1

Introduction

Lesson Objectives

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

- Define the goals of the course
- List the features of Oracle Database 12*c*
- Describe the salient features of Oracle Cloud
- Discuss the theoretical and physical aspects of a relational database
- Describe Oracle server's implementation of RDBMS and object relational database management system (ORDBMS)
- Identify the development environments that can be used for this course
- Describe the database and schema used in this course

ORACLE

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 Oracle SQL Developer and SQL*Plus as development environments used for executing SQL statements, and for formatting and reporting purposes.

Lesson Agenda

- Course objectives, agenda, and appendixes used in the course
- Overview of Oracle Database 12*c* and related products
- Overview of relational database management concepts and terminologies
- Introduction to SQL and its development environments
- The Human Resource(HR) Schema and the tables used in the Course
- Oracle database 12*c* SQL Documentation and Additional Resources

ORACLE

Course Objectives

After completing this course, you should be able to:

- Identify the major components of Oracle Database
- 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 complex queries to retrieve data from multiple tables
- Run data manipulation language (DML) statements to update data in Oracle Database
- Run data definition language (DDL) statements to create and manage schema objects

ORACLE

This course offers you an introduction to the Oracle 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.

Course Agenda

- Day 1:
  - Introduction
  - Retrieving Data Using the SQL `SELECT` Statement
  - Restricting and Sorting Data
  - Using Single-Row Functions to Customize Output
- Day 2:
  - Using Conversion Functions and Conditional Expressions
  - Reporting Aggregated Data Using the Group Functions
  - Displaying Data from Multiple Tables Using Joins
  - Using Subqueries to Solve Queries

Course Agenda

- Day 3:
  - Using the Set Operators
  - Managing Tables Using DML Statements
  - Introduction to Data Definition Language

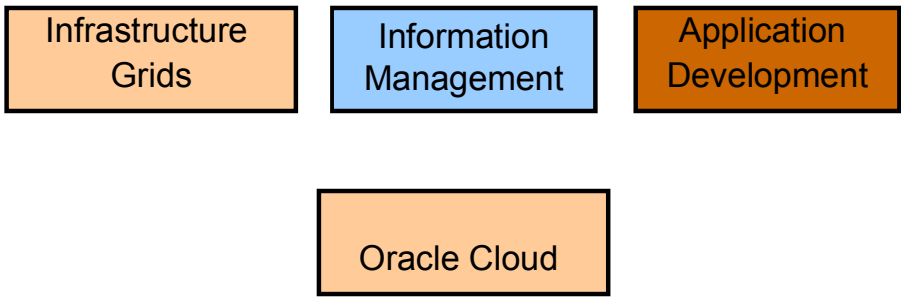
Appendices and Practices Used in the Course

- Appendix A: Table Descriptions
- Appendix B: Using SQL Developer
- Appendix C: Using SQL*Plus
- Appendix D: Commonly Used SQL Commands
- Activity Guide
  - Practices and Solutions
  - Additional Practices and Solutions

Lesson Agenda

- Course objectives, agenda, and appendixes used in the course
- Overview of Oracle Database 12*c* and related products
- Overview of relational database management concepts and terminologies
- Introduction to SQL and its development environments
- The Human Resource(HR) Schema and the tables used in this course
- Oracle database 12*c* SQL Documentation and Additional Resources

Oracle Database 12*c*: Focus Areas



Oracle Database 12*c* offers extensive features across the following focus areas:

- **Infrastructure Grids:** The Infrastructure Grid technology of Oracle enables pooling of low-cost servers and storage to form systems that deliver the highest quality of service in terms of manageability, high availability, and performance. Oracle Database 12*c* consolidates and extends the benefits of grid computing. Apart from taking full advantage of grid computing, Oracle Database 11*g* has unique change assurance features to manage changes in a controlled and cost effective manner.
- **Information Management:** Oracle Database 12*c* extends the existing information management capabilities in content management, information integration, and information life-cycle management areas. Oracle provides content management of advanced data types such as Extensible Markup Language (XML), text, spatial, multimedia, medical imaging, and semantic technologies.
- **Application Development:** Oracle Database 12*c* has capabilities to use and manage all the major application development environments such as PL/SQL, Java/JDBC, .NET and Windows, PHP, SQL Developer, and Application Express.
- **Oracle Cloud:** The Oracle Cloud is an enterprise cloud for business. It provides an integrated collection of application and platform cloud services that are based upon best in class products and open Java and SQL standards.

Oracle Database 12*c*

Manageability

High availability

Performance

Security

Information integration

Organizations need to support multiple terabytes of information for users who demand fast and secure access to business applications round the clock. The database systems must be reliable and must be able to recover quickly in the event of any kind of failure. Oracle Database 12*c* is designed along the following feature areas to help organizations manage infrastructure grids easily and deliver high-quality service:

- **Manageability:** By using some of the change assurance, management automation, and fault diagnostics features, the database administrators (DBAs) can increase their productivity, reduce costs, minimize errors, and maximize quality of service. Some of the useful features that promote better management are Database Replay facility, the SQL Performance Analyzer, and the Automatic SQL Tuning facility. Real-Time Database Operations Monitoring.
- **Enterprise Manager Database Express 12*c*** is a web-based tool for managing Oracle databases. Enterprise Manager Database Express greatly simplifies database performance diagnostics by consolidating the relevant database performance screens into a consolidated view called Database Performance Hub. DBAs get a single, consolidated view of the current real-time and historical view of the database performance across multiple dimensions such as database load, monitored SQL and PL/SQL, and Active Session History (ASH) in a single page for the selected time period.

- **High availability:** By using the high availability features, you can reduce the risk of down time and data loss. These features improve online operations and enable faster database upgrades.
- **Performance:** By using capabilities such as SecureFiles, compression for online transaction processing (OLTP), Real Application Clusters (RAC) optimizations, Result Caches, and so on, you can greatly improve the performance of your database. Oracle Database 12*c* enables organizations to manage large, scalable, transactional, and data warehousing systems that deliver fast data access using low-cost modular storage.
- **Security:** Oracle Database 12*c* helps organizations protect their information with unique secure configurations, data encryption and masking, and sophisticated auditing capabilities. It delivers a secure and scalable platform for reliable and fast access to all types of information by using the industry-standard interfaces.
- **Information integration:** Oracle Database 12*c* has many features to better integrate data throughout the enterprise. It also supports advanced information life-cycle management capabilities. This helps you manage the changing data in your database.

Oracle Fusion Middleware

Portfolio of leading, standards-based, and customer-proven software products that spans a range of tools and services from Java EE and developer tools, through integration services, business intelligence, collaboration, and content management

Oracle Fusion Middleware is a comprehensive and well-integrated family of products that offers complete support for development, deployment, and management of Service-Oriented Architecture (SOA). SOA facilitates the development of modular business services that can be easily integrated and reused, thereby reducing development and maintenance costs, and providing higher quality of services. Oracle Fusion Middleware's pluggable architecture enables you to leverage your investments in any existing application, system, or technology. Its unbreakable core technology minimizes the disruption caused by planned or unplanned outages.

Some of the products from the Oracle Fusion Middleware family include:

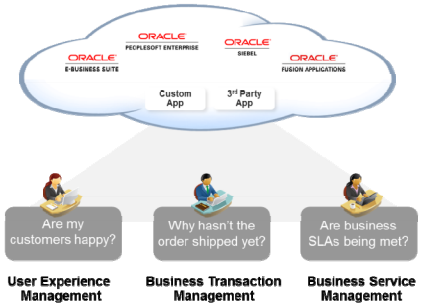
- **Application Server:** Java EE, Web Services
- **SOA and Process Management:** BPEL Process Manager, SOA Governance
- **Development Tools:** Oracle Application Development Framework, JDeveloper, SOA Suite
- **Business Intelligence:** Oracle Business Activity Monitoring, Oracle Data Integrator
- **Enterprise Management:** Enterprise Manager
- **Identity Management:** Oracle Identity Management
- **Content Management:** Oracle Content Database Suite
- **User Interaction:** Portal, Rich Internet Apps

Oracle Enterprise Manager Cloud Control

- Create and manage a complete set of cloud services.
- Manage all phases of cloud life cycle.
- Manage the entire cloud stack
- Monitor the health of all components
- Identify, understand, and resolve business problems

Complete life cycle

Complete stack



Complete integration

Self-Service IT I Simple and Automated I Business-Driven

ORACLE

Enterprise Manager Cloud Control is a management tool that provides monitoring and management capabilities for Oracle and non-Oracle components. It is a complete, integrated, and business-driven cloud management solution in a single product, which is referred to as "Total Cloud Control."

Using Enterprise Manager Cloud Control, you can:

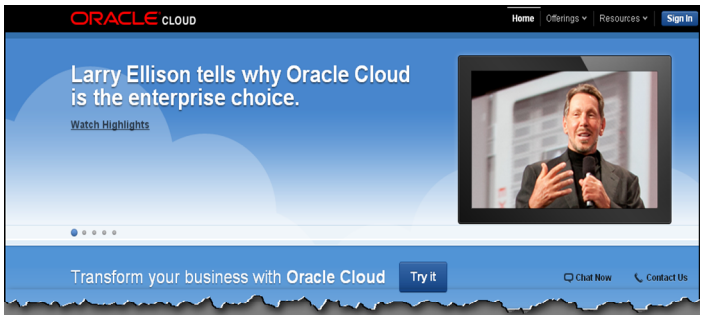
- Create and manage a complete set of cloud services, including: Infrastructure-as-a-service, Database-as-a-service, Platform-as-a-service, and others
- Manage all phases of cloud life cycle
- Manage the entire cloud stack: from application to disk, including engineered systems (Exa series) and with integrated support capabilities
- Monitor the health of all components, the hosts that they run on, and the key business processes that they support
- Identify, understand, and resolve business problems through the unified and correlated management of User Experience, Business Transactions, and Business Services across all your packaged and custom applications

Oracle Cloud

The Oracle Cloud is an enterprise cloud for business. It consists of many different services which share some common characteristics:

- On-demand self-service
- Resource pooling
- Rapid elasticity
- Measured service
- Broad network access

www.cloud.oracle.com



The Oracle Cloud is an enterprise cloud for business. It provides an integrated collection of application and platform cloud services that are based upon best-in-class products and open Java and SQL standards. The top two benefits of cloud computing are speed and cost.

As a result, the applications and databases deployed in the Oracle Cloud are portable and you can be easily moved them to or from a private cloud or on-premise environment.

- All Cloud Services can be provisioned through a self-service interface. Users can get their Cloud Services delivered on an integrated development and deployment platform with tools to rapidly extend and create new services.
- Oracle Cloud services are built on Oracle Exalogic Elastic Cloud and Oracle Exadata Database Machine, together offering a platform that delivers extreme performance, redundancy, and scalability.

Here are five essential characteristics of Oracle Cloud services:

- **On-demand self-service:** Provisioning, monitoring, and management control
- **Resource pooling:** Implies sharing and a level of abstraction between consumers and services
- **Rapid elasticity:** Ability to quickly scale up or down as needed
- **Measured service:** Metering utilization for either internal chargeback (private cloud) or external billing (public cloud)
- **Broad network access:** Access through a browser on any networked device

Oracle Cloud Services

Oracle Cloud provides three types of services:

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

ORACLE

(SaaS) generally refers to applications that are delivered to end users over the Internet. Oracle CRM On Demand is an example of a SaaS offering that provides both multitenant as well as single-tenant options, depending on the customer's preferences.

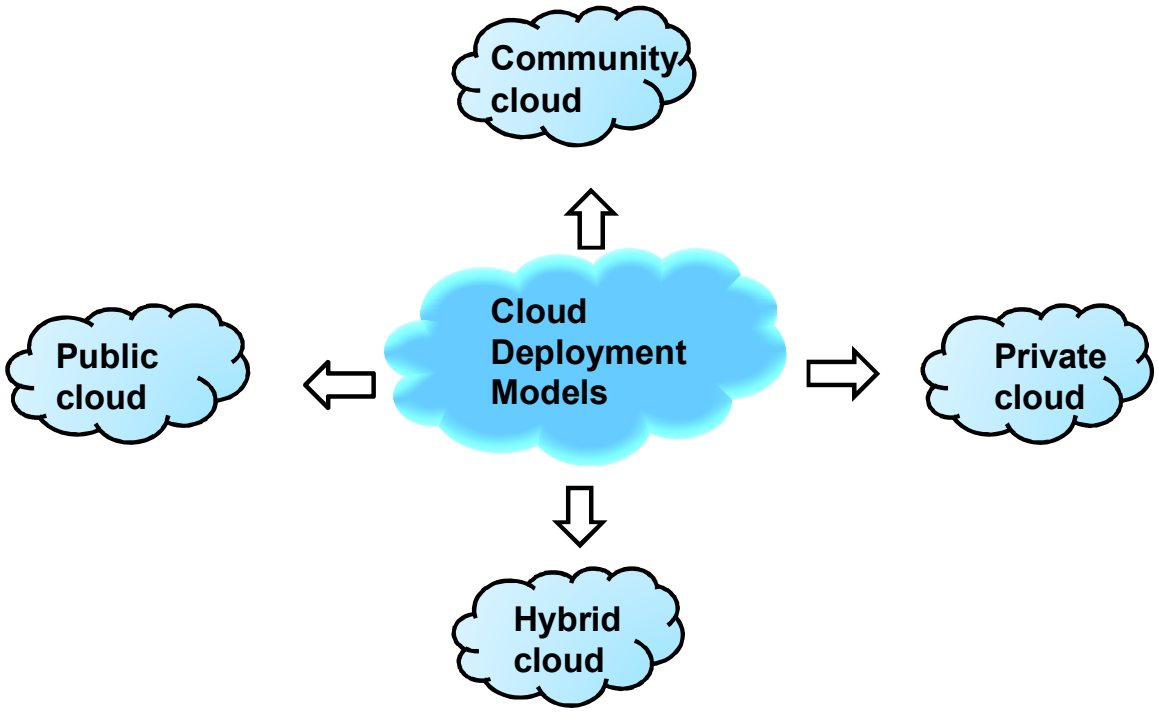
(PaaS) generally refers to an application development and deployment platform delivered as a service to developers, enabling them to quickly build and deploy a SaaS application to end users. The platform typically includes databases, middleware, and development tools, all delivered as a service via the Internet

(IaaS) refers to computing hardware (servers, storage, and network) delivered as a service. This service typically includes the associated software as well as operating systems, virtualization, clustering, and so on. Examples of IaaS in the public cloud include Amazon's Elastic Compute Cloud (EC2) and Simple Storage Service (S3).

The database cloud is built within an enterprise's private cloud environment, as a PaaS model. The database cloud provides on-demand access to database services in a self-service, elastically scalable, and metered manner. The database cloud offers compelling advantages in cost, quality of service, and agility. You can deploy a database within a virtual machine in a IaaS platform.

You can rapidly deploy Database clouds on Oracle Exadata which is a preintegrated and optimized hardware platform that supports both OLTP and DW workloads.

Cloud Deployment Models



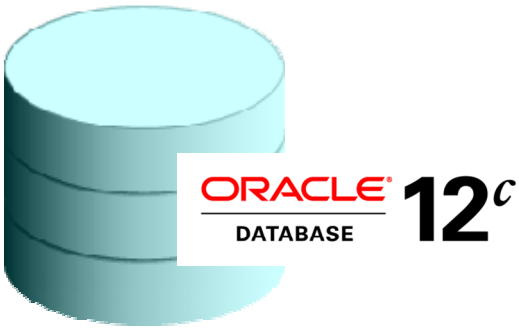
- **Private cloud:** A single organization uses a private cloud, which it typically controls, manages, and hosts in private data centers. However, the organization can also outsource hosting and operation to a third-party service provider. Amazon's Virtual Private Cloud is an example of a private cloud in an external provider setting
- **Public cloud:** Multiple organizations(tenants) uses private cloud on a shared basis; hosted and managed by a third-party service provider. Example: Amazon Elastic Compute Cloud (EC2), IBM's Blue Cloud, Sun Cloud, Google AppEngine, etc.
- **Community cloud:** A group of related organizations, who want to make use of a common cloud computing environment uses the community cloud. It is managed by the participating organizations or by a third-party managed service provider. It is hosted internally or externally. Example: A community might consist of the different branches of the military, all the universities in a given region, or all the suppliers to a large manufacturer.
- **Hybrid cloud:** A single organization that wants to adopt both private and public clouds for a single application uses the hybrid cloud. A third model, the hybrid cloud, is maintained by both internal and external providers For example, an organization might use a public cloud service, such as Amazon Simple Storage Service (Amazon S3) for archived data but continue to maintain in-house(private cloud) storage for operational customer data.

Lesson Agenda

- Course objectives, agenda, and appendixes used in the course
- Overview of Oracle Database 12*c* and related products
- Overview of relational database management concepts and terminologies
- Introduction to SQL and its development environments
- The Human Resource(HR) Schema and the tables used in this course
- Oracle database 12*c* SQL Documentation and Additional Resources

Relational and Object Relational Database Management Systems

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



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

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

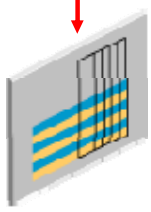
It includes several features for improved performance and functionality of the OLTP applications, such as better sharing of run-time data structures, larger buffer caches, and deferrable constraints. Data warehouse applications benefit from enhancements such as parallel execution of insert, update, and delete operations; partitioning; and parallel-aware query optimization. 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, refer to *Oracle Database Concepts for 10g or 11g database*.

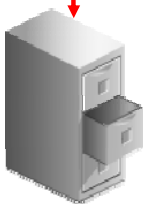
Data Storage on Different Media

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping		
4	60	IT		
5	80	Sales		
6	90	Executive		
7	110	Accounting		
8	190	Contracting		

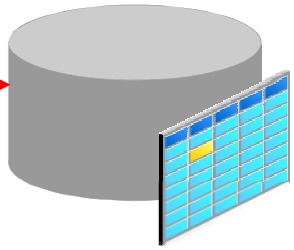
	GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
1	A	1000	2999
2	B	3000	5999
3	C	6000	9999
4	D	10000	14999
5	E	15000	24999
6	F	25000	40000



Electronic spreadsheet



Filing cabinet



Database

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 its employees, departments, and salaries. These pieces of information are called *data*.

Organizations can store data in 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

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-interface products, thereby providing a total solution.

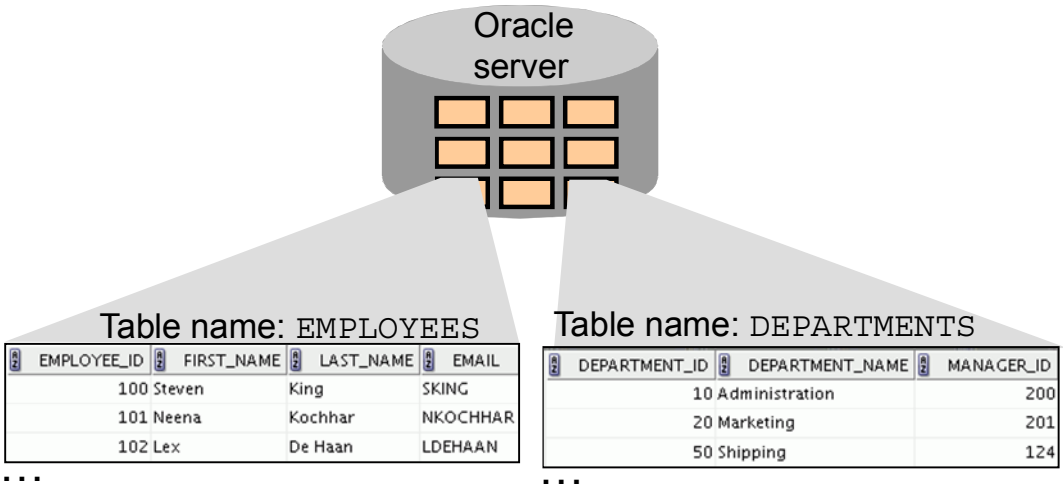
Components of the Relational Model

- 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, refer to *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 controlled by the Oracle server.

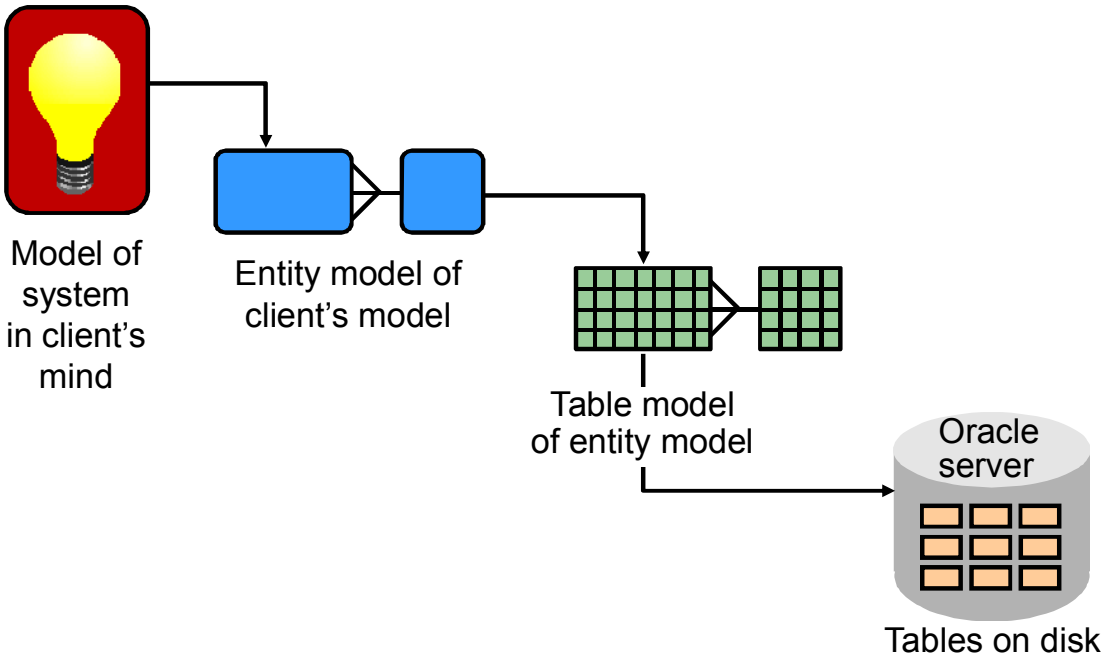


ORACLE

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



Models are the 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

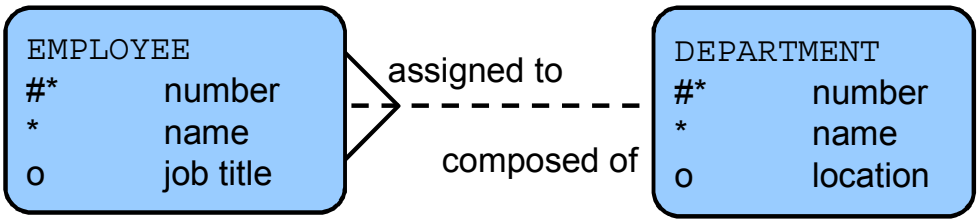
Models help to communicate the concepts that are in people's minds. They can be used to do the following:

- 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

In an effective system, data is divided into discrete categories or entities. An entity relationship (ER) model is an illustration of the 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 the business. Although businesses can change their activities, the type of information tends to remain constant. Therefore, the data structures also tend to be constant.

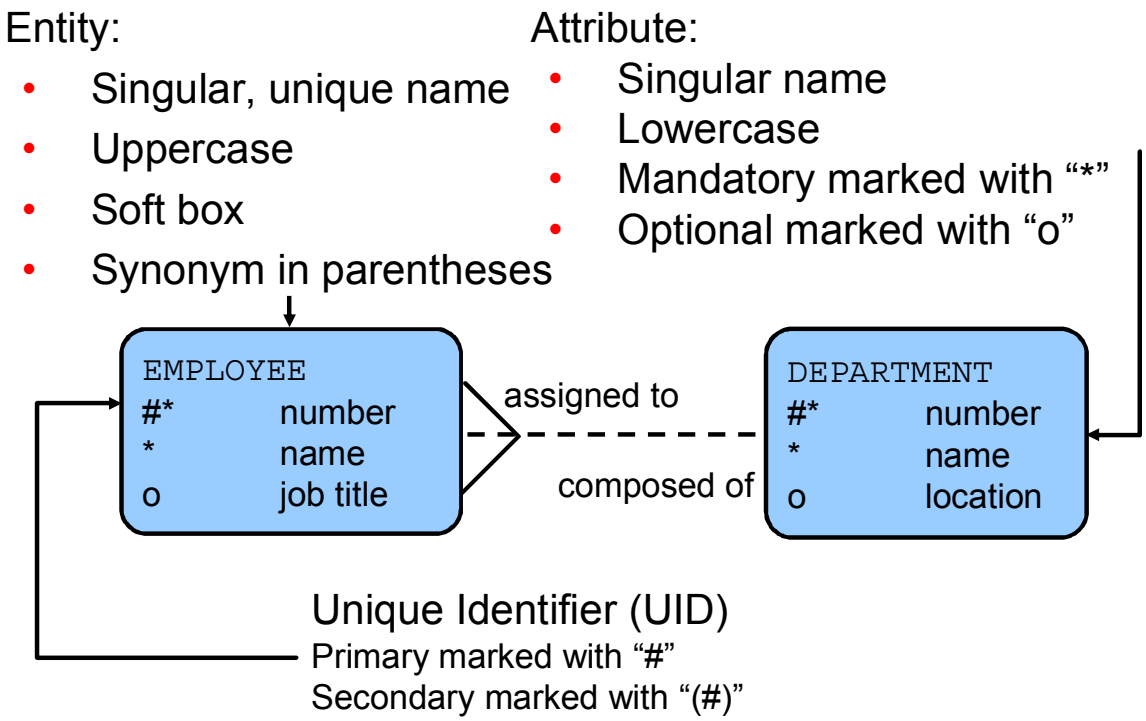
Benefits of ER Modeling

- 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

Key Components

- **Entity:** An aspect of significance about which information must 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



ORACLE

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

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*

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”

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

Each source entity {may be | must be} in relation {one and only one | one or more} with the destination entity.

Note: The convention is to read clockwise.

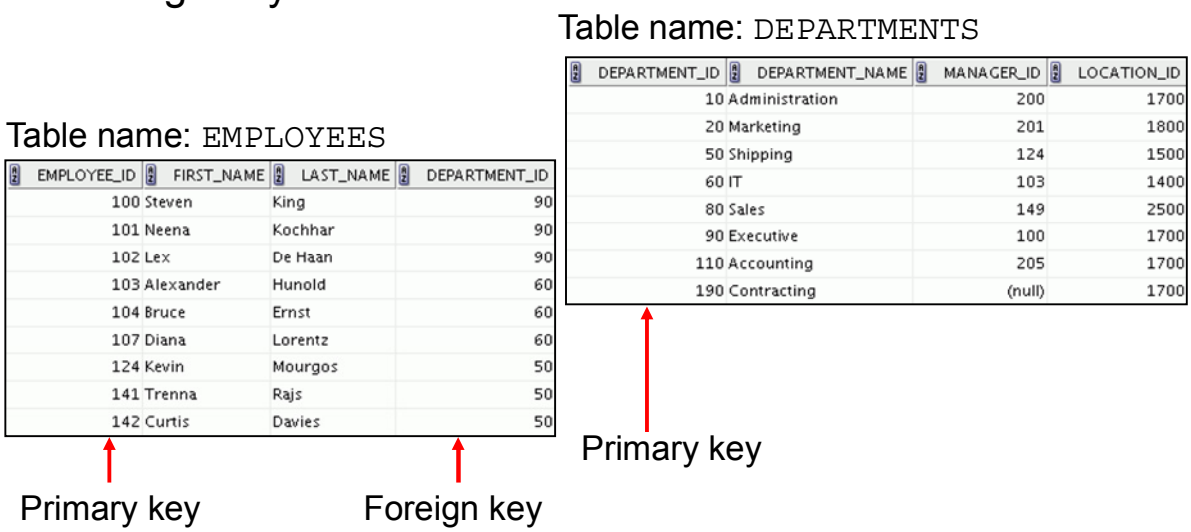
Unique Identifiers

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 hash sign “#”.
- Tag secondary UIDs with a hash sign in parentheses (#).

Relating Multiple Tables

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



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. By 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 the department data by storing it in a separate table.

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; otherwise, it must be null.
- A foreign key must reference either a primary key or a unique key column.

Relational Database Terminology

ORACLE

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 the 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.

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.

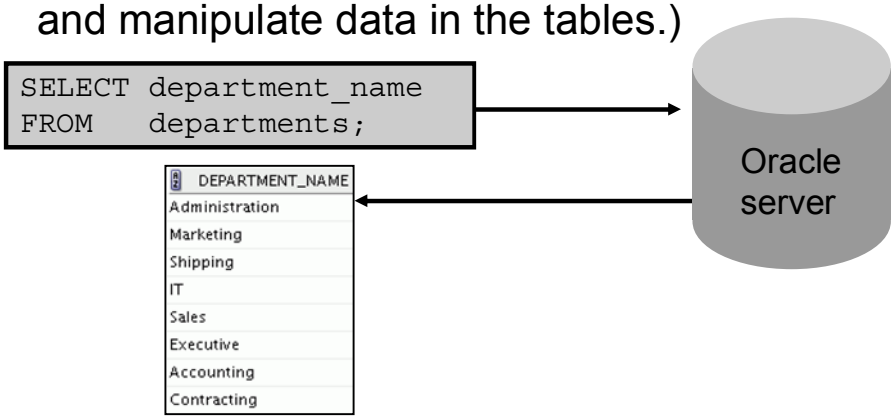
Lesson Agenda

- Course objectives, agenda, and appendixes used in the course
- Overview of Oracle Database 12*c* and related products
- Overview of relational database management concepts and terminologies
- Introduction to SQL and its development environments
- The Human Resource(HR) Schema and the tables used in this course
- Oracle database 12*c* SQL Documentation and Additional Resources

Using SQL to Query Your Database

Structured query language (SQL) is:

- The ANSI standard language for operating relational databases
- Efficient, easy to learn, and use
- Functionally complete (With SQL, you can define, retrieve, and manipulate data in the tables.)



ORACLE

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. SQL is also compliant to ISO Standard (SQL 1999).

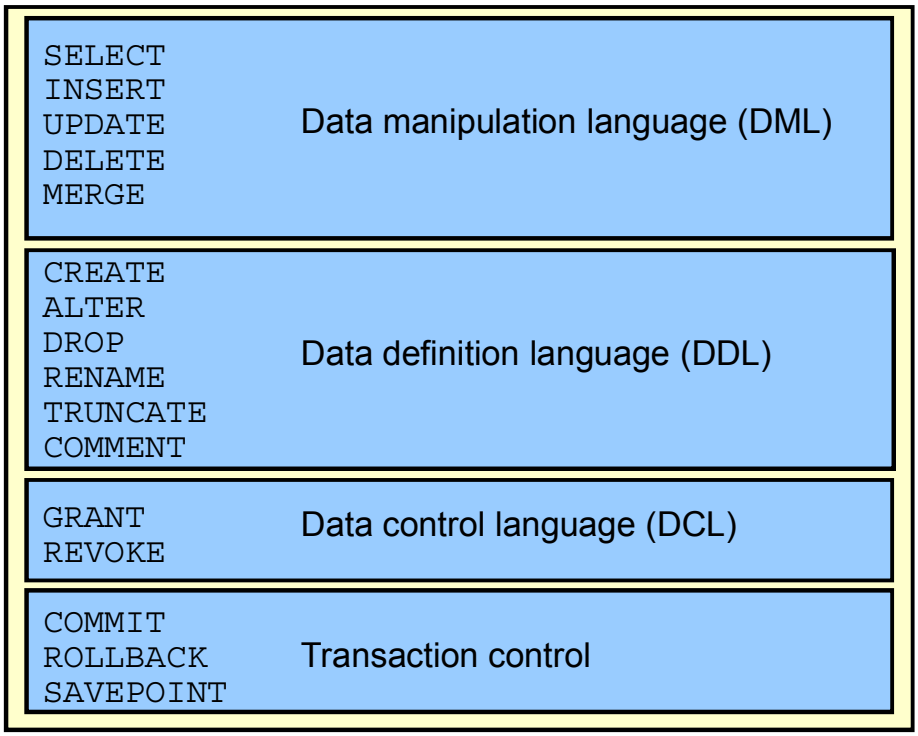
SQL is a set of statements with which all programs and users access data in an Oracle Database. Application programs and Oracle tools often allow users access to the database without using SQL directly, but these applications, in turn, must use SQL when executing the user's request.

SQL provides statements for a variety of tasks, including:

- Querying data
- Inserting, updating, and deleting rows in a table
- Creating, replacing, altering, and dropping objects
- Controlling access to the database and its objects
- Guaranteeing database consistency and integrity

SQL unifies all of the preceding tasks in one consistent language and enables you to work with data at a logical level.

SQL Statements Used in the Course



SQL Statements

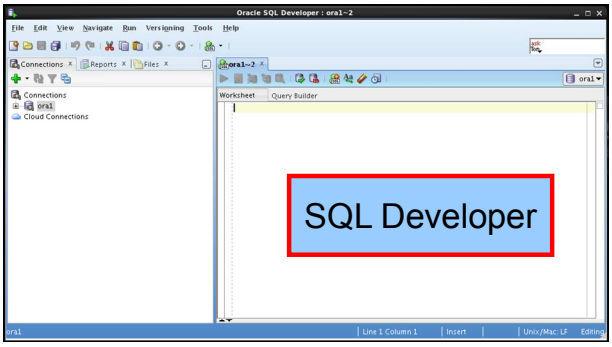
SQL statements supported by Oracle comply with industry standards. Oracle Corporation ensures future compliance with evolving standards by actively involving key personnel in SQL standards committees. The industry-accepted committees are ANSI and International Standards Organization (ISO). Both ANSI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

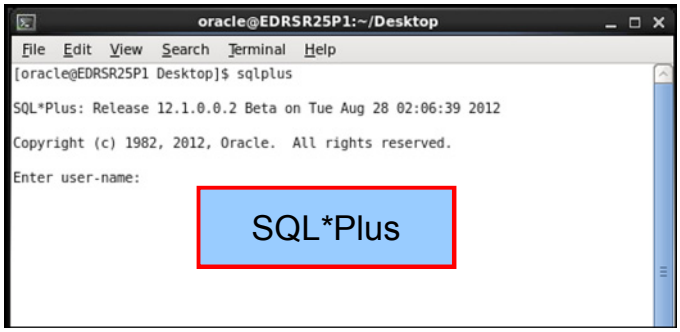
Statement	Description
SELECT INSERT UPDATE DELETE MERGE	Retrieves data from the database, enters new rows, changes existing rows, and removes unwanted rows from tables in the database, respectively. Collectively known as *data manipulation language* (DML)
CREATE ALTER DROP RENAME TRUNCATE COMMENT	Sets up, changes, and removes data structures from tables. Collectively known as *data definition language* (DDL)
GRANT REVOKE	Provides 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

Development Environments for SQL

There are two development environments for this course:

- The primary tool is Oracle SQL Developer.
- SQL*Plus command-line interface can also be used.





SQL Developer

This course is developed using Oracle SQL Developer as the tool for running the SQL statements discussed in the examples in the lessons and the practices. SQL Developer is the default tool for this class.

SQL*Plus

The SQL*Plus environment can also be used to run all SQL commands covered in this course.

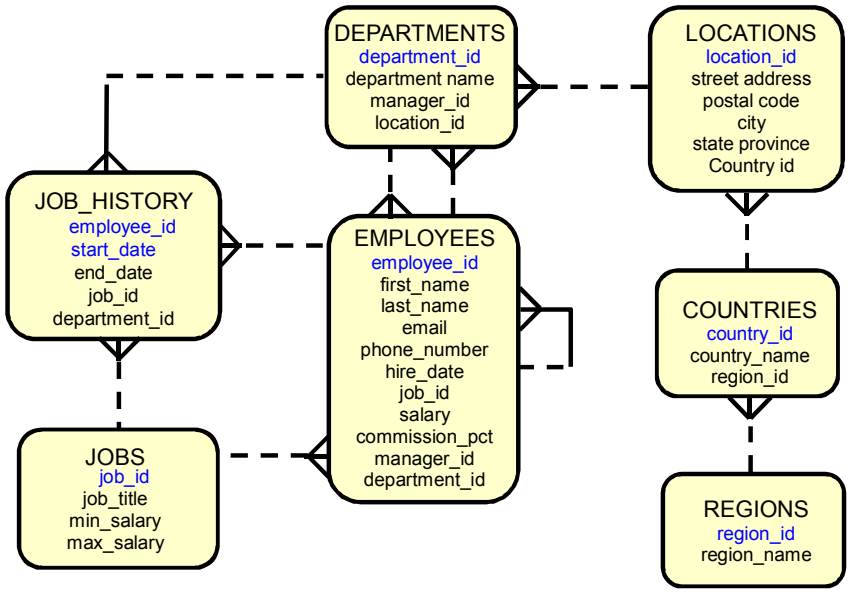
Note

- See Appendix B for information about using SQL Developer, including simple instructions on installation process.
- See Appendix C for information about using SQL*Plus.

Lesson Agenda

- Course objectives, agenda, and appendixes used in the course
- Overview of Oracle Database 12*c* and related products
- Overview of relational database management concepts and terminologies
- Introduction to SQL and its development environments
- The Human Resource(HR) Schema and the tables used in this course
- Oracle database 12*c* SQL Documentation and Additional Resources

Human Resources (HR) Schema



ORACLE

Human Resources (HR) Schema Description

The Human Resources (HR) schema is a part of the Oracle Sample Schemas that can be installed in an Oracle Database. The practice sessions in this course use data from the HR schema.

Table Descriptions

- REGIONS contains rows that represent a region such as America, Asia, and so on.
- COUNTRIES contains rows for countries, each of which is associated with a region.
- LOCATIONS contains the specific address of a specific office, warehouse, or production site of a company in a particular country.
- DEPARTMENTS shows details about the departments in which the employees work. Each department may have a relationship representing the department manager in the EMPLOYEES table.
- EMPLOYEES contains details about each employee working for a department. Some employees may not be assigned to any department.
- JOBS contains the job types that can be held by each employee.
- JOB_HISTORY contains the job history of the employees. If an employee changes departments within a job or changes jobs within a department, a new row is inserted into this table with the earlier job information of the employee.

Tables Used in the Course

EMPLOYEES

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP	17000
3	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP	17000
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-06	AC_MGR	12008
5	104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-07	IT_PROG	6000
6	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-07	IT_PROG	4200
7	124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-07	ST_MAN	5800
8	141	Trenna	Rajs	TRAJS	650.121.8009	17-OCT-03	ST_CLERK	3500
9	142	Curtis	Davies	CDAVIES	650.121.2994	29-JAN-05	ST_CLERK	3100
10	143	Randall	Matos	RMATOS	650.121.2874	15-MAR-06	ST_CLERK	2600
11	144	Peter	Vargas	PVARGAS	650.121.2004	09-JUL-06	ST_CLERK	2500
12	149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-08	SA_MAN	10500
13	174	Ellen	Abel	EABEL	011.44.1644.429267	11-MAY-04	SA_REP	11000
14	176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	24-MAR-06	SA_REP	8600
15	178	Kimberely	Grant	KGRANT	011.44.1644.429263	24-MAY-07	SA_REP	7000
16	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-03	AD_ASST	4400
17	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-04	MK_MAN	13000
18	202	Pat	Fay	PFAY	603.123.6666	17-AUG-05	MK_REP	6000
19	205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-02	AC_MGR	12008
20	206	William	Gietz	WGIETZ	515.123.8181	07-JUN-02	AC_ACCOUNT	8300

	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

JOB_GRADES

	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

DEPARTMENTS

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

Apart from these tables, you will also use the other tables listed in the previous slide such as the LOCATIONS and the JOB_HISTORY table.

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

Lesson Agenda

- Course objectives, agenda, and appendixes used in the course
- Overview of Oracle Database 12*c* and related products
- Overview of relational database management concepts and terminologies
- Introduction to SQL and its development environments
- The Human Resource(HR) Schema and the tables used in this course
- Oracle database 12*c* SQL Documentation and Additional Resources

Oracle Database Documentation

- Oracle Database New Features Guide
- Oracle Database Reference
- Oracle Database SQL Language Reference
- Oracle Database Concepts
- Oracle Database SQL Developer User's Guide

Navigate to http://www.oracle.com/pls/db102/homepage to access the Oracle Database 10*g* documentation library.

Navigate to http://www.oracle.com/pls/db112/homepage to access the Oracle Database 11*g* documentation library.

Additional Resources

For additional information about Oracle Database 12*c*, refer to the following:

- *Oracle Database 12c: New Features eStudies*
- *Oracle Learning Library:*
  - http://www.oracle.com/goto/oll
- *Oracle Cloud :*
  - www.cloud.oracle.com

Summary

In this lesson, you should have learned that:

- Oracle Database 12c extends:
  - The benefits of infrastructure grids
  - The existing information management capabilities
  - The capabilities to use the major application development environments such as PL/SQL, Oracle Java/JDBC, .NET, XML, and so on
  - Oracle Cloud
- The database is based on ORDBMS
- 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 SQL

ORACLE

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 with which you store and manage information by using SQL
- Oracle Fusion Middleware with which you develop, deploy, and manage modular business services that can be integrated and reused
- Oracle Enterprise Manager 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.

Practice 1: Overview

This practice covers the following topics:

- Starting Oracle SQL Developer
- Creating a new database connection
- Browsing the `HR` tables

In this practice, you perform the following:

- Start Oracle SQL Developer and create a new connection to the ora1 account.
- Use Oracle SQL Developer to examine data objects in the ora1 account. The ora1 account contains the HR schema tables.

Note the following location for the lab files:

/home/oracle/labs/sql1/labs

If you are asked to save any lab files, save them in this location.

In any practice, there may be exercises that are prefaced with the phrases “If you have time” or “If you want an extra challenge.” Work on these exercises only if you have completed all other exercises within the allocated time and would like a further challenge to your skills.

Perform the practices slowly and precisely. You can experiment with saving and running command files. If you have any questions at any time, ask your instructor.

Note: All written practices use Oracle SQL Developer as the development environment. Although it is recommended that you use Oracle SQL Developer, you can also use SQL*Plus that is available in this course.