**Practical 1(A)**

**Problem statement-** Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym, different constraints etc.

**//To show databases**

mysql> **show databases;**

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

| sakila |

| sys |

| world |

+--------------------+

6 rows in set (0.03 sec)

**//To create new database**

mysql> **create database DBMS ;**

Query OK, 1 row affected (0.01 sec)

**//To use the new or existing database**

**mysql> use DBMS;**

Database changed

**//To create new table**

**mysql> create table Employee(Employee\_Id integer(10) not null,Employee\_Name varchar(100) not null,Designation varchar(40),Department varchar(40),Salary float(20),City varchar(20),primary key (Employee\_Id));**

Query OK, 0 rows affected, 1 warning (0.02 sec)

**// To insert values in table**

mysql> **insert into Employee values(1,'Harsh','Engineer', 'Computer',100000,'Pune');**

Query OK, 1 row affected (0.01 sec)

mysql> **insert into Employee values(2,'Amit','Engineer', 'Mechanical',90000,'Pune');**

Query OK, 1 row affected (0.01 sec)

mysql> **insert into Employee values(3,'Tanay','Engineer', 'ENTC',80000,'Satara');**

Query OK, 1 row affected (0.00 sec)

mysql**> insert into Employee values(4,'Parth','Engineer', 'IT',70000,'Nashik');**

Query OK, 1 row affected (0.00 sec)

mysql**> insert into Employee values(5,'Omkar','Engineer', 'Civil',50000,'Solapur');**

Query OK, 1 row affected (0.02 sec)

**//To display table**

mysql> **select \* from Employee;**

+-------------+---------------+-------------+------------+--------+---------+

| Employee\_Id | Employee\_Name | Designation | Department | Salary | City |

+-------------+---------------+-------------+------------+--------+---------+

| 1 | Harsh | Engineer | Computer | 100000 | Pune |

| 2 | Amit | Engineer | Mechanical | 90000 | Pune |

| 3 | Tanay | Engineer | ENTC | 80000 | Satara |

| 4 | Parth | Engineer | IT | 70000 | Nashik |

| 5 | Omkar | Engineer | Civil | 50000 | Solapur |

+-------------+---------------+-------------+------------+--------+---------+

5 rows in set (0.02 sec)

**// To create view**

mysql> **create view Employee\_view as**

**-> select Employee\_Id,Employee\_NAME,City from Employee where Salary > 70000;**

Query OK, 0 rows affected (0.01 sec)

mysql> **select \* from Employee\_view;**

+-------------+---------------+--------+

| Employee\_Id | Employee\_NAME | City |

+-------------+---------------+--------+

| 1 | Harsh | Pune |

| 3 | Tanay | Satara |

+-------------+---------------+--------+

2 rows in set (0.00 sec)

mysql> **select Employee\_Name,Salary from Employee where Salary>70000;**

+---------------+--------+

| Employee\_Name | Salary |

+---------------+--------+

| Harsh | 100000 |

| Tanay | 80000 |

+---------------+--------+

2 rows in set (0.00 sec)

**//Index**

mysql> **create index idx1 on Employee(Email);**

Query OK, 0 rows affected (0.02 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql**> show index from Employee;**

+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+

| Table | Non\_unique | Key\_name | Seq\_in\_index | Column\_name | Collation | Cardinality | Sub\_part | Packed | Null | Index\_type | Comment | Index\_comment | Visible | Expression |

+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+

| employee | 0 | PRIMARY | 1 | Employee\_Id | A | 4 | NULL | NULL | | BTREE | | | YES | NULL |

| employee | 1 | idx1 | 1 | Email | A | 1 | NULL | NULL | YES | BTREE | | | YES | NULL |

+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+

2 rows in set (0.01 sec)

mysql> **drop index idx1 on Employee;**

Query OK, 0 rows affected (0.01 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> **show index from Employee;**

+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+

| Table | Non\_unique | Key\_name | Seq\_in\_index | Column\_name | Collation | Cardinality | Sub\_part | Packed | Null | Index\_type | Comment | Index\_comment | Visible | Expression |

+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+

| employee | 0 | PRIMARY | 1 | Employee\_Id | A | 4 | NULL | NULL | | BTREE | | | YES | NULL |

+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+

1 row in set (0.00 sec)

**Practical 1(B)**

**Problem statement-** • Write at least 10 SQL queries on the suitable database application using SQL DML statements.

**//To create new table**

**mysql> create table Employee(Employee\_Id integer(10) not null,Employee\_Name varchar(100) not null,Designation varchar(40),Department varchar(40),Salary float(20),City varchar(20),primary key (Employee\_Id));**

Query OK, 0 rows affected, 1 warning (0.02 sec)

**// To insert values in table**

mysql> **insert into Employee values(1,'Harsh','Engineer', 'Computer',100000,'Pune');**

Query OK, 1 row affected (0.01 sec)

mysql> **insert into Employee values(2,'Amit','Engineer', 'Mechanical',90000,'Pune');**

Query OK, 1 row affected (0.01 sec)

mysql> **insert into Employee values(3,'Tanay','Engineer', 'ENTC',80000,'Satara');**

Query OK, 1 row affected (0.00 sec)

mysql**> insert into Employee values(4,'Parth','Engineer', 'IT',70000,'Nashik');**

Query OK, 1 row affected (0.00 sec)

mysql**> insert into Employee values(5,'Omkar','Engineer', 'Civil',50000,'Solapur');**

Query OK, 1 row affected (0.02 sec)

**//To display table**

mysql> **select \* from Employee;**

+-------------+---------------+-------------+------------+--------+---------+

| Employee\_Id | Employee\_Name | Designation | Department | Salary | City |

+-------------+---------------+-------------+------------+--------+---------+

| 1 | Harsh | Engineer | Computer | 100000 | Pune |

| 2 | Amit | Engineer | Mechanical | 90000 | Pune |

| 3 | Tanay | Engineer | ENTC | 80000 | Satara |

| 4 | Parth | Engineer | IT | 70000 | Nashik |

| 5 | Omkar | Engineer | Civil | 50000 | Solapur |

+-------------+---------------+-------------+------------+--------+---------+

5 rows in set (0.02 sec)

**//To delete row**

mysql**> Delete from Employee where Salary=90000;**

Query OK, 1 row affected (0.00 sec)

mysql> **select \* from Employee;**

+-------------+---------------+-------------+------------+--------+---------+

| Employee\_Id | Employee\_Name | Designation | Department | Salary | City |

+-------------+---------------+-------------+------------+--------+---------+

| 1 | Harsh | Engineer | Computer | 100000 | Pune |

| 3 | Tanay | Engineer | ENTC | 80000 | Satara |

| 4 | Parth | Engineer | IT | 70000 | Nashik |

| 5 | Omkar | Engineer | Civil | 50000 | Solapur |

+-------------+---------------+-------------+------------+--------+---------+

4 rows in set (0.00 sec)

**//Updating values**

mysql> **update Employee set Employee\_Name='Pratik' where Salary=70000;**

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> **select \* from Employee;**

+-------------+---------------+-------------+------------+--------+---------+

| Employee\_Id | Employee\_Name | Designation | Department | Salary | City |

+-------------+---------------+-------------+------------+--------+---------+

| 1 | Harsh | Engineer | Computer | 100000 | Pune |

| 3 | Tanay | Engineer | ENTC | 80000 | Satara |

| 4 | Pratik | Engineer | IT | 70000 | Nashik |

| 5 | Omkar | Engineer | Civil | 50000 | Solapur |

+-------------+---------------+-------------+------------+--------+---------+

4 rows in set (0.00 sec)

**Practical No.2**

**Problem statement-**Write at least10 SQL queries for suitable database application using SQL DML statements. Note: Instructor will design the queries which demonstrate the use of concepts like all types of Join ,Sub-Query and View

**mysql> create table customers ( c\_id int(10),c\_name varchar(40),c\_email varchar(40));**

**mysql> insert into customers values(1,'Vinod','vinod@thapa.com');**

**mysql> insert into customers values(2,'Ram','ram@roy.com');**

**mysql> insert into customers values(3,'Ravi','ravi@khan.com');**

**mysql> insert into customers values(4,'Nitin','nitin@puri.com');**

**mysql> select \* from customers;**

+------+--------+-----------------+

| c\_id | c\_name | c\_email |

+------+--------+-----------------+

| 1 | Vinod | vinod@thapa.com |

| 2 | Ram | ram@roy.com |

| 3 | Ravi | ravi@khan.com |

| 4 | Nitin | nitin@puri.com |

+------+--------+-----------------+

**mysql> create table orders(o\_id int(20),o\_date varchar(20),o\_amount int(20),c\_id int(20));**

**mysql> insert into orders values(1,'2019-05-05',55,1);**

**mysql> insert into orders values(2,'2019-08-06',85,2);**

**mysql> insert into orders values(3,'2019-08-05',155,1);**

**mysql> insert into orders values(4,'2019-05-12',95,3);**

**mysql> select\* from orders;**

+------+------------+----------+------+

| o\_id | o\_date | o\_amount | c\_id |

+------+------------+----------+------+

| 1 | 2019-05-05 | 55 | 1 |

| 2 | 2019-08-06 | 85 | 2 |

| 3 | 2019-08-05 | 155 | 1 |

| 4 | 2019-05-12 | 95 | 3 |

+------+------------+----------+------+

**//INNER JOIN**

**mysql> select \* from customers inner join orders where customers.c\_id=orders.c\_id;**

+------+--------+-----------------+------+------------+----------+------+

| c\_id | c\_name | c\_email | o\_id | o\_date | o\_amount | c\_id |

+------+--------+-----------------+------+------------+----------+------+

| 1 | Vinod | vinod@thapa.com | 1 | 2019-05-05 | 55 | 1 |

| 2 | Ram | ram@roy.com | 2 | 2019-08-06 | 85 | 2 |

| 1 | Vinod | vinod@thapa.com | 3 | 2019-08-05 | 155 | 1 |

| 3 | Ravi | ravi@khan.com | 4 | 2019-05-12 | 95 | 3 |

+------+--------+-----------------+------+------------+----------+------+

**//LEFT JOIN**

**mysql> select customers.c\_id,c\_name,o\_amount from customers left join orders on customers.c\_id=orders.c\_id;**

+------+--------+----------+

| c\_id | c\_name | o\_amount |

+------+--------+----------+

| 1 | Vinod | 155 |

| 1 | Vinod | 55 |

| 2 | Ram | 85 |

| 3 | Ravi | 95 |

| 4 | Nitin | NULL |

+------+--------+----------+

**// RIGHT JOIN**

**mysql> select customers.c\_id,c\_name,o\_amount from customers right join orders on customers.c\_id=orders.c\_id;**

+------+--------+----------+

| c\_id | c\_name | o\_amount |

+------+--------+----------+

| 1 | Vinod | 55 |

| 2 | Ram | 85 |

| 1 | Vinod | 155 |

| 3 | Ravi | 95 |

+------+--------+----------+

**//View**

**mysql> create view view1 as select c\_id,c\_name,c\_email from customers where c\_id>2;**

**Query OK, 0 rows affected (0.02 sec)**

**mysql> select \* from view1;**

+------+--------+----------------+

| c\_id | c\_name | c\_email |

+------+--------+----------------+

| 3 | Ravi | ravi@khan.com |

| 4 | Nitin | nitin@puri.com |

+------+--------+----------------+

2 rows in set (0.00 sec)

**//Subquery**

**mysql> select \* from customers;**

+------+--------+-----------------+

| c\_id | c\_name | c\_email |

+------+--------+-----------------+

| 1 | Vinod | vinod@thapa.com |

| 2 | Ram | ram@roy.com |

| 3 | Ravi | ravi@khan.com |

| 4 | Nitin | nitin@puri.com |

+------+--------+-----------------+

4 rows in set (0.00 sec)

**mysql> select \* from orders;**

+------+------------+----------+------+

| o\_id | o\_date | o\_amount | c\_id |

+------+------------+----------+------+

| 1 | 2019-05-05 | 55 | 1 |

| 2 | 2019-08-06 | 85 | 2 |

| 3 | 2019-08-05 | 155 | 1 |

| 4 | 2019-05-12 | 95 | 3 |

+------+------------+----------+------+

4 rows in set (0.01 sec)

**mysql> select c\_id ,c\_name,c\_email from customers where c\_id in (select c\_id from orders where o\_amount>60);**

+------+--------+-----------------+

| c\_id | c\_name | c\_email |

+------+--------+-----------------+

| 1 | Vinod | vinod@thapa.com |

| 2 | Ram | ram@roy.com |

| 3 | Ravi | ravi@khan.com |

+------+--------+-----------------+

3 rows in set (0.00 sec)

**Practical No: 03**

**Problem statement-**Design and Develop MongoDB Queries using CRUD operations. (Use CRUD operations, SAVE method, logical operators etc.). 4. Unnamed PL/SQLc

test> show dbs

Employee 72.00 KiB

admin 40.00 KiB

config 72.00 KiB

local 40.00 KiB

test> use company

switched to db company

company> show dbs

Employee 72.00 KiB

admin 40.00 KiB

config 72.00 KiB

local 40.00 KiB

company> db.employee.insert({

... empname:"yash",

... Empaddress:"pune"

... })

DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

{

acknowledged: true,

insertedIds: { '0': ObjectId("65298e64321b8a1817b5bc9d") }

}

company> show collections

employee

company> db.employee.find().pretty()

[

{

\_id: ObjectId("65298e64321b8a1817b5bc9d"),

empname: 'yash',

Empaddress: 'pune'

}

]

company> db.employee.count()

DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.

1

company> db.employee.insert({

...

... empname:"satish",

... empaddress:"mumbai"})

{

acknowledged: true,

insertedIds: { '0': ObjectId("65298f98321b8a1817b5bc9e") }

}

company> db.employee.find().pretty()

[

{

\_id: ObjectId("65298e64321b8a1817b5bc9d"),

empname: 'yash',

Empaddress: 'pune'

},

{

\_id: ObjectId("65298f98321b8a1817b5bc9e"),

empname: 'satish',

empaddress: 'mumbai'

}

]

company> db.employee.count()

2

company> db.employee.update({empname:"satish"},{$set:{empaddress:"solapur"}})

DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

company> db.employee.find().pretty()

[

{

\_id: ObjectId("65298e64321b8a1817b5bc9d"),

empname: 'yash',

Empaddress: 'pune'

},

{

\_id: ObjectId("65298f98321b8a1817b5bc9e"),

empname: 'satish',

empaddress: 'solapur'

}

]

company> db.employee.remove({empname:"satish"})

DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.

{ acknowledged: true, deletedCount: 1 }

company> db.employee.find().pretty()

[

{

\_id: ObjectId("65298e64321b8a1817b5bc9d"),

empname: 'yash',

Empaddress: 'pune'

}

]

company>

**Practical 4**

**Problem Statement-**. Unnamed PL/SQLcode block: Use of Control structure and Exception handling is mandatory. Suggested Problem statement: Consider Tables

: 1. Borrower (Roll\_no, Name, Date\_of\_Issue, Name\_of\_Book, Status)

2. Fine (Roll\_no, Date, Amt) • Accept Roll\_no and Name\_of\_Book from user.

• Check the number of days (from Date\_of\_Issue).

• If days are between 15 to 30 then fine amount will be Rs 5per day.

• If no. of days>30, per day fine will be Rs 50 per day and for days less than 30, Rs. 5 per day.

• After submitting the book, status will change from I to R.

• If condition of fine is true, then details will be stored into fine table.

• Also handles the exception by named exception handler or user define exception handler.

**mysql> create table Borrower1(Roll\_no int,Name varchar(200),Date\_of\_issue date, book\_name varchar(50),satus varchar(50))**

**;**

Query OK, 0 rows affected (0.02 sec)

**mysql> insert into Borrower1 values(1,"Amit",'2018-07-01',"oop","issued");**

Query OK, 1 row affected (0.01 sec)

**mysql> insert into Borrower1 values(2,"Ray",'2018-06-01',"dbms","issued");**

Query OK, 1 row affected (0.00 sec)

**mysql> insert into Borrower1 values(3,"Suresh",'2018-05-01',"toc","returned");**

Query OK, 1 row affected (0.00 sec)

**mysql> insert into Borrower1 values(4,"Rohan",'2018-06-15',"hci","returned");**

Query OK, 1 row affected (0.00 sec)

**mysql> insert into Borrower1 values(5,"Ram",'2018-05-15',"dsa","issued");**

Query OK, 1 row affected (0.00 sec)

**mysql> Select \* from Borrower1;**

+---------+--------+---------------+-----------+----------+

| Roll\_no | Name | Date\_of\_issue | book\_name | satus |

+---------+--------+---------------+-----------+----------+

| 1 | Amit | 2018-07-01 | oop | issued |

| 2 | Ray | 2018-06-01 | dbms | issued |

| 3 | Suresh | 2018-05-01 | toc | returned |

| 4 | Rohan | 2018-06-15 | hci | returned |

| 5 | Ram | 2018-05-15 | dsa | issued |

+---------+--------+---------------+-----------+----------+

5 rows in set (0.00 sec)

**mysql> create table Fine(Roll\_no int,fine\_date date,amount int);**

Query OK, 0 rows affected (0.01 sec)

**mysql> delimiter //**

**mysql> create procedure B(roll\_new int,book\_name varchar(20))**

**begin**

**declare X integer;**

**declare continue handler for not found**

**begin**

**select 'NOT FOUND';**

**end;**

**select datediff(curdate(),Date\_of\_issue)into X from Borrower1**

**where Roll\_no=roll\_new;**

**if(X>15&&X<30)**

**then**

**insert into fine values(roll\_new,curdate(),(X\*5));**

**end if;**

**if(X>30)**

**then**

**insert into fine values(roll\_new,curdate(),(X\*50));**

**end if;**

**update Borrower1 set status ='returned' where Roll\_no=roll\_new;**

**end;**

**//**

**mysql> call B(1,'oop');//**

**Query OK, 1 row affected (0.01 sec)**

**mysql> select \* from Fine;//**

+---------+------------+--------+

| Roll\_no | fine\_date | amount |

+---------+------------+--------+

| 1 | 2023-10-16 | 96650 |

+---------+------------+--------+

1 row in set (0.00 sec)

**mysql> call B(2,'dbms');//**

**Query OK, 1 row affected (0.01 sec)**

mysql> select \* from Fine;//

+---------+------------+--------+

| Roll\_no | fine\_date | amount |

+---------+------------+--------+

| 1 | 2023-10-16 | 96650 |

| 2 | 2023-10-16 | 98150 |

+---------+------------+--------+

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

**mysql> call B(3,'toc');**

**-> call B(4,'hci');**

**-> call B(5,'dsa');//**

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

**Query OK, 0 rows affected (0.01 sec)**

**Query OK, 1 row affected (0.01 sec)**

**mysql> Select \*from Fine;**

**-> //**

+---------+------------+--------+

| Roll\_no | fine\_date | amount |

+---------+------------+--------+

| 1 | 2023-10-16 | 96650 |

| 2 | 2023-10-16 | 98150 |

| 3 | 2023-10-16 | 99700 |

| 4 | 2023-10-16 | 97450 |

| 5 | 2023-10-16 | 99000 |

+---------+------------+--------+

5 rows in set (0.00 sec)

**Practical 5(A)**

**Problem statement-**· Design and develop SQL DML statements to demonstrate exporting tables to external files of different file formats ex. CSV, XLSX, TXT, etc.

**Exporting**

mysql> use dbms;

Database changed

mysql> select \*from Borrower1;

+---------+--------+---------------+-----------+----------+

| Roll\_no | Name   | Date\_of\_issue | book\_name | satus    |

+---------+--------+---------------+-----------+----------+

|       1 | Amit   | 2018-07-01    | oop       | returned |

|       2 | Ray    | 2018-06-01    | dbms      | returned |

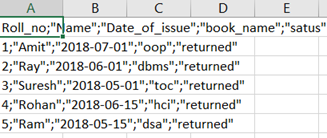
|       3 | Suresh | 2018-05-01    | toc       | returned |

|       4 | Rohan  | 2018-06-15    | hci       | returned |

|       5 | Ram    | 2018-05-15    | dsa       | returned |

+---------+--------+---------------+-----------+----------+

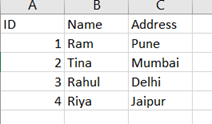
5 rows in set (0.00 sec)



**Practical 5(B)**

**Problem statement-** · Design and develop SQL DML statements to demonstrate importing data from external files of different file formats ex. CSV, XLSX, TXT, etc.

**CSV file**



**Importing data to sql**

**mysql> SELECT \* FROM python\_connect.sample\_csv;**

+------+-------+---------+

| ID   | Name  | Address |

+------+-------+---------+

|    1 | Ram   | Pune    |

|    2 | Tina  | Mumbai  |

|    3 | Rahul | Delhi   |

|    4 | Riya  | Jaipur  |

+------+-------+---------+

4 rows in set (0.00 sec)

**Practical 6**

**Problem statement-**Write a PL/SQL block of code using parameterized Cursor that will merge the data available in the newly created table N\_Roll\_Call with the data available in the table O\_Roll\_Call. If the data in the first table already exists in the second table then that data should be skipped. Note: Instructor will frame the problem statement for writing PL/SQL block using all types of Cursors in line with above statement.

**mysql> create table o\_rollcall(roll\_no int,name varchar(50),address varchar(50));**

Query OK, 0 rows affected (0.02 sec)

**mysql> create table n\_rollcall(roll\_no int,name varchar(20),address varchar(20));**

Query OK, 0 rows affected (0.01 sec)

**mysql> insert into o\_rollcall values(1,'Ram','Pune');**

Query OK, 1 row affected (0.00 sec)

**mysql> insert into o\_rollcall values(2,'Suresh','Mumbai');**

Query OK, 1 row affected (0.00 sec)

**mysql> insert into o\_rollcall values(3,'Karan','Solapur');**

Query OK, 1 row affected (0.00 sec)

**mysql> insert into o\_rollcall values(4,'Hina','kolkata');**

Query OK, 1 row affected (0.00 sec)

**mysql> insert into o\_rollcall values(5,'Sham','Jaipur');**

Query OK, 1 row affected (0.00 sec)

**mysql> delimiter //**

mysql> create procedure p3(in r1 int)

begin

declare r2 int;

declare exit\_loop boolean;

declare c1 cursor for select roll\_no from o\_rollcall

where roll\_no>r1;

declare continue handler for not found set

exit\_loop=true;

open c1;

e\_loop:loop

fetch c1 into r2;

if not exists(select \* from n\_rollcall where

roll\_no=r2)

then

insert into n\_rollcall select \* from o\_rollcall where

roll\_no=r2;

end if;

if exit\_loop

then

close c1;

leave e\_loop;

end if;

end loop e\_loop;

end

//

Query OK, 0 rows affected (0.01 sec)

**mysql> call p3(0);//**

Query OK, 0 rows affected (0.01 sec)

**mysql> select \* from n\_rollcall;**

**-> //**

+---------+--------+---------+

| roll\_no | name | address |

+---------+--------+---------+

| 1 | Ram | Pune |

| 2 | Suresh | Mumbai |

| 3 | Karan | Solapur |

| 4 | Hina | kolkata |

| 5 | Sham | Jaipur |

+---------+--------+---------+

5 rows in set (0.00 sec)

**mysql> insert into o\_rollcall values(6,'Tina','Gujarat');//**

Query OK, 1 row affected (0.00 sec)

**mysql> select \* from n\_rollcall;**

**-> //**

+---------+--------+---------+

| roll\_no | name | address |

+---------+--------+---------+

| 1 | Ram | Pune |

| 2 | Suresh | Mumbai |

| 3 | Karan | Solapur |

| 4 | Hina | kolkata |

| 5 | Sham | Jaipur |

+---------+--------+---------+

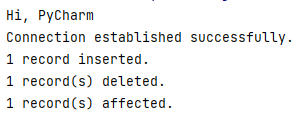
5 rows in set (0.00 sec)

**Practical 7**

**Problem statement-**Write a program to implement MySQL/Oracle database connectivity with any front end language to implement Database navigation operations (add, delete, edit etc.)

import mysql.connector  
  
*# Establish the connection*mydb = mysql.connector.connect(  
  host="localhost",  
  user="root",  
  password="root123",  
  database="python\_connect"  
)  
  
*# Check if the connection is established*if mydb.is\_connected():  
    print("Connection established successfully.")  
  
*# Function to add data*def add\_data(Name, Age):  
    cursor = mydb.cursor()  
    sql = "INSERT INTO info (Name, Age) VALUES (%s, %s)"  
    val = (Name, Age)  
    cursor.execute(sql, val)  
    mydb.commit()  
    print(cursor.rowcount, "record inserted.")  
  
*# Function to delete data*def delete\_data(id):  
    cursor = mydb.cursor()  
    sql = ("DELETE FROM info WHERE id = %s")  
    val = (id,)  
    cursor.execute(sql, val)  
    mydb.commit()  
    print(cursor.rowcount, "record(s) deleted.")  
  
*# Function to edit data*def edit\_data(id, name, age):  
    cursor = mydb.cursor()  
    sql = "UPDATE info SET name = %s, age = %s WHERE id = %s"  
    val = (name, age, id)  
    cursor.execute(sql, val)  
    mydb.commit()  
    print(cursor.rowcount, "record(s) affected.")  
  
*# Example usage of functions*add\_data("John Doe", 30)  
delete\_data(1)  
edit\_data(2, "Jane Doe", 35)  
  
*# Close the connection*mydb.close()

**OUTPUT**

****

**Mysql**

**mysql> create table info(id int auto\_increment primary key ,Name varchar(45),Age int );**

**Query OK, 0 rows affected (0.01 sec)**

**mysql> select \* from info;**

+----+----------+------+

| id | Name     | Age  |

+----+----------+------+

|  2 | Jane Doe |   35 |

|  3 | Tina     |   22 |

|  4 | John Doe |   30 |

+----+----------+------+

3 rows in set (0.00 sec)