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—Suppr
ess 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 sued 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;

FROM employees;

Queries

False. ANNUAL SALARY is not surrounded by quotes which can lead to error. AS keyword is missing before ANNUAL SALARY

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

```
DESC departments;
SELECT * FROM departments;
```

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;
```

4. Provide an alias STARTDATE for the hire date.

```
SELECT hire_date AS `STARTDATE` FROM employees;
```

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

```
SELECT UNIQUE job_id FROM employees;
```

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;
```

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

C. Sruthi 21/9/24

```
SELECT employee_id || ', ' || first_name || ', ' || last_name
  || ', ' || email || ', ' || phone_number || ', ' || hire_date
  || ', ' || job_id || ', ' || salary || ', ' || commission_pct
  || ', ' || manager_id || ', ' || department_id AS 'THE OUTPUT'
FROM employees
```