1. Create an employee database with 5 columns
   (emp_id, emp_name, emp_age, emp_salary, job_role) and 10 records. Choose a unique column as a primary key.
   Note: Only two job roles available ("Data Analyst","ML Engineer").

```
MariaDB [mysql]> create table employee(emp_id int,emp_name varchar(20),emp_age int,emp_salary int,job_role varchar(20),primary key(emp_id));
Query OK, 0 rows affected (0.349 sec)

MariaDB [mysql]> insert into employee values(100,"john",22,20000,"data analyst"),(101,"ben",23,30000,"ml engineer")
    -> ,(102,"lee",22,25000,"ml engineer")
    -> ,(103,"bob",24,15000,"data analyst")
    -> ,(104,"rock",23,25000,"ml engineer")
    -> ,(105,"ram",23,25000,"data analyst")
    -> ,(106,"mary",24,15000,"data analyst")
    -> ,(107,"stella",23,20000,"ml engineer")
    -> ,(108,"rohit",25,30000,"data analyst")
    -> ,(109,"rohit",26,30000,"data analyst");
Query OK, 10 rows affected (0.081 sec)
Records: 10  Duplicates: 0  Warnings: 0

MariaDB [mysql]> select * from employee;
+--------+----------+---------+------------+-------------+
| emp_id | emp_name | emp_age | emp_salary | job_role    |
+--------+----------+---------+------------+-------------+
|    100 | john     |      22 |      20000 | data analyst |
|    101 | ben      |      23 |      30000 | ml engineer  |
|    102 | lee      |      22 |      25000 | ml engineer  |
|    103 | bob      |      24 |      15000 | data analyst |
|    104 | rock     |      23 |      25000 | ml engineer  |
|    105 | ram      |      23 |      25000 | data analyst |
|    106 | mary     |      24 |      15000 | data analyst |
|    107 | stella   |      23 |      20000 | ml engineer  |
|    108 | rohit    |      25 |      30000 | data analyst |
|    109 | rohit    |      26 |      30000 | data analyst |
+--------+----------+---------+------------+-------------+
10 rows in set (0.000 sec)

MariaDB [mysql]>
```

I have created employee table with given columns using CREATE Command and inserted the values using INSERT Command

2. Retrieve the name of the employee who is having the highest salary.

```
MariaDB [mysql]> select emp_name from employee where emp_salary=(select max(emp_salary) from employee);
+----------+
| emp_name |
+----------+
| ben      |
| rohit    |
| rohit    |
+----------+
3 rows in set (0.078 sec)

MariaDB [mysql]>
```

I used a sub query to get the highest emp_salary and then select their names, here there are 3 employees with same highest emp_salary

3. Retrieve distinct names of the employees whose salary is in between [50000,59999] using where clause.

```
MariaDB [mysql]> update employee set emp_salary=50000 where emp_id=103;
Query OK, 1 row affected (0.346 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [mysql]> update employee set emp_salary=55000 where emp_id=104;
Query OK, 1 row affected (0.105 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [mysql]> update employee set emp_salary=57000 where emp_id=102;
Query OK, 1 row affected (0.331 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [mysql]> select * from employee;
+--------+----------+---------+------------+--------------+
| emp_id | emp_name | emp_age | emp_salary | job_role     |
+--------+----------+---------+------------+--------------+
|    100 | john     |      22 |      20000 | data analyst |
|    101 | ben      |      23 |      30000 | ml engineer  |
|    102 | lee      |      22 |      57000 | ml engineer  |
|    103 | bob      |      24 |      50000 | data analyst |
|    104 | rock     |      23 |      55000 | ml engineer  |
|    105 | ram      |      23 |      25000 | data analyst |
|    106 | mary     |      24 |      15000 | data analyst |
|    107 | stella   |      23 |      20000 | ml engineer  |
|    108 | rohit    |      25 |      30000 | data analyst |
|    109 | rohit    |      26 |      30000 | data analyst |
+--------+----------+---------+------------+--------------+
10 rows in set (0.000 sec)

MariaDB [mysql]> select distinct emp_name from employee where emp_salary between 50000 and 59999;
+----------+
| emp_name |
+----------+
| lee      |
| bob      |
| rock     |
+----------+
3 rows in set (0.033 sec)

MariaDB [mysql]>
```

Initially table didn't have salaries in between 50000 and 59999, so I updated some values for them. Here I have used where clause with between and selected distinct emp_names from the table

4. Modify the Data Analyst job role into Data Scientist.

```
MariaDB [mysql]> update employee set job_role="data scientist" where job_role="data analyst";
Query OK, 6 rows affected (0.178 sec)
Rows matched: 6  Changed: 6  Warnings: 0

MariaDB [mysql]> select * from employee;
+--------+----------+---------+------------+----------------+
| emp_id | emp_name | emp_age | emp_salary | job_role       |
+--------+----------+---------+------------+----------------+
|    100 | john     |      22 |      20000 | data scientist |
|    101 | ben      |      23 |      30000 | ml engineer    |
|    102 | lee      |      22 |      57000 | ml engineer    |
|    103 | bob      |      24 |      50000 | data scientist |
|    104 | rock     |      23 |      55000 | ml engineer    |
|    105 | ram      |      23 |      25000 | data scientist |
|    106 | mary     |      24 |      15000 | data scientist |
|    107 | stella   |      23 |      20000 | ml engineer    |
|    108 | rohit    |      25 |      30000 | data scientist |
|    109 | rohit    |      26 |      30000 | data scientist |
+--------+----------+---------+------------+----------------+
10 rows in set (0.000 sec)

MariaDB [mysql]>
```

Used UPDATE Command to update job role column value from “data analyst” to “data scientist”

5. Retrieve the job role which is having more than 3 employees.

```
MariaDB [mysql]> select job_role from employee group by job_role having count(*)>3;
+----------------+
| job_role       |
+----------------+
| data scientist |
| ml engineer    |
+----------------+
2 rows in set (0.008 sec)

MariaDB [mysql]>
```

Used GROUP BY Command with HAVING to select job roles with COUNT to count the job roles that are more then 3

6. Retrieve the last three rows sorted by columns salary in descending order, age in ascending order.

```
MariaDB [mysql]> select * from employee order by emp_salary desc,emp_age limit 7,3;
+--------+----------+---------+------------+----------------+
| emp_id | emp_name | emp_age | emp_salary | job_role       |
+--------+----------+---------+------------+----------------+
|    100 | john     |      22 |      20000 | data scientist |
|    107 | stella   |      23 |      20000 | ml engineer    |
|    106 | mary     |      24 |      15000 | data scientist |
+--------+----------+---------+------------+----------------+
3 rows in set (0.000 sec)

MariaDB [mysql]>
```

Using ORDER BY Command to order the table in ascending and descending order respective with emp_age and emp_salary, here limit specifies the last 3 rows

7. Retrieve distinct names of the employees whose salary is in between [50000,59999] using wildcards.

```
MariaDB [mysql]> select distinct emp_name from employee where emp_salary like '5____';
+----------+
| emp_name |
+----------+
| lee      |
| bob      |
| rock     |
+----------+
3 rows in set (0.001 sec)

MariaDB [mysql]>
```

wildcard character is used to substitute one or more characters, here we substituted last 4 digits with wildcards so they can be anything and starting value to look using WHERE and LIKE is 5. So finally, we get epm_salary between 50000 and 59999

8. Retrieve the job roles in sorted order of average salary.

```
MariaDB [mysql]> select job_role from employee group by job_role order by avg(emp_salary);
+----------------+
| job_role       |
+----------------+
| data scientist |
| ml engineer    |
+----------------+
2 rows in set (0.001 sec)

MariaDB [mysql]>
```

Using GROUP BY to select job roles with sorted order of average salary using ORDER BY with average salary

9. Update highest salary with lowest salary.
   Hint and Note: Use nested queries.

```
MariaDB [mysql]> update employee set emp_salary=(select min(emp_salary) from employee) where emp_salary=(select max(emp_salary) from employee);
Query OK, 1 row affected (0.075 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [mysql]> select * from employee;
+--------+----------+---------+------------+----------------+
| emp_id | emp_name | emp_age | emp_salary | job_role       |
+--------+----------+---------+------------+----------------+
|    100 | john     |      22 |      20000 | data scientist |
|    101 | ben      |      23 |      30000 | ml engineer    |
|    102 | lee      |      22 |      15000 | ml engineer    |
|    103 | bob      |      24 |      50000 | data scientist |
|    104 | rock     |      23 |      55000 | ml engineer    |
|    105 | ram      |      23 |      25000 | data scientist |
|    106 | mary     |      24 |      15000 | data scientist |
|    107 | stella   |      23 |      20000 | ml engineer    |
|    108 | rohit    |      25 |      30000 | data scientist |
|    109 | rohit    |      26 |      30000 | data scientist |
+--------+----------+---------+------------+----------------+
10 rows in set (0.000 sec)

MariaDB [mysql]>
```

Using UPDATE command with nested queries to get the highest salary and lowest salary then setting highest salary with lowest salary we found with query

10. Delete all employees whose age is greater than 65.

```
MariaDB [mysql]> select * from employee;
+--------+----------+---------+------------+----------------+
| emp_id | emp_name | emp_age | emp_salary | job_role       |
+--------+----------+---------+------------+----------------+
|    100 | john     |      22 |      20000 | data scientist |
|    101 | ben      |      23 |      30000 | ml engineer    |
|    102 | lee      |      22 |      15000 | ml engineer    |
|    103 | bob      |      70 |      50000 | data scientist |
|    104 | rock     |      66 |      55000 | ml engineer    |
|    105 | ram      |      23 |      25000 | data scientist |
|    106 | mary     |      68 |      15000 | data scientist |
|    107 | stella   |      23 |      20000 | ml engineer    |
|    108 | rohit    |      25 |      30000 | data scientist |
|    109 | rohit    |      26 |      30000 | data scientist |
+--------+----------+---------+------------+----------------+
10 rows in set (0.000 sec)

MariaDB [mysql]> delete from employee where emp_age>65;
Query OK, 3 rows affected (0.093 sec)

MariaDB [mysql]> select * from employee;
+--------+----------+---------+------------+----------------+
| emp_id | emp_name | emp_age | emp_salary | job_role       |
+--------+----------+---------+------------+----------------+
|    100 | john     |      22 |      20000 | data scientist |
|    101 | ben      |      23 |      30000 | ml engineer    |
|    102 | lee      |      22 |      15000 | ml engineer    |
|    105 | ram      |      23 |      25000 | data scientist |
|    107 | stella   |      23 |      20000 | ml engineer    |
|    108 | rohit    |      25 |      30000 | data scientist |
|    109 | rohit    |      26 |      30000 | data scientist |
+--------+----------+---------+------------+----------------+
7 rows in set (0.000 sec)

MariaDB [mysql]>
```

Using DELETE Command with where clause to delete entries with age values greater than 65