model
- User-defined data types and objects
- Fully compatible with relational database
- Support of multimedia and large objects
- High-quality database server features

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, is positioned on the right side of a red horizontal bar.

Copyright © 2009, Oracle. All rights reserved.

About the Oracle Server

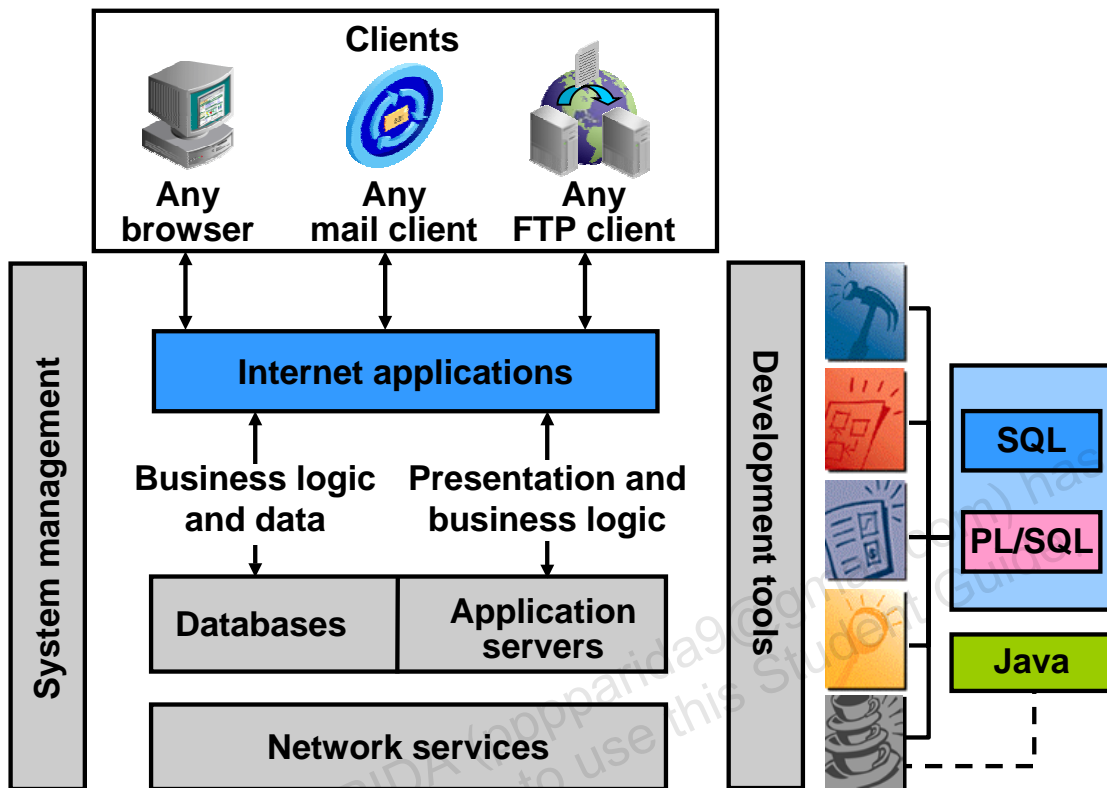
The Oracle server supports both the relational and object relational models.

The Oracle server extends the data-modeling capabilities to support an object relational database model that brings object-oriented programming, complex data types, complex business objects, and full compatibility with the relational world.

It includes several features for improved performance and functionality of online transaction processing (OLTP) applications, such as better sharing of run-time data structures, larger buffer caches, and deferrable constraints. Data warehouse applications benefit from enhancements such as parallel execution of insert, update, and delete operations; partitioning; and parallel-aware query optimization. Operating within the Network Computing Architecture (NCA) framework, the Oracle model supports client/server and Web-based applications that are distributed and multitiered.

For more information about the relational and object relational model, see the *Database Concepts* manual.

Oracle Internet Platform



Copyright © 2009, Oracle. All rights reserved.

Oracle Internet Platform

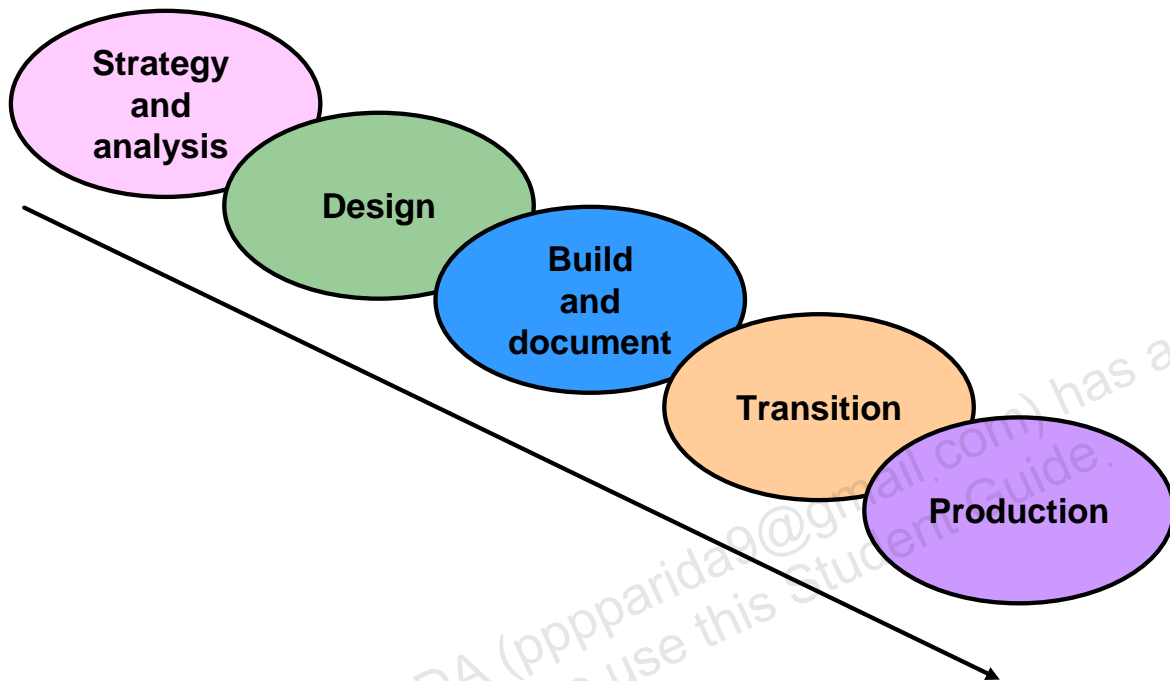
To develop an e-commerce application, you need a product that can store and manage the data, a product that can provide a run-time environment for your applications implementing business logic, and a product that can monitor and diagnose the application after it is integrated. The Oracle 10g products provide all the necessary components to develop your application.

Oracle offers a comprehensive high-performance Internet platform for e-commerce and data warehousing. The integrated Oracle Internet Platform includes everything needed to develop, deploy, and manage Internet applications, including these three core pieces:

- Browser-based clients to process presentation
- Application servers to execute business logic and serve presentation logic to browser-based clients
- Databases to execute database-intensive business logic and serve data

Oracle offers a wide variety of the most advanced graphical user interface (GUI)-driven development tools to build business applications, as well as a large suite of software applications for many areas of business and industry. Oracle Developer Suite includes tools to develop forms and reports and to build data warehouses. Stored procedures, functions, and packages can be written using SQL, PL/SQL, or Java.

System Development Life Cycle



ORACLE

Copyright © 2009, Oracle. All rights reserved.

System Development Life Cycle

From concept to production, you can develop a database by using the system-development life cycle, which contains multiple stages of development. This top-down, systematic approach to database development transforms business information requirements into an operational database.

Strategy and Analysis Phase

- Study and analyze the business requirements. Interview users and managers to identify the information requirements. Incorporate the enterprise and application mission statements as well as any future system specifications.
- Build models of the system. Transfer the business narrative into a graphical representation of business information needs and rules. Confirm and refine the model with the analysts and experts.

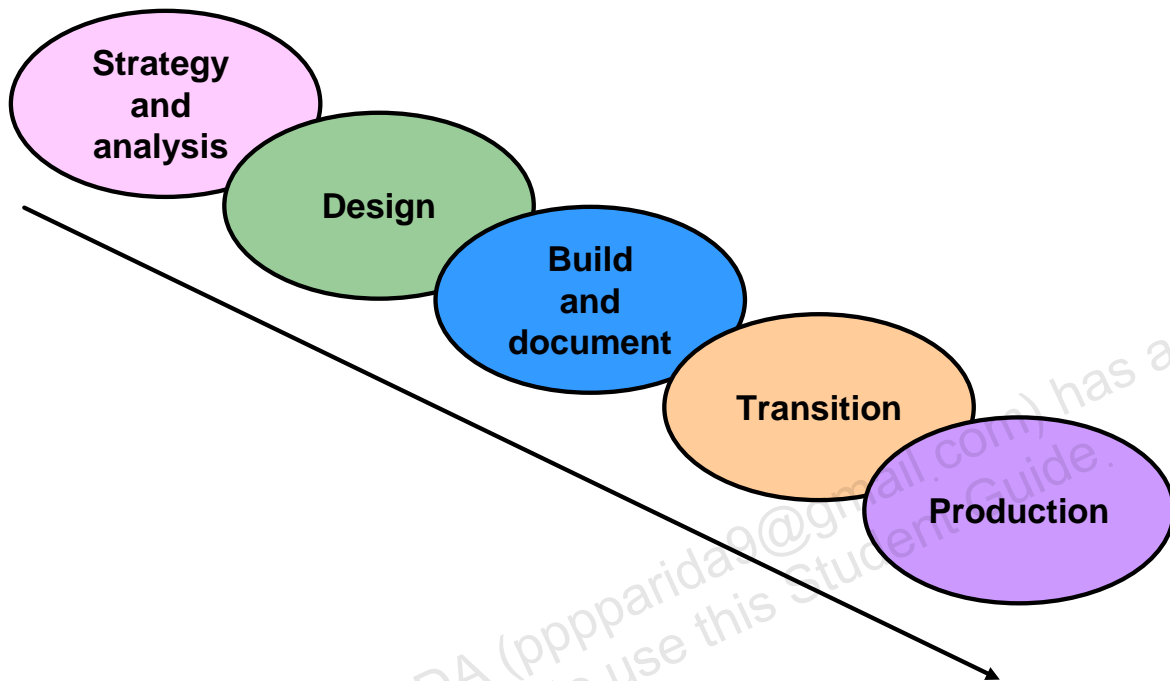
Design Phase

Design the database based on the model developed in the strategy and analysis phase.

Build and Documentation Phase

- Build the prototype system. Write and execute the commands to create the tables and supporting objects for the database.
- Develop user documentation, help text, and operations manuals to support the use and operation of the system.

System Development Life Cycle



Copyright © 2009, Oracle. All rights reserved.

System Development Life Cycle (continued)

Transition Phase

Refine the prototype. Move an application into production with user-acceptance testing, conversion of existing data, and parallel operations. Make any modifications required.

Production Phase

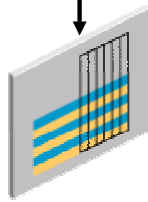
Roll out the system to the users. Operate the production system. Monitor its performance, and enhance and refine the system.

Note: The various phases of the system development life cycle can be carried out iteratively. This course focuses on the Build phase of the system development life cycle.

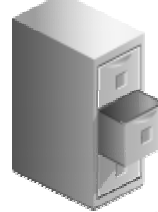
Data Storage on Different Media

| DEPARTMENT_ID | DEPARTMENT_NAME | MANAGER_ID | LOCATION_ID |
|---------------|-------------------|------------|-------------|
| 1 | 10 Administration | 200 | 1700 |
| 2 | 20 Marketing | 201 | 1800 |
| 3 | 50 Shipping | 124 | 1500 |
| 4 | 60 IT | 103 | 1400 |
| 5 | 80 Sales | 149 | 2500 |
| 6 | 90 Executive | 100 | 1700 |
| 7 | 110 Accounting | 205 | 1700 |
| 8 | 190 Contracting | (null) | 1700 |

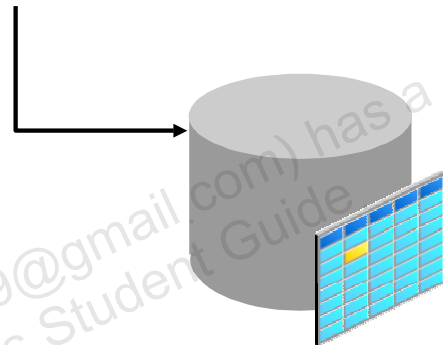
| GRADE_LEVEL | LOWEST_SAL | HIGHEST_SAL |
|-------------|------------|-------------|
| 1 A | 1000 | 2999 |
| 2 B | 3000 | 5999 |
| 3 C | 6000 | 9999 |
| 4 D | 10000 | 14999 |
| 5 E | 15000 | 24999 |
| 6 F | 25000 | 40000 |



Electronic spreadsheet



Filing cabinet



Database

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Storing Information

Every organization has some information needs. A library keeps a list of members, books, due dates, and fines. A company needs to save information about employees, departments, and salaries. These pieces of information are called *data*.

Organizations can store data on various media and in different formats, such as a hard-copy document in a filing cabinet or data stored in electronic spreadsheets or in databases.

A *database* is an organized collection of information.

To manage databases, you need a database management system (DBMS). A DBMS is a program that stores, retrieves, and modifies data in databases on request. There are four main types of databases: *hierarchical*, *network*, *relational*, and (most recently) *object relational*.

Relational Database Concept

- Dr. E. F. Codd proposed the relational model for database systems in 1970.
- It is the basis for the relational database management system (RDBMS).
- The relational model consists of the following:
 - Collection of objects or relations
 - Set of operators to act on the relations
 - Data integrity for accuracy and consistency

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Relational Model

The principles of the relational model were first outlined by Dr. E. F. Codd in a June 1970 paper titled “A Relational Model of Data for Large Shared Data Banks.” In this paper, Dr. Codd proposed the relational model for database systems.

The common models used at that time were hierarchical and network, or even simple flat-file data structures. Relational database management systems (RDBMS) soon became very popular, especially for their ease of use and flexibility in structure. In addition, a number of innovative vendors, such as Oracle, supplemented the RDBMS with a suite of powerful application development and user products, providing a total solution.

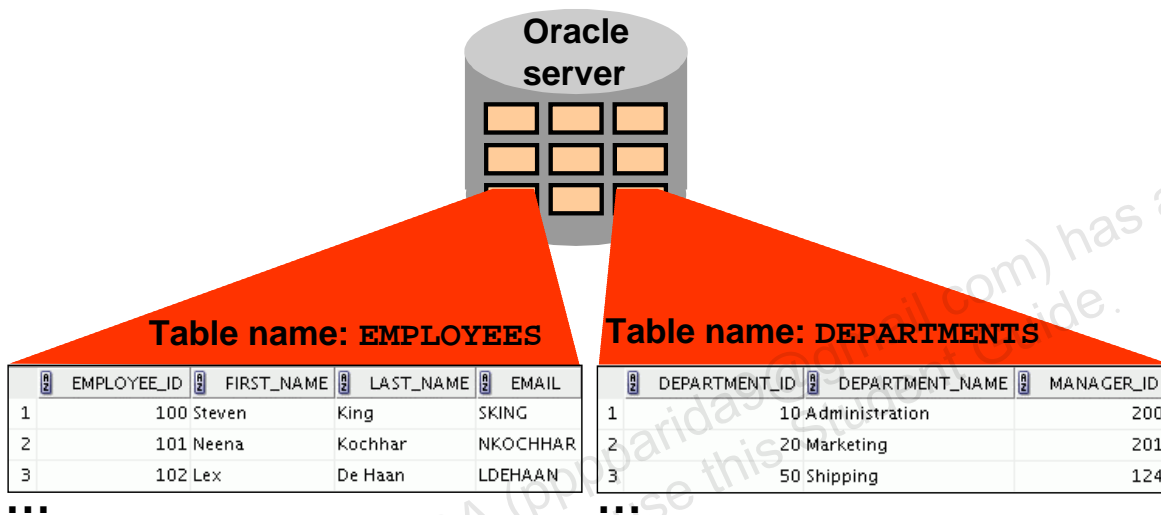
Components of the Relational Model

- Collections of objects or relations that store the data
- A set of operators that can act on the relations to produce other relations
- Data integrity for accuracy and consistency

For more information, see *An Introduction to Database Systems, Eighth Edition* (Addison-Wesley: 2004), written by Chris Date.

Definition of a Relational Database

A relational database is a collection of relations or two-dimensional tables.



ORACLE

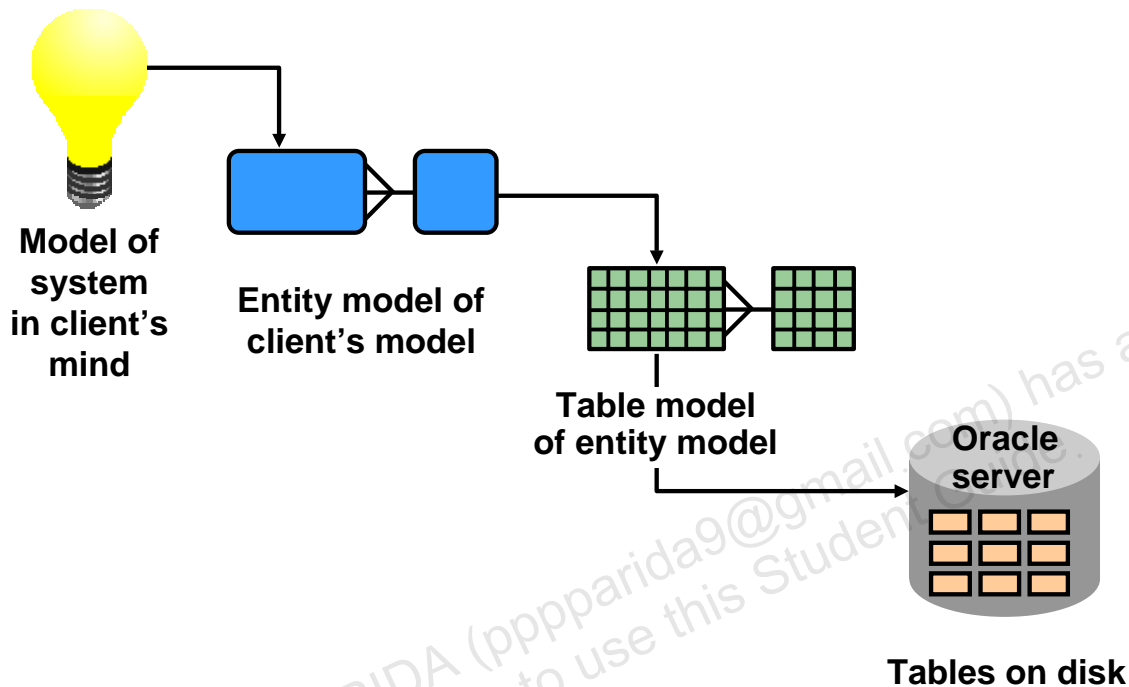
Copyright © 2009, Oracle. All rights reserved.

Definition of a Relational Database

A relational database uses relations or two-dimensional tables to store information.

For example, you might want to store information about all the employees in your company. In a relational database, you create several tables to store different pieces of information about your employees, such as an employee table, a department table, and a salary table.

Data Models



Copyright © 2009, Oracle. All rights reserved.

Data Models

Models are a cornerstone of design. Engineers build a model of a car to work out any details before putting it into production. In the same manner, system designers develop models to explore ideas and improve the understanding of database design.

Purpose of Models

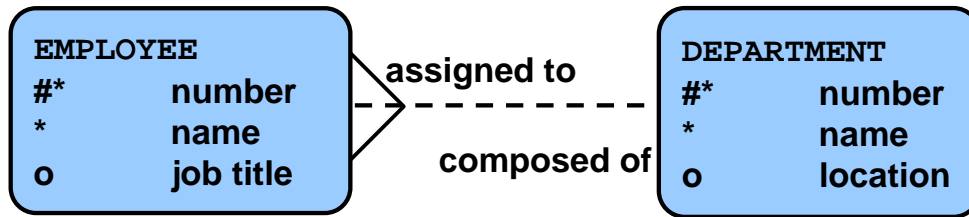
Models help communicate the concepts that are in people's minds. They can be used to do the following:

- Communicate
- Categorize
- Describe
- Specify
- Investigate
- Evolve
- Analyze
- Imitate

The objective is to produce a model that fits a multitude of these uses, can be understood by an end user, and contains sufficient detail for a developer to build a database system.

Entity Relationship Model

- Create an entity relationship diagram from business specifications or narratives:



- Scenario
 - "... Assign one or more employees to a department ..."
 - "... Some departments do not yet have assigned employees ..."

ORACLE

Copyright © 2009, Oracle. All rights reserved.

ER Modeling

In an effective system, data is divided into discrete categories or entities. An entity relationship (ER) model is an illustration of various entities in a business and the relationships among them. An ER model is derived from business specifications or narratives and built during the analysis phase of the system development life cycle. ER models separate the information required by a business from the activities performed within a business. Although businesses can change their activities, the type of information tends to remain constant. Therefore, the data structures also tend to be constant.

Benefits of ER Modeling

- Documents information for the organization in a clear, precise format
- Provides a clear picture of the scope of the information requirement
- Provides an easily understood pictorial map for database design
- Offers an effective framework for integrating multiple applications

ER Modeling (continued)

Key Components

- **Entity:** A thing of significance about which information needs to be known. Examples are departments, employees, and orders.
- **Attribute:** Something that describes or qualifies an entity. For example, for the employee entity, the attributes would be the employee number, name, job title, hire date, department number, and so on. Each of the attributes is either required or optional. This state is called *optionality*.
- **Relationship:** A named association between entities showing optionality and degree. Examples are employees and departments, and orders and items.

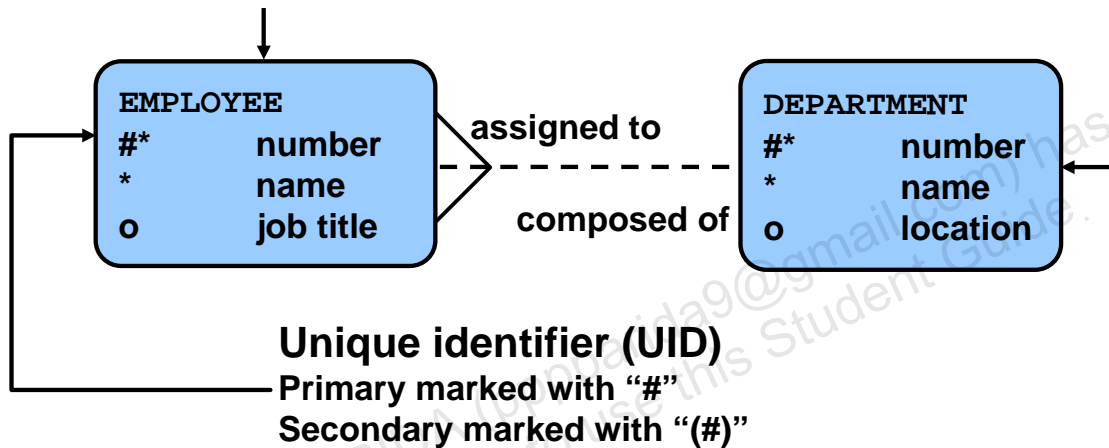
Entity Relationship Modeling Conventions

Entity

- Singular, unique name
- Uppercase
- Soft box
- Synonym in parentheses

Attribute

- Singular name
- Lowercase
- Mandatory marked with *
- Optional marked with “o”



ORACLE

Copyright © 2009, Oracle. All rights reserved.

ER Modeling Conventions

Entities

To represent an entity in a model, use the following conventions:

- Singular, unique entity name
- Entity name in uppercase
- Soft box
- Optional synonym names in uppercase within parentheses: ()

Attributes

To represent an attribute in a model, use the following conventions:

- Singular name in lowercase
- Asterisk (*) tag for mandatory attributes (that is, values that *must* be known)
- Letter “o” tag for optional attributes (that is, values that *may* be known)

Relationships

| Symbol | Description |
|-------------|--|
| Dashed line | Optional element indicating “maybe” |
| Solid line | Mandatory element indicating “must be” |
| Crow’s foot | Degree element indicating “one or more” |
| Single line | Degree element indicating “one and only one” |

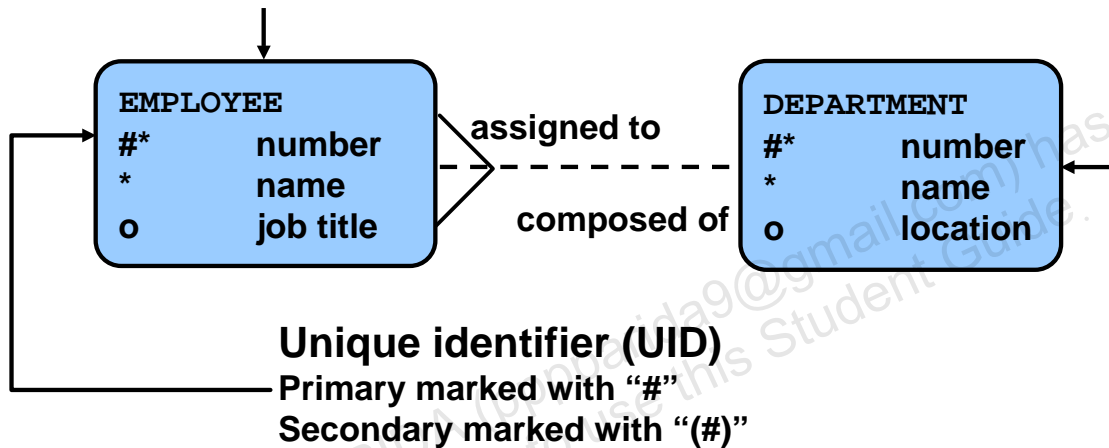
Entity Relationship Modeling Conventions

Entity

- Singular, unique name
- Uppercase
- Soft box
- Synonym in parentheses

Attribute

- Singular name
- Lowercase
- Mandatory marked with *
- Optional marked with "o"



ORACLE

Copyright © 2009, Oracle. All rights reserved.

ER Modeling Conventions (continued)

Relationships

Each direction of the relationship contains:

- **A label:** for example, *taught by* or *assigned to*
- **An optionality:** either *must be* or *maybe*
- **A degree:** either *one and only one* or *one or more*

Note: The term *cardinality* is a synonym for the term *degree*.

Each source entity {may be | must be} relationship name {one and only one | one or more} destination entity.

Note: The convention is to read clockwise.

Unique Identifiers

A unique identifier (UID) is any combination of attributes or relationships, or both, that serves to distinguish occurrences of an entity. Each entity occurrence must be uniquely identifiable.

- **Tag each attribute that is part of the UID with a number sign: #**
- **Tag secondary UIDs with a number sign in parentheses: (#)**

Relating Multiple Tables

- Each row of data in a table is uniquely identified by a primary key (PK).
- You can logically relate data from multiple tables using foreign keys (FK).

Table name: **EMPLOYEES**

| | EMPLOYEE_ID | FIRST_NAME | LAST_NAME | DEPARTMENT_ID |
|---|-------------|------------|-----------|---------------|
| 1 | 200 | Jennifer | Whalen | 10 |
| 2 | 201 | Michael | Hartstein | 20 |
| 3 | 202 | Pat | Fay | 20 |
| 4 | 205 | Shelley | Higgins | 110 |
| 5 | 206 | William | Gietz | 110 |

...

Primary key

Foreign key Primary key

Table name: **DEPARTMENTS**

| | DEPARTMENT_ID | DEPARTMENT_NAME | MANAGER_ID | LOCATION_ID |
|---|---------------|-----------------|------------|-------------|
| 1 | 10 | Administration | 200 | 1700 |
| 2 | 20 | Marketing | 201 | 1800 |
| 3 | 50 | Shipping | 124 | 1500 |
| 4 | 60 | IT | 103 | 1400 |
| 5 | 80 | Sales | 149 | 2500 |
| 6 | 90 | Executive | 100 | 1700 |
| 7 | 110 | Accounting | 205 | 1700 |
| 8 | 190 | Contracting | (null) | 1700 |

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Relating Multiple Tables

Each table contains data that describes exactly one entity. For example, the EMPLOYEES table contains information about employees. Categories of data are listed across the top of each table, and individual cases are listed below. Using a table format, you can readily visualize, understand, and use information.

Because data about different entities is stored in different tables, you may need to combine two or more tables to answer a particular question. For example, you may want to know the location of the department where an employee works. In this scenario, you need information from the EMPLOYEES table (which contains data about employees) and the DEPARTMENTS table (which contains information about departments). With an RDBMS, you can relate the data in one table to the data in another by using the foreign keys. A foreign key is a column (or a set of columns) that refers to a primary key in the same table or another table.

You can use the ability to relate data in one table to data in another to organize information in separate, manageable units. Employee data can be kept logically distinct from department data by storing it in a separate table.

Relating Multiple Tables (continued)

Guidelines for Primary Keys and Foreign Keys

- You cannot use duplicate values in a primary key.
- Primary keys generally cannot be changed.
- Foreign keys are based on data values and are purely logical (not physical) pointers.
- A foreign key value must match an existing primary key value or unique key value, or else it must be null.
- A foreign key must reference either a primary key or a unique key column.

Relational Database Terminology

| 2 | 3 | 4 | | | | |
|-------------|------------|-----------|-----------|----------------|---------------|--------|
| 1 | 2 | 3 | | | | |
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | COMMISSION_PCT | DEPARTMENT_ID | |
| 1 | 200 | Jennifer | Whalen | 4400 | (null) | 10 |
| 2 | 201 | Michael | Hartstein | 13000 | (null) | 20 |
| 3 | 202 | Pat | Fay | 6000 | (null) | 20 |
| 4 | 205 | Shelley | Higgins | 12000 | (null) | 110 |
| 5 | 206 | William | Gietz | 8300 | (null) | 110 |
| 6 | 100 | Steven | King | 24000 | (null) | 90 |
| 7 | 101 | Neena | Kochhar | 17000 | (null) | 90 |
| 8 | 102 | Lex | De Haan | 17000 | (null) | 90 |
| 9 | 103 | Alexander | Hunold | 9000 | (null) | 60 |
| 10 | 104 | Bruce | Ernst | 6000 | (null) | 60 |
| 11 | 107 | Diana | Lorentz | 4200 | (null) | 60 |
| 12 | 124 | Kevin | Mourgos | 5800 | (null) | 50 |
| 13 | 141 | Trenna | Rajs | 3500 | (null) | 50 |
| 14 | 142 | Curtis | Davies | 3100 | (null) | 50 |
| 15 | 143 | Randall | Matos | 2600 | (null) | 50 |
| 16 | 144 | Peter | Vargas | 2500 | (null) | 50 |
| 17 | 149 | Eleni | Zlotkey | 10500 | 0.2 | 80 |
| 18 | 174 | Ellen | Abel | 11000 | 0.3 | 80 |
| 19 | 176 | Jonathon | Taylor | 8600 | 0.2 | 80 |
| 20 | 178 | Kimberely | Grant | 7000 | 0.15 | (null) |

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Terminology Used in a Relational Database

A relational database can contain one or many tables. A *table* is the basic storage structure of an RDBMS. A table holds all the data necessary about something in the real world, such as employees, invoices, or customers.

The slide shows the contents of the *EMPLOYEES table* or *relation*. The numbers indicate the following:

1. A single *row* (or *tuple*) representing all data required for a particular employee. Each row in a table should be identified by a primary key, which permits no duplicate rows. The order of rows is insignificant; specify the row order when the data is retrieved.
2. A *column* or attribute containing the employee number. The employee number identifies a *unique* employee in the *EMPLOYEES* table. In this example, the employee number column is designated as the *primary key*. A primary key must contain a value, and the value must be unique.
3. A column that is not a key value. A column represents one kind of data in a table; in this example, the data is the salaries of all the employees. Column order is insignificant when storing data; specify the column order when the data is retrieved.

Terminology Used in a Relational Database (continued)

4. A column containing the department number, which is also a *foreign key*. A foreign key is a column that defines how tables relate to each other. A foreign key refers to a primary key or a unique key in the same table or in another table. In the example, DEPARTMENT_ID *uniquely* identifies a department in the DEPARTMENTS table.
5. A *field* can be found at the intersection of a row and a column. There can be only one value in it.
6. A field may have no value in it. This is called a null value. In the EMPLOYEES table, only those employees who have the role of sales representative have a value in the COMMISSION_PCT (commission) field.

Relational Database Properties

A relational database:

- Can be accessed and modified by executing structured query language (SQL) statements
- Contains a collection of tables with no physical pointers
- Uses a set of operators

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Properties of a Relational Database

In a relational database, you do not specify the access route to the tables, and you do not need to know how the data is arranged physically.

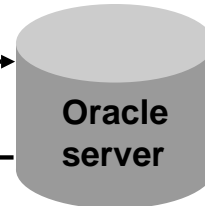
To access the database, you execute a structured query language (SQL) statement, which is the American National Standards Institute (ANSI) standard language for operating relational databases. The language contains a large set of operators for partitioning and combining relations. The database can be modified by using the SQL statements.

Communicating with an RDBMS Using SQL

SQL statement is entered.

```
SELECT department_name  
FROM departments;
```

**Statement is sent to
Oracle server.**



| DEPARTMENT_NAME |
|------------------|
| 1 Administration |
| 2 Marketing |
| 3 Shipping |
| 4 IT |
| 5 Sales |
| 6 Executive |
| 7 Accounting |
| 8 Contracting |

ORACLE

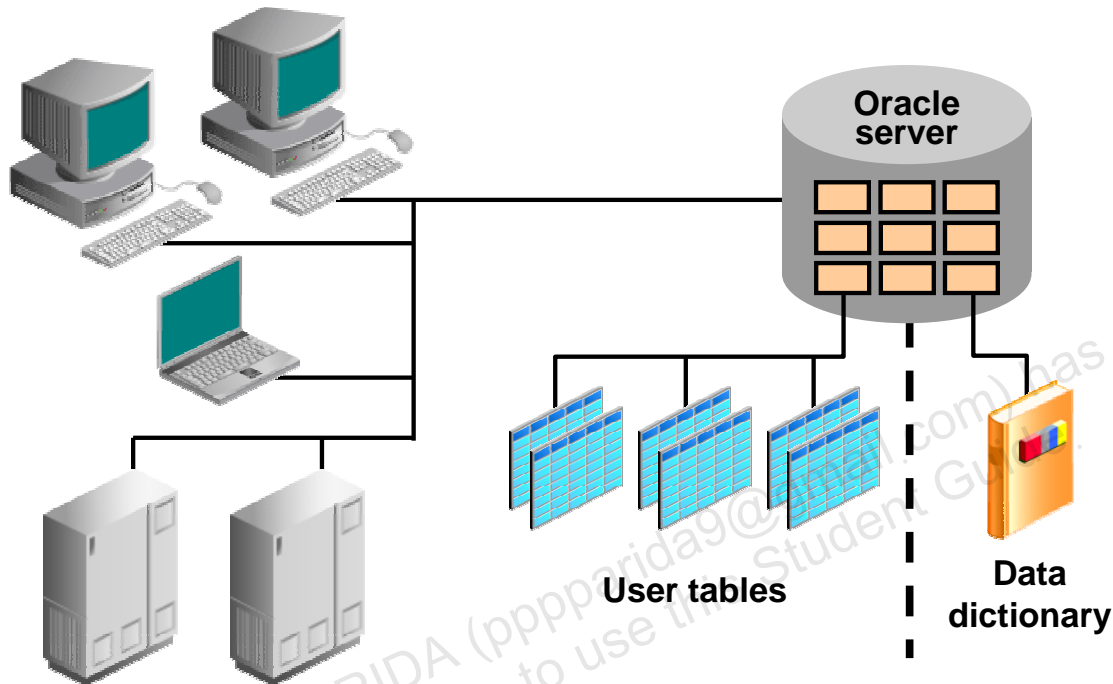
Copyright © 2009, Oracle. All rights reserved.

Structured Query Language

Using SQL, you can communicate with the Oracle server. SQL has the following advantages:

- Efficient
- Easy to learn and use
- Functionally complete (With SQL, you can define, retrieve, and manipulate data in the tables.)

Oracle's Relational Database Management System



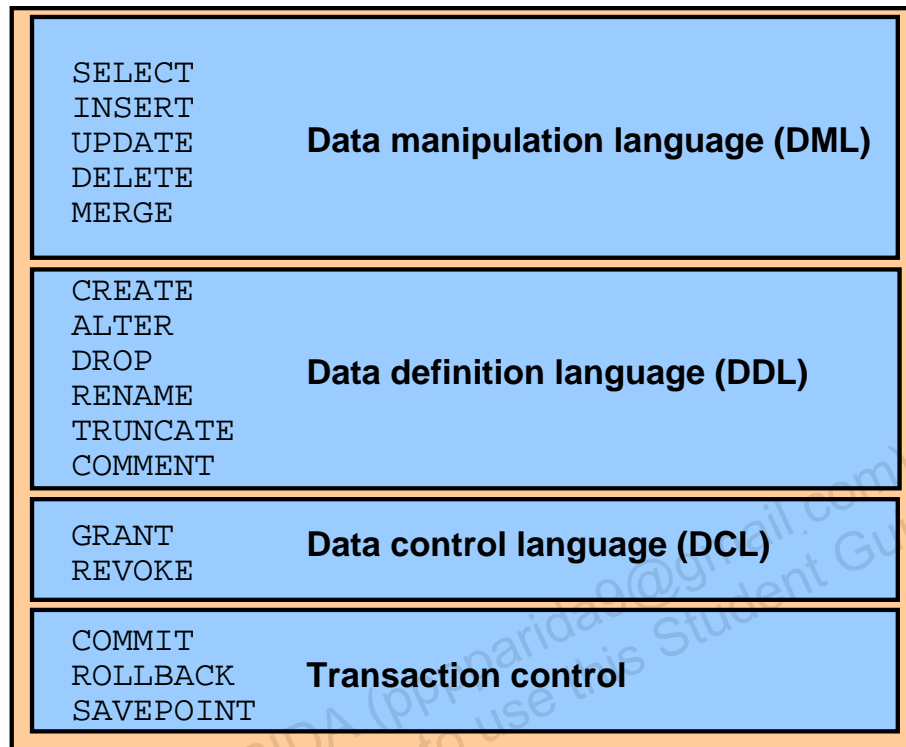
Copyright © 2009, Oracle. All rights reserved.

Oracle's Relational Database Management System

Oracle provides a flexible RDBMS called Oracle Database 10g. Using its features, you can store and manage data with all the advantages of a relational structure plus PL/SQL, an engine that provides you with the ability to store and execute program units. Oracle Database 10g also supports Java and XML. The Oracle server offers the options of retrieving data based on optimization techniques. It includes security features that control how a database is accessed and used. Other features include consistency and protection of data through locking mechanisms.

The Oracle10g release provides an open, comprehensive, and integrated approach to information management. An Oracle server consists of an Oracle Database and an Oracle server instance. Every time a database is started, a system global area (SGA) is allocated and Oracle background processes are started. The SGA is an area of memory that is used for database information shared by the database users. The combination of the background processes and memory buffers is called an Oracle *instance*.

SQL Statements



ORACLE

Copyright © 2009, Oracle. All rights reserved.

SQL Statements

Oracle SQL complies with industry-accepted standards. Oracle Corporation ensures future compliance with evolving standards by actively involving key personnel in SQL standards committees. Industry-accepted committees are American National Standards Institute (ANSI) and International Standards Organization (ISO). Both ANSI and ISO have accepted SQL as the standard language for relational databases.

| Statement | Description |
|--|---|
| SELECT INSERT UPDATE DELETE MERGE | Retrieves data from the database, enters new rows, changes existing rows, and removes unwanted rows from tables in the database, respectively. Collectively known as <i>data manipulation language</i> (DML). |
| CREATE ALTER DROP RENAME TRUNCATE COMMENT | Sets up, changes, and removes data structures from tables. Collectively known as <i>data definition language</i> (DDL). |
| GRANT REVOKE | Gives or removes access rights to both the Oracle database and the structures within it. |
| COMMIT ROLLBACK SAVEPOINT | Manages the changes made by DML statements. Changes to the data can be grouped together into logical transactions. |

Tables Used in the Course

EMPLOYEES

| | EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER | HIRE_DATE | JOB_ID | SALARY |
|----|-------------|------------|-----------|----------|--------------|-----------|------------|--------|
| 1 | 200 | Jennifer | Whalen | JWHALEN | 515.123.4444 | 17-SEP-87 | AD_ASST | 4400 |
| 2 | 201 | Michael | Hartstein | MHARTSTE | 515.123.5555 | 17-FEB-96 | MK_MAN | 13000 |
| 3 | 202 | Pat | Fay | PFAY | 603.123.6666 | 17-AUG-97 | MK_REP | 6000 |
| 4 | 205 | Shelley | Higgins | SHIGGINS | 515.123.8080 | 07-JUN-94 | AC_MGR | 12000 |
| 5 | 206 | William | Gietz | WGIEZT | 515.123.8181 | 07-JUN-94 | AC_ACCOUNT | 8300 |
| 6 | 100 | Steven | King | SKING | 515.123.4567 | 17-JUN-87 | AD_PRES | 24000 |
| 7 | 101 | Neena | Kochhar | NKOCHHAR | 515.123.4568 | 21-SEP-89 | AD_VP | 17000 |
| 8 | 102 | Lex | De Haan | LDEHAAN | 515.123.4569 | 13-JAN-93 | AD_VP | 17000 |
| 9 | 103 | Alexander | Hunold | AHUNOLD | 590.423.4567 | 03-JAN-90 | IT_PROG | 9000 |
| 10 | 104 | Bruce | Ernst | BERNST | 590.423.4568 | 21-MAY-91 | IT_PROG | 6000 |
| 11 | 107 | Diana | Lorentz | DLORENTZ | 590.423.5567 | 07-FEB-99 | IT_PROG | 4200 |
| 12 | 124 | Kevin | Mourgos | KMOURGOS | 650.123.5234 | 16-NOV-99 | ST_MAN | 5800 |
| 13 | 141 | Trenna | Rajs | TRAJS | 650.121.8009 | 17-OCT-95 | ST_CLERK | 3500 |
| 14 | 142 | Curtis | Davies | CDAVIES | 650.121.2994 | 29-JAN-97 | ST_CLERK | 3100 |
| 15 | 143 | Randall | Matos | RMATOS | 650.121.2874 | 15-MAR-98 | ST_CLERK | 2600 |
| 16 | 144 | Peter | Vargas | PVARGAS | 650.121.2004 | 09-JUL-98 | ST_CLERK | 2500 |

| | DEPARTMENT_ID | DEPARTMENT_NAME | MANAGER_ID | LOCATION_ID | | GRADE_LEVEL | LOWEST_SAL | HIGHEST_SAL |
|---|---------------|-----------------|------------|-------------|-------------|-------------|------------|-------------|
| 1 | 10 | Administration | 200 | 1700 | 1644.429018 | 1 A | 1000 | 2999 |
| 2 | 20 | Marketing | 201 | 1800 | 1644.429265 | 2 B | 3000 | 5999 |
| 3 | 50 | Shipping | 124 | 1500 | 1644.429263 | 3 C | 6000 | 9999 |
| 4 | 60 | IT | 103 | 1400 | | 4 D | 10000 | 14999 |
| 5 | 80 | Sales | 149 | 2500 | | 5 E | 15000 | 24999 |
| 6 | 90 | Executive | 100 | 1700 | | 6 F | 25000 | 40000 |
| 7 | 110 | Accounting | 205 | 1700 | | | | |
| 8 | 190 | Contracting | (null) | 1700 | | | | |

DEPARTMENTS

JOB_GRADES

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Tables Used in the Course

The following main tables are used in this course:

- EMPLOYEES table: Gives details of all the employees
- DEPARTMENTS table: Gives details of all the departments
- JOB_GRADES table: Gives details of salaries for various grades

Note: The structure and data for all the tables are provided in Appendix B.

Summary

- Oracle Database 10g is the database for grid computing.
- The database is based on the object relational database management system.
- Relational databases are composed of relations, managed by relational operations, and governed by data integrity constraints.
- With the Oracle server, you can store and manage information by using the SQL language and PL/SQL engine.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Summary

Relational database management systems are composed of objects or relations. They are managed by operations and governed by data integrity constraints.

Oracle Corporation produces products and services to meet your RDBMS needs. The main products are the following:

- Oracle Database 10g, with which you store and manage information by using SQL
- Oracle Application Server 10g, with which you run all of your applications
- Oracle Enterprise Manager 10g Grid Control, which you use to manage and automate administrative tasks across sets of systems in a grid environment

SQL

The Oracle server supports ANSI-standard SQL and contains extensions. SQL is the language that is used to communicate with the server to access, manipulate, and control data.