| Ex.No.: 3 | | WRITING BASIC SQL SELECT STATEMENTS |
|---|---|---|
| Date: | 2/8/24 | |

## 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.

**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 deparment_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;

SELECT employee_id, last_name sal*12 AS ANNUAL_SALARY FROM employees;

**Queries**

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

Desc Employees table;

Select * from Employees_table;

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_table;
```

4. Provide an alias STARTDATE for the hire date.

```
select hire_date as start_date from Employees_table;
```

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

```
Select distinct job_id from Employees_table;
```

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 "Employees Title"
from Employees_table;
```

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 || ', ' || last_name || ', ' || salary AS
THE_OUTPUT FROM Employees.
```

| Evaluation Procedure | Marks awarded |
|---|---|
| Query(5) | 5 |
| Execution (5) | 5 |
| Viva(5) | 5 |
| Total (15) | 15 |
| Faculty Signature | |