Title: Writing Basic SQL Select Statement

Author: S. Santhosh Kumar

Objectives

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

- 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

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

- **DISTINCT**: Suppresses duplicates.
- **Alias**: Gives selected columns different headings.

Examples:

1.
   ```
   sql
   SELECT * FROM departments;
   ```

2.
   ```
   sql
   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.

Examples:

1. **sql**
   ```
   SELECT last_name, salary, salary+300 FROM employees;
   ```
2. **sql**
   ```
   SELECT last_name, salary, 12*salary+100 FROM employees;
   ```
   - **This statement is different from:**

     sql
     ```
     SELECT last_name, salary, 12*(salary+100) FROM employees;
     ```
3. **sql**
   ```
   SELECT last_name, job_id, salary, commission_pct FROM employees;
   ```
4. **sql**
   ```
   SELECT last_name, job_id, salary, 12*salary*commission_pct FROM
   employees;
   ```

Using Column Alias

To rename a column heading with or without the `AS` keyword.

Examples:

1. **sql**
   ```
   SELECT last_name AS Name FROM employees;
   ```
2. **sql**
   ```
   SELECT last_name AS "Name", salary*12 AS "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:

sql
```
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 the `DISTINCT` keyword.

Example:

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

Displaying Table Structure

Using the `DESC` keyword.

Syntax:

```
DESC table_name;
```

Example:

```
DESC employees;
```

Exercises with Input and Output

Sample Data

Let's assume the `employees` table contains the following data:

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
101	John	Smith	jsmith@xyz.com	123-456-7890	2020-01-15	IT_PROG	6000	0.10	100	60
102	Jane	Doe	jdoe@xyz.com	987-654-3210	2019-02-18	HR_REP	4500	0.15	101	40
103	Emily	Johnson	ejohnson@xyz.com	555-555-5555	2021-03-20	IT_PROG	5200	0.20	100	60
104	Michael	Brown	mbrown@xyz.com	111-222-3333	2018-04-22	AD_ASST	3000	0.05	102	10

QUERIES IN SQL

1. True or False: Identify the Errors

Input:

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

Errors Identified:

- Missing comma between `last_name` and `sal*12 ANNUAL SALARY`.
- The correct statement should be:

  ```
  SELECT employee_id, last_name, salary*12 AS ANNUAL_SALARY FROM
  employees;
  ```

Correct Output:

employee_id	last_name	ANNUAL_SALARY
101	Smith	72000
102	Doe	54000
103	Johnson	62400
104	Brown	36000

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

Input:

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

Output:

```
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| department_id     | int(11)     | NO   | PRI | NULL    |       |
| department_name   | varchar(30) | NO   |     | NULL    |       |
| manager_id        | int(11)     | YES  |     | NULL    |       |
| location_id       | int(11)     | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
```

Assuming the `departments` table contains:

department_id	department_name	manager_id	location_id
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	202	1900
40	Human Resources	203	2000
60	IT	204	2100

Output:

department_id	department_name	manager_id	location_id
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	202	1900
40	Human Resources	203	2000
60	IT	204	2100

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

Input:

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

Output:

employee_id	last_name	job_id	hire_date
101	Smith	IT_PROG	2020-01-15
102	Doe	HR_REP	2019-02-18
103	Johnson	IT_PROG	2021-03-20
104	Brown	AD_ASST	2018-04-22

4. Provide an alias STARTDATE for the hire date.

Input:

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

Output:

START DATE
2020-01-15
2019-02-18
2021-03-20
2018-04-22

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

Input:

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

Output:

job_id
IT_PROG
HR_REP
AD_ASST

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

Input:

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

Output:

EMPLOYEE AND TITLE
Smith, IT_PROG
Doe, HR_REP
Johnson, IT_PROG

EMPLOYEE AND TITLE
Brown, AD_ASST

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

Input:

```
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;
```

Output:

THE_OUTPUT
101, John, Smith, jsmith@xyz.com, 123-456-7890, 2020-01-15, IT_PROG, 6000, 0.10, 100, 60
102, Jane, Doe, jdoe@xyz.com, 987-654-3210, 2019-02-18, HR_REP, 4500, 0.15, 101, 40
103, Emily, Johnson, ejohnson@xyz.com, 555-555-5555, 2021-03-20, IT_PROG, 5200, 0.20, 100, 60
104, Michael, Brown, mbrown@xyz.com, 111-222-3333, 2018-04-22, AD_ASST, 3000, 0.05, 102, 10