

Data Base Management System Lab Manual[22MCAL27]



Prepared By,

Roopa R

Assistant Professor

Department of MCA

BIET

Bapuji Educational Association (Regd.)

Bapuji Institute of Engineering and Technology, Davangere

Department of Master of Computer Applications (MCA)

1. Create the following tables with properly specifying primary keys, Foreign keys and solve following queries:

BRANCH (Branchid,Branchname,Hod)

STUDENT (USN,Name,Address,Branchid,sem)

Author (Authorid,Authorname,Country,Age)

Book(Bookid,Book_name,Authorid,Publisher,Branchid)

BORROW (USN,Boookid, Borrowed_Date)

Execute the following queries:

- i) List the details of student who are all studying in 2ns MCA.
- ii) List the students who are not borrowed any books
- iii) Display the USN, Student name, Branch_name, book_name, author_name, books_borrowed_date of 2nd sem MCA students who borrowed books.
- iv) Display the number of books written by each Author.
- v) Display the student details who borrowed more than two books.
- vi) Display the student details who borrowed books of more than one Author.
- vii) Display the Book names in descending order of their names.
- viii) List the details of students who borrowed the books which are all published by the same Publisher.

Create Table:

Create table Branch

(Branchid int primary key,

Bname varchar(10),

HOD varchar(10));

Create table Student

(USN varchar(10) primary key,

Name varchar(10),

Address varchar(15),

branchid int references branch(branchid),

Sem int);

Create table Author

```
( Authorid int primary key,  
  Aname varchar(10),  
  Country varchar(10),  
  Age int(2));
```

Create table Book

```
( bookid varchar(10),  
  bname varchar(10),  
  Authorid int references author(authorid),  
  Publisher varchar(10),  
  branchid int references branch(branchid));
```

Create table Borrow

```
( USN varchar(10) references student(usn),  
  Bookid int references book(bookid) ,  
  Borrowed_date date);
```

Insert values:

```
Insert into branch value (1,'MCA', ShankarGowda B.B');
```

```
//similarly insert other records.
```

```
mysql> insert into branch value(  
  -> 1,'MCA', 'ShankarGowda B.B');  
Query OK, 1 row affected, 1 warning (0.03 sec)  
  
mysql> insert into branch value(  
  -> 2,'MBA', 'Ueeresh');  
Query OK, 1 row affected (0.03 sec)  
  
mysql> insert into branch value(  
  -> 3,'CSE', 'Nirmala C.R');  
Query OK, 1 row affected, 1 warning (0.03 sec)  
  
mysql>  
mysql>  
mysql> insert into branch value(  
  -> 4,'ISE', 'Vinutha');  
Query OK, 1 row affected (0.03 sec)  
  
mysql> insert into branch value(  
  -> 5,'E&E', 'Dheeresh');  
Query OK, 1 row affected (0.03 sec)
```

```
mysql> select * from branch;
+-----+-----+-----+
| Branchid | Bname | HOD      |
+-----+-----+-----+
| 1        | MCA   | ShankarGow |
| 2        | MBA   | Ueeresh   |
| 3        | CSE   | Nirmala C. |
| 4        | ISE   | Vinutha   |
| 5        | E&E   | Dheeresh  |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> insert into student value(
-> '22MCA01', 'Rajesh', 'Davangere',1,2);
Query OK, 1 row affected (0.03 sec)

mysql> insert into student value(
-> '22MCA02', 'Rakesh', 'Davangere',2,3);
Query OK, 1 row affected (0.03 sec)

mysql> insert into student value(
-> '22MCA03', 'Ramesh', 'Davangere',3,6);
Query OK, 1 row affected (0.02 sec)

mysql> insert into student value(
-> '22MCA04', 'Ramya', 'Davangere',4,7);
Query OK, 1 row affected (0.16 sec)

mysql>
mysql> insert into student value(
-> '22MCA05', 'Pooja', 'Davangere',5,4);
Query OK, 1 row affected (0.03 sec)
```

```
mysql> select * from student;
+-----+-----+-----+-----+-----+
| USN      | Name   | Address   | branchid | Sem    |
+-----+-----+-----+-----+-----+
| 22MCA01  | Rajesh | Davangere | 1        | 2      |
| 22MCA02  | Rakesh | Davangere | 2        | 3      |
| 22MCA03  | Ramesh | Davangere | 3        | 6      |
| 22MCA04  | Ramya  | Davangere | 4        | 7      |
| 22MCA05  | Pooja  | Davangere | 5        | 4      |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> insert into book value(
-> 1111,'C prog',123,'pearson',1);
Query OK, 1 row affected (0.03 sec)

mysql> insert into book value(
-> 2222,'dbms',124,'himalaya',2);
Query OK, 1 row affected (0.03 sec)

mysql> insert into book value(
-> 3333,'oops',125,'sapna',3);
Query OK, 1 row affected (0.03 sec)

mysql> insert into book value(
-> 4444,'unix',126,'subhash',4);
Query OK, 1 row affected (0.05 sec)

mysql> insert into book value(
-> 5555,'CN',127,'pearson',5);
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from book;
```

bookid	bname	Authorid	Publisher	branchid
1111	C prog	123	pearson	1
2222	dbms	124	himalaya	2
3333	oops	125	sapna	3
4444	unix	126	subhash	4
5555	CN	127	pearson	5
222	dbms	123	pearson	1

```
6 rows in set (0.00 sec)
```

```
mysql> insert into author value(
-> 123,'Kottur','India',55);
Query OK, 1 row affected (0.02 sec)

mysql> insert into author value(
-> 124,'navathe','India',45);
Query OK, 1 row affected (0.03 sec)

mysql> insert into author value(
-> 125,'riche','uk',50);
Query OK, 1 row affected (0.06 sec)

mysql> insert into author value(
-> 126,'sumitaba','india',55);
Query OK, 1 row affected (0.03 sec)

mysql> insert into author value(
-> 12,'sumitaba','india',55);
Query OK, 1 row affected (0.03 sec)
```

```
mysql> select * from author;
```

Authorid	Aname	Country	Age
123	Kottur	India	55
124	navathe	India	45
125	riche	uk	50
126	sumitaba	india	55
127	sumitaba	india	55

```
5 rows in set (0.00 sec)
```

```
mysql> insert into borrow values(
-> '22MCA01', 2222, '2023-01-22'),
-> ('22MCA01', 3333, '2023-04-21'),
-> ('22MCA03', 5555, '2023-03-22'),
-> ('22MCA05', 2222, '2023-01-20'),
-> ('22MCA02', 1111, '2023-09-18');
Query OK, 5 rows affected (0.03 sec)
```

```
mysql> select * from borrow;
+-----+-----+-----+
| USN      | Bookid | Borrowed_date |
+-----+-----+-----+
| 22MCA01  | 2222   | 2023-01-22    |
| 22MCA01  | 3333   | 2023-04-21    |
| 22MCA03  | 5555   | 2023-03-22    |
| 22MCA05  | 2222   | 2023-01-20    |
| 22MCA02  | 1111   | 2023-09-18    |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Query 1 : List the details of Student who are all studying in 2nd sem MCA

```
mysql> select * from student where sem = 2 and branchid in
-> (select branchid from branch where bname = 'MCA');
+-----+-----+-----+-----+-----+
| USN      | Name   | Address   | branchid | Sem |
+-----+-----+-----+-----+-----+
| 22MCA01  | Rajesh | Davangere | 1        | 2   |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Query 2: List the students who are not borrowed any books

```
mysql> select * from student where USN not in
-> (select USN from borrow);
+-----+-----+-----+-----+-----+
| USN      | Name   | Address   | branchid | Sem |
+-----+-----+-----+-----+-----+
| 22MCA04  | Ranya  | Davangere | 4        | 7   |
+-----+-----+-----+-----+-----+
1 row in set (0.03 sec)
```

Query 3: Display the USN, Student name, Branch_name, book_name, author_name, books_borrowed_date of 2nd sem MCA students who borrowed books

```
mysql> select student.usn,student.name,branch.bname,book.bname,aname,
-> borrowed_date from student,branch,book,author,borrow where
-> student.usn=borrow.usn and borrow.bookid = book.bookid and
-> book.authorid = author.authorid and student.branchid =branch.branchid
-> and student.sem = 2 and branch.bname = 'MCA';
+-----+-----+-----+-----+-----+-----+
| usn      | name   | bname | bname | aname | borrowed_date |
+-----+-----+-----+-----+-----+-----+
| 22MCA01  | Rajesh | MCA   | dbms  | navathe | 2023-01-22    |
| 22MCA01  | Rajesh | MCA   | oops  | riche  | 2023-04-21    |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Query 4: Display the number of books written by each Author.

```
mysql> select count(*) ,authorid from book group by authorid;
+-----+-----+
| count(*) | authorid |
+-----+-----+
| 1        | 123      |
| 1        | 124      |
| 1        | 125      |
| 1        | 126      |
| 1        | 127      |
+-----+-----+
5 rows in set (0.00 sec)
```

Query 5: Display the student details who borrowed more than two books.

```
mysql> select name,address,branchid,sem,count(borrow.usn)
-> from student,borrow where borrow.usn=student.usn
-> group by name,address,branchid,sem having count(borrow.usn)>=2;
+-----+-----+-----+-----+-----+
| name   | address | branchid | sem | count(borrow.usn) |
+-----+-----+-----+-----+-----+
| Rajesh | Davangere | 1        | 2   | 2                 |
+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

Query 6: Display the student details who borrowed books of more than one Author.

```
mysql> select * from student where usn in (select usn from borrow group by usn
-> having count(usn)>=2);
+-----+-----+-----+-----+-----+
| USN    | Name   | Address | branchid | Sem |
+-----+-----+-----+-----+-----+
| 22MCA01 | Rajesh | Davangere | 1        | 2   |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Query 7: Display the Book names in descending order of their names.

```
mysql> select bname from book order by bname desc;
+-----+
| bname |
+-----+
| unix  |
| oops  |
| dbms  |
| dbms  |
| CN    |
| C prog |
+-----+
6 rows in set (0.00 sec)
```

Query 8: List the details of students who borrowed the books which are all published by the same Publisher.

```
mysql> select * from student s where exists(select usn,publisher from borrow
-> join book on borrow.bookid = book.bookid where s.usn = borrow.usn
-> group by usn having count(distinct publisher) =1);
+-----+-----+-----+-----+-----+
| USN    | Name   | Address | branchid | Sem |
+-----+-----+-----+-----+-----+
| 22MCA01 | Rajesh | Davangere | 1        | 2   |
+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

2. Create the following schema.**STUDENT(USN,name,date_of_birth,branch,mark1,mark2,mark3,total,GPA)**

Execute the following queires:

- i) Update the column total by adding the column mark1,mark2,mark3.
- ii) Find the GPA score of all the students.
- iii) Find the students who born on a particular year of birth from the date _of_birth column.
- iv) List the students who are studying in a particular branch of study.
- v) Find the maximum GPA score of the student branch-wise.
- vi) Find the students whose name starts with alphabet "S".
- vii) Find the students whose name ends with alphapets "AR".
- viii) Delete the student details whose USN is given as 1001.

Create Table:

```
create table StudentR(  
    USN varchar(10) primary key,  
    Snamevarchar(20),  
    Date_of_Birth Date,  
    Branch varchar(10),  
    Mark1 int,  
    Mark2 int,  
    Mark3 int,  
    Total int,  
    GPA int);
```

Insert Values:

```
insert into StudentR value(  
'4BD22MC001', Abhishek,'1999/07/02','MCA', 91, 82, 86 ,0,0);  
  
// similarly insert other records
```



```
mysql> select * from StudentR;
+-----+-----+-----+-----+-----+-----+-----+-----+
| USN    | Sname  | Date_of_Birth | Branch | Mark1 | Mark2 | Mark3 | Total |
| GPA    |        |                |        |       |       |       |       |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 4BD22MC001 | Abhishek | 1999-07-02 | MCA | 91 | 82 | 86 | NULL |
| NULL |        |                |        |       |       |       |       |
| 4BD22MC002 | Ajay S   | 1998-03-02 | MCA | 71 | 82 | 76 | 0 |
| 0 |        |                |        |       |       |       |       |
| 4BD22MC003 | Anjan    | 1999-06-12 | MCA | 81 | 92 | 76 | 0 |
| 0 |        |                |        |       |       |       |       |
| 4BD22MC007 | Chandan  | 1998-03-02 | MCA | 71 | 82 | 76 | 0 |
| 0 |        |                |        |       |       |       |       |
| 4BD22MC014 | Naveen   | 1999-02-02 | MCA | 81 | 82 | 83 | 0 |
| 0 |        |                |        |       |       |       |       |
| 4BD22MC015 | Rakesh   | 1999-07-02 | MCA | 91 | 81 | 76 | 0 |
| 0 |        |                |        |       |       |       |       |
| 4BD22MC018 | Prasad   | 1999-07-02 | MBA | 71 | 82 | 76 | 0 |
| 0 |        |                |        |       |       |       |       |
| 4BD22MC020 | Chethan  | 2000-03-02 | MCA | 91 | 72 | 76 | 0 |
| 0 |        |                |        |       |       |       |       |
| 4BD22MC024 | Srinivas | 1998-07-02 | CS  | 71 | 72 | 76 | 0 |
| 0 |        |                |        |       |       |       |       |
+-----+-----+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

i) Update the column total by adding the columns mark1,mark2,mark3

Solution:

Update StudentR set total = mark1 + mark2 + mark3;

Update StudentR set GPA = total / 3;

```
mysql> select * from StudentR;
+-----+-----+-----+-----+-----+-----+-----+-----+
| USN    | Sname  | Date_of_Birth | Branch | Mark1 | Mark2 | Mark3 | Total |
| GPA    |        |                |        |       |       |       |       |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 4BD22MC001 | Abhishek | 1999-07-02 | MCA | 91 | 82 | 86 | 259 |
| 86 |        |                |        |       |       |       |       |
| 4BD22MC002 | Ajay S   | 1998-03-02 | MCA | 71 | 82 | 76 | 229 |
| 76 |        |                |        |       |       |       |       |
| 4BD22MC003 | Anjan    | 1999-06-12 | MCA | 81 | 92 | 76 | 249 |
| 83 |        |                |        |       |       |       |       |
| 4BD22MC007 | Chandan  | 1998-03-02 | MCA | 71 | 82 | 76 | 229 |
| 76 |        |                |        |       |       |       |       |
| 4BD22MC014 | Naveen   | 1999-02-02 | MCA | 81 | 82 | 83 | 246 |
| 82 |        |                |        |       |       |       |       |
| 4BD22MC015 | Rakesh   | 1999-07-02 | MCA | 91 | 81 | 76 | 248 |
| 83 |        |                |        |       |       |       |       |
| 4BD22MC018 | Prasad   | 1999-07-02 | MBA | 71 | 82 | 76 | 229 |
| 76 |        |                |        |       |       |       |       |
| 4BD22MC020 | Chethan  | 2000-03-02 | MCA | 91 | 72 | 76 | 239 |
| 80 |        |                |        |       |       |       |       |
| 4BD22MC024 | Srinivas | 1998-07-02 | CS  | 71 | 72 | 76 | 219 |
| 73 |        |                |        |       |       |       |       |
+-----+-----+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

ii) Find the GPA score of all the students:

Solution:

Select sname,GPA from StudentR;

```
mysql> select sname,GPA from StudentR;
+-----+-----+
| sname | GPA |
+-----+-----+
| Abhishek | 86 |
| Ajay S | 76 |
| Anjan | 83 |
| Chandan | 76 |
| Naveen | 82 |
| Rakesh | 83 |
| Prasad | 76 |
| Chethan | 80 |
| Srinivas | 73 |
+-----+-----+
9 rows in set (0.00 sec)
```

iii) Find the students who born on particular year of birth from date_of_birth column.

Solution:

Select * from student where date_of_birth like '1998%';

```
mysql> select * from studentR where date_of_birth like '1998%';
+-----+-----+-----+-----+-----+-----+-----+-----+
| USN | Sname | Date_of_Birth | Branch | Mark1 | Mark2 | Mark3 | Total |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 4BD22MC002 | Ajay S | 1998-03-02 | MCA | 71 | 82 | 76 | 229 |
| 4BD22MC007 | Chandan | 1998-03-02 | MCA | 71 | 82 | 76 | 229 |
| 4BD22MC024 | Srinivas | 1998-07-02 | CS | 71 | 72 | 76 | 219 |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)
```

iv)List the students who are studying in particular branch of study.

Solution:

Select * from StudentR where branch = 'CS';

```
mysql> select * from studentR where date_of_birth like '1998%';
+-----+-----+-----+-----+-----+-----+-----+-----+
| USN | Sname | Date_of_Birth | Branch | Mark1 | Mark2 | Mark3 | Total |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 4BD22MC002 | Ajay S | 1998-03-02 | MCA | 71 | 82 | 76 | 229 |
| 4BD22MC007 | Chandan | 1998-03-02 | MCA | 71 | 82 | 76 | 229 |
| 4BD22MC024 | Srinivas | 1998-07-02 | CS | 71 | 72 | 76 | 219 |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)
```

v) Find the maximum GPA score of the student branch-wise;

Solution:

Select branch, max(GPA) from student group by branch;

```
mysql> select branch,max(GPA) from studentR group by branch;
+-----+-----+
| branch | max(GPA) |
+-----+-----+
| CS     | 73       |
| MBA    | 76       |
| MCA    | 86       |
+-----+-----+
3 rows in set (0.00 sec)
```

vi) Find the students whose name starts with the alphabet "S".

Solution:

Select * from StudentR where sname like 'S%';

```
mysql> select * from studentR where sname like 's%';
+-----+-----+-----+-----+-----+-----+-----+-----+
| USN      | Sname   | Date_of_Birth | Branch | Mark1 | Mark2 | Mark3 | Total |
| GPA     |         |               |        |       |       |       |       |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 4BD22MC024 | Srinivas | 1998-07-02   | CS     | 71    | 72    | 76    | 219   |
| 73       |         |               |        |       |       |       |       |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

vii) Find the students whose name ends with the alphabes 'AN'.

Solution:

Select * from StudentR where sname like '%AN';

```
mysql> select * from studentR
-> where sname like '%AN';
+-----+-----+-----+-----+-----+-----+-----+-----+
| USN      | Sname   | Date_of_Birth | Branch | Mark1 | Mark2 | Mark3 | Total |
| GPA     |         |               |        |       |       |       |       |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 4BD22MC003 | Anjan   | 1999-06-12   | MCA    | 81    | 92    | 76    | 249   |
| 83       |         |               |        |       |       |       |       |
| 4BD22MC007 | Chandan | 1998-03-02   | MCA    | 71    | 82    | 76    | 229   |
| 76       |         |               |        |       |       |       |       |
| 4BD22MC020 | Chethan | 2000-03-02   | MCA    | 91    | 72    | 76    | 239   |
| 80       |         |               |        |       |       |       |       |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

viii) Delete the student details whose USN is given as 001.

Solution:

Delete FROM student where USN = '4BD22MC001';

```
mysql> delete FROM studentR where USN = '4BD22MC001' ;  
Query OK, 1 row affected (0.08 sec)
```

```
mysql> SELECT * FROM studentR;
```

USN GPA	Sname	Date_of_Birth	Branch	Mark1	Mark2	Mark3	Total
4BD22MC002 76	Ajay S	1998-03-02	MCA	71	82	76	229
4BD22MC003 83	Anjan	1999-06-12	MCA	81	92	76	249
4BD22MC007 76	Chandan	1998-03-02	MCA	71	82	76	229
4BD22MC014 82	Naveen	1999-02-02	MCA	81	82	83	246
4BD22MC015 83	Rakesh	1999-07-02	MCA	91	81	76	248
4BD22MC018 76	Prasad	1999-07-02	MBA	71	82	76	229
4BD22MC020 80	Chethan	2000-03-02	MCA	91	72	76	239
4BD22MC024 73	Srinivas	1998-07-02	CS	71	72	76	219

3. Design 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 there are many teams are contesting each having a Teamid, Team_name, City, 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, Adress(involves city, area_name, pincode). A team can play many matches. Each Match played between the two teams in 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.

- i) Display the youngest player (in terms of age) Name, Team name, age in which he belongs of the tournament.
- ii) List the details of the stadium where the maximum number of matches were played.
- iii) List the details of the player who is not a captain but got the man_of_match award at least in two matches.
- iv) Display the Team details who won the maximum matches.
- v) Display the team name where all its won matches played in the stadium.

Create Table:

```
create table TeamR(
    Tid    varchar(10) primary key,
    Tname  varchar(10),
    coach  varchar(10),
    capid  varchar(4),
    city   varchar(10));
```

Insert values:

```
insert into teamR values('T001', 'India','Clark','PA01','Mumbai');
```

// similarly add values

```
mysql> select * from TeamR;
+----+-----+-----+-----+-----+
| Tid | Tname | coach | capid | city |
+----+-----+-----+-----+-----+
| T001 | India | Clark | PA01 | Mumbai |
| T002 | Pakistan | Mulla | PB01 | Gulf |
| T003 | Africa | Smith | PC01 | Gerf |
| T004 | Australia | Sammy | PD01 | Denmark |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Create table PlayerR(

```
    Pid    varchar(4) primary key,
    pname  varchar(10) Not Null,
    age    int(2),
    Tid    varchar(10) references TeamR(tid));
```

Insert values:

```
insert into playerR values('PA01','KHOLI','25','T001');
```

//Similarly add values

```
mysql> SELECT * FROM PLAYERR;
+----+-----+-----+-----+
| Pid | pname | age | Tid |
+----+-----+-----+-----+
| PA01 | KHOLI | 25 | T001 |
| PA02 | DHONI | 7 | T001 |
| PA03 | INSHANT | 32 | T001 |
| PB01 | SMARK | 30 | T002 |
| PB02 | SMITH | 25 | T002 |
| PB03 | SHAYNE | 35 | T002 |
| PC01 | SAMM | 35 | T003 |
| PC02 | WANDER | 33 | T003 |
| PC03 | UALKY | 34 | T003 |
| PD01 | UALKY | 24 | T004 |
| PD02 | BAHU | 21 | T004 |
| PD03 | FUGU | 28 | T004 |
| PD04 | RUGU | 82 | T004 |
+----+-----+-----+-----+
13 rows in set (0.00 sec)
```

```

create table stadiumR(
    sid      varchar(4) primary key,
    sname    varchar(10) Not null,
    pincode  int(6),
    city     varchar(10),
    area     varchar(10));

insert into stadiumR values('S001','INDIA','509123','MUMBAI','SAHPUR');
// similarly add some more values.

```

```

mysql> Select * from stadiumR;
+----+-----+-----+-----+-----+
| sid | sname | pincode | city   | area   |
+----+-----+-----+-----+-----+
| S001 | INDIA | 509123 | MUMBAI | SAHPUR |
| S002 | AFRICA | 509124 | DENMARK | OCEAN  |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

```

create table matchesR(
    mid      varchar(4) primary key,
    mdate    date,
    time     varchar(10),
    sid      varchar(4) references StadiumR(Sid),
    team_id1 varchar(4) references TeamR(tid),
    team_id2 varchar(4) references TeamR(tid),
    winning_team_id varchar(4) references TeamR(tid),
    man_of_match varchar(4) references playerR(pid));

insert into matchesR values (M1,'2022/07/02','10AM','S001','T001','T002','T001','PA01');
// Similarly add some more values

```

```

mysql> SELECT * FROM MATCHESr;
+----+-----+-----+-----+-----+-----+-----+
| mid | mdate   | time | sid | team_id1 | team_id2 | winning_team_id | man_o
f_match |
+----+-----+-----+-----+-----+-----+-----+
| M1  | 2022-07-02 | 10AM | S001 | T001     | T002     | T001             | PA01
| M2  | 2021-06-02 | 3AM  | S002 | T003     | T004     | T003             | PC01
| M3  | 2021-06-10 | 9AM  | S001 | T001     | T003     | T001             | PA02
| M4  | 2023-06-10 | 11AM | S002 | T001     | T004     | T001             | PA02
| M5  | 2022-05-10 | 11AM | S001 | T001     | T002     | T001             | PA02
+----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

create table playerR_pno(
pid varchar(4) references playerR(pid),
phone int(10));
insert into player values('PA01',9845234212);

```

```

mysql> SELECT * FROM PLAYERR_PNO;
+----+-----+
| pid | phone |
+----+-----+
| PA01 | 2147483647 |
| PA02 | 2147483647 |
| PA03 | 2147483647 |
| PA01 | 2147483647 |
| PA02 | 2147483647 |
| PB01 | 2147483647 |
| PB01 | 2147483647 |
| PC01 | 2147483647 |
+----+-----+
8 rows in set (0.00 sec)

```

- i) Display the youngest player (in terms of age) Name, Team name, age in which he belongs of the tournament.

Solution:

```

select pname,tname,age from player P,team t where
p.tid = t.tid and age = (select min(age) from player);

```

```

mysql> select pname,tname,age from player P,team t where
-> p.tid = t.tid and age = (select min(age) from player);
+----+-----+-----+
| pname | tname | age |
+----+-----+-----+
| BAHU | AUSTRALIA | 21 |
+----+-----+-----+
1 row in set (0.03 sec)

```

- ii) List the details of the stadium where the maximum number of matches were played.

Solution:

```

select * from stadiumR where sid in (select sid from matchesR
group by sid having count(sid) = (select count(sid) from matchesR group by sid order by sid asc limit 1 ));

```

```

mysql> select * from stadiumR where sid in (select sid from matchesR
-> group by sid having count(sid) = (select count(sid) from matchesR group b
y sid order by sid asc limit 1 ));
+----+-----+-----+-----+-----+
| sid | sname | pincode | city | area |
+----+-----+-----+-----+-----+
| S001 | INDIA | 509123 | MUMBAI | SAHPUR |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

- iii) List the details of the player who is not a captain but got the man_of_match award at least in two matches.

Solution:

```
select * from playerR where pid not in (select capid from team)
and pid in(select man_of_match from matchesR group by
man_of_match having count(man_of_match)>=2);
```

```
mysql> select * from playerR where pid not in (select capid from team)
-> and pid in(select man_of_match from matchesR group by
-> man_of_match having count(man_of_match)>=2);
+----+-----+-----+-----+
| Pid | pname | age | Tid |
+----+-----+-----+-----+
| PA02 | DHONI | 7 | T001 |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

vi) Display the Team details who won the maximum matches.

Solution:

```
select * from teamR where tid in (select winning_team_id from matchesR group by winning_team_id
having count(winning_team_id) = (select count(winning_team_id) from
matchesR group by winning_team_id order by winning_team_id desc limit 1));
```

```
;
+----+-----+-----+-----+-----+
| Tid | Tname | coach | capid | city |
+----+-----+-----+-----+-----+
| T003 | Africa | Smith | PC01 | Gerf |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

iv) Display the team name where all its won matches played in the stadium.

Solution:

```
select tname from teamR where tid in( select winning_team_id from matchesR group
by(winning_team_id) in (select count(winning_team_id) from matchesR group by
winning_team_id order by winning_team_id));
```

```
+----+-----+
| tname |
+----+-----+
| India |
+----+-----+
1 row in set (0.00 sec)
```

4. A country wants to conduct an election for the parliament. A country having 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 House no, city, state, pincode). Each voter belongs to only one constituency. There are many candidates contesting in the election. Each candidates are 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 a 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 belongs to different parties. Each voter votes only one candidate of his/her constituency.

Queries:

- i. List the details of the candidates who are contesting from more than one constituencies which are belongs to different states.**
- ii. Display the state name having maximum number of constituencies.**
- iii. Create a stored procedure to insert the tuple into the voter table by checking the voter age. If voter's age is at least 18 years old, then insert the tuple into the voter else display the "Not an eligible voter msg".**
- iv. 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.**
- v. Create a TRIGGER to UPDATE the count of " Number_of_voters" of the respective constituency in "CONSTITUENCY" table , AFTER inserting a tuple into the "VOTERS" table.**

Create Table:

```
create table constituency
```

```
(cons_id int(20) primary key,
```

```
csname varchar(20),
```

```
csstate varchar(20),
```

```
no_of_voters int(10));
```

```
insert into constituency values(111,'rajajinagar','karnataka',4);
```

```
insert into constituency values(222,'ramnagar','kerala',1);
```

```
insert into constituency values(333,'Chittur','Andrapradesh',1);
```

```
mysql> select * from constituency;
```

cons_id	csname	csstate	no_of_voters
111	rajajinagar	karnataka	4
222	ramnagar	kerala	1
333	Chittur	Andrapradesh	1

```
create table party
```

```
(pid int(20) primary key,
```

```
pname varchar(20),
```

```
psymbol varchar(10));
```

```
insert into party values(876,'bjp','lotus');
```

```
insert into party values(877,'congress','hand');
```

```
mysql> select * from party;
```

pid	pname	psymbol
876	bjp	lotus
877	congress	hand

```
2 rows in set (0.00 sec)
```

```
create table candidates
```

```
(cand_id int(12) primary key,
```

```
phone_no int(10),
```

```
age int(2),
```

```
state varchar(20),
```

```
name varchar(20),
```

```
pid int references party(pid));
```

```
insert into candidates values(121,9538904626,23,'kerala','raksha',876);
```

```
insert into candidates values(122,9740777502,24,'karnataka','veena',877);
```

```
select *from candidates;
```

```
mysql> select *from candidates;
+-----+-----+-----+-----+-----+-----+
| cand_id | phone_no | age | state | name | pid |
+-----+-----+-----+-----+-----+-----+
|      121 | 2147483647 | 23 | kerala | raksha | 876 |
|      122 | 2147483647 | 24 | karnataka | veena | 877 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
create table contest
```

```
(cons_id int(20) references constituency(cons_id),
```

```
cand_id int(12) references candidates(cons_id),
```

```
primary key(cons_id,cand_id));
```

```
insert into contest values(121,122);
```

```
insert into contest values(222,121);
```

```
insert into contest values(222,122);
```

```
insert into contest values(122,122);
```

```
select * from contest;
```

```
mysql> select * from contest;
+-----+-----+
| cons_id | cand_id |
+-----+-----+
|      121 |      122 |
|      122 |      122 |
|      222 |      121 |
|      222 |      122 |
+-----+-----+
4 rows in set (0.00 sec)
```

```

create table voter
(vid int(20) primary key,
vname varchar(20),
vage int(5),
vaddr varchar(20),
cons_id int(20) references constituency(cons_id),
cand_id int(12) references candidates(cand_id));

insert into voter values(345,'prashanth',21,'kanakpur',222,122);
insert into voter values(346,'prakash',23,'ramnagar',111,121);
insert into voter values(348,'nagesh',30,'mandya',111,121);
insert into voter values(349,'nagesh',30,'mandya',111,121);

select * from voter;

```

```

mysql> select * from voter;
+----+-----+-----+-----+-----+-----+
| vid | vname   | vage | vaddr   | cons_id | cand_id |
+----+-----+-----+-----+-----+-----+
| 345 | prashanth | 21 | kanakpur | 222 | 122 |
| 346 | prakash  | 23 | ramnagar | 111 | 121 |
| 348 | nagesh   | 30 | mandya   | 111 | 121 |
| 349 | nagesh   | 30 | mandya   | 111 | 121 |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

Queries:

- List the details of the candidates who are contesting from more than one constituencies
Which are belongs to different states.**

```

select * from candidates where cand_id in (select cand_id from contest
join constituency on contest.cons_id=constituency.cons_id
group by cand_id having count(distinct(csstate))>=1);

```

```
mysql> select * from candidates where cand_id in (select cand_id from contest
-> join constituency on contest.cons_id=constituency.cons_id
-> group by cand_id having count(distinct(csstate))>=1);
```

cand_id	phone_no	age	state	name	pid
121	2147483647	23	kerala	raksha	876
122	2147483647	24	karnataka	veena	877

```
2 rows in set (0.00 sec)
```

2. Display the state name having maximum number of constituencies .

```
select csstate from (select csstate, count(*) As num_constituency from constituency
group by csstate order by num_constituency desc limit 1) max_constituency;
```

```
mysql> select csstate from (select csstate, count(*) As num_constituency from constituency
-> group by csstate order by num_constituency desc limit 1) max_constituency;
```

csstate
karnataka

```
1 row in set (0.00 sec)
```

3. Create a stored procedure to insert the tuple into the voter table by checking the voter age. If Voters age is atleast 18 years old, then insert the tuple into the voter else display the “ Not an eligible voter msg”.

```
DELIMITER //
```

```
Create Procedure ageChk( In vid varchar(10), In vname varchar(20),In vage int,
```

```
In vaddr varchar(30), in vcand_id varchar(10))
```

```
BEGIN
```

```
declare msg varchar(50);
```

```
if vage >= 18 then
```

```
insert into voter values(vid,vname,vage,vaddr,vcand_id);
```

```
set msg = "Row Inserted....";
```

```
else
```

```
set msg = "Age less than 18";
```

```
end if;
```

```
select msg;
```

```
End;
```

```
//
```

```
mysql> call agechk('v07','Avi',14,'Davangere','co1');
-> //
+-----+
| msg   |
+-----+
| Age less than 18 |
+-----+
1 row in set (0.01 sec)
```

```
mysql> select * from voter;
-> //
+-----+-----+-----+-----+-----+
| voter_id | name      | age  | address      | cid  |
+-----+-----+-----+-----+-----+
| v01      | akhil     | 24   | kasaragod    | c01  |
| v02      | hari      | 26   | kasaragod    | c01  |
| v03      | meena     | 52   | kasaragod    | c01  |
| v04      | arun      | 18   | manjeshwara  | c03  |
| v05      | dhruv     | 19   | agrar        | c02  |
| v06      | akhila    | 32   | manjeshwara  | c03  |
| v07      | Avi       | 34   | agrar        | c02  |
| v10      | Natraj    | 30   | Kasaragod    | c03  |
| v11      | Raja      | 33   | agrar        | c02  |
| v12      | Vaishali  | 45   | Poona        | c02  |
| v13      | Amnju     | 43   | Mum          | c02  |
| v22      | Rajan     | 34   | agrar        | c02  |
| v30      | Avi       | 50   | agrar        | c02  |
+-----+-----+-----+-----+-----+
13 rows in set (0.00 sec)
```

- v) 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.


```
mysql>
mysql>
mysql>
mysql> delimiter //
mysql> CREATE PROCEDURE consName(conname1 varchar(50))
-> BEGIN
-> select no_of_voters from
-> constituency where name=conname1;
-> END //
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> Delimiter ;
mysql> call consName('agrara');
+-----+
| no_of_voters |
+-----+
| 1056790      |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

- vi) Create a TRIGGER to UPDATE the count of “Number_of_voters” of the respective constituency in “CONSTITUENCY” table , AFTER inserting a tuple into the “VOTERS” table.

```
C:\WINDOWS\system32\cmd.exe

mysql>
mysql>
mysql>
mysql> delimiter //
mysql> create trigger updT after
-> insert on voter for each row
-> begin
-> update constituency set
-> no_of_voters=no_of_voters+1 where cid=new.cid;
-> end //
Query OK, 0 rows affected (0.08 sec)

mysql> delimiter ;
mysql> insert into voter values('v25','Nagaraj',35,'Kasaragod','c01')
Query OK, 1 row affected (0.02 sec)

mysql> select * from voter;
+-----+-----+-----+-----+-----+
| voter_id | name   | age  | address  | cid   |
+-----+-----+-----+-----+-----+
| v01      | akhil  | 24   | kasaragod | c01   |
| v02      | hari   | 26   | kasaragod | c01   |
```


- 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, belongs to a state, Number of kilometres away from the 02.03.2021 updated 52/ 104 capital city of that state,history. There are many Tourists visits tourist places every year. Each tourist is identified uniquely by using Tourist_id, having a Name, age, Country and multiple emailids. A tourist visits many Tourist places, it is also required to record the visted_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 in the same date or at different dates.

Queries:

- i. List the state name which is having maximum number of tourist places.
- ii. List details of Tourist place where maximum number of tourists visited.
- iii. List the details of tourists visited all tourist places of the state "KARNATAKA".
- iv. Display the details of the tourists visited at least one tourist place of the state, but visited all states tourist places.
- v. Display the details of the tourist place visited by the tourists of all country.

Create Table:

```
create table Tourist_place01(  
Tpid int primary key,  
TpName varchar(20),  
State varchar(20),  
Kilometer int(3),  
History varchar(20));
```

```
insert into TOURIST_place01 values('11','tajmhal','delhi','100','monument');  
insert into TOURIST_place01 values('12','munnar','kerala','200','hillstationb');  
insert into TOURIST_place01 values('13','modren','goa','50','beach');
```

```
mysql> select * from tourist_place01;  
+-----+-----+-----+-----+-----+  
| Tpid | TpName | State | Kilometer | History |  
+-----+-----+-----+-----+-----+  
| 11 | tajmhal | delhi | 100 | monument |  
| 12 | munnar | kerala | 200 | hillstationb |  
| 13 | modren | goa | 50 | beach |  
+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

```
create table Tourist01(  
Tid int primary key,  
Name varchar(20),  
Age int(3),  
Country varchar(20));
```

```
insert into Tourist01 values(22,'PRAKESH','40','DUBAI');  
insert into Tourist01 values(23,'ABDHUL','35','INDIA');  
insert into Tourist01 values(24,'ARNOLD','45','AMERICA');
```

```
mysql> select * from Tourist01;
```

Tid	Name	Age	Country
22	PRAKESH	40	DUBAI
23	ABDHUL	35	INDIA
24	ARNOLD	45	AMERICA

```
3 rows in set (0.00 sec)
```

```
create table Visits01(
  Tpid int References Tourist_place01(Tpid),
  Tid int References Tourist01(Tid),
  V_date date,
  primary key (Tpid,Tid));
```

```
insert into Visits01 values(12,23,'2014/11/13');
```

```
mysql> select * from visits01;
```

Tpid	Tid	V_date
11	23	2011-08-13
11	24	2015-10-13
12	23	2014-11-13
13	22	2011-10-13
13	23	2010-08-13
14	24	2017-05-10

```
6 rows in set (0.00 sec)
```

```
create table email01(
  Tid int References Tourist01(Tid),
  email varchar(30),
  primary key (tid,email));
```

```
insert into email01 values(23,'bhanu@gmal.com');
insert into email01 values(22,'manu@gmal.com');
insert into email01 values(25,'thanu@gmal.com');
```

```
mysql>
mysql> select * from email01;
+-----+
| Tid | email |
+-----+
| 22 | manu@gmail.com |
| 23 | bhanu@gmail.com |
| 25 | thanu@gmail.com |
+-----+
3 rows in set (0.00 sec)
```

Query:

1. List the state name which is have a maximum number of tourist places.

Select state, count(tpid) from tourist_place01 group by state order by count(tpid)
Desc limit 1;

```
mysql> Select state, count(tpid) from tourist_place01 group by state order by count(tpid)
-> Desc limit 1;
+-----+-----+
| state | count(tpid) |
+-----+-----+
| delhi |          1 |
+-----+-----+
1 row in set (0.00 sec)
```

2. List the details of tourist place where maximum number of tourist visited.

Select tp.tpid,tp.tpname,tp.state, count(tv.tid)
From tourist_place01 tp,visits01 tv where tv.tpid=tp.tpid group by tv.tpid order by
count(tv.tpid) desc limit 1;

```
mysql> Select tp.tpid,tp.tpname,tp.state, count(tv.tid)
-> From tourist_place01 tp,visits01 tv where tv.tpid=tp.tpid group by tv.tpid order by count(tv.tpid) desc limit 1;
+-----+-----+-----+-----+
| tpid | tpname | state | count(tv.tid) |
+-----+-----+-----+-----+
| 11 | tajmahal | delhi |          2 |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

3. List the details of tourist visited all on tourist places of the state karnataka

Select * from tourist01 t where t.tid in (select tid from visits01
Join tourist01 on visits01.tpid = tourist01.tpid
Where state = 'karnataka' group by tid having count(state) in (
Select count(state) from tourist01 where state= 'karnataka'));

4. Display the details of the tourists visited atleast one tourists place of the states but visited all the states tourist places

Select * from tourist01 t where t.tid in (select tid from visits01 join tourist01 on visits01.tpid = tourist01.tpid group by tid having count(distinct state) in (select count(distinct state) from tourist01));

5. Display the details of the tourist place visited by the tourist of all country.

Select * from tourist_place01 where tpid in (select tpid from visits01 join tourist01 on visits01.tid=tourist01.tid group by tpid having count(distinct country)=(select count(distinct country) from tourist01));