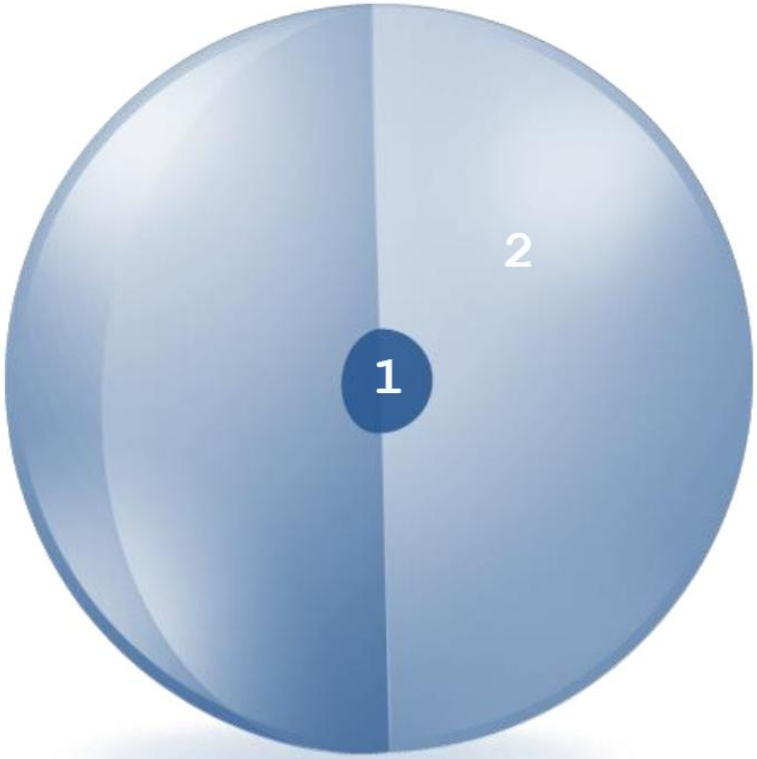
Lesson 1

# Retrieving Data Using the SQL SELECT Statement

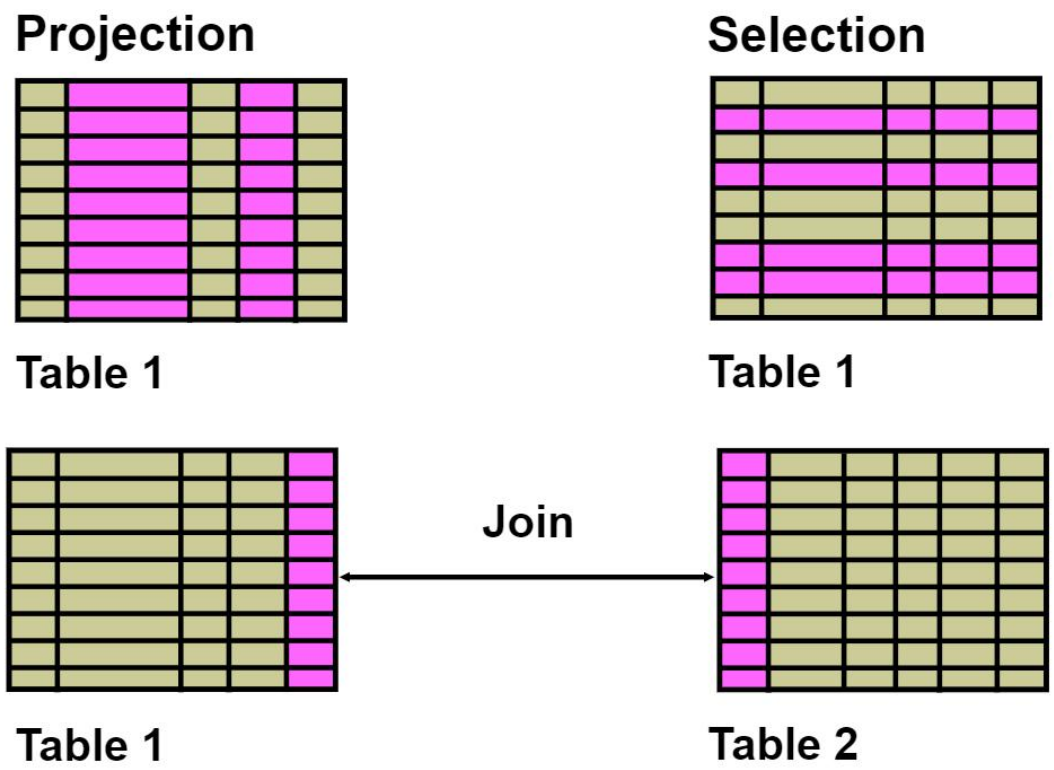
ORACLE

# What You will Learn at the end of this Session ?




1. List the capabilities of SQL SELECT statements
2. Execute a basic SELECT statement

# Capabilities of SQL SELECT Statements



# Basic SELECT Statement

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM    table;
```


**SELECT** identifies the columns to be displayed.

**FROM** identifies the table containing those columns.

# Selecting All Columns

```
SELECT *
FROM   inventories ;
```

| | PRODUCT_ID | WAREHOUSE_ID | QUANTITY_ON_HAND |
|---|---|---|---|
| 1 | 3108 | 8 | 122 |
| 2 | 3110 | 8 | 123 |
| 3 | 3112 | 8 | 123 |
| 4 | 3117 | 8 | 124 |
| 5 | 3124 | 8 | 125 |
| 6 | 3127 | 8 | 125 |
| 7 | 3129 | 8 | 126 |
| 8 | 3134 | 8 | 149 |
| 9 | 3139 | 8 | 150 |
| 10 | 3140 | 8 | 150 |
| 11 | 3143 | 8 | 151 |

...

# Selecting Specific Columns

```
SELECT product_id, quantity_on_hand
FROM   inventories ;
```

| | PRODUCT_ID | QUANTITY_ON_HAND |
|---|---|---|
| 1 | 3108 | 122 |
| 2 | 3110 | 123 |
| 3 | 3112 | 123 |
| 4 | 3117 | 124 |
| 5 | 3124 | 125 |
| 6 | 3127 | 125 |
| 7 | 3129 | 126 |
| 8 | 3134 | 149 |
| 9 | 3139 | 150 |
| 10 | 3140 | 150 |
| 11 | 3143 | 151 |

...

# Writing SQL Statements

SQL statements are not case sensitive

SQL statements can be entered on one or more lines.

Keywords cannot be abbreviated or split across lines.


Clauses are usually placed on separate lines.

Indents are used to enhance readability.

In SQL*Plus, you are required to end each SQL statement with a semicolon (;).

In SQL Developer, SQL statements can be optionally terminated by a semicolon (;). Semicolons are required when you execute multiple SQL statements

ORACLE

# Column Heading Defaults


Default heading alignment: Left-aligned
Default heading display: Uppercase


Character and Date column headings are left-aligned.
Number column headings are right-aligned.
Default heading display: Uppercase

# Column Heading Defaults

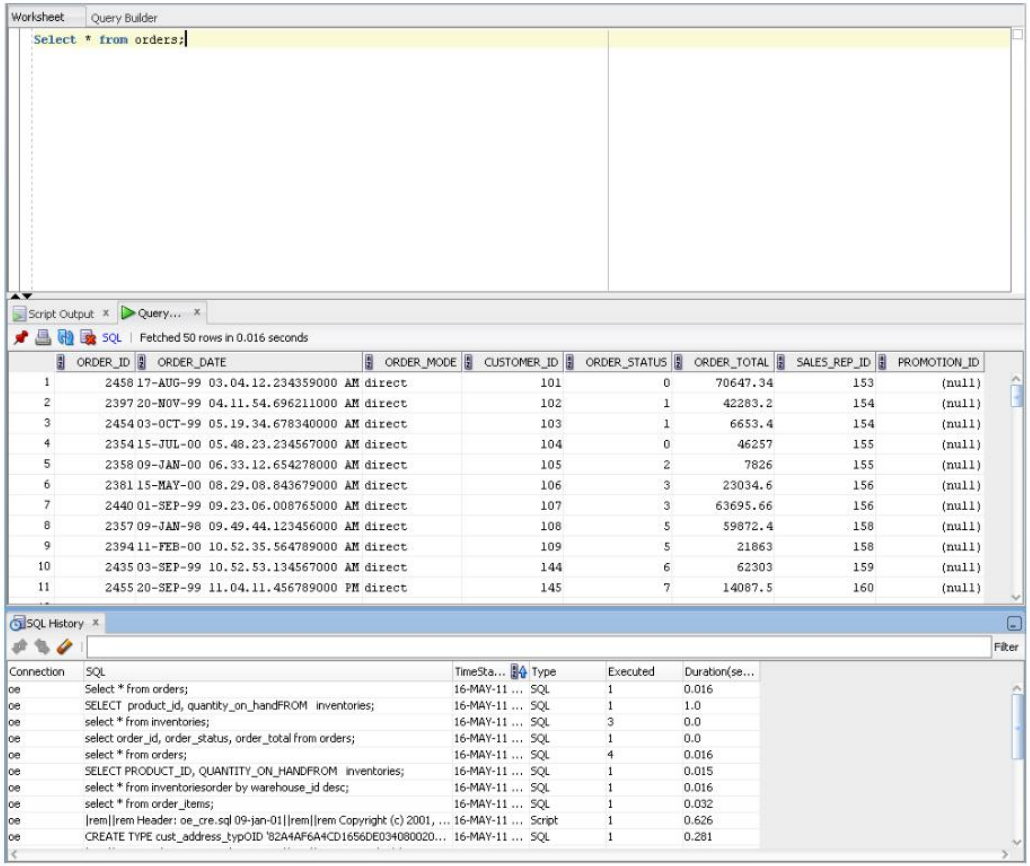
| | EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER |
|---|---|---|---|---|---|
| 1 | 100 | Steven | King | SKING | 515.123.4567 |
| 2 | 101 | Neena | Kochhar | NKOCHHAR | 515.123.4568 |
| 3 | 102 | Lex | De Haan | LDEHAAN | 515.123.4569 |
| 4 | 103 | Alexander | Hunold | AHUNOLD | 590.423.4567 |
| 5 | 104 | Bruce | Ernst | BERNST | 590.423.4568 |
| 6 | 105 | David | Austin | DAUSTIN | 590.423.4569 |
| 7 | 106 | Valli | Pataballa | VPATABAL | 590.423.4560 |
| 8 | 107 | Diana | Lorentz | DLORENTZ | 590.423.5567 |
| 9 | 108 | Nancy | Greenberg | NGREENBE | 515.124.4569 |
| 10 | 109 | Daniel | Faviet | DFAVIET | 515.124.4169 |

# SQL Developer sample screenshot

Worksheet

Query Result

History



# Arithmetic Expressions

•Create expressions with number and date data by using arithmetic operators.

| Operator | Description |
|---|---|
| + | Add |
| - | Subtract |
| * | Multiply |
| / | Divide |

# Using Arithmetic Operators

```
SELECT   product_id,  quantity_on_hand, quantity_on_hand+200
FROM   inventories ;
```

| | PRODUCT_ID | QUANTITY_ON_HAND | QUANTITY_ON_HAND+200 |
|---|---|---|---|
| 1 | 3108 | 122 | 322 |
| 2 | 3110 | 123 | 323 |
| 3 | 3112 | 123 | 323 |
| 4 | 3117 | 124 | 324 |
| 5 | 3124 | 125 | 325 |
| 6 | 3127 | 125 | 325 |
| 7 | 3129 | 126 | 326 |
| 8 | 3134 | 149 | 349 |
| 9 | 3139 | 150 | 350 |
| 10 | 3140 | 150 | 350 |
| 11 | 3143 | 151 | 351 |

# Operator Precedence

```
SELECT   product_id, quantity_on_hand, 12*quantity_on_hand+200
FROM   inventories ;
```

1

| | PRODUCT_ID | QUANTITY_ON_HAND | 12*QUANTITY_ON_HAND+200 |
|---|---|---|---|
| 1 | 3108 | 122 | 1664 |
| 2 | 3110 | 123 | 1676 |
| 3 | 3112 | 123 | 1676 |
| 4 | 3117 | 124 | 1688 |

```
SELECT   product_id, quantity_on_hand, 12*(quantity_on_hand+200)
FROM   inventories ;
```

2

| | PRODUCT_ID | QUANTITY_ON_HAND | 12*(QUANTITY_ON_HAND+200) |
|---|---|---|---|
| 1 | 3108 | 122 | 3864 |
| 2 | 3110 | 123 | 3876 |
| 3 | 3112 | 123 | 3876 |
| 4 | 3117 | 124 | 3888 |

# What is a NULL value ?

**What is a NULL value ?**

**If a row does not have an entry for a particular column, that value is said to be NULL..**

**What is a NULL value ?**

**It is the absence of any character, zero, blank space etc.**

**What is a NULL value ?**

**Arithmetic operations on a NULL value always return a NULL value.**

ORACLE

# Defining a Null Value

- Null is a value that is unavailable, unassigned, unknown, or inapplicable.
- Null is not the same as zero or a blank space.

```
SELECT   order_id,  ROUND (order_date)  "ORDER_DATE",
         customer_id, promotion_id
FROM   orders ;
```

| | ORDER_ID | ORDER_DATE | CUSTOMER_ID | PROMOTION_ID |
|---|---|---|---|---|
| 1 | 2458 | 17-AUG-99 | 101 | (null) |
| 2 | 2397 | 20-NOV-99 | 102 | (null) |
| 3 | 2454 | 03-OCT-99 | 103 | (null) |
| 4 | 2354 | 15-JUL-00 | 104 | (null) |
| 5 | 2358 | 09-JAN-00 | 105 | (null) |

▪ ▪ ▪

**Note:** Round() will be explained later during the course of the presentation.

# Null Values in Arithmetic Expressions

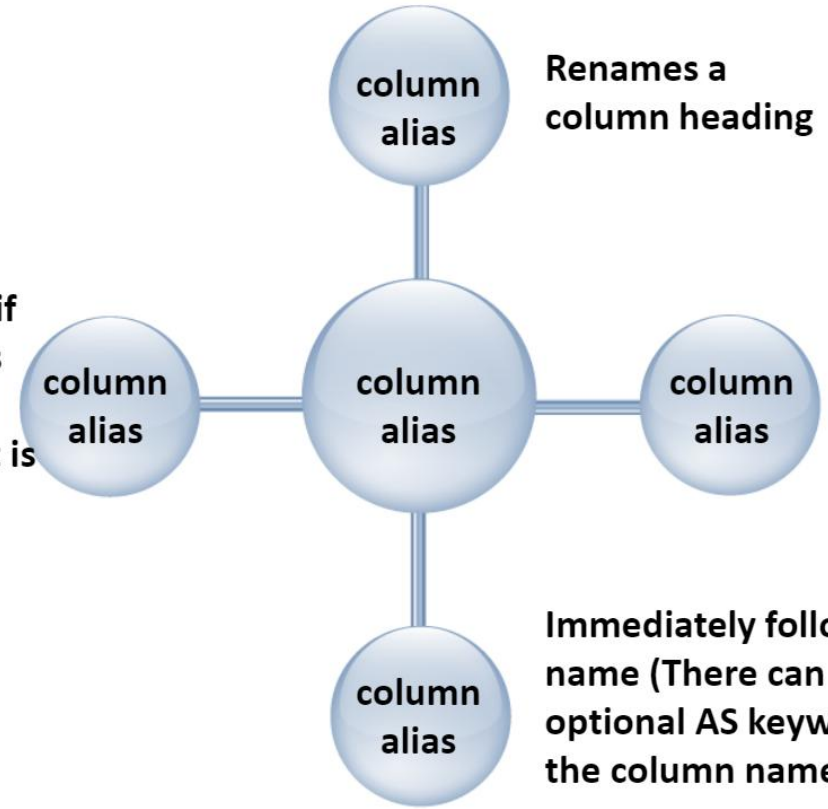
•Arithmetic expressions containing a null value evaluate to null.

```
SELECT   order_id, 12*order_id*promotion_id
FROM   orders ;
```

| | ORDER_ID | 12*ORDER_ID*PROMOTION_ID |
|---|---|---|
| 1 | 2458 | (null) |
| 2 | 2397 | (null) |
| 3 | 2454 | (null) |
| 4 | 2354 | (null) |
| 5 | 2358 | (null) |
| 6 | 2381 | (null) |
| 7 | 2440 | (null) |
| 8 | 2357 | (null) |
| 9 | 2394 | (null) |
| 10 | 2435 | (null) |
| 11 | 2455 | (null) |

■ ■ ■

# Defining a Column Alias




Renames a column heading

Requires double quotation marks if it contains spaces or special characters, or if it is case-sensitive

Is useful with calculations

Immediately follows the column name (There can also be the optional AS keyword between the column name and the alias.)

# Using Column Aliases

```
SELECT   product_id  AS  Product , quantity_on_hand  Quantity
FROM   inventories ;
```

| | PRODUCT | QUANTITY |
|---|---|---|
| 1 | 3108 | 122 |
| 2 | 3110 | 123 |
| 3 | 3112 | 123 |
| 4 | 3117 | 124 |

■ ■ ■

```
SELECT   order_id "Order " ,  ROUND(order_date) "Date of Order"
FROM orders ;
```

| | Order | Date of Order |
|---|---|---|
| 1 | 2458 | 17-AUG-99 |
| 2 | 2397 | 20-NOV-99 |
| 3 | 2454 | 03-OCT-99 |
| 4 | 2354 | 15-JUL-00 |

■ ■ ■

# Concatenation Operator

- A concatenation operator:
  - Links columns or character strings to other columns
  - Is represented by two vertical bars (||)
  - Creates a resultant column that is a character expression

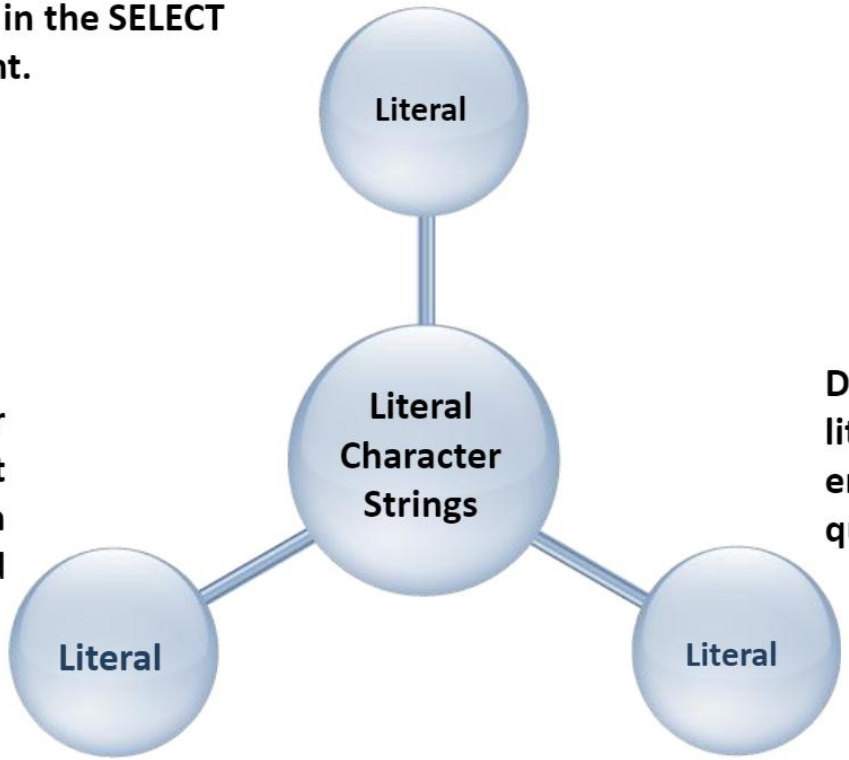
```
SELECT first_name || last_name AS "NAME"
FROM customers ;
```

| | NAME |
|---|---|
| 1 | EllenAbel |
| 2 | SundarAnde |
| 3 | MozheAtkinson |
| 4 | DavidAustin |
| 5 | HermannBaer |
| 6 | ShelliBaida |
| 7 | AmitBanda |
| 8 | ElizabethBates |

■ ■ ■

# Literal Character Strings

A literal is a character, a number, or a date that is included in the SELECT statement.




Each character string is output once for each row returned

Date and character literal values must be enclosed within single quotation marks.

ORACLE

# Using Literal Character Strings

```
SELECT   product_id  ||  ' is in Warehouse '  ||  warehouse_id
         AS  "Product-Warehouse"
FROM   inventories ;
```

| | Product-Warehouse |
|---|---|
| 1 | 1733 is in Warehouse 1 |
| 2 | 1734 is in Warehouse 1 |
| 3 | 1737 is in Warehouse 1 |
| 4 | 1738 is in Warehouse 1 |
| 5 | 1745 is in Warehouse 1 |
| 6 | 1748 is in Warehouse 1 |
| 7 | 2278 is in Warehouse 1 |

■ ■ ■

# Alternative Quote (q) Operator

- Specify your own quotation mark delimiter.
- Select any delimiter.
- Increase readability and usability.

```
SELECT   department_name  || q ' [ Department's Manager Id: ] '
     ||  manager_id
     AS " Department and Manager "
FROM   departments ;
```

| | Department and Manager |
|---|---|
| 1 | Administration Department's Manager Id: 200 |
| 2 | Marketing Department's Manager Id: 201 |
| 3 | Shipping Department's Manager Id: 124 |
| 4 | IT Department's Manager Id: 103 |
| 5 | Sales Department's Manager Id: 149 |
| 6 | Executive Department's Manager Id: 100 |
| 7 | Accounting Department's Manager Id: 205 |
| 8 | Contracting Department's Manager Id: |

# Duplicate Rows

- The default display of queries is all rows, including duplicate rows.

1

```
SELECT   department_id
FROM   employees ;
```

| | DEPARTMENT_ID |
|---|---|
| 1 | 10 |
| 2 | 20 |
| 3 | 20 |
| 4 | 110 |
| 5 | 110 |

...

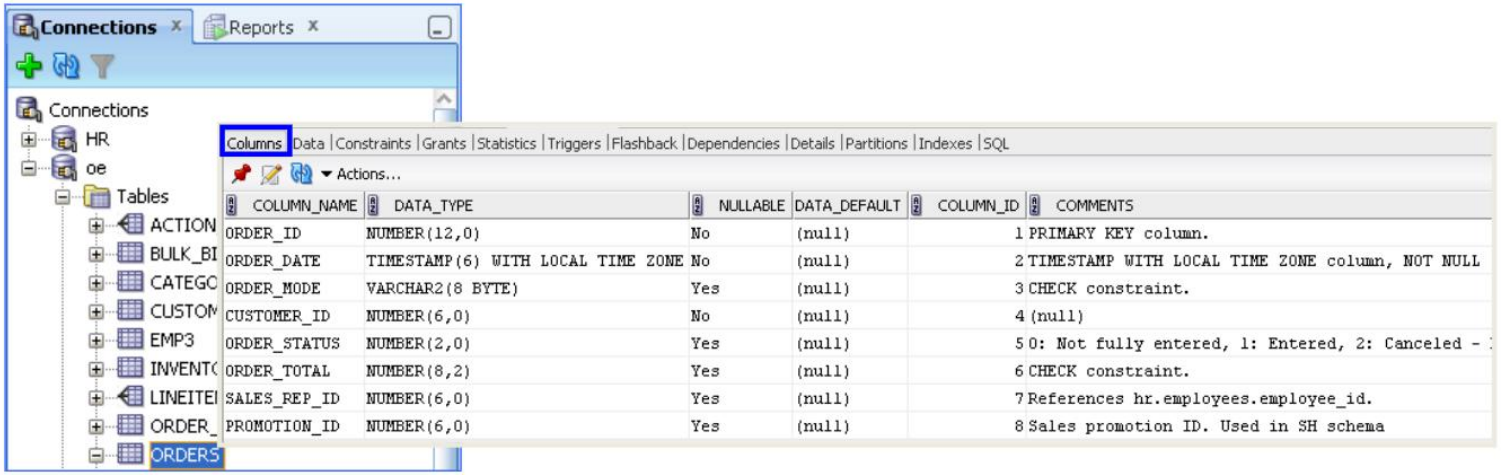
2

```
SELECT  DISTINCT  department_id
FROM   employees ;
```

| | DEPARTMENT_ID |
|---|---|
| 1 | (null) |
| 2 | 20 |
| 3 | 90 |
| 4 | 110 |
| 5 | 50 |
| 6 | 80 |
| 7 | 10 |
| 8 | 60 |

# Displaying the Table Structure using SQL Developer

– Select the required table in the “Connections” tree in the SQL Developer and use the “Columns” tab to view the table structure.




Here, the table structure of the Orders table is displayed.

# Using the DESCRIBE Command

- Use the DESCRIBE command to display the structure of a table.

```
DESCRIBE   orders ;
```

```
Name          Null     Type
------------ -------- ------------------------------
ORDER_ID      NOT NULL NUMBER(12)
ORDER_DATE    NOT NULL TIMESTAMP(6) WITH LOCAL TIME ZONE
ORDER_MODE             VARCHAR2(8)
CUSTOMER_ID   NOT NULL NUMBER(6)
ORDER_STATUS           NUMBER(2)
ORDER_TOTAL            NUMBER(8,2)
SALES_REP_ID           NUMBER(6)
PROMOTION_ID           NUMBER(6)
```

ORACLE

# Quiz

- Identify the `SELECT` statements that execute successfully.

```
1.SELECT first_name, last_name, job_id, salary*12
  AS Yearly Sal
 FROM   employees;

2.SELECT first_name, last_name, job_id, salary*12
  "yearly sal"
 FROM   employees;

3.SELECT first_name, last_name, job_id, salary AS
  "yearly sal"
 FROM   employees;

4.SELECT first_name+last_name AS name, job_Id,
  salary*12 yearly sal
 FROM   employees;
```

# Quiz

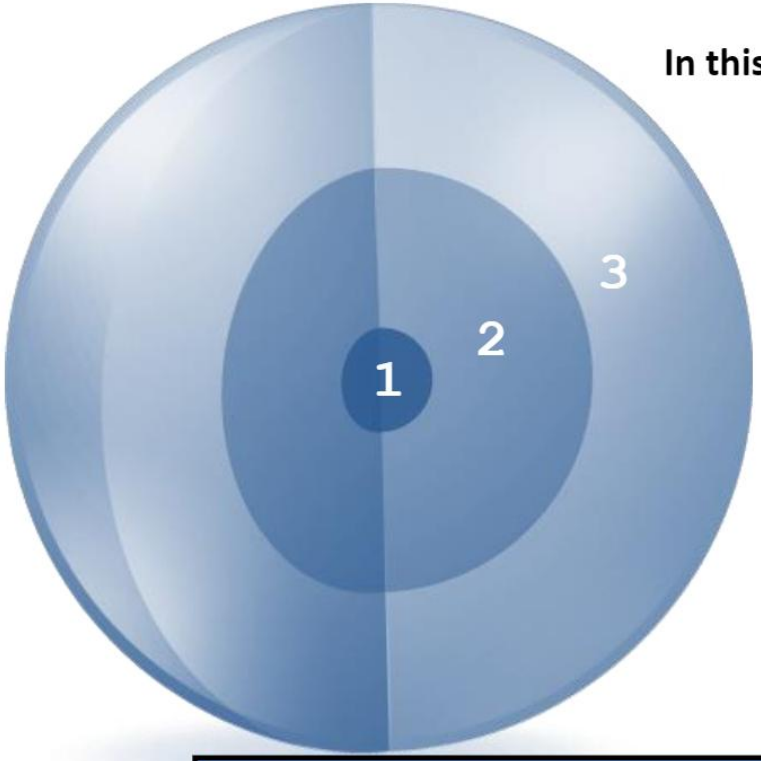
- Identify the SELECT statements that execute successfully.

5.
```
SELECT product_id, warehouse_id AS "Product",
"Warehouse"
FROM   employees;
```

6.
```
SELECT order_id|| is in ||order_mode|| mode AS
 "Order Mode"
FROM   inventories;
```

7. Write an SQL query to display all the quantity_on_hand in the warehouse with warehouse_id

# Session Summary




**In this lesson, you should have learned how to:**

1. Returns all rows and columns from a table
2. Returns specified columns from a table
3. Uses column aliases to display more descriptive column headings

**Syntax :**

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM table;
```

ORACLE

# Practice 1: Overview This practice covers the following topics: