Ddl commands

PROGRAM 1- DDL

1. **Create a database and two tables-Student and Course with suitable attributes. Insert values into it.**

```
mysql> create database colleges;
Query OK, 1 row affected (0.00 sec)

mysql> use colleges;
Database changed
mysql> create table Student(Roll_no int primary key,
    -> Std_name varchar(20),DOB date,House_name varchar(20),
    -> Place varchar(20),Ph_no int,Blood_group varchar(20));
Query OK, 0 rows affected (0.10 sec)
```

2. **Display the tables in database**

```
mysql> create table Course(Course_id int primary key,
    -> Course_name varchar(20),Course_duration varchar(20));
Query OK, 0 rows affected (0.05 sec)

mysql> show tables;
+--------------------+
| Tables_in_colleges |
+--------------------+
| course             |
| student            |
+--------------------+
2 rows in set (0.05 sec)
```

3. **Display the structure of student table**

```
mysql> desc Student;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Roll_no     | int(11)     | NO   | PRI | NULL    |       |
| Std_name    | varchar(20) | YES  |     | NULL    |       |
| DOB         | date        | YES  |     | NULL    |       |
| House_name  | varchar(20) | YES  |     | NULL    |       |
| Place       | varchar(20) | YES  |     | NULL    |       |
| Ph_no       | int(11)     | YES  |     | NULL    |       |
| Blood_group | varchar(20) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
7 rows in set (0.07 sec)
```

4. **Drop the column-Blood group in student table**

```
mysql> alter table Student drop column Blood_group;
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc Student;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| Roll_no    | int(11)     | NO   | PRI | NULL    |       |
| Std_name   | varchar(20) | YES  |     | NULL    |       |
| DOB        | date        | YES  |     | NULL    |       |
| House_name | varchar(20) | YES  |     | NULL    |       |
| Place      | varchar(20) | YES  |     | NULL    |       |
| Ph_no      | int(11)     | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

5. **Add a new column Aadhar no. to student table**

```
mysql> alter table Student add column (Adhar_no int(20));
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc Student;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| Roll_no    | int(11)     | NO   | PRI | NULL    |       |
| Std_name   | varchar(20) | YES  |     | NULL    |       |
| DOB        | date        | YES  |     | NULL    |       |
| House_name | varchar(20) | YES  |     | NULL    |       |
| Place      | varchar(20) | YES  |     | NULL    |       |
| Ph_no      | int(11)     | YES  |     | NULL    |       |
| Adhar_no   | int(20)     | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

6. **Change the datatype of Phone number to integer in student table**

```
mysql> alter table Student modify column Ph_no int;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc Student;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| Roll_no    | int(11)     | NO   | PRI | NULL    |       |
| Std_name   | varchar(20) | YES  |     | NULL    |       |
| DOB        | date        | YES  |     | NULL    |       |
| House_name | varchar(20) | YES  |     | NULL    |       |
| Place      | varchar(20) | YES  |     | NULL    |       |
| Ph_no      | int(11)     | YES  |     | NULL    |       |
| Adhar_no   | int(20)     | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

7. **Drop the tables and database**

```
mysql> drop table Student;
Query OK, 0 rows affected (0.00 sec)

mysql> drop table Course;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
Empty set (0.00 sec)

mysql> drop database colleges;
Query OK, 0 rows affected (0.00 sec)
```

8. **Truncate student table.**

```
+---------+----------+------------+------------+---------+-------+-------------+
| Roll_no | Std_name | DOB        | House_name | Place   | Ph_no | Blood_group |
+---------+----------+------------+------------+---------+-------+-------------+
|       1 | rahul    | 2001-01-02 | hridhyam   | varanad | 85794 | b+ve        |
+---------+----------+------------+------------+---------+-------+-------------+
1 row in set (0.02 sec)

mysql> truncate table Student;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from Student;
Empty set (0.00 sec)
```

9. **Rename student table**

```
mysql> rename table Student to Student_details;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+--------------------+
| Tables_in_colleges |
+--------------------+
| student_details    |
+--------------------+
1 row in set (0.00 sec)
```

Constraints

1.Create a table – Person ,with id(PK),Name(not null),Aadhar no(not null and unique) ,Age(only individuals who are 19 years old or above can have their data entered or updated in the system)

```
mysql> create database Orders;
Query OK, 1 row affected (0.00 sec)

mysql> use Orders;
Database changed
mysql> create table Person(Person_id int primary key,Name varchar(20)
    -> not null,Adhar_no int not null unique,Age int,check(Age>18));
Query OK, 0 rows affected (0.05 sec)

mysql> show tables;
+------------------+
| Tables_in_orders |
+------------------+
| person           |
+------------------+
1 row in set (0.00 sec)
```

2. Create a table-Orders with ID(PK),Order no(not null),set foreign key with Person,item attribute should have default value as Bag

```
mysql> create table Orders(Order_id int primary key,Order_no int
    -> not null,Person_id int,foreign key(Person_id)references
    -> Person(Person_id));
Query OK, 0 rows affected (0.04 sec)

mysql> show tables;
+------------------+
| Tables_in_orders |
+------------------+
| orders           |
| person           |
+------------------+
2 rows in set (0.00 sec)
```

3. Describe the structure of Person and Orders

```
mysql> desc Person;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| Person_id | int(11)     | NO   | PRI | NULL    |       |
| Name      | varchar(20) | NO   |     | NULL    |       |
| Adhar_no  | int(11)     | NO   | UNI | NULL    |       |
| Age       | int(11)     | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> desc Orders;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| Order_id  | int(11) | NO   | PRI | NULL    |       |
| Order_no  | int(11) | NO   |     | NULL    |       |
| Person_id | int(11) | YES  | MUL | NULL    |       |
+-----------+---------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Program 2 -Dml commands

1. **Create database and two table- Employee and Department with suitable attributes.**

```
mysql> create database organizations;
Query OK, 1 row affected (0.03 sec)

mysql> use organizations;
Database changed
mysql> 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,
    -> designation varchar(20));
Query OK, 0 rows affected (0.09 sec)

mysql> create table department(deptno int primary key,dept_name varchar(20),location varchar(20));
Query OK, 0 rows affected (0.07 sec)

mysql> show tables;
+--------------------------+
| Tables_in_organizations |
+--------------------------+
| department               |
| employee                 |
+--------------------------+
2 rows in set (0.14 sec)
```

2. **Insert values into the table Employee**

```
mysql> insert into employee values(101,"john",
    -> "1990-02-12","no123",123456,1,10000,"professor");
Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values(102,"mathew",
    -> "1990-03-12","no124",123356,1,10000,"professor");
Query OK, 1 row affected (0.00 sec)

mysql> insert into employee values(103,"preethi",
    -> "1990-01-12","no134",123256,2,7000,"clerk");
Query OK, 1 row affected (0.00 sec)
```

3. **Display the structure of Employee Table**

```
mysql> insert into employee values(105,"anny","1932-04-11","no120",132454,4,3000,"peon");
Query OK, 1 row affected (0.00 sec)

mysql> select * from employee;
+--------+----------+------------+----------+-----------+---------+--------+-------------+
| emp_no | emp_name | dob        | address  | mobile_no | dept_no | salary | designation |
+--------+----------+------------+----------+-----------+---------+--------+-------------+
|    101 | john     | 1990-02-12 | no123    |    123456 |       1 |  10000 | professor   |
|    102 | mathew   | 1990-03-12 | no124    |    123356 |       1 |  10000 | professor   |
|    103 | preethi  | 1990-01-12 | no134    |    123256 |       2 |   7000 | clerk       |
|    104 | rahma    | 1990-01-12 | no534    |    123256 |       2 |   7000 | clerk       |
|    105 | anny     | 1932-04-11 | no120    |    132454 |       4 |   3000 | peon        |
+--------+----------+------------+----------+-----------+---------+--------+-------------+
5 rows in set (0.00 sec)
```

4. **Insert values into table Department and display all records from the table.**

```
mysql> insert into department values(1,"mca","block_c");
Query OK, 1 row affected (0.00 sec)

mysql> insert into department values(2,"mechanical","block_d");
Query OK, 1 row affected (0.00 sec)

mysql> insert into department values(3,"civil","block_a");
Query OK, 1 row affected (0.00 sec)

mysql> insert into department values(4,"barch","block_b");
Query OK, 1 row affected (0.00 sec)

mysql> insert into department values(5,"eee","block_f");
Query OK, 1 row affected (0.00 sec)

mysql> select * from department;
+--------+------------+----------+
| deptno | dept_name  | 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 (0.00 sec)
```

5. **Display the employee number and name of whose department number is 2. Display employee number, name, department number, designation, salary of employees in descending order of salary. Display the employee number, name of the employee whose salary is between 2000 and 3000.**

```
mysql> select emp_no,emp_name from employee where dept_no=2;
+--------+----------+
| emp_no | emp_name |
+--------+----------+
|    103 | preethi  |
|    104 | rahma    |
+--------+----------+
2 rows in set (0.02 sec)

mysql> select emp_no,emp_name,dept_no,salary,designation
    -> from employee order by salary desc;
+--------+----------+---------+--------+-------------+
| emp_no | emp_name | dept_no | salary | designation |
+--------+----------+---------+--------+-------------+
|    101 | john     |       1 |  10000 | professor   |
|    102 | mathew   |       1 |  10000 | professor   |
|    103 | preethi  |       2 |   7000 | clerk       |
|    104 | rahma    |       2 |   7000 | clerk       |
|    105 | anny     |       4 |   3000 | peon        |
+--------+----------+---------+--------+-------------+
5 rows in set (0.00 sec)

mysql> select emp_no,emp_name from employee
    -> where salary between "2000"and"5000";
+--------+----------+
| emp_no | emp_name |
+--------+----------+
|    105 | anny     |
+--------+----------+
1 row in set (0.03 sec)
```

6. **Display designation from employee without duplicate values**

```
mysql> select designation from employee group by designation;
+-------------+
| designation |
+-------------+
| clerk       |
| peon        |
| professor   |
+-------------+
3 rows in set (0.00 sec)
```

7. **Change the salary of employees to 45000 whose designation is manager**

```
mysql> update employee set salary="45000" where designation="manager";
Query OK, 0 rows affected (0.01 sec)
Rows matched: 0  Changed: 0  Warnings: 0
```

8. **Display all records from the table employees**
   Change mobile number of employees named John
   Display all records from table employees

```
mysql> select * from employee;
+--------+----------+------------+---------+-----------+---------+--------+-------------+
| emp_no | emp_name | dob        | address | mobile_no | dept_no | salary | designation |
+--------+----------+------------+---------+-----------+---------+--------+-------------+
|    101 | john     | 1990-02-12 | no123   |    123456 |       1 |  10000 | professor   |
|    102 | mathew   | 1990-03-12 | no124   |    123356 |       1 |  10000 | professor   |
|    103 | preethi  | 1990-01-12 | no134   |    123256 |       2 |  45000 | clerk       |
|    104 | rahma    | 1990-01-12 | no534   |    123256 |       2 |  45000 | clerk       |
|    105 | anny     | 1932-04-11 | no120   |    132454 |       4 |   3000 | peon        |
+--------+----------+------------+---------+-----------+---------+--------+-------------+
5 rows in set (0.00 sec)

mysql> update employee set mobile_no="987654"
    -> where emp_name="john";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from employee;
+--------+----------+------------+---------+-----------+---------+--------+-------------+
| emp_no | emp_name | dob        | address | mobile_no | dept_no | salary | designation |
+--------+----------+------------+---------+-----------+---------+--------+-------------+
|    101 | john     | 1990-02-12 | no123   |    987654 |       1 |  10000 | professor   |
|    102 | mathew   | 1990-03-12 | no124   |    123356 |       1 |  10000 | professor   |
|    103 | preethi  | 1990-01-12 | no134   |    123256 |       2 |  45000 | clerk       |
|    104 | rahma    | 1990-01-12 | no534   |    123256 |       2 |  45000 | clerk       |
|    105 | anny     | 1932-04-11 | no120   |    132454 |       4 |   3000 | peon        |
+--------+----------+------------+---------+-----------+---------+--------+-------------+
5 rows in set (0.00 sec)
```

9. **Delete all employees whose salary is equal to Rs 300. Retrieve the name,mobile number of all employees whose name start with "a","m". Retrieve the emp_no,name,salary of all employees working as "peon","clerk".**

```
mysql> delete from employee where salary="3000";
Query OK, 1 row affected (0.03 sec)

mysql> select * from employee;
+--------+----------+------------+---------+-----------+---------+--------+-------------+
| emp_no | emp_name | dob        | address | mobile_no | dept_no | salary | designation |
+--------+----------+------------+---------+-----------+---------+--------+-------------+
|    101 | john     | 1990-02-12 | no123   |    987654 |       1 |  10000 | professor   |
|    102 | mathew   | 1990-03-12 | no124   |    123356 |       1 |  10000 | professor   |
|    103 | preethi  | 1990-01-12 | no134   |    123256 |       2 |  45000 | clerk       |
|    104 | rahma    | 1990-01-12 | no534   |    123256 |       2 |  45000 | clerk       |
+--------+----------+------------+---------+-----------+---------+--------+-------------+
4 rows in set (0.00 sec)

mysql> select emp_name,mobile_no from employee where emp_name like"a%";
Empty set (0.00 sec)

mysql> select emp_name,mobile_no from employee where emp_name like"m%";
+----------+-----------+
| emp_name | mobile_no |
+----------+-----------+
| mathew   |    123356 |
+----------+-----------+
1 row in set (0.00 sec)

mysql> select emp_no,emp_name,salary
    -> from employee
    -> where designation="peon"
    -> or designation="clerk";
+--------+----------+--------+
| emp_no | emp_name | salary |
+--------+----------+--------+
|    103 | preethi  |  45000 |
|    104 | rahma    |  45000 |
+--------+----------+--------+
2 rows in set (0.00 sec)
```

Program 3 Dcl commands

1. **Create a database and use that database.**
   Create a table

```
C:\wamp\bin\mysql\mysql5.7.14\bin>mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> create database Colleges;
Query OK, 1 row affected (0.00 sec)

mysql> use Colleges;
Database changed
mysql> create table Student(Std_id int primary key,Std_name varchar(20));
Query OK, 0 rows affected (0.05 sec)
```

2. **Display table**

```
mysql> desc Student;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| Std_id   | int(11)     | NO   | PRI | NULL    |       |
| Std_name | varchar(20) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

3. **Create a user "schoolteacher" identified by "@teacherss755" and GRANT select,insert,update**

```
mysql> create user schoolteacher identified by '@teacherss755';
Query OK, 0 rows affected (0.02 sec)

mysql> grant select,insert,update on Student to schoolteacher;
Query OK, 0 rows affected (0.06 sec)

mysql> quit
Bye
```

4. **Login as user and use database**
   Show tables

```
C:\wamp\bin\mysql\mysql5.7.14\bin>mysql -u schoolteacher -p
Enter password: **************
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use Colleges;
Database changed
mysql> show tables;
+--------------------+
| Tables_in_colleges |
+--------------------+
| student            |
+--------------------+
1 row in set (0.00 sec)
```

5. **Insert values into table**

```
mysql> insert into Student(Std_id,Std_name)values(1,'Rahul'),
    -> (2,'Ziya'),(3,'Diya');
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

6. **Update student name where std_id = 1**

Delete student name where std_id = 1

```
mysql> select * from Student;
+--------+----------+
| Std_id | Std_name |
+--------+----------+
|      1 | Rahul    |
|      2 | Ziya     |
|      3 | Diya     |
+--------+----------+
3 rows in set (0.00 sec)

mysql> update Student set Std_name = 'Ciya' where Std_id = 1;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Student;
+--------+----------+
| Std_id | Std_name |
+--------+----------+
|      1 | Ciya     |
|      2 | Ziya     |
|      3 | Diya     |
+--------+----------+
3 rows in set (0.00 sec)

mysql> delete from Student where Std_id = 1;
ERROR 1142 (42000): DELETE command denied to user 'schoolteacher'@'localhost' fo
r table 'student'
mysql> quit
Bye
```

7. Grant delete to user

```
C:\wamp\bin\mysql\mysql5.7.14\bin>mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use Colleges;
Database changed
mysql> grant delete on Student to schoolteacher;
Query OK, 0 rows affected (0.00 sec)

mysql> quit
Bye

C:\wamp\bin\mysql\mysql5.7.14\bin>
```

8. Delete values and display tables

```
C:\wamp\bin\mysql\mysql5.7.14\bin>mysql -u schoolteacher -p
Enter password: **************
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use Colleges;
Database changed
mysql> select * from Student;
+--------+----------+
| Std_id | Std_name |
+--------+----------+
|      1 | Ciya     |
|      2 | Ziya     |
|      3 | Diya     |
+--------+----------+
3 rows in set (0.00 sec)

mysql> delete from Student where Std_id = 1;
Query OK, 1 row affected (0.03 sec)

mysql> select * from Student;
+--------+----------+
| Std_id | Std_name |
+--------+----------+
|      2 | Ziya     |
|      3 | Diya     |
+--------+----------+
2 rows in set (0.00 sec)

mysql> quit
Bye
```

9. Revoke insert, update and delete from user

```
C:\wamp\bin\mysql\mysql5.7.14\bin>mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use Colleges;
Database changed
mysql> revoke insert,delete on Student from schoolteacher;
Query OK, 0 rows affected (0.03 sec)

mysql> quit
Bye
```

10. Insert values to the table

Delete student name where std_id = 1

```
C:\wamp\bin\mysql\mysql5.7.14\bin>mysql -u schoolteacher -p
Enter password: **************
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use Colleges;
Database changed
mysql> select * from Student;
+--------+----------+
| Std_id | Std_name |
+--------+----------+
|      2 | Ziya     |
|      3 | Diya     |
+--------+----------+
2 rows in set (0.00 sec)

mysql> insert into Student(Std_id,Std_name)values(1,'Miya');
ERROR 1142 (42000): INSERT command denied to user 'schoolteacher'@'localhost' fo
r table 'student'
mysql> delete from Student where Std_id = 2;
ERROR 1142 (42000): DELETE command denied to user 'schoolteacher'@'localhost' fo
r table 'student'
mysql> quit
Bye
```

Prgm 4 Tcl commands

1. Create a table MCA students and insert values into table

```
SQL> insert into MCAstudents
  2  values(4, 'Georgia','F',2500);

1 row created.

SQL> insert into MCAstudents
  2  values(5, 'Veronica','F',1500);

1 row created.

SQL> select * from MCAstudents;

        ID NAME                 GENDER        STIPEND
---------- -------------------- ---------- ----------
         1 Alex                 M                1500
         2 Denny                M                2000
         3 Eva                  F                1500
         4 Georgia              F                2500
         5 Veronica             F                1500

SQL>
```

```
SQL> create table MCAstudents(
  2  ID int primary key,
  3  Name varchar(20),
  4  Gender varchar(10),
  5  Stipend int);

Table created.

SQL> insert into MCAstudents
  2  values(1, 'Alex','M',1500);

1 row created.

SQL> insert into MCAstudents
  2  values(2, 'Denny','M',2000);

1 row created.

SQL> insert into MCAstudents
  2  values(3, 'Eva','F',1500);

1 row created.
```

2. **Update the stipend of the student where name= Eva**
3. **Delete a row where stipend =2000**
4. **Perform commit**

```
SQL> update MCAstudents set Stipend=3000
  2  where Name='Eva';

1 row updated.

SQL> select * from MCAstudents;

        ID NAME                 GENDER        STIPEND
---------- -------------------- ---------- ----------
         1 Alex                 M                1500
         2 Denny                M                2000
         3 Eva                  F                3000
         4 Georgia              F                2500
         5 Veronica             F                1500

SQL> delete from MCAstudents where Stipend=2000;

1 row deleted.

SQL> select * from MCAstudents;

        ID NAME                 GENDER        STIPEND
---------- -------------------- ---------- ----------
         1 Alex                 M                1500
         3 Eva                  F                3000
         4 Georgia              F                2500
         5 Veronica             F                1500

SQL> commit;

Commit complete.
```

5. **Delete a row where name = Georiga**
6. **Delete the ttable**
7. **Perform rollback**

```
SQL> delete from MCAstudents where Name='Georgia';

1 row deleted.

SQL> select * from MCAstudents;

        ID NAME                 GENDER        STIPEND
---------- -------------------- ---------- ----------
         1 Alex                 M                1500
         3 Eva                  F                3000
         5 Veronica             F                1500

SQL> delete from MCAstudents;

3 rows deleted.

SQL> select * from MCAstudents;

no rows selected

SQL> rollback;

Rollback complete.

SQL> select * from MCAstudents;

        ID NAME                 GENDER        STIPEND
---------- -------------------- ---------- ----------
         1 Alex                 M                1500
         3 Eva                  F                3000
         4 Georgia              F                2500
         5 Veronica             F                1500
```

Prgm 5 join

Inner join

1.Retrieve all employees along with their department names (show NULL if no department is assigned).

```
mysql> use organisation;
Database changed
mysql> select * from Department1;
+---------+------------+
| Dept_id | Dept_name  |
+---------+------------+
| D1      | NULL       |
| D2      | Sales      |
| D3      | Marketing  |
| D4      | Finance    |
| D5      | Executive  |
+---------+------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM employee;
+--------+--------------+---------+--------+
| Emp_id | Emp_name     | Dept_id | salary |
+--------+--------------+---------+--------+
|    101 | NULL         | D1      |  32000 |
|    102 | Manoj        | D2      |  35000 |
|    103 | Geetha       | D3      |  35000 |
|    104 | NULL         | D4      |  36000 |
|    105 | Geethanjali  | D5      |  50000 |
+--------+--------------+---------+--------+
5 rows in set (0.00 sec)

mysql> select employee.Emp_id,employee.Emp_name,Department1.Dept_name
    -> FROM employee JOIN Department1 ON employee.Dept_id=Department1.Dept_id;
+--------+--------------+------------+
| Emp_id | Emp_name     | Dept_name  |
+--------+--------------+------------+
|    101 | NULL         | NULL       |
|    102 | Manoj        | Sales      |
|    103 | Geetha       | Marketing  |
|    104 | NULL         | Finance    |
|    105 | Geethanjali  | Executive  |
+--------+--------------+------------+
5 rows in set (0.02 sec)
```

2.Retrieve employees earning more than 50,000 along with their department names.

3. Find the total number of employees in each department.

```
mysql> use organisation;
Database changed
mysql> select employee.Emp_id,employee.Emp_name,employee.salary,
    -> Department1.Dept_name FROM employee INNER JOIN Department1 ON
    -> employee.Dept_id=Department1.Dept_id WHERE employee.salary>35000;
+--------+--------------+--------+------------+
| Emp_id | Emp_name     | salary | Dept_name  |
+--------+--------------+--------+------------+
|    104 | NULL         |  36000 | Marketing  |
|    105 | Geethanjali  |  50000 | Executive  |
+--------+--------------+--------+------------+
2 rows in set (0.00 sec)

mysql> select Department1.Dept_name,COUNT(employee.Emp_id)
    -> as totalemployees from employee INNER JOIN
    -> Department1 on employee.Dept_id=Department1.Dept_id
    -> GROUP BY Department1.Dept_name;
+------------+-----------------+
| Dept_name  | totalemployees  |
+------------+-----------------+
| NULL       |               1 |
| Executive  |               1 |
| Marketing  |               2 |
| Sales      |               1 |
+------------+-----------------+
4 rows in set (0.04 sec)
```

Left join

1.Retrieve all employees along with their department names (show NULL if no department is assigned).
2. Retrieve employees who are not assigned to any department.

```
mysql> select employee.Emp_id,employee.Emp_name,Department1.Dept_name
    -> from employee LEFT JOIN Department1 on
    -> employee.Dept_id=Department1.Dept_id;
+--------+-------------+-----------+
| Emp_id | Emp_name    | Dept_name |
+--------+-------------+-----------+
|    101 | NULL        | NULL      |
|    102 | Manoj       | Sales     |
|    103 | Geetha      | Marketing |
|    104 | NULL        | Marketing |
|    105 | Geethanjali | Executive |
+--------+-------------+-----------+
5 rows in set (0.02 sec)

mysql> select employee.Emp_id,employee.Emp_name from employee
    -> LEFT JOIN Department1 on employee.Dept_id=Department1.Dept_id
    -> where Department1.Dept_id is NULL;
Empty set (0.03 sec)

mysql> select employee.Emp_id,employee.Emp_name from employee
    -> LEFT JOIN Department1 on employee.Dept_id=Department1.Dept_id
    -> where Department1.Dept_name is NULL;
+--------+----------+
| Emp_id | Emp_name |
+--------+----------+
|    101 | NULL     |
+--------+----------+
1 row in set (0.00 sec)
```

3. Retrieve all employees and replace NULL department names with 'Not Assigned'.

```
mysql> select employee.Emp_id,employee.Emp_name,
    -> COALESCE(department1.Dept_name,'Not Assigned')
    -> as Department_name from employee LEFT
    -> JOIN department1 on employee.Dept_id=department1.Dept_id;
+--------+-------------+-----------------+
| Emp_id | Emp_name    | Department_name |
+--------+-------------+-----------------+
|    101 | NULL        | Not Assigned    |
|    102 | Manoj       | Sales           |
|    103 | Geetha      | Marketing       |
|    104 | NULL        | Marketing       |
|    105 | Geethanjali | Executive       |
+--------+-------------+-----------------+
5 rows in set (0.04 sec)
```

RIGHT JOIN

1.Retrieve all departments along with employees working in them (show NULL if no employees are assigned).

```
mysql> select employee.Emp_id,employee.Emp_name,Department1.Dept_name
    -> from employee RIGHT JOIN Department1 on
    -> employee.Dept_id=Department1.Dept_id;
+--------+-------------+-----------+
| Emp_id | Emp_name    | Dept_name |
+--------+-------------+-----------+
|    101 | NULL        | NULL      |
|    102 | Manoj       | Sales     |
|    103 | Geetha      | Marketing |
|    104 | NULL        | Marketing |
|    105 | Geethanjali | Executive |
+--------+-------------+-----------+
5 rows in set (0.00 sec)

mysql> select Department1.Dept_id,Department1.Dept_name
    -> from employee RIGHT JOIN Department1 on
    -> employee.Dept_id=Department1.Dept_id
    -> where employee.Emp_name is NULL;
+---------+-----------+
| Dept_id | Dept_name |
+---------+-----------+
| D1      | NULL      |
| D4      | Marketing |
+---------+-----------+
2 rows in set (0.00 sec)
```

2. Retrieve all departments and count the number of employees in each (including departments with zero employees)
3. Find departments that have no employees assigned.

```
mysql> select Department1.Dept_name,COUNT(employee.Emp_id)
    -> as totalemployees from employee RIGHT JOIN
    -> Department1 on employee.Dept_id=Department1.Dept_id
    -> GROUP BY Department1.Dept_name;
+-----------+-----------------+
| Dept_name | totalemployees  |
+-----------+-----------------+
| NULL      |               1 |
| Executive |               1 |
| Marketing |               2 |
| Sales     |               1 |
+-----------+-----------------+
4 rows in set (0.00 sec)

mysql> select Department1.Dept_id,Department1.Dept_name
    -> from employee RIGHT JOIN Department1 on
    -> employee.Dept_id=Department1.Dept_id
    -> where employee.Emp_name is NULL;
+---------+-----------+
| Dept_id | Dept_name |
+---------+-----------+
| D1      | NULL      |
| D4      | Marketing |
+---------+-----------+
2 rows in set (0.00 sec)
```

Prgm 6 aggregate

Table 1: Employees

```
mysql> use organisation;
Database changed
mysql> create table employees(Emp_id int primary key,
    -> Emp_name varchar(20),Dept_name varchar(20),
    -> Salary int);
Query OK, 0 rows affected (0.23 sec)

mysql> insert into employees values
    -> (101,"Alice Johnson","Sales",35000),
    -> (102,"Bob Smith","Sales",36000),
    -> (103,"Charlie Brown","Marketing",37000),
    -> (104,"David Willson","IT",38000),
    -> (105,"Eva Green","HR",45000),
    -> (106,"Ian Black","Finance",40000);
Query OK, 6 rows affected (0.12 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

Table 2: Sales

```
mysql> create table Sales(Sale_id int primary key,
    -> Emp_id int,Sale_date date,Sale_Amt int,
    -> foreign key(Emp_id)references employees
    -> (Emp_id));
Query OK, 0 rows affected (0.07 sec)

mysql> insert into Sales values
    -> (1,101,'2022-01-12',15000),
    -> (2,101,'2022-02-11',20000),
    -> (3,102,'2022-01-08',20000),
    -> (4,103,'2022-03-07',21000),
    -> (5,104,'2022-03-06',11000),
    -> (6,105,'2022-02-05',12000),
    -> (7,106,'2022-07-07',18000);
Query OK, 7 rows affected (0.11 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

Show Tables

```
mysql> select * from employees;
+--------+---------------+-----------+--------+
| Emp_id | Emp_name      | Dept_name | Salary |
+--------+---------------+-----------+--------+
|    101 | Alice Johnson | Sales     |  35000 |
|    102 | Bob Smith     | Sales     |  36000 |
|    103 | Charlie Brown | Marketing |  37000 |
|    104 | David Willson | IT        |  38000 |
|    105 | Eva Green     | HR        |  45000 |
|    106 | Ian Black     | Finance   |  40000 |
+--------+---------------+-----------+--------+
6 rows in set (0.00 sec)

mysql> select * from Sales;
+---------+--------+------------+----------+
| Sale_id | Emp_id | Sale_date  | Sale_Amt |
+---------+--------+------------+----------+
|       1 |    101 | 2022-01-12 |    15000 |
|       2 |    101 | 2022-02-11 |    20000 |
|       3 |    102 | 2022-01-08 |    20000 |
|       4 |    103 | 2022-03-07 |    21000 |
|       5 |    104 | 2022-03-06 |    11000 |
|       6 |    105 | 2022-02-05 |    12000 |
|       7 |    106 | 2022-07-07 |    18000 |
+---------+--------+------------+----------+
7 rows in set (0.00 sec)
```

1. What is the average salary of employees in the Sales department?
2. What is the total sale amount for each employee?

```
mysql> select AVG(Salary) as Average_Salary
    -> from employees where Dept_name="Sales";
+----------------+
| Average_Salary |
+----------------+
|     35500.0000 |
+----------------+
1 row in set (0.04 sec)

mysql> select Emp_id, SUM(Sale_Amt) as
    -> Total_Sale_Amount from sales
    -> group by Emp_id;
+--------+--------------------+
| Emp_id | Total_Sale_Amount  |
+--------+--------------------+
|    101 |              35000 |
|    102 |              20000 |
|    103 |              21000 |
|    104 |              11000 |
|    105 |              12000 |
|    106 |              18000 |
+--------+--------------------+
6 rows in set (0.08 sec)
```

3. What is the maximum sale amount for each department?
4. What is the minimum salary of employees in the Marketing department?

```
mysql> select Dept_name,MAX(Sale_Amt) as
    -> Max_Sale_Amount from Sales join
    -> employees on Sales.Emp_id=employees.Emp_id
    -> group by Dept_name;
+-----------+------------------+
| Dept_name | Max_Sale_Amount  |
+-----------+------------------+
| Finance   |            18000 |
| HR        |            12000 |
| IT        |            11000 |
| Marketing |            21000 |
| Sales     |            20000 |
+-----------+------------------+
5 rows in set (0.03 sec)

mysql> select MIN(Salary) from employees
    -> where Dept_name="Marketing";
+-------------+
| MIN(Salary) |
+-------------+
|       37000 |
+-------------+
1 row in set (0.00 sec)
```

5. What is the total salary of all employees?
6. What is the average sale amount for each quarter of the year?

```
mysql> select SUM(Salary) as Total_Salary
    -> from employees;
+--------------+
| Total_Salary |
+--------------+
|       231000 |
+--------------+
1 row in set (0.00 sec)

mysql> select QUARTER(Sale_date) as
    -> Quarter, AVG(Sale_Amt) as
    -> Avg_Sale_Amount from Sales
    -> group by Quarter;
+---------+-----------------+
| Quarter | Avg_Sale_Amount |
+---------+-----------------+
|       1 |      16500.0000 |
|       3 |      18000.0000 |
+---------+-----------------+
2 rows in set (0.02 sec)
```

7. What is the count of employees in each department?
8. What is the highest salary earned by an employee in the IT department?

```
mysql> select Dept_name,COUNT(*) as
    -> Employees_Count from employees
    -> group by Dept_name;
+-----------+-----------------+
| Dept_name | Employees_Count |
+-----------+-----------------+
| Finance   |               1 |
| HR        |               1 |
| IT        |               1 |
| Marketing |               1 |
| Sales     |               2 |
+-----------+-----------------+
5 rows in set (0.02 sec)

mysql> select MAX(Salary) as Max_Salary
    -> from employees where Dept_name
    -> ="IT";
+------------+
| Max_Salary |
+------------+
|      38000 |
+------------+
1 row in set (0.00 sec)
```

9. How many sales were made by employees in the Sales department in the year 2022?

10. What is the total sale amount for each department, and which department has the highest total sale amount?

```
mysql> select COUNT(*) as Sale_Count
    -> from Sales join employees on
    -> Sales.Emp_id=employees.Emp_id
    -> where employees.Dept_name="Sales"
    -> AND YEAR(Sales.Sale_date)=2022;
+------------+
| Sale_Count |
+------------+
|          3 |
+------------+
1 row in set (0.02 sec)

mysql> select employees.Dept_name,
    -> SUM(Sales.Sale_Amt) as
    -> Total_Sale_Amount from
    -> employees join Sales on
    -> employees.Emp_id=Sales.emp_id
    -> group by employees.Dept_name
    -> order by Total_Sale_Amount DESC;
+-----------+-------------------+
| Dept_name | Total_Sale_Amount |
+-----------+-------------------+
| Sales     |             55000 |
| Marketing |             21000 |
| Finance   |             18000 |
| HR        |             12000 |
| IT        |             11000 |
+-----------+-------------------+
5 rows in set (0.02 sec)
```

Prgm no 8 sum and average

```
SQL> start C:\Users\CAD3\Desktop\sum.sql
Enter value for a: 5
old   2:  a number:=&a;
new   2:  a number:=5;
Enter value for b: 6
old   3:  b number:=&b;
new   3:  b number:=6;
Enter value for c: 6
old   4:  c number:=&c;
new   4:  c number:=6;
sum=17
average=5.66666666666666666666666666666666666667

PL/SQL procedure successfully completed.
```

pgrm no 9 simple interest

```
SQL> set serveroutput on;
SQL> start C:\Users\CAD3\Desktop\dbms\simple.sql
Enter value for p: 150
old   7: p:=&p;
new   7: p:=150;
Enter value for r: 4
old   8: r:=&r;
new   8: r:=4;
Enter value for t: 5
old   9: t:=&t;
new   9: t:=5;
Simple Interest = 30

PL/SQL procedure successfully completed.
```

Prgm 10 area

```
SQL> start C:\Users\CAD3\Desktop\dbms\circleradius.sql

Table created.


PL/SQL procedure successfully completed.

SQL> select * from areas;

    RADIUS       AREA
---------- ----------
         3      28.26
         4      50.24
         5       78.5
         6     113.04
         7     153.86
```

Pgrm 11 factorial

```
SQL> start C:\Users\CAD3\Desktop\dbms\factorial.sql
Enter value for n: 25
old   6: n:=&n;
new   6: n:=25;
25! = 15511210043330985984000000

PL/SQL procedure successfully completed.

SQL>
```

Prgm 12 reverse

```
SQL> start C:\Users\CAD3\Desktop\dbms\Reversenum.sql
Enter value for n: 64
old   7: N := &N;
new   7: N := 64;
THE REVERSED DIGITS OF 64 = 46

PL/SQL procedure successfully completed.

SQL>
```

Prgm 13 greatest

```
SQL> set serveroutput on;
SQL> edit greatnum.sql

SQL> start C:\Users\CAD3\Desktop\dbms\greatnum.sql
Enter value for a: 15
old   2: a number := &a;
new   2: a number := 15;
Enter value for b: 13
old   3: b number := &b;
new   3: b number := 13;
Enter value for c: 45
old   4: c number := &c;
new   4: c number := 45;
45 is greatest.

PL/SQL procedure successfully completed.
```

Prgm14 fibonacci

```
SQL> start C:\Users\CAD3\Desktop\dbms\fibonacci.sql
0 1 1 2 3 5 8 13 21 34

PL/SQL procedure successfully completed.

SQL>
```

Prgm 15 sum of digits

```
SQL> set serveroutput on;
SQL> edit sumnum.sql

SQL> start C:\Users\CAD3\Desktop\dbms\sumnum.sql
Enter value for n: 3456
old   6: N:=&N;
new   6: N:=3456;
THE SUM OF THE DIGITS = 18

PL/SQL procedure successfully completed.

SQL>
```

Prgm 17

Procedure program to insert values into student table

```
SQL> create table CEStudent(ID number(10) primary key,
  2  name varchar(30), Mark number(30));

Table created.

SQL> create or replace procedure insertuser(ID in number,
  2  name in varchar2, Mark in number)
  3  is
  4  begin
  5  insert into CEStudent values(ID,Name,Mark);
  6  end;
  7  /

Procedure created.

SQL> call insertuser (1,'Aleena',78);

Call completed.

SQL> call insertuser (2,'Alex',82);

Call completed.
```

```
SQL> call insertuser (3,'George',75);

Call completed.

SQL> call insertuser (4,'Harish',73);

Call completed.

SQL> call insertuser (5,'John',69);

Call completed.

SQL> select * from CEStudent;

        ID NAME                                 MARK
---------- ------------------------------ ----------
         1 Aleena                                 78
         2 Alex                                   82
         3 George                                 75
         4 Harish                                 73
         5 John                                   69

SQL>
```

Prgm 18

Procedure program to find a maximum of two number.

```
SQL> create or replace procedure
  2  max(a in number,b in number,
  3  max_value out number) is
  4  begin
  5   if a>b then
  6     max_value:=a;
  7   else
  8     max_value:=b;
  9   end if;
 10  end;
 11  /

Procedure created.

SQL> start C:\Users\CAD3\Desktop\dbms\largestnum.sql;
Enter value for a: 5
old   2:   num1 number:=&a;
new   2:   num1 number:=5;
Enter value for b: 8
old   3:   num2 number:=&b;
new   3:   num2 number:=8;
The maximum value is: 8

PL/SQL procedure successfully completed.
```

Prgm19

Implement function to find maximum salary from employee table

```
SQL> create table Employees(Emp_id int primary key,
  2  name varchar(20),
  3  salary int);

Table created.

SQL> insert into Employees
  2  values(1,'Ankit',30000);

1 row created.

SQL> insert into Employees
  2  values(2,'Maria',40000);

1 row created.

SQL> insert into Employees
  2  values(3,'Ram',25000);

1 row created.
```

```
SQL> create or replace function
  2  MaxSal return number is
  3  Maximum number:=0;
  4  begin
  5  select max(salary) into maximum from Employees;
  6  return Maximum;
  7  end;
  8  /

Function created.

SQL> set serveroutput on;
SQL> edit emptable.sql;

SQL> start C:\Users\CAD3\Desktop\dbms\emptable.sql;
Maximum salary=40000

PL/SQL procedure successfully completed.
```

Prgm 20

Familiarise after insert,delete and update trigger.

```
SQL> create table empl (eid int,ename varchar(30),
  2  deptname varchar(30),deptid int);

Table created.

SQL> insert into empl values(101,'raman','electronics',201);

1 row created.

SQL> insert into empl values(102,'kumar','computer',202);

1 row created.

SQL> insert into empl values(103,'unni','mechanic',203);

1 row created.

SQL> insert into empl values(104,'hari','mechanic',204);

1 row created.

SQL> create table trig_logg (logs varchar(200));

Table created.
```

```
SQL> select * from empl;

       EID ENAME                          DEPTNAME
---------- ------------------------------ ------------------------------
    DEPTID
----------
       101 raman                          electronics
       201

       102 kumar                          computer
       202

       103 unni                           mechanic
       203


       EID ENAME                          DEPTNAME
---------- ------------------------------ ------------------------------
    DEPTID
----------
       104 hari                           mechanic
       204
```

```
SQL> select * from trig_logg;

no rows selected

SQL> create trigger trig_empl_insert after insert on empl
  2  for each row
  3  begin
  4  insert into trig_logg values('A new row in empl table has been created');
  5  end;
  6  /

Trigger created.

SQL> insert into empl values (110,'Jeevan','MCA',309);

1 row created.

SQL> select * from trig_logg;

LOGS
--------------------------------------------------------------------------------
A new row in empl table has been created

SQL> create trigger trig_empl_update after update on empl
  2  for each row
  3  begin
  4  insert into trig_logg values('a new row in empl table has been updated');
  5  end;
  6  /

Trigger created.

SQL> update empl set ename='Niranjan' where ename='kumar';

1 row updated.

SQL> select * from trig_logg;

LOGS
--------------------------------------------------------------------------------
A new row in empl table has been created
a new row in empl table has been updated
```

```
SQL> create trigger trig_empl_delete after delete empl
  2  for each row
  3  begin
  4  insert into trig_logg values('a new row in empl table has been deleted');
  5  end;
  6  /
create trigger trig_empl_delete after delete empl
                                        *
ERROR at line 1:
ORA-00969: missing ON keyword


SQL> create trigger trig_empl_delete after delete on empl
  2  for each row
  3  begin
  4  insert into trig_logg values('a new row in empl table has been deleted');
  5  end;
  6  /

Trigger created.

SQL> delete from empl where eid=110;

1 row deleted.

SQL> select * from trig_logg;

LOGS
--------------------------------------------------------------------------------
A new row in empl table has been created
a new row in empl table has been updated
a new row in empl table has been deleted
```

Prgm 21

Familiarise old and new command in trigger

```
SQL> select * from emps;

no rows selected

SQL> select * from empl;

       EID ENAME                          DEPTNAME
---------- ------------------------------ ------------------------------
    DEPTID
----------
       101 raman                          electronics
       201

       102 Niranjan                       computer
       202

       103 unni                           mechanic
       203


       EID ENAME                          DEPTNAME
---------- ------------------------------ ------------------------------
    DEPTID
----------
       104 hari                           mechanic
       204
```

```
SQL> insert into empl values(109,'mohan','electronics',209);

1 row created.

SQL> select * from trig_logg;

LOGS
--------------------------------------------------------------------------------
A new row in empl table has been created
a new row in empl table has been updated
a new row in empl table has been deleted
A new row in empl table has been created
```

```
SQL> create trigger trig_empl_updates after update on empl
  2  for each row
  3  begin
  4  insert into trig_logg values(concat('you have updated an emloyee with name ',: NEW.ename));
  5  insert into trig_logg values(concat('you have removed an emloyee with name ',: OLD.ename));
  6  end;
  7  /

Trigger created.

SQL> update empl set ename='rahul' where ename='kumar';

0 rows updated.

SQL> update empl set ename='rahul' where ename='hari';

1 row updated.

SQL> select * from trig_logg;

LOGS
--------------------------------------------------------------------------------
A new row in empl table has been created
a new row in empl table has been updated
a new row in empl table has been deleted
A new row in empl table has been created
you have updated an emloyee with name rahul
you have removed an emloyee with name hari
a new row in empl table has been updated

7 rows selected.
```

Prgm 22

Program to retrieve data from employee table using cursor

```
SQL> set serveroutput on
SQL> edit stdcur.sql

SQL> start C:\Users\CAD3\Desktop\dbms\stdcur.sql
Employee id :3 Employee name :Ram

PL/SQL procedure successfully completed.

SQL>
```

Prgm 24
MongoDB program

AIM: Create a database College with following collections:

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

```
> use college45;
switched to db college45
> db.student.insertMany([{id:1, name:'arun',
... age:24,address:'karott',mobile:8767567785,semester:2},
... {id:2, name:'tharun',age:23,address:'thazhath',mobile:874372288,
... semester:1}]);
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("68354fd92c7e1d315b88bf1f"),
                ObjectId("68354fd92c7e1d315b88bf20")
        ]
}
> db.student.find().pretty();
{
        "_id" : ObjectId("68354eb22c7e1d315b88bf1d"),
        "id" : 1,
        "name" : "Arun",
        "mobile" : 9876543210,
        "age" : 24,
        "address" : "karott",
        "semester" : 2
}
{
        "_id" : ObjectId("68354eb22c7e1d315b88bf1e"),
        "id" : 2,
        "name" : "Tharun",
        "age" : 23
}
```

```
{
        "_id" : ObjectId("68354fd92c7e1d315b88bf1f"),
        "id" : 1,
        "name" : "arun",
        "age" : 24,
        "address" : "karott",
        "mobile" : 8767567785,
        "semester" : 2
}
{
        "_id" : ObjectId("68354fd92c7e1d315b88bf20"),
        "id" : 2,
        "name" : "tharun",
        "age" : 23,
        "address" : "thazhath",
        "mobile" : 874372288,
        "semester" : 1
}
```

2. Faculty (id, name, dept-name, salary, job_role)

```
> db.faculty.insertMany([{fid:11,name:'prasanth',
... dept_name:'mca',salary:24000,job_role:'clerk',
... age:26},{fid:12,name:'sreeshanth',dept_name:
... 'cse',job_role:'assistant_prof',age:37},
... {fid:13,name:'rasanth',dept_name:'ec',salary:
... 25000,job_role:'programmer',age:28},{fid:14,
... name:'suneesh',dept_name:'mca',salary:25000,
... job_role:'hod',age:32}]);
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("683551d62c7e1d315b88bf21"),
                ObjectId("683551d62c7e1d315b88bf22"),
                ObjectId("683551d62c7e1d315b88bf23"),
                ObjectId("683551d62c7e1d315b88bf24")
        ]
}
> db.faculty.find().pretty();
{
        "_id" : ObjectId("683551d62c7e1d315b88bf21"),
        "fid" : 11,
        "name" : "prasanth",
        "dept_name" : "mca",
        "salary" : 24000,
        "job_role" : "clerk",
        "age" : 26
}
```

```
{
        "_id" : ObjectId("683551d62c7e1d315b88bf22"),
        "fid" : 12,
        "name" : "sreeshanth",
        "dept_name" : "cse",
        "job_role" : "assistant_prof",
        "age" : 37
}
{
        "_id" : ObjectId("683551d62c7e1d315b88bf23"),
        "fid" : 13,
        "name" : "rasanth",
        "dept_name" : "ec",
        "salary" : 25000,
        "job_role" : "programmer",
        "age" : 28
}
{
        "_id" : ObjectId("683551d62c7e1d315b88bf24"),
        "fid" : 14,
        "name" : "suneesh",
        "dept_name" : "mca",
        "salary" : 25000,
        "job_role" : "hod",
        "age" : 32
}
```

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

```
> db.faculity.find({salary:{$gt:25000}}).pretty();
{
        "_id" : ObjectId("68303afb9304bea81b2758fd"),
        "f_id" : 12,
        "name" : "sreesanth",
        "dept_name" : "cse",
        "salary" : 26000,
        "job_role" : "Assistant_prof",
        "age" : 37
}
```

2. Find the documents with the 'salary' less than 25000 .

```
> db.faculty.find({salary:{$lt:25000}}).pretty();
{
        "_id" : ObjectId("683551d62c7e1d315b88bf21"),
        "fid" : 11,
        "name" : "prasanth",
        "dept_name" : "mca",
        "salary" : 24000,
        "job_role" : "clerk",
        "age" : 26
}
```

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

```
> db.faculty.find({salary:{$gte:25000}}).pretty();
{
        "_id" : ObjectId("683551d62c7e1d315b88bf23"),
        "fid" : 13,
        "name" : "rasanth",
        "dept_name" : "ec",
        "salary" : 25000,
        "job_role" : "programmer",
        "age" : 28
}
{
        "_id" : ObjectId("683551d62c7e1d315b88bf24"),
        "fid" : 14,
        "name" : "suneesh",
        "dept_name" : "mca",
        "salary" : 25000,
        "job_role" : "hod",
        "age" : 32
}
```

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

```
> db.faculty.find({salary:{$lte:25000}}).pretty();
{
        "_id" : ObjectId("683551d62c7e1d315b88bf21"),
        "fid" : 11,
        "name" : "prasanth",
        "dept_name" : "mca",
        "salary" : 24000,
        "job_role" : "clerk",
        "age" : 26
}
{
        "_id" : ObjectId("683551d62c7e1d315b88bf23"),
        "fid" : 13,
        "name" : "rasanth",
        "dept_name" : "ec",
        "salary" : 25000,
        "job_role" : "programmer",
        "age" : 28
}
{
        "_id" : ObjectId("683551d62c7e1d315b88bf24"),
        "fid" : 14,
        "name" : "suneesh",
        "dept_name" : "mca",
        "salary" : 25000,
        "job_role" : "hod",
        "age" : 32
}
>
```

5. **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("683551d62c7e1d315b88bf21"),
        "fid" : 11,
        "name" : "prasanth",
        "dept_name" : "mca",
        "salary" : 24000,
        "job_role" : "clerk",
        "age" : 26
}
{
        "_id" : ObjectId("683551d62c7e1d315b88bf22"),
        "fid" : 12,
        "name" : "sreeshanth",
        "dept_name" : "cse",
        "job_role" : "assistant_prof",
        "age" : 37
}
{
        "_id" : ObjectId("683551d62c7e1d315b88bf24"),
        "fid" : 14,
        "name" : "suneesh",
        "dept_name" : "mca",
        "salary" : 25000,
        "job_role" : "hod",
        "age" : 32
}
>
```

6. **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("68354eb22c7e1d315b88bf1d"),
        "id" : 1,
        "name" : "Arun",
        "mobile" : 9876543210,
        "age" : 24,
        "address" : "karott",
        "semester" : 2
}
{
        "_id" : ObjectId("68354eb22c7e1d315b88bf1e"),
        "id" : 2,
        "name" : "Tharun",
        "age" : 23
}
{
        "_id" : ObjectId("68354fd92c7e1d315b88bf1f"),
        "id" : 1,
        "name" : "arun",
        "age" : 24,
        "address" : "karott",
        "mobile" : 8767567785,
        "semester" : 2
}
```

7. Find documents where the value of the D_id field is not equal to D_01 in the department collection.

```
> db.department.find({did:{$ne:'D_01'}}).pretty();
{
        "_id" : ObjectId("683553882c7e1d315b88bf26"),
        "did" : "D_02",
        "dept_name" : "ce",
        "location" : "CIVIL Block"
}
{
        "_id" : ObjectId("683553882c7e1d315b88bf27"),
        "did" : "D_03",
        "dept_name" : "cse",
        "location" : "CS Block"
}
```

8. 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:{$gt:30}},{age:{$lt:40}}]}]}).pretty();
{
        "_id" : ObjectId("683551d62c7e1d315b88bf22"),
        "fid" : 12,
        "name" : "sreeshanth",
        "dept_name" : "cse",
        "job_role" : "assistant_prof",
        "age" : 37
}
>
```

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

job_role is equal to "Programer" or "Clerk"

```
> db.faculty.find({$or:[{job_role:'programmer'},
... {job_role:'clerk'}]}).pretty();
{
        "_id" : ObjectId("683551d62c7e1d315b88bf21"),
        "fid" : 11,
        "name" : "prasanth",
        "dept_name" : "mca",
        "salary" : 24000,
        "job_role" : "clerk",
        "age" : 26
}
{
        "_id" : ObjectId("683551d62c7e1d315b88bf23"),
        "fid" : 13,
        "name" : "rasanth",
        "dept_name" : "ec",
        "salary" : 25000,
        "job_role" : "programmer",
        "age" : 28
}
>
```

10. 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:{$ne:'CIVIL Block'}}]}).pretty();
{
        "_id" : ObjectId("683553882c7e1d315b88bf27"),
        "did" : "D_03",
        "dept_name" : "cse",
        "location" : "CS Block"
}
>
>
> db.faculty.find({age:{$not:{$gte:40}}}).pretty();
{
        "_id" : ObjectId("683551d62c7e1d315b88bf21"),
        "fid" : 11,
        "name" : "prasanth",
        "dept_name" : "mca",
        "salary" : 24000,
        "job_role" : "clerk",
        "age" : 26
}
{
        "_id" : ObjectId("683551d62c7e1d315b88bf22"),
        "fid" : 12,
        "name" : "sreeshanth",
        "dept_name" : "cse",
        "job_role" : "assistant_prof",
        "age" : 37
}
```

Rhythmbox

```
{
        "_id" : ObjectId("683551d62c7e1d315b88bf23"),
        "fid" : 13,
        "name" : "rasanth",
        "dept_name" : "ec",
        "salary" : 25000,
        "job_role" : "programmer",
        "age" : 28
}
{
        "_id" : ObjectId("683551d62c7e1d315b88bf24"),
        "fid" : 14,
        "name" : "suneesh",
        "dept_name" : "mca",
        "salary" : 25000,
        "job_role" : "hod",
        "age" : 32
}
```

11. Find documents where the job_role field exists and equal to "HOD". 1

```
> db.faculty.find({$and:[{job_role:{$exists:true}},
... {job_role:'hod'}]}).pretty();
{ Screenshot
        "_id" : ObjectId("683551d62c7e1d315b88bf24"),
        "fid" : 14,
        "name" : "suneesh",
        "dept_name" : "mca",
        "salary" : 25000,
        "job_role" : "hod",
        "age" : 32
}
```

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

```
> db.department.find({location:{$exists:true}}).pretty();
{
        "_id" : ObjectId("68303b289304bea81b275900"),
        "d_id" : "D_01",
        "dept_name" : "MCA",
        "location" : "MCA Block"
}
{
        "_id" : ObjectId("68303b289304bea81b275901"),
        "d_id" : "D_02",
        "dept_name" : "ce",
        "location" : "civil Block"
}
{
        "_id" : ObjectId("68303b289304bea81b275902"),
        "d_id" : "D_03",
        "dept_name" : "cse",
        "location" : "cs Block"
}
```