

EXERCISE-4

Writing Basic SQL SELECT Statements

OBJECTIVES

After the completion of this exercise, the students will be able to do the following:

- List the capabilities of SQL SELECT Statement
- Execute a basic SELECT statement

Capabilities of SQL SELECT statement

A SELECT statement retrieves information from the database. Using a select statement, we can perform

- ✓ Projection: To choose the columns in a table
- ✓ Selection: To choose the rows in a table
- ✓ Joining: To bring together the data that is stored in different tables

Basic SELECT Statement

Syntax

```
SELECT *|DISTINCT Column_name| alias  
FROM table_name;
```

NOTE:

DISTINCT—Suppress the duplicates.

Alias—gives selected columns different headings.

Example: 1

```
SELECT * FROM departments;
```

Example: 2

```
SELECT location_id, department_id FROM departments;
```

Writing SQL Statements

- SQL statements are not case sensitive
- SQL statements can be 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

Using Arithmetic Expressions

Basic Arithmetic operators like *, /, +, - can be used

Example:1

```
SELECT last_name, salary, salary+300 FROM employees;
```

Example:2

```
SELECT last_name, salary, 12*salary+100 FROM employees;
```

The statement is not same as

```
SELECT last_name, salary, 12*(salary+100) FROM employees;
```

Example:3

SELECT last_name, job_id, salary, commission_pct FROM employees;

Example:4

SELECT last_name, job_id, salary, 12*salary*commission_pct FROM employees;

Using Column Alias

- To rename a column heading with or without AS keyword.

Example:1

SELECT last_name AS Name
FROM employees;

Example: 2

SELECT last_name "Name" salary*12 "Annual Salary "
FROM employees;

Concatenation Operator

- Concatenates columns or character strings to other columns
- Represented by two vertical bars (||)
- Creates a resultant column that is a character expression

Example:

SELECT last_name||job_id AS "EMPLOYEES JOB" FROM employees;

Using Literal Character String

- A literal is a character, a number, or a date included in the SELECT list.
- Date and character literal values must be enclosed within single quotation marks.

Example:

SELECT last_name||'is a'||job_id AS "EMPLOYEES JOB" FROM employees;

Eliminating Duplicate Rows

- Using DISTINCT keyword.

Example:

SELECT DISTINCT department_id FROM employees;

Displaying Table Structure

- Using DESC keyword.

Syntax

DESC table_name;

Example:

DESC employees;

Find the Solution for the following;

True OR False

1. The following statement executes successfully.

Identify the Errors

SELECT employee_id, last_name
sal*12 ANNUAL SALARY

FROM employees;

Queries

```
SELECT employee_id, last_name, sal*12 AS "ANNUAL SALARY"  
FROM employees;
```

Output:

EMPLOYEE_ID	LAST_NAME	ANNUAL SALARY
101	King	288000

2. Show the structure of departments the table. Select all the data from it.

DESC departments;

```
SELECT * FROM departments;
```

Output (Structure Example):

COLUMN_NAME	DATA_TYPE	NULLABLE
DEPARTMENT_ID	NUMBER	No
DEPARTMENT_NAME	VARCHAR2	No
MANAGER_ID	NUMBER	Yes
LOCATION_ID	NUMBER	Yes

Output (Data Example):

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700

3. Create a query to display the last name, job code, hire date, and employee number for each employee, with employee number appearing first.

```
SELECT employee_id, last_name, job_id, hire_date  
FROM employees;
```

Output:

EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE
101	King	AD_PRES	17-JUN-1987

4. Provide an alias STARTDATE for the hire date.

```
SELECT employee_id, last_name, hire_date AS "STARTDATE"  
FROM employees;
```

Output:

EMPLOYEE_ID	LAST_NAME	STARTDATE
-------------	-----------	-----------

5. Create a query to display unique job codes from the employee table.

```
SELECT DISTINCT job_id  
FROM employees;
```

Output:

```
JOB_ID  
AD_PRES  
ST_CLERK  
IT_PROG
```

6. Display the last name concatenated with the job ID , separated by a comma and space, and name the column EMPLOYEE and TITLE.

```
SELECT last_name || ', ' || job_id AS "EMPLOYEE AND TITLE"  
FROM employees;
```

Output:

```
EMPLOYEE AND TITLE  
King, AD_PRES  
Kochhar, AD_VP
```

7. Create a query to display all the data from the employees table. Separate each column by a comma. Name the column THE_OUTPUT.

```
SELECT employee_id || ', ' || first_name || ', ' || last_name || ', ' ||  
job_id || ', ' || salary AS "THE_OUTPUT"  
FROM employees;
```

Output:

```
THE_OUTPUT  
101, Steven, King, AD_PRES, 24000
```