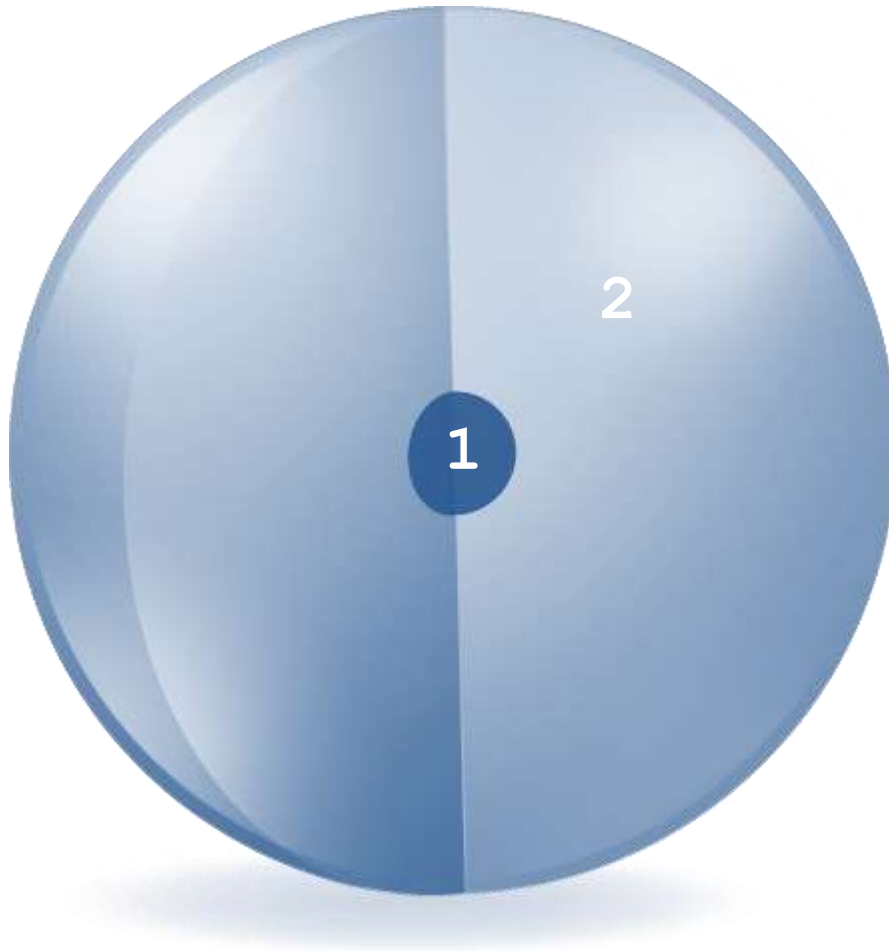


Retrieving Data Using the SQL SELECT Statement

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

Projection

Table 1

Selection

Table 1

Join

Table 1

Table 2

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

...



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

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

Column Heading Defaults

A light blue 3D rectangular box with a slight shadow underneath.

SQL Developer






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

A light blue 3D rectangular box with a slight shadow underneath.

SQL*Plus

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

The screenshot displays the Oracle SQL Developer interface. At the top, the 'Worksheet' tab is active, showing a query: `Select * from orders;`. Below the worksheet, the 'Script Output' tab is active, displaying the query result. The result is a table with 11 rows and 9 columns: ORDER_ID, ORDER_DATE, ORDER_MODE, CUSTOMER_ID, ORDER_STATUS, ORDER_TOTAL, SALES_REP_ID, and PROMOTION_ID. The bottom panel shows the 'SQL History' tab, which lists the executed queries, their connection names, and execution details.

ORDER_ID	ORDER_DATE	ORDER_MODE	CUSTOMER_ID	ORDER_STATUS	ORDER_TOTAL	SALES_REP_ID	PROMOTION_ID
1	2458 17-AUG-99 03.04.12.234359000	AM direct	101	0	70647.34	153	(null)
2	2397 20-NOV-99 04.11.54.696211000	AM direct	102	1	42283.2	154	(null)
3	2454 03-OCT-99 05.19.34.678340000	AM direct	103	1	6653.4	154	(null)
4	2354 15-JUL-00 05.40.23.234567000	AM direct	104	0	46257	155	(null)
5	2358 09-JAN-00 06.33.12.654278000	AM direct	105	2	7826	155	(null)
6	2381 15-MAY-00 08.29.08.843679000	AM direct	106	3	23034.6	156	(null)
7	2440 01-SEP-99 09.23.06.088765000	AM direct	107	3	63695.66	156	(null)
8	2357 09-JAN-98 09.49.44.123456000	AM direct	108	5	59872.4	158	(null)
9	2394 11-FEB-00 10.52.35.564789000	AM direct	109	5	21863	158	(null)
10	2435 03-SEP-99 10.52.53.134567000	AM direct	144	6	62303	159	(null)
11	2455 20-SEP-99 11.04.11.456789000	PM direct	145	7	14087.5	160	(null)

Connection	SQL	TimeSta...	Type	Executed	Duration(se...
oe	Select * from orders;	16-MAY-11 ...	SQL	1	0.016
oe	SELECT product_id, quantity_on_hand FROM inventories;	16-MAY-11 ...	SQL	1	1.0
oe	select * from inventories;	16-MAY-11 ...	SQL	3	0.0
oe	select order_id, order_status, order_total from orders;	16-MAY-11 ...	SQL	1	0.0
oe	select * from orders;	16-MAY-11 ...	SQL	4	0.016
oe	SELECT PRODUCT_ID, QUANTITY_ON_HAND FROM inventories;	16-MAY-11 ...	SQL	1	0.015
oe	select * from inventories order by warehouse_id desc;	16-MAY-11 ...	SQL	1	0.016
oe	select * from order_items;	16-MAY-11 ...	SQL	1	0.032
oe	[rem][rem Header: oe_cre.sql 09-Jan-01][rem][rem Copyright (c) 2001, ...	16-MAY-11 ...	Script	1	0.626
oe	CREATE TYPE cust_address_typID '%24A46A4CD1656DE034080020...	16-MAY-11 ...	SQL	1	0.281

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.

- 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.

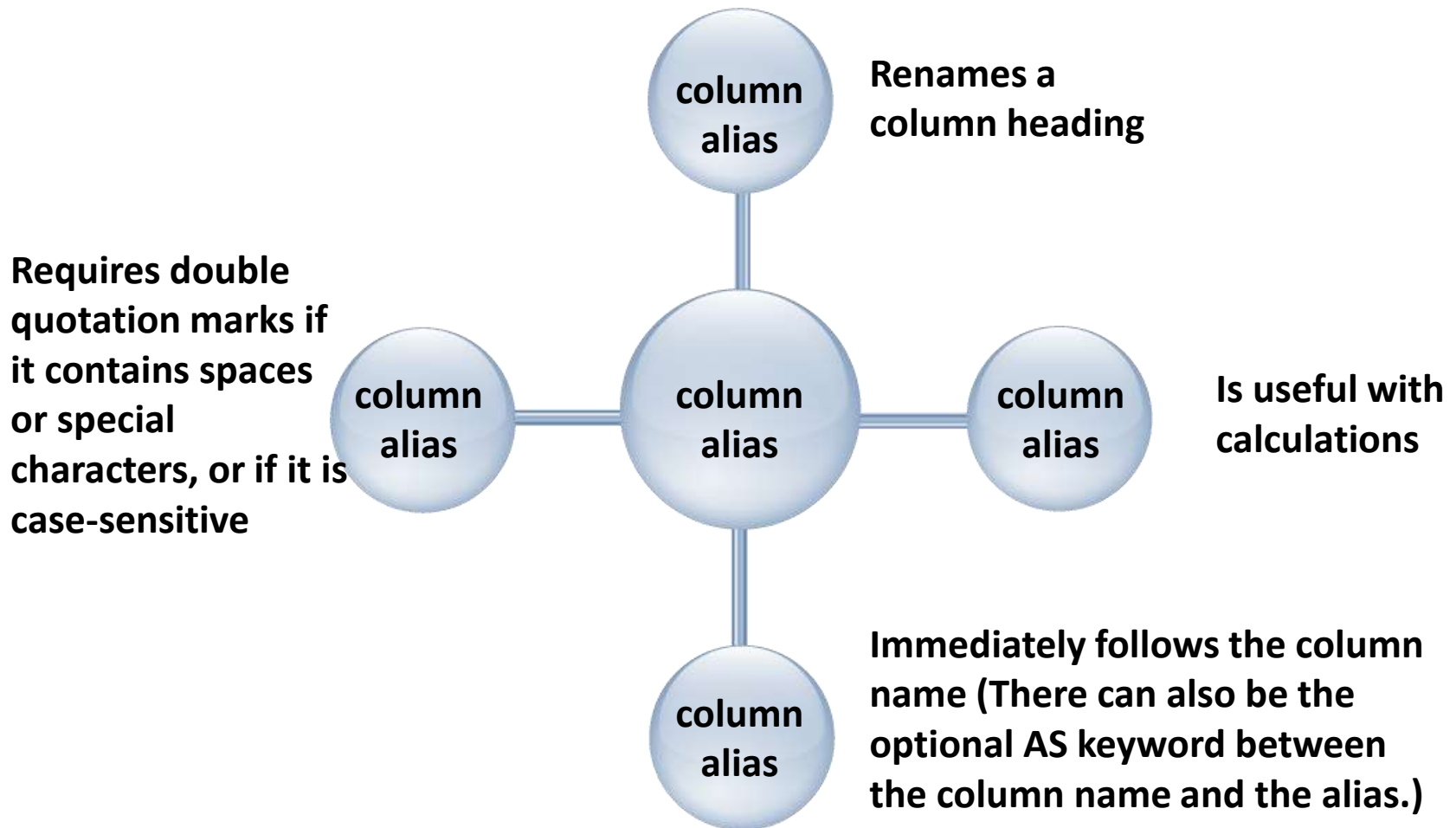
Null Values in Arithmetic Expressions

- 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)

■ ■ ■



```
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

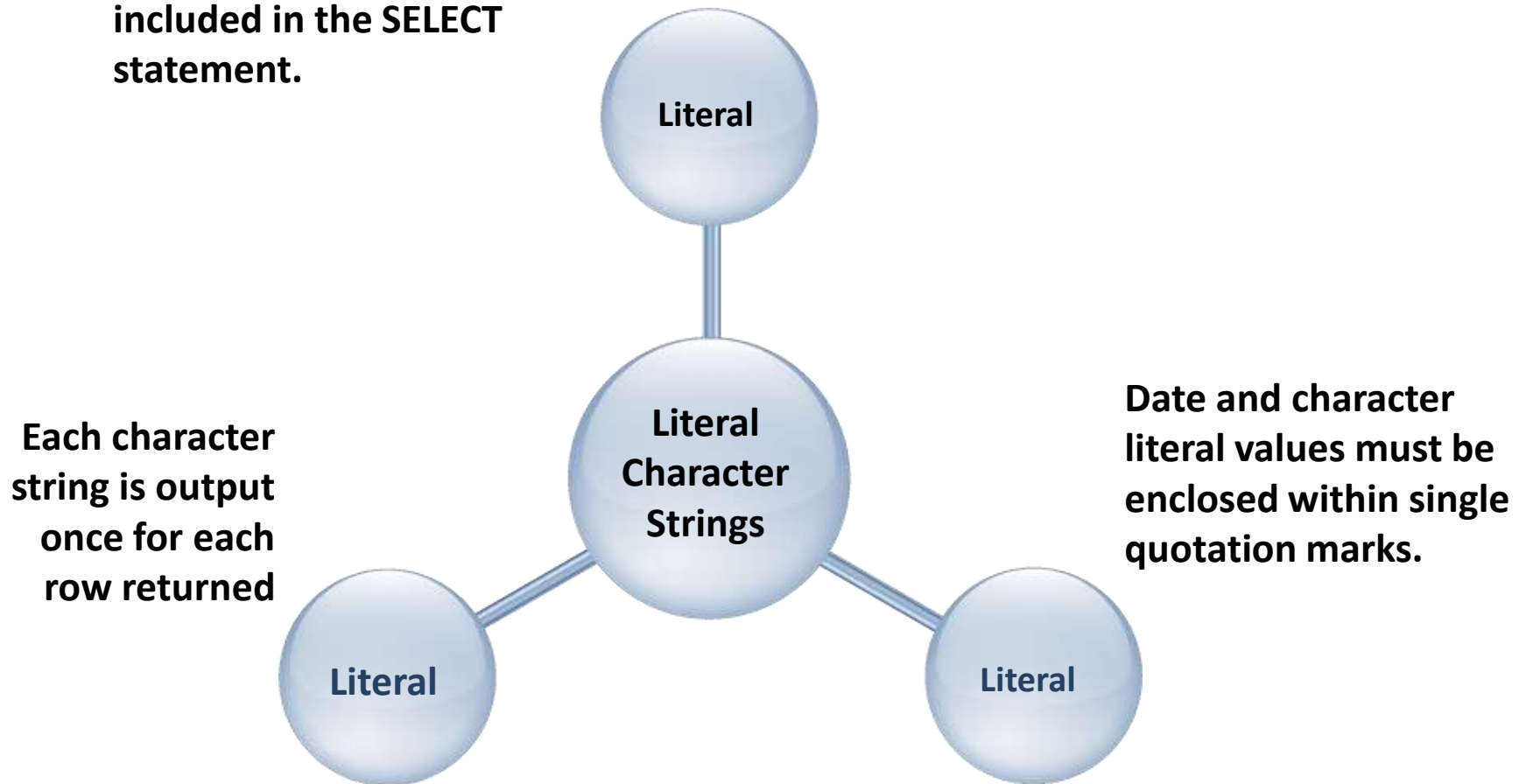
- A concatenation operator:
 - Links columns or character strings to other columns
 - Is represented by two vertical bars (||)
 - Creates a resultant column that is a character expression

```
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

■ ■ ■

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



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

...

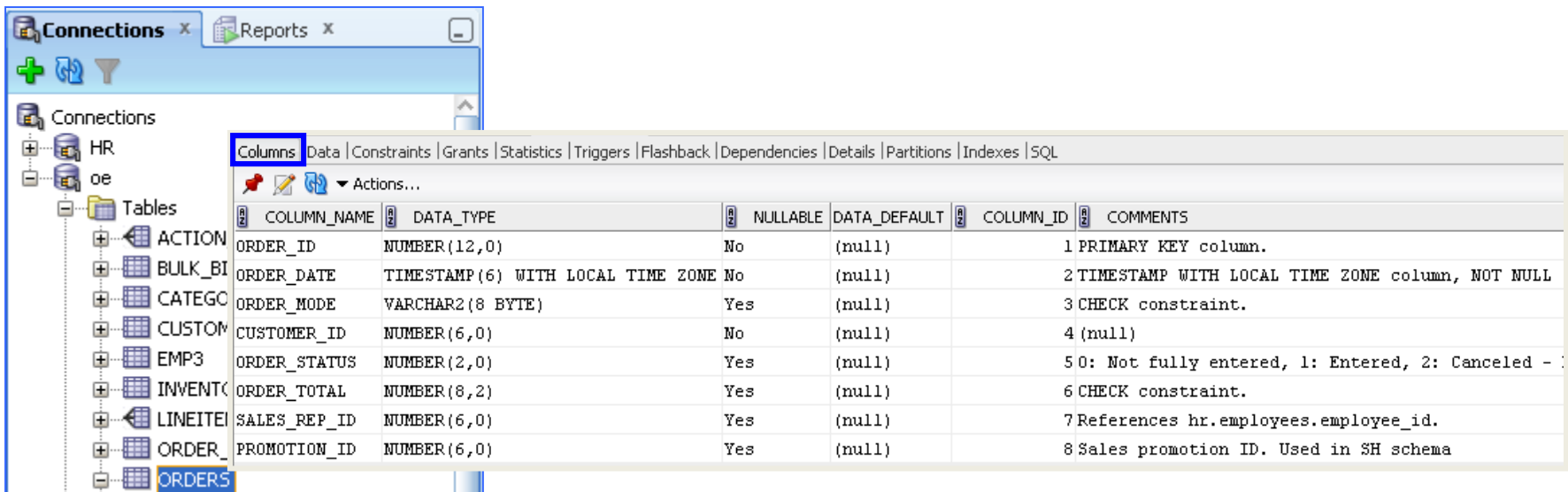
2

```
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.



The screenshot shows the SQL Developer interface. On the left, the 'Connections' tree is expanded, showing the 'Tables' folder under the 'oe' connection. The 'ORDERS' table is selected. The main pane displays the 'Columns' tab for the 'ORDERS' table, showing a list of columns with their data types, nullability, and comments.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
ORDER_ID	NUMBER(12,0)	No	(null)	1	PRIMARY KEY column.
ORDER_DATE	TIMESTAMP(6) WITH LOCAL TIME ZONE	No	(null)	2	TIMESTAMP WITH LOCAL TIME ZONE column, NOT NULL
ORDER_MODE	VARCHAR2(8 BYTE)	Yes	(null)	3	CHECK constraint.
CUSTOMER_ID	NUMBER(6,0)	No	(null)	4	(null)
ORDER_STATUS	NUMBER(2,0)	Yes	(null)	5	0: Not fully entered, 1: Entered, 2: Canceled -
ORDER_TOTAL	NUMBER(8,2)	Yes	(null)	6	CHECK constraint.
SALES_REP_ID	NUMBER(6,0)	Yes	(null)	7	References hr.employees.employee_id.
PROMOTION_ID	NUMBER(6,0)	Yes	(null)	8	Sales promotion ID. Used in SH schema

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)

- 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;
```

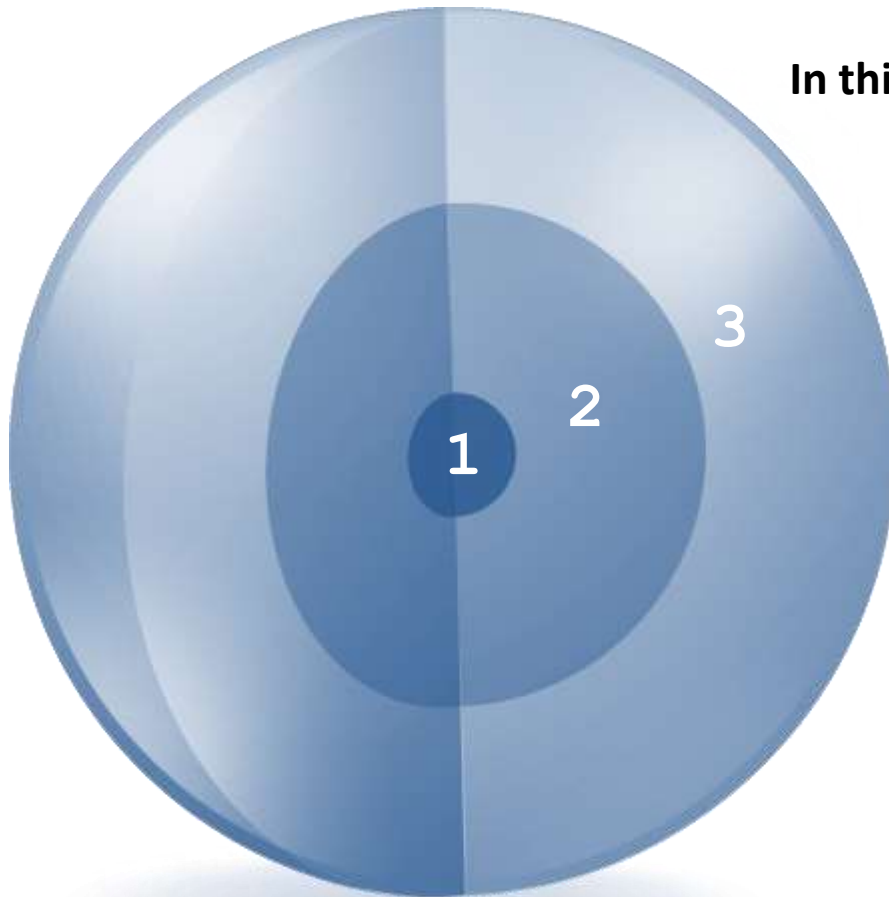
```
4.SELECT first_name+last_name AS name, job_Id,
   salary*12 yearly sal
FROM   employees;
```

- 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



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;
```

Practice 1: Overview

This practice covers the following topics:

