**1. Consider the following schema: STUDENT (USN, name, date_of_birth, branch, mark1, mark2,mark3, total, GPA).**
**Execute the following queries:**
**a. Update the column total by adding the columns mark1, mark2, mark3.**
**b. Find the GPA score of all the students.**
**c. Find the students who were born on a particular year of birth from the date_of_birth**
**d. List the students who are studying in a particular branch of study.**
**e. Find the maximum GPA score of the student branch-wise.**
**f. Find the students whose name starts with the alphabet "S".**
**g. Find the students whose name ends with the alphabet "AR".**
**h. Delete the student details whose USN is given as 1001.**
**give the answer for the given question in MariaDB**

=

```
MariaDB [lab]> CREATE TABLE STUDENT (
    ->     USN INT PRIMARY KEY,
    ->     name VARCHAR(50),
    ->     date_of_birth DATE,
    ->     branch VARCHAR(20),
    ->     mark1 INT,
    ->     mark2 INT,
    ->     mark3 INT,
    ->     total INT,
    ->     GPA DECIMAL(3,2)
    -> );
Query OK, 0 rows affected (0.040 sec)

MariaDB [lab]> INSERT INTO STUDENT (USN, name, date_of_birth, branch, mark1,
mark2, mark3, total, GPA) VALUES
    -> (1001, 'ravi', '2000-05-12', 'CSE', 80, 75, 90, NULL, 8.50),
    -> (1002, 'sruthi', '2001-11-23', 'ECE', 70, 65, 60, NULL, 7.20),
    -> (1003, 'amar', '2000-07-19', 'CSE', 85, 88, 92, NULL, 9.10),
    -> (1004, 'laksh', '1999-02-10', 'MECH', 60, 55, 65, NULL, 6.80),
    -> (1005, 'anu', '2000-12-02', 'CIVIL', 72, 78, 80, NULL, 8.00);
Query OK, 5 rows affected (0.034 sec)
Records: 5  Duplicates: 0  Warnings: 0

a)MariaDB [lab]> UPDATE STUDENT
    -> SET total = mark1 + mark2 + mark3;
Query OK, 5 rows affected (0.030 sec)
Rows matched: 5  Changed: 5  Warnings: 0

MariaDB [lab]> select * from student;
+------+--------+---------------+--------+-------+-------+-------+-------+------+
| USN  | name   | date_of_birth | branch | mark1 | mark2 | mark3 | total | GPA  |
+------+--------+---------------+--------+-------+-------+-------+-------+------+
| 1001 | ravi   | 2000-05-12    | CSE    |    80 |    75 |    90 |   245 | 8.50 |
| 1002 | sruthi | 2001-11-23    | ECE    |    70 |    65 |    60 |   195 | 7.20 |
| 1003 | amar   | 2000-07-19    | CSE    |    85 |    88 |    92 |   265 | 9.10 |
| 1004 | laksh  | 1999-02-10    | MECH   |    60 |    55 |    65 |   180 | 6.80 |
```

```
| 1005 | anu    | 2000-12-02    | CIVIL |    72 |    78 |    80 |   230 | 8.00 |
+------+--------+---------------+--------+-------+-------+-------+-------+------+
5 rows in set (0.028 sec)

b)MariaDB [lab]> SELECT USN, name, GPA FROM STUDENT;
+------+--------+------+
| USN  | name   | GPA  |
+------+--------+------+
| 1001 | ravi   | 8.50 |
| 1002 | sruthi | 7.20 |
| 1003 | amar   | 9.10 |
| 1004 | laksh  | 6.80 |
| 1005 | anu    | 8.00 |
+------+--------+------+
5 rows in set (0.002 sec)

c)MariaDB [lab]> SELECT * FROM STUDENT
    -> WHERE YEAR(date_of_birth) = 2000;
+------+------+---------------+--------+-------+-------+-------+-------+------+
| USN  | name | date_of_birth | branch | mark1 | mark2 | mark3 | total | GPA  |
+------+------+---------------+--------+-------+-------+-------+-------+------+
| 1001 | ravi | 2000-05-12    | CSE    |    80 |    75 |    90 |   245 | 8.50 |
| 1003 | amar | 2000-07-19    | CSE    |    85 |    88 |    92 |   265 | 9.10 |
| 1005 | anu  | 2000-12-02    | CIVIL |    72 |    78 |    80 |   230 | 8.00 |
+------+------+---------------+--------+-------+-------+-------+-------+------+
3 rows in set (0.027 sec)

d)MariaDB [lab]> SELECT * FROM STUDENT
    -> WHERE branch = 'CSE';
+------+------+---------------+--------+-------+-------+-------+-------+------+
| USN  | name | date_of_birth | branch | mark1 | mark2 | mark3 | total | GPA  |
+------+------+---------------+--------+-------+-------+-------+-------+------+
| 1001 | ravi | 2000-05-12    | CSE    |    80 |    75 |    90 |   245 | 8.50 |
| 1003 | amar | 2000-07-19    | CSE    |    85 |    88 |    92 |   265 | 9.10 |
+------+------+---------------+--------+-------+-------+-------+-------+------+
2 rows in set (0.006 sec)

e)MariaDB [lab]> SELECT branch, MAX(GPA) AS max_GPA
    -> FROM STUDENT
    -> GROUP BY branch;
+--------+---------+
| branch | max_GPA |
+--------+---------+
| CIVIL  |    8.00 |
| CSE    |    9.10 |
| ECE    |    7.20 |
| MECH   |    6.80 |
+--------+---------+
4 rows in set (0.029 sec)
```

f)MariaDB [lab]> SELECT * FROM STUDENT
   -> WHERE name LIKE 'S%';
+------+--------+--------------+--------+-------+-------+-------+-------+------+
| USN  | name   | date_of_birth | branch | mark1 | mark2 | mark3 | total | GPA  |
+------+--------+--------------+--------+-------+-------+-------+-------+------+
| 1002 | sruthi | 2001-11-23    | ECE    |    70 |    65 |    60 |   195 | 7.20 |
+------+--------+--------------+--------+-------+-------+-------+-------+------+
1 row in set (0.027 sec)

g)MariaDB [lab]> SELECT * FROM STUDENT
   -> WHERE name LIKE '%AR';
+------+------+--------------+--------+-------+-------+-------+-------+------+
| USN  | name | date_of_birth | branch | mark1 | mark2 | mark3 | total | GPA  |
+------+------+--------------+--------+-------+-------+-------+-------+------+
| 1003 | amar | 2000-07-19    | CSE    |    85 |    88 |    92 |   265 | 9.10 |
+------+------+--------------+--------+-------+-------+-------+-------+------+
1 row in set (0.001 sec)

h)MariaDB [lab]> DELETE FROM STUDENT
   -> WHERE USN = 1001;
Query OK, 1 row affected (0.027 sec)

MariaDB [lab]> SELECT * FROM STUDENT;
+------+--------+--------------+--------+-------+-------+-------+-------+------+
| USN  | name   | date_of_birth | branch | mark1 | mark2 | mark3 | total | GPA  |
+------+--------+--------------+--------+-------+-------+-------+-------+------+
| 1002 | sruthi | 2001-11-23    | ECE    |    70 |    65 |    60 |   195 | 7.20 |
| 1003 | amar   | 2000-07-19    | CSE    |    85 |    88 |    92 |   265 | 9.10 |
| 1004 | laksh  | 1999-02-10    | MECH   |    60 |    55 |    65 |   180 | 6.80 |
| 1005 | anu    | 2000-12-02    | CIVIL  |    72 |    78 |    80 |   230 | 8.00 |
+------+--------+--------------+--------+-------+-------+-------+-------+------+
4 rows in set (0.001 sec)

**2. Consider the following database of student enrollment in courses and books adopted for each course.**
**STUDENT (regno#: string, name: string, major: string, bdate: date)**
**COURSE (course#: int, cname: string, dept: String)**
**TEXT(book_ISBN#: int, book_title: string, publisher: string,author:string)**
**ENROLL (regno#: string, course#: int, sem: int, marks: int)**
**BOOK_ADOPTION (course#: int, sem: int, book_ ISBN: int)**
**Execute SQL queries for the following:**
**a. List out the student details, and their course details. The records should be ordered in asemester-wise manner.**
**b. List out the student details under a particular department whose name is ordered semester-wise.**
**c. List out all the book details under a particular course.**
**d. Find out the Courses in which the number of students studying are more than 2.**
**e. Find out the Publisher who has published more than 2 books.**

```
MariaDB [(none)]> use dbms2;
Database changed
MariaDB [dbms2]> CREATE TABLE STUDENT (
    ->     regno VARCHAR(10) PRIMARY KEY,
    ->     name VARCHAR(50),
    ->     major VARCHAR(30),
    ->     bdate DATE
    -> );
Query OK, 0 rows affected (0.016 sec)

MariaDB [dbms2]> CREATE TABLE TEXT (
    ->     book_ISBN INT PRIMARY KEY,
    ->     book_title VARCHAR(100),
    ->     publisher VARCHAR(50),
    ->     author VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.007 sec)

MariaDB [dbms2]> CREATE TABLE COURSE (
    -> course INT PRIMARY KEY,
    -> cname VARCHAR(50),
    ->     dept VARCHAR(30)
    -> );
Query OK, 0 rows affected (0.039 sec)
```

```
MariaDB [dbms2]> CREATE TABLE ENROLL (
    ->     regno VARCHAR(10),
    ->     course INT,
    ->     sem INT,
    ->     marks INT,
    ->     PRIMARY KEY (regno, course),
    ->     FOREIGN KEY (regno) REFERENCES STUDENT(regno),
    ->     FOREIGN KEY (course) REFERENCES COURSE(course)
    -> );
Query OK, 0 rows affected (0.036 sec)

MariaDB [dbms2]> CREATE TABLE BOOK_ADOPTION (
    ->     course INT,
    ->     sem INT,
    ->     book_ISBN INT,
    ->     PRIMARY KEY (course, sem, book_ISBN),
    ->     FOREIGN KEY (course) REFERENCES COURSE(course),
    ->     FOREIGN KEY (book_ISBN) REFERENCES TEXT(book_ISBN)
    -> );
Query OK, 0 rows affected (0.014 sec)


MariaDB [dbms2]> INSERT INTO STUDENT VALUES ('S1', 'Aishwarya', 'CS', '2003-05-
12');
Query OK, 1 row affected (0.029 sec)

MariaDB [dbms2]> INSERT INTO STUDENT VALUES ('S2', 'Rahul', 'IT', '2002-11-02');
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms2]> INSERT INTO STUDENT VALUES ('S3', 'Sneha', 'CS', '2003-08-21');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO STUDENT VALUES ('S4', 'Arjun', 'EC', '2001-12-14');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms2]> INSERT INTO STUDENT VALUES ('S5', 'Kiran', 'CS', '2002-06-06');
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms2]>
MariaDB [dbms2]> INSERT INTO COURSE VALUES (101, 'DBMS', 'CS');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO COURSE VALUES (102, 'Networks', 'IT');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO COURSE VALUES (103, 'Digital Logic', 'EC');
Query OK, 1 row affected (0.001 sec)
```

```
MariaDB [dbms2]> INSERT INTO TEXT VALUES (1001, 'Database System Concepts',
'McGrawHill', 'Korth');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO TEXT VALUES (1002, 'Computer Networks', 'Pearson',
'Tanenbaum');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO TEXT VALUES (1003, 'Digital Logic Design', 'Wiley',
'Morris Mano');
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms2]> INSERT INTO TEXT VALUES (1004, 'Operating Systems',
'McGrawHill', 'Silberschatz');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms2]> INSERT INTO TEXT VALUES (1005, 'Data Structures', 'Pearson', 'Mark
Allen');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO TEXT VALUES (1006, 'Machine Learning Basics',
'McGrawHill', 'Andrew Ng');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO ENROLL VALUES ('S1', 101, 1, 85);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO ENROLL VALUES ('S2', 101, 1, 78);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO ENROLL VALUES ('S3', 101, 2, 90);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO ENROLL VALUES ('S4', 103, 2, 75);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO ENROLL VALUES ('S5', 101, 1, 88);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO BOOK_ADOPTION VALUES (101, 1, 1001);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO BOOK_ADOPTION VALUES (102, 1, 1002);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO BOOK_ADOPTION VALUES (103, 2, 1003);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms2]> INSERT INTO BOOK_ADOPTION VALUES (101, 2, 1004);
Query OK, 1 row affected (0.001 sec)
```

```
MariaDB [dbms2]> INSERT INTO BOOK_ADOPTION VALUES (101, 1, 1005);
Query OK, 1 row affected (0.003 sec)

a) MariaDB [dbms2]> SELECT S.regno, S.name, S.major, S.bdate,
    ->        C.course, C.cname, C.dept,
    ->        E.sem, E.marks
    -> FROM STUDENT S
    -> JOIN ENROLL E ON S.regno = E.regno
    -> JOIN COURSE C ON E.course = C.course
    -> ORDER BY E.sem;
+-------+-----------+-------+------------+--------+---------------+------+------+-------+
| regno | name      | major | bdate      | course | cname         | dept | sem  | marks |
+-------+-----------+-------+------------+--------+---------------+------+------+-------+
| S2    | Rahul     | IT    | 2002-11-02 |    101 | DBMS          | CS   |    1 |    78 |
| S1    | Aishwarya | CS    | 2003-05-12 |    101 | DBMS          | CS   |    1 |    85 |
| S5    | Kiran     | CS    | 2002-06-06 |    101 | DBMS          | CS   |    1 |    88 |
| S4    | Arjun     | EC    | 2001-12-14 |    103 | Digital Logic | EC   |    2 |    75 |
| S3    | Sneha     | CS    | 2003-08-21 |    101 | DBMS          | CS   |    2 |    90 |
+-------+-----------+-------+------------+--------+---------------+------+------+-------+
5 rows in set (0.002 sec)

b) MariaDB [dbms2]> SELECT S.regno, S.name, S.major, S.bdate,
    ->        C.course, C.cname, C.dept, E.sem
    -> FROM STUDENT S
    -> JOIN ENROLL E ON S.regno = E.regno
    -> JOIN COURSE C ON E.course = C.course
    -> WHERE C.dept = 'CS'
    -> ORDER BY E.sem;
+-------+-----------+-------+------------+--------+-------+------+------+
| regno | name      | major | bdate      | course | cname | dept | sem  |
+-------+-----------+-------+------------+--------+-------+------+------+
| S2    | Rahul     | IT    | 2002-11-02 |    101 | DBMS  | CS   |    1 |
| S1    | Aishwarya | CS    | 2003-05-12 |    101 | DBMS  | CS   |    1 |
| S5    | Kiran     | CS    | 2002-06-06 |    101 | DBMS  | CS   |    1 |
| S3    | Sneha     | CS    | 2003-08-21 |    101 | DBMS  | CS   |    2 |
+-------+-----------+-------+------------+--------+-------+------+------+
4 rows in set (0.001 sec)

c) MariaDB [dbms2]> SELECT T.book_ISBN, T.book_title, T.publisher, T.author
    -> FROM TEXT T
    -> JOIN BOOK_ADOPTION B ON T.book_ISBN = B.book_ISBN
    -> WHERE B.course = 101;
+-----------+-------------------------+------------+--------------+
| book_ISBN | book_title              | publisher  | author       |
+-----------+-------------------------+------------+--------------+
|      1001 | Database System Concepts | McGrawHill | Korth        |
|      1005 | Data Structures          | Pearson    | Mark Allen   |
|      1004 | Operating Systems        | McGrawHill | Silberschatz |
+-----------+-------------------------+------------+--------------+
3 rows in set (0.000 sec)
```

d) MariaDB [dbms2]> SELECT C.course, C.cname, COUNT(E.regno) AS student_count
    -> FROM COURSE C
    -> JOIN ENROLL E ON C.course = E.course
    -> GROUP BY C.course, C.cname
    -> HAVING COUNT(E.regno) > 2;

| course | cname | student_count |
|---|---|---|
| 101 | DBMS | 4 |

1 row in set (0.000 sec)

e) MariaDB [dbms2]> SELECT publisher, COUNT(*) AS book_count
    -> FROM TEXT
    -> GROUP BY publisher
    -> HAVING COUNT(*) > 2;

| publisher | book_count |
|---|---|
| McGrawHill | 3 |

1 row in set (0.000 sec)

**3. Design an ER-diagram for the following scenario, Convert the same into a relational model and then solve the following queries. Consider a Cricket Tournament "ABC CUP" organized by an organization. In the tournament, many teams are contesting each having a Teamid, Team_Name, City and a coach. Each team is uniquely identified by using Teamid. A team can have many Players and a captain. Each player is uniquely identified by Playerid, having a Name, and multiple phone numbers, age. A player represents only one team. There are many Stadiums to conduct matches. Each stadium is identified using Stadiumid, having a stadium_name, Address (involves city,area_name, pincode). A team can play many matches. Each match is played between the two teams on the scheduled date and time in the predefined Stadium. Each match is identified uniquely by using Matchid. Each match won by any of the one team that also wants to record in the database. For each match man_of_the match award given to a player.**
**Execute the following queries:**
**a. Display the youngest player (in terms of age) Name, Team name, age in which he belongs to the tournament.**
**b. List the details of the stadium where the maximum number of matches were played.**
**c. List the details of the player who is not a captain but got the man_of_match award at least in two matches.**
**d. Display the Team details who won the maximum matches.**
**e. Display the team's name where all its won matches played in the same stadium.**

```
MariaDB [dbms2]> CREATE TABLE TEAM (
    ->     Teamid INT PRIMARY KEY,
    ->     Team_Name VARCHAR(50),
    ->     City VARCHAR(50),
    ->     Coach VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.035 sec)

MariaDB [dbms2]>
MariaDB [dbms2]> CREATE TABLE PLAYER (
    ->     Playerid INT PRIMARY KEY,
    ->     Name VARCHAR(50),
    ->     Age INT,
    ->     Teamid INT,
    ->     FOREIGN KEY (Teamid) REFERENCES TEAM(Teamid)
    -> );
Query OK, 0 rows affected (0.014 sec)

MariaDB [dbms2]>
MariaDB [dbms2]> CREATE TABLE PLAYER_PHONE (
    ->     Playerid INT,
    ->     Phone_No VARCHAR(15),
    ->     PRIMARY KEY (Playerid, Phone_No),
    ->     FOREIGN KEY (Playerid) REFERENCES PLAYER(Playerid)
    -> );
Query OK, 0 rows affected (0.011 sec)
```

```
MariaDB [dbms2]>
MariaDB [dbms2]> CREATE TABLE STADIUM (
    ->     Stadiumid INT PRIMARY KEY,
    ->     Stadium_Name VARCHAR(50),
    ->     City VARCHAR(50),
    ->     Area_Name VARCHAR(50),
    ->     Pincode VARCHAR(10)
    -> );
Query OK, 0 rows affected (0.011 sec)

MariaDB [dbms2]> CREATE TABLE MATCHDETAIL(
    ->     Matchid INT PRIMARY KEY,
    ->     Match_Date DATE,
    ->     Match_Time TIME,
    ->     Stadiumid INT,
    ->     Team1 INT,
    ->     Team2 INT,
    ->     Winner_Teamid INT,
    ->     MOM_Playerid INT,
    ->     FOREIGN KEY(Stadiumid) REFERENCES STADIUM(Stadiumid),
    ->     FOREIGN KEY(Team1) REFERENCES TEAM(Teamid),
    ->     FOREIGN KEY(Team2) REFERENCES TEAM(Teamid),
    ->     FOREIGN KEY(Winner_Teamid) REFERENCES TEAM(Teamid),
    ->     FOREIGN KEY(MOM_Playerid) REFERENCES PLAYER(Playerid)
    -> );
Query OK, 0 rows affected (0.023 sec)

INSERT INTO TEAM VALUES (1, 'Warriors', 'Mumbai', 'Coach A', NULL);
INSERT INTO TEAM VALUES (2, 'Titans', 'Delhi', 'Coach B', NULL);
INSERT INTO TEAM VALUES (3, 'Rangers', 'Bangalore', 'Coach C', NULL);


INSERT INTO PLAYER VALUES (101, 'Rohit', 24, 1);
INSERT INTO PLAYER VALUES (102, 'Virat', 22, 1);
INSERT INTO PLAYER VALUES (103, 'Hardik', 27, 1);

INSERT INTO PLAYER VALUES (201, 'Dhoni', 28, 2);
INSERT INTO PLAYER VALUES (202, 'Pant', 21, 2);
INSERT INTO PLAYER VALUES (203, 'Iyer', 23, 2);

INSERT INTO PLAYER VALUES (301, 'Rahul', 26, 3);
INSERT INTO PLAYER VALUES (302, 'Samson', 25, 3);
INSERT INTO PLAYER VALUES (303, 'Shami', 29, 3);

INSERT INTO PLAYER_PHONE VALUES (101, '9876543210');
INSERT INTO PLAYER_PHONE VALUES (102, '9123456780');
INSERT INTO PLAYER_PHONE VALUES (201, '9988776655');
```

```
UPDATE TEAM SET Captain_Playerid = 101 WHERE Teamid = 1;  -- Rohit
UPDATE TEAM SET Captain_Playerid = 201 WHERE Teamid = 2;  -- Dhoni
UPDATE TEAM SET Captain_Playerid = 301 WHERE Teamid = 3;  -- Rahul

INSERT INTO STADIUM VALUES (11, 'Wankhede', 'Mumbai', 'Marine Drive', '400020');
INSERT INTO STADIUM VALUES (12, 'Eden Gardens', 'Kolkata', 'BBD Bagh', '700001');
INSERT INTO STADIUM VALUES (13, 'Chinnaswamy', 'Bangalore', 'MG Road', '560001');

INSERT INTO MATCHDETAIL VALUES (1001, '2025-01-10', '16:00:00', 11, 1, 2, 1, 102);
INSERT INTO MATCHDETAIL VALUES (1002, '2025-01-15', '19:00:00', 11, 1, 3, 3, 301);
INSERT INTO MATCHDETAIL VALUES (1003, '2025-01-20', '18:00:00', 12, 2, 3, 2, 202);
INSERT INTO MATCHDETAIL VALUES (1004, '2025-01-25', '20:00:00', 11, 1, 2, 1, 102);


a) MariaDB [dbms2]> SELECT P.Name, T.Team_Name, P.Age
    -> FROM PLAYER P
    -> JOIN TEAM T ON P.Teamid = T.Teamid
    -> WHERE P.Age = (SELECT MIN(Age) FROM PLAYER);
+------+-----------+------+
| Name | Team_Name | Age  |
+------+-----------+------+
| Pant | Titans    |   21 |
+------+-----------+------+
1 row in set (0.002 sec)

b) MariaDB [dbms2]> SELECT S.Stadiumid, S.Stadium_Name, S.City, S.Area_Name,
S.Pincode, COUNT(M.Matchid) AS Total_Matches
    -> FROM STADIUM S
    -> JOIN MATCHDETAIL M ON S.Stadiumid = M.Stadiumid
    -> GROUP BY S.Stadiumid, S.Stadium_Name, S.City, S.Area_Name, S.Pincode
    -> ORDER BY Total_Matches DESC
    -> LIMIT 1;
+-----------+-------------+--------+-------------+---------+---------------+
| Stadiumid | Stadium_Name | City   | Area_Name    | Pincode | Total_Matches |
+-----------+-------------+--------+-------------+---------+---------------+
|        11 | Wankhede     | Mumbai | Marine Drive | 400020  |             3 |
+-----------+-------------+--------+-------------+---------+---------------+
1 row in set (0.001 sec)

c) MariaDB [dbms2]> SELECT P.Playerid, P.Name, COUNT(M.Matchid) AS MOM_Count
    -> FROM PLAYER P
    -> JOIN MATCHDETAIL M ON P.Playerid = M.MOM_Playerid
    -> WHERE P.Playerid NOT IN (SELECT Captain_Playerid FROM TEAM)
    -> GROUP BY P.Playerid, P.Name
    -> HAVING COUNT(M.Matchid) >= 2;
+----------+-------+-----------+
| Playerid | Name  | MOM_Count |
+----------+-------+-----------+
|      102 | Virat |         2 |
+----------+-------+-----------+
```

1 row in set (0.002 sec)

d) MariaDB [dbms2]> SELECT T.Teamid, T.Team_Name, COUNT(M.Matchid) AS Wins
    -> FROM TEAM T
    -> JOIN MATCHDETAIL M ON T.Teamid = M.Winner_Teamid
    -> GROUP BY T.Teamid, T.Team_Name
    -> ORDER BY Wins DESC
    -> LIMIT 1;
+--------+-----------+------+
| Teamid | Team_Name | Wins |
+--------+-----------+------+
|      1 | Warriors  |    2 |
+--------+-----------+------+
1 row in set (0.001 sec)

e) MariaDB [dbms2]> SELECT T.Teamid, T.Team_Name
    -> FROM TEAM T
    -> JOIN MATCHDETAIL M ON T.Teamid = M.Winner_Teamid
    -> GROUP BY T.Teamid, T.Team_Name
    -> HAVING COUNT(DISTINCT M.Stadiumid) = 1;
+--------+-----------+
| Teamid | Team_Name |
+--------+-----------+
|      1 | Warriors  |
|      2 | Titans    |
|      3 | Rangers   |
+--------+-----------+
3 rows in set (0.001 sec)

**4. A country wants to conduct an election for parliament. A country has many constituencies. Each constituency is identified uniquely by Constituency_id, having the Name, belongs to a state, Number_of_voters. A constituency can have many voters. Each voter is uniquely identified by using Voter id, having the Name, age, address (involves Houseno, city, state, pincode). Each voter belongs to only one constituency. Many candidates are contesting in the election. Each candidate is uniquely identified by using candidate_id, having Name, phone_no, age, state. A candidate belongs to only one party. There are many parties. Each party is uniquely identified by using Party_id, having Party_Name, Party_symbol. A candidate can contest from many constituencies under the same party. A party can have many candidates contesting from different constituencies. No constituency having the candidates from the same party. A constituency can have many contesting candidates belonging to different parties. Each voter votes for only one candidate of his/her constituency.**
**Execute the following queries:**

a. **List the details of the candidates who are contesting from more than one constituency which belongs to different states.**

b. **Display the state name having the maximum number of constituencies.**

c. **Create a stored procedure to insert the tuple into the voter table by checking the voter age. If the voter's age is at least 18 years old, then insert the tuple into the voter else display the"Not an eligible voter msg".**

d. **Create a stored procedure to display the number_of_voters in the specified constituency. Where the constituency name is passed as an argument to the stored procedure.**

e. **Create a TRIGGER to UPDATE the count of "Number of voters" of the respectiveconstituency in the "CONSTITUENCY" table, after inserting a tuple into the "VOTERS" table.give the create table ,insert table and queries for the given questions**

```
MariaDB [dbms4]> CREATE TABLE PARTY (
    ->     Party_id INT PRIMARY KEY,
    ->     Party_Name VARCHAR(50) NOT NULL,
    ->     Party_symbol VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.020 sec)

MariaDB [dbms4]> CREATE TABLE CONSTITUENCY (
    ->     Constituency_id INT PRIMARY KEY,
    ->     Name VARCHAR(50) NOT NULL,
    ->     State VARCHAR(50) NOT NULL,
    ->     Number_of_voters INT DEFAULT 0
    -> );
Query OK, 0 rows affected (0.017 sec)
```

```
MariaDB [dbms4]> CREATE TABLE CANDIDATE (
    ->     Candidate_id INT PRIMARY KEY,
    ->     Name VARCHAR(50) NOT NULL,
    ->     Phone_no VARCHAR(15),
    ->     Age INT,
    ->     State VARCHAR(50),
    ->     Party_id INT,
    ->     FOREIGN KEY (Party_id) REFERENCES PARTY(Party_id)
    -> );
Query OK, 0 rows affected (0.020 sec)

MariaDB [dbms4]> CREATE TABLE CANDIDATE_CONSTITUENCY (
    ->     Candidate_id INT,
    ->     Constituency_id INT,
    ->     PRIMARY KEY (Candidate_id, Constituency_id),
    ->     FOREIGN KEY (Candidate_id) REFERENCES CANDIDATE(Candidate_id),
    ->     FOREIGN KEY (Constituency_id) REFERENCES
CONSTITUENCY(Constituency_id)
    -> );
Query OK, 0 rows affected (0.018 sec)

MariaDB [dbms4]> CREATE TABLE VOTER (
    ->     Voter_id INT PRIMARY KEY,
    ->     Name VARCHAR(50),
    ->     Age INT,
    ->     HouseNo VARCHAR(10),
    ->     City VARCHAR(50),
    ->     State VARCHAR(50),
    ->     Pincode VARCHAR(10),
    ->     Constituency_id INT,
    ->     FOREIGN KEY (Constituency_id) REFERENCES CONSTITUENCY
(Constituency_id)
    -> );
Query OK, 0 rows affected (0.015 sec)

MariaDB [dbms4]>  INSERT INTO PARTY VALUES (1, 'Democratic Party', 'DP');
Query OK, 1 row affected (0.007 sec)

MariaDB [dbms4]> INSERT INTO PARTY VALUES (2, 'National Party', 'NP');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO PARTY VALUES (3, 'People’s Front', 'PF');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO CONSTITUENCY VALUES (101, 'North City',
'Karnataka', 0);
Query OK, 1 row affected (0.003 sec)
```

```
MariaDB [dbms4]> INSERT INTO CONSTITUENCY VALUES (102, 'South City',
'Karnataka', 0);
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms4]> INSERT INTO CONSTITUENCY VALUES (103, 'East Town',
'Maharashtra', 0);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO CONSTITUENCY VALUES (104, 'West Town',
'Maharashtra', 0);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO CANDIDATE VALUES (201, 'Ravi Kumar',
'9876543210', 45, 'Karnataka', 1);
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms4]> INSERT INTO CANDIDATE VALUES (202, 'Meera Singh',
'9876500000', 39, 'Maharashtra', 2);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO CANDIDATE VALUES (203, 'Arjun Rao', '9876511111',
50, 'Karnataka', 3);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO CANDIDATE VALUES (204, 'Sneha Patel',
'9876522222', 42, 'Maharashtra', 1);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]>
MariaDB [dbms4]> INSERT INTO CANDIDATE_CONSTITUENCY VALUES (201, 101);
Query OK, 1 row affected (0.004 sec)

MariaDB [dbms4]> INSERT INTO CANDIDATE_CONSTITUENCY VALUES (201, 103);
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms4]> INSERT INTO CANDIDATE_CONSTITUENCY VALUES (202, 104);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO CANDIDATE_CONSTITUENCY VALUES (203, 102);
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms4]> INSERT INTO CANDIDATE_CONSTITUENCY VALUES (204, 103);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO VOTER VALUES (301, 'Amit Sharma', 25, '12A',
'Bangalore', 'Karnataka', '560001', 101);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms4]> INSERT INTO VOTER VALUES (302, 'Neha Verma', 32, '45B',
'Bangalore', 'Karnataka', '560002', 101);
```

```
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO VOTER VALUES (303, 'Suresh Reddy', 28, '78C', 'Pune',
'Maharashtra', '411001', 103);
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> SELECT c.Candidate_id, c.Name, c.Age, c.State, p.Party_Name
    -> FROM CANDIDATE c
    -> JOIN CANDIDATE_CONSTITUENCY cc1 ON c.Candidate_id = cc1.Candidate_id
    -> JOIN CONSTITUENCY con1 ON cc1.Constituency_id = con1.Constituency_id
    -> JOIN CANDIDATE_CONSTITUENCY cc2 ON c.Candidate_id = cc2.Candidate_id
    -> JOIN CONSTITUENCY con2 ON cc2.Constituency_id = con2.Constituency_id
    -> JOIN PARTY p ON c.Party_id = p.Party_id
    -> WHERE con1.State <> con2.State;
+--------------+------------+------+-----------+-----------------+
| Candidate_id | Name       | Age  | State     | Party_Name      |
+--------------+------------+------+-----------+-----------------+
|          201 | Ravi Kumar |   45 | Karnataka | Democratic Party |
|          201 | Ravi Kumar |   45 | Karnataka | Democratic Party |
+--------------+------------+------+-----------+-----------------+
2 rows in set (0.002 sec)

MariaDB [dbms4]> SELECT State, COUNT(*) AS No_of_constituencies
    -> FROM CONSTITUENCY
    -> GROUP BY State
    -> ORDER BY No_of_constituencies DESC
    -> LIMIT 1;
+-------------+----------------------+
| State       | No_of_constituencies |
+-------------+----------------------+
| Maharashtra |                    2 |
+-------------+----------------------+
1 row in set (0.001 sec)

MariaDB [dbms4]> DELIMITER $$
MariaDB [dbms4]>
MariaDB [dbms4]> CREATE PROCEDURE InsertVoter(
    ->     IN v_id INT, IN v_name VARCHAR(50), IN v_age INT,
    ->     IN v_hno VARCHAR(10), IN v_city VARCHAR(50), IN v_state VARCHAR(50),
    ->     IN v_pin VARCHAR(10), IN v_const INT
    -> )
    -> BEGIN
    ->     IF v_age >= 18 THEN
    ->         INSERT INTO VOTER(Voter_id, Name, Age, HouseNo, City, State, Pincode,
Constituency_id)
    ->         VALUES(v_id, v_name, v_age, v_hno, v_city, v_state, v_pin, v_const);
    ->     ELSE
    ->         SELECT 'Not an eligible voter' AS Message;
    ->     END IF;
    -> END $$
```

```
Query OK, 0 rows affected (0.040 sec)

MariaDB [dbms4]>
MariaDB [dbms4]> DELIMITER ;
MariaDB [dbms4]> DELIMITER $$
MariaDB [dbms4]>
MariaDB [dbms4]> CREATE PROCEDURE VoterCount(IN c_name VARCHAR(50))
    -> BEGIN
    ->     SELECT COUNT(*) AS Total_Voters
    ->     FROM VOTER v
    ->     JOIN CONSTITUENCY c ON v.Constituency_id = c.Constituency_id
    ->     WHERE c.Name = c_name;
    -> END $$
Query OK, 0 rows affected (0.007 sec)

MariaDB [dbms4]>
MariaDB [dbms4]> DELIMITER ;
MariaDB [dbms4]> DELIMITER $$
MariaDB [dbms4]>
MariaDB [dbms4]> CREATE TRIGGER update_voter_count
    -> AFTER INSERT ON VOTER
    -> FOR EACH ROW
    -> BEGIN
    ->     UPDATE CONSTITUENCY
    ->     SET Number_of_voters = Number_of_voters + 1
    ->     WHERE Constituency_id = NEW.Constituency_id;
    -> END $$
Query OK, 0 rows affected (0.014 sec)

MariaDB [dbms4]>
MariaDB [dbms4]> DELIMITER ;
MariaDB [dbms4]>
```

**5. Design an ER-diagram for the following scenario, Convert the same into a relational model, normalize Relations into a suitable Normal form and then solve the following queries. A country can have many Tourist places. Each Tourist place is identified by using tourist_place_id, having a name, belonging to a state, capital city of that state, history. There are many Tourists visiting tourist places every year. Each tourist is identified uniquely by using Tourist_id, having a Name, age, Country and multiple email ids. A tourist visits many Tourist places, it is also required to record the visited_date in the database. A tourist can visit a Tourist place many times at different dates. A Tourist place can be visited by many tourists either on the same date or at different dates.**

**Queries:**

**a. List the state name which has the maximum number of tourist places.**

**b. List details of Tourist places where the maximum number of tourists visited.**

**c. List the details of tourists visiting all tourist places from the state "KARNATAKA".**

**d.Display the details of the tourists who visited at least one tourist place of the state butvisited all tourist places in all states.**

**e. Display the details of the tourist places visited by the tourists of all countries.give the create table ,insert table and queries for the given questions**

```
MariaDB [dbms4]> CREATE TABLE TOURIST_PLACE (
    ->     Tourist_place_id INT PRIMARY KEY,
    ->     Name VARCHAR(100),
    ->     State VARCHAR(50),
    ->     Capital_city VARCHAR(50),
    ->     History TEXT
    -> );
Query OK, 0 rows affected (0.015 sec)

MariaDB [dbms4]> CREATE TABLE TOURIST (
    ->     Tourist_id INT PRIMARY KEY,
    ->     Name VARCHAR(100),
    ->     Age INT,
    ->     Country VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.015 sec)

MariaDB [dbms4]> CREATE TABLE TOURIST_EMAIL (
    ->     Tourist_id INT,
    ->     Email_id VARCHAR(100),
    ->     PRIMARY KEY (Tourist_id, Email_id),
    ->     FOREIGN KEY (Tourist_id) REFERENCES TOURIST(Tourist_id)
    -> );
Query OK, 0 rows affected (0.017 sec)

MariaDB [dbms4]> CREATE TABLE VISIT (
```

```
    ->     Tourist_id INT,
    ->     Tourist_place_id INT,
    ->     Visited_date DATE,
    ->     PRIMARY KEY (Tourist_id, Tourist_place_id, Visited_date),
    ->     FOREIGN KEY (Tourist_id) REFERENCES TOURIST(Tourist_id),
    ->     FOREIGN KEY (Tourist_place_id) REFERENCES
TOURIST_PLACE(Tourist_place_id)
    -> );
Query OK, 0 rows affected (0.018 sec)

MariaDB [dbms4]> INSERT INTO TOURIST_PLACE VALUES (1, 'Mysore Palace',
'KARNATAKA', 'Bangalore', 'Historic royal palace');
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms4]> INSERT INTO TOURIST_PLACE VALUES (2, 'Hampi',
'KARNATAKA', 'Bangalore', 'UNESCO heritage site');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO TOURIST_PLACE VALUES (3, 'Gateway of India',
'MAHARASHTRA', 'Mumbai', 'Historic monument');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO TOURIST_PLACE VALUES (4, 'Ajanta Caves',
'MAHARASHTRA', 'Mumbai', 'Ancient caves');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO TOURIST_PLACE VALUES (5, 'Taj Mahal', 'UTTAR
PRADESH', 'Lucknow', 'World wonder');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]>
MariaDB [dbms4]> INSERT INTO TOURIST VALUES (101, 'Amit Sharma', 30, 'India');
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms4]> INSERT INTO TOURIST VALUES (102, 'John Smith', 40, 'USA');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO TOURIST VALUES (103, 'Maria Garcia', 28, 'Spain');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]>
MariaDB [dbms4]> INSERT INTO TOURIST_EMAIL VALUES (101, 'amit@gmail.com');
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms4]> INSERT INTO TOURIST_EMAIL VALUES (101, 'amit@work.com');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO TOURIST_EMAIL VALUES (102, 'john@yahoo.com');
Query OK, 1 row affected (0.001 sec)
```

```
MariaDB [dbms4]> INSERT INTO TOURIST_EMAIL VALUES (103,
'maria@hotmail.com');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]>
MariaDB [dbms4]> INSERT INTO VISIT VALUES (101, 1, '2024-01-01');
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms4]> INSERT INTO VISIT VALUES (101, 2, '2024-02-10');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms4]> INSERT INTO VISIT VALUES (101, 5, '2024-03-05');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO VISIT VALUES (102, 3, '2024-02-15');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO VISIT VALUES (102, 5, '2024-04-20');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO VISIT VALUES (103, 1, '2024-02-25');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO VISIT VALUES (103, 2, '2024-03-01');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO VISIT VALUES (103, 3, '2024-03-15');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO VISIT VALUES (103, 4, '2024-03-25');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]> INSERT INTO VISIT VALUES (103, 5, '2024-04-05');
Query OK, 1 row affected (0.001 sec)

MariaDB [dbms4]>
MariaDB [dbms4]> SELECT State, COUNT(*) AS No_of_places
    -> FROM TOURIST_PLACE
    -> GROUP BY State
    -> ORDER BY No_of_places DESC
    -> LIMIT 1;
+-------------+--------------+
| State       | No_of_places |
+-------------+--------------+
| MAHARASHTRA |            2 |
+-------------+--------------+
1 row in set (0.001 sec)

MariaDB [dbms4]>
MariaDB [dbms4]>
```

```
MariaDB [dbms4]>
MariaDB [dbms4]> SELECT tp.Tourist_place_id, tp.Name, tp.State, COUNT(DISTINCT
v.Tourist_id) AS No_of_tourists
    -> FROM TOURIST_PLACE tp
    -> JOIN VISIT v ON tp.Tourist_place_id = v.Tourist_place_id
    -> GROUP BY tp.Tourist_place_id, tp.Name, tp.State
    -> ORDER BY No_of_tourists DESC
    -> LIMIT 1;
+-----------------+-----------+--------------+----------------+
| Tourist_place_id | Name      | State        | No_of_tourists |
+-----------------+-----------+--------------+----------------+
|               5 | Taj Mahal | UTTAR PRADESH |              3 |
+-----------------+-----------+--------------+----------------+
1 row in set (0.001 sec)

MariaDB [dbms4]> SELECT t.Tourist_id, t.Name, t.Country
    -> FROM TOURIST t
    -> WHERE NOT EXISTS (
    ->     SELECT tp.Tourist_place_id
    ->     FROM TOURIST_PLACE tp
    ->     WHERE tp.State = 'KARNATAKA'
    ->     EXCEPT
    ->     SELECT v.Tourist_place_id
    ->     FROM VISIT v
    ->     WHERE v.Tourist_id = t.Tourist_id
    -> );
+------------+--------------+---------+
| Tourist_id | Name         | Country |
+------------+--------------+---------+
|        101 | Amit Sharma  | India   |
|        103 | Maria Garcia | Spain   |
+------------+--------------+---------+
2 rows in set (0.003 sec)

MariaDB [dbms4]> SELECT t.Tourist_id, t.Name, t.Country
    -> FROM TOURIST t
    -> WHERE NOT EXISTS (
    ->     SELECT DISTINCT tp.State
    ->     FROM TOURIST_PLACE tp
    ->     EXCEPT
    ->     SELECT DISTINCT tp2.State
    ->     FROM VISIT v
    ->     JOIN TOURIST_PLACE tp2 ON v.Tourist_place_id = tp2.Tourist_place_id
    ->     WHERE v.Tourist_id = t.Tourist_id
    -> );
+------------+--------------+---------+
| Tourist_id | Name         | Country |
+------------+--------------+---------+
|        103 | Maria Garcia | Spain   |
+------------+--------------+---------+
```

1 row in set (0.002 sec)

```
MariaDB [dbms4]> SELECT tp.Tourist_place_id, tp.Name, tp.State
    -> FROM TOURIST_PLACE tp
    -> WHERE NOT EXISTS (
    ->     SELECT DISTINCT t.Country
    ->     FROM TOURIST t
    ->     EXCEPT
    ->     SELECT DISTINCT t2.Country
    ->     FROM VISIT v
    ->     JOIN TOURIST t2 ON v.Tourist_id = t2.Tourist_id
    ->     WHERE v.Tourist_place_id = tp.Tourist_place_id
    -> );
+-----------------+-----------+---------------+
| Tourist_place_id | Name      | State         |
+-----------------+-----------+---------------+
|               5 | Taj Mahal | UTTAR PRADESH |
+-----------------+-----------+---------------+
1 row in set (0.001 sec)
```