EXPERIMENT - II
BASIC SQL QUERIES - I

July 29, 2019

ADITHYA D RAJAGOPAL
ROLL NO : 9
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING TRIVANDRUM

AIM

To study the basic SQL queries such as:

1. SELECT
2. INSERT
3. UPDATE
4. DELETE

SELECT STATEMENT

The SELECT statement is used to display the selected attributes of a table.

SYNTAX

SELECT column1,column2,...,columnN FROM <TableName> ;

INSERT STATEMENT

The INSERT statement is used to insert values into a table.

SYNTAX

INSERT INTO <TableName> (column1,column2,...,columnN)
VALUES (value1,value2,..valueN);

UPDATE STATEMENT

The UPDATE statement is used to modify the existing records in a table.

SYNTAX

UPDATE <TableName>
SET column1 = value1, column2 = value2...., columnN = valueN
WHERE [condition];

DELETE STATEMENT

The DELETE statement is used to delete the existing records from a table.

SYNTAX

DELETE FROM <TableName>
WHERE [condition];

QUESTIONS

Create a table named Employee and populate the table as shown below.

EMP_ID	EMP_NAME	DEPT	SALARY
1	Michael	Production	$2500
2	Joe	Production	$2500
3	Smith	Sales	$2250
4	David	Marketing	$2900
5	Richard	Sales	$1600
6	Jessy	Marketing	$1800
7	Jane	Sales	$2000
8	Janet	Production	$3000
9	Neville	Marketing	$2750
10	Richardson	Sales	$1800

```
postgres=# CREATE TABLE Employee (
postgres(# EMP_ID INT PRIMARY KEY NOT NULL,
postgres(# EMP_NAME TEXT NOT NULL,
postgres(# DEPT TEXT NOT NULL,
postgres(# SALARY TEXT);
CREATE TABLE
postgres=# INSERT INTO Employee VALUES (1,'Michael','Production','$2500');
INSERT 0 1
postgres=# INSERT INTO Employee VALUES (2,'Joe','Production','$2500');
INSERT 0 1
postgres=# INSERT INTO Employee VALUES (3,'Smith','Sales','$2250');
INSERT 0 1
postgres=# INSERT INTO Employee VALUES (4,'David','Marketing','$2900');
INSERT 0 1
postgres=# INSERT INTO Employee VALUES (5,'Richard','Sales','$1600');
INSERT 0 1
postgres=# INSERT INTO Employee VALUES (6,'Jessy','Marketing','$1800');
INSERT 0 1
postgres=# INSERT INTO Employee VALUES (7,'Jane','Sales','$2000');
INSERT 0 1
postgres=# INSERT INTO Employee VALUES (8,'Janet','Production','$3000');
INSERT 0 1
postgres=# INSERT INTO Employee VALUES (9,'Neville','Marketing','$2750');
INSERT 0 1
postgres=# INSERT INTO Employee VALUES (10,'Richardson','Sales','$1800');
INSERT 0 1
postgres=#
```

1. Display the details of all the employees.

```
postgres=# SELECT * FROM Employee;
 emp_id |  emp_name  |    dept    | salary
--------+------------+------------+--------
      1 | Michael    | Production | $2500
      2 | Joe        | Production | $2500
      3 | Smith      | Sales      | $2250
      4 | David      | Marketing  | $2900
      5 | Richard    | Sales      | $1600
      6 | Jessy      | Marketing  | $1800
      7 | Jane       | Sales      | $2000
      8 | Janet      | Production | $3000
      9 | Neville    | Marketing  | $2750
     10 | Richardson | Sales      | $1800
(10 rows)

postgres=#
```

2. Display the names and idâĂŹs of all employees.

```
postgres=# SELECT EMP_ID,EMP_NAME FROM Employee;
 emp_id |  emp_name
--------+------------
      1 | Michael
      2 | Joe
      3 | Smith
      4 | David
      5 | Richard
      6 | Jessy
      7 | Jane
      8 | Janet
      9 | Neville
     10 | Richardson
(10 rows)

postgres=#
```

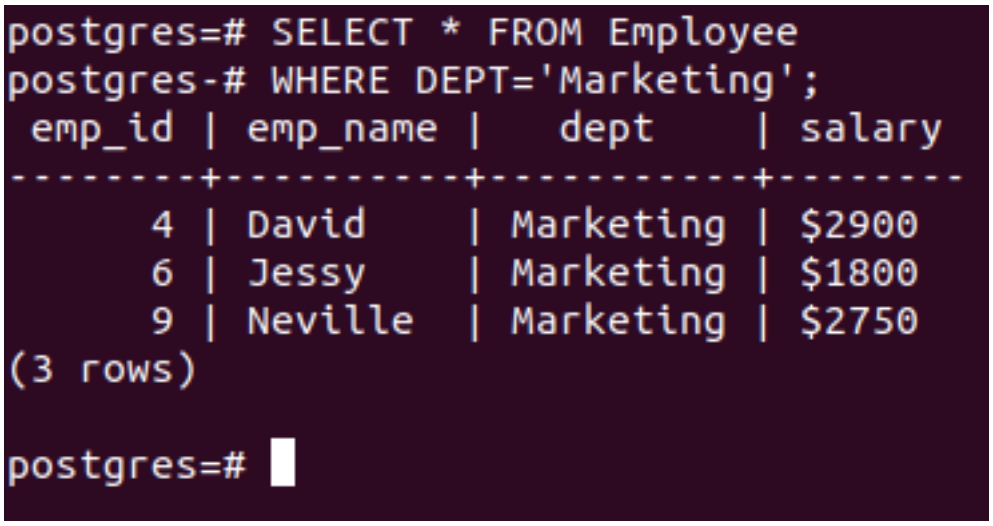
3. Delete the entry corresponding to employee id:10.

```
postgres=# DELETE FROM Employee
postgres-# WHERE EMP_ID=10;
DELETE 1
postgres=# SELECT * FROM Employee;
 emp_id | emp_name |    dept    | salary
--------+----------+------------+--------
      1 | Michael  | Production | $2500
      2 | Joe      | Production | $2500
      3 | Smith    | Sales      | $2250
      4 | David    | Marketing  | $2900
      5 | Richard  | Sales      | $1600
      6 | Jessy    | Marketing  | $1800
      7 | Jane     | Sales      | $2000
      8 | Janet    | Production | $3000
      9 | Neville  | Marketing  | $2750
(9 rows)

postgres=#
```

4. Insert a new tuple to the table. The salary field of the new employee should be kept NULL.

```
postgres=# INSERT INTO Employee(EMP_ID,EMP_NAME,DEPT) VALUES (10,'Richardson','Sales');
INSERT 0 1
postgres=# SELECT * FROM Employee;
 emp_id |  emp_name  |    dept    | salary
--------+------------+------------+--------
      1 | Michael    | Production | $2500
      2 | Joe        | Production | $2500
      3 | Smith      | Sales      | $2250
      4 | David      | Marketing  | $2900
      5 | Richard    | Sales      | $1600
      6 | Jessy      | Marketing  | $1800
      7 | Jane       | Sales      | $2000
      8 | Janet      | Production | $3000
      9 | Neville    | Marketing  | $2750
     10 | Richardson | Sales      |
(10 rows)

postgres=#
```

5. Find the details of all employees working in the marketing department.



6. Add the salary details of the newly added employee.

```
postgres=# UPDATE Employee
postgres-# SET SALARY='$1900'
postgres-# WHERE EMP_ID=10;
UPDATE 1
postgres=# SELECT * FROM Employee;
 emp_id |  emp_name  |    dept    | salary
--------+------------+------------+--------
      1 | Michael    | Production | $2500
      2 | Joe        | Production | $2500
      3 | Smith      | Sales      | $2250
      4 | David      | Marketing  | $2900
      5 | Richard    | Sales      | $1600
      6 | Jessy      | Marketing  | $1800
      7 | Jane       | Sales      | $2000
      8 | Janet      | Production | $3000
      9 | Neville    | Marketing  | $2750
     10 | Richardson | Sales      | $1900
(10 rows)

postgres=#
```

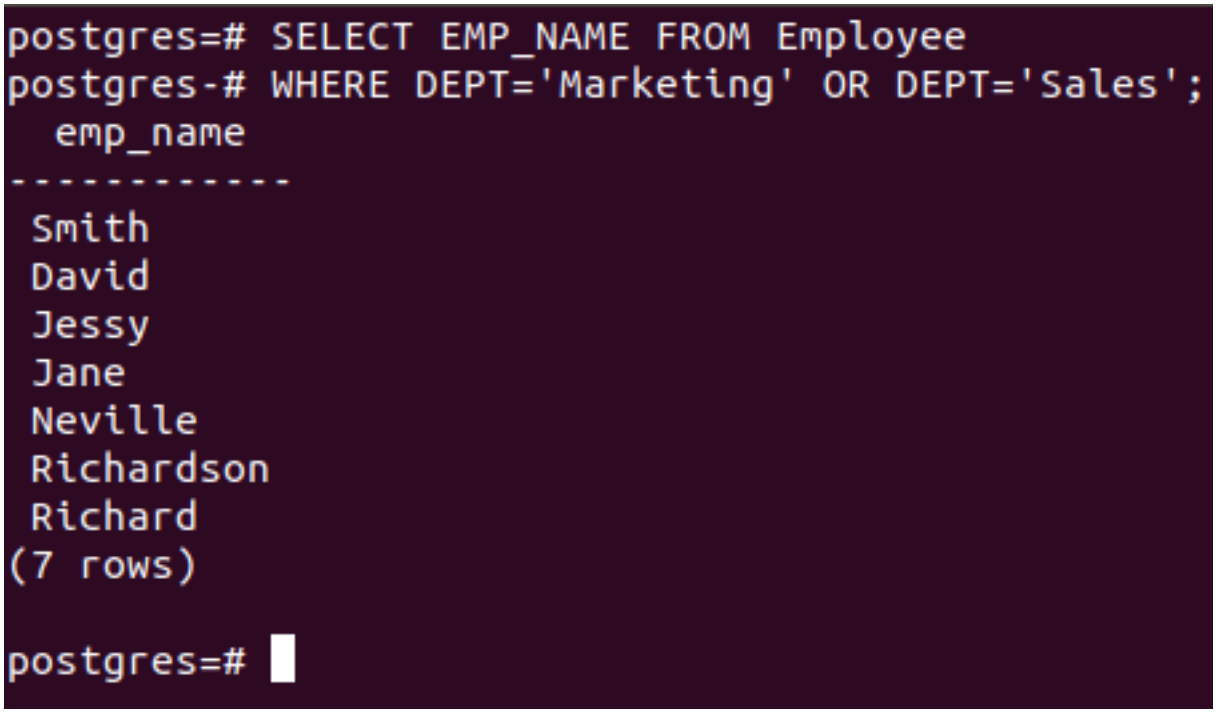
7. Update the salary of Richard to $1900.

```
postgres=# UPDATE Employee
postgres-# SET SALARY='$1900'
postgres-# WHERE EMP_NAME='Richard';
UPDATE 1
postgres=# SELECT * FROM Employee;
 emp_id |  emp_name  |    dept    | salary
--------+------------+------------+--------
      1 | Michael    | Production | $2500
      2 | Joe        | Production | $2500
      3 | Smith      | Sales      | $2250
      4 | David      | Marketing  | $2900
      6 | Jessy      | Marketing  | $1800
      7 | Jane       | Sales      | $2000
      8 | Janet      | Production | $3000
      9 | Neville    | Marketing  | $2750
     10 | Richardson | Sales      | $1900
      5 | Richard    | Sales      | $1900
(10 rows)

postgres=#
```

8. Find the details of all employees who are working for marketing and has a salary greater than $2000.

```
postgres=# SELECT * FROM Employee
postgres-# WHERE DEPT='Marketing'
postgres-# AND SALARY>'$2000';
 emp_id | emp_name |   dept    | salary
--------+----------+-----------+--------
      4 | David    | Marketing | $2900
      9 | Neville  | Marketing | $2750
(2 rows)

postgres=#
```

9. List the names of all employees working in the sales department and marketing department.



10. List the names and department of all employees whose salary is between $2300 and $3000.

```
postgres=# SELECT EMP_NAME,DEPT FROM Employee
postgres-# WHERE SALARY>='$2300' AND SALARY<='$3000';
 emp_name |    dept
----------+------------
 Michael  | Production
 Joe      | Production
 David    | Marketing
 Janet    | Production
 Neville  | Marketing
(5 rows)

postgres=#
```

11. Update the salary of all employees working in production department 12%.

```
postgres=# UPDATE Employee
SET SALARY=1.12*SALARY
WHERE DEPT='Production';
UPDATE 3
postgres=# SELECT * FROM Employee;
 emp_id |  emp_name  |    dept    | salary
--------+------------+------------+--------
      4 | David      | Marketing  | $2900
      6 | Jessy      | Marketing  | $1800
      7 | Jane       | Sales      | $2000
      9 | Neville    | Marketing  | $2750
     10 | Richardson | Sales      | $1900
      5 | Richard    | Sales      | $1900
      3 | Smith      | Sales      | $2250
      1 | Michael    | Production | $2800
      2 | Joe        | Production | $2800
      8 | Janet      | Production | $3360
(10 rows)

postgres=#
```

12. Display the names of all employees whose salary is less than $2000 or working for sales department.

```
postgres=# SELECT EMP_NAME FROM Employee
postgres-# WHERE SALARY<='$2000' AND DEPT='Sales';
  emp_name
------------
 Jane
 Richardson
 Richard
(3 rows)

postgres=#
```

RESULT

The query was executed and the output was obtained.