# NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

Azad Hind Fauj Marg
Sector - 3 , Dwarka
New Delhi

Practical File

DBMS CACSC05

Submitted by:
Name: Piyush Tiwari
Roll No: 2022UCA1912

Submitted to:
Mrs. Riya Mangal

# Assignment 1

**Creating the table**:

CREATE TABLE Employee(Emp_id INT PRIMARY KEY, First_name VARCHAR(30), Last_name VARCHAR(30), Phone_No VARCHAR(13), Hire_date DATE, Job_id INT, Emp_Salary INT, Comission_Pct INT, Manager_id INT, Department_id INT);

INSERT INTO Employee VALUES(47401, "Rama", "Rao", "8965324170", "2011-09-12", 301, 60000, 601, 100, 60);

INSERT INTO Employee VALUES(47402, "Ranga", "Reddy", "7020321450", "2004-11-11", 302, 56464, 602, 101, 70);

INSERT INTO Employee values(47403,'Raja',"Shekhar",9848002255,'2004-08-12',303,58451,603,103,80);

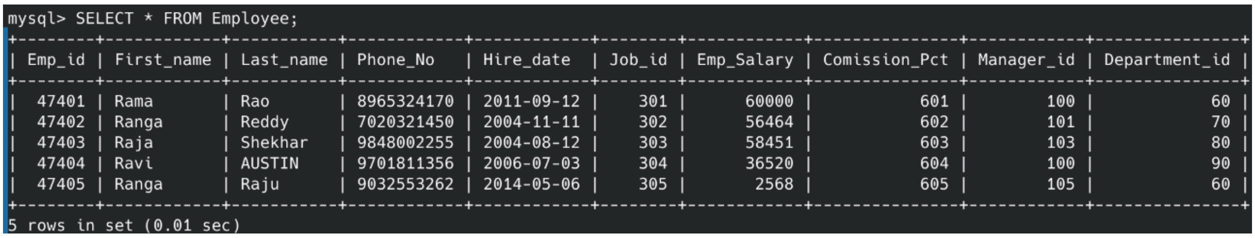
INSERT INTO Employee values(47404,'Ravi','AUSTIN',9701811356,'2006-07-03',304,36520,604,100,90);

INSERT INTO Employee values(47405,'Ranga','Raju',9032553262,'2014-05-06',305,2568,605,105,60);

**Queries**:

A.

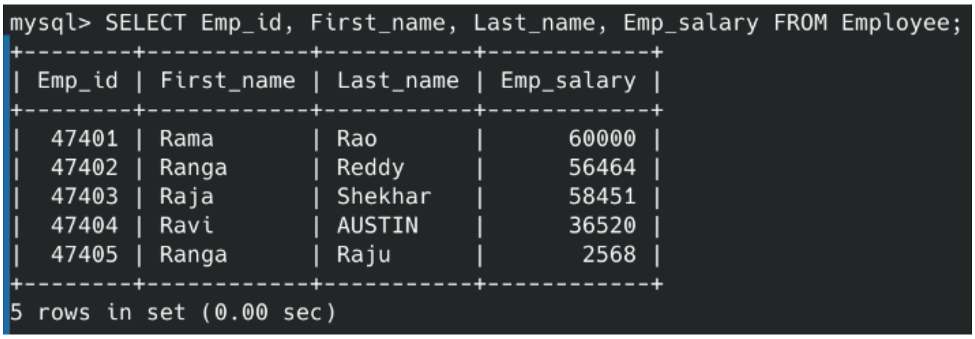
SELECT * FROM Employee;



```
mysql> SELECT * FROM Employee;
+--------+------------+-----------+------------+------------+--------+------------+---------------+------------+---------------+
| Emp_id | First_name | Last_name | Phone_No   | Hire_date  | Job_id | Emp_Salary | Comission_Pct | Manager_id | Department_id |
+--------+------------+-----------+------------+------------+--------+------------+---------------+------------+---------------+
|  47401 | Rama       | Rao       | 8965324170 | 2011-09-12 |    301 |      60000 |           601 |        100 |            60 |
|  47402 | Ranga      | Reddy     | 7020321450 | 2004-11-11 |    302 |      56464 |           602 |        101 |            70 |
|  47403 | Raja       | Shekhar   | 9848002255 | 2004-08-12 |    303 |      58451 |           603 |        103 |            80 |
|  47404 | Ravi       | AUSTIN    | 9701811356 | 2006-07-03 |    304 |      36520 |           604 |        100 |            90 |
|  47405 | Ranga      | Raju      | 9032553262 | 2014-05-06 |    305 |       2568 |           605 |        105 |            60 |
+--------+------------+-----------+------------+------------+--------+------------+---------------+------------+---------------+
5 rows in set (0.01 sec)
```

B.

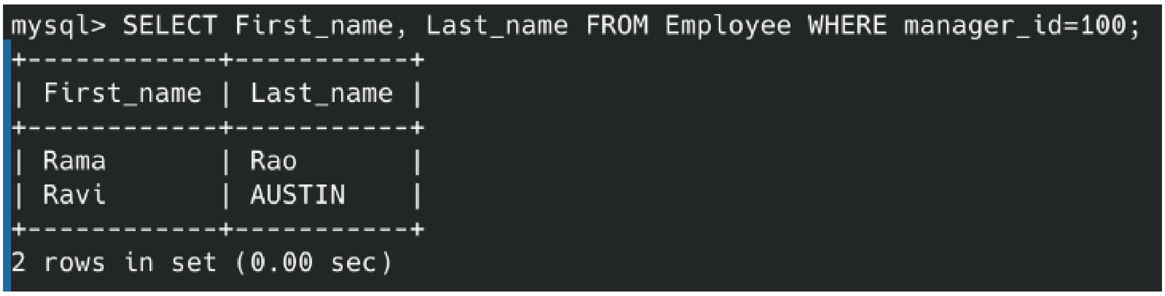
SELECT Emp_id, First_name, Last_name, Emp_salary FROM Employee;



```
mysql> SELECT Emp_id, First_name, Last_name, Emp_salary FROM Employee;
+--------+------------+-----------+------------+
| Emp_id | First_name | Last_name | Emp_salary |
+--------+------------+-----------+------------+
|  47401 | Rama       | Rao       |      60000 |
|  47402 | Ranga      | Reddy     |      56464 |
|  47403 | Raja       | Shekhar   |      58451 |
|  47404 | Ravi       | AUSTIN    |      36520 |
|  47405 | Ranga      | Raju      |       2568 |
+--------+------------+-----------+------------+
5 rows in set (0.00 sec)
```

C.

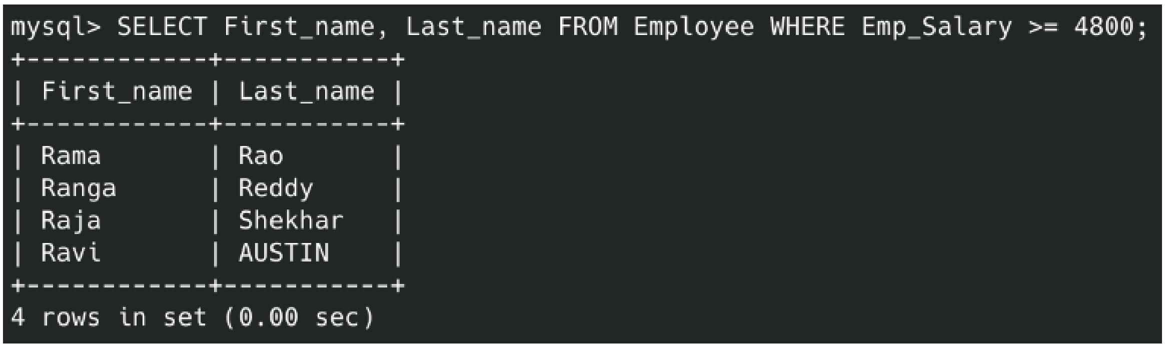
SELECT First_name, Last_name FROM Employee WHERE manager_id=100;



```
mysql> SELECT First_name, Last_name FROM Employee WHERE manager_id=100;
+------------+-----------+
| First_name | Last_name |
+------------+-----------+
| Rama       | Rao       |
| Ravi       | AUSTIN    |
+------------+-----------+
2 rows in set (0.00 sec)
```

D.

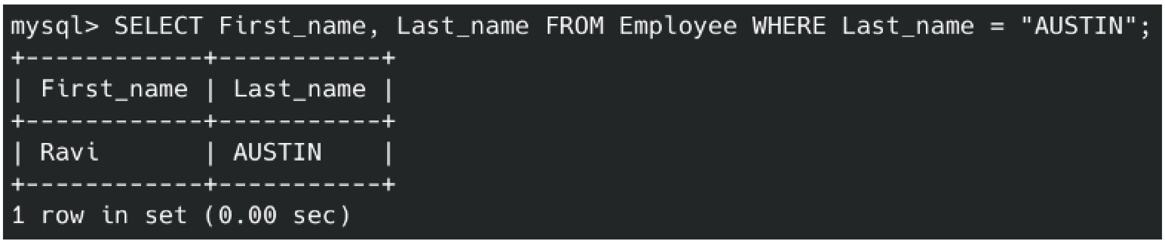
SELECT First_name, Last_name FROM Employee WHERE Emp_Salary >= 4800;



```
mysql> SELECT First_name, Last_name FROM Employee WHERE Emp_Salary >= 4800;
+------------+-----------+
| First_name | Last_name |
+------------+-----------+
| Rama       | Rao       |
| Ranga      | Reddy     |
| Raja       | Shekhar   |
| Ravi       | AUSTIN    |
+------------+-----------+
4 rows in set (0.00 sec)
```

E.

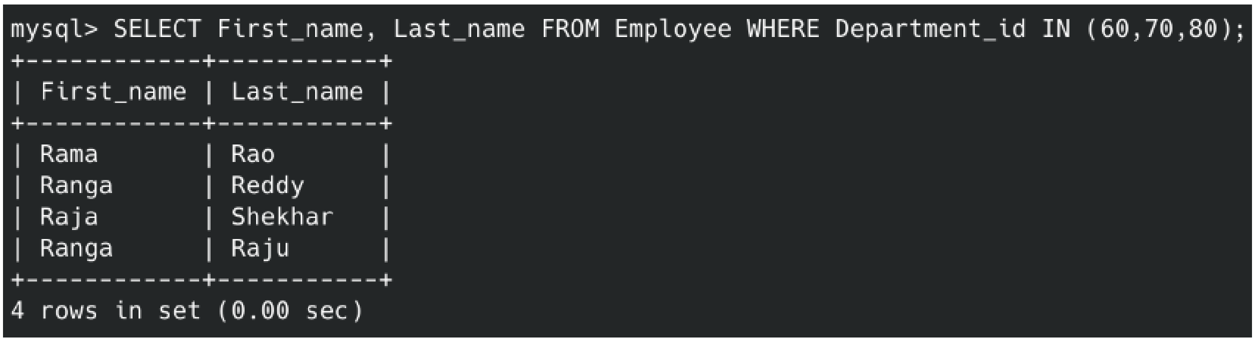
SELECT First_name, Last_name FROM Employee WHERE Last_name = "AUSTIN";



```
mysql> SELECT First_name, Last_name FROM Employee WHERE Last_name = "AUSTIN";
+------------+-----------+
| First_name | Last_name |
+------------+-----------+
| Ravi       | AUSTIN    |
+------------+-----------+
1 row in set (0.00 sec)
```

F.

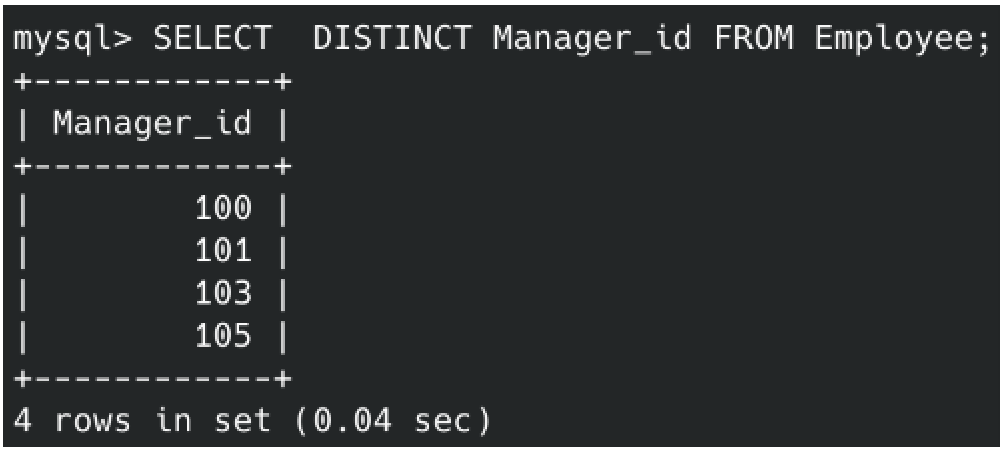
SELECT First_name, Last_name FROM Employee WHERE Department_id IN (60,70,80);



```
mysql> SELECT First_name, Last_name FROM Employee WHERE Department_id IN (60,70,80);
+------------+-----------+
| First_name | Last_name |
+------------+-----------+
| Rama       | Rao       |
| Ranga      | Reddy     |
| Raja       | Shekhar   |
| Ranga      | Raju      |
+------------+-----------+
4 rows in set (0.00 sec)
```

G.

SELECT  DISTINCT Manager_id FROM Employee;



```
mysql> SELECT  DISTINCT Manager_id FROM Employee;
+------------+
| Manager_id |
+------------+
|        100 |
|        101 |
|        103 |
|        105 |
+------------+
4 rows in set (0.04 sec)
```

# Assignment 2

**Creating the table**:

CREATE TABLE CUSTOMERS(ID int PRIMARY KEY, name varchar(30), age int, address varchar(50), salary int);

INSERT INTO CUSTOMERS VALUES(1, 'Chandan', 24, 'Indore', 10000);

INSERT INTO CUSTOMERS VALUES(2, 'Niranjan', 22, 'Hyderabad', 4500);

INSERT INTO CUSTOMERS VALUES(3, 'Raushan', 27, 'Bhopal', 8500);

INSERT INTO CUSTOMERS VALUES(4, 'Vikas', 25, 'Mumbai', 6500);

INSERT INTO CUSTOMERS VALUES(5, 'Komal', 23, 'Kota', 2000);

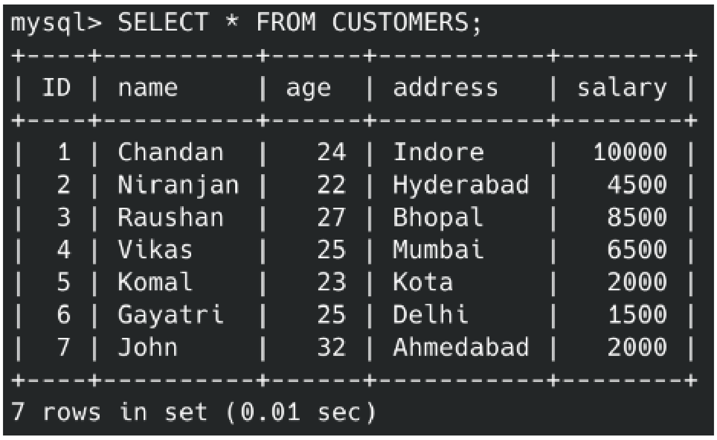
INSERT INTO CUSTOMERS VALUES(6, 'Gayatri', 25, 'Delhi', 1500);

INSERT INTO CUSTOMERS VALUES(7, 'John', 32, 'Ahmedabad', 2000);

**Queries**:

A.

SELECT * FROM CUSTOMERS;



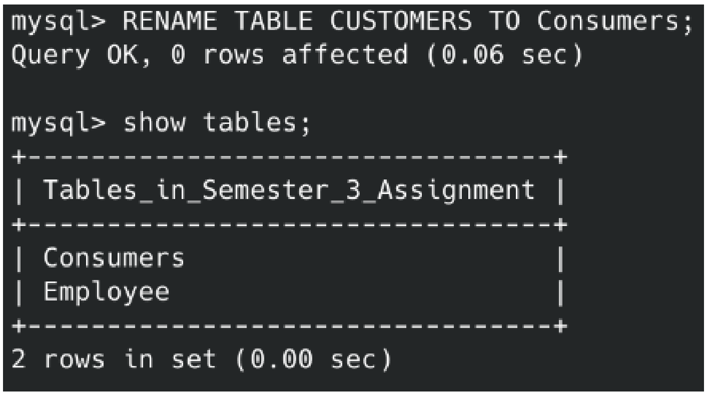
```
mysql> SELECT * FROM CUSTOMERS;
+----+----------+------+-----------+--------+
| ID | name     | age  | address   | salary |
+----+----------+------+-----------+--------+
|  1 | Chandan  |   24 | Indore    |  10000 |
|  2 | Niranjan |   22 | Hyderabad |   4500 |
|  3 | Raushan  |   27 | Bhopal    |   8500 |
|  4 | Vikas    |   25 | Mumbai    |   6500 |
|  5 | Komal    |   23 | Kota      |   2000 |
|  6 | Gayatri  |   25 | Delhi     |   1500 |
|  7 | John     |   32 | Ahmedabad |   2000 |
+----+----------+------+-----------+--------+
7 rows in set (0.01 sec)
```

B.

ALTER TABLE CUSTOMERS RENAME TO Consumers;

OR

RENAME TABLE CUSTOMERS TO Consumers;

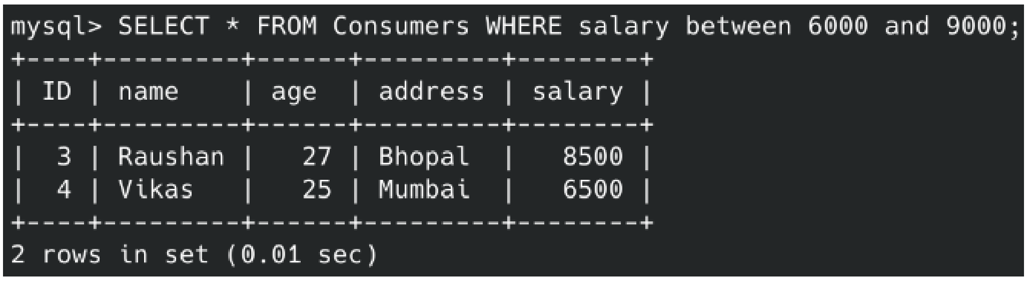


```
mysql> RENAME TABLE CUSTOMERS TO Consumers;
Query OK, 0 rows affected (0.06 sec)

mysql> show tables;
+-------------------------------+
| Tables_in_Semester_3_Assignment |
+-------------------------------+
| Consumers                     |
| Employee                      |
+-------------------------------+
2 rows in set (0.00 sec)
```

C.

SELECT * FROM Consumers WHERE salary between 6000 and 9000;

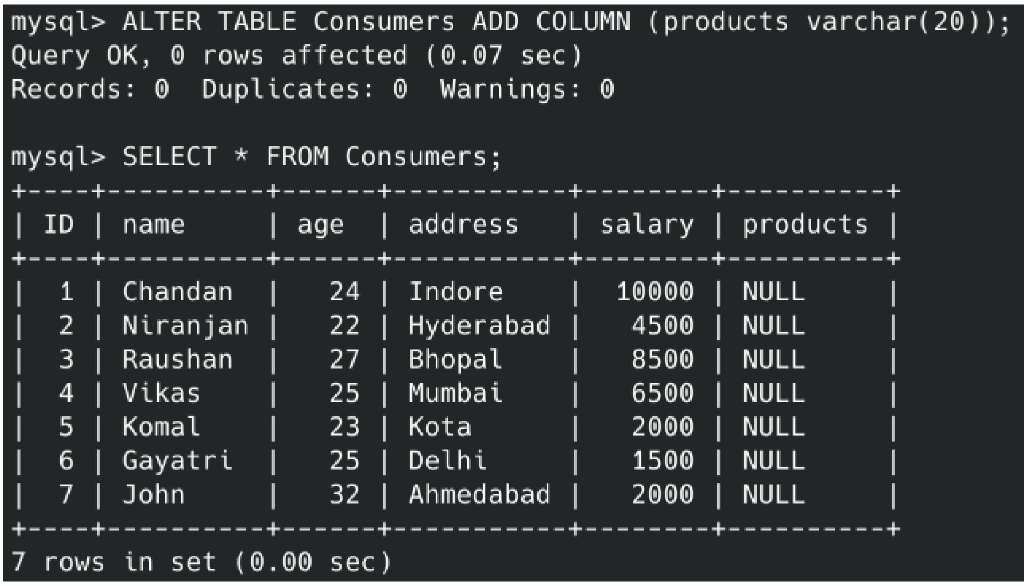


```
mysql> SELECT * FROM Consumers WHERE salary between 6000 and 9000;
+----+---------+------+---------+--------+
| ID | name    | age  | address | salary |
+----+---------+------+---------+--------+
|  3 | Raushan |   27 | Bhopal  |   8500 |
|  4 | Vikas   |   25 | Mumbai  |   6500 |
+----+---------+------+---------+--------+
2 rows in set (0.01 sec)
```

D.

ALTER TABLE Consumers ADD COLUMN (products varchar(20));

SELECT * FROM Consumers;



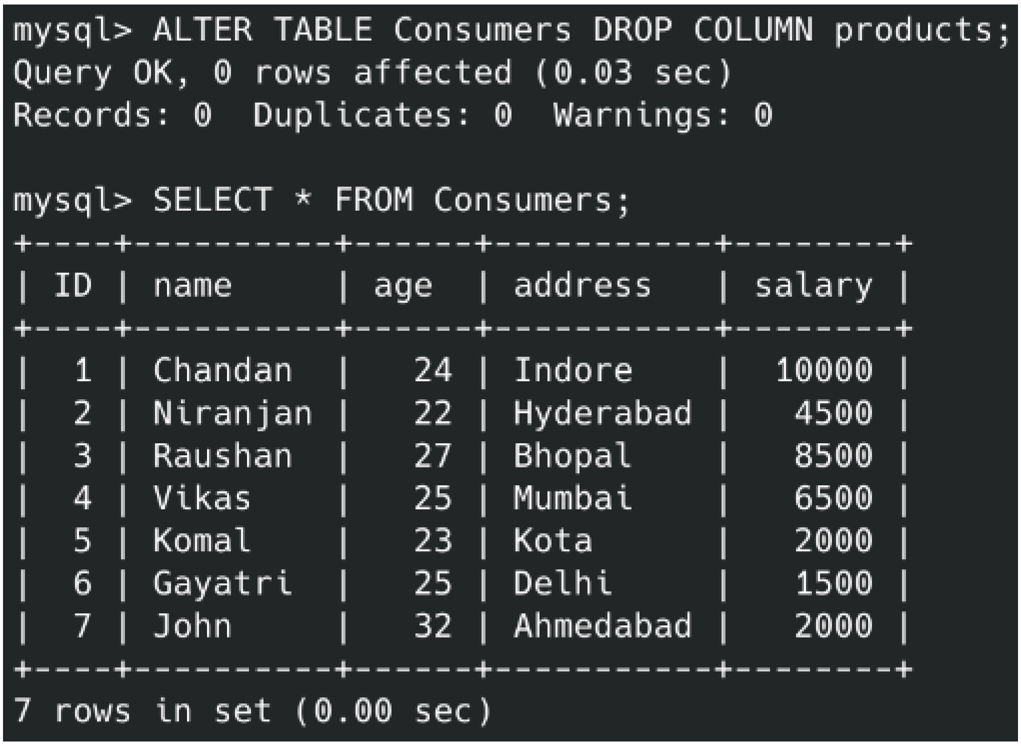
```
mysql> ALTER TABLE Consumers ADD COLUMN (products varchar(20));
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Consumers;
+----+----------+------+-----------+--------+----------+
| ID | name     | age  | address   | salary | products |
+----+----------+------+-----------+--------+----------+
|  1 | Chandan  |   24 | Indore    |  10000 | NULL     |
|  2 | Niranjan |   22 | Hyderabad |   4500 | NULL     |
|  3 | Raushan  |   27 | Bhopal    |   8500 | NULL     |
|  4 | Vikas    |   25 | Mumbai    |   6500 | NULL     |
|  5 | Komal    |   23 | Kota      |   2000 | NULL     |
|  6 | Gayatri  |   25 | Delhi     |   1500 | NULL     |
|  7 | John     |   32 | Ahmedabad |   2000 | NULL     |
+----+----------+------+-----------+--------+----------+
7 rows in set (0.00 sec)
```

E.

ALTER TABLE Consumers DROP COLUMN products;

SELECT * FROM Consumers;



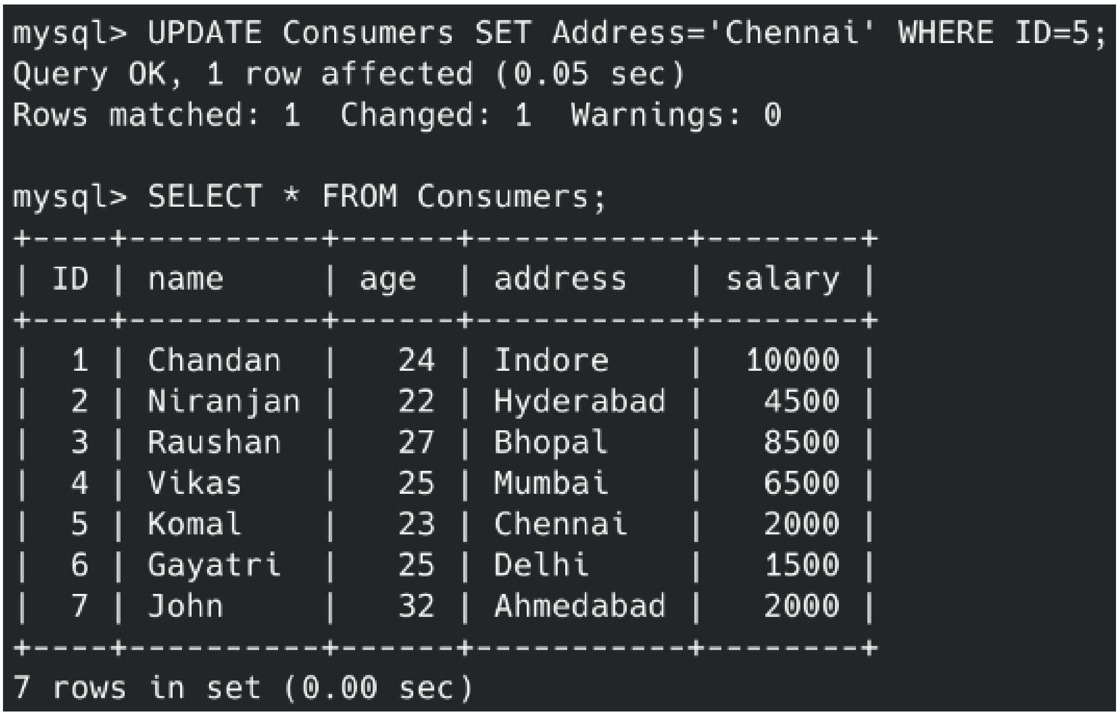
```
mysql> ALTER TABLE Consumers DROP COLUMN products;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Consumers;
+----+----------+------+-----------+--------+
| ID | name     | age  | address   | salary |
+----+----------+------+-----------+--------+
|  1 | Chandan  |   24 | Indore    |  10000 |
|  2 | Niranjan |   22 | Hyderabad |   4500 |
|  3 | Raushan  |   27 | Bhopal    |   8500 |
|  4 | Vikas    |   25 | Mumbai    |   6500 |
|  5 | Komal    |   23 | Kota      |   2000 |
|  6 | Gayatri  |   25 | Delhi     |   1500 |
|  7 | John     |   32 | Ahmedabad |   2000 |
+----+----------+------+-----------+--------+
7 rows in set (0.00 sec)
```

F.

UPDATE Consumers SET Address='Chennai' WHERE ID=5;

SELECT * FROM Consumers;

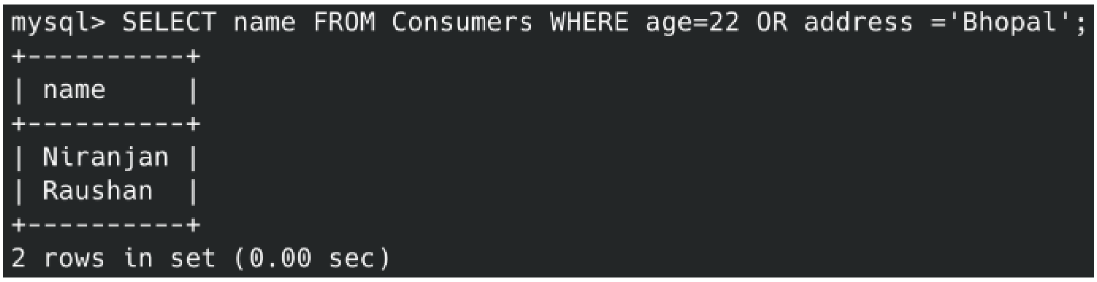


```
mysql> UPDATE Consumers SET Address='Chennai' WHERE ID=5;
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Consumers;
+----+----------+------+-----------+--------+
| ID | name     | age  | address   | salary |
+----+----------+------+-----------+--------+
|  1 | Chandan  |   24 | Indore    |  10000 |
|  2 | Niranjan |   22 | Hyderabad |   4500 |
|  3 | Raushan  |   27 | Bhopal    |   8500 |
|  4 | Vikas    |   25 | Mumbai    |   6500 |
|  5 | Komal    |   23 | Chennai   |   2000 |
|  6 | Gayatri  |   25 | Delhi     |   1500 |
|  7 | John     |   32 | Ahmedabad |   2000 |
+----+----------+------+-----------+--------+
7 rows in set (0.00 sec)
```

G.

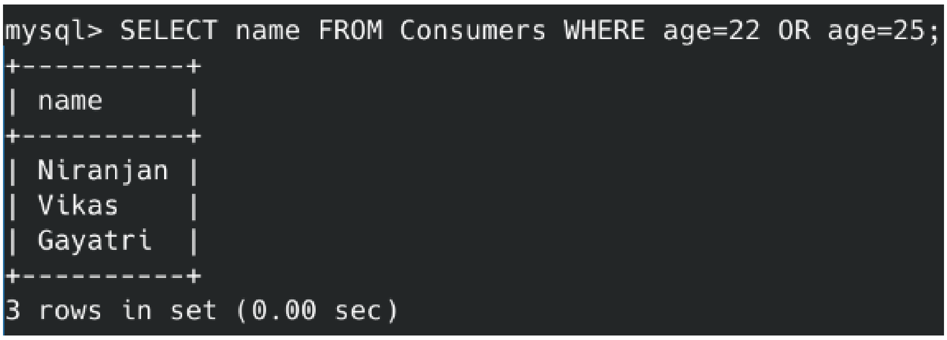
SELECT name FROM Consumers WHERE age=22 OR address ='Bhopal';



```
mysql> SELECT name FROM Consumers WHERE age=22 OR address ='Bhopal';
+----------+
| name     |
+----------+
| Niranjan |
| Raushan  |
+----------+
2 rows in set (0.00 sec)
```

H.

SELECT name FROM Consumers WHERE age=22 OR age=25;

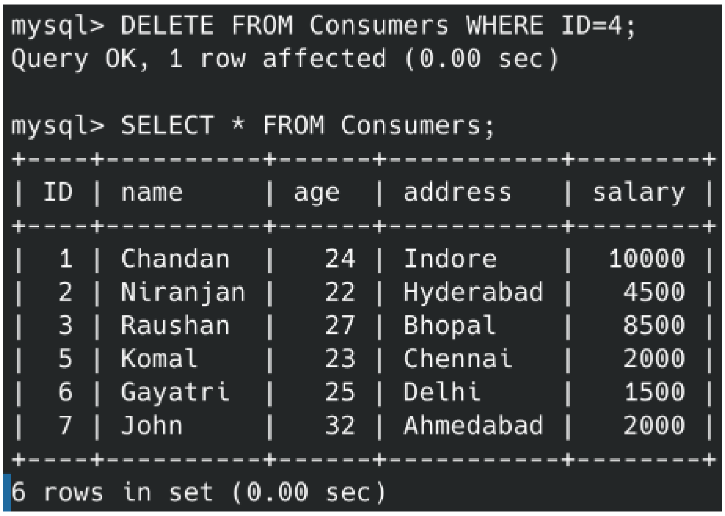


```
mysql> SELECT name FROM Consumers WHERE age=22 OR age=25;
+----------+
| name     |
+----------+
| Niranjan |
| Vikas    |
| Gayatri  |
+----------+
3 rows in set (0.00 sec)
```

I.

DELETE FROM Consumers WHERE ID=4;

SELECT * FROM Consumers;

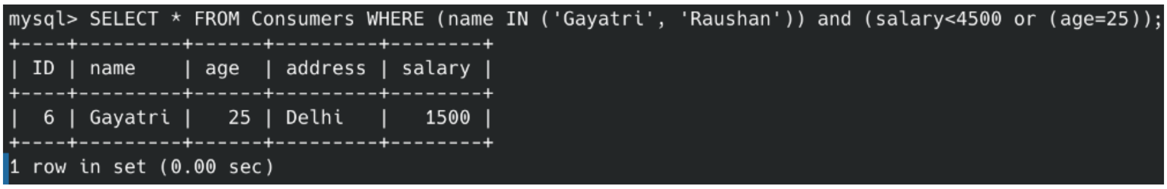


```
mysql> DELETE FROM Consumers WHERE ID=4;
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Consumers;
+----+----------+------+-----------+--------+
| ID | name     | age  | address   | salary |
+----+----------+------+-----------+--------+
|  1 | Chandan  |   24 | Indore    |  10000 |
|  2 | Niranjan |   22 | Hyderabad |   4500 |
|  3 | Raushan  |   27 | Bhopal    |   8500 |
|  5 | Komal    |   23 | Chennai   |   2000 |
|  6 | Gayatri  |   25 | Delhi     |   1500 |
|  7 | John     |   32 | Ahmedabad |   2000 |
+----+----------+------+-----------+--------+
6 rows in set (0.00 sec)
```

J.

SELECT * FROM Consumers WHERE (name IN ('Gayatri', 'Raushan')) and (salary<4500 or (age=25));



```
mysql> SELECT * FROM Consumers WHERE (name IN ('Gayatri', 'Raushan')) and (salary<4500 or (age=25));
+----+---------+------+---------+--------+
| ID | name    | age  | address | salary |
+----+---------+------+---------+--------+
|  6 | Gayatri |   25 | Delhi   |   1500 |
+----+---------+------+---------+--------+
1 row in set (0.00 sec)
```

# Assignment 3

**Creating the table:**

CREATE TABLE CUSTOMERS(ID int PRIMARY KEY, name varchar(30), age int, address varchar(50), salary int);

INSERT INTO CUSTOMERS VALUES(1, 'Chandan', 24, 'Indore', 10000);

INSERT INTO CUSTOMERS VALUES(2, 'Niranjan', 22, 'Hyderabad', 4500);

INSERT INTO CUSTOMERS VALUES(3, 'Raushan', 27, 'Bhopal', 8500);

INSERT INTO CUSTOMERS VALUES(4, 'Vikas', 25, 'Mumbai', 6500);

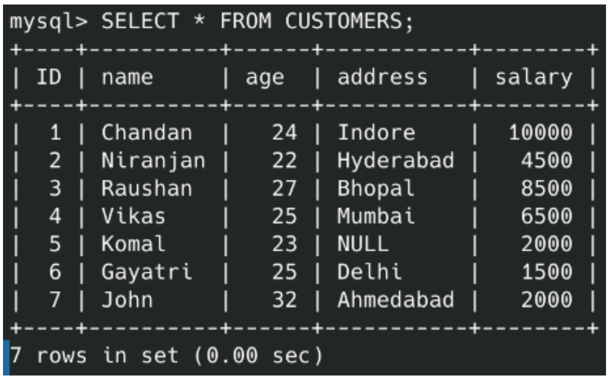
INSERT INTO CUSTOMERS VALUES(5, 'Komal', 23, NULL, 2000);

INSERT INTO CUSTOMERS VALUES(6, 'Gayatri', 25, 'Delhi', 1500);

INSERT INTO CUSTOMERS VALUES(7, 'John', 32, 'Ahmedabad', 2000);

A.

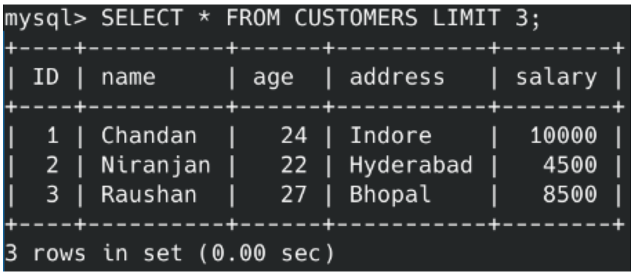
SELECT * FROM CUSTOMERS;



```
mysql> SELECT * FROM CUSTOMERS;
+----+----------+------+-----------+--------+
| ID | name     | age  | address   | salary |
+----+----------+------+-----------+--------+
|  1 | Chandan  |   24 | Indore    |  10000 |
|  2 | Niranjan |   22 | Hyderabad |   4500 |
|  3 | Raushan  |   27 | Bhopal    |   8500 |
|  4 | Vikas    |   25 | Mumbai    |   6500 |
|  5 | Komal    |   23 | NULL      |   2000 |
|  6 | Gayatri  |   25 | Delhi     |   1500 |
|  7 | John     |   32 | Ahmedabad |   2000 |
+----+----------+------+-----------+--------+
7 rows in set (0.00 sec)
```

B.

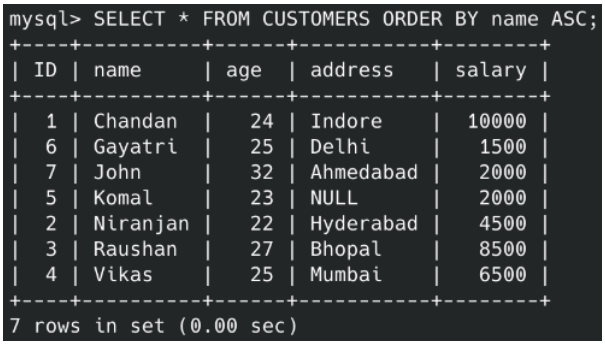
SELECT * FROM CUSTOMERS LIMIT 3;



```
mysql> SELECT * FROM CUSTOMERS LIMIT 3;
+----+----------+------+-----------+--------+
| ID | name     | age  | address   | salary |
+----+----------+------+-----------+--------+
|  1 | Chandan  |   24 | Indore    |  10000 |
|  2 | Niranjan |   22 | Hyderabad |   4500 |
|  3 | Raushan  |   27 | Bhopal    |   8500 |
+----+----------+------+-----------+--------+
3 rows in set (0.00 sec)
```

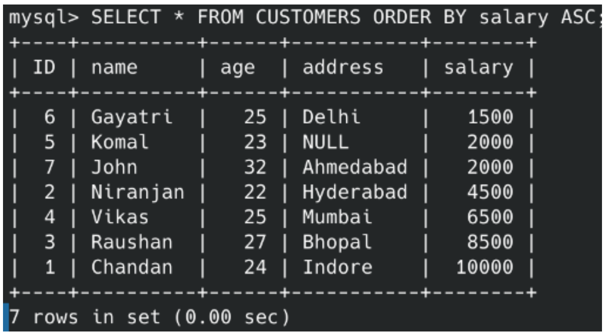
C.

SELECT * FROM CUSTOMERS ORDER BY name ASC;



```
mysql> SELECT * FROM CUSTOMERS ORDER BY name ASC;
+----+----------+------+-----------+--------+
| ID | name     | age  | address   | salary |
+----+----------+------+-----------+--------+
|  1 | Chandan  |   24 | Indore    |  10000 |
|  6 | Gayatri  |   25 | Delhi     |   1500 |
|  7 | John     |   32 | Ahmedabad |   2000 |
|  5 | Komal    |   23 | NULL      |   2000 |
|  2 | Niranjan |   22 | Hyderabad |   4500 |
|  3 | Raushan  |   27 | Bhopal    |   8500 |
|  4 | Vikas    |   25 | Mumbai    |   6500 |
+----+----------+------+-----------+--------+
7 rows in set (0.00 sec)
```

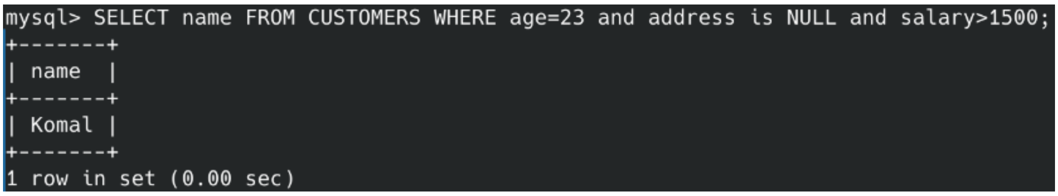
SELECT * FROM CUSTOMERS ORDER BY salary ASC;



```
mysql> SELECT * FROM CUSTOMERS ORDER BY salary ASC;
+----+----------+------+-----------+--------+
| ID | name     | age  | address   | salary |
+----+----------+------+-----------+--------+
|  6 | Gayatri  |   25 | Delhi     |   1500 |
|  5 | Komal    |   23 | NULL      |   2000 |
|  7 | John     |   32 | Ahmedabad |   2000 |
|  2 | Niranjan |   22 | Hyderabad |   4500 |
|  4 | Vikas    |   25 | Mumbai    |   6500 |
|  3 | Raushan  |   27 | Bhopal    |   8500 |
|  1 | Chandan  |   24 | Indore    |  10000 |
+----+----------+------+-----------+--------+
7 rows in set (0.00 sec)
```

D.

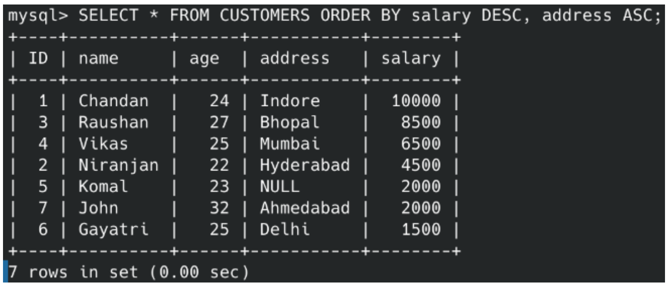
SELECT name FROM CUSTOMERS WHERE age=23 and address is NULL and salary>1500;



```
mysql> SELECT name FROM CUSTOMERS WHERE age=23 and address is NULL and salary>1500;
+-------+
| name  |
+-------+
| Komal |
+-------+
1 row in set (0.00 sec)
```

E.

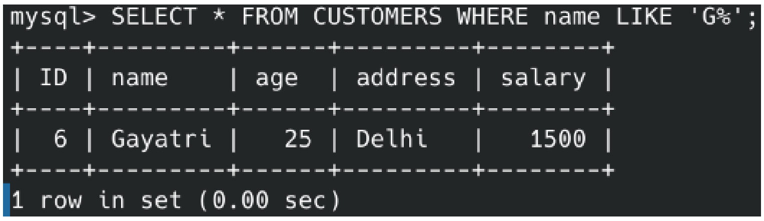
SELECT * FROM CUSTOMERS ORDER BY salary DESC, address ASC;



```
mysql> SELECT * FROM CUSTOMERS ORDER BY salary DESC, address ASC;
+----+----------+------+-----------+--------+
| ID | name     | age  | address   | salary |
+----+----------+------+-----------+--------+
|  1 | Chandan  |   24 | Indore    |  10000 |
|  3 | Raushan  |   27 | Bhopal    |   8500 |
|  4 | Vikas    |   25 | Mumbai    |   6500 |
|  2 | Niranjan |   22 | Hyderabad |   4500 |
|  5 | Komal    |   23 | NULL      |   2000 |
|  7 | John     |   32 | Ahmedabad |   2000 |
|  6 | Gayatri  |   25 | Delhi     |   1500 |
+----+----------+------+-----------+--------+
7 rows in set (0.00 sec)
```

F.

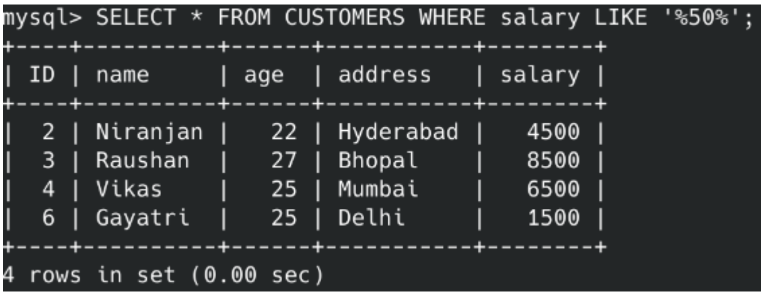
SELECT * FROM CUSTOMERS WHERE name LIKE 'G%';



```
mysql> SELECT * FROM CUSTOMERS WHERE name LIKE 'G%';
+----+---------+------+---------+--------+
| ID | name    | age  | address | salary |
+----+---------+------+---------+--------+
|  6 | Gayatri |   25 | Delhi   |   1500 |
+----+---------+------+---------+--------+
1 row in set (0.00 sec)
```

G.

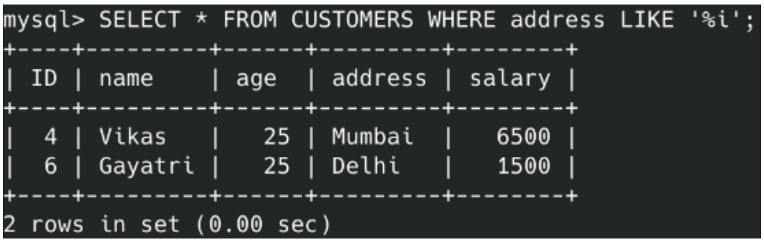
SELECT * FROM CUSTOMERS WHERE salary LIKE '%50%';



```
mysql> SELECT * FROM CUSTOMERS WHERE salary LIKE '%50%';
+----+----------+------+-----------+--------+
| ID | name     | age  | address   | salary |
+----+----------+------+-----------+--------+
|  2 | Niranjan |   22 | Hyderabad |   4500 |
|  3 | Raushan  |   27 | Bhopal    |   8500 |
|  4 | Vikas    |   25 | Mumbai    |   6500 |
|  6 | Gayatri  |   25 | Delhi     |   1500 |
+----+----------+------+-----------+--------+
4 rows in set (0.00 sec)
```

H.

SELECT * FROM CUSTOMERS WHERE address LIKE '%i';



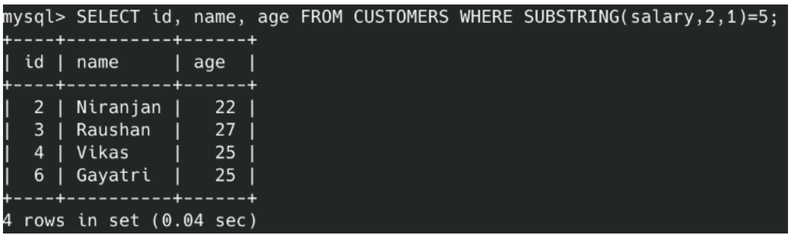
```
mysql> SELECT * FROM CUSTOMERS WHERE address LIKE '%i';
+----+---------+------+---------+--------+
| ID | name    | age  | address | salary |
+----+---------+------+---------+--------+
|  4 | Vikas   |   25 | Mumbai  |   6500 |
|  6 | Gayatri |   25 | Delhi   |   1500 |
+----+---------+------+---------+--------+
2 rows in set (0.00 sec)
```

I.

SELECT id, name, age FROM CUSTOMERS WHERE salary LIKE "_5%";

OR

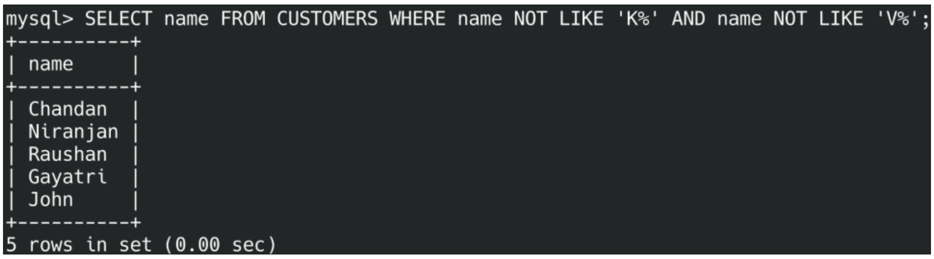
SELECT id, name, age FROM CUSTOMERS WHERE SUBSTRING(salary,2,1)=5;



```
mysql> SELECT id, name, age FROM CUSTOMERS WHERE SUBSTRING(salary,2,1)=5;
+----+----------+------+
| id | name     | age  |
+----+----------+------+
|  2 | Niranjan |   22 |
|  3 | Raushan  |   27 |
|  4 | Vikas    |   25 |
|  6 | Gayatri  |   25 |
+----+----------+------+
4 rows in set (0.04 sec)
```

J.

SELECT name FROM CUSTOMERS WHERE name NOT LIKE 'K%' AND name NOT LIKE 'V%';



```
mysql> SELECT name FROM CUSTOMERS WHERE name NOT LIKE 'K%' AND name NOT LIKE 'V%';
+----------+
| name     |
+----------+
| Chandan  |
| Niranjan |
| Raushan  |
| Gayatri  |
| John     |
+----------+
5 rows in set (0.00 sec)
```

# Assignment 4

**Creating the table**:

CREATE TABLE CUSTOMERS(ID int PRIMARY KEY, name varchar(30), age int, address varchar(50), salary int);

INSERT INTO CUSTOMERS VALUES(1, 'Chandan', 32, 'Hyderabad', 10000);

INSERT INTO CUSTOMERS VALUES(2, 'Niranjan', 32, 'Hyderabad', 4500);

INSERT INTO CUSTOMERS VALUES(3, 'Raushan', 23, 'Delhi', 8500);

INSERT INTO CUSTOMERS VALUES(4, 'Vikas', 25, 'Delhi', 6500);

INSERT INTO CUSTOMERS VALUES(5, 'Komal', 23, 'Bhopal', 2000);

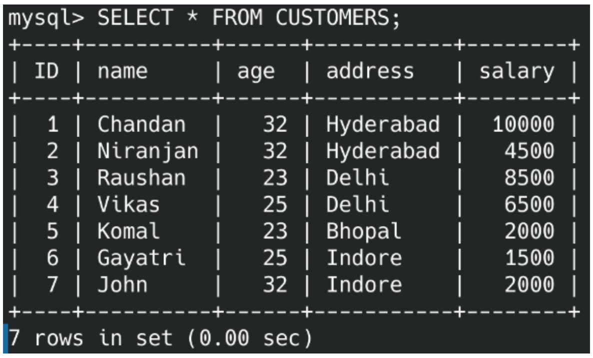
INSERT INTO CUSTOMERS VALUES(6, 'Gayatri', 25, 'Indore', 1500);

INSERT INTO CUSTOMERS VALUES(7, 'John', 32, 'Indore', 2000);

**Queries**:

A.

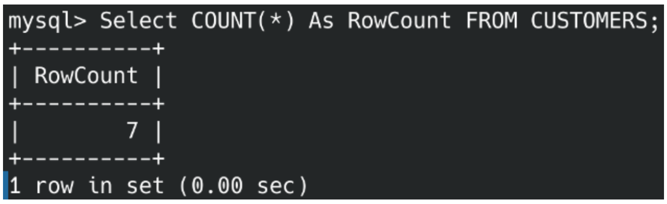
SELECT * FROM CUSTOMERS;



```
mysql> SELECT * FROM CUSTOMERS;
+----+----------+------+-----------+--------+
| ID | name     | age  | address   | salary |
+----+----------+------+-----------+--------+
|  1 | Chandan  |   32 | Hyderabad |  10000 |
|  2 | Niranjan |   32 | Hyderabad |   4500 |
|  3 | Raushan  |   23 | Delhi     |   8500 |
|  4 | Vikas    |   25 | Delhi     |   6500 |
|  5 | Komal    |   23 | Bhopal    |   2000 |
|  6 | Gayatri  |   25 | Indore    |   1500 |
|  7 | John     |   32 | Indore    |   2000 |
+----+----------+------+-----------+--------+
7 rows in set (0.00 sec)
```

B.

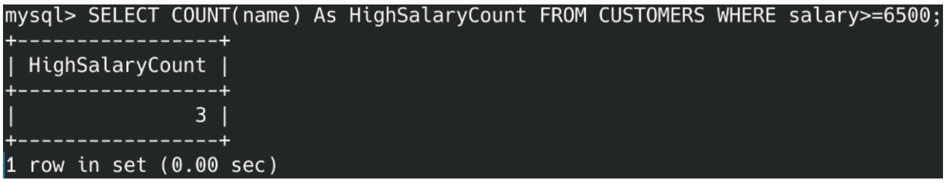
Select COUNT(*) As RowCount FROM CUSTOMERS;



```
mysql> Select COUNT(*) As RowCount FROM CUSTOMERS;
+----------+
| RowCount |
+----------+
|        7 |
+----------+
1 row in set (0.00 sec)
```

C.

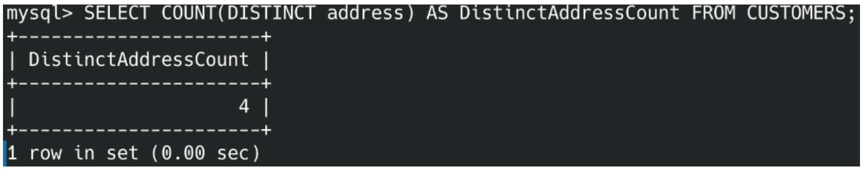
SELECT COUNT(name) As HighSalaryCount FROM CUSTOMERS WHERE salary>=6500;



```
mysql> SELECT COUNT(name) As HighSalaryCount FROM CUSTOMERS WHERE salary>=6500;
+-----------------+
| HighSalaryCount |
+-----------------+
|               3 |
+-----------------+
1 row in set (0.00 sec)
```

D.

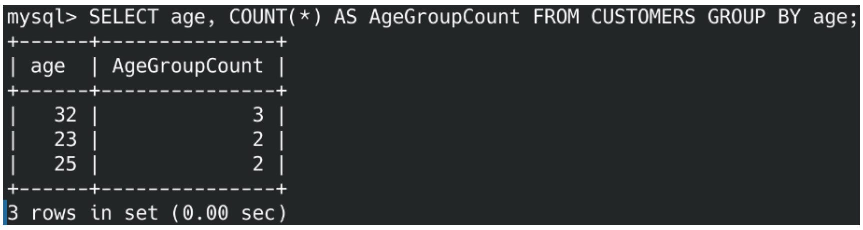
SELECT COUNT(DISTINCT address) AS DistinctAddressCount FROM CUSTOMERS;



```
mysql> SELECT COUNT(DISTINCT address) AS DistinctAddressCount FROM CUSTOMERS;
+----------------------+
| DistinctAddressCount |
+----------------------+
|                    4 |
+----------------------+
1 row in set (0.00 sec)
```

E.

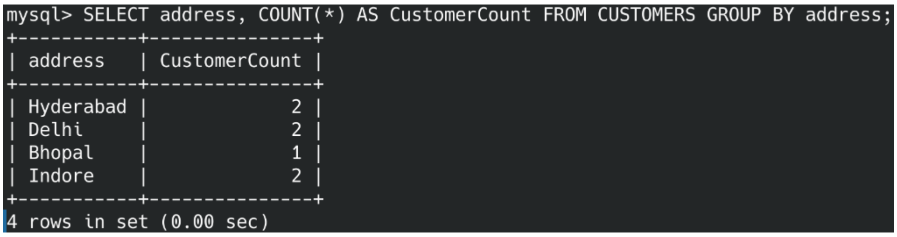
SELECT age, COUNT(*) AS AgeGroupCount FROM CUSTOMERS GROUP BY age;



```
mysql> SELECT age, COUNT(*) AS AgeGroupCount FROM CUSTOMERS GROUP BY age;
+------+---------------+
| age  | AgeGroupCount |
+------+---------------+
|   32 |             3 |
|   23 |             2 |
|   25 |             2 |
+------+---------------+
3 rows in set (0.00 sec)
```

F.

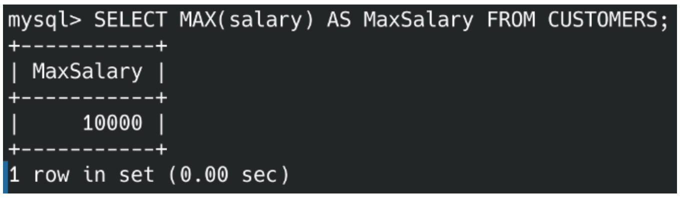
SELECT address, COUNT(*) AS CustomerCount FROM CUSTOMERS GROUP BY address;



```
mysql> SELECT address, COUNT(*) AS CustomerCount FROM CUSTOMERS GROUP BY address;
+-----------+---------------+
| address   | CustomerCount |
+-----------+---------------+
| Hyderabad |             2 |
| Delhi     |             2 |
| Bhopal    |             1 |
| Indore    |             2 |
+-----------+---------------+
4 rows in set (0.00 sec)
```

G.

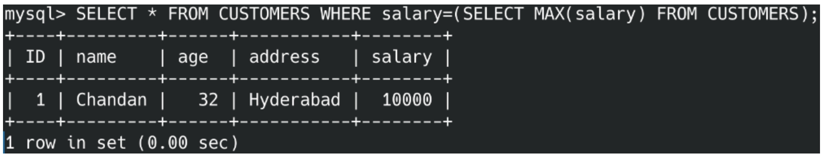
SELECT MAX(salary) AS MaxSalary FROM CUSTOMERS;



```
mysql> SELECT MAX(salary) AS MaxSalary FROM CUSTOMERS;
+-----------+
| MaxSalary |
+-----------+
|     10000 |
+-----------+
1 row in set (0.00 sec)
```

H.

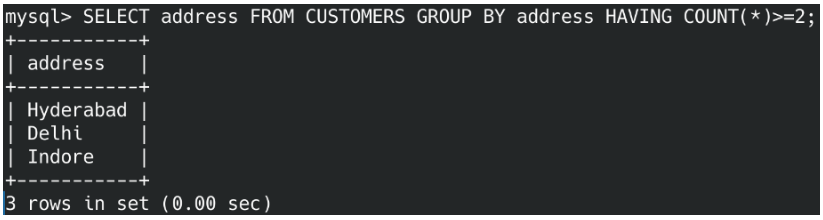
SELECT * FROM CUSTOMERS WHERE salary=(SELECT MAX(salary) FROM CUSTOMERS);



```
mysql> SELECT * FROM CUSTOMERS WHERE salary=(SELECT MAX(salary) FROM CUSTOMERS);
+----+---------+------+-----------+--------+
| ID | name    | age  | address   | salary |
+----+---------+------+-----------+--------+
|  1 | Chandan |   32 | Hyderabad |  10000 |
+----+---------+------+-----------+--------+
1 row in set (0.00 sec)
```

I.

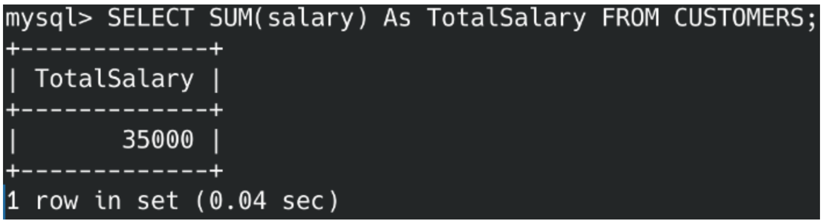
SELECT address FROM CUSTOMERS GROUP BY address HAVING COUNT(*)>=2;



```
mysql> SELECT address FROM CUSTOMERS GROUP BY address HAVING COUNT(*)>=2;
+-----------+
| address   |
+-----------+
| Hyderabad |
| Delhi     |
| Indore    |
+-----------+
3 rows in set (0.00 sec)
```

J.

SELECT SUM(salary) As TotalSalary FROM CUSTOMERS;



```
mysql> SELECT SUM(salary) As TotalSalary FROM CUSTOMERS;
+-------------+
| TotalSalary |
+-------------+
|       35000 |
+-------------+
1 row in set (0.04 sec)
```

# Assignment 5

**<u>Creating the table</u>**:

CREATE TABLE CUSTOMERS(ID int PRIMARY KEY, name varchar(30), age int, address varchar(50), salary int);

INSERT INTO CUSTOMERS VALUES(1, 'Chandan', 32, 'Hyderabad', 10000);

INSERT INTO CUSTOMERS VALUES(2, 'Niranjan', 32, 'Hyderabad', 4500);

INSERT INTO CUSTOMERS VALUES(3, 'Raushan', 23, 'Delhi', 8500);

INSERT INTO CUSTOMERS VALUES(4, 'Vikas', 25, 'Delhi', 6500);

INSERT INTO CUSTOMERS VALUES(5, 'Komal', 23, 'Bhopal', 2000);

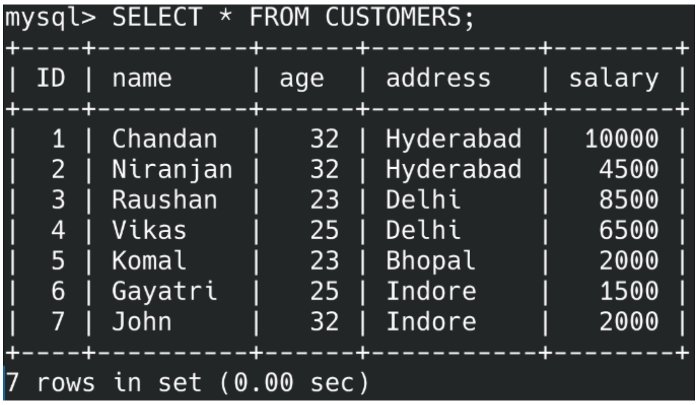
INSERT INTO CUSTOMERS VALUES(6, 'Gayatri', 25, 'Indore', 1500);

INSERT INTO CUSTOMERS VALUES(7, 'John', 32, 'Indore', 2000);

**<u>Queries</u>**:

A.

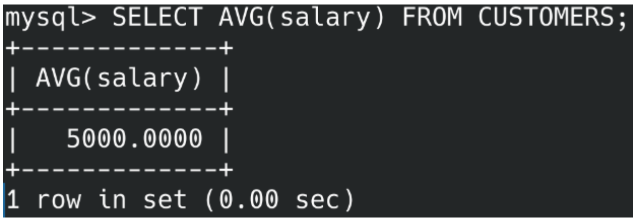
SELECT * FROM CUSTOMERS;



```
mysql> SELECT * FROM CUSTOMERS;
+----+----------+------+-----------+--------+
| ID | name     | age  | address   | salary |
+----+----------+------+-----------+--------+
|  1 | Chandan  |   32 | Hyderabad |  10000 |
|  2 | Niranjan |   32 | Hyderabad |   4500 |
|  3 | Raushan  |   23 | Delhi     |   8500 |
|  4 | Vikas    |   25 | Delhi     |   6500 |
|  5 | Komal    |   23 | Bhopal    |   2000 |
|  6 | Gayatri  |   25 | Indore    |   1500 |
|  7 | John     |   32 | Indore    |   2000 |
+----+----------+------+-----------+--------+
7 rows in set (0.00 sec)
```

B.

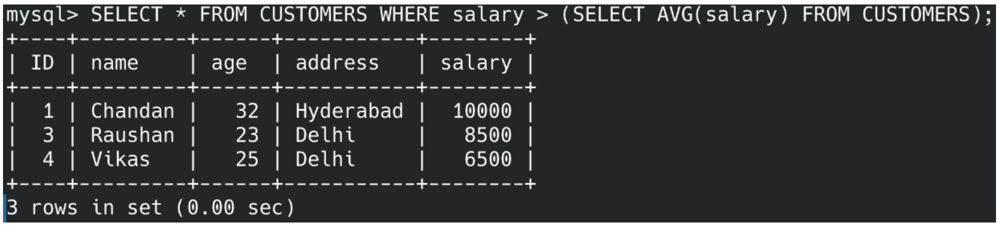
SELECT AVG(salary) FROM CUSTOMERS;



```
mysql> SELECT AVG(salary) FROM CUSTOMERS;
+-------------+
| AVG(salary) |
+-------------+
|   5000.0000 |
+-------------+
1 row in set (0.00 sec)
```

C.

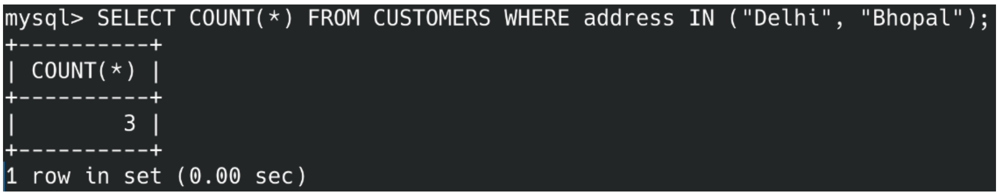
SELECT * FROM CUSTOMERS WHERE salary > (SELECT AVG(salary) FROM CUSTOMERS);



```
mysql> SELECT * FROM CUSTOMERS WHERE salary > (SELECT AVG(salary) FROM CUSTOMERS);
+----+---------+------+-----------+--------+
| ID | name    | age  | address   | salary |
+----+---------+------+-----------+--------+
|  1 | Chandan |   32 | Hyderabad |  10000 |
|  3 | Raushan |   23 | Delhi     |   8500 |
|  4 | Vikas   |   25 | Delhi     |   6500 |
+----+---------+------+-----------+--------+
3 rows in set (0.00 sec)
```

D.

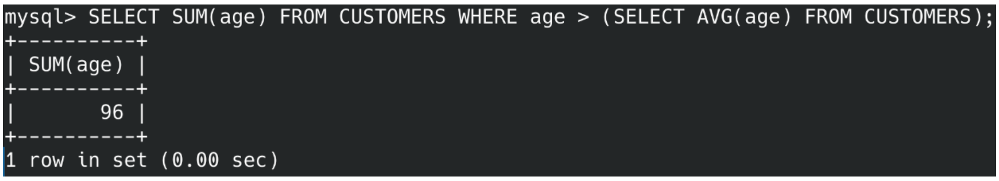
SELECT COUNT(*) FROM CUSTOMERS WHERE address IN ("Delhi", "Bhopal");



```
mysql> SELECT COUNT(*) FROM CUSTOMERS WHERE address IN ("Delhi", "Bhopal");
+----------+
| COUNT(*) |
+----------+
|        3 |
+----------+
1 row in set (0.00 sec)
```

E.

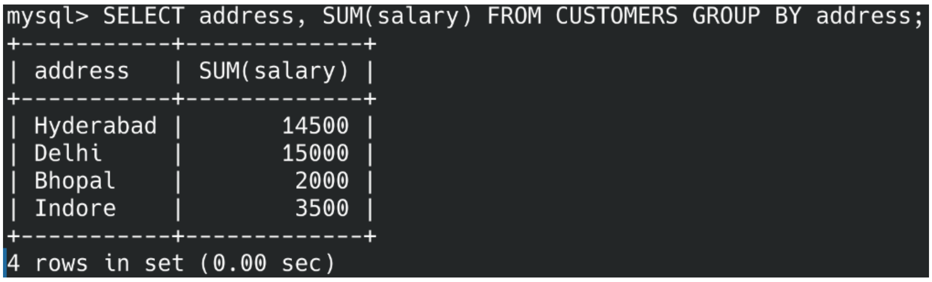
SELECT SUM(age) FROM CUSTOMERS WHERE age > (SELECT AVG(age) FROM CUSTOMERS);



```
mysql> SELECT SUM(age) FROM CUSTOMERS WHERE age > (SELECT AVG(age) FROM CUSTOMERS);
+----------+
| SUM(age) |
+----------+
|       96 |
+----------+
1 row in set (0.00 sec)
```

F.

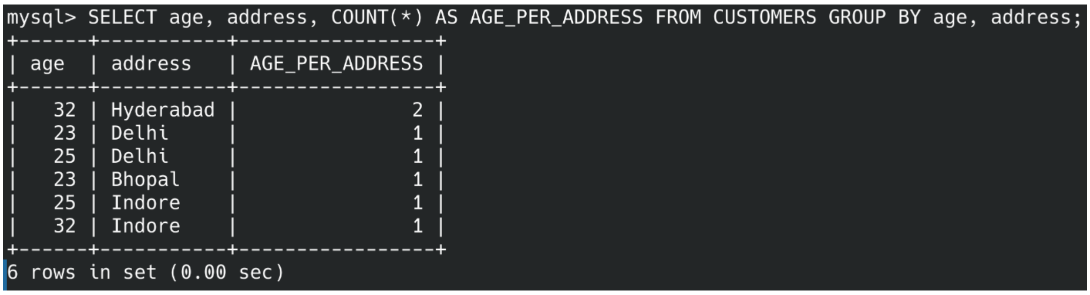
SELECT address, SUM(salary) FROM CUSTOMERS GROUP BY address;



```
mysql> SELECT address, SUM(salary) FROM CUSTOMERS GROUP BY address;
+-----------+-------------+
| address   | SUM(salary) |
+-----------+-------------+
| Hyderabad |       14500 |
| Delhi     |       15000 |
| Bhopal    |        2000 |
| Indore    |        3500 |
+-----------+-------------+
4 rows in set (0.00 sec)
```

G.

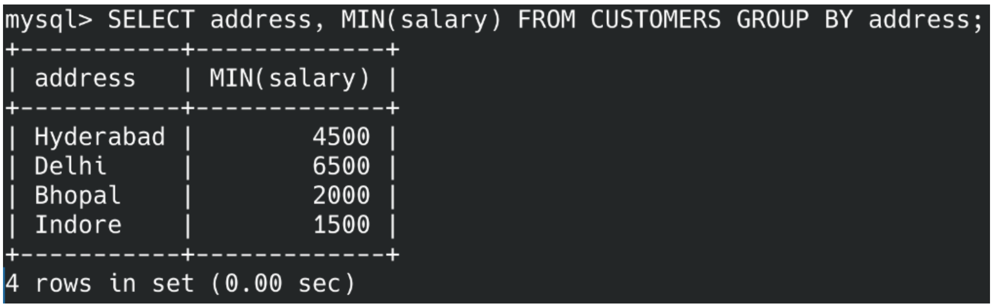
SELECT age, address, COUNT(*) AS AGE_PER_ADDRESS FROM CUSTOMERS GROUP BY age, address;



```
mysql> SELECT age, address, COUNT(*) AS AGE_PER_ADDRESS FROM CUSTOMERS GROUP BY age, address;
+------+-----------+-----------------+
| age  | address   | AGE_PER_ADDRESS |
+------+-----------+-----------------+
|   32 | Hyderabad |               2 |
|   23 | Delhi     |               1 |
|   25 | Delhi     |               1 |
|   23 | Bhopal    |               1 |
|   25 | Indore    |               1 |
|   32 | Indore    |               1 |
+------+-----------+-----------------+
6 rows in set (0.00 sec)
```

H.

SELECT address, MIN(salary) FROM CUSTOMERS GROUP BY address;



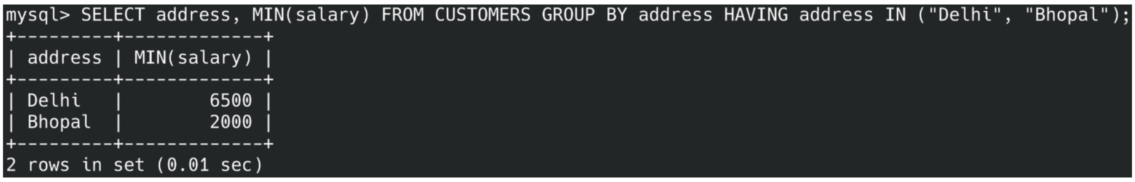
```
mysql> SELECT address, MIN(salary) FROM CUSTOMERS GROUP BY address;
+-----------+-------------+
| address   | MIN(salary) |
+-----------+-------------+
| Hyderabad |        4500 |
| Delhi     |        6500 |
| Bhopal    |        2000 |
| Indore    |        1500 |
+-----------+-------------+
4 rows in set (0.00 sec)
```

I.

SELECT address, MIN(salary) FROM CUSTOMERS GROUP BY address HAVING address IN ("Delhi", "Bhopal");

OR

SELECT address, MIN(salary) FROM CUSTOMERS WHERE address IN ("Delhi", "Bhopal") GROUP BY address;

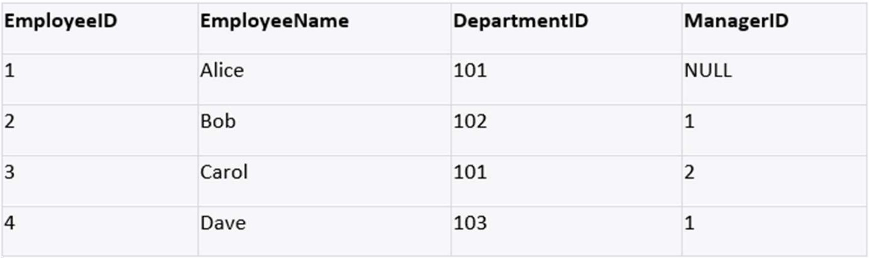


```
mysql> SELECT address, MIN(salary) FROM CUSTOMERS GROUP BY address HAVING address IN ("Delhi", "Bhopal");
+---------+-------------+
| address | MIN(salary) |
+---------+-------------+
| Delhi   |        6500 |
| Bhopal  |        2000 |
+---------+-------------+
2 rows in set (0.01 sec)
```

# Assignment 6

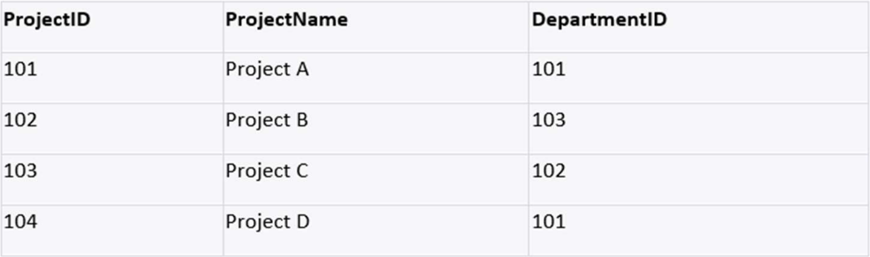
1. Create two tables with the given fields and data in the tables below. The name of the tables should be EMPLOYEE and PROJECT.

**Employees Table:**

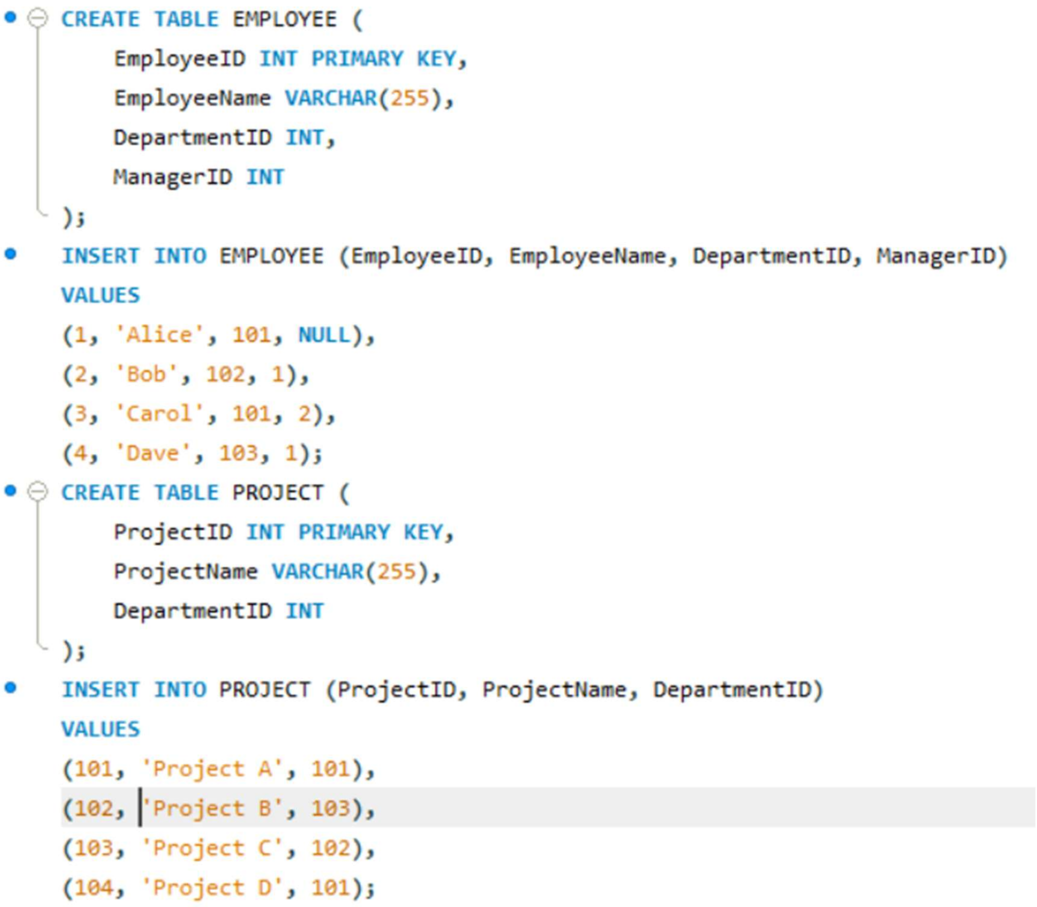


| EmployeeID | EmployeeName | DepartmentID | ManagerID |
|---|---|---|---|
| 1 | Alice | 101 | NULL |
| 2 | Bob | 102 | 1 |
| 3 | Carol | 101 | 2 |
| 4 | Dave | 103 | 1 |

**Projects Table:**

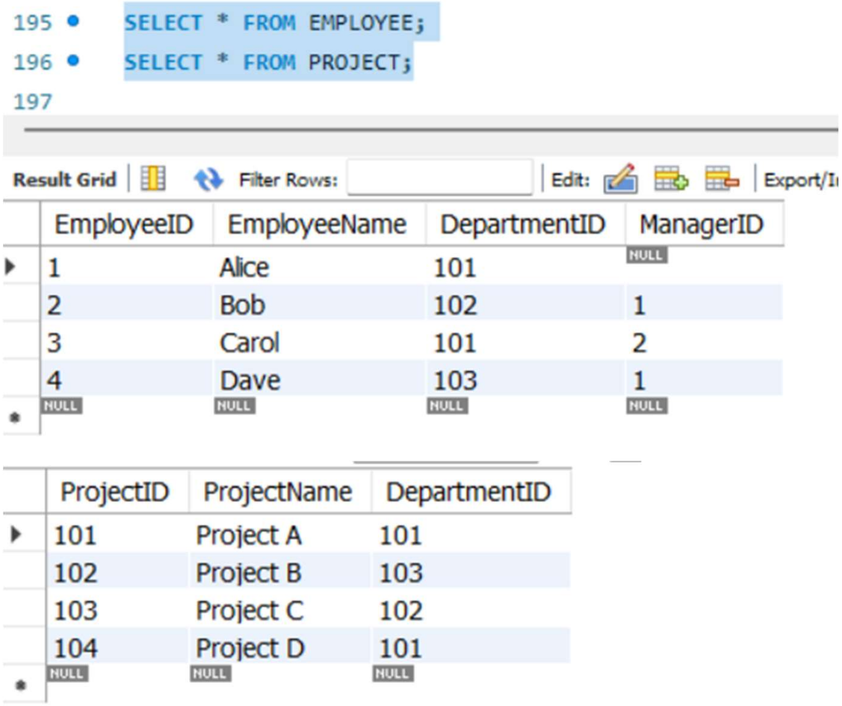


| ProjectID | ProjectName | DepartmentID |
|---|---|---|
| 101 | Project A | 101 |
| 102 | Project B | 103 |
| 103 | Project C | 102 |
| 104 | Project D | 101 |



```
CREATE TABLE EMPLOYEE (
    EmployeeID INT PRIMARY KEY,
    EmployeeName VARCHAR(255),
    DepartmentID INT,
    ManagerID INT
);
INSERT INTO EMPLOYEE (EmployeeID, EmployeeName, DepartmentID, ManagerID)
VALUES
(1, 'Alice', 101, NULL),
(2, 'Bob', 102, 1),
(3, 'Carol', 101, 2),
(4, 'Dave', 103, 1);
CREATE TABLE PROJECT (
    ProjectID INT PRIMARY KEY,
    ProjectName VARCHAR(255),
    DepartmentID INT
);
INSERT INTO PROJECT (ProjectID, ProjectName, DepartmentID)
VALUES
(101, 'Project A', 101),
(102, 'Project B', 103),
(103, 'Project C', 102),
(104, 'Project D', 101);
```

2. Write an SQL query to print each and every record from both of the table in table format.

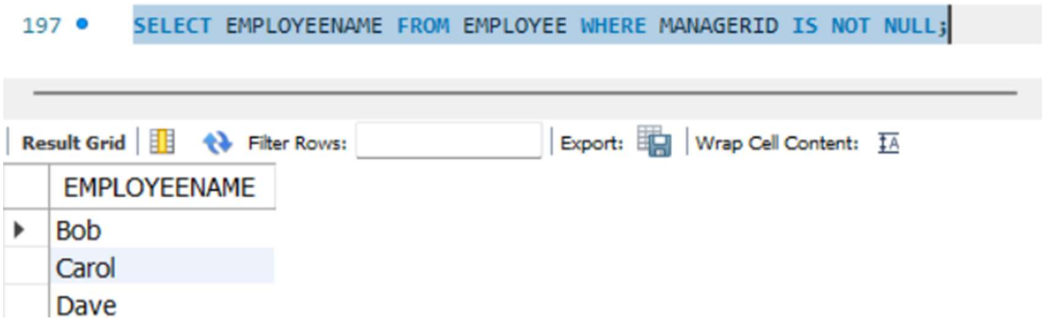


```
SELECT * FROM EMPLOYEE;
SELECT * FROM PROJECT;
```

| EmployeeID | EmployeeName | DepartmentID | ManagerID |
|---|---|---|---|
| 1 | Alice | 101 | NULL |
| 2 | Bob | 102 | 1 |
| 3 | Carol | 101 | 2 |
| 4 | Dave | 103 | 1 |
| NULL | NULL | NULL | NULL |

| ProjectID | ProjectName | DepartmentID |
|---|---|---|
| 101 | Project A | 101 |
| 102 | Project B | 103 |
| 103 | Project C | 102 |
| 104 | Project D | 101 |
| NULL | NULL | NULL |

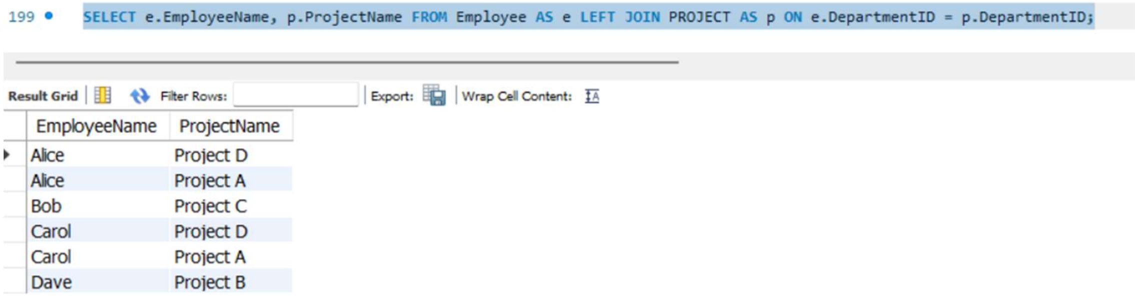
3. Write an SQL query to find the names of employees who are managers.



```
SELECT EMPLOYEENAME FROM EMPLOYEE WHERE MANAGERID IS NOT NULL;
```

| EMPLOYEENAME |
|---|
| Bob |
| Carol |
| Dave |

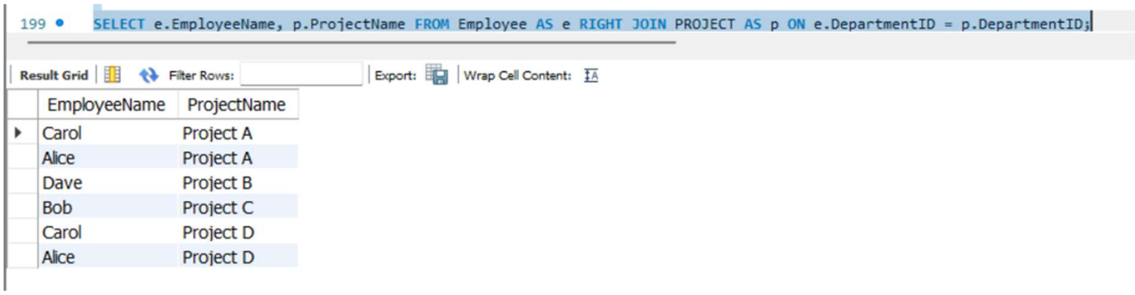
4. Using LEFT JOIN, retrieve the names of all employees and the projects they are assigned to. Additionally, include employees who do not have any project assignments.



```
SELECT e.EmployeeName, p.ProjectName FROM Employee AS e LEFT JOIN PROJECT AS p ON e.DepartmentID = p.DepartmentID;
```

| EmployeeName | ProjectName |
|---|---|
| Alice | Project D |
| Alice | Project A |
| Bob | Project C |
| Carol | Project D |
| Carol | Project A |
| Dave | Project B |

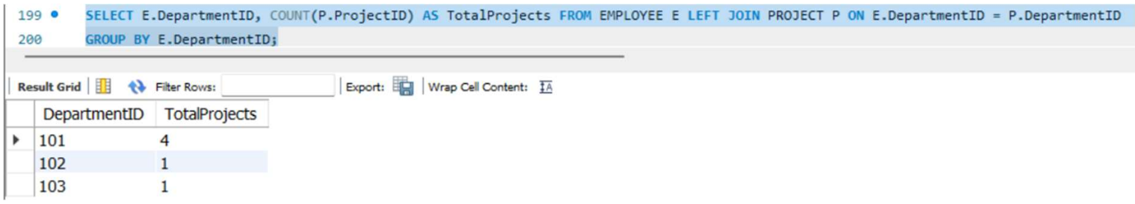
5. Perform a RIGHT JOIN to find the names of employees and the projects they are working on (including projects without assigned employees).



```
SELECT e.EmployeeName, p.ProjectName FROM Employee AS e RIGHT JOIN PROJECT AS p ON e.DepartmentID = p.DepartmentID;
```

| EmployeeName | ProjectName |
|---|---|
| Carol | Project A |
| Alice | Project A |
| Dave | Project B |
| Bob | Project C |
| Carol | Project D |
| Alice | Project D |

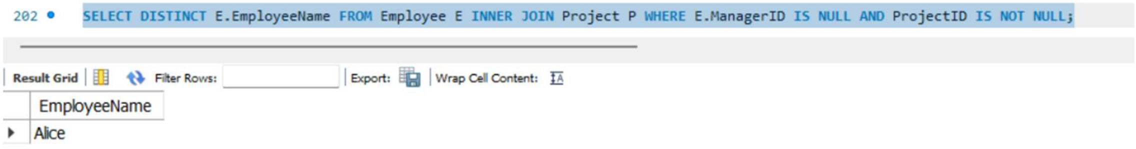
6. Write a query to count the total number of projects in each department. Include departments with no projects in the result using a LEFT JOIN.



```
SELECT E.DepartmentID, COUNT(P.ProjectID) AS TotalProjects FROM EMPLOYEE E LEFT JOIN PROJECT P ON E.DepartmentID = P.DepartmentID
GROUP BY E.DepartmentID;
```

| DepartmentID | TotalProjects |
|---|---|
| 101 | 4 |
| 102 | 1 |
| 103 | 1 |

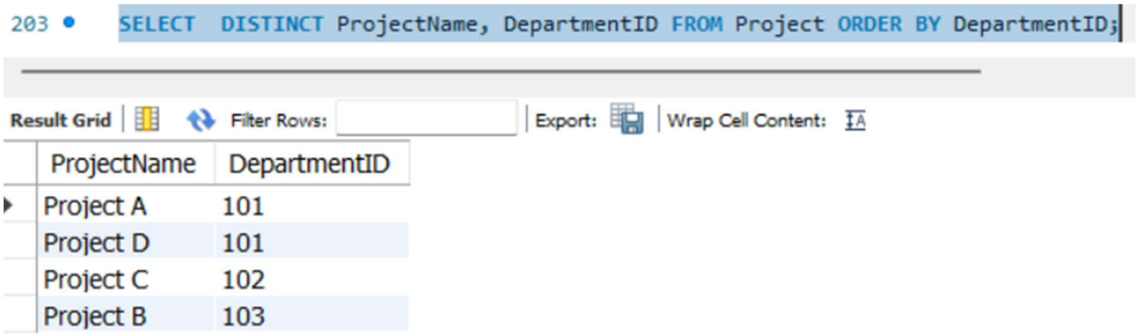
7. Use INNER JOIN to retrieve the names of employees who are not managers but are working on projects. Exclude employees who are managers.



```sql
SELECT DISTINCT E.EmployeeName FROM Employee E INNER JOIN Project P WHERE E.ManagerID IS NULL AND ProjectID IS NOT NULL;
```

| EmployeeName |
|---|
| Alice |

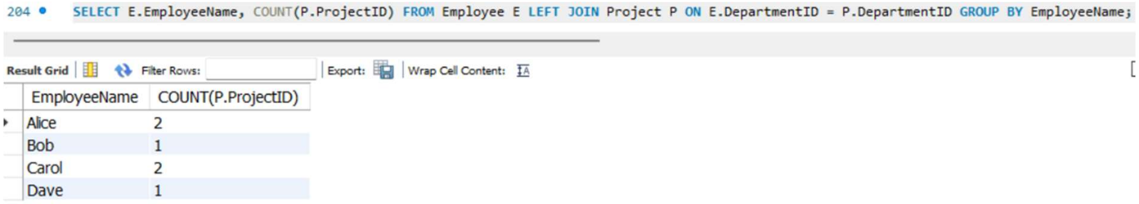
8. Write an SQL query that lists all projects and their respective department names, sorted by department name. Ensure that projects without a department are also included in the list.



```sql
SELECT  DISTINCT ProjectName, DepartmentID FROM Project ORDER BY DepartmentID;
```

| ProjectName | DepartmentID |
|---|---|
| Project A | 101 |
| Project D | 101 |
| Project C | 102 |
| Project B | 103 |

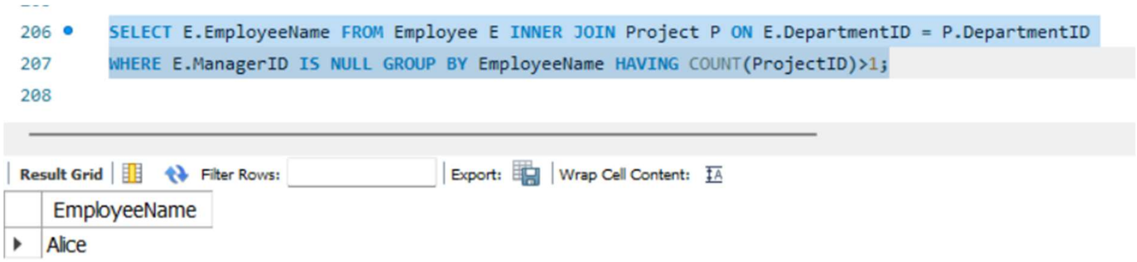
9. Perform a LEFT JOIN to find the names of employees and the total number of projects each employee is working on. Include employees who have no project assignments.



```sql
SELECT E.EmployeeName, COUNT(P.ProjectID) FROM Employee E LEFT JOIN Project P ON E.DepartmentID = P.DepartmentID GROUP BY EmployeeName;
```

| EmployeeName | COUNT(P.ProjectID) |
|---|---|
| Alice | 2 |
| Bob | 1 |
| Carol | 2 |
| Dave | 1 |

10. Write an SQL query to find the names of employees who are not managers but are working on projects from multiple departments.



```
SELECT E.EmployeeName FROM Employee E INNER JOIN Project P ON E.DepartmentID = P.DepartmentID
WHERE E.ManagerID IS NULL GROUP BY EmployeeName HAVING COUNT(ProjectID)>1;
```

```
EmployeeName
Alice
```