

## Lab Cycle : 1

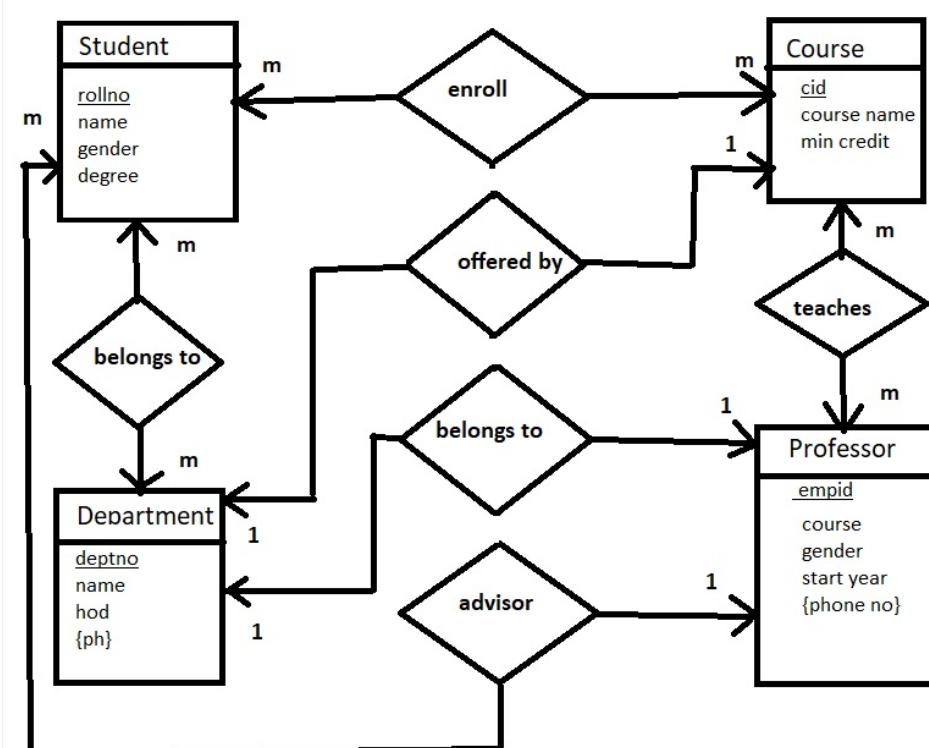
### Experiment : 1

Date : 11-04-2022

AIM: A student is described by unique roll no, name, degree, year, sex. Department have department no which is unique, name, hod and phone no. A professor is described by employee id which is unique, name, sex, start year, phone no.

Department offers courses which is described by unique course id, name, min.credits. A student can enrol for a course which is described by roll no, course id. Database also stores information about professors teaching for different courses which is described by employee id, course id, sem, year and class room. Create a schema with required integrity constraints, Insert tuples into tables? Also draw the ER Diagram?

### ER-Diagram



Query:

- `create table department(dept_id int primary key,dname varchar(30),hod varchar(20));`
- `desc department;`

Output:

| Field   | Type        | Null | Key | Default | Extra |
|---------|-------------|------|-----|---------|-------|
| dept_id | int         | NO   | PRI | NULL    |       |
| dname   | varchar(30) |      | YES |         | NULL  |
| hod     | varchar(20) |      | YES |         | NULL  |

- `create table dept_ph(dept_id int,phno numeric(10),primary key(dept_id,phno));`
- `desc dept_ph;`

Output:

| Field   | Type          | Null | Key | Default | Extra |
|---------|---------------|------|-----|---------|-------|
| dept_id | int           | NO   | PRI | NULL    |       |
| phno    | decimal(10,0) |      | NO  | PRI     | NULL  |

- `create table professor(empid int primary key,pname varchar(20),sex varchar(7),strt_yr int,dept_id int,foreign key(dept_id)references department(dept_id)on delete cascade);`
- `desc professor;`

**Output:**

| Field   | Type        | Null | Key | Default | Extra |
|---------|-------------|------|-----|---------|-------|
| empid   | int         | NO   | PRI | NULL    |       |
| pname   | varchar(20) |      | YES |         | NULL  |
| sex     | varchar(5)  |      | YES |         | NULL  |
| strt_yr | int         | YES  |     | NULL    |       |
| dept_id | int         | YES  | MUL | NULL    |       |

- `create table prof_ph(empid int,phno numeric(10),primary key(empid,phno));`
- `desc prof_ph;`

**Output:**

| Field | Type          | Null | Key | Default | Extra |
|-------|---------------|------|-----|---------|-------|
| empid | int           | NO   | PRI | NULL    |       |
| phno  | decimal(10,0) |      | NO  | PRI     | NULL  |

- `create table student(rollno int primary key,sname varchar(20),degree varchar(20),year int,sex varchar(7),dept_id int,empid int,foreign key(dept_id)references department(dept_id)on delete cascade,foreign key(empid)references professor(empid)on delete cascade);`
- `desc student;`

**Output:**

| Field   | Type        | Null | Key | Default | Extra |
|---------|-------------|------|-----|---------|-------|
| rollno  | int         | NO   | PRI | NULL    |       |
| sname   | varchar(20) |      | YES |         | NULL  |
| degree  | varchar(20) |      | YES |         | NULL  |
| year    | int         | YES  |     | NULL    |       |
| sex     | varchar(7)  |      | YES |         | NULL  |
| dept_id | int         | YES  | MUL | NULL    |       |
| empid   | int         | YES  | MUL | NULL    |       |

- `create table course(cid int primary key,cname varchar(20),min_cred int,dept_id int,empid int,foreign key(dept_id)references department(dept_id)on delete cascade,foreign key(empid)references professor(empid)on delete cascade);`
- `desc course;`

**Output:**

| Field    | Type        | Null | Key | Default | Extra |
|----------|-------------|------|-----|---------|-------|
| cid      | int         | NO   | PRI | NULL    |       |
| cname    | varchar(20) |      | YES |         | NULL  |
| min_cred |             | int  | YES |         | NULL  |
| dept_id  | int         | YES  | MUL | NULL    |       |
| empid    | int         | YES  | MUL | NULL    |       |

- `create table enrollment(rollno int,cid int,primary key(rollno,cid));`
- `desc enrollment;`

**Output:**

| Field  | Type | Null | Key | Default | Extra |
|--------|------|------|-----|---------|-------|
| rollno | int  | NO   | PRI | NULL    |       |
| cid    | int  | NO   | PRI | NULL    |       |

- `insert into department values (121,'MCA','Prof. Reena'), (122,'CSE','Prof. Madhav'), (123,'MBA','Prof. John');`
- `select * from department;`

**Output:**

| dept_id | dname | hod          |
|---------|-------|--------------|
| 121     | MCA   | Prof. Reena  |
| 122     | CSE   | Prof. Madhav |
| 123     | MBA   | Prof. John   |

- `insert into professor values(1001,'Alice','Female',1998,121),  
(1002,'Madhu','Male',2002,121),(1003,'Nidhin','Male',1992,122),(1004,'Sheela','Female',2001,123),(1005,'Viji','Female',1990,123),(1006,'Arun','Male',1990,122);`
- `select * from professor;`

| empid | pname  | sex    | strt_yr | dept_id |
|-------|--------|--------|---------|---------|
| 1001  | Alice  | Female | 1998    | 121     |
| 1002  | Madhu  | Male   | 2002    | 121     |
| 1003  | Nidhin | Male   | 1992    | 122     |
| 1004  | Sheela | Female | 2001    | 123     |
| 1005  | Viji   | Female | 1990    | 123     |
| 1006  | Arun   | Male   | 1990    | 122     |

- `insert into student values(1,'Anju','BCA',2019,'Female',121,1001),  
(2,'Botto','BscCS',2018,'Male',122,1003),(3,'Arjun','BscPh',2019,'Male',121,1002),(4,  
'Jinu','BBA',2019,'Male',123,1004);`
- `select * from student;`

Output:

| rollno | sname | degree | year | sex    | dept_id | empid |
|--------|-------|--------|------|--------|---------|-------|
| 1      | Anju  | BCA    | 2019 | Female | 121     | 1001  |
| 2      | Botto | Bsc CS | 2018 | Male   | 122     | 1003  |
| 3      | Arjun | Bsc Ph | 2019 | Male   | 121     | 1002  |
| 4      | Jinu  | BBA    | 2019 | Male   | 123     | 1004  |

- `insert into prof_ph values (1001,9999999991), (1002,9999999992),  
(1003,9999999993), (1004,9999999994),(1005,9999999995), (1006,9999999996);`
- `select * from prof_ph;`

**Output:**

| <b>empid</b> | <b>phno</b> |
|--------------|-------------|
| 1001         | 9999999991  |
| 1002         | 9999999992  |
| 1003         | 9999999993  |
| 1004         | 9999999994  |
| 1005         | 9999999995  |
| 1006         | 9999999996  |

- **insert into dept\_ph values (121,9999999980), (122,9999999981), (123,9999999982);**
- **select \* from dep\_ph;**

**Output:**

| <b>dept_id</b> | <b>phno</b> |
|----------------|-------------|
| 121            | 9999999980  |
| 122            | 9999999981  |
| 123            | 9999999982  |

- **insert into course value(201,'Btech CSE',8,122,1003),(202,'Mtech CSE',10,122,1003), (203,'MCA',9,121,1001),(204,'MBA',9,123,1005);**
- **select \* from course;**

**Output:**

| <b>cid</b> | <b>cname</b> | <b>min_cred</b> | <b>dept_id</b> | <b>empid</b> |
|------------|--------------|-----------------|----------------|--------------|
| 201        | Btech CSE    | 8               | 122            | 1003         |
| 202        | Mtech CSE    | 10              | 122            | 1003         |
| 203        | MCA          | 9               | 121            | 1001         |
| 204        | MBA          | 9               | 123            | 1005         |

- **insert into enrollment values (1,203), (2,201), (3,203), (4,204);**

- **select \* from enrollment;**

Output:

| rollno | cid |
|--------|-----|
| 1      | 203 |
| 2      | 201 |
| 3      | 203 |
| 4      | 204 |

1. Delete tuples from professor table with start year as 1992?

- **delete from professor where strt\_yr=1992;**
- **select \* from professor;**

| empid | pname  | sex    | strt_yr | dept_id |
|-------|--------|--------|---------|---------|
| 1001  | Alice  | Female | 1998    | 121     |
| 1002  | Madhu  | Male   | 2002    | 121     |
| 1004  | Sheela | Female | 2001    | 123     |
| 1005  | viji   | Female | 1990    | 123     |
| 1006  | Arun   | Male   | 1990    | 122     |

2. Change phone no of all professors working in mca department to 0476-4567878?

- **update prof\_ph set phno=04764567878 where empid in(select empid from professor where dept\_id=(select dept\_id from department where dname='MCA'));**
- **select \* from prof\_ph;**

| empid | phno       |
|-------|------------|
| 1001  | 4764567878 |
| 1002  | 4764567878 |
| 1003  | 9999999993 |
| 1004  | 9999999994 |
| 1005  | 9999999995 |
| 1006  | 9999999996 |

3. List the name of professors who joined after 1990 in alphabetic order?

- **select \* from professor where strt\_yr>1990 order by pname;**

| empid | pname  | sex    | strt_yr | dept_id |
|-------|--------|--------|---------|---------|
| 1001  | Alice  | Female | 1998    | 121     |
| 1002  | Madhu  | Male   | 2002    | 121     |
| 1004  | Sheela | Female | 2001    | 123     |

4. Obtain name of professors who have joined between 1990 and 2000?

- **select \* from professor where strt\_yr between 1990 and 2000;**

| empid | pname | sex    | strt_yr | dept_id |
|-------|-------|--------|---------|---------|
| 1001  | Alice | Female | 1998    | 121     |
| 1005  | Viji  | Female | 1990    | 123     |
| 1006  | Arun  | Male   | 1990    | 122     |

5. Obtain the roll no, name of all women professors in mca department?

- **select empid,pname from professor where sex='Female' and dept\_id=(select dept\_id from department where dname='MCA');**

| empid | pname |
|-------|-------|
| 1001  | Alice |

6. Get the employee id, name and phone no of professors in the mca dept. who have joined after 2000?

- **select e.empid,e.pname,p.phno from professor e,prof\_ph p where e.empid=p.empid and e strt\_yr>2000 and e.dept\_id=(select dept\_id from department where dname='MCA');**

| empid | pname | phno       |
|-------|-------|------------|
| 1002  | Madhu | 4764567878 |

7. Get the details of students whose advisor name is Alice.

- **select \* from student where empid=(select empid from professor where pname='Alice');**

| rollno | sname | degree | year | sex    | dept_id | empid |
|--------|-------|--------|------|--------|---------|-------|
| 1      | Anju  | BCA    | 2019 | Female | 121     | 1001  |

8. Get the rollno and name of student whose gender is same as advisor.

- **select s.rollno,s.sname from student s,professor p where s.empid=p.empid and s.sex=p.sex;**

| rollno | sname |
|--------|-------|
| 1      | Anju  |
| 3      | Arjun |

9. Get the empid and name of senior most professor?

- **select empid,pname from professor where strt\_yr=(select min(strt\_yr) from professor);**

| empid | pname |
|-------|-------|
| 1005  | Viji  |
| 1006  | Arun  |

10. Get the empid and name of professor who advised atleast one women student

- `select empid,pname from professor where empid in(select empid from student where sex='Female');`

| empid | pname |
|-------|-------|
| 1001  | Alice |

11. Obtain the roll number of students who are currently enrolled for either btech or mca course?

- `select rollno from enrollment where cid in(select cid from course where cname like '%btech%' or cname='MCA');`

| rollno |
|--------|
| 1      |
| 3      |

12. Find out the total number of students in mca dept?

- `select count(*) from student where dept_id=(select dept_id from department where dname='MCA');`

| count(*) |
|----------|
| 2        |

13. Display the list of course name along with the name of faculty handling the subject

- `select c.cname,p.pname from course c,professor p where c.empid=p.empid;`

| cid | cname | min_cred | dept_id | empid |
|-----|-------|----------|---------|-------|
| 203 | MCA   | 9        | 121     | 1001  |
| 204 | MBA   | 9        | 123     | 1005  |

14. Find out the total number of students in each course?

- **select count(\*),cid from enrollment group by cid;**

| count(*) | cid |
|----------|-----|
| 2        | 203 |
| 1        | 201 |
| 1        | 204 |

15. Create a view which contains name ,employee id and phone nos of professors who joined in cse department in or after 1889?

- **create view employee as select e.empid,e.pname,p.phno from professor e,prof\_ph p where e.dept\_id=(select dept\_id from department where dname='CSE') and e.empid=p.empid and e.strt\_yr>=1990;**
- **select \* from employee;**

| empid | pname | phno       |
|-------|-------|------------|
| 1006  | Arun  | 9999999996 |

16. Use the created view to obtain the names of professors in cse department who joined after 1889 and whose name starts with M or ends with n?

- **select \* from employee where pname like 'M%' or pname like '%n';**

| empid | pname | phno       |
|-------|-------|------------|
| 1006  | Arun  | 9999999996 |

17. obtain the name,department id and the total no of courses offered by each department?

- **select d.dname,c.\* from department as d join(select count(cid),dept\_id from course group by dept\_id) c using (dept\_id);**

| dname | count(cid) | dept_id |
|-------|------------|---------|
| MCA   | 1          | 121     |
| MBA   | 1          | 123     |

18. List the details of students in mca department along with advisor name and course enrolled.

- `select s.rollno,s.sname,p.pname,c cname from student s,professor p,course c where s.empid=p.empid and c.cid in(select cid from enrollment where s.rollno=rollno) and s.dept_id=(select dept_id from department where dname='MCA');`

| rollno | sname | pname | cname |
|--------|-------|-------|-------|
| 1      | Anju  | Alice | MCA   |
| 3      | Arjun | Madhu | MCA   |

**Lab Cycle : 1**

**Experiment : 2**

**Date : 19-04-2022**

**AIM:** Data Definition Language (DDL) - These SQL commands are used for creating, modifying, and dropping the structure of database objects. The commands are CREATE, ALTER, DROP, RENAME, and TRUNCATE.

**A. Consider the database for a College. Write Queries to implement the following:**

1. Create a database
  - **CREATE DATABASE college;**
  
2. Select the current database
  - **USE college;**

```
MariaDB [college1]> use college_
Database changed
```

3. Create the following tables:
  - a) Student(roll\_no,name,dob,address,phone\_no,blood\_grp)
    - **CREATE TABLE student(rollno int primary key,name varchar(20),dob date,house\_name varchar(20),place varchar(20),phone\_no varchar(20),blood\_grp varchar(20));**

```
MariaDB [college_]> create Table Student(roll_no int PRIMARY KEY ,name varchar(20),dob date,
address varchar(20),phone_no varchar(10),blood_grp varchar(10));
Query OK, 0 rows affected (0.041 sec)
```

3. Create the following tables:
  - b) Course(Course\_id,Course\_name, course\_duration)
    - **CREATE TABLE course(course\_id int primary key,course\_name varchar(20),couse\_duration varchar(20));**

```
MariaDB [college_]> create Table Course(course_id int PRIMARY KEY , course_name varchar(20), course_duration int);
Query OK, 0 rows affected (0.042 sec)
```

4. List all tables in the current database.

- **SHOW TABLES;**

```
MariaDB [college_]> SHOW TABLES;
+-----+
| Tables_in_college_ |
+-----+
| course
| student
+-----+
2 rows in set (0.001 sec)
```

5. Display the structure of the Student table.

- **DESC student;**

```
MariaDB [college_]> DESC Student;
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| roll_no    | int(11)   | NO   | PRI | NULL    |       |
| name       | varchar(20)| YES  |     | NULL    |       |
| dob        | date      | YES  |     | NULL    |       |
| address    | varchar(20)| YES  |     | NULL    |       |
| phone_no   | varchar(10)| YES  |     | NULL    |       |
| blood_grp  | varchar(10)| YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.029 sec)
```

6. Drop the column blood\_grp from Student table.

- **ALTER TABLE student DROP COLUMN blood\_grp;**

```
MariaDB [college_]> ALTER TABLE Student DROP COLUMN blood_grp;
Query OK, 0 rows affected (0.016 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [college_]> DESC Student;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| roll_no | int(11) | NO   | PRI | NULL    |
| name    | varchar(20)| YES  |     | NULL    |
| dob     | date    | YES  |     | NULL    |
| address | varchar(20)| YES  |     | NULL    |
| phone_no | varchar(10)| YES  |     | NULL    |
+-----+-----+-----+-----+-----+
5 rows in set (0.030 sec)
```

7. Add a new column Adar\_no to the table Student.

- **ALTER TABLE student ADD COLUMN (Adar\_no int(20));**

```
MariaDB [college_]> DESC Student;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| roll_no | int(11) | NO   | PRI | NULL    |
| name    | varchar(20)| YES  |     | NULL    |
| dob     | date    | YES  |     | NULL    |
| address | varchar(20)| YES  |     | NULL    |
| phone_no | varchar(10)| YES  |     | NULL    |
| Adar_no | int(20)  | YES  |     | NULL    |
+-----+-----+-----+-----+-----+
6 rows in set (0.028 sec)
```

8. Change the datatype of phone\_no from varchar to int

- **ALTER TABLE student MODIFY COLUMN phone\_no int;**

```
MariaDB [college_]> ALTER TABLE Student MODIFY phone_no int;
Query OK, 0 rows affected (0.095 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [college_]> DESC Student;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| roll_no | int(11) | NO   | PRI   | NULL    |       |
| name    | varchar(20)| YES  |        | NULL    |       |
| dob     | date    | YES  |        | NULL    |       |
| address | varchar(20)| YES  |        | NULL    |       |
| phone_no | int(11) | YES  |        | NULL    |       |
| Adar_no  | int(20) | YES  |        | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.031 sec)
```

9. Drop the tables.

- **DROP TABLE student;**
- **DROP TABLE course;**

```
MariaDB [college_]> DROP TABLE Student;
Query OK, 0 rows affected (0.018 sec)

MariaDB [college_]> DROP TABLE Course;
Query OK, 0 rows affected (0.018 sec)
```

10. Delete the database.

- **DROP DATABASE college;**

```
MariaDB [college_]> DROP DATABASE college_;
Query OK, 0 rows affected (0.002 sec)
```

- B. Consider the database for an organization. Write Queries to implement the following:
1. Create a database
    - **CREATE DATABASE organization;**

```
MariaDB [(none)]> CREATE DATABASE organization;
Query OK, 1 row affected (0.002 sec)
```

2. Select the current database

- **USE organization;**

```
MariaDB [(none)]> USE organization;
Database changed
MariaDB [organization]>
```

3. Create the following tables:

- a) Employee(emp\_no,emp\_name, DOB, address, mobile\_no, dept\_no, salary)
  - **CREATE TABLE Employee(emp\_no int PRIMARY KEY, emp\_name varchar(20), DOB date, address varchar(20), mobile\_no int, dept\_no int, salary int);**

```
MariaDB [organization]> CREATE TABLE Employee (emp_no int PRIMARY KEY, emp_name varchar(20),
DOB date, address varchar(20),mobile_no int,dept_no int,Salary int);
Query OK, 0 rows affected (0.040 sec)
```

- b) Department(dept\_no,dept\_name, location)
  - **CREATE TABLE Department (deptno int PRIMARY KEY,dept\_name varchar(20),location varchar(20));**

```
MariaDB [organization]> CREATE TABLE Department(deptno int PRIMARY KEY,deptname varchar(20),
location varchar(20));
Query OK, 0 rows affected (0.037 sec)
```

4. List all tables in the current database.

- **SHOW TABLES;**

```
MariaDB [organization]> SHOW TABLES;
+-----+
| Tables_in_organization |
+-----+
| department
| employee
+-----+
2 rows in set (0.001 sec)
```

5. Display the structure of the Employee table.

- **DESC Employee;**

```
MariaDB [organization]> DESC Employee;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| emp_no | int(11) | NO   | PRI   | NULL    |       |
| emp_name | varchar(20) | YES  |       | NULL    |       |
| DOB | date   | YES  |       | NULL    |       |
| address | varchar(20) | YES  |       | NULL    |       |
| mobile_no | int(11) | YES  |       | NULL    |       |
| dept_no | int(11) | YES  |       | NULL    |       |
| Salary | int(11) | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+
7 rows in set (0.027 sec)
```

6. Add a new column Designation to the table Employee.

- **ALTER TABLE Employee ADD COLUMN (designation varchar(20));**

```
MariaDB [organization]> ALTER TABLE Employee ADD COLUMN (designation varchar(20));
Query OK, 0 rows affected (0.016 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [organization]> DESC Employee;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| emp_no | int(11) | NO   | PRI   | NULL    |       |
| emp_name | varchar(20) | YES  |       | NULL    |       |
| DOB | date   | YES  |       | NULL    |       |
| address | varchar(20) | YES  |       | NULL    |       |
| mobile_no | int(11) | YES  |       | NULL    |       |
| dept_no | int(11) | YES  |       | NULL    |       |
| Salary | int(11) | YES  |       | NULL    |       |
| designation | varchar(20) | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+
8 rows in set (0.026 sec)
```

7. Drop the column location from Department table.

- **ALTER TABLE Department DROP COLUMN location;**

```
MariaDB [organization]> ALTER TABLE Department DROP COLUMN location;
Query OK, 0 rows affected (0.017 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [organization]> DESC Department;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| deptno | int(11) | NO   | PRI   | NULL    |       |
| deptname | varchar(20) | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.025 sec)
```

8. Display the structure of the Employee table.

- **DESC Employee;**

```
MariaDB [organization]> DESC Employee;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| emp_no | int(11) | NO   | PRI   | NULL    |       |
| emp_name | varchar(20) | YES  |       | NULL    |       |
| DOB | date   | YES  |       | NULL    |       |
| address | varchar(20) | YES  |       | NULL    |       |
| mobile_no | int(11) | YES  |       | NULL    |       |
| dept_no | int(11) | YES  |       | NULL    |       |
| Salary | int(11) | YES  |       | NULL    |       |
| designation | varchar(20) | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+
8 rows in set (0.025 sec)
```

## B. Constraints

Write queries to implement the following:

1. 1. Create new table Person with attributes PersonID(with constraints NOT NULL,PRIMARY KEY),Name (constraints-NOT NULL),Adar(constraints-NOT NULL UNIQUE), Age (constraints CHECK>18)

- **CREATE TABLE Person(PersonID int Primary Key, Name varchar(20) NOT NULL,adar int NOT NULL UNIQUE, Age int, CHECK(Age>18));**

```
MariaDB [organization]> CREATE TABLE Person(PersonID int Primary Key, Name varchar(20) NOT NULL,
adar int NOT NULL UNIQUE, Age int, CHECK(Age>18));
Query OK, 0 rows affected (0.035 sec)
```

2. CREATE TABLE Orders with attributes OrderID(with constraints-NOT NULL, PRIMARY KEY),OrderNumber(constraints-NOT NULL) and PersonID( set FOREIGN KEY on attribute PersonID referencing the column PersonId of Person table)

- **CREATE TABLE Orders(orderID int PRIMARY KEY,orderNumber int NOT NULL,PersonID int,FOREIGN KEY(PersonID) REFERENCES Person(PersonID));**

```
MariaDB [organization]> CREATE TABLE Orders(orderID int PRIMARY KEY,orderNumber int NOT NULL,PersonID int,  
FOREIGN KEY(PersonID) REFERENCES person(PersonID));  
Query OK, 0 rows affected (0.054 sec)
```

3. Display the structure of Person tables.

- **DESC Person;**

```
MariaDB [organization]> DESC Person;  
+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+  
| PersonID | int(11) | NO | PRI | NULL |  
| Name | varchar(20) | NO | NULL | NULL |  
| adar | int(11) | NO | UNI | NULL |  
| Age | int(11) | YES | NULL | NULL |  
+-----+-----+-----+-----+-----+  
4 rows in set (0.030 sec)
```

4. Display the structure of Orders tables.

- **DESC orders;**

```
MariaDB [organization]> DESC orders;  
+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+  
| orderID | int(11) | NO | PRI | NULL |  
| orderNumber | int(11) | NO | NULL | NULL |  
| PersonID | int(11) | YES | MUL | NULL |  
+-----+-----+-----+-----+-----+  
3 rows in set (0.032 sec)
```

## **Lab Cycle : 1**

### **Experiment : 3**

**Date : 19-04-2022**

#### **AIM: 2a) DML Commands:**

Data Manipulation Language (DML) - These SQL commands are used for storing, retrieving, modifying,

and deleting data. These commands are: SELECT, INSERT, UPDATE, and DELETE

Consider the database for an organization. Write Queries to implement the following:

1. Add 5 rows into the table Employee and Department.

- **INSERT INTO Employee VALUES(101,"John","1990-02-12",  
"no152",1462137,1,10000,"professor"), (102,"Mathew","1982-06-  
21","no168",1421627,1,10000,"professor"), (103,"Preethi","1990-04-  
05","no102",1234612,2,7000,"Clerk"), (104,"Rahma","1995-11-  
11","no192",1241623,3,55000,"Manager"), (105,"Anny","1979-04-  
11","no103",1469210,4,3000,"Peon");**
- **INSERT INTO Department VALUES(1,"MCA","Block C"),  
(1,"Mechanical","Block D"), (1,"Civil","Block A"), (1,"BArch","Block B"),  
(1,"EEE","Block F");**

```
MariaDB [organization]> INSERT INTO Employee VALUES (101,"John","1990-02-12","no152",1462137,1,10000,"Professor"),  
-> (102,"Mathew","1982-06-21","no168",1421627,1,10000,"Professor"),  
-> (103,"Preethi","1990-04-05","no102",1234612,2,7000,"Clerk"),  
-> (104,"Rahma","1995-11-11","no192",1241623,3,55000,"Manager"),  
-> (105,"Anny","1979-04-11","no103",1469210,4,3000,"Peon");  
Query OK, 5 rows affected (0.005 sec)  
Records: 5  Duplicates: 0  Warnings: 0  
  
MariaDB [organization]> INSERT INTO Department VALUES (1,"MCA","Block C"),  
-> (2,"Mechanical","Block D"),  
-> (3,"Civil","Block A"),  
-> (4,"BArch","Block B"),  
-> (5,"EEE","Block F");  
Query OK, 5 rows affected (0.007 sec)  
Records: 5  Duplicates: 0  Warnings: 0
```

2. Display all the records from the above tables.

- **SELECT \* FROM Employee;**
- **SELECT \* FROM Department;**

```
MariaDB [organization]> SELECT * FROM Employee;
+-----+-----+-----+-----+-----+-----+-----+
| emp_no | emp_name | DOB      | address | mobile_no | dept_no | Salary   | designation |
+-----+-----+-----+-----+-----+-----+-----+
| 101    | John     | 1990-02-12 | no152   | 1462137  | 1       | 10000   | Professor   |
| 102    | Mathew   | 1982-06-21 | no168   | 1421627  | 1       | 10000   | Professor   |
| 103    | Preethi  | 1990-04-05 | no102   | 1234612  | 2       | 7000    | Clerk      |
| 104    | Rahma   | 1995-11-11 | no192   | 1241623  | 3       | 55000   | Manager    |
| 105    | Anny    | 1979-04-11 | no103   | 1469210  | 4       | 3000    | Peon      |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.000 sec)

MariaDB [organization]> SELECT * FROM Department;
+-----+-----+-----+
| deptno | deptname | location |
+-----+-----+-----+
| 1      | MCA      | Block C  |
| 2      | Mechanical | Block D  |
| 3      | Civil     | Block A  |
| 4      | BArch    | Block B  |
| 5      | EEE      | Block F  |
+-----+-----+-----+
5 rows in set (0.000 sec)
```

3. Display the emp\_no and name of employees from department no 2.

- **SELECT emp\_no,emp\_name FROM Employee WHERE dept\_no=2;**

```
MariaDB [organization]> SELECT emp_no,emp_name FROM Employee WHERE dept_no=2;
+-----+-----+
| emp_no | emp_name |
+-----+-----+
| 103    | Preethi  |
+-----+-----+
1 row in set (0.000 sec)
```

4. Display emp\_no, emp\_name ,designation,deptno and salary of employees in the descending order of salary.

- **SELECT emp\_no,emp\_name,dept\_no,salary,designation FROM Employee ORDER BY salary DESC;**

```
MariaDB [organization]> SELECT emp_no,emp_name,dept_no,salary,designation FROM Employee ORDER BY salary DESC;
+-----+-----+-----+-----+-----+
| emp_no | emp_name | dept_no | salary | designation |
+-----+-----+-----+-----+-----+
| 104    | Rahma   | 3       | 55000 | Manager    |
| 101    | John    | 1       | 10000 | Professor  |
| 102    | Mathew  | 1       | 10000 | Professor  |
| 103    | Preethi | 2       | 7000  | Clerk      |
| 105    | Anny    | 4       | 3000  | Peon      |
+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)
```

5. Display the emp\_no ,name of employees whose salary is between 2000 and 5000

- **SELECT emp\_no,emp\_name FROM Employee WHERE salary BETWEEN 2000 and 5000;**

```
MariaDB [organization]> SELECT emp_no,emp_name FROM Employee WHERE salary BETWEEN 2000 and 5000;
+-----+-----+
| emp_no | emp_name |
+-----+-----+
|    105 | Anny      |
+-----+-----+
1 row in set (0.000 sec)
```

6. Display the designations without duplicate values.

- **SELECT Designation FROM Employee GROUP BY designation;**

```
MariaDB [organization]> SELECT designation FROM Employee GROUP BY designation;
+-----+
| designation |
+-----+
| Clerk      |
| Manager    |
| Peon       |
| Professor  |
+-----+
4 rows in set (0.000 sec)
```

7. Change the salary of employees to 45000 whose designation is 'Manager'

- **UPDATE Employee set salary = 45000 WHERE designation="Manager";**

```
MariaDB [organization]> UPDATE Employee SET salary=4500 WHERE Designation="Manager";
Query OK, 1 row affected (0.006 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [organization]> SELECT * FROM Employee;
+-----+-----+-----+-----+-----+-----+-----+
| emp_no | emp_name | DOB        | address   | mobile_no | dept_no | Salary | designation |
+-----+-----+-----+-----+-----+-----+-----+
|    101 | John     | 1990-02-12 | no152    | 1462137  |      1  | 10000 | Professor   |
|    102 | Mathew   | 1982-06-21 | no168    | 1421627  |      1  | 10000 | Professor   |
|    103 | Preethi  | 1990-04-05 | no102    | 1234612  |      2  | 7000  | Clerk      |
|    104 | Rahma    | 1995-11-11 | no192    | 1241623  |      3  | 4500  | Manager    |
|    105 | Anny     | 1979-04-11 | no103    | 1469210  |      4  | 3000  | Peon      |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.000 sec)
```

8. Change the mobile number of employees named John

- **UPDATE Employee set mobile\_no= 9142327 WHERE name="John";**

```
MariaDB [organization]> UPDATE Employee SET mobile_no=9142327 WHERE emp_name="John";
Query OK, 1 row affected (0.005 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [organization]> SELECT * FROM Employee;
+-----+-----+-----+-----+-----+-----+-----+
| emp_no | emp_name | DOB      | address | mobile_no | dept_no | Salary | designation |
+-----+-----+-----+-----+-----+-----+-----+
| 101   | John     | 1990-02-12 | no152   | 9142327  | 1       | 10000  | Professor    |
| 102   | Mathew   | 1982-06-21 | no168   | 1421627  | 1       | 10000  | Professor    |
| 103   | Preethi  | 1990-04-05 | no102   | 1234612  | 2       | 7000   | Clerk        |
| 104   | Rahma    | 1995-11-11 | no192   | 1241623  | 3       | 4500   | Manager      |
| 105   | Anny     | 1979-04-11 | no103   | 1469210  | 4       | 3000   | Peon        |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.000 sec)
```

9. Delete all employees whose salary is equal to Rs.7000

- **DELETE FROM Employee WHERE salary=7000;**

```
MariaDB [organization]> DELETE FROM Employee WHERE salary=7000;
Query OK, 1 row affected (0.006 sec)
```

```
MariaDB [organization]> SELECT * FROM Employee;
+-----+-----+-----+-----+-----+-----+-----+
| emp_no | emp_name | DOB      | address | mobile_no | dept_no | Salary | designation |
+-----+-----+-----+-----+-----+-----+-----+
| 101   | John     | 1990-02-12 | no152   | 9142327  | 1       | 10000  | Professor    |
| 102   | Mathew   | 1982-06-21 | no168   | 1421627  | 1       | 10000  | Professor    |
| 104   | Rahma    | 1995-11-11 | no192   | 1241623  | 3       | 4500   | Manager      |
| 105   | Anny     | 1979-04-11 | no103   | 1469210  | 4       | 3000   | Peon        |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.000 sec)
```

10. Retrieve the name,mobile number of all employees whose name start with “A”

- **SELECT emp\_name,mobile\_no FROM Employee WHERE emp\_name LIKE "A%";**

```
MariaDB [organization]> SELECT emp_name,mobile_no FROM Employee WHERE emp_name LIKE "A%";
+-----+-----+
| emp_name | mobile_no |
+-----+-----+
| Anny     | 1469210  |
+-----+-----+
1 row in set (0.000 sec)
```

11. Retrieve the empno,name,salary of all employees working as peon and clerk.

- **SELECT emp\_no,emp\_name,salary FROM Employee WHERE Designation = "Peon" OR Designation = "Clerk";**

```
MariaDB [organization]> SELECT emp_no,emp_name,salary FROM Employee WHERE designation="Peon" OR
designation="clerk";
+-----+-----+
| emp_no | emp_name | salary |
+-----+-----+
|    105 | Anny     |   3000 |
+-----+-----+
1 row in set (0.000 sec)
```

## Lab Cycle : 2

### Experiment : 1

Date : 05-05-2022

**AIM:** creates a stored procedure that selects Customers from a particular City from the "Customers" table

#### SOURCE CODE:

- **create table customer(cust\_id int, cust\_name varchar(30), location varchar(30));**
- **insert into customer values (1, 'Alen', 'Kottayam'), (2, 'Bipin', 'Calicut'), (3, 'Chery', 'Palakkad');**

cust.sql

```
drop procedure cust;
delimiter |
create procedure cust()
begin
    select * from customer;
end |
delimiter ;
```

source ~/adbms/sp/cust.sql;

call cust;

#### OUTPUT

The screenshot shows a terminal window with three tabs. The active tab displays the following MySQL session:

```
mysql: [Warning] Using a password on an unencrypted connection may be insecure!
Enter password:
Your MySQL Connection id is 79
Server version: 10.3.34-MariaDB-0ubuntu20.04.1 Ubuntu 20.04
Copyright (c) 2000, 2019, Oracle, MariaDB Corporation Ab and others.
Type 'help'; or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> use adbms;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with 'A'
Database changed
MariaDB [(none)]> create table customer(cust_id int, cust_name varchar(30), location varchar(30));
Query OK, 0 rows affected (0.01 sec)

MariaDB [(none)]> insert into customer values (1, 'Alen', 'Kottayam'), (2, 'Bipin', 'Calicut'), (3, 'Chery', 'Palakkad');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [(none)]> source ~/adbms/sp/cust.sql;
Query OK, 0 rows affected (0.00 sec)
PROCEDURE adbms.cust does not exist

MariaDB [(none)]> call cust;
+-----+-----+
| cust_id | cust_name | location |
+-----+-----+
| 1 | Alen | Kottayam |
| 2 | Bipin | Calicut |
| 3 | Chery | Palakkad |
+-----+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> 
```

## Experiment : 2

Date : 05-05-2022

AIM: Write a stored procedure to display the details of a particular customer where name is passed as a parameter

### SOURCE CODE

- **create table customer(cust\_id int, cust\_name varchar(30), location varchar(30));**
- **insert into customer values (1, 'Alen', 'Kottayam'), (2, 'Bipin', 'Calicut'), (3, 'Chery', 'Palakkad');**
- **select \* from customer;**

#### cust1.sql

```
drop procedure cust1;
delimiter |
create procedure cust1(in name varchar(30))
begin
    select * from customer where cust_name = name;
end
|
delimiter ;
```

source ~/adbms/sp/cust1.sql;  
call cust1('Chery');

### OUTPUT

The screenshot shows a terminal window with four tabs open. The active tab displays the execution of a MySQL stored procedure named cust1. The command `source ~/adbms/sp/cust1.sql;` is run, followed by `call cust1('Chery');`. The output shows the creation of the procedure and its execution, displaying the customer record for 'Chery'.

```
sreejith@jarvis:~$ mysql -u sreejith -p
Enter password:
Welcome to MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 71
Server version: 10.3.34-MariaDB-ubuntu.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use adbms;
MariaDB [adbms]> source ~/adbms/sp/cust1.sql;
MariaDB [adbms]> call cust1('Chery');
+----+-----+-----+
| cust_id | cust_name | location |
+----+-----+-----+
|      3 | Chery     | Palakkad |
+----+-----+-----+
1 row in set (0.000 sec)

Query OK, 1 rows affected (0.009 sec)

MariaDB [adbms]> 
```

## **Experiment : 3**

**Date : 05-05-2022**

AIM: Write a stored procedure to find and display a student is passed or failed by passing rollnumber

### SOURCE CODE

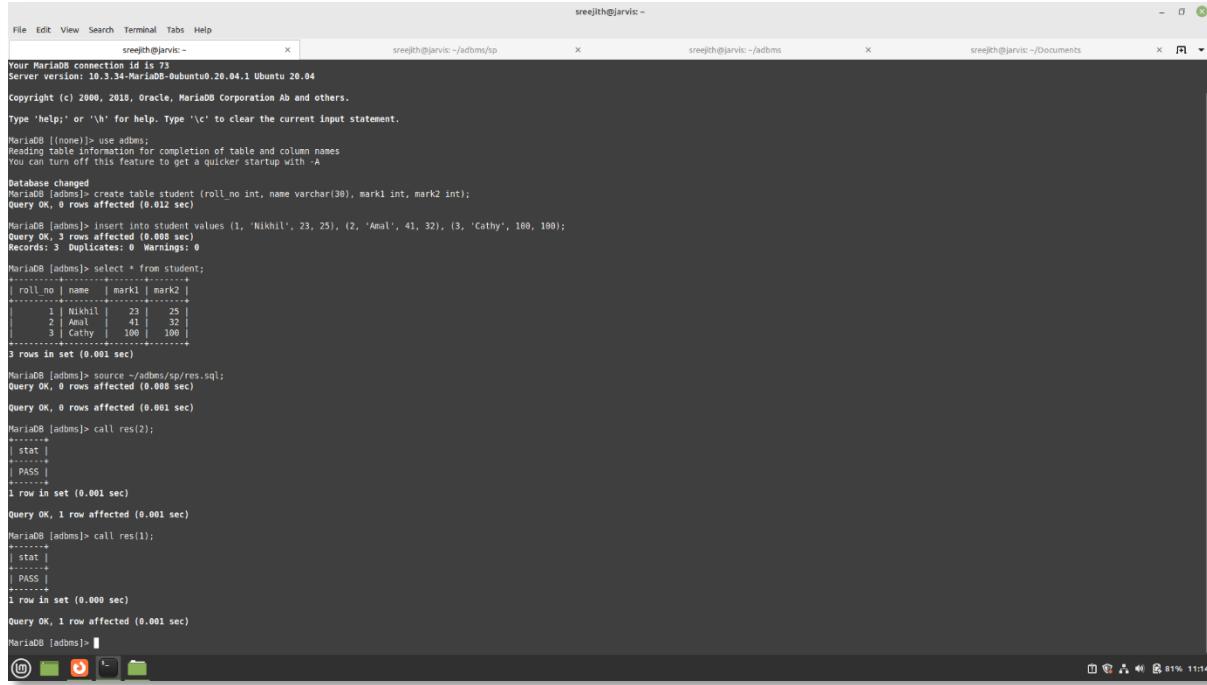
- **create table student (roll\_no int, name varchar(30), mark1 int, mark2 int);**
- **insert into student values (1, 'Nikhil', 23, 25), (2, 'Amal', 41, 32), (3, 'Cathy', 100, 100)**

### res.sql

```
drop procedure res;
delimiter |
create procedure res(in rno int)
begin
    declare m1, m2, percentage, tot int;
    declare stat varchar(30);
    select mark1, mark2 into m1, m2 from student where
roll_no=rno;
    set tot = m1+m2;
    set percentage = (tot*150)/100;
    if(percentage>40) then
        set stat = 'PASS';
    else
        set stat = 'FAIL';
    end if;
    select stat;
end
|
delimiter ;
```

```
source ~/adbms/sp/res.sql;
call res(2);
call res(1);
```

## OUTPUT



The screenshot shows a terminal window with four tabs open, all titled 'sreejith@jarvis:~'. The content of the tabs is as follows:

```
Your MariaDB connection id is 73
Server version: 10.3.34-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> use adbms;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with 'A'
Database changed
MariaDB [adbms]> create table student (roll_no int, name varchar(30), mark1 int, mark2 int);
Query OK, 0 rows affected (0.012 sec)

MariaDB [adbms]> insert into student values (1, 'Nikhil', 23, 25), (2, 'Anil', 41, 32), (3, 'Cathy', 100, 100);
Query OK, 3 rows affected (0.008 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [adbms]> select * from student;
+-----+-----+-----+
| roll_no | name   | mark1 | mark2 |
+-----+-----+-----+
| 1      | Nikhil | 23    | 25    |
| 2      | Anil   | 41    | 32    |
| 3      | Cathy  | 100   | 100   |
+-----+-----+-----+
3 rows in set (0.001 sec)

MariaDB [adbms]> source ./adbms/sp/res.sql;
Query OK, 0 rows affected (0.000 sec)

Query OK, 0 rows affected (0.001 sec)

MariaDB [adbms]> call res(2);
+-----+
| stat |
+-----+
| PASS |
+-----+
1 row in set (0.001 sec)

Query OK, 1 row affected (0.001 sec)

MariaDB [adbms]> call res(1);
+-----+
| stat |
+-----+
| PASS |
+-----+
1 row in set (0.000 sec)

Query OK, 1 row affected (0.001 sec)

MariaDB [adbms]>
```

## **Experiment : 4**

**Date : 05-05-2022**

AIM: Create a stored procedure to determine membership of customer based on the following credits;

Above 5000 = Membership Platinum

1000 to 5000 = Gold

< 1000 = silver

### SOURCE CODE

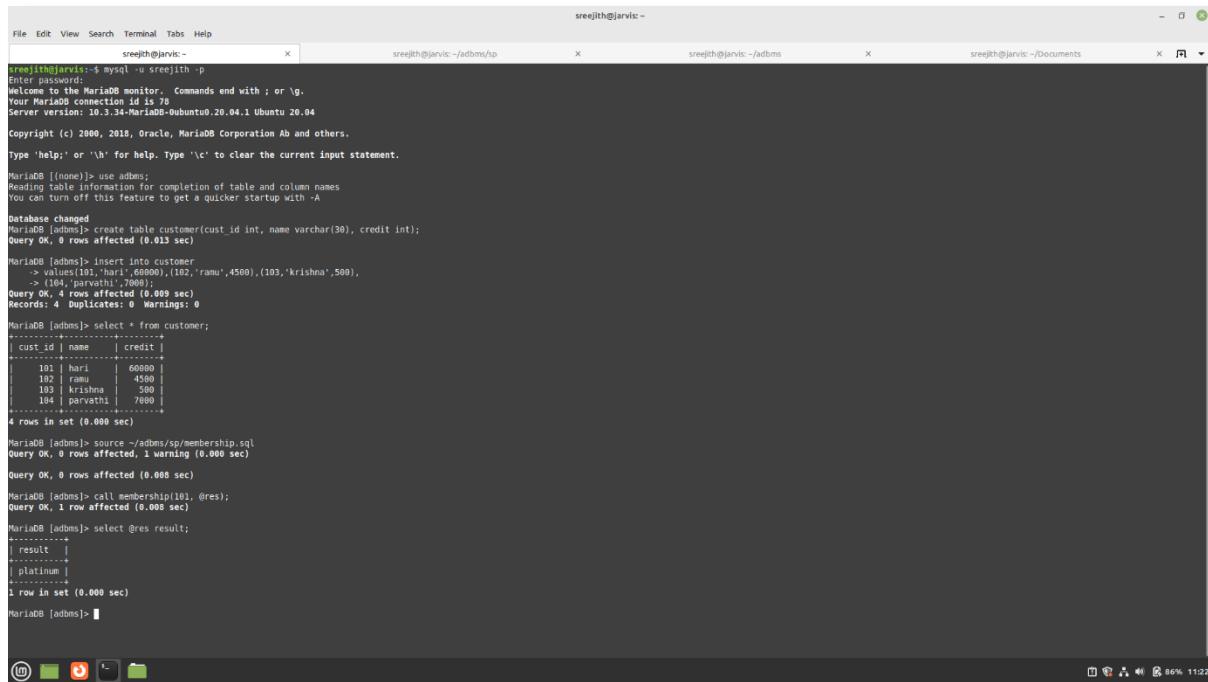
- `create table customer(cust_id int, name varchar(30), credit int);`
- `insert into customer values(101,'hari',60000), (102,'ramu',4500), (103,'krishna',500), (104,'parvathi',7000);`
- `select * from customer;`

### membership.sql

```
drop procedure if exists membership;
delimiter $$ 
create procedure membership(in cust int, out res varchar(30))
begin
declare cred int;
select credit into cred from customer where cust_id=cust
if cred >5000 then
set res='platinum';
elseif (cred>1000 && cred<5000) then
set res='gold';
else
set res='silver';
end if;
end $$ 
delimiter ;
```

```
source ~/adbms/sp/membership.sql
call membership(101, @res);
select @res result;
```

## OUTPUT



The screenshot shows a Linux desktop environment with three terminal windows open. The first window is titled 'sreejith@jarvis:~' and contains the MySQL command-line interface. The user has connected to a database named 'adems'. They have created a table 'customer' with columns 'cust\_id', 'name', and 'credit'. Four rows are inserted into the table: (101, 'hari', 60000), (102, 'ramu', 4500), (103, 'krishna', 500), and (104, 'parvathi', 7000). The user then runs a query to select all data from the 'customer' table, which returns four rows. Next, they run a 'source' command to execute a script named 'membership.sql'. This script contains a stored procedure 'call membership(101, @res)'. The user then runs a query to select the result of this procedure, which returns a single row labeled 'platinum'. The second window is titled 'sreejith@jarvis:~/adems/sp' and the third is titled 'sreejith@jarvis:~/Documents'. The bottom of the screen shows the Unity desktop interface with its characteristic icons and status bar.

```
sreejith@jarvis:~$ mysql -u sreejith -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 78
Server version: 10.3.34-MariaDB-Ubuntu0.20.04.1 Ubuntu 20.04
Copyright (c) 2000, 2019, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use adems;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with '-A'

Database changed
MariaDB [adems]> create table customer(cust_id int, name varchar(30), credit int);
Query OK, 0 rows affected (0.013 sec)

MariaDB [adems]> insert into customer
    values(101,'hari',60000),(102,'ramu',4500),(103,'krishna',500),
    > (104,'parvathi',7000);
Query OK, 4 rows affected (0.009 sec)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [adems]> select * from customer;
+-----+-----+-----+
| cust_id | name   | credit |
+-----+-----+-----+
| 101    | hari   | 60000 |
| 102    | ramu   | 4500   |
| 103    | krishna| 500    |
| 104    | parvathi| 7000  |
+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [adems]> source ~/adems/sp/membership.sql
Query OK, 0 rows affected, 1 warning (0.000 sec)

Query OK, 0 rows affected (0.008 sec)

MariaDB [adems]> call membership(101, @res);
query OK, 1 row affected (0.008 sec)

MariaDB [adems]> select @res result;
+-----+
| result |
+-----+
| platinum |
+-----+
1 row in set (0.000 sec)

MariaDB [adems]>
```

## **Experiment : 5**

**Date : 10-05-2022**

AIM: Copy contents from one table to another using Cursor

### SOURCE COD

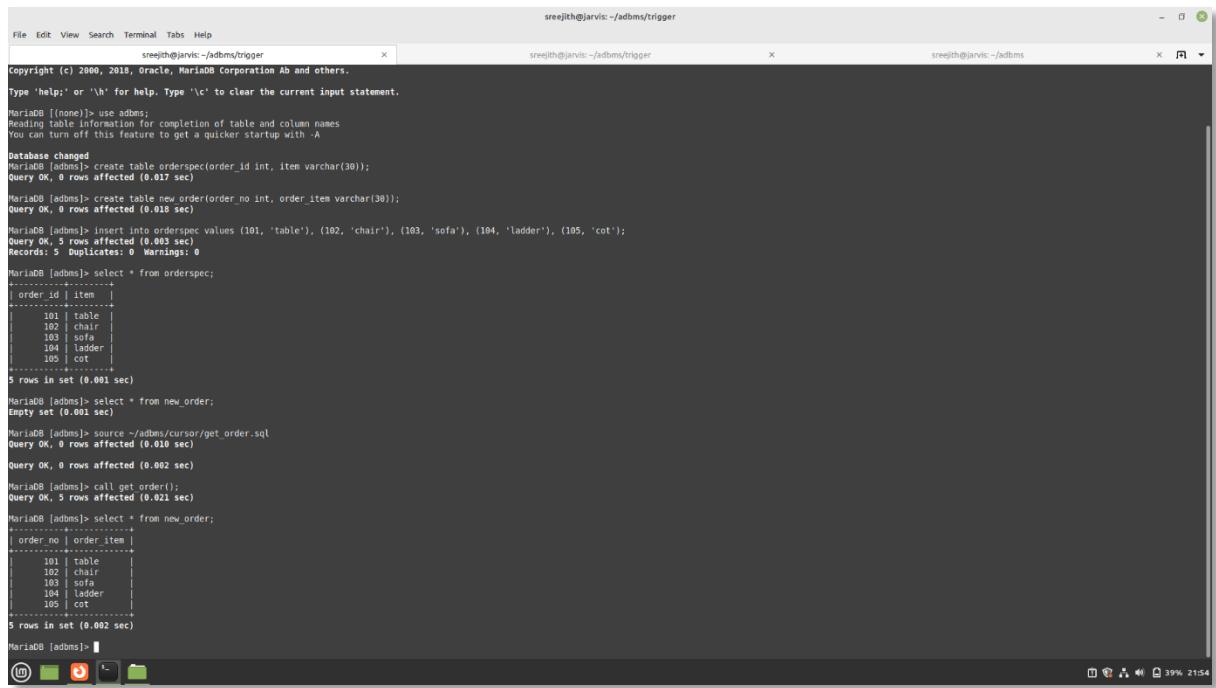
- **create table orderspec(order\_id int, item varchar(30));**
- **create table orderspec(order\_id int, item varchar(30));**
- **insert into orderspec values (101, 'table'), (102, 'chair'), (103, 'sofa'), (104, 'ladder'), (105, 'cot')**
- **select \* from orderspec;**
- **select \* from new\_order;**

### get\_order.sql

```
drop procedure if exists get_order;
delimiter $$ 
create procedure get_order()
begin
    declare flag int default 0;
    declare order_n int;
    declare order_ite varchar(30);
    declare c1 cursor for select order_id, item from orderspec;
    declare continue handler for sqlstate '02000' set flag=1;
    open c1;
    repeat
        fetch c1 into order_n, order_ite;
        if(!flag) then
            insert into new_order(order_no, order_item) values
            (order_n, order_ite);
        end if;
        until flag end repeat;
    end
$$
delimiter ;
```

```
source ~/adbms/cursor/get_order.sql;
call get_order();
```

## OUTPUT



The screenshot shows three terminal windows side-by-side, all titled "sreejith@jarvis: ~/adbsms/trigger". Each window displays an Oracle SQL\*Plus session. The session starts with copyright information, then creates a table named "orderspec" with columns "order\_id" (int) and "item" (varchar(30)). It then inserts five rows of data: (101, 'table'), (102, 'chair'), (103, 'sofa'), (104, 'ladder'), and (105, 'cot'). A select query is run to verify the data, showing the same five rows. Subsequent commands include "select \* from new\_order", "source /adbsms/cursor/get\_order.sql", and "call get\_order();". Finally, another select query is run on the "new\_order" table, which also returns the five rows of data.

```
sreejith@jarvis: ~/adbsms/trigger
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use adbsms;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [adbsms]> create table orderspec(order_id int, item varchar(30));
Query OK, 0 rows affected (0.017 sec)

MariaDB [adbsms]> create table new_order(order_no int, order_item varchar(30));
Query OK, 0 rows affected (0.018 sec)

MariaDB [adbsms]> insert into orderspec values (101, 'table'), (102, 'chair'), (103, 'sofa'), (104, 'ladder'), (105, 'cot');
Query OK, 5 rows affected (0.003 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [adbsms]> select * from orderspec;
+-----+-----+
| order_id | item   |
+-----+-----+
|    101   | table  |
|    102   | chair  |
|    103   | sofa   |
|    104   | ladder |
|    105   | cot    |
+-----+-----+
5 rows in set (0.001 sec)

MariaDB [adbsms]> select * from new_order;
Empty set (0.001 sec)

MariaDB [adbsms]> source /adbsms/cursor/get_order.sql
Query OK, 0 rows affected (0.010 sec)

Query OK, 0 rows affected (0.002 sec)

MariaDB [adbsms]> call get_order();
Query OK, 5 rows affected (0.021 sec)

MariaDB [adbsms]> select * from new_order;
+-----+-----+
| order_no | order_item |
+-----+-----+
|    101   | table      |
|    102   | chair      |
|    103   | sofa       |
|    104   | ladder     |
|    105   | cot        |
+-----+-----+
5 rows in set (0.002 sec)

MariaDB [adbsms]>
```

## **Experiment : 6**

**Date : 10-05-2022**

AIM: Write a stored procedure using cursor to calculate salary of each employee. Employee table have following attributes (emp\_id,emp\_name,no of working days,designation,salary)

Designation=Assistant Professor=1750/day

Designation= Clerk=750/day

Designation= Programmer=1250/day

### SOURCE CODE

- `create table employee (emp_id int, emp_name varchar(30), no_of_working_days int, designation varchar(30), salary int);`
- `insert into employee (emp_id, emp_name, no_of_working_days, designation) values (1, 'sam', 30, 'Assistant Professor'), (2, 'vivek', 60, 'Programmer'), (3, 'suresh', 20, 'Clerk');`
- `select * from employee;`

### setsal.sql

```
drop procedure if exists setsal;
delimiter $$ 
create procedure setsal()
begin
    declare flag int default 0;
    declare sal, id, nowd int;
    declare des varchar(30);
    declare c1 cursor for select emp_id, designation,
no_of_working_days from employee;
    declare continue handler for sqlstate '02000' set flag=1;
    open c1;
    repeat
        fetch c1 into id, des, nowd;
        if(!flag) then
            if(des = 'Assistant Professor') then
                set sal = nowd*1750;
            elseif(des = 'Programmer') then
                set sal = nowd*1250;
            else
                set sal = nowd*750;
            end if;
            update employee set salary = sal where emp_id = id;
        end if;
    end repeat;
end;
```

```

        until flag end repeat;
end
$$
delimiter ;

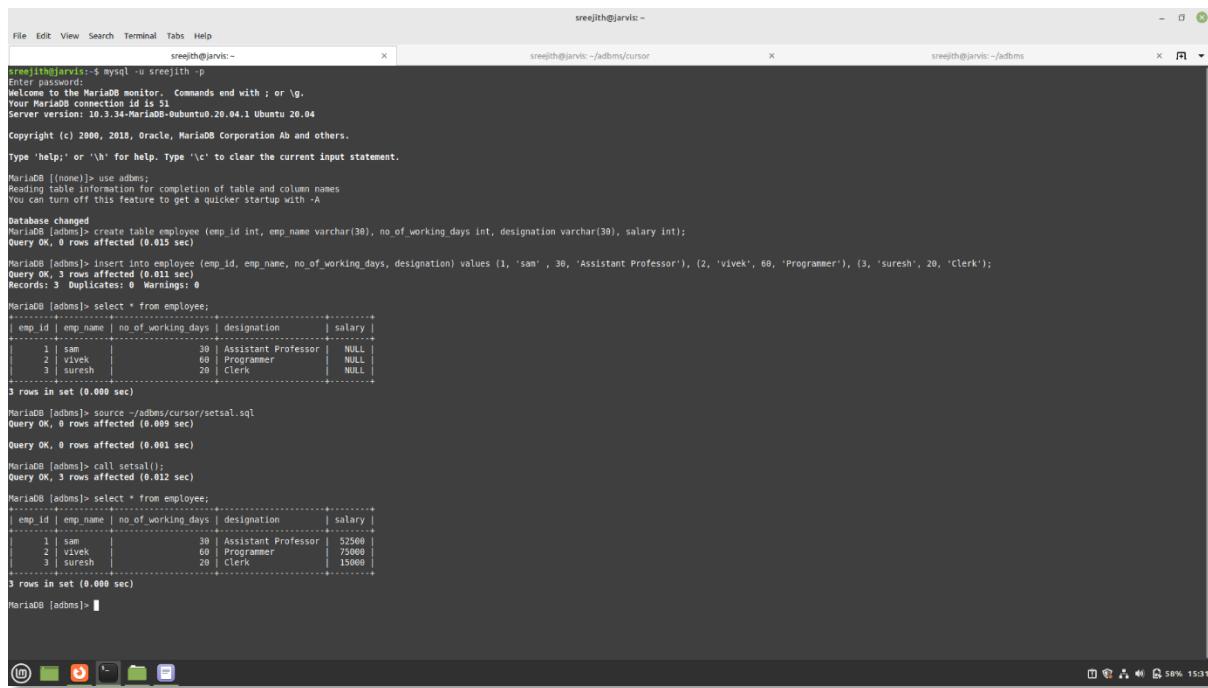
```

```

source ~/adbms/cursor/setsal.sql
call setsal();

```

## OUTPUT



```

sreejith@jarvis:~ mysql -u sreejith -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 51
Server version: 10.3.34-MariaDB-Ubuntu 20.04.1 Ubuntu 20.04
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use adbms;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [adbms]> create table employee (emp_id int, emp_name varchar(30), no_of_working_days int, designation varchar(30), salary int);
Query OK, 0 rows affected (0.015 sec)

MariaDB [adbms]> insert into employee (emp_id, emp_name, no_of_working_days, designation) values (1, 'sam', 30, 'Assistant Professor'), (2, 'vivek', 60, 'Programmer'), (3, 'suresh', 20, 'Clerk');
Query OK, 3 rows affected (0.011 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [adbms]> select * from employee;
+-----+-----+-----+-----+
| emp_id | emp_name | no_of_working_days | designation | salary |
+-----+-----+-----+-----+
| 1 | sam | 30 | Assistant Professor | NULL |
| 2 | vivek | 60 | Programmer | NULL |
| 3 | suresh | 20 | Clerk | NULL |
+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [adbms]> source ~/adbms/cursor/setsal.sql
Query OK, 0 rows affected (0.009 sec)

Query OK, 0 rows affected (0.001 sec)

MariaDB [adbms]> call setsal();
Query OK, 3 rows affected (0.012 sec)

MariaDB [adbms]> select * from employee;
+-----+-----+-----+-----+
| emp_id | emp_name | no_of_working_days | designation | salary |
+-----+-----+-----+-----+
| 1 | sam | 30 | Assistant Professor | 52500 |
| 2 | vivek | 60 | Programmer | 75000 |
| 3 | suresh | 20 | Clerk | 15000 |
+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [adbms]>

```

## **Experiment : 7**

**Date : 13-05-2022**

AIM: Implement procedure to calculate the electricity bill of all customers.

Electricity board charges the following rates to domestic uses to find the consumption of energy.

For first 100 units Rs:2 per unit

101 to 200 units Rs:2.5 per unit

201 to 300 units Rs: 3 per unit

Above 300 units Rs: 4 per unit

Consider the table Bill (customer\_id,name, pre\_reading,cur\_reading,unit,amount)

### SOURCE CODE

- `create table bill(customer_id int,name varchar(30),pre_reading int,cur_reading int,unit int,amount decimal(10,2));`
- `insert into bill(customer_id,name,pre_reading,cur_reading) values(101,'abishek',300,650), (102,'krishnan',450,500),(103,'manu',230,870), (104,'lijo',200,1000);`
- `select * from bill;`

### ebill.sql

```
drop procedure if exists ebill;
delimiter $$ 
create procedure ebill()
begin
    declare flag int default 0;
    declare eno, pr, cr, units int;
    declare amt decimal(10, 2);
    declare c1 cursor for select customer_id, pre_reading,
cur_reading from bill;
    declare continue handler for sqlstate '02000' set flag=1;
    open c1;
    repeat
        fetch c1 into eno, pr, cr;
        if(!flag) then
            set units = cr-pr;
            if(units<=100) then
                set amt = units*2;
```

```

    elseif(units>100 && units<=200) then
        set amt = units*2.5;
    elseif(units>200 && units<=300) then
        set amt = units*3;
    else
        set amt = units*4;
    end if;
    update bill set unit=units, amount=amt where
customer_id=eno;
    end if;
    until flag end repeat;
    close c1;
end
$$
delimiter ;

```

source ~/adbms/cursor/ebill.sql

call ebill();

## OUTPUT

```

sreejith@jarvis:~$ mysql -u sreejith -p
sreejith@jarvis:~$ 
MariaDB [(none)]> use adbms;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
MariaDB [adbms]> create table bill(customer_id int,name varchar(30),pre_reading int,cur_reading
-> int,unit int,amount decimal(10,2));
Query OK, 0 rows affected (0.016 sec)

MariaDB [adbms]> insert into bill(customer_id,name,pre_reading,cur_reading) values(101,'abishek',300,650), (102,'krishnan',450,500),(103,'manu',230,870), (104,'lijo',200,1900);
Query OK, 4 rows affected (0.012 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [adbms]> select * from bill;
+-----+-----+-----+-----+-----+
| customer_id | name   | pre_reading | cur_reading | unit | amount |
+-----+-----+-----+-----+-----+
| 101 | abishek | 300 | 650 | NULL | NULL |
| 102 | krishnan | 450 | 500 | NULL | NULL |
| 103 | manu   | 230 | 870 | NULL | NULL |
| 104 | lijo   | 200 | 1000 | NULL | NULL |
+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [adbms]> source ~/adbms/cursor/ebill.sql
Query OK, 0 rows affected (0.000 sec)

Query OK, 0 rows affected (0.002 sec)

MariaDB [adbms]> call ebill();
Query OK, 4 rows affected (0.013 sec)

MariaDB [adbms]> select * from bill;
+-----+-----+-----+-----+-----+
| customer_id | name   | pre_reading | cur_reading | unit | amount |
+-----+-----+-----+-----+-----+
| 101 | abishek | 300 | 650 | 350 | 1400.00 |
| 102 | krishnan | 450 | 500 | 50 | 2500.00 |
| 103 | manu   | 230 | 870 | 640 | 2560.00 |
| 104 | lijo   | 200 | 1000 | 800 | 3200.00 |
+-----+-----+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [adbms]>

```

## Experiment : 8

Date : 13-05-2022

AIM: Write a stored procedure using a Cursor to calculate TOTMARK (Mark1+ Mark2 +Mark3), AVGMARK (TOTMARK/3), RES (PASS if M1 and M2 are greater than or equal to 50 else FAIL). The program should update all the records based on the above conditions.

### SOURCE CODE

- create table student (regno int,name varchar(30),mark1 int,mark2 int, mark3 int,totmark int, avgmark decimal(12,2),res varchar(30))
- insert into student(regno,name,mark1,mark2,mark3) values(101,'ramesh',67,78,90),(102,'manoj',45,43,23),(103,'hareesh',57,33,78),(104,'unni',76,78,73)
- select \* from student;

### updResult.sql

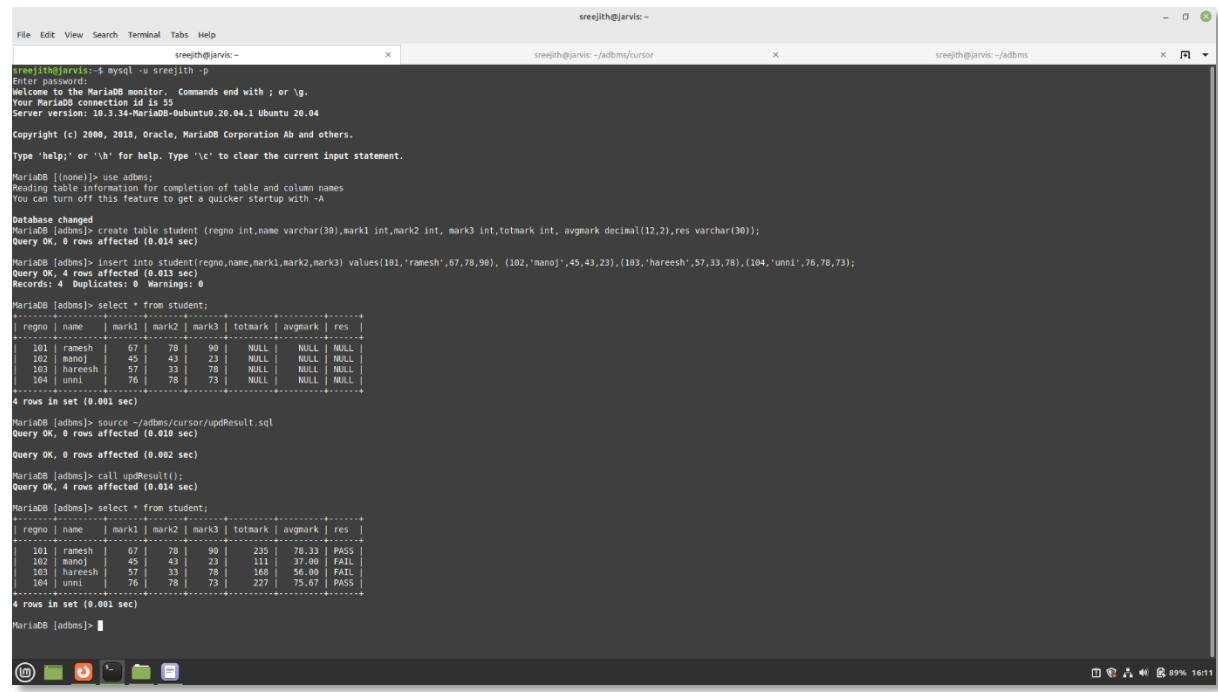
```
drop procedure if exists updResult;
delimiter $$ 
create procedure updResult()
begin
    declare r, m1, m2, m3, t int;
    declare a numeric(10, 2);
    declare result char(10);
    declare flag int default 0;
    declare c1 cursor for select regno, mark1, mark2, mark3 from student;
    declare continue handler for sqlstate '02000' set flag=1;
    open c1;
    repeat
        fetch c1 into r, m1, m2, m3;
        if(!flag) then
            set t = m1+m2+m3;
            set a = t/3;
            if(m1>50 and m2>50 and m3>50) then
                set result = 'PASS';
            else
                set result = 'FAIL';
            end if;
            update student set totmark=t, avgmark=a, res=result where regno=r;
        end if;
    until flag end repeat;
end
```

```
$$
delimiter ;
```

```
source ~/adbms/cursor/updResult.sql
```

```
call updResult();
```

## OUTPUT



```
sreejith@jarvis:~
```

```
sreejith@jarvis:~
```

```
sreejith@jarvis:~/adbms/cursor
```

```
sreejith@jarvis:~/adbms
```

```
sreejith@jarvis:~
```

```
MySQL [(none)]> mysql -u sreejith -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 55
Server version: 10.3.34-MariaDB-Ubuntu0.20.04.1 Ubuntu 20.04
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> use adbms;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with '-A'
Database changed
MariaDB [adbms]> create table student (regno int,name varchar(30),mark1 int,mark2 int,mark3 int,totmark int, avgmark decimal(12,2),res varchar(30));
Query OK, 0 rows affected (0.014 sec)

MariaDB [adbms]> insert into student (regno,name,mark1,mark2,mark3) values(101,'ramesh',67,78,90),(102,'manoj',45,43,23),(103,'hareesh',57,33,78),(104,'unni',76,78,73);
Query OK, 4 rows affected (0.013 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [adbms]> select * from student;
+-----+-----+-----+-----+-----+-----+-----+-----+
| regno | name  | mark1 | mark2 | mark3 | totmark | avgmark | res   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 101  | ramesh | 67   | 78   | 90   | NULL    | 78.33  | PASS  |
| 102  | manoj  | 45   | 43   | 23   | NULL    | 39.67  | FAIL  |
| 103  | hareesh | 57   | 33   | 78   | NULL    | 56.00  | FAIL  |
| 104  | unni   | 76   | 78   | 73   | NULL    | 75.67  | PASS  |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [adbms]> source ~/adbms/cursor/updResult.sql
Query OK, 0 rows affected (0.010 sec)

Query OK, 0 rows affected (0.002 sec)

MariaDB [adbms]> call updResult();
Query OK, 4 rows affected (0.014 sec)

MariaDB [adbms]> select * from student;
+-----+-----+-----+-----+-----+-----+-----+-----+
| regno | name  | mark1 | mark2 | mark3 | totmark | avgmark | res   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 101  | ramesh | 67   | 78   | 90   | 235    | 78.33  | PASS  |
| 102  | manoj  | 45   | 43   | 23   | 111    | 37.00  | FAIL  |
| 103  | hareesh | 57   | 33   | 78   | 168    | 56.00  | FAIL  |
| 104  | unni   | 76   | 78   | 73   | 227    | 75.67  | PASS  |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [adbms]>
```

## Experiment : 9

Date : 17-05-2022

AIM: Familiarise after insert,delete and update triggers

### SOURCE CODE

- create table emps (eid int, ename varchar(30), deptname varchar(30),deptid int);
- insert into emps values(101,'raman','electronics',201),(102,'kumar','computer',202),(103,'unni','mechanic',203),(104,'hareesh','computer',204);
- create table trig\_logg (logs varchar(200));
- select \* from emps;
- source ~/adbms/trigger/ai.sql
- insert into emps values(110, 'ramesh', 'elect', 309);
- select \* from trig\_logg;
- source ~/adbms/trigger/au.sql
- update emps set ename = 'krishna' where ename='krishnan';
- select \* from trig\_logg;
- source ~/adbms/trigger/ad.sql
- delete from emps where eid=110;
- select \* from trig\_logg;

#### ai.sql

```
delimiter $$  
create trigger trig_emp after insert on emps  
for each row  
begin  
insert into trig_logg values('A new row in emps table has been  
inserted');  
end $$  
delimiter ;
```

#### au.sql

```
delimiter $$  
create trigger trig_emp_update after update on emps  
for each row
```

```

begin

    insert into trig_log values('A new row in emps table has
been updated');

end $$

delimiter ;

```

### ad.sql

```

delimiter $$

create trigger trig_emp_delete after delete on emps
for each row

begin

    insert into trig_log values ('A row in emps table has
been deleted');

end $$

delimiter ;

```

### OUTPUT

```

File Edit View Search Terminal Tabs Help
sreejith@jarvis: ~
sreejith@jarvis: ~
sreejith@jarvis: ~

sreejith@jarvis:~$ mysql -u sreejith -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 56
Server version: 10.3.34-MariaDB-Ubuntu20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use adbs;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [adbs]> create table emps (eid int, ename varchar(30), deptname varchar(30),deptid int);
Query OK, 0 rows affected (0.017 sec)

MariaDB [adbs]> insert into emps values(101,'raman','electronics',201),(102,'kumar','computer',202),
    > (103,'unni','mechanic',203),(104,'hareesh','computer',204);
Query OK, 4 rows affected (0.011 sec)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [adbs]> create table trig_log (logs varchar(200));
Query OK, 0 rows affected (0.012 sec)

MariaDB [adbs]> select * from emps;
+----+-----+-----+-----+
| eid | ename | deptname | deptid |
+----+-----+-----+-----+
| 101 | raman | electronics | 201 |
| 102 | kumar | computer | 202 |
| 103 | unni | mechanic | 203 |
| 104 | hareesh | computer | 204 |
+----+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [adbs]> source ~/adbs/trigger/ad.sql
Query OK, 0 rows affected (0.012 sec)

MariaDB [adbs]> insert into emps values(110, 'ramesh', 'elect', 309);
Query OK, 1 row affected (0.011 sec)

MariaDB [adbs]> select * from trig_log;
+-----+
| logs |
+-----+
| A new row in emps table has been inserted |
+-----+
1 row in set (0.001 sec)

MariaDB [adbs]>

```

```

sreejith@jarvis: ~
File Edit View Search Terminal Tabs Help
sreejith@jarvis: ~
MariaDB [adms] > select * from emps;
+---+-----+-----+-----+
| eid | ename | deptname | deptid |
+---+-----+-----+-----+
| 101 | raman | electronics | 201 |
| 102 | kumar | computer | 202 |
| 103 | krishnan | mechanic | 203 |
| 104 | hareesh | computer | 204 |
| 110 | ramesh | elect | 309 |
+---+-----+-----+-----+
5 rows in set (0.00 sec)

MariaDB [adms] > update emps set ename = 'krishna' where ename='krishnan';
Query OK, 1 row affected (0.010 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [adms] > select * from trig_logg;
ERROR 1146 (42S02): Table 'adms.trig_logg' doesn't exist
MariaDB [adms] > select * from trig_logg;
+-----+
| logg |
+-----+
| A new row in emps table has been inserted |
| A new row in emps table has been updated |
| A new row in emps table has been updated |
+-----+
3 rows in set (0.001 sec)

MariaDB [adms] > select * from emps;
+---+-----+-----+-----+
| eid | ename | deptname | deptid |
+---+-----+-----+-----+
| 101 | raman | electronics | 201 |
| 102 | kumar | computer | 202 |
| 103 | krishna | mechanic | 203 |
| 104 | hareesh | computer | 204 |
| 110 | ramesh | elect | 309 |
+---+-----+-----+-----+
5 rows in set (0.001 sec)

MariaDB [adms] > delete from emps where eid=110;
Query OK, 1 row affected (0.009 sec)

MariaDB [adms] > select * from trig_logg;
+-----+
| logg |
+-----+
| A new row in emps table has been inserted |
| A new row in emps table has been updated |
| A new row in emps table has been updated |
| A row in emps table has been deleted |
+-----+
4 rows in set (0.000 sec)

MariaDB [adms] >

```

## Experiment : 10

Date : 17-05-2022

AIM: Familiarise OLD and NEW command in triggers

SOURCE CODE

- select \* from emps;
- insert into emps values (109, 'mohan', 'elect', 209);
- select \* from trig\_logg;
- source ~/adbms/trigger/auon.sql;
- update emps set ename = 'gokul' where ename = 'kumar';
- delet from emps where eid=109;

auon.sql

```
delimiter $$

create trigger trig_emp_update_old_new after update on emps
for each row
begin
    insert into trig_logg values (concat('You have updated an
employee with name:', NEW.ename));
    insert into trig_logg values (concat('You have removed an
employee with name:', OLD.ename));
end $$

delimiter ;
```

OUTPUT

The screenshot shows a terminal window with three panes. The left pane shows the MySQL command-line interface with the following session:

```
sreejith@jarvis:~/adbms/trigger
Enter password:
Welcome to MariaDB monitor.  Commands end with ; or \q.
Your MariaDB connection id is 62
server version: 10.3.34-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04
Copyright (c) 2000, 2020, Oracle, MariaDB Corporation Ab and others.
Type 'help' or '\h' for help. Type '\c' to clear the current input statement.
For server manual, type '\m'.
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
MariaDB [db01]> select * from emps;
+----+-----+-----+-----+
| eid | ename | deptid |
+----+-----+-----+
| 101 | Deepak | electronics | 203
| 102 | kumar | computer | 202
| 103 | Krish | mechanical | 203
| 104 | krish | computer | 204
| 109 | kris | elect | 209
+----+-----+-----+
5 rows in set (0.001 sec)

MariaDB [db01]> insert into emps values (109, 'mohan', 'elect', 209);
Query OK, 1 row affected (0.009 sec)

MariaDB [db01]> select * from trig_logg;
+-----+
| logg |
+-----+
| A new row in emps table has been inserted
| You have inserted an employee with name:mohan |
+-----+
2 rows in set (0.000 sec)

MariaDB [db01]> update emps set ename = 'gokul' where ename = 'kumar';
Query OK, 1 row affected (0.009 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [db01]> select * from trig_logg;
+-----+
| logg |
+-----+
| A new row in emps table has been inserted
| You have inserted an employee with name:mohan |
| A new row in emps table has been updated
| You have updated an employee with name:gokul |
| You have removed an employee with name:kumar |
+-----+
5 rows in set (0.001 sec)
```

The middle and right panes show the output of the trigger logs, which capture the insertion of a new row, the update of an existing row, and the subsequent removal of the old row.

## Experiment : 11

Date : 20-05-2022

AIM: Library issue register program

### SOURCE CODE

- create table issue (memberno int, bookno int, issuedate date, status int default 1);
- source ~/adbms/trigger/maxissue.sql
- insert into issue value(101, 306, '2020-10-11', 1);

### maxissue.sql

```
drop trigger if exists maxissue;
delimiter |
create trigger maxissue before insert on issue
for each row
begin
    declare cnt int default 0;
    select count(*) into cnt from issue where
memberno=new.memberno and status=1;
    if(cnt>=3) then
        signal sqlstate value '45000' set
message_text='maximum books issued!!!';
    end if;
end |
delimiter ;
```

## OUTPUT

**Lab Cycle : 3**

**Experiment : 1**

**Date : 20-05-2022**

AIM: MongoDB installation

STEPS:

1. Open the terminal and write the following command:
  - sudo apt -get install mongodb
  - put the administrator password to execute the task
2. Modify the update index using the following command and then start the mongodb service
  - sudo apt -get update
  - sudo service mongodb start
3. Start the mongodb shell by typing 'mongo' in terminal.

MongoDB versions:

2021 . 2022 . MongoDB 5.0 & Rapid Release

## **Lab Cycle : 4**

### **Experiment : 1**

**Date : 24-05-2022**

AIM: Create a database College with following collections:

1. Student (id, name, age, address, mobile, semester)

- `use colleges45;`
- `db.student.insertMany([{"id:1,name:'Arun', age:24, address:"karott", mobile:9605002123, semester:2},{id:2,name:'Tharun', age:23, address:"thazath", mobile:9605002023, semester:1}]);`

```
switched to db college45
{
    "acknowledged" : true,
    "insertedIds" : [
        ObjectId("62a99ea3df587c228d2d57f4"),
        ObjectId("62a99ea3df587c228d2d57f5")
    ]
}
```

- `db.student.find().pretty();`

---

Output:

```
switched to db colleges45
{
    "_id" : ObjectId("62a99fea05b493cf33088385"),
    "id" : 1,
    "name" : "Arun",
    "age" : 24,
    "address" : "karott",
    "mobile" : 9605002123,
    "semester" : 2
}
{
    "_id" : ObjectId("62a99fea05b493cf33088386"),
    "id" : 2,
    "name" : "Tharun",
    "age" : 23,
    "address" : "thazath",
    "mobile" : 9605002023,
    "semester" : 1
}
```

## 2. Faculty (id, name, dept-name, salary, job\_role)

- db.faculty.insertMany ([{f\_id:11, name:'prasanth', dept\_name:'mca', salary:24000, job\_role:'clerk', age:26}, {f\_id:12, name:'sreesanth', dept\_name:'cse', salary:26000, job\_role:'Assistant\_Prof', age:37}, {f\_id:13, name:'rasanth', dept\_name:'ec', salary:25000, job\_role:'programmer', age:28}, {f\_id:14, name:'suneesh', dept\_name:'mca', salary:25000, job\_role:'HOD', age:32}]);

```
{  
    "acknowledged" : true,  
    "insertedIds" : [  
        ObjectId("62a9a9dbb47cee6f52218221"),  
        ObjectId("62a9a9dbb47cee6f52218222"),  
        ObjectId("62a9a9dbb47cee6f52218223"),  
        ObjectId("62a9a9dbb47cee6f52218224")  
    ]  
}
```

- db.faculty.find().pretty()

Output:

```
switched to db colleges45  
{  
    "_id" : ObjectId("62a9a1312065d0ba5a85c845"),  
    "d_id" : "D_01",  
    "dept_name" : "mca",  
    "location" : "MCA Block"  
}  
{  
    "_id" : ObjectId("62a9a1312065d0ba5a85c846"),  
    "d_id" : "D_02",  
    "dept_name" : "ce",  
    "location" : "Civil Block"  
}  
{  
    "_id" : ObjectId("62a9a1312065d0ba5a85c847"),  
    "d_id" : "D_03",  
    "dept_name" : "cse",  
    "location" : "CS Block"  
}
```

### 3. Department (id, dept-name, location)

- `db.department.insertMany ( [{d_id:'D_01', dept_name:'mca', location:'MCA Block'}, {d_id:'D_02', dept_name:'ce', location:'Civil Block'}, {d_id:'D_03', dept_name:'cse', location:'CS Block'}]);`

Output:

```
switched to db colleges45
{
    "acknowledged" : true,
    "insertedIds" : [
        ObjectId("62a9a1312065d0ba5a85c845"),
        ObjectId("62a9a1312065d0ba5a85c846"),
        ObjectId("62a9a1312065d0ba5a85c847")
    ]
}
```

- `db.department.find().pretty();`

Output:

```
switched to db colleges45
{
    "_id" : ObjectId("62a9a1312065d0ba5a85c845"),
    "d_id" : "D_01",
    "dept_name" : "mca",
    "location" : "MCA Block"
}
{
    "_id" : ObjectId("62a9a1312065d0ba5a85c846"),
    "d_id" : "D_02",
    "dept_name" : "ce",
    "location" : "Civil Block"
}
{
    "_id" : ObjectId("62a9a1312065d0ba5a85c847"),
    "d_id" : "D_03",
    "dept_name" : "cse",
    "location" : "CS Block"
}
```

Querires:

1. Retrieve the document with the exact `_id` value “1”.

- `db.student.find({id:1}).pretty();`

```
{  
    "_id" : ObjectId("62a99fea05b493cf33088385"),  
    "id" : 1,  
    "name" : "Arun",  
    "age" : 24,  
    "address" : "karott",  
    "mobile" : 9605002123,  
    "semester" : 2  
}
```

2. Retrieve the documents where the `salary` is greater than 25000.

- `db.faculty.find({salary:{$gt:25000}}).pretty();`

```
{  
    "_id" : ObjectId("62a99f9efc81994727713b22"),  
    "f_id" : 12,  
    "name" : "sreesanth",  
    "dept_name" : "cse",  
    "salary" : 26000,  
    "job_role" : "HOD",  
    "age" : 37  
}
```

3. Find the documents with the `salary` less than 25000

- `db.faculty.find({salary:{$lt:25000}}).pretty();`

```
{  
    "_id" : ObjectId("62a99f9efc81994727713b21"),  
    "f_id" : 11,  
    "name" : "prasanth",  
    "dept_name" : "mca",  
    "salary" : 12000,  
    "job_role" : "clerk",  
    "age" : 26  
}
```

4. Find documents with 'salary' greater than or equal to 25000.

- `db.faculty.find({salary:{$gte:25000}}).pretty();`

```
{  
    "_id" : ObjectId("62a99f9efc81994727713b22"),  
    "f_id" : 12,  
    "name" : "sreesanth",  
    "dept_name" : "cse",  
    "salary" : 26000,  
    "job_role" : "HOD",  
    "age" : 37  
}  
{  
    "_id" : ObjectId("62a99f9efc81994727713b23"),  
    "f_id" : 13,  
    "name" : "rasanth",  
    "dept_name" : "ec",  
    "salary" : 25000,  
    "job_role" : "programmer",  
    "age" : 28  
}  
{  
    "_id" : ObjectId("62a99f9efc81994727713b24"),  
    "f_id" : 14,  
    "name" : "suneesh",  
    "dept_name" : "mca",  
    "salary" : 25000,  
    "job_role" : "Assistant_Prof",  
    "age" : 32  
}
```

5. Write query returns documents where the salary is less than or equal to 15000.

- `db.faculty.find({salary:{$lte:15000}}).pretty();`

```
{  
    "_id" : ObjectId("62a99f9efc81994727713b21"),  
    "f_id" : 11,  
    "name" : "prasanth",  
    "dept_name" : "mca",  
    "salary" : 12000,  
    "job_role" : "clerk",  
    "age" : 26  
}
```

6. Write query returns documents where the dept-name field contains the given values(mca, cse).

- `db.faculty.find({dept_name:{$in :['mca','cse']} }).pretty();`

```
{  
    "_id" : ObjectId("62a99f9efc81994727713b21"),  
    "f_id" : 11,  
    "name" : "prasanth",  
    "dept_name" : "mca",  
    "salary" : 12000,  
    "job_role" : "clerk",  
    "age" : 26  
}  
{  
    "_id" : ObjectId("62a99f9efc81994727713b22"),  
    "f_id" : 12,  
    "name" : "sreesanth",  
    "dept_name" : "cse",  
    "salary" : 26000,  
    "job_role" : "HOD",  
    "age" : 37  
}  
{  
    "_id" : ObjectId("62a99f9efc81994727713b24"),  
    "f_id" : 14,  
    "name" : "suneesh",  
    "dept_name" : "mca",  
    "salary" : 25000,  
    "job_role" : "Assistant_Prof",  
    "age" : 32  
}
```

7. Find documents where the semester fields do not contain the given values (1 and 3)

- `db.student.find({semester:{$nin :[1,3]} }).pretty();`

```
{  
    "_id" : ObjectId("62a99fea05b493cf33088385"),  
    "id" : 1,  
    "name" : "Arun",  
    "age" : 24,  
    "address" : "karott",  
    "mobile" : 9605002123,  
    "semester" : 2  
}
```

8. Find documents where the value of the D\_id field is not equal to D\_01 in the department collection.

- `db.department.find({d_id:{$ne:'D_01'}}).pretty();`

```
{  
    "_id" : ObjectId("62a9a1312065d0ba5a85c846") ,  
    "d_id" : "D_02" ,  
    "dept_name" : "ce" ,  
    "location" : "Civil Block"  
}  
{  
    "_id" : ObjectId("62a9a1312065d0ba5a85c847") ,  
    "d_id" : "D_03" ,  
    "dept_name" : "cse" ,  
    "location" : "CS Block"  
}
```

9. Find documents that match both the following conditions on faculty collections

- ❖ job\_role is equal to “Assistant Professor”
- ❖ age is between 25 and 35

- `db.faculty.find({$and:[{job_role:'Assistant_Prof'}, {$and:[{age:{$gt:25}}, {age:{$lt:35}}]}]}).pretty();`

```
{  
    "_id" : ObjectId("62a99f9efc81994727713b24") ,  
    "f_id" : 14 ,  
    "name" : "suneesh" ,  
    "dept_name" : "mca" ,  
    "salary" : 25000 ,  
    "job_role" : "Assistant_Prof" ,  
    "age" : 32  
}
```

10. Find documents that match either of the following conditions.

- ❖ job\_role is equal to “Programmer” or “Clerk”

- `db.faculty.find({$or:[{job_role:'programmer'}, {job_role:'clerk'}]}).pretty();`

```
{
    "_id" : ObjectId("62a99f9efc81994727713b21"),
    "f_id" : 11,
    "name" : "prasanth",
    "dept_name" : "mca",
    "salary" : 12000,
    "job_role" : "clerk",
    "age" : 26
}
{
    "_id" : ObjectId("62a99f9efc81994727713b23"),
    "f_id" : 13,
    "name" : "rasanth",
    "dept_name" : "ec",
    "salary" : 25000,
    "job_role" : "programmer",
    "age" : 28
}
```

11. Find documents that do not match either of the following conditions.

- ❖ Location is equal to “MCA Block” or “Civil Block”
- db.department.find({\$and:[{location:{\$ne:'MCA Block'}},{location:{\$ne:'Civil Block'}}]}).pretty()

```
{
    "_id" : ObjectId("62a9a1312065d0ba5a85c847"),
    "d_id" : "D_03",
    "dept_name" : "cse",
    "location" : "CS Block"
}
```

12. Find documents where they do not match the given condition on student collection.

- ❖ age is not greater than or equal to 40
- db.faculty.find({age:{\$not:{\$gte:40}}}).pretty()

```
{
    "_id" : ObjectId("62a99f9efc81994727713b21"),
    "f_id" : 11,
    "name" : "prasanth",
    "dept_name" : "mca",
    "salary" : 12000,
    "job_role" : "clerk",
    "age" : 26
}
{
    "_id" : ObjectId("62a99f9efc81994727713b22"),
    "f_id" : 12,
    "name" : "sreesanth",
    "dept_name" : "cse",
    "salary" : 26000,
    "job_role" : "HOD",
    "age" : 37
}
{
    "_id" : ObjectId("62a99f9efc81994727713b23"),
    "f_id" : 13,
    "name" : "rasanth",
    "dept_name" : "ec",
    "salary" : 25000,
    "job_role" : "programmer",
    "age" : 28
}
{
    "_id" : ObjectId("62a99f9efc81994727713b24"),
    "f_id" : 14,
    "name" : "suneesh",
    "dept_name" : "mca",
    "salary" : 25000,
    "job_role" : "Assistant_Prof",
    "age" : 32
}
```

13. Find documents where the job\_role field exists and equal to “HOD”.

- `db.faculty.find({$and:[{job_role:{$exists:true}},{job_role:'HOD'}]}).pretty();`

```
{
    "_id" : ObjectId("62a99f9efc81994727713b22"),
    "f_id" : 12,
    "name" : "sreesanth",
    "dept_name" : "cse",
    "salary" : 26000,
    "job_role" : "HOD",
    "age" : 37
}
```

14. Find documents with an address field on department collections.

- `db.department.find({address:{$exists:true}}).pretty();`

```
{  
    "_id" : ObjectId("62a9a1312065d0ba5a85c845"),  
    "d_id" : "D_01",  
    "dept_name" : "mca",  
    "location" : "MCA Block"  
}  
{  
    "_id" : ObjectId("62a9a1312065d0ba5a85c846"),  
    "d_id" : "D_02",  
    "dept_name" : "ce",  
    "location" : "Civil Block"  
}  
{  
    "_id" : ObjectId("62a9a1312065d0ba5a85c847"),  
    "d_id" : "D_03",  
    "dept_name" : "cse",  
    "location" : "CS Block"  
}
```

## **Lab Cycle : 5**

### **Experiment : 1**

**Date : 30-05-2022**

AIM: Assume that you have two collection, student and college

1. Documents in student collection contains 'name, address, mob, date of birth, Qualification (including course (starting from sslc, plus to, bsc, mca, mark), location, district').
2. Document in college collection include 'name, location, established year, district'

### Queries.

1. find out the total number of students in collection

- **db.student.aggregate([ { \$group : { \_id:null, numberOfStudents : { \$sum: 1 } }, { \$project: { \_id: 0 } } }])**

or

**db.student.aggregate([ { \$count : "numberOfStudents" } ])**

```
db.student.aggregate([
  { $group : { _id:null, numberOfStudents : { $sum: 1 } },
    { $project: { _id: 0 } }
  }
])
// or
db.student.aggregate([
  { $count : "numberOfStudents" }
])
// db.college.aggregate([
//   { $sort: { name: 1 } },
//   { $limit: 10 }
])
```

Output:

```
switched to db data213
college
student
{ "numberOfStudents" : 4 }
{ "numberOfStudents" : 4 }
```

2. How many colleges are in district TVM

- **db.college.aggregate([ { \$match : { District:"Trivandrum" }}, { \$count : "Colleges in trivandrum" }]).pretty();**

```
db.college.aggregate([
  { $match : { District:"Trivandrum"}},
  { $count : "Colleges in trivandrum "}
]).pretty();
```

Output:

```
switched to db data213
college
student
{ "Colleges in trivandrum " : 1 }
```

### 3. Display the details of student and college residing in same district

- **db.student.aggregate([{\$lookup:{from:"college",localField:"District",foreignField: "district",as: "college details" }}]).pretty();**

Output:

```
switched to db data213
college
student
{
  "_id" : ObjectId("629600ab18c6e93aefa42388"),
  "name" : "jayaraj",
  "address" : "kannankuzha",
  "mob" : "9897962356",
  "dob" : "20-04-2000",
  "location" : "panpuzha",
  "district" : "kottayam",
  "qualification" : [
    {
      "sslc" : "pass",
      "mark" : 689
    },
    {
      "plustwo" : "pass",
      "mark" : 976
    },
    {
      "bsc" : "pass",
      "mark" : 2790
    }
  ]
}
```

```
        "plustwo" : "pass",
        "mark" : 976
    },
    {
        "bsc" : "pass",
        "mark" : 2790
    },
    {
        "mca" : "pass",
        "mark" : 1789
    }
],
"college details" : [
    {
        "_id" : ObjectId("6295a2028d2f3ba6ee19c024"),
        "name" : "RIT",
        "Location" : "Vellor",
        "established_year" : "1991",
        "District" : "Kottayam"
    },
    {
        "_id" : ObjectId("6295a2028d2f3ba6ee19c025"),
        "name" : "CET",
        "Location" : "Sreekaryam",
        "established_year" : "1939",
    },
    {
        "_id" : ObjectId("6295a2028d2f3ba6ee19c025"),
        "name" : "CET",
        "Location" : "Sreekaryam",
        "established_year" : "1939",
        "District" : "Trivandram"
    },
    {
        "_id" : ObjectId("6295a2028d2f3ba6ee19c026"),
        "name" : "TKM",
        "Location" : "karikod",
        "established_year" : "1958",
        "District" : "Kollam"
    },
    {
        "_id" : ObjectId("6295a2028d2f3ba6ee19c027"),
        "name" : "GCEK",
        "Location" : "Kannur",
        "established_year" : "1986",
        "District" : "Kannur"
    }
]
}
```

```
{
  "_id" : ObjectId("629600ab18c6e93aefa42389"),
  "name" : "Arjun",
  "address" : "kizhakethil",
  "mob" : "9797662353",
  "dob" : 2000,
  "location" : "mangalm",
  "district" : "trivandram",
  "qualification" : [
    {
      "sslc" : "pass",
      "mark" : 567
    },
    {
      "plustwo" : "pass",
      "mark" : 789
    },
    {
      "bsc" : "pass",
      "mark" : 2600
    },
    {
      "mca" : "pass",
      "mark" : 1899
    }
  ]
}
```

```

      "mca" : "pass",
      "mark" : 1899
    }
  ],
  "college details" : [
    {
      "_id" : ObjectId("6295a2028d2f3ba6ee19c024"),
      "name" : "RIT",
      "Location" : "Vellor",
      "established_year" : "1991",
      "District" : "Kottayam"
    },
    {
      "_id" : ObjectId("6295a2028d2f3ba6ee19c025"),
      "name" : "CET",
      "Location" : "Sreekaryam",
      "established_year" : "1939",
      "District" : "Trivandram"
    },
    {
      "_id" : ObjectId("6295a2028d2f3ba6ee19c026"),
      "name" : "TKM",
      "Location" : "karikod",
      "established_year" : "1958",
      "District" : "Kollam"
    }
  ]
},
```

```

        "established_year" : "1958",
        "District" : "Kollam"
    },
    {
        "_id" : ObjectId("6295a2028d2f3ba6ee19c027"),
        "name" : "GCEK",
        "Location" : "Kannur",
        "established_year" : "1986",
        "District" : "Kannur"
    }
]
}
{
    "_id" : ObjectId("629600ab18c6e93aefa4238a"),
    "name" : "Gokul",
    "address" : "krishnabhavan",
    "mob" : "9097962254",
    "dob" : 1999,
    "location" : "kanjirapalli",
    "district" : "kottayam",
    "qualification" : [
        {
            "sslc" : "pass",
            "mark" : 738
        },
        {
            "hsc" : "distinction",
            "mark" : 856
        }
    ]
}

```

4. Display the name and establishment year of colleges in district ways.

- **db.college.aggregate([  
    { \$sort: {"district":1}},  
    { \$project: { \_id:0, name:1 ,established\_year:1} }  
]).pretty();**
- **db.student.update({}, { \$unset: {}})**

```

db.college.aggregate([
    { $sort: {"district":1}},
    { $project: { _id:0, name:1 ,established_year:1} }
]).pretty();

```

Output:

```

switched to db data213
college
student
[{"name": "RIT", "established_year": "1991"}, {"name": "CET", "established_year": "1939"}, {"name": "TKM", "established_year": "1958"}, {"name": "GCEK", "established_year": "1986"}]

```

5. Display the qualification details of students along with the mark in sorted order(mark)

- **db.student.aggregate([
 {\$sort: {"mark":1}},
 {\$project: {id:0, qualification:1}}]). pretty ();**

```
db.student.aggregate([
  { $sort: {"mark":1}},
  { $project: { _id:0, qualification:1 } }
]).pretty();
```

## 6. Display the details of college in sorted order( name)

- **db.college.aggregate([  
  { \$sort: { name:1 }}  
]).pretty();**

```
db.college.aggregate([
  { $sort: { name:1 }}
]).pretty();
```

Output:

```
switched to db data213
college
student
{
  "_id" : ObjectId("6295a2028d2f3ba6ee19c025"),
  "name" : "CET",
  "Location" : "Sreekaryam",
  "established_year" : "1939",
  "District" : "Trivandram"
}
{
  "_id" : ObjectId("6295a2028d2f3ba6ee19c027"),
  "name" : "GCEK",
  "Location" : "Kannur",
  "established_year" : "1986",
  "District" : "Kannur"
}
{
  "_id" : ObjectId("6295a2028d2f3ba6ee19c024"),
  "name" : "RIT",
  "Location" : "Vellor",
  "established_year" : "1991",
  "District" : "Kottayam"
```

## 7. Find the total number of students who passed mca degree

- **db.student.aggregate([{{ \$match: {qualification:{ \$elemMatch: { mca:"pass" }}}}, { \$group : { id:null, numberOfStudents : { \$sum: 1 } } }, { \$project: { \_id: 0 } }}]).pretty();**

```

db.student.aggregate([
  {
    $match: {
      qualification: { $elemMatch: { mca:"pass" } }
    }
  },
  {
    $group : {
      _id:null,
      numberOfStudents : { $sum: 1 }
    }
  },
  { $project: { _id: 0 } }
]).pretty();

```

Output:

```

switched to db data213
college
student
{ "numberOfStudents" : 3 }

```

8. List the details of college in descending order based on the establishment year.

- **db.college.aggregate([{\$sort: { established\_year:-1 }}]).pretty();**

```

db.college.aggregate([
  { $sort: { established_year:-1 } }
]).pretty();

```

```

{
  "_id" : ObjectId("6295a2028d2f3ba6ee19c024"),
  "name" : "RIT",
  "Location" : "Vellor",
  "established_year" : "1991",
  "District" : "Kottayam"
}
{
  "_id" : ObjectId("6295a2028d2f3ba6ee19c027"),
  "name" : "GCEK",
  "Location" : "Kannur",
  "established_year" : "1986",
  "District" : "Kannur"
}
{
  "_id" : ObjectId("6295a2028d2f3ba6ee19c026"),
  "name" : "TKM",
  "Location" : "Karikod",
  "established_year" : "1958",
  "District" : "Kollam"
}
{
  "_id" : ObjectId("6295a2028d2f3ba6ee19c025"),
  "name" : "CET",
  "Location" : "Sreekaryam",
}

```

9. Find the total mark secured in MCA degree of students.

- **db.student.aggregate([{\$project:{qualification:\$arrayElemAt:[{\$qualification:'\$qualification',3]}},{\$group:{\_id:null,totalMarks:{sum:"\$qualification.mark"}},{\$project:{\_id:0}}}]**

```

> db.student.aggregate([
  { $project : { qualification: { $arrayElemAt:[{$qualification:3} ] } } },
  { $group : { _id:null,totalMarks : { $sum:"$qualification.mark" } } },
  { $project:{_id:0 } })
{ "totalMarks" : 6380 }

```

10. Find the total mark secured in MCA degree of students who born after year 1995

- **db.student.aggregate([{\$match:{dob:{\$gt:1995}}},{\$projrct:{qualification:{\$arrayElemAt:[‘\$qualification’,3]}},{\$group:{\_id:null,totalMarks:{\$sum:”\$qualification.mark”}}},{\$project:{\_id:0}}])**

```
> db.student.aggregate([ {$match : {dob:{$gt:1995}}}, { $project : { qualification:{ $arrayElemAt:[ '$qualification',3 ] } } }, { $group : { _id:null,totalMarks : {$sum: "$qualification.mark"} } }, { $project:{_id:0} } ])
```

{ "totalMarks" : 3300 }

11. Find the document that matches regular expression “mi” with name field of student document

- **db.student.find({name:{\$regex:"mi"},\$option:"i"}).pretty();**

```
> db.student.find({name:{$regex:"mi"},$options:"i"}).pretty();
{
  "_id" : ObjectId("629dfed3da3fefaa2d4a217d"),
  "name" : "milly",
  "address" : "xyz",
  "mob" : "8521165452",
  "dob" : 2000,
  "qualification" : [
    {
      "sslc" : "pass",
      "mark" : 950
    },
    {
      "plustwo" : "pass",
      "mark" : 780
    },
    {
      "bsc" : "pass",
      "mark" : 1230
    },
    {
      "mca" : "fail",
      "mark" : 250
    }
  ],
  "location" : "valayar",
  "district" : "PKD"
}
```

12. Use aggregate function to display the name of the student only that matches “a”

- **db.student.aggregate([{\$project:{name:1}}]);**
- **db.student.aggregate([{\$match:{name:{\$regex:/^a/i}}},{\$project:{\$project:{\_id:0,name:1}}});**

```
> db.student.aggregate([ { $project:{name:1} }])
{ "_id" : ObjectId("629dfed3da3fefaa2d4a2179"), "name" : "Anusree" }
{ "_id" : ObjectId("629dfed3da3fefaa2d4a217a"), "name" : "Sruthy" }
{ "_id" : ObjectId("629dfed3da3fefaa2d4a217b"), "name" : "Aswin" }
{ "_id" : ObjectId("629dfed3da3fefaa2d4a217c"), "name" : "Rajesh" }
{ "_id" : ObjectId("629dfed3da3fefaa2d4a217d"), "name" : "milly" }
> db.student.aggregate([ {$match:{name:{$regex:/^a/i}}}, { $project:{_id:0,name:1} }])
{ "name" : "Anusree" }
{ "name" : "Aswin" }
```

