# RAMANUJAN COLLEGE

University of Delhi

# Database Management Practical File

NAME-RAVIKANT MAURYA

COLLEGE ROLL.NO-20221433

EXAM ROLL.NO-22020570026

COURSE-B.SC.(H)COMPUTER SCIENCE

# DATABASE MANAGEMENT SYSTEM

## (I) CREATING A DATABASE NAME AS "STUDENTSOCIETY" FOR A COLLEGE'S STUDENT-SOCIETY DATABASE IN MYSQL.

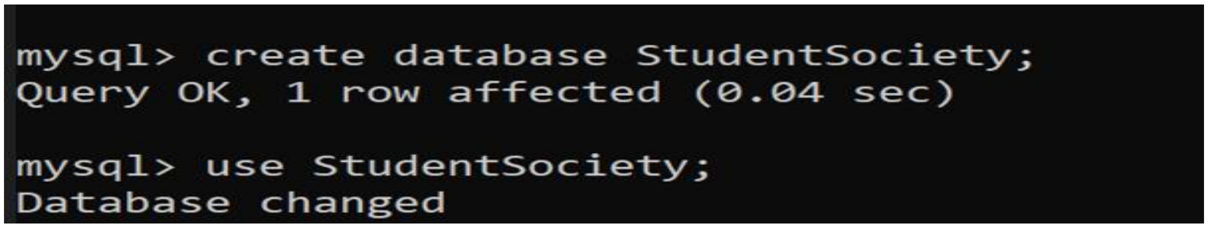
## Creating Database and Tables

### Creating database:

**Query:** *create database studentsociety;*

*use studentsociety;*

**Output:**



```
mysql> create database StudentSociety;
Query OK, 1 row affected (0.04 sec)

mysql> use StudentSociety;
Database changed
```

***<u>Creating table student:</u>***

**<u>Query:</u>** *Create table student(*

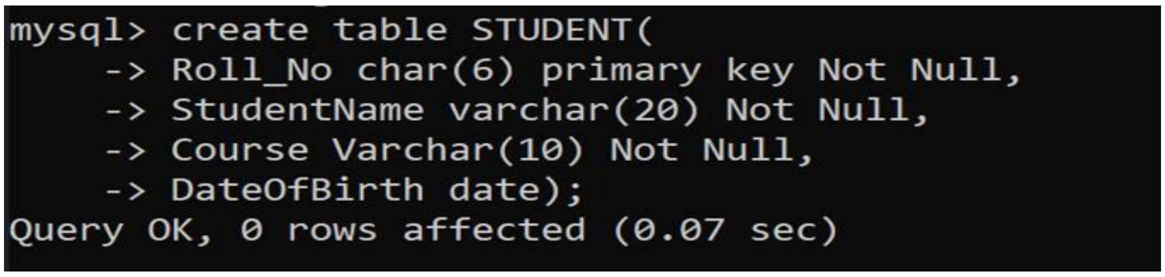
*Roll_No  char(6) primary key Not Null,*

*StudentName varchar(20) Not Null,*

*Course varchar(10) Not Null,*

*DateofBirth date);*

***<u>Output:</u>***



```
mysql> create table STUDENT(
    -> Roll_No char(6) primary key Not Null,
    -> StudentName varchar(20) Not Null,
    -> Course Varchar(10) Not Null,
    -> DateOfBirth date);
Query OK, 0 rows affected (0.07 sec)
```

***<u>Creating table Society:</u>***

**<u>Query:</u>** *create table society(*

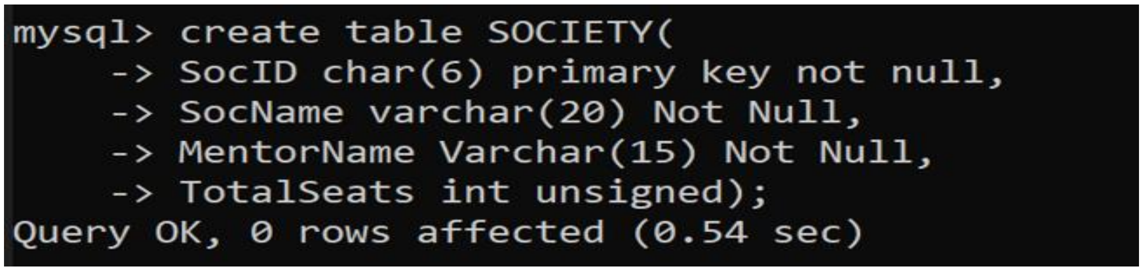
*SocID char(6)  primary key not null,*

*SocName varchar(20) not null,*

*MentorName varchar(15) Not Null,*

*TotalSeats int unsigned);*

***Output:***



```
mysql> create table SOCIETY(
    -> SocID char(6) primary key not null,
    -> SocName varchar(20) Not Null,
    -> MentorName Varchar(15) Not Null,
    -> TotalSeats int unsigned);
Query OK, 0 rows affected (0.54 sec)
```

***Creating table Enrollment:***

***Query:*** *create table enrollment(*

*Roll_No char(6) Not Null,*

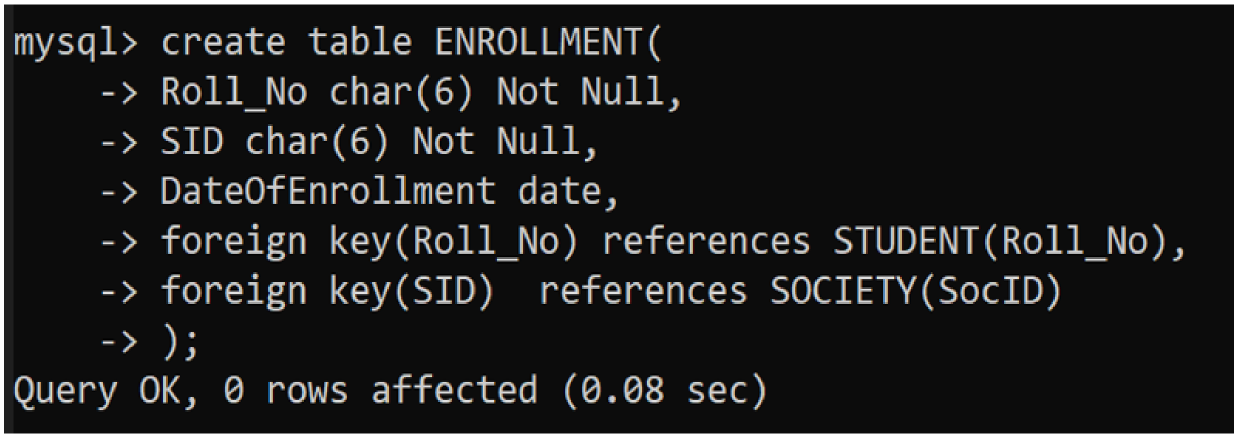
*SID char(6) Not Null,*

*DateofEnrollment date,*

*foreign key (Roll_No) references Student(Roll_No),*

*foreign key(SID) references Society(SocID);*

***Output:***



```
mysql> create table ENROLLMENT(
    -> Roll_No char(6) Not Null,
    -> SID char(6) Not Null,
    -> DateOfEnrollment date,
    -> foreign key(Roll_No) references STUDENT(Roll_No),
    -> foreign key(SID)  references SOCIETY(SocID)
    -> );
Query OK, 0 rows affected (0.08 sec)
```

***<u>Inserting values in Student Table:</u>***

***<u>Query:</u>*** *insert into student values*

```
("x2345","Aman","BSC(H)CHEM","2001-01-23"),
("x2348","Mohit","BSC(H)CHEM","2002-01-20"),
("x2349","Harsh","BSC(H)CHEM","2002-08-21"),
("x2350","Ashish","BSC(H)CHEM","2001-08-11"),
("x2351","Sandeep","BSC(H)CHEM","2001-09-15"),
("x1234","Rahul","BSC(H)CHEM","2001-10-28"),
("x1235","Anamika","BSC(H)CHEM","2007-11-22"),
("x1238","Aditya","BSC(H)CHEM","2001-11-15"),
("y1239","Varun","BSC(H)CS","2001-11-21"),
("y1240","Arun","BSC(H)CS","2001-01-01"),
("y1241","Mishti","BSC(H)CS","2001-02-05"),
("y1242","Ankit","BSC(H)CS","2001-03-07"),
("y1243","Kavita","BSC(H)CS","2001-06-02"),
("y1244","Kiran","BSC(H)CS","2002-07-12"),
("y1245","Karan","BSC(H)CS","2003-08-22"),
 ("y1246","Suraj","BSC(H)CS","2001-09-14"),
 ("y1247","Sachin","BSC(H)CS","2001-05-12"),
 ("zz4669","Annu","BA(HONS)","2003-11-22"),
("zX4669","Nilam","BA(HONS)","2004-03-21"),
 ("zX4679","Muskan","BA(HONS)","2002-01-12"),
("zX4689","Sheetal","BA(HONS)","2001-02-11"),
 ("zX4699","Riya","BA(HONS)","2001-02-10"),
("624699","Jyoti","BA(HONS)","2001-12-11"),
("624694","Priya","BA(HONS)","2001-07-12"),
  ("z1249","Nitin","BSC(H)CS","2001-02-12"),
 ("x1249","Nishant","BSC(H)CS","2001-03-17"),
  ("x4569","Sanjay","BSC(H)CS","2001-05-23"),
  ("z4579","Arjun","BSC(H)CS","2001-06-14"),
  ("z4589","Himanshu","BSC(H)CS","2003-04-17"),
  ("x4599","Ritik","BSC(H)MATH","2002-03-19"),
  ("x4619","Ritesh","BSC(H)MATH","2003-04-11"),
  ("x4629","Vijay","BSC(H)MATH","2002-06-13"),
```

*("z4639","Vishal","BSC(H)MATH","2001-10-24"),*

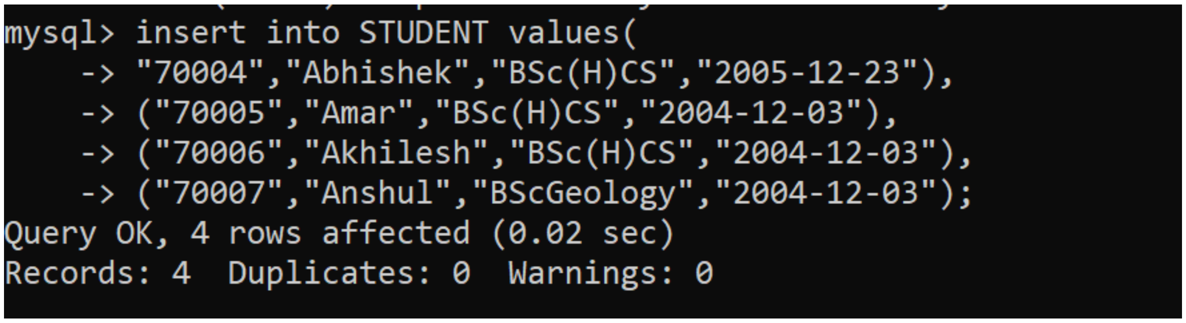
*("z4649","Vikram","BSC(H)MATH","2001-11-26"),*

*("z4659","Divya","BSC(H)MATH","2010-11-26"),*

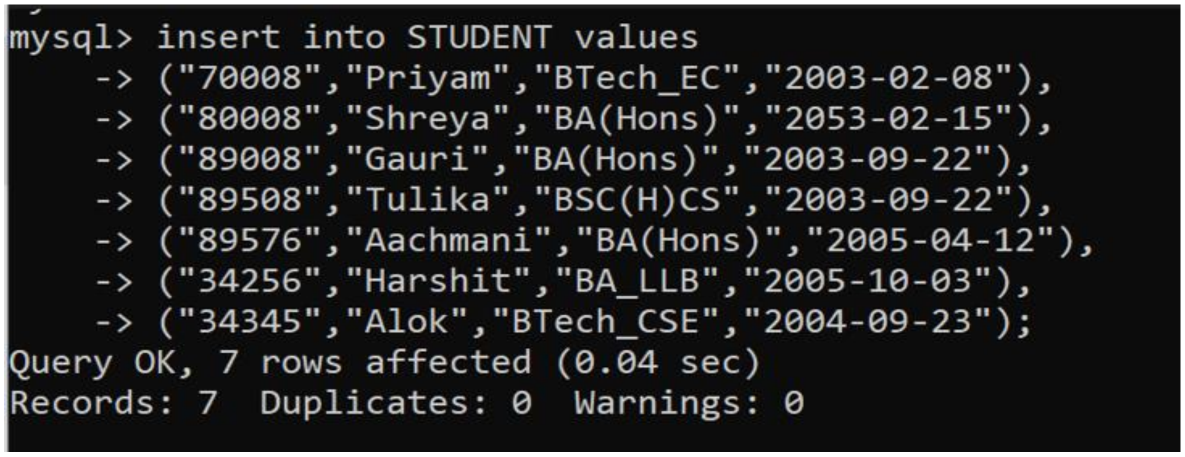
*("z4669","Tanu","BSC(H)MATH","2011-12-22"),*

*("z4679","Neha","BSC(H)MATH","2006-10-22");*

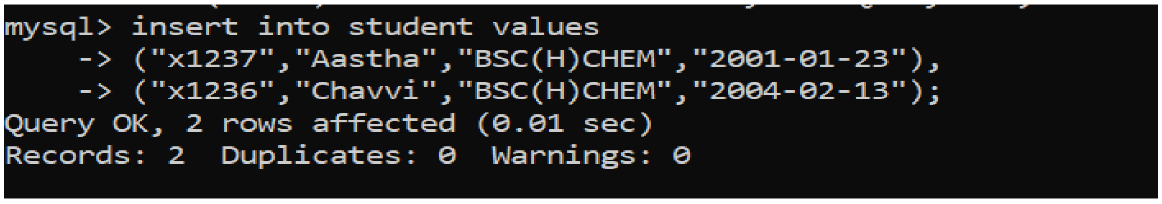
***Output:***



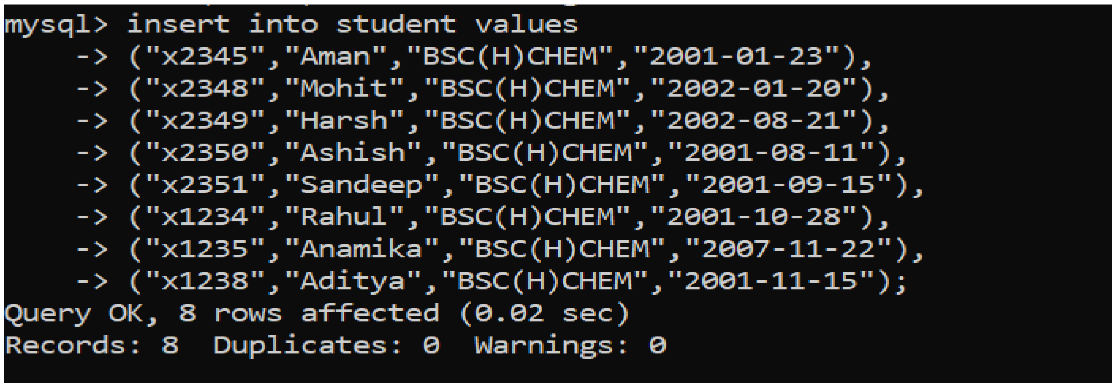
```
mysql> insert into STUDENT values(
    -> "70004","Abhishek","BSc(H)CS","2005-12-23"),
    -> ("70005","Amar","BSc(H)CS","2004-12-03"),
    -> ("70006","Akhilesh","BSc(H)CS","2004-12-03"),
    -> ("70007","Anshul","BScGeology","2004-12-03");
Query OK, 4 rows affected (0.02 sec)
Records: 4  Duplicates: 0  Warnings: 0
```



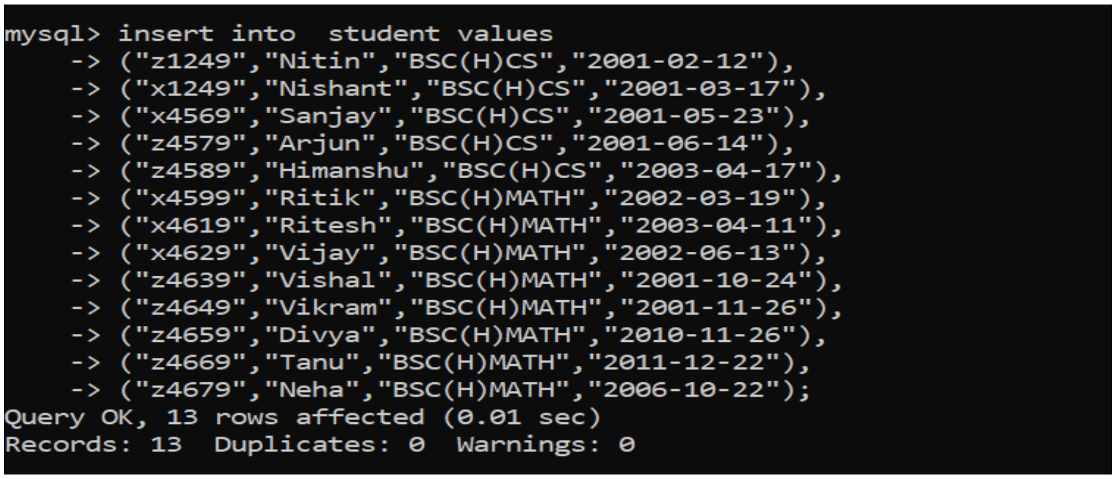
```
mysql> insert into STUDENT values
    -> ("70008","Priyam","BTech_EC","2003-02-08"),
    -> ("80008","Shreya","BA(Hons)","2053-02-15"),
    -> ("89008","Gauri","BA(Hons)","2003-09-22"),
    -> ("89508","Tulika","BSC(H)CS","2003-09-22"),
    -> ("89576","Aachmani","BA(Hons)","2005-04-12"),
    -> ("34256","Harshit","BA_LLB","2005-10-03"),
    -> ("34345","Alok","BTech_CSE","2004-09-23");
Query OK, 7 rows affected (0.04 sec)
Records: 7  Duplicates: 0  Warnings: 0
```



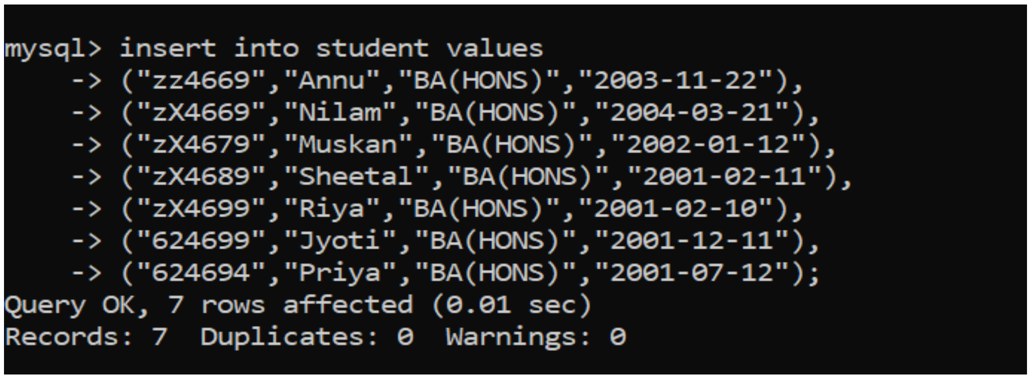
```
mysql> insert into student values
    -> ("x1237","Aastha","BSC(H)CHEM","2001-01-23"),
    -> ("x1236","Chavvi","BSC(H)CHEM","2004-02-13");
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0
```



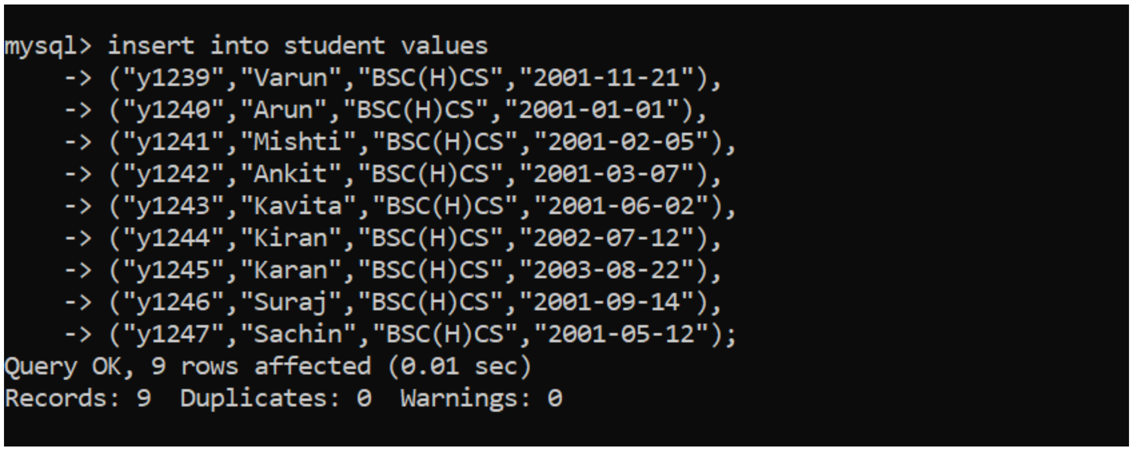
```
mysql> insert into student values
    -> ("x2345","Aman","BSC(H)CHEM","2001-01-23"),
    -> ("x2348","Mohit","BSC(H)CHEM","2002-01-20"),
    -> ("x2349","Harsh","BSC(H)CHEM","2002-08-21"),
    -> ("x2350","Ashish","BSC(H)CHEM","2001-08-11"),
    -> ("x2351","Sandeep","BSC(H)CHEM","2001-09-15"),
    -> ("x1234","Rahul","BSC(H)CHEM","2001-10-28"),
    -> ("x1235","Anamika","BSC(H)CHEM","2007-11-22"),
    -> ("x1238","Aditya","BSC(H)CHEM","2001-11-15");
Query OK, 8 rows affected (0.02 sec)
Records: 8  Duplicates: 0  Warnings: 0
```



```
mysql> insert into  student values
    -> ("z1249","Nitin","BSC(H)CS","2001-02-12"),
    -> ("x1249","Nishant","BSC(H)CS","2001-03-17"),
    -> ("x4569","Sanjay","BSC(H)CS","2001-05-23"),
    -> ("z4579","Arjun","BSC(H)CS","2001-06-14"),
    -> ("z4589","Himanshu","BSC(H)CS","2003-04-17"),
    -> ("x4599","Ritik","BSC(H)MATH","2002-03-19"),
    -> ("x4619","Ritesh","BSC(H)MATH","2003-04-11"),
    -> ("x4629","Vijay","BSC(H)MATH","2002-06-13"),
    -> ("z4639","Vishal","BSC(H)MATH","2001-10-24"),
    -> ("z4649","Vikram","BSC(H)MATH","2001-11-26"),
    -> ("z4659","Divya","BSC(H)MATH","2010-11-26"),
    -> ("z4669","Tanu","BSC(H)MATH","2011-12-22"),
    -> ("z4679","Neha","BSC(H)MATH","2006-10-22");
Query OK, 13 rows affected (0.01 sec)
Records: 13  Duplicates: 0  Warnings: 0
```



```
mysql> insert into student values
    -> ("zz4669","Annu","BA(HONS)","2003-11-22"),
    -> ("zX4669","Nilam","BA(HONS)","2004-03-21"),
    -> ("zX4679","Muskan","BA(HONS)","2002-01-12"),
    -> ("zX4689","Sheetal","BA(HONS)","2001-02-11"),
    -> ("zX4699","Riya","BA(HONS)","2001-02-10"),
    -> ("624699","Jyoti","BA(HONS)","2001-12-11"),
    -> ("624694","Priya","BA(HONS)","2001-07-12");
Query OK, 7 rows affected (0.01 sec)
Records: 7  Duplicates: 0  Warnings: 0
```



```
mysql> insert into student values
    -> ("y1239","Varun","BSC(H)CS","2001-11-21"),
    -> ("y1240","Arun","BSC(H)CS","2001-01-01"),
    -> ("y1241","Mishti","BSC(H)CS","2001-02-05"),
    -> ("y1242","Ankit","BSC(H)CS","2001-03-07"),
    -> ("y1243","Kavita","BSC(H)CS","2001-06-02"),
    -> ("y1244","Kiran","BSC(H)CS","2002-07-12"),
    -> ("y1245","Karan","BSC(H)CS","2003-08-22"),
    -> ("y1246","Suraj","BSC(H)CS","2001-09-14"),
    -> ("y1247","Sachin","BSC(H)CS","2001-05-12");
Query OK, 9 rows affected (0.01 sec)
Records: 9  Duplicates: 0  Warnings: 0
```

***Inserting values in society table:***

***Query:*** *insert into society values*

*(“123”,”Debating”,”RameshGupta”,”45”),*

*(“223”,”Dancing”,”AakashSingh”,”65”),*

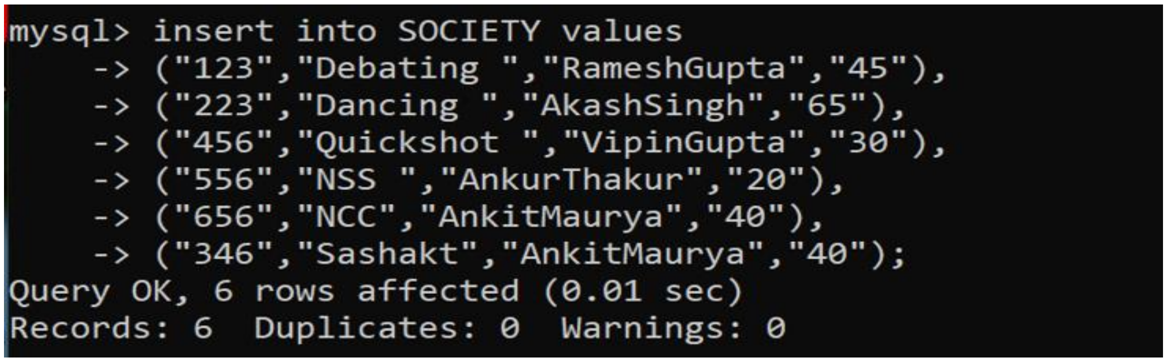
*(“456”,”Quicksort”,”VipinGupta”,”30”),*

*(“556”,”NSS”,”AnkurThakur”,”20”),*

*(“656”,”NCC”,”AnkitMaurya”,”40”),*

*(“346”,”Sashakt”,”AnkitMaurya”,”40”);*

***Output:***



```
mysql> insert into SOCIETY values
    -> ("123","Debating ","RameshGupta","45"),
    -> ("223","Dancing ","AkashSingh","65"),
    -> ("456","Quickshot ","VipinGupta","30"),
    -> ("556","NSS ","AnkurThakur","20"),
    -> ("656","NCC","AnkitMaurya","40"),
    -> ("346","Sashakt","AnkitMaurya","40");
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

***Inserting values in Enrollment table:***

***Query:*** *insert into Enrollment values*

*(“70005”,”223”,”2024-03-23”),*

*(“70006”,”456”,”2024-04-28”),*

*(“70007”,”556”,”2024-01-30”),*

*(“70008”,”656”,”2024-02-28”),*

*(“80008”,”346”,”2024-02-20”),*

*(“89008”,”123”,”2024-02-25”),*

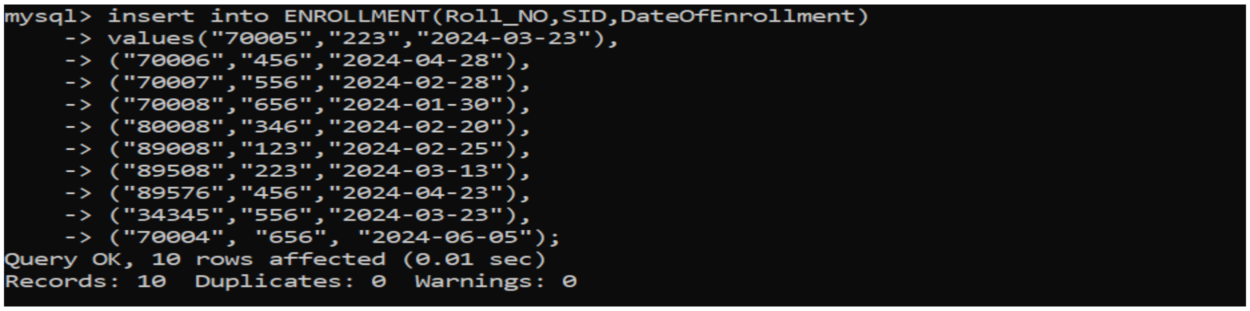
*(“89508”,”223”,”2024-03-13”),*

*(“89576”,”456”,”2024-04-23”),*

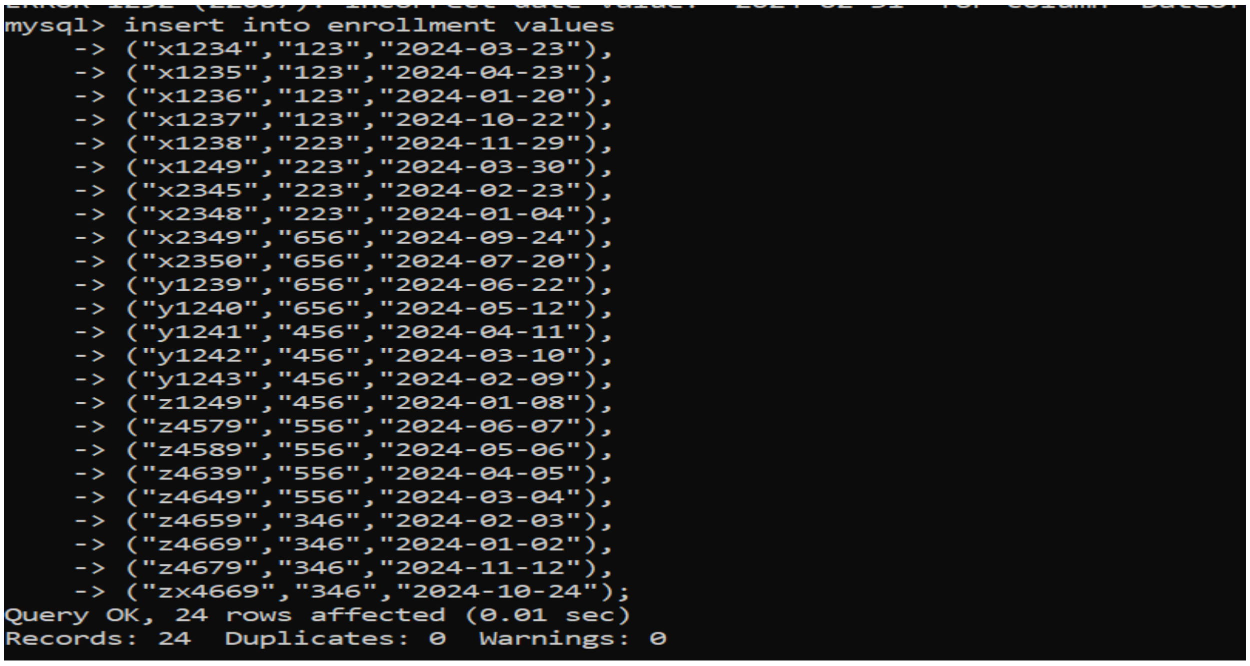
*(“34345”,”556”,”2024-03-23”),*

*(“70004”,”656”,”2024-06-05”);*

***Output:***



```
mysql> insert into ENROLLMENT(Roll_NO,SID,DateOfEnrollment)
    -> values("70005","223","2024-03-23"),
    -> ("70006","456","2024-04-28"),
    -> ("70007","556","2024-02-28"),
    -> ("70008","656","2024-01-30"),
    -> ("80008","346","2024-02-20"),
    -> ("89008","123","2024-02-25"),
    -> ("89508","223","2024-03-13"),
    -> ("89576","456","2024-04-23"),
    -> ("34345","556","2024-03-23"),
    -> ("70004", "656", "2024-06-05");
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

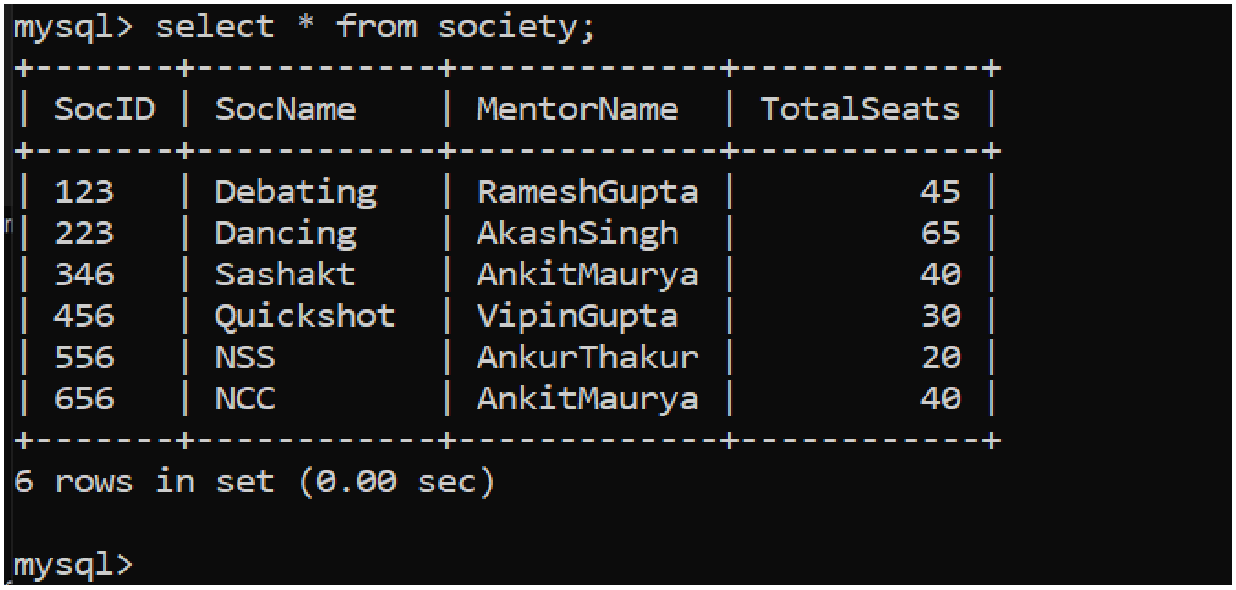


```
mysql> insert into enrollment values
    -> ("x1234","123","2024-03-23"),
    -> ("x1235","123","2024-04-23"),
    -> ("x1236","123","2024-01-20"),
    -> ("x1237","123","2024-10-22"),
    -> ("x1238","223","2024-11-29"),
    -> ("x1249","223","2024-03-30"),
    -> ("x2345","223","2024-02-23"),
    -> ("x2348","223","2024-01-04"),
    -> ("x2349","656","2024-09-24"),
    -> ("x2350","656","2024-07-20"),
    -> ("y1239","656","2024-06-22"),
    -> ("y1240","656","2024-05-12"),
    -> ("y1241","456","2024-04-11"),
    -> ("y1242","456","2024-03-10"),
    -> ("y1243","456","2024-02-09"),
    -> ("z1249","456","2024-01-08"),
    -> ("z4579","556","2024-06-07"),
    -> ("z4589","556","2024-05-06"),
    -> ("z4639","556","2024-04-05"),
    -> ("z4649","556","2024-03-04"),
    -> ("z4659","346","2024-02-03"),
    -> ("z4669","346","2024-01-02"),
    -> ("z4679","346","2024-11-12"),
    -> ("zx4669","346","2024-10-24");
Query OK, 24 rows affected (0.01 sec)
Records: 24  Duplicates: 0  Warnings: 0
```

## ***Checking Entries of each table:***

***Query:*** *select*from society;*

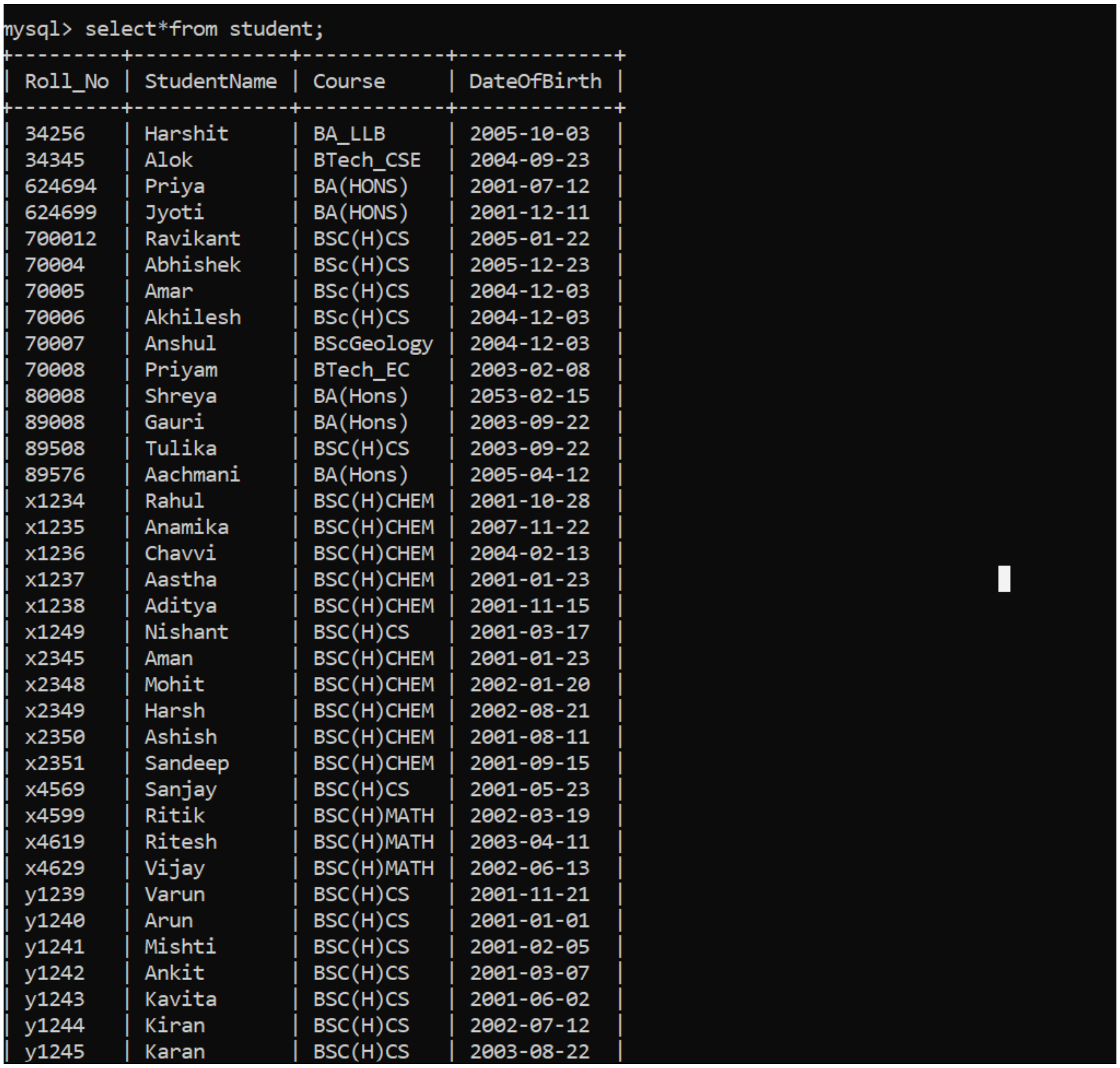
***Output:***



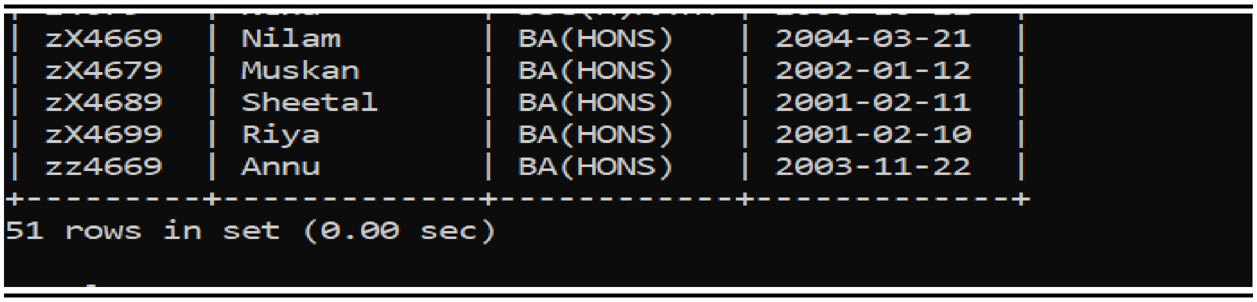
```
mysql> select * from society;
+-------+-----------+------------+------------+
| SocID | SocName   | MentorName | TotalSeats |
+-------+-----------+------------+------------+
| 123   | Debating  | RameshGupta |        45 |
| 223   | Dancing   | AkashSingh |         65 |
| 346   | Sashakt   | AnkitMaurya |        40 |
| 456   | Quickshot | VipinGupta |         30 |
| 556   | NSS       | AnkurThakur |        20 |
| 656   | NCC       | AnkitMaurya |        40 |
+-------+-----------+------------+------------+
6 rows in set (0.00 sec)

mysql>
```

**Query:** *select*from student;*

***Output:***



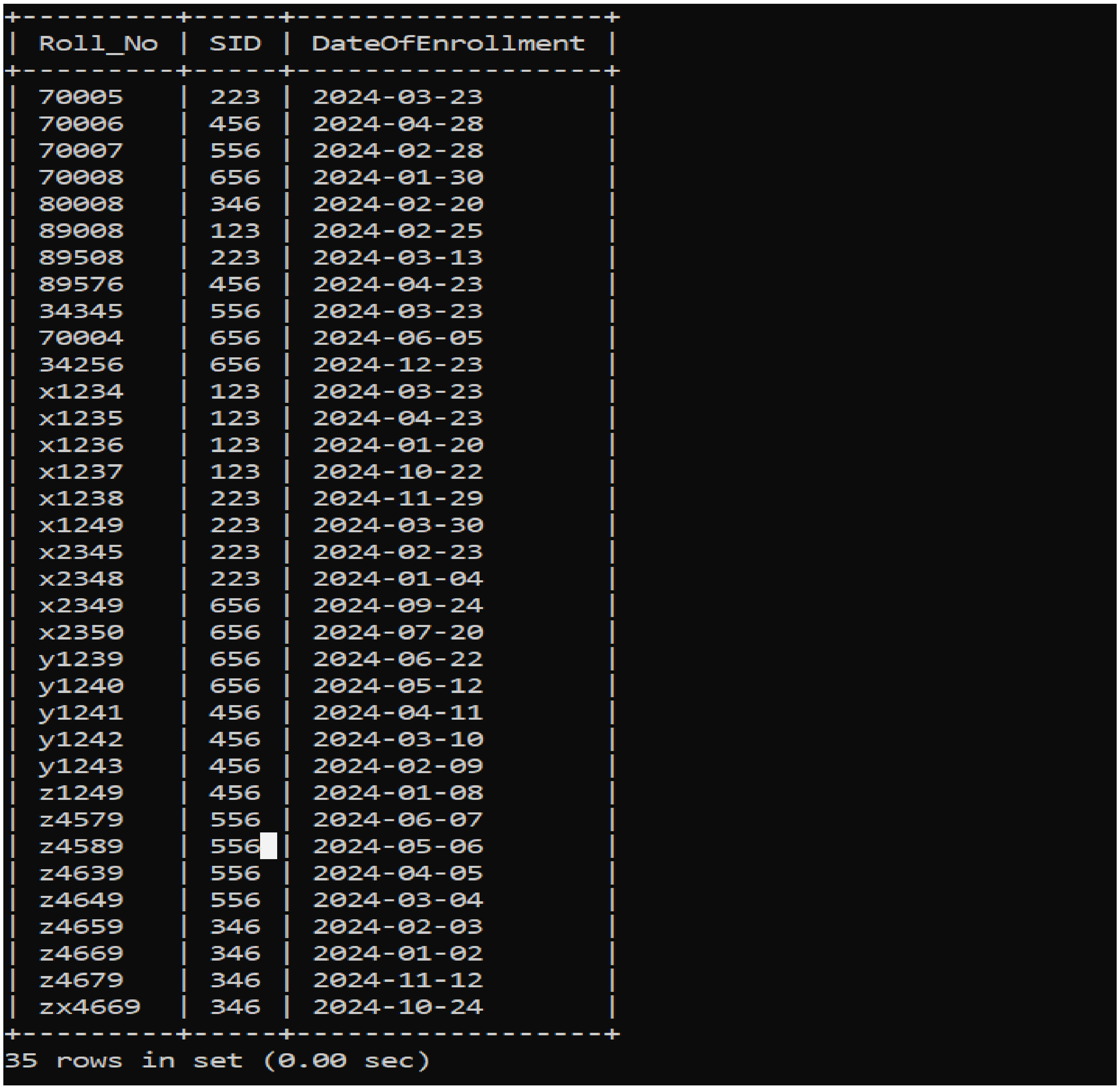
```
mysql> select*from student;
+---------+-------------+------------+------------+
| Roll_No | StudentName | Course     | DateOfBirth |
+---------+-------------+------------+------------+
| 34256   | Harshit     | BA_LLB     | 2005-10-03 |
| 34345   | Alok        | BTech_CSE  | 2004-09-23 |
| 624694  | Priya       | BA(HONS)   | 2001-07-12 |
| 624699  | Jyoti       | BA(HONS)   | 2001-12-11 |
| 700012  | Ravikant    | BSC(H)CS   | 2005-01-22 |
| 70004   | Abhishek    | BSc(H)CS   | 2005-12-23 |
| 70005   | Amar        | BSc(H)CS   | 2004-12-03 |
| 70006   | Akhilesh    | BSc(H)CS   | 2004-12-03 |
| 70007   | Anshul      | BScGeology | 2004-12-03 |
| 70008   | Priyam      | BTech_EC   | 2003-02-08 |
| 80008   | Shreya      | BA(Hons)   | 2053-02-15 |
| 89008   | Gauri       | BA(Hons)   | 2003-09-22 |
| 89508   | Tulika      | BSC(H)CS   | 2003-09-22 |
| 89576   | Aachmani    | BA(Hons)   | 2005-04-12 |
| x1234   | Rahul       | BSC(H)CHEM | 2001-10-28 |
| x1235   | Anamika     | BSC(H)CHEM | 2007-11-22 |
| x1236   | Chavvi      | BSC(H)CHEM | 2004-02-13 |
| x1237   | Aastha      | BSC(H)CHEM | 2001-01-23 |
| x1238   | Aditya      | BSC(H)CHEM | 2001-11-15 |
| x1249   | Nishant     | BSC(H)CS   | 2001-03-17 |
| x2345   | Aman        | BSC(H)CHEM | 2001-01-23 |
| x2348   | Mohit       | BSC(H)CHEM | 2002-01-20 |
| x2349   | Harsh       | BSC(H)CHEM | 2002-08-21 |
| x2350   | Ashish      | BSC(H)CHEM | 2001-08-11 |
| x2351   | Sandeep     | BSC(H)CHEM | 2001-09-15 |
| x4569   | Sanjay      | BSC(H)CS   | 2001-05-23 |
| x4599   | Ritik       | BSC(H)MATH | 2002-03-19 |
| x4619   | Ritesh      | BSC(H)MATH | 2003-04-11 |
| x4629   | Vijay       | BSC(H)MATH | 2002-06-13 |
| y1239   | Varun       | BSC(H)CS   | 2001-11-21 |
| y1240   | Arun        | BSC(H)CS   | 2001-01-01 |
| y1241   | Mishti      | BSC(H)CS   | 2001-02-05 |
| y1242   | Ankit       | BSC(H)CS   | 2001-03-07 |
| y1243   | Kavita      | BSC(H)CS   | 2001-06-02 |
| y1244   | Kiran       | BSC(H)CS   | 2002-07-12 |
| y1245   | Karan       | BSC(H)CS   | 2003-08-22 |
```



```
| zX4669  | Nilam       | BA(HONS)   | 2004-03-21 |
| zX4679  | Muskan      | BA(HONS)   | 2002-01-12 |
| zX4689  | Sheetal     | BA(HONS)   | 2001-02-11 |
| zX4699  | Riya        | BA(HONS)   | 2001-02-10 |
| zz4669  | Annu        | BA(HONS)   | 2003-11-22 |
+---------+-------------+------------+------------+
51 rows in set (0.00 sec)
```

***Query:*** *select*from enrollment;*

***Output:***



```
+---------+-----+------------------+
| Roll_No | SID | DateOfEnrollment |
+---------+-----+------------------+
| 70005   | 223 | 2024-03-23       |
| 70006   | 456 | 2024-04-28       |
| 70007   | 556 | 2024-02-28       |
| 70008   | 656 | 2024-01-30       |
| 80008   | 346 | 2024-02-20       |
| 89008   | 123 | 2024-02-25       |
| 89508   | 223 | 2024-03-13       |
| 89576   | 456 | 2024-04-23       |
| 34345   | 556 | 2024-03-23       |
| 70004   | 656 | 2024-06-05       |
| 34256   | 656 | 2024-12-23       |
| x1234   | 123 | 2024-03-23       |
| x1235   | 123 | 2024-04-23       |
| x1236   | 123 | 2024-01-20       |
| x1237   | 123 | 2024-10-22       |
| x1238   | 223 | 2024-11-29       |
| x1249   | 223 | 2024-03-30       |
| x2345   | 223 | 2024-02-23       |
| x2348   | 223 | 2024-01-04       |
| x2349   | 656 | 2024-09-24       |
| x2350   | 656 | 2024-07-20       |
| y1239   | 656 | 2024-06-22       |
| y1240   | 656 | 2024-05-12       |
| y1241   | 456 | 2024-04-11       |
| y1242   | 456 | 2024-03-10       |
| y1243   | 456 | 2024-02-09       |
| z1249   | 456 | 2024-01-08       |
| z4579   | 556 | 2024-06-07       |
| z4589   | 556 | 2024-05-06       |
| z4639   | 556 | 2024-04-05       |
| z4649   | 556 | 2024-03-04       |
| z4659   | 346 | 2024-02-03       |
| z4669   | 346 | 2024-01-02       |
| z4679   | 346 | 2024-11-12       |
| zx4669  | 346 | 2024-10-24       |
+---------+-----+------------------+
35 rows in set (0.00 sec)
```

## (1.) ***Retrieve names of students enrolled in any society***

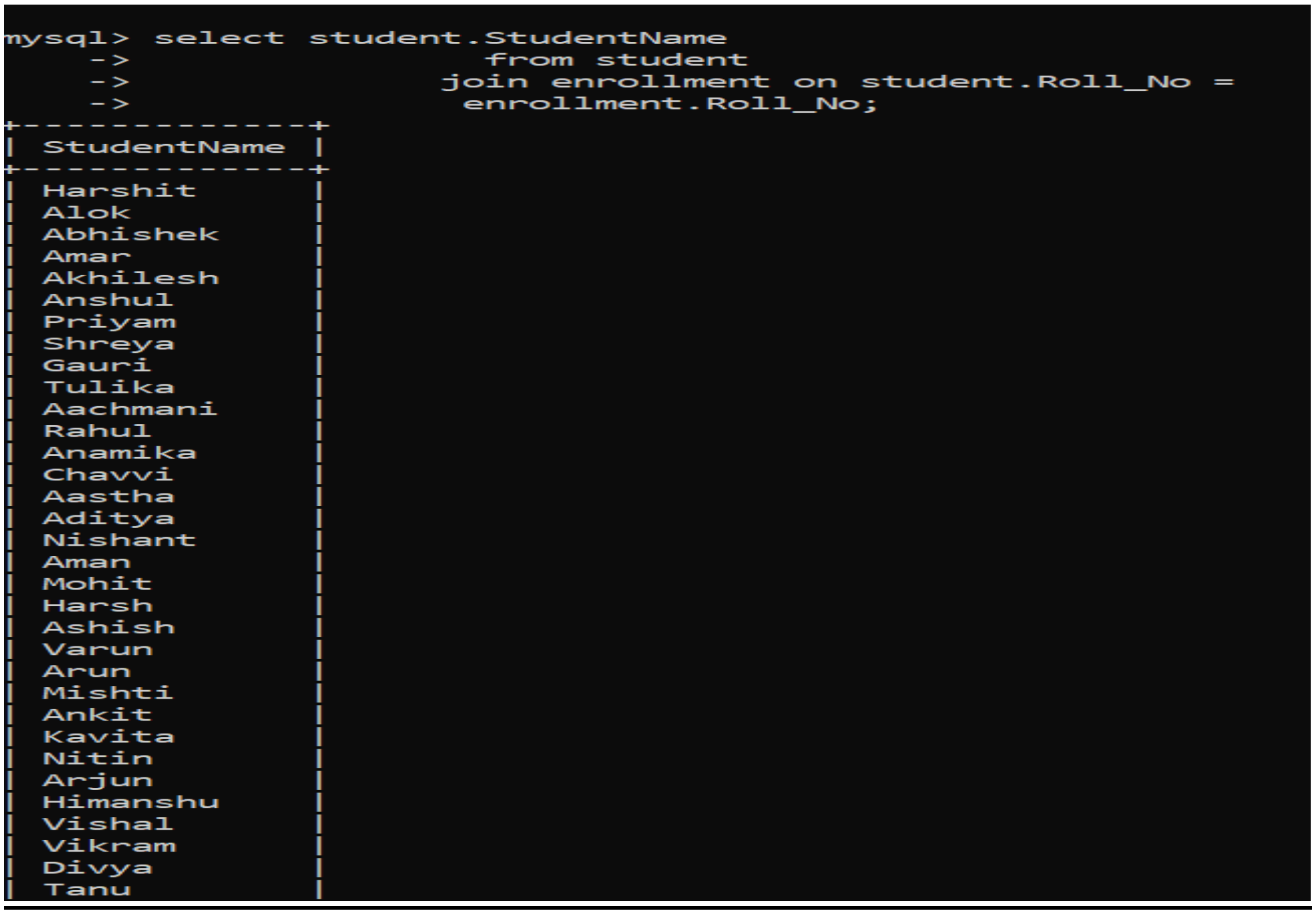
***Query:*** *select student.StudentName*

*from student*

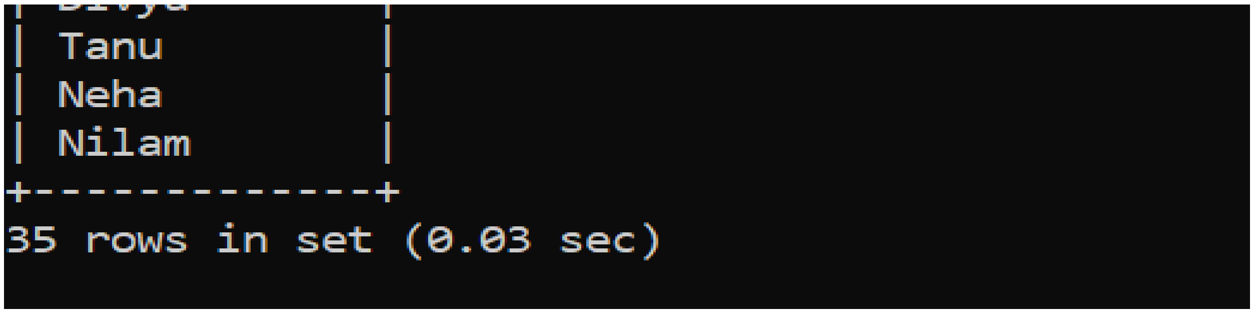
*join enrollment on student.Roll_No =*

*enrollment.Roll_No;*

***Output:***



```
mysql> select student.StudentName
    ->              from student
    ->             join enrollment on student.Roll_No =
    ->              enrollment.Roll_No;
+-------------+
| StudentName |
+-------------+
| Harshit     |
| Alok        |
| Abhishek    |
| Amar        |
| Akhilesh    |
| Anshul      |
| Priyam      |
| Shreya      |
| Gauri       |
| Tulika      |
| Aachmani    |
| Rahul       |
| Anamika     |
| Chavvi      |
| Aastha      |
| Aditya      |
| Nishant     |
| Aman        |
| Mohit       |
| Harsh       |
| Ashish      |
| Varun       |
| Arun        |
| Mishti      |
| Ankit       |
| Kavita      |
| Nitin       |
| Arjun       |
| Himanshu    |
| Vishal      |
| Vikram      |
| Divya       |
| Tanu        |
```



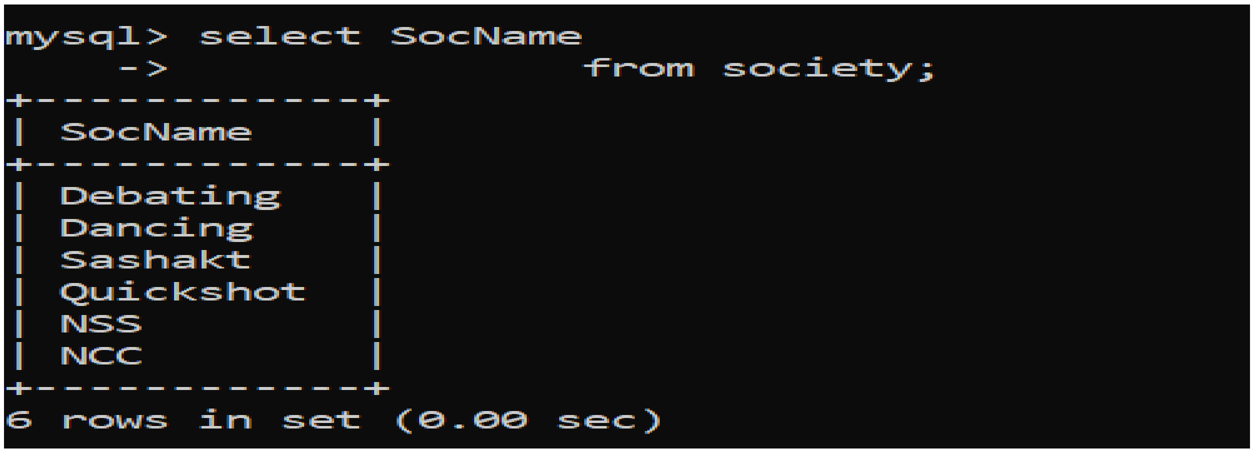
```
| Divya        |
| Tanu         |
| Neha         |
| Nilam        |
+-------------+
35 rows in set (0.03 sec)
```

## (2.) ***Retrieve all society names.***

***Query:*** *select SocName*

*from society;*

***Output:***



```
mysql> select SocName
    ->                    from society;
+-------------+
| SocName     |
+-------------+
| Debating    |
| Dancing     |
| Sashakt     |
| Quickshot   |
| NSS         |
| NCC         |
+-------------+
6 rows in set (0.00 sec)
```

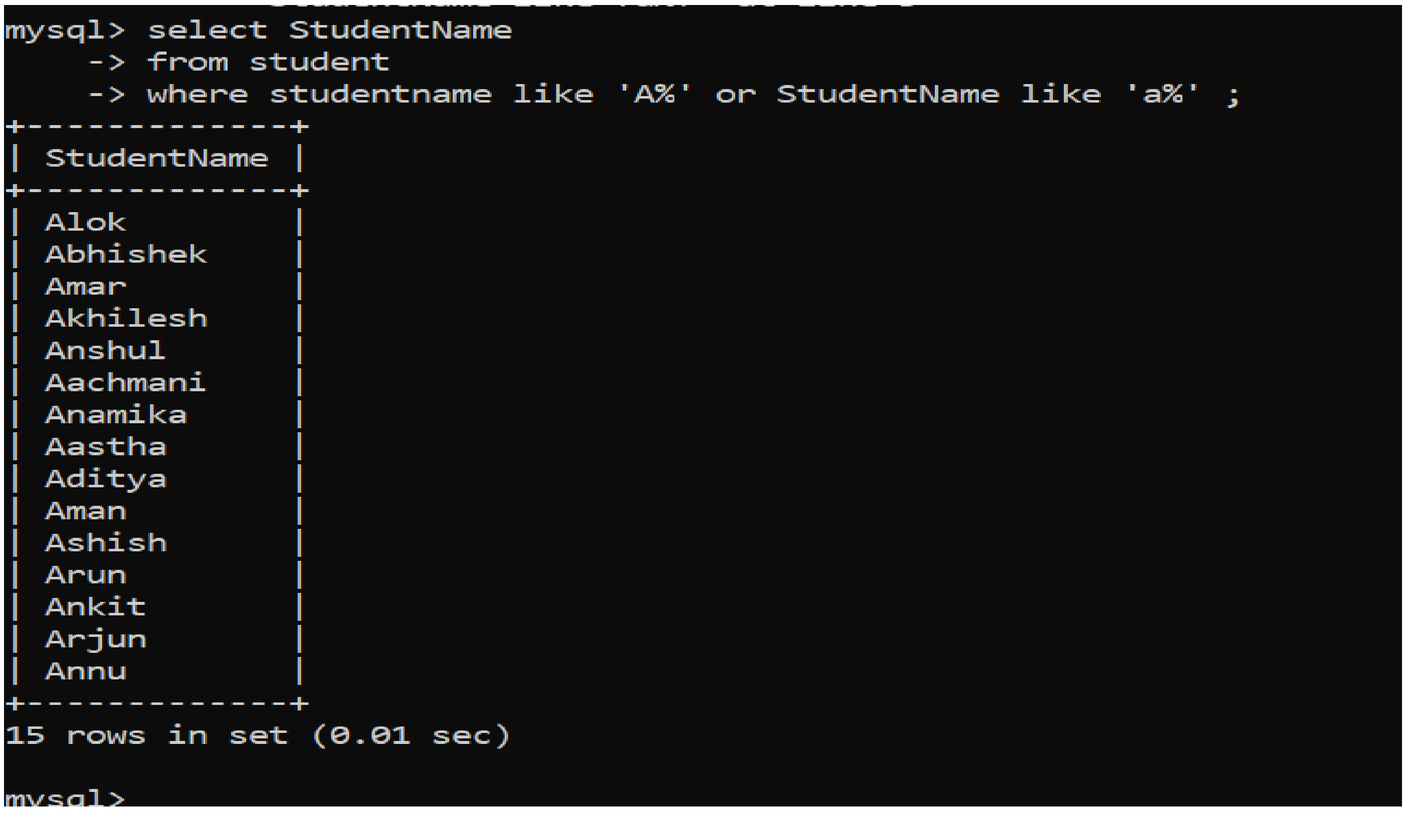
## (3.) ***Retrieve students' names starting with letter ‘A’.***

***Query:*** *select StudentName*

*from student*

*where StudentName like ‘A%’ or StudentName like ‘a%’;*

**Output:**



```
mysql> select StudentName
    -> from student
    -> where studentname like 'A%' or StudentName like 'a%' ;
+-------------+
| StudentName |
+-------------+
| Alok        |
| Abhishek    |
| Amar        |
| Akhilesh    |
| Anshul      |
| Aachmani    |
| Anamika     |
| Aastha      |
| Aditya      |
| Aman        |
| Ashish      |
| Arun        |
| Ankit       |
| Arjun       |
| Annu        |
+-------------+
15 rows in set (0.01 sec)

mysql>
```

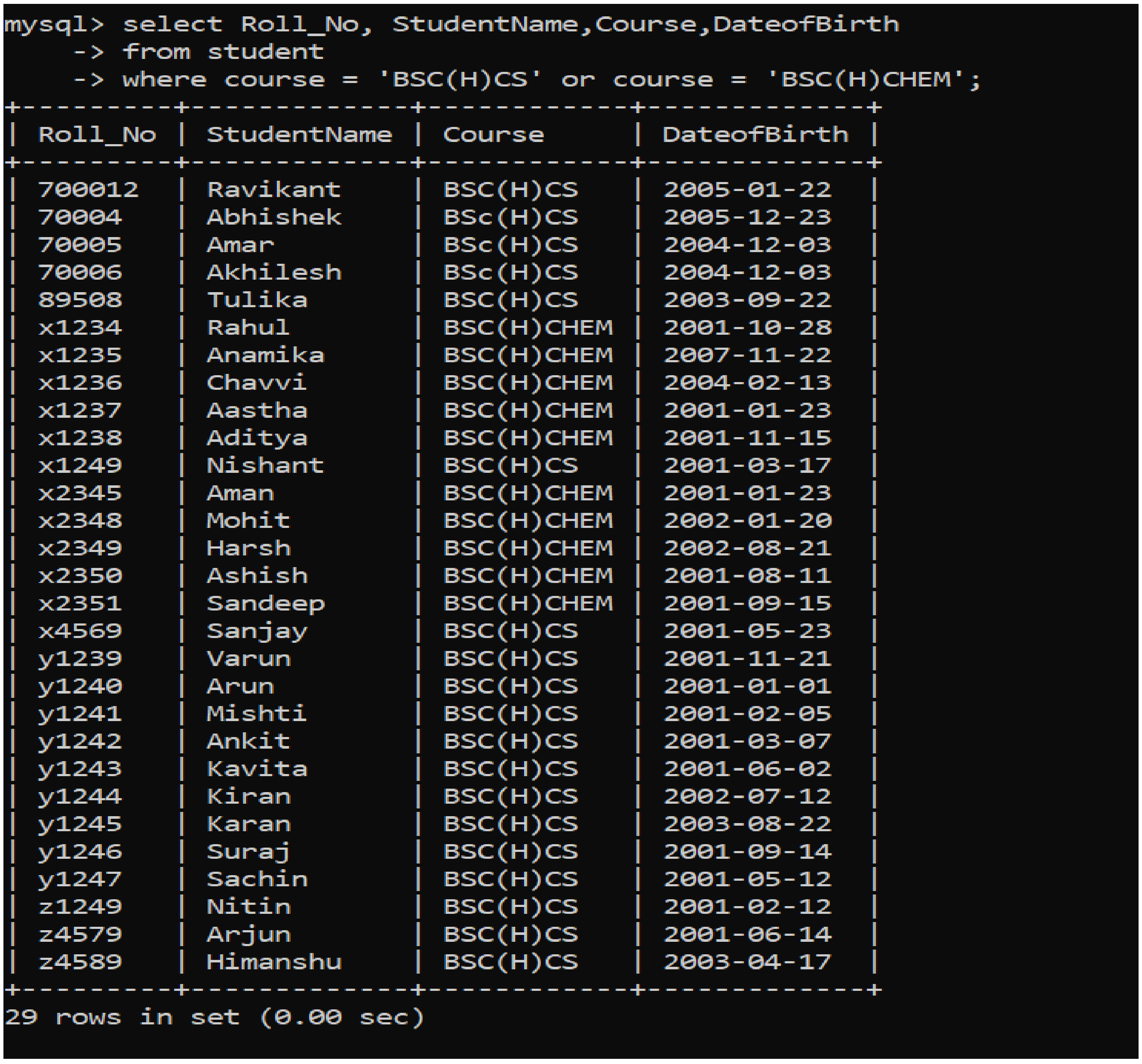
## (4). Retrieve students' details studying in courses ‘computer science’ or ‘chemistry’.

**Query:-** *select Roll_No,StudentName,Course,DateofBirth*

*from student*

*where Course=‘BSC(H)CS’ or course=‘BSC(H)CHEM’ ;*

**Output:-**


```
mysql> select Roll_No, StudentName,Course,DateofBirth
    -> from student
    -> where course = 'BSC(H)CS' or course = 'BSC(H)CHEM';
+---------+-------------+------------+-------------+
| Roll_No | StudentName | Course     | DateofBirth |
+---------+-------------+------------+-------------+
| 700012  | Ravikant    | BSC(H)CS   | 2005-01-22  |
| 70004   | Abhishek    | BSc(H)CS   | 2005-12-23  |
| 70005   | Amar        | BSc(H)CS   | 2004-12-03  |
| 70006   | Akhilesh    | BSc(H)CS   | 2004-12-03  |
| 89508   | Tulika      | BSC(H)CS   | 2003-09-22  |
| x1234   | Rahul       | BSC(H)CHEM | 2001-10-28  |
| x1235   | Anamika     | BSC(H)CHEM | 2007-11-22  |
| x1236   | Chavvi      | BSC(H)CHEM | 2004-02-13  |
| x1237   | Aastha      | BSC(H)CHEM | 2001-01-23  |
| x1238   | Aditya      | BSC(H)CHEM | 2001-11-15  |
| x1249   | Nishant     | BSC(H)CS   | 2001-03-17  |
| x2345   | Aman        | BSC(H)CHEM | 2001-01-23  |
| x2348   | Mohit       | BSC(H)CHEM | 2002-01-20  |
| x2349   | Harsh       | BSC(H)CHEM | 2002-08-21  |
| x2350   | Ashish      | BSC(H)CHEM | 2001-08-11  |
| x2351   | Sandeep     | BSC(H)CHEM | 2001-09-15  |
| x4569   | Sanjay      | BSC(H)CS   | 2001-05-23  |
| y1239   | Varun       | BSC(H)CS   | 2001-11-21  |
| y1240   | Arun        | BSC(H)CS   | 2001-01-01  |
| y1241   | Mishti      | BSC(H)CS   | 2001-02-05  |
| y1242   | Ankit       | BSC(H)CS   | 2001-03-07  |
| y1243   | Kavita      | BSC(H)CS   | 2001-06-02  |
| y1244   | Kiran       | BSC(H)CS   | 2002-07-12  |
| y1245   | Karan       | BSC(H)CS   | 2003-08-22  |
| y1246   | Suraj       | BSC(H)CS   | 2001-09-14  |
| y1247   | Sachin      | BSC(H)CS   | 2001-05-12  |
| z1249   | Nitin       | BSC(H)CS   | 2001-02-12  |
| z4579   | Arjun       | BSC(H)CS   | 2001-06-14  |
| z4589   | Himanshu    | BSC(H)CS   | 2003-04-17  |
+---------+-------------+------------+-------------+
29 rows in set (0.00 sec)
```

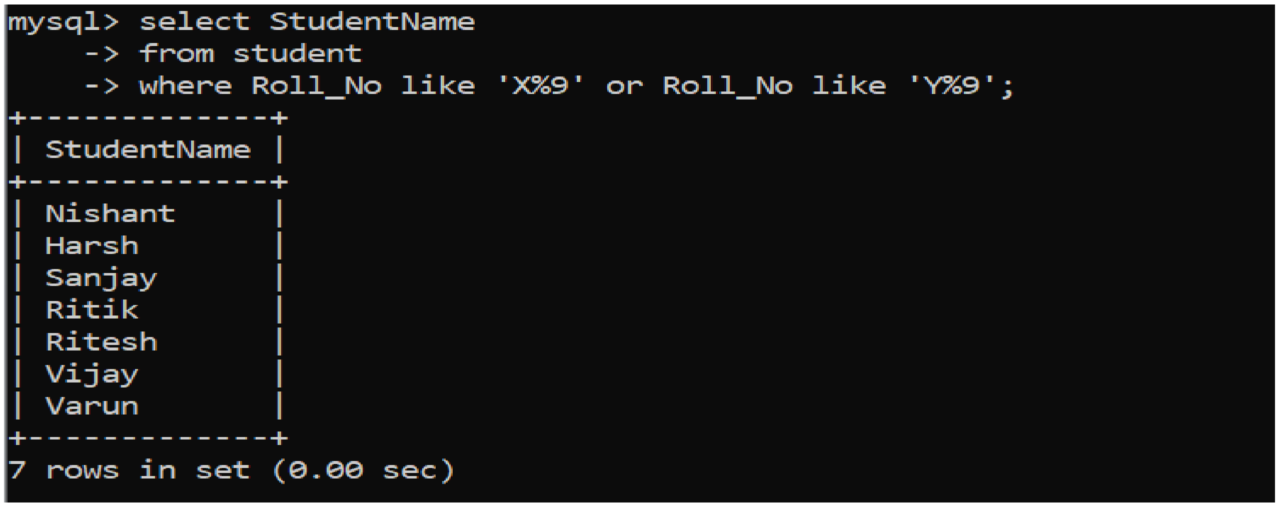
***(5.)Retrieve students' names whose roll no either starts with 'X' or 'Z' and ends with '9'.***

***Query:-*** *select StudentName*

*from student*

*where Roll_No like ‘X%9’ or Roll_No like ‘Y%9’;*

**Output:-**



```
mysql> select StudentName
    -> from student
    -> where Roll_No like 'X%9' or Roll_No like 'Y%9';
+-------------+
| StudentName |
+-------------+
| Nishant     |
| Harsh       |
| Sanjay      |
| Ritik       |
| Ritesh      |
| Vijay       |
| Varun       |
+-------------+
7 rows in set (0.00 sec)
```

## (6.) Find society details with more than N TotalSeats where N is to be input by the user.

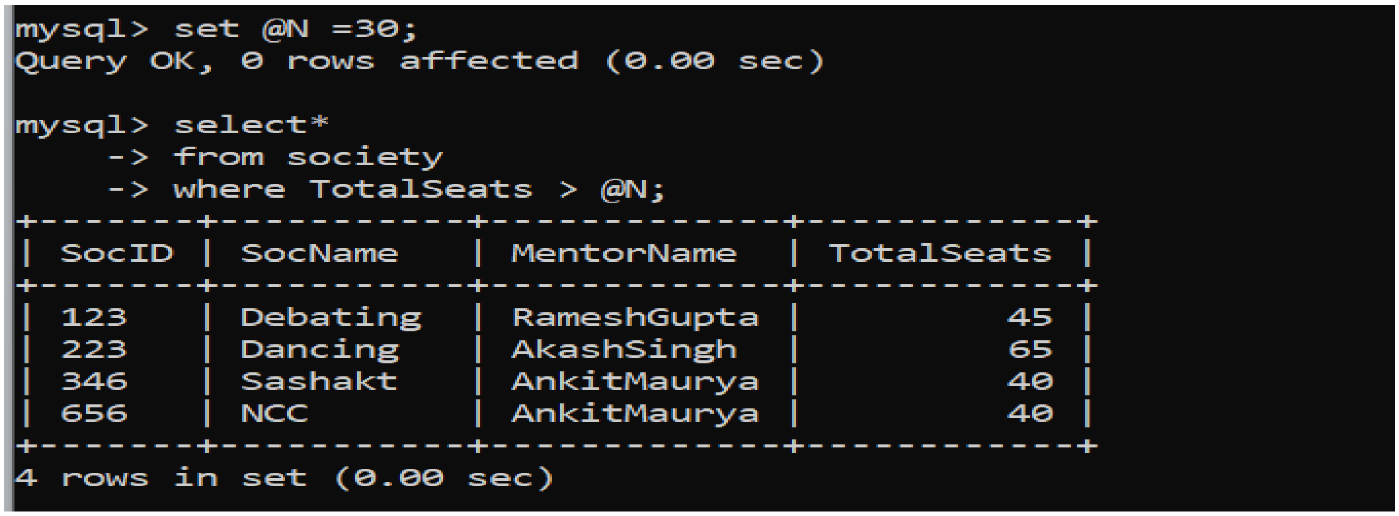
**Query:** *set @N = 30;*

*Select **

*from society*

*where TotalSeats > @N;*

**Output:**



```
mysql> set @N =30;
Query OK, 0 rows affected (0.00 sec)

mysql> select*
    -> from society
    -> where TotalSeats > @N;
+-------+-----------+------------+------------+
| SocID | SocName   | MentorName | TotalSeats |
+-------+-----------+------------+------------+
| 123   | Debating  | RameshGupta |        45 |
| 223   | Dancing   | AkashSingh  |        65 |
| 346   | Sashakt   | AnkitMaurya |        40 |
| 656   | NCC       | AnkitMaurya |        40 |
+-------+-----------+------------+------------+
4 rows in set (0.00 sec)
```

***<u>(7.) Update society table for mentor name of a specific society.</u>***

***<u>Query:-</u>*** *select* from society;*

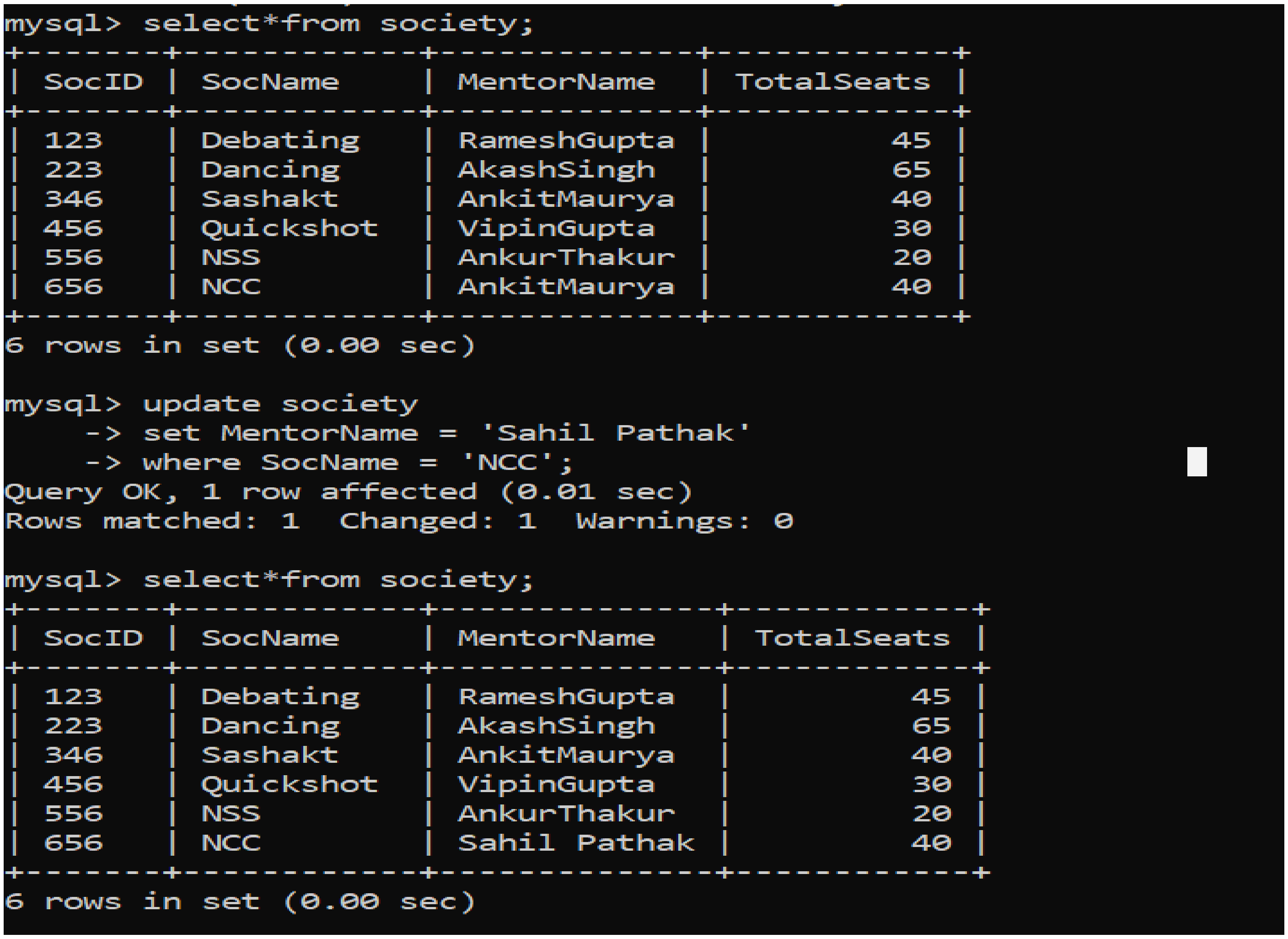
*update society*

*set MentorName = 'Sahil Pathak'*

*where SocName = 'NCC';*

*select * from society;*

***<u>Output:-</u>***



```
mysql> select*from society;
+-------+-----------+------------+------------+
| SocID | SocName   | MentorName | TotalSeats |
+-------+-----------+------------+------------+
| 123   | Debating  | RameshGupta |        45 |
| 223   | Dancing   | AkashSingh  |        65 |
| 346   | Sashakt   | AnkitMaurya |        40 |
| 456   | Quickshot | VipinGupta  |        30 |
| 556   | NSS       | AnkurThakur |        20 |
| 656   | NCC       | AnkitMaurya |        40 |
+-------+-----------+------------+------------+
6 rows in set (0.00 sec)

mysql> update society
    -> set MentorName = 'Sahil Pathak'
    -> where SocName = 'NCC';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select*from society;
+-------+-----------+--------------+------------+
| SocID | SocName   | MentorName   | TotalSeats |
+-------+-----------+--------------+------------+
| 123   | Debating  | RameshGupta  |         45 |
| 223   | Dancing   | AkashSingh   |         65 |
| 346   | Sashakt   | AnkitMaurya  |         40 |
| 456   | Quickshot | VipinGupta   |         30 |
| 556   | NSS       | AnkurThakur  |         20 |
| 656   | NCC       | Sahil Pathak |         40 |
+-------+-----------+--------------+------------+
6 rows in set (0.00 sec)
```

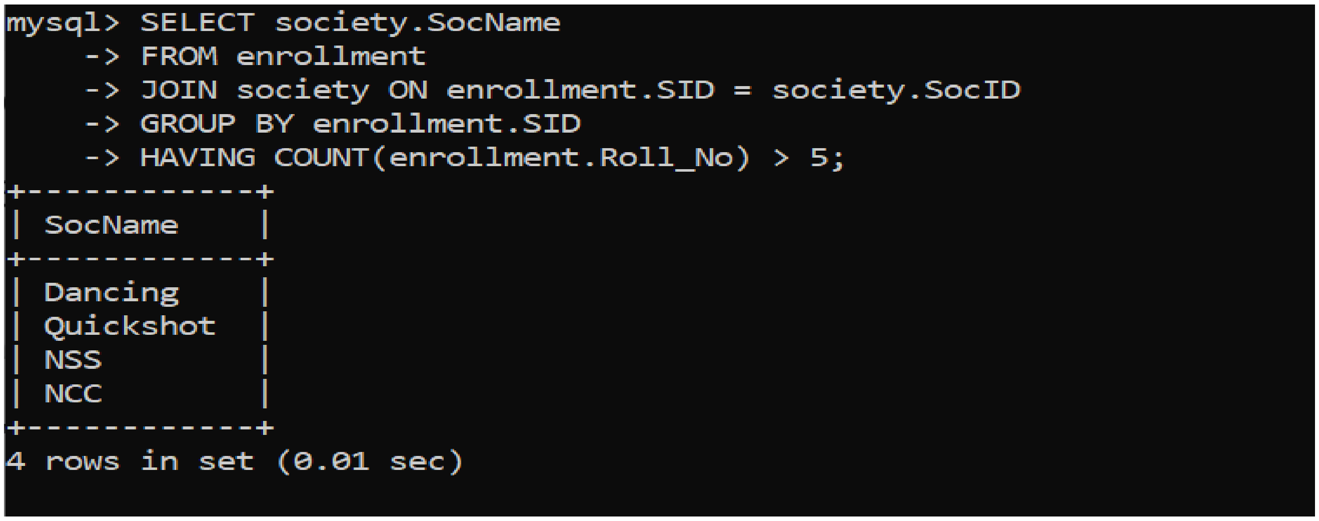
## ***<u>(8.) Find society names in which more than five students have enrolled .</u>***

***<u>Query:-</u>*** *SELECT society.SocName*

```
FROM enrollment
JOIN society ON enrollment.SID = society.SocID
GROUP BY enrollment.SID
```

*HAVING COUNT(enrollment.Roll_No) > 5;*

**Output:-**



```
mysql> SELECT society.SocName
    -> FROM enrollment
    -> JOIN society ON enrollment.SID = society.SocID
    -> GROUP BY enrollment.SID
    -> HAVING COUNT(enrollment.Roll_No) > 5;
+------------+
| SocName    |
+------------+
| Dancing    |
| Quickshot  |
| NSS        |
| NCC        |
+------------+
4 rows in set (0.01 sec)
```

## (9.) Find the name of youngest student enrolled in society 'NSS' .

**Query:-** *select StudentName*

*from student*

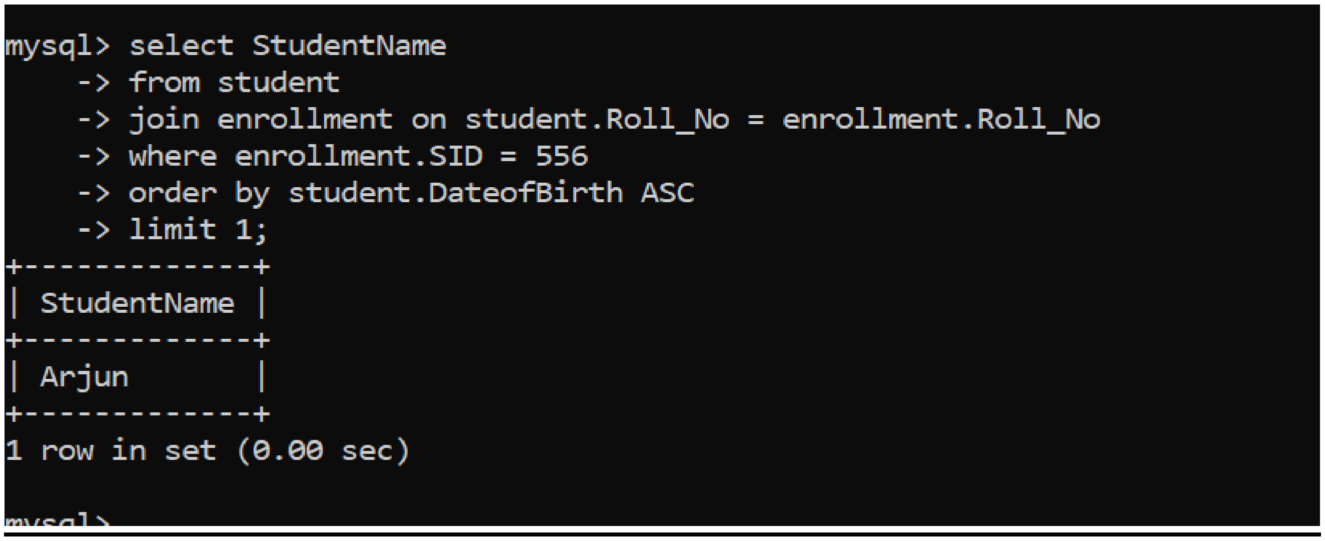
*join enrollment on student.Roll_No= enrollment.Roll_No*

*where enrollment.SID = 556*

*order by student.DateofBirth ASC*

*limit 1;*

**Output:-**



```
mysql> select StudentName
    -> from student
    -> join enrollment on student.Roll_No = enrollment.Roll_No
    -> where enrollment.SID = 556
    -> order by student.DateofBirth ASC
    -> limit 1;
+-------------+
| StudentName |
+-------------+
| Arjun       |
+-------------+
1 row in set (0.00 sec)

mysql>
```

## ***(10.) Find the name of most popular society (on the basis of enrolled students).***

***Query:-*** *select SocName*

*from society*

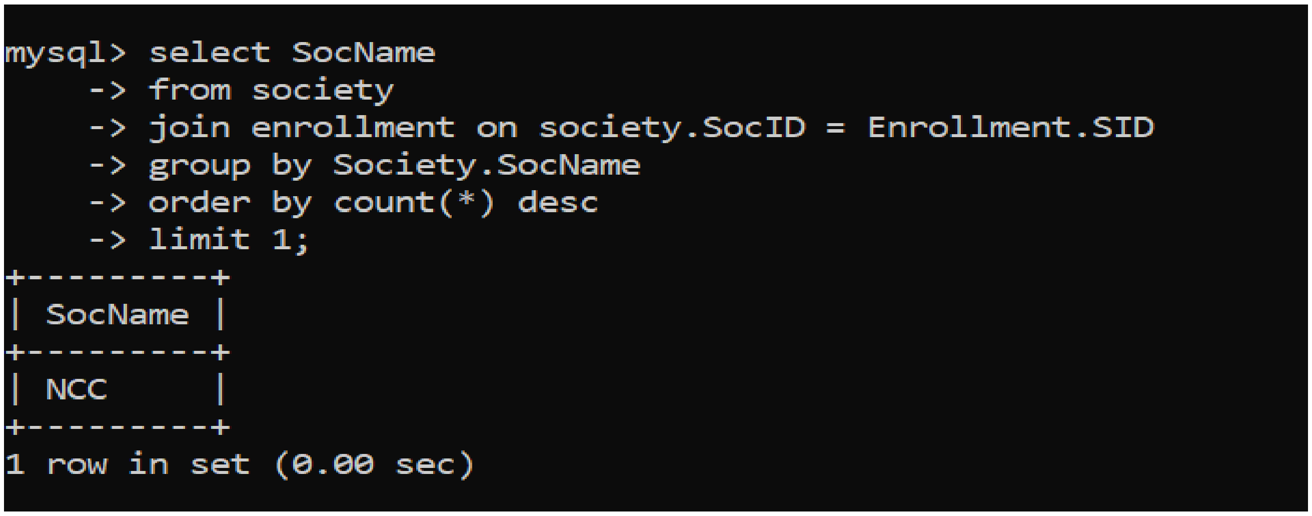
*join enrollment on society.SocID = Enrollment.SID*

*group by Society.SocName*

*order by count(*) desc*

*limit 1;*

***Output:-***



```
mysql> select SocName
    -> from society
    -> join enrollment on society.SocID = Enrollment.SID
    -> group by Society.SocName
    -> order by count(*) desc
    -> limit 1;
+---------+
| SocName |
+---------+
| NCC     |
+---------+
1 row in set (0.00 sec)
```

## ***(11.) Find the name of two least popular societies (on the basis of enrolled students).***

***Query:-*** *select society.SocName*

*from society*

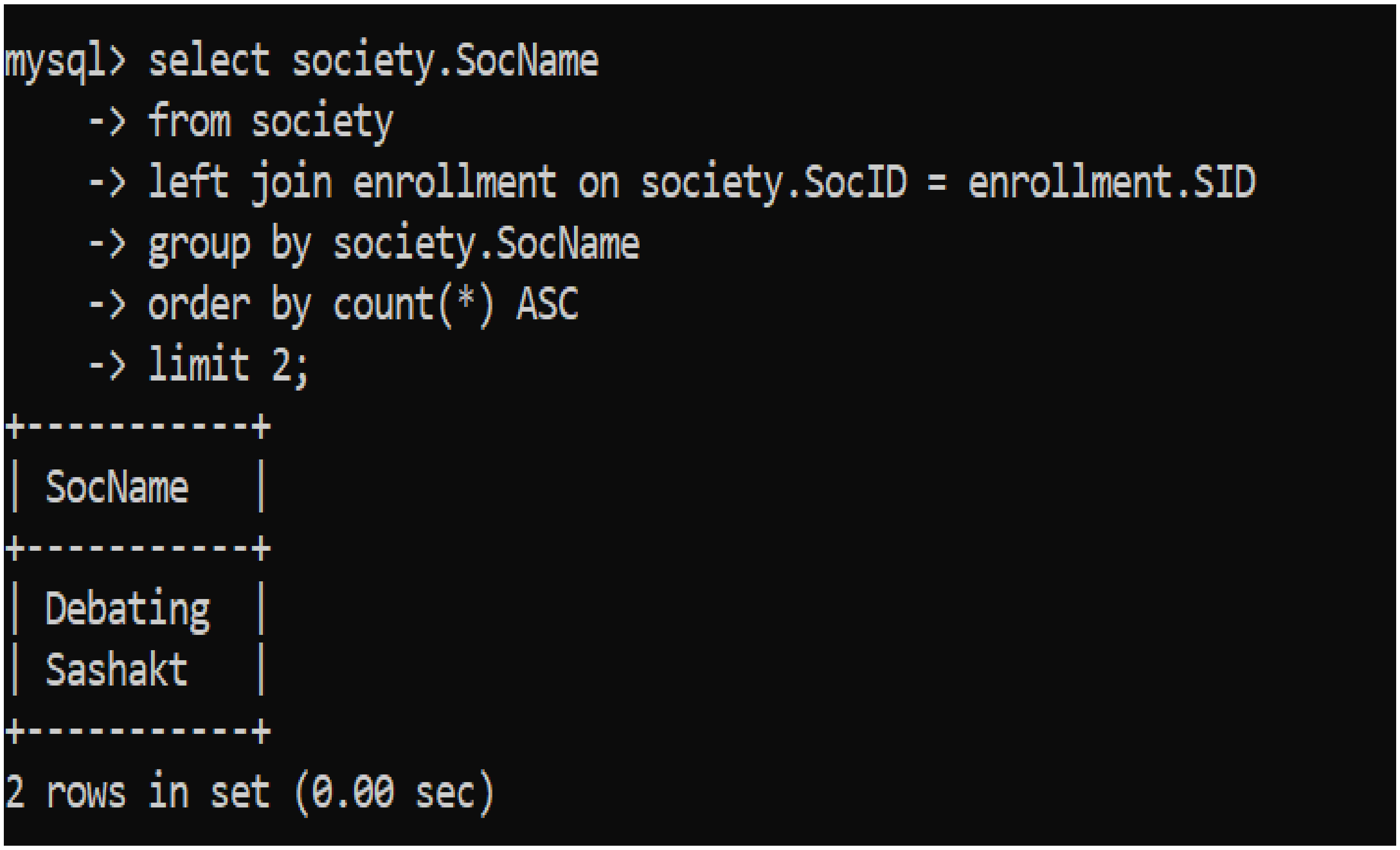
*left join enrollment on society.SocID = enrollment.SID*

*group by society.SocName*

*order by count(*) ASC*

*limit 2;*

***Output:-***



```
mysql> select society.SocName
    -> from society
    -> left join enrollment on society.SocID = enrollment.SID
    -> group by society.SocName
    -> order by count(*) ASC
    -> limit 2;
+-----------+
| SocName   |
+-----------+
| Debating  |
| Sashakt   |
+-----------+
2 rows in set (0.00 sec)
```

***<u>(12.) Find the student names who are not enrolled in any society.</u>***

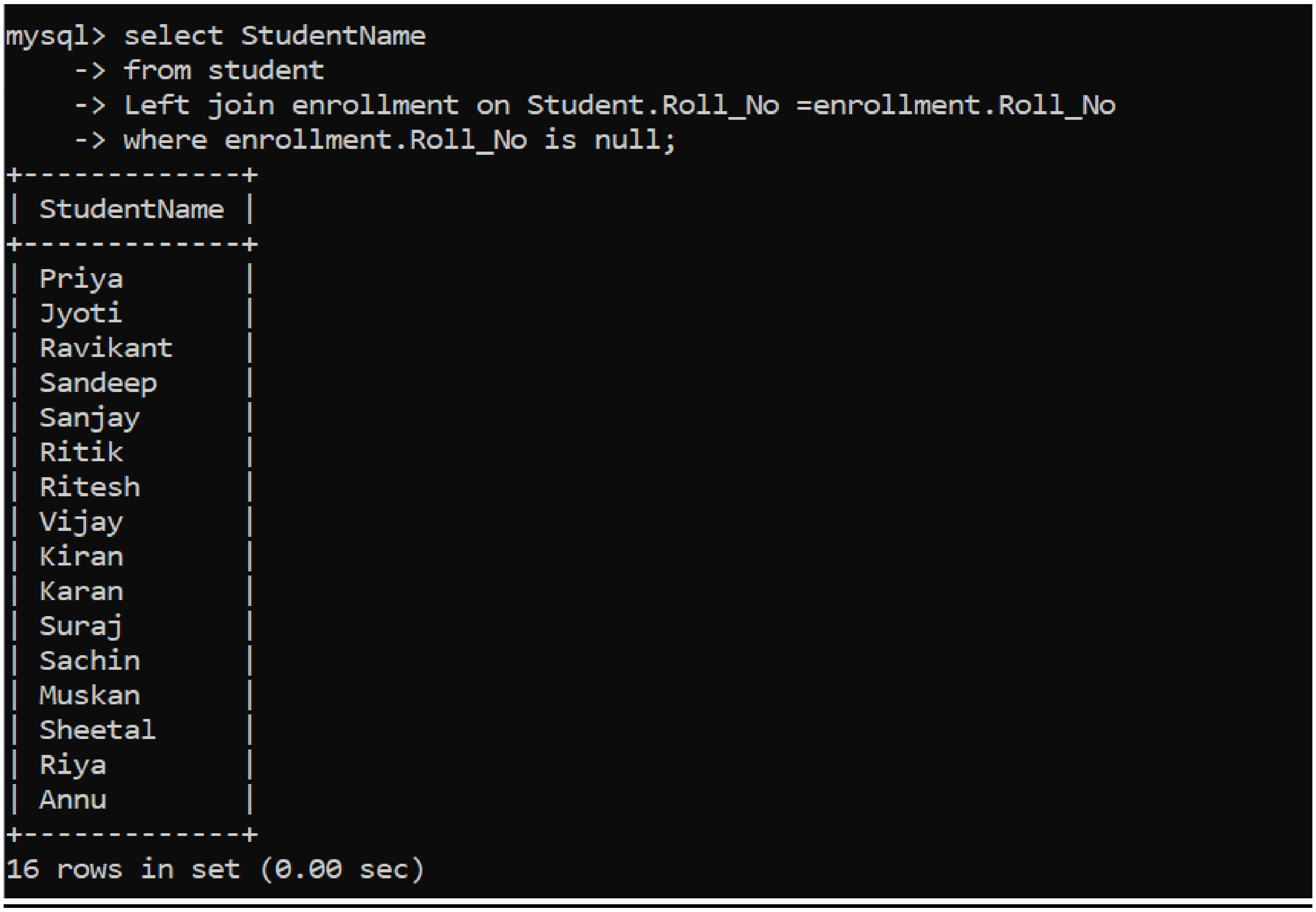
**<u>Query:-</u>** *select StudentName*

*from student*

*Left join enrollment on Student.Roll_No = enrollment.Roll_No*

*where enrollment.Roll_No is null;*

***<u>Output:-</u>***



```
mysql> select StudentName
    -> from student
    -> Left join enrollment on Student.Roll_No =enrollment.Roll_No
    -> where enrollment.Roll_No is null;
+-------------+
| StudentName |
+-------------+
| Priya       |
| Jyoti       |
| Ravikant    |
| Sandeep     |
| Sanjay      |
| Ritik       |
| Ritesh      |
| Vijay       |
| Kiran       |
| Karan       |
| Suraj       |
| Sachin      |
| Muskan      |
| Sheetal     |
| Riya        |
| Annu        |
+-------------+
16 rows in set (0.00 sec)
```

***<u>(13.) Find the student names enrolled in at least two societies.</u>***

***<u>Query:-</u>*** *select studentName*

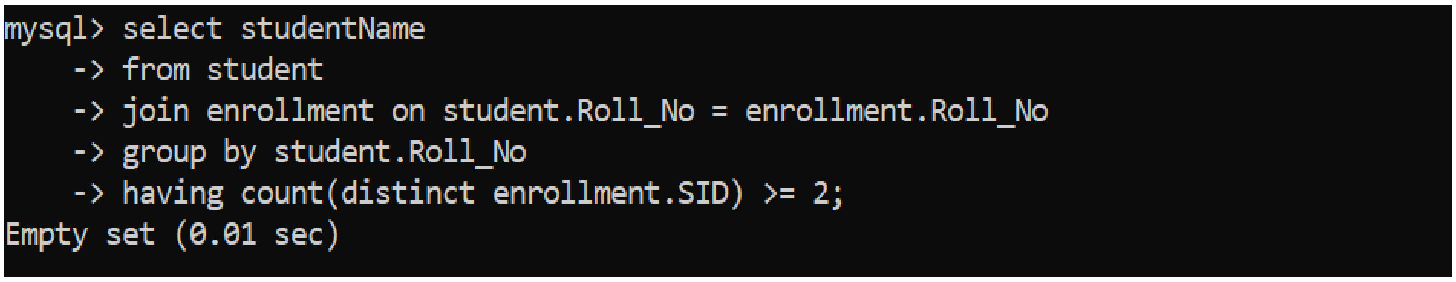
*from student*

*join enrollment on student.Roll_No = enrollment.Roll_No*

*group by student.Roll_No*

*having count(distinct enrollment.SID) >= 2;*

**Output:-**



```
mysql> select studentName
    -> from student
    -> join enrollment on student.Roll_No = enrollment.Roll_No
    -> group by student.Roll_No
    -> having count(distinct enrollment.SID) >= 2;
Empty set (0.01 sec)
```

## (14.) Find society names in which maximum students are enrolled.

**Query:-** *select society.SocName*

*from society*

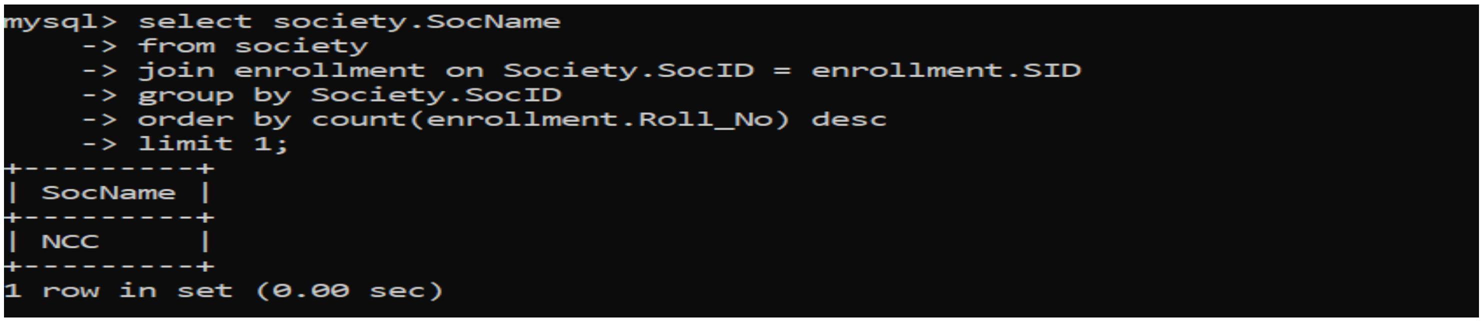
*join enrollment on Society.SocID = enrollment.SID*

*group by Society.SocID*

*order by count(enrollment.Roll_No) desc*

*limit 1;*

**Output:-**



```
mysql> select society.SocName
    -> from society
    -> join enrollment on Society.SocID = enrollment.SID
    -> group by Society.SocID
    -> order by count(enrollment.Roll_No) desc
    -> limit 1;
+---------+
| SocName |
+---------+
| NCC     |
+---------+
1 row in set (0.00 sec)
```

## ***(15.) Find names of all students who have enrolled in any society and society names in which at least one student has enrolled.***

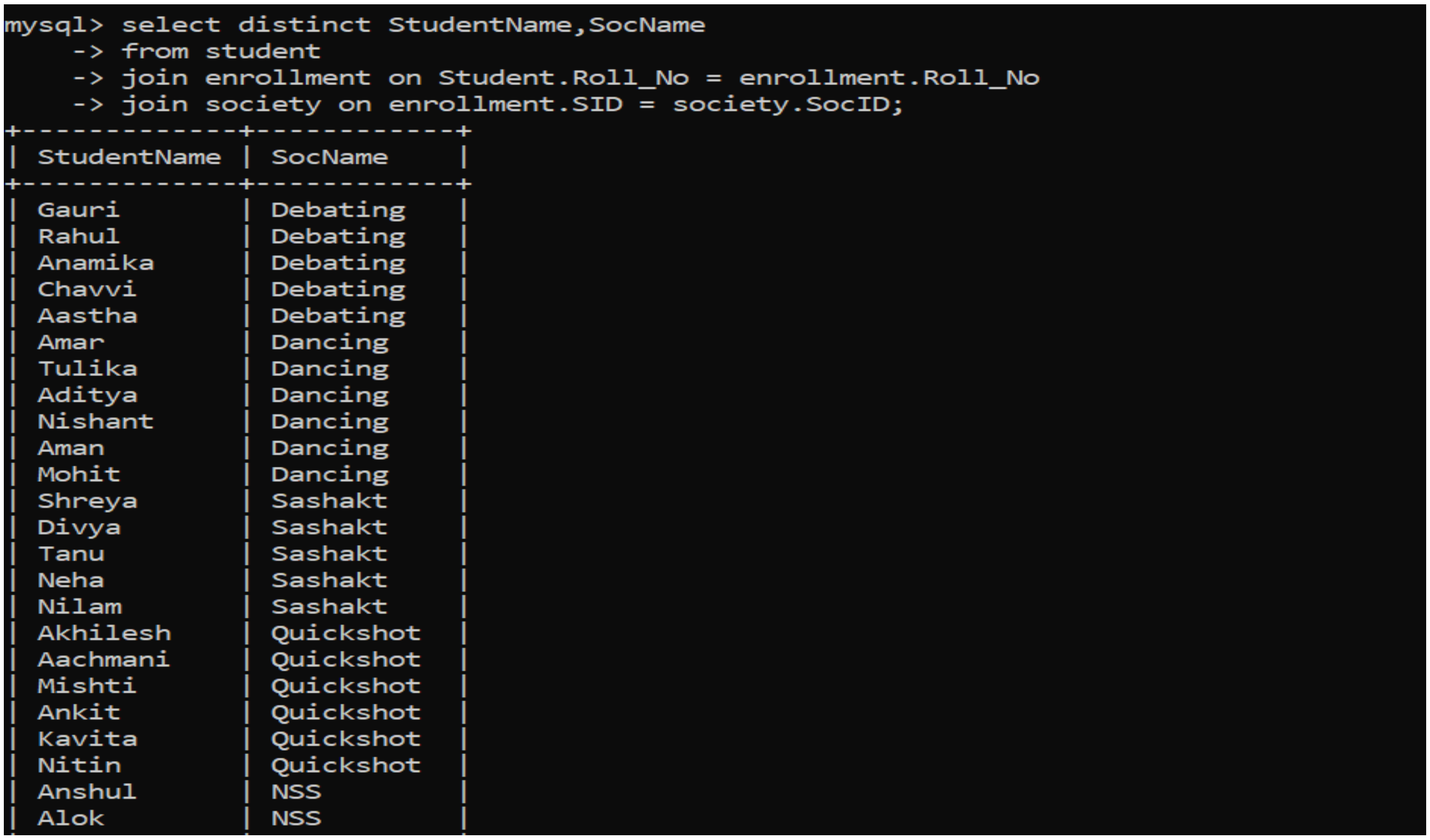
***Query:-*** *select distinct StudentName,SocName*

*from student*

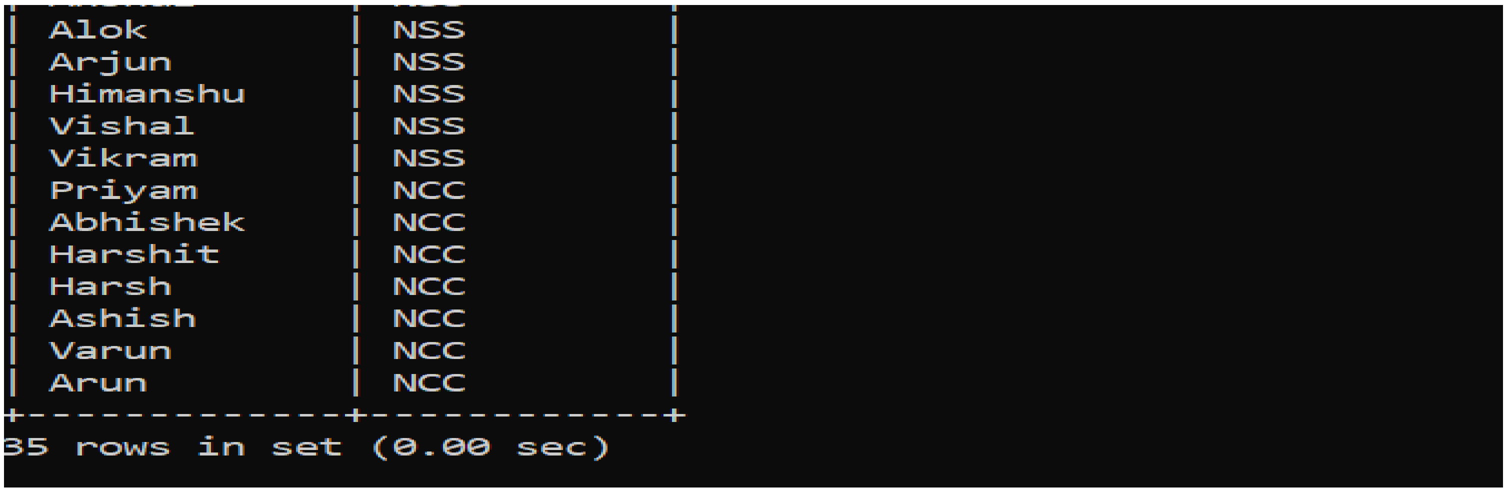
*join enrollment on Student.Roll_No = enrollment.Roll_No*

*join society on enrollment.SID = society.SocID;*

***Output:-***



```
mysql> select distinct StudentName,SocName
    -> from student
    -> join enrollment on Student.Roll_No = enrollment.Roll_No
    -> join society on enrollment.SID = society.SocID;
+-------------+------------+
| StudentName | SocName    |
+-------------+------------+
| Gauri       | Debating   |
| Rahul       | Debating   |
| Anamika     | Debating   |
| Chavvi      | Debating   |
| Aastha      | Debating   |
| Amar        | Dancing    |
| Tulika      | Dancing    |
| Aditya      | Dancing    |
| Nishant     | Dancing    |
| Aman        | Dancing    |
| Mohit       | Dancing    |
| Shreya      | Sashakt    |
| Divya       | Sashakt    |
| Tanu        | Sashakt    |
| Neha        | Sashakt    |
| Nilam       | Sashakt    |
| Akhilesh    | Quickshot  |
| Aachmani    | Quickshot  |
| Mishti      | Quickshot  |
| Ankit       | Quickshot  |
| Kavita      | Quickshot  |
| Nitin       | Quickshot  |
| Anshul      | NSS        |
| Alok        | NSS        |
```


```
| Alok         | NSS         |
| Arjun        | NSS         |
| Himanshu     | NSS         |
| Vishal       | NSS         |
| Vikram       | NSS         |
| Priyam       | NCC         |
| Abhishek     | NCC         |
| Harshit      | NCC         |
| Harsh        | NCC         |
| Ashish       | NCC         |
| Varun        | NCC         |
| Arun         | NCC         |
+--------------+-------------+
35 rows in set (0.00 sec)
```

***(16.) Find names of students who are enrolled in any of the three societies 'Debating', 'Dancing' and 'Sashakt'.***

***Query:-*** *select distinct StudentName*

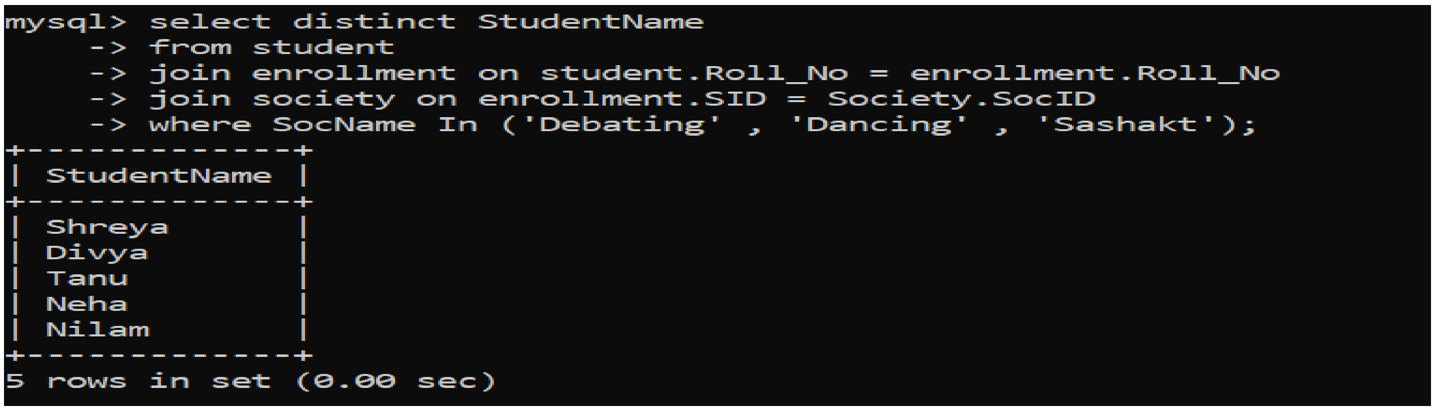
*from student*

*join enrollment on student.Roll_No = enrollment.Roll_No*

*join society on enrollment.SID = Society.SocID*

*where SocName In ('Debating' , 'Dancing' , 'Sashakt');*

**Output:-**



```
mysql> select distinct StudentName
    -> from student
    -> join enrollment on student.Roll_No = enrollment.Roll_No
    -> join society on enrollment.SID = Society.SocID
    -> where SocName In ('Debating' , 'Dancing' , 'Sashakt');
+-------------+
| StudentName |
+-------------+
| Shreya      |
| Divya       |
| Tanu        |
| Neha        |
| Nilam       |
+-------------+
5 rows in set (0.00 sec)
```

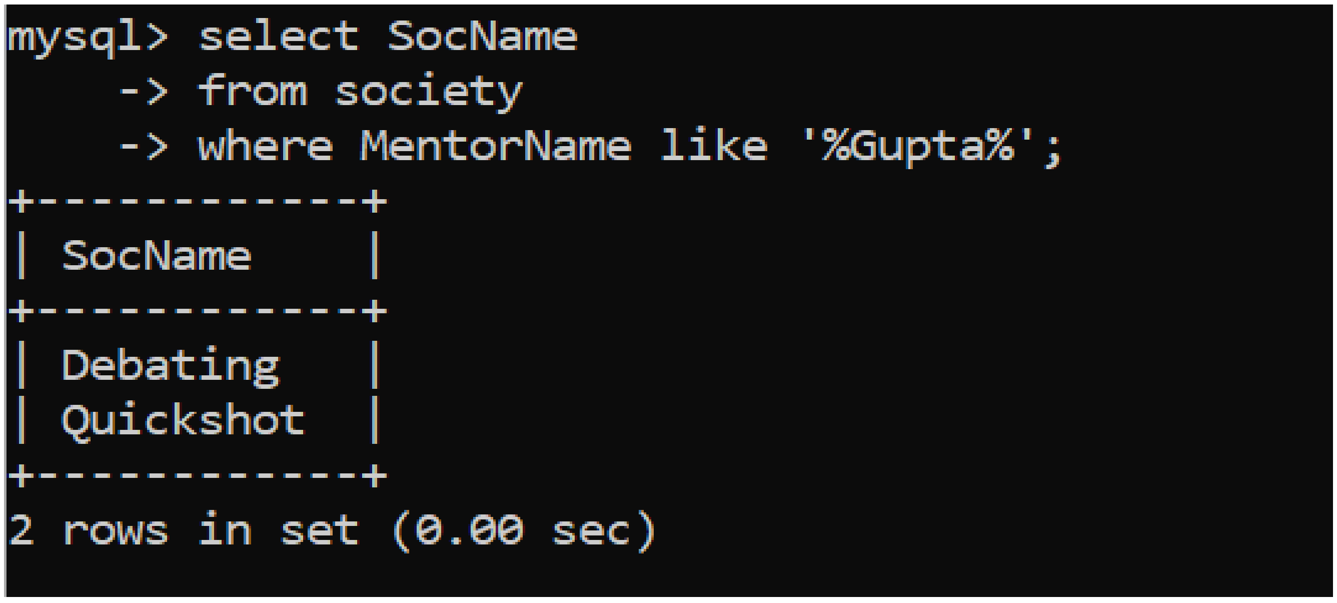
## (17.) Find society names such that its mentor has a name with 'Gupta' in it.

**Query:-** *select SocName*

*from society*

*where MentorName like '%Gupta%';*

**Output:-**



```
mysql> select SocName
    -> from society
    -> where MentorName like '%Gupta%';
+-----------+
| SocName   |
+-----------+
| Debating  |
| Quickshot |
+-----------+
2 rows in set (0.00 sec)
```

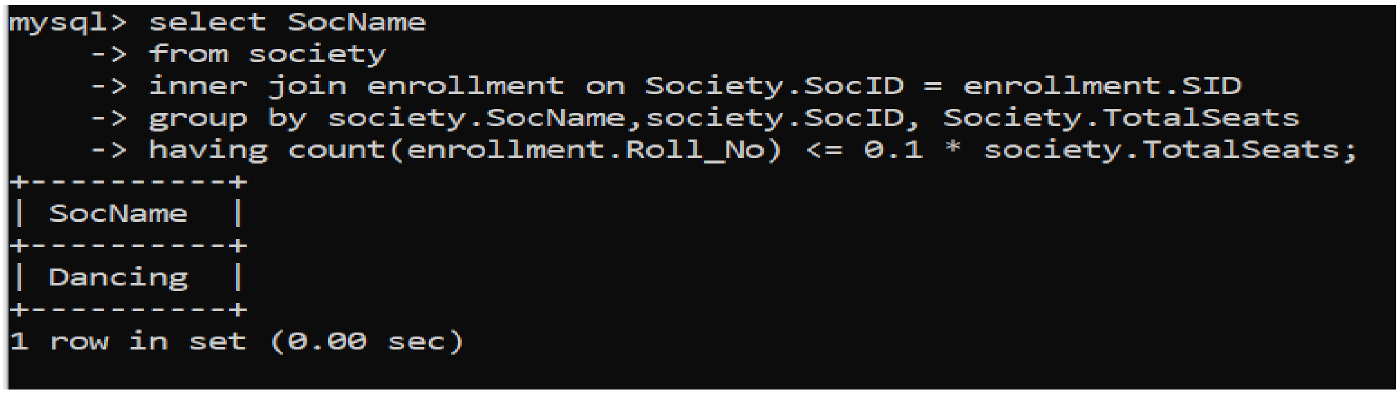
***(18.) Find the society names in which the number of enrolled students is only 10% of its capacity.***

***Query:-*** *select SocName*

*from society*

*inner join enrollment on Society.SocID = enrollment.SID group by society.SocName,society.SocID, Society.TotalSeats having count(enrollment.Roll_No) <= 0.1 * society.TotalSeats;*

***Output:-***



```
mysql> select SocName
    -> from society
    -> inner join enrollment on Society.SocID = enrollment.SID
    -> group by society.SocName,society.SocID, Society.TotalSeats
    -> having count(enrollment.Roll_No) <= 0.1 * society.TotalSeats;
+----------+
| SocName  |
+----------+
| Dancing  |
+----------+
1 row in set (0.00 sec)
```

***(19.) Display the vacant seats for each society.***

***Query:-*** *select SocName,*

*TotalSeats-count(enrollment.Roll_No) As vacant_Seats*

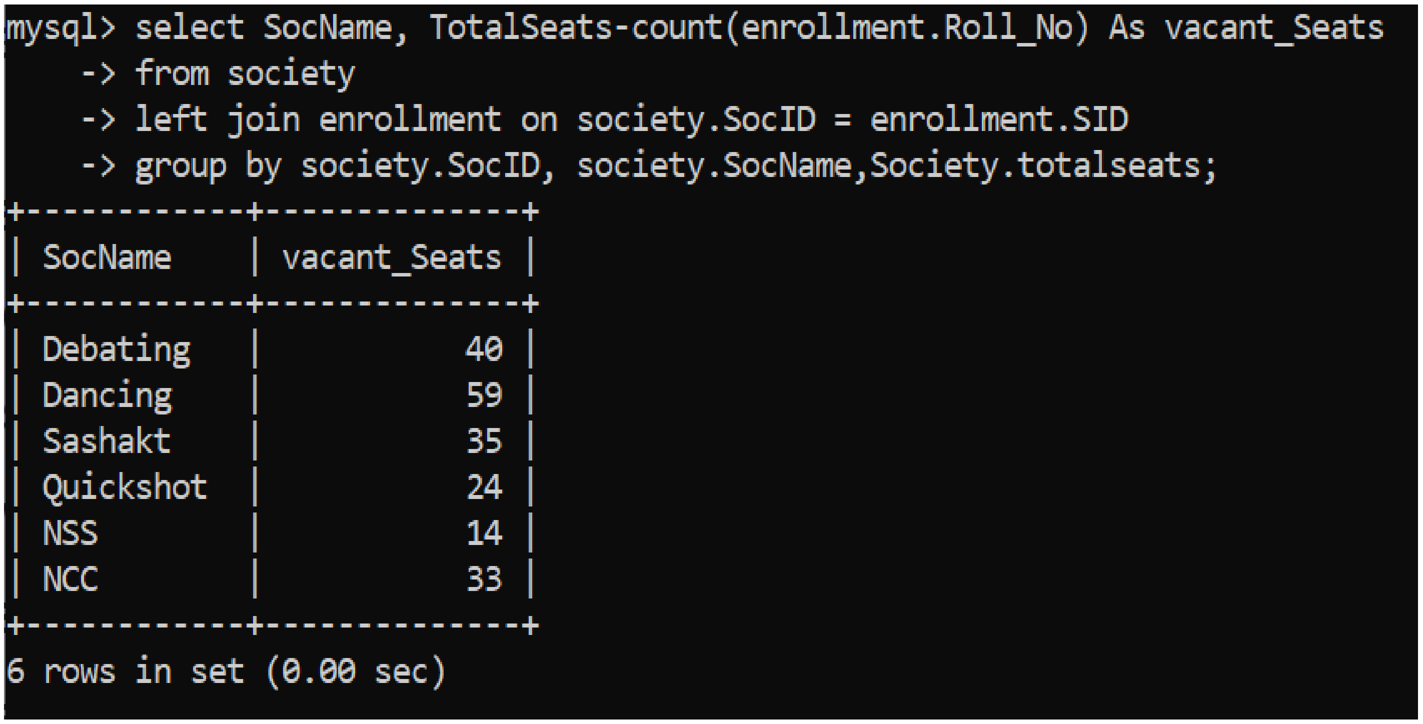
*from society*

*left join enrollment on society.SocID = enrollment.SID*

*group by society.SocID, society.SocName ,*

*Society.totalseats;*

***Output:-***



```
mysql> select SocName, TotalSeats-count(enrollment.Roll_No) As vacant_Seats
    -> from society
    -> left join enrollment on society.SocID = enrollment.SID
    -> group by society.SocID, society.SocName,Society.totalseats;
+-----------+-------------+
| SocName   | vacant_Seats |
+-----------+-------------+
| Debating  |           40 |
| Dancing   |           59 |
| Sashakt   |           35 |
| Quickshot |           24 |
| NSS       |           14 |
| NCC       |           33 |
+-----------+-------------+
6 rows in set (0.00 sec)
```

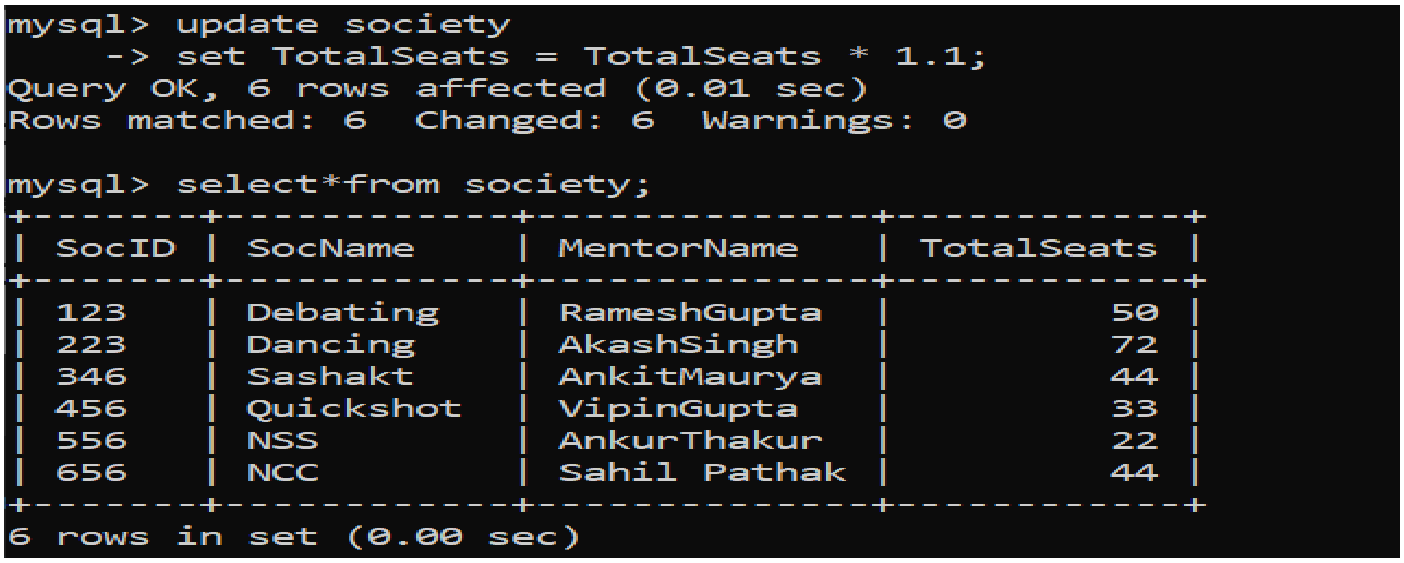
## ***(20.) Increment Total Seats of each society by 10%.***

***Query:-*** *update society*

*set TotalSeats = TotalSeats * 1.1;*

*select*from society;*

***Output:-***



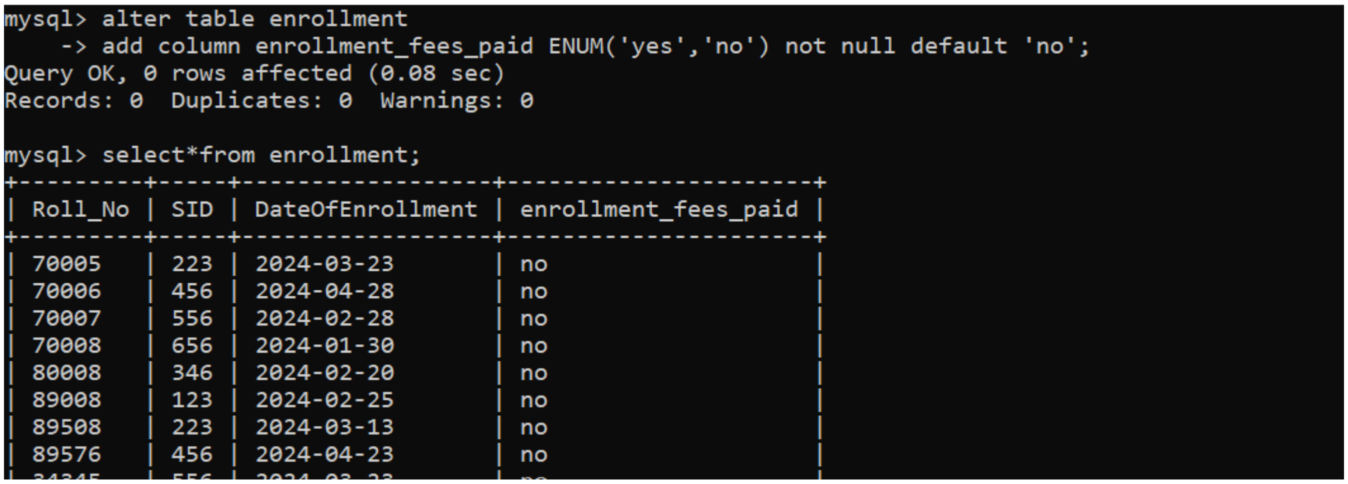
```
mysql> update society
    -> set TotalSeats = TotalSeats * 1.1;
Query OK, 6 rows affected (0.01 sec)
Rows matched: 6  Changed: 6  Warnings: 0

mysql> select*from society;
+-------+------------+-------------+------------+
| SocID | SocName    | MentorName  | TotalSeats |
+-------+------------+-------------+------------+
| 123   | Debating   | RameshGupta |         50 |
| 223   | Dancing    | AkashSingh  |         72 |
| 346   | Sashakt    | AnkitMaurya |         44 |
| 456   | Quickshot  | VipinGupta  |         33 |
| 556   | NSS        | AnkurThakur |         22 |
| 656   | NCC        | Sahil Pathak|         44 |
+-------+------------+-------------+------------+
6 rows in set (0.00 sec)
```

## ***(21.) Add the enrollment fees paid ('yes'/'No') field in the enrollment table.***

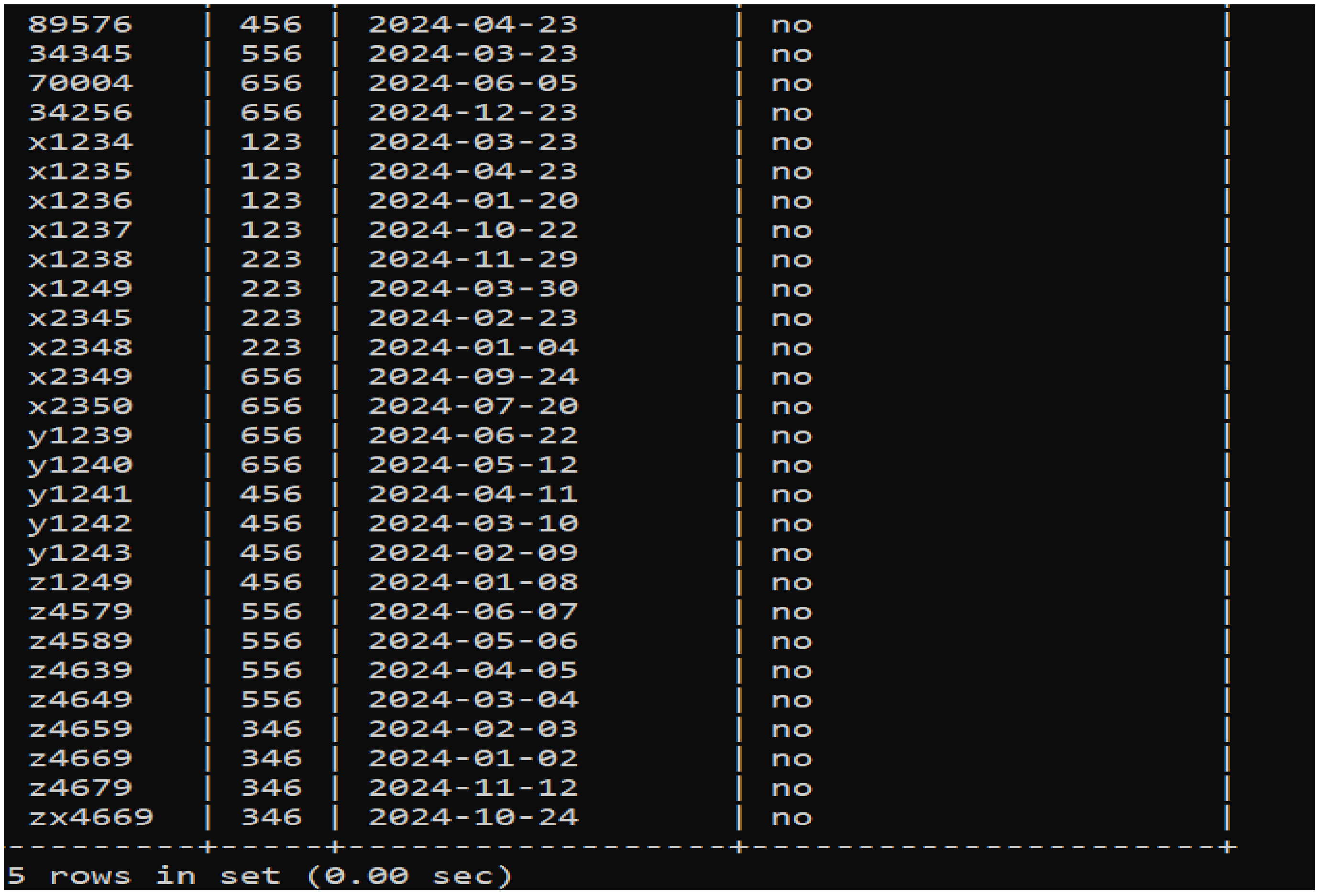
***Query:-*** *alter table enrollment*

*add column enrollment_fees_paid ENUM('yes','no') not null default 'no';*

***Output:-***



```
mysql> alter table enrollment
    -> add column enrollment_fees_paid ENUM('yes','no') not null default 'no';
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select*from enrollment;
+---------+-----+----------------+----------------------+
| Roll_No | SID | DateOfEnrollment | enrollment_fees_paid |
+---------+-----+----------------+----------------------+
| 70005   | 223 | 2024-03-23     | no                   |
| 70006   | 456 | 2024-04-28     | no                   |
| 70007   | 556 | 2024-02-28     | no                   |
| 70008   | 656 | 2024-01-30     | no                   |
| 80008   | 346 | 2024-02-20     | no                   |
| 89008   | 123 | 2024-02-25     | no                   |
| 89508   | 223 | 2024-03-13     | no                   |
| 89576   | 456 | 2024-04-23     | no                   |
```



```
 89576    | 456 | 2024-04-23         | no                   |
 34345    | 556 | 2024-03-23         | no                   |
 70004    | 656 | 2024-06-05         | no                   |
 34256    | 656 | 2024-12-23         | no                   |
 x1234    | 123 | 2024-03-23         | no                   |
 x1235    | 123 | 2024-04-23         | no                   |
 x1236    | 123 | 2024-01-20         | no                   |
 x1237    | 123 | 2024-10-22         | no                   |
 x1238    | 223 | 2024-11-29         | no                   |
 x1249    | 223 | 2024-03-30         | no                   |
 x2345    | 223 | 2024-02-23         | no                   |
 x2348    | 223 | 2024-01-04         | no                   |
 x2349    | 656 | 2024-09-24         | no                   |
 x2350    | 656 | 2024-07-20         | no                   |
 y1239    | 656 | 2024-06-22         | no                   |
 y1240    | 656 | 2024-05-12         | no                   |
 y1241    | 456 | 2024-04-11         | no                   |
 y1242    | 456 | 2024-03-10         | no                   |
 y1243    | 456 | 2024-02-09         | no                   |
 z1249    | 456 | 2024-01-08         | no                   |
 z4579    | 556 | 2024-06-07         | no                   |
 z4589    | 556 | 2024-05-06         | no                   |
 z4639    | 556 | 2024-04-05         | no                   |
 z4649    | 556 | 2024-03-04         | no                   |
 z4659    | 346 | 2024-02-03         | no                   |
 z4669    | 346 | 2024-01-02         | no                   |
 z4679    | 346 | 2024-11-12         | no                   |
 zx4669   | 346 | 2024-10-24         | no                   |
---------+-----+-------------------+----------------------+
5 rows in set (0.00 sec)
```

## ***(22.) Update date of enrollment of society id ‘s1’ to ‘2018-01-15’, ‘s2’ to current date and ‘s3’ to ‘2018-01-02’.***

***Query:-*** *update enrollment*

*set DateofEnrollment=*

*case SID*

*when'NSS' then '2025-01-22'*

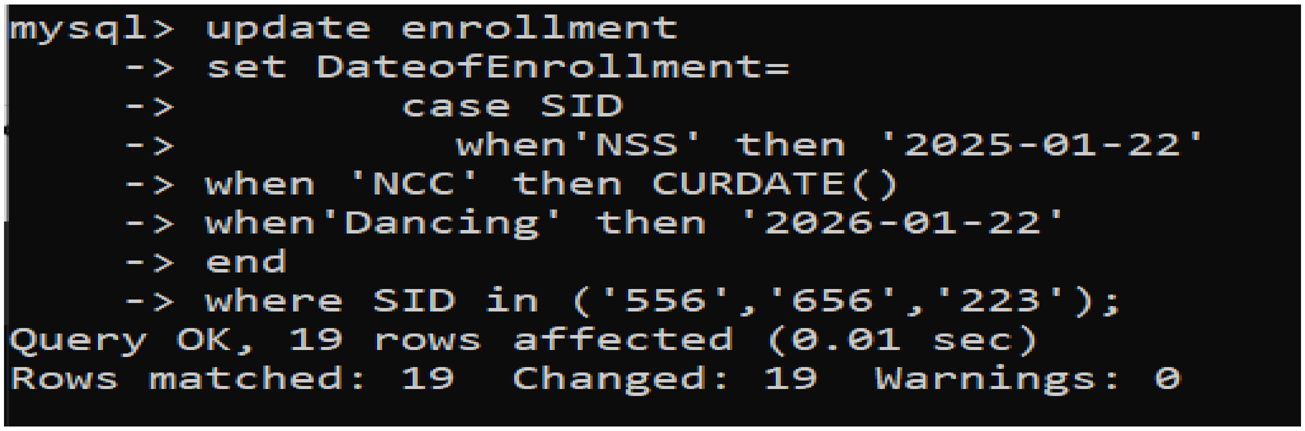
*when 'NCC' then CURDATE()*

*when 'Dancing' then '2026-01-22'*

*end*

*where SID in ('556','656','223');*

***Output:-***



```
mysql> update enrollment
    -> set DateofEnrollment=
    ->          case SID
    ->             when'NSS' then '2025-01-22'
    -> when 'NCC' then CURDATE()
    -> when'Dancing' then '2026-01-22'
    -> end
    -> where SID in ('556','656','223');
Query OK, 19 rows affected (0.01 sec)
Rows matched: 19  Changed: 19  Warnings: 0
```

***(23.) Create a view to keep track of society names with the total number of students enrolled in it.***

***Query:-*** *create view SocietyEnrollment As*

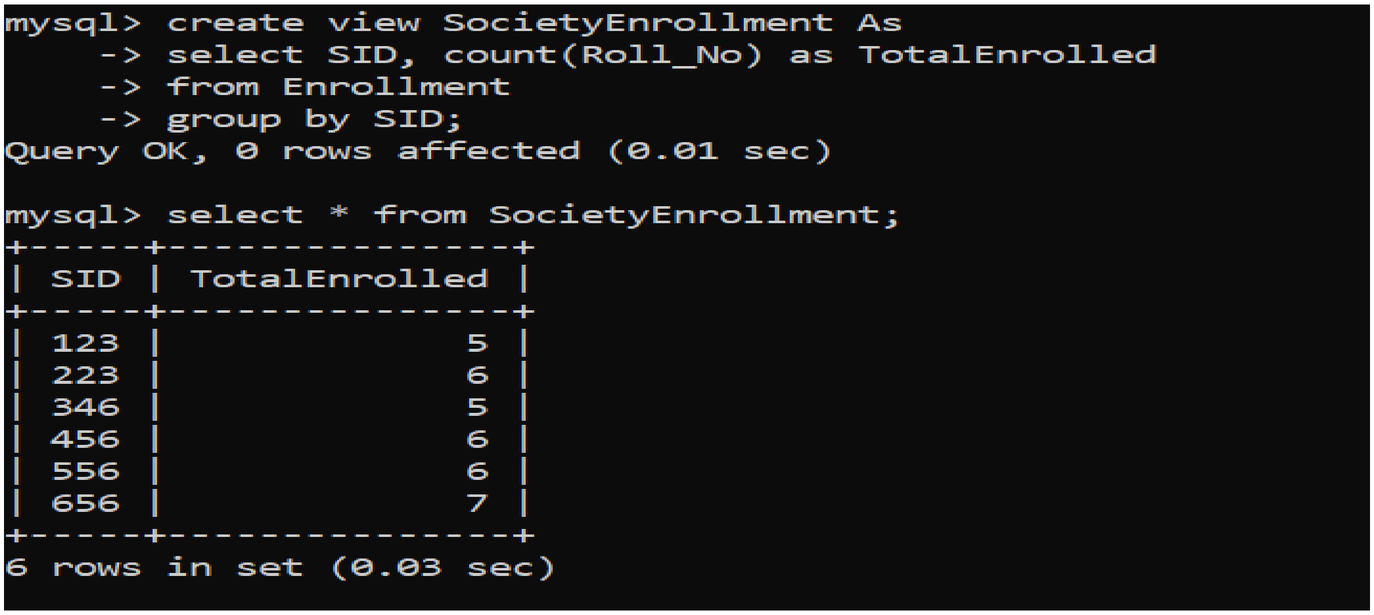
*select SID, count(Roll_No) as TotalEnrolled*

*from Enrollment*

*group by SID;*

*select * from SocietyEnrollment;*

***Output:-***



```
mysql> create view SocietyEnrollment As
    -> select SID, count(Roll_No) as TotalEnrolled
    -> from Enrollment
    -> group by SID;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from SocietyEnrollment;
+-----+---------------+
| SID | TotalEnrolled |
+-----+---------------+
| 123 |             5 |
| 223 |             6 |
| 346 |             5 |
| 456 |             6 |
| 556 |             6 |
| 656 |             7 |
+-----+---------------+
6 rows in set (0.03 sec)
```

## ***(24.) Find student names enrolled in all the societies.***

***Query:-*** *select StudentName*

*from Student*

*where Roll_No in (*

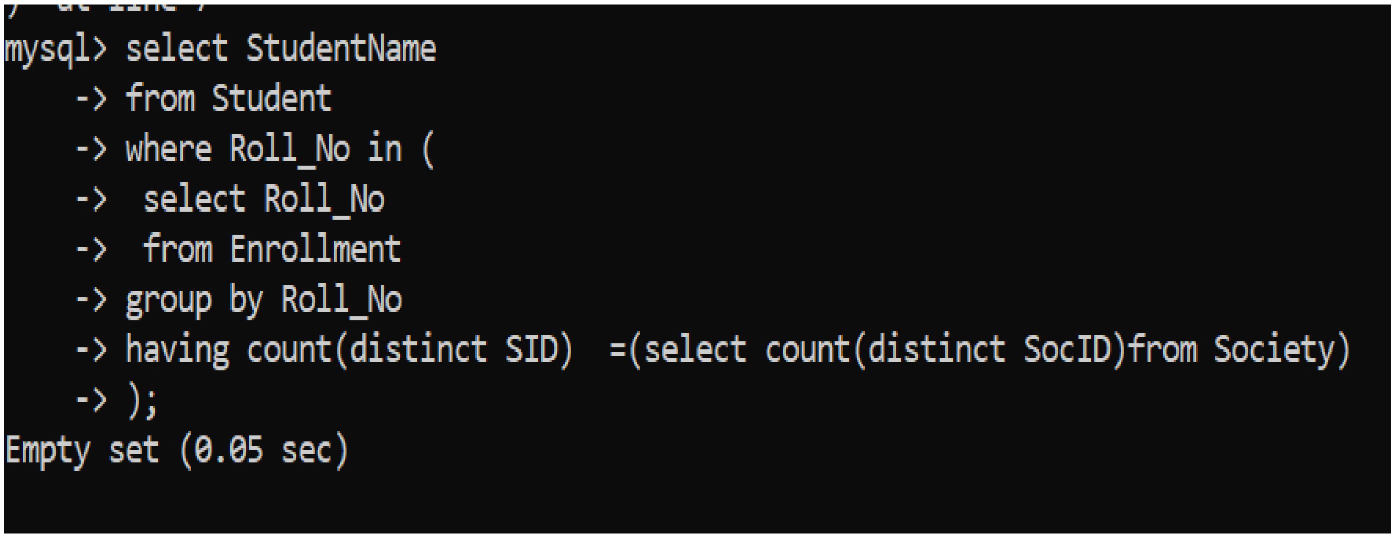
*select Roll_No*

*from Enrollment*

*group by Roll_No*

*having count(distinct SID) =(select count(distinct SocID)from Society));*

***<u>Output:-</u>***



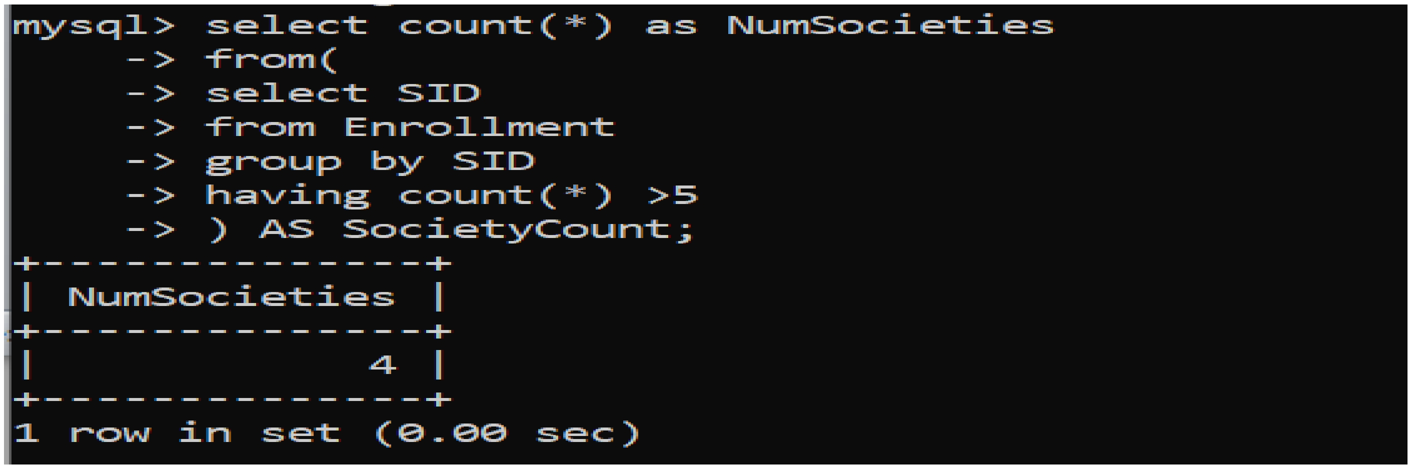
```
mysql> select StudentName
    -> from Student
    -> where Roll_No in (
    ->  select Roll_No
    ->  from Enrollment
    -> group by Roll_No
    -> having count(distinct SID)  =(select count(distinct SocID)from Society)
    -> );
Empty set (0.05 sec)
```

## ***<u>(25.) Count the number of societies with more than 5 students enrolled in it.</u>***

***<u>Query:-</u>*** *select count(*) as NumSocieties*

```
         from(
        select SID
        from Enrollment
        group by SID
        having count(*) >5
        ) AS SocietyCount;
```

***<u>Output:-</u>***



```
mysql> select count(*) as NumSocieties
    -> from(
    -> select SID
    -> from Enrollment
    -> group by SID
    -> having count(*) >5
    -> ) AS SocietyCount;
+--------------+
| NumSocieties |
+--------------+
|            4 |
+--------------+
1 row in set (0.00 sec)
```

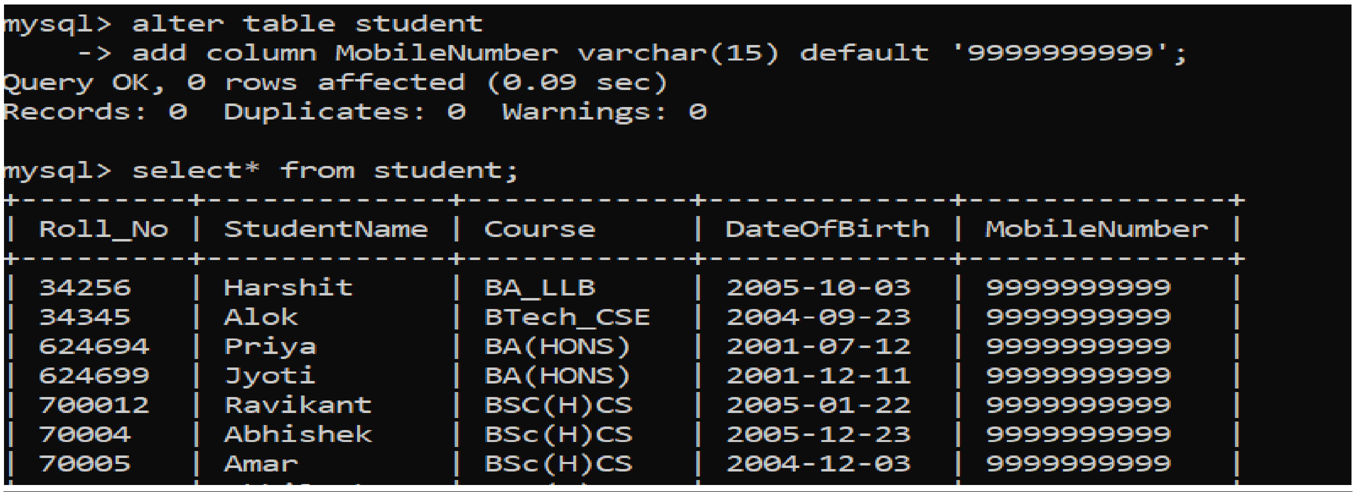
## <u>*(26.) Add column Mobile number in student table with default value ‘9999999999’.*</u>

***<u>Query:-</u>*** *alter table student*

*add column MobileNumber varchar(15) default '9999999999';*

*select *from student;*

***<u>Output:-</u>***



```
mysql> alter table student
    -> add column MobileNumber varchar(15) default '9999999999';
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select* from student;
+---------+-------------+------------+-------------+--------------+
| Roll_No | StudentName | Course     | DateOfBirth | MobileNumber |
+---------+-------------+------------+-------------+--------------+
| 34256   | Harshit     | BA_LLB     | 2005-10-03  | 9999999999   |
| 34345   | Alok        | BTech_CSE  | 2004-09-23  | 9999999999   |
| 624694  | Priya       | BA(HONS)   | 2001-07-12  | 9999999999   |
| 624699  | Jyoti       | BA(HONS)   | 2001-12-11  | 9999999999   |
| 700012  | Ravikant    | BSC(H)CS   | 2005-01-22  | 9999999999   |
| 70004   | Abhishek    | BSc(H)CS   | 2005-12-23  | 9999999999   |
| 70005   | Amar        | BSc(H)CS   | 2004-12-03  | 9999999999   |
```

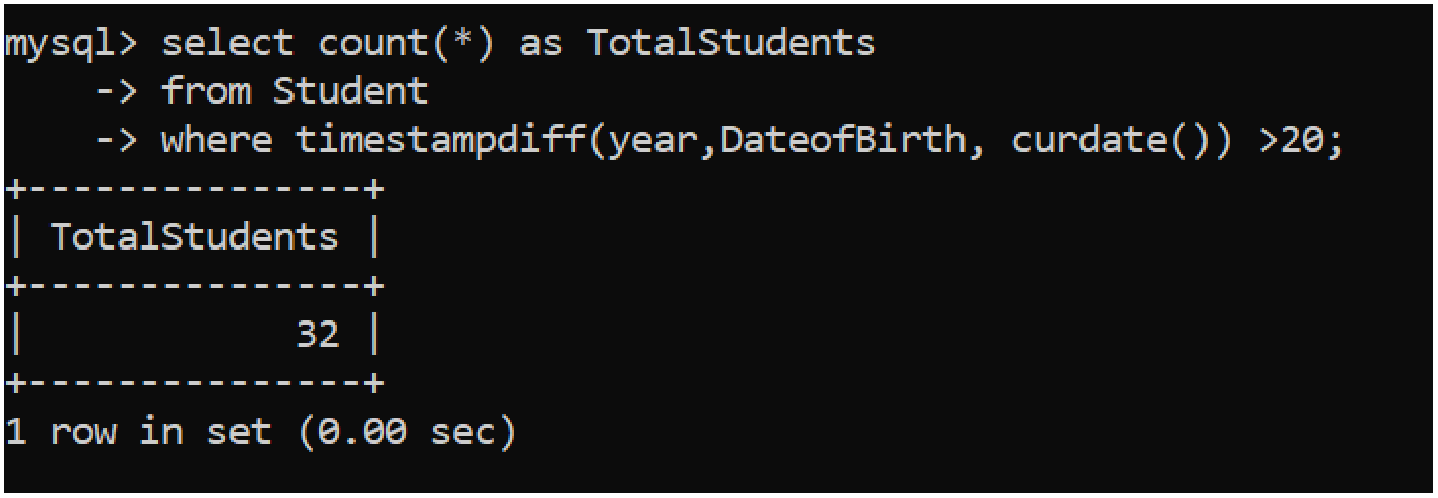
***<u>(27.) Find the total number of students whose age is > 20 years.</u>***

***<u>Query:-</u>*** *select count(*) as TotalStudents*

*from Student*

*where timestampdiff(year,DateofBirth, curdate()) >20;*

***<u>Output:-</u>***



```
mysql> select count(*) as TotalStudents
    -> from Student
    -> where timestampdiff(year,DateofBirth, curdate()) >20;
+---------------+
| TotalStudents |
+---------------+
|            32 |
+---------------+
1 row in set (0.00 sec)
```

***<u>(28.) Find names of students who are born in 2001 and are enrolled in at least one society.</u>***

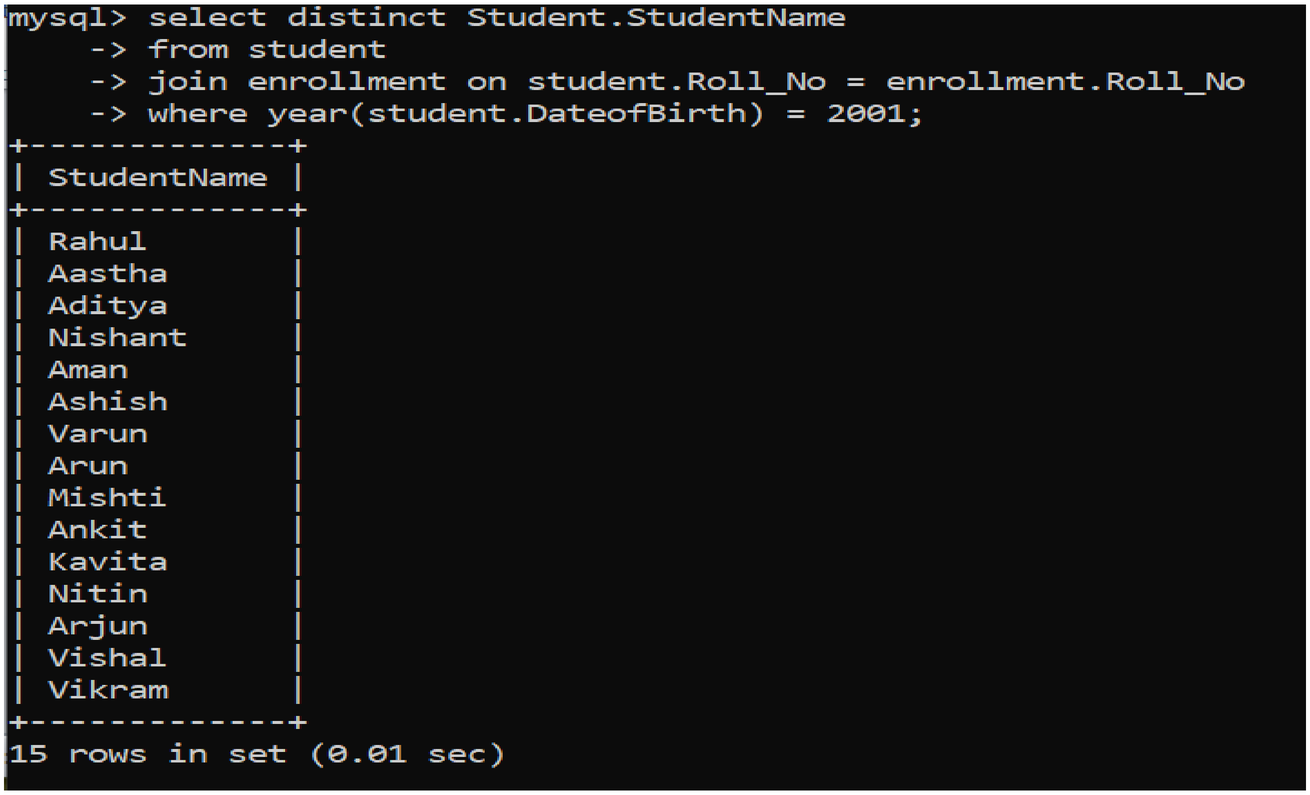
***<u>Query:-</u>*** *select distinct Student.StudentName*

*from student*

*join enrollment on student.Roll_No = enrollment.Roll_No*

*where year(student.DateofBirth) = 2001;*

***Output:-***



```
mysql> select distinct Student.StudentName
    -> from student
    -> join enrollment on student.Roll_No = enrollment.Roll_No
    -> where year(student.DateofBirth) = 2001;
+-------------+
| StudentName |
+-------------+
| Rahul       |
| Aastha      |
| Aditya      |
| Nishant     |
| Aman        |
| Ashish      |
| Varun       |
| Arun        |
| Mishti      |
| Ankit       |
| Kavita      |
| Nitin       |
| Arjun       |
| Vishal      |
| Vikram      |
+-------------+
15 rows in set (0.01 sec)
```

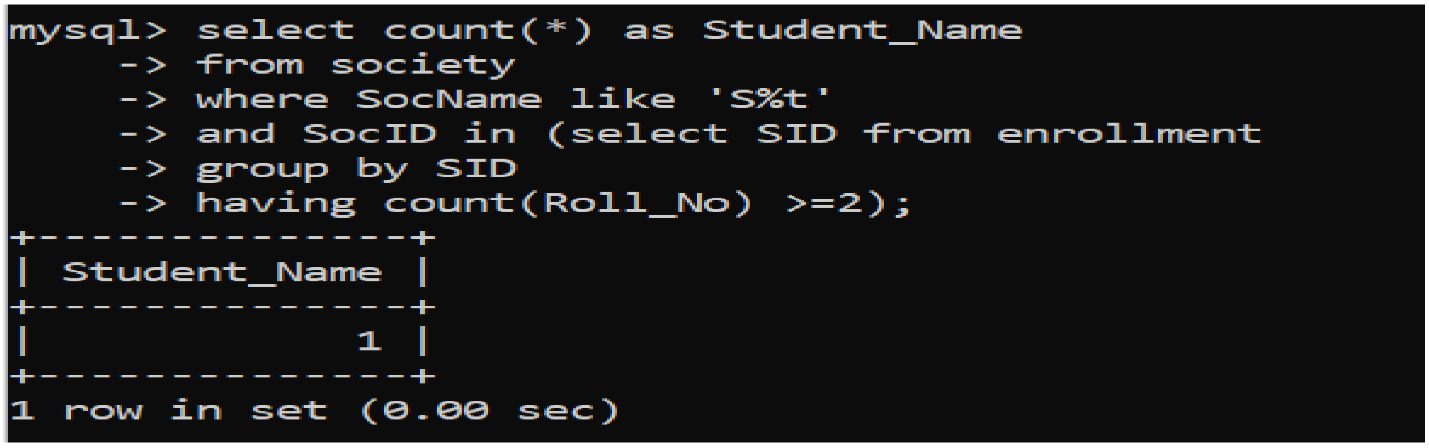
## ***(29.) Count all societies whose name starts with 'S' and ends with 't' and at least 5 students are enrolled in the society.***

***Query:-*** *select count(*) as Student_Name*

```
from society
where SocName like 'S%t'
and SocID in (select SID from enrollment
group by SID
```

*having count(Roll_No) >=2);*

***Output:-***



```
mysql> select count(*) as Student_Name
    -> from society
    -> where SocName like 'S%t'
    -> and SocID in (select SID from enrollment
    -> group by SID
    -> having count(Roll_No) >=2);
+--------------+
| Student_Name |
+--------------+
|            1 |
+--------------+
1 row in set (0.00 sec)
```

## ***(30.) Display the following information: Society name Mentor name Total Capacity Total Enrolled Unfilled Seats.***

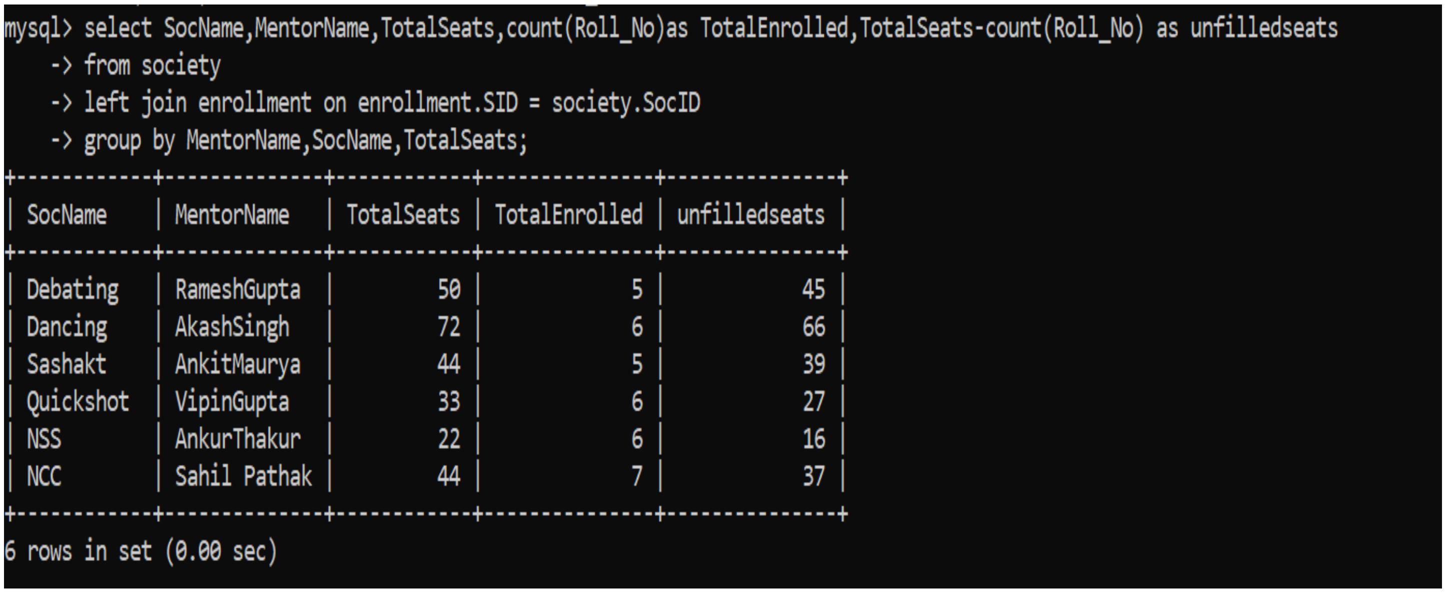
***Query:-*** *select SocName,MentorName,TotalSeats,count(Roll_No)as TotalEnrolled,TotalSeats-count(Roll_No) as unfilledseats*

*from society*

*left join enrollment on enrollment.SID = society.SocID*

*group by MentorName,SocName,TotalSeats;*

***Output:-***



```
mysql> select SocName,MentorName,TotalSeats,count(Roll_No)as TotalEnrolled,TotalSeats-count(Roll_No) as unfilledseats
    -> from society
    -> left join enrollment on enrollment.SID = society.SocID
    -> group by MentorName,SocName,TotalSeats;
+-----------+-------------+------------+---------------+---------------+
| SocName   | MentorName  | TotalSeats | TotalEnrolled | unfilledseats |
+-----------+-------------+------------+---------------+---------------+
| Debating  | RameshGupta |         50 |             5 |            45 |
| Dancing   | AkashSingh  |         72 |             6 |            66 |
| Sashakt   | AnkitMaurya |         44 |             5 |            39 |
| Quickshot | VipinGupta  |         33 |             6 |            27 |
| NSS       | AnkurThakur |         22 |             6 |            16 |
| NCC       | Sahil Pathak|         44 |             7 |            37 |
+-----------+-------------+------------+---------------+---------------+
6 rows in set (0.00 sec)
```

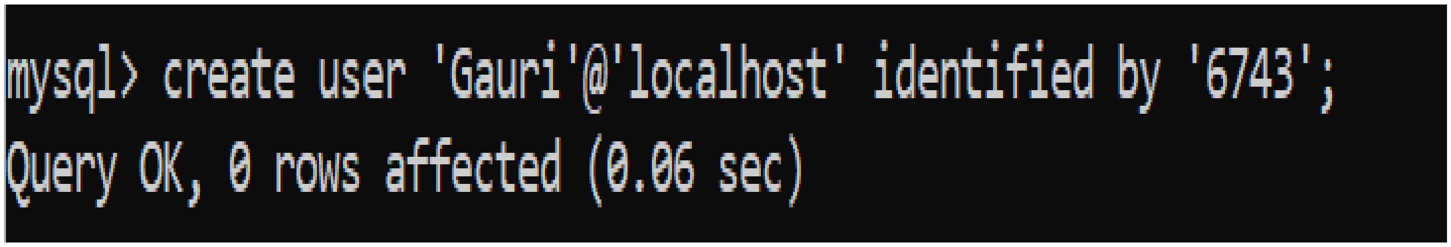
## ***(II). Do the following database administration commands:***

## ***create user, create role, grant privileges to a role, revoke privileges from a role, create index.***

***CREATE USER:-***

***Query:-*** *create user 'Gauri'@'localhost' identified by '6743';*

***Output:-***

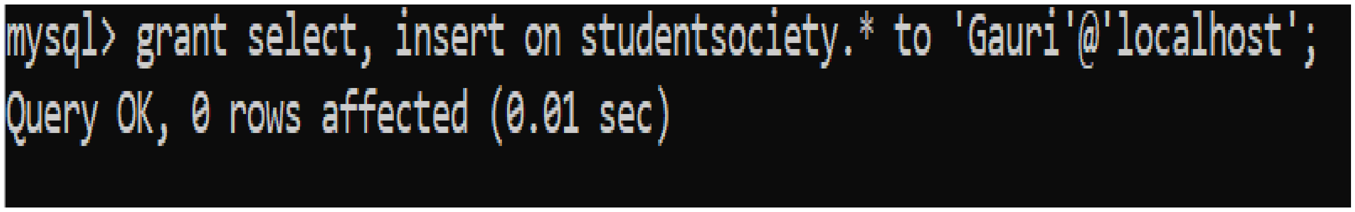


```
mysql> create user 'Gauri'@'localhost' identified by '6743';
Query OK, 0 rows affected (0.06 sec)
```

***GRANT PRIVILEGES TO A ROLE:-***

***Query:-*** *grant select, insert on studentsociety.* to 'Gauri'@'localhost';*

***Output:-***

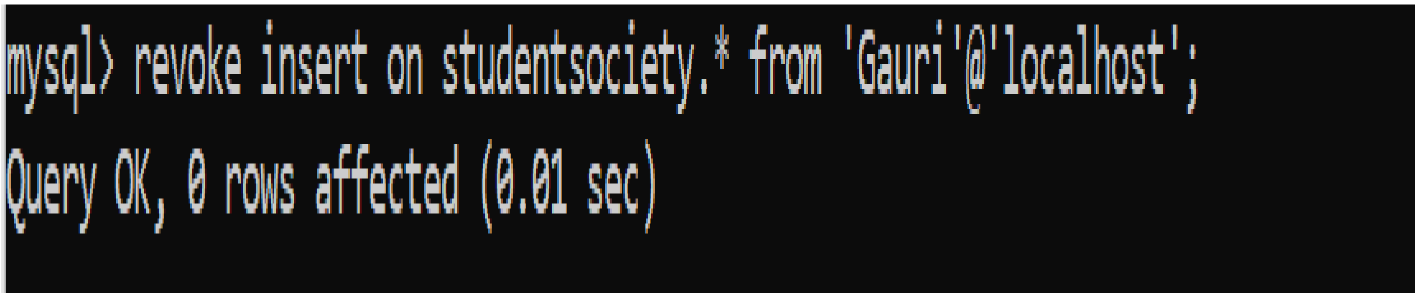


```
mysql> grant select, insert on studentsociety.* to 'Gauri'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

***REVOKE PRIVILEGES FROM A ROLE:-***

***Query:-*** *revoke insert on studentsociety.* from 'Gauri'@'localhost';*

***Output:-***

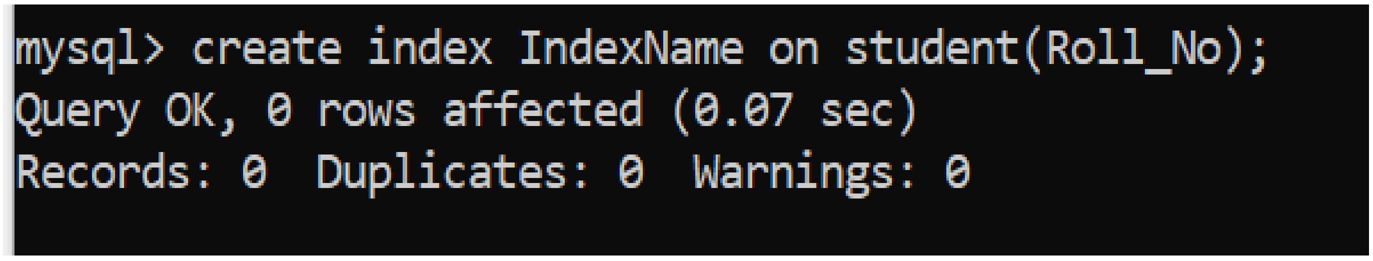


```
mysql> revoke insert on studentsociety.* from 'Gauri'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

***CREATE INDEX:-***

***Query:-*** *create index IndexName on student(Roll_No);*

***Output:-***

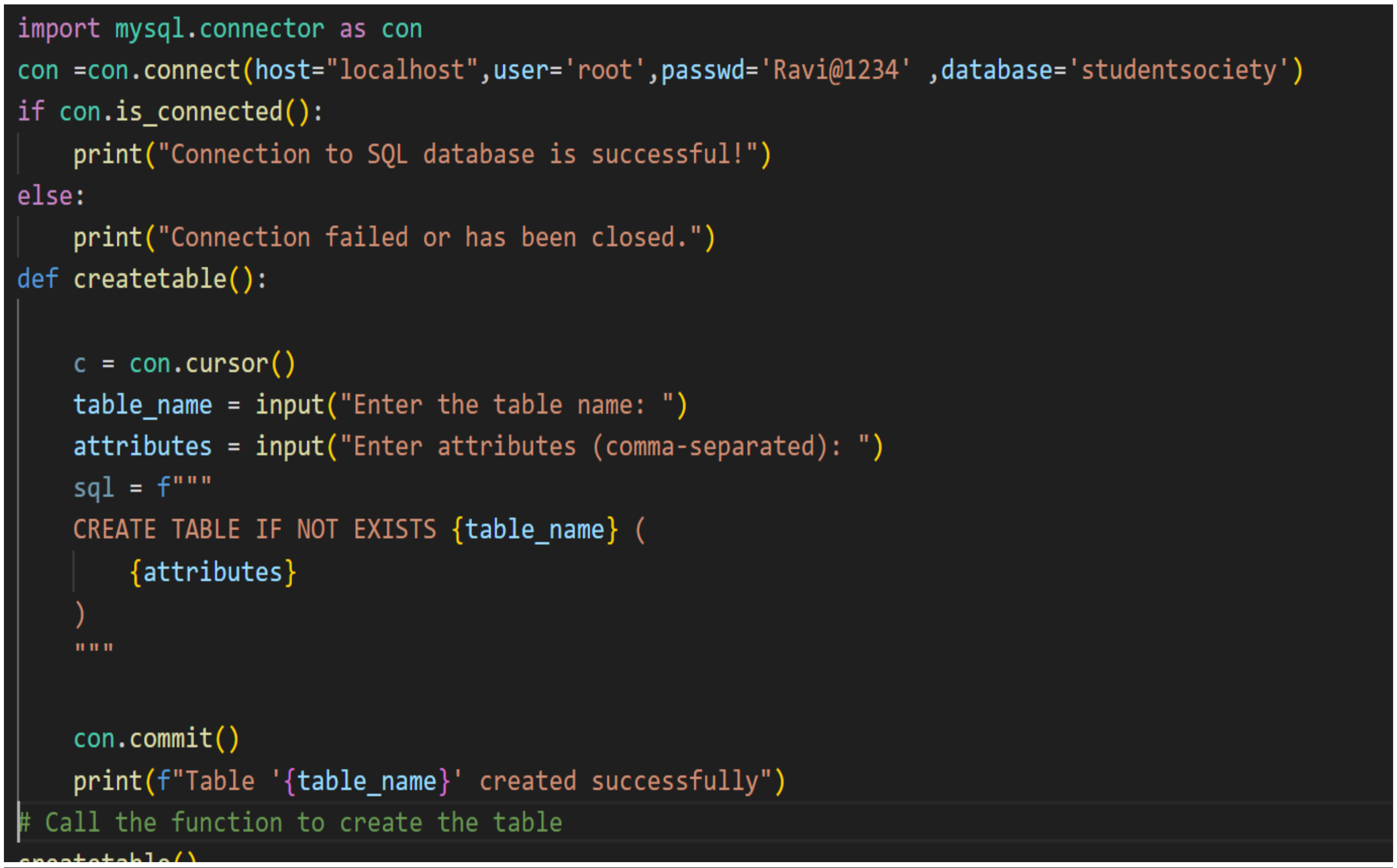


```
mysql> create index IndexName on student(Roll_No);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

## ***(III.) Execute queries given in part I through a high-level language using ODBC connection.***

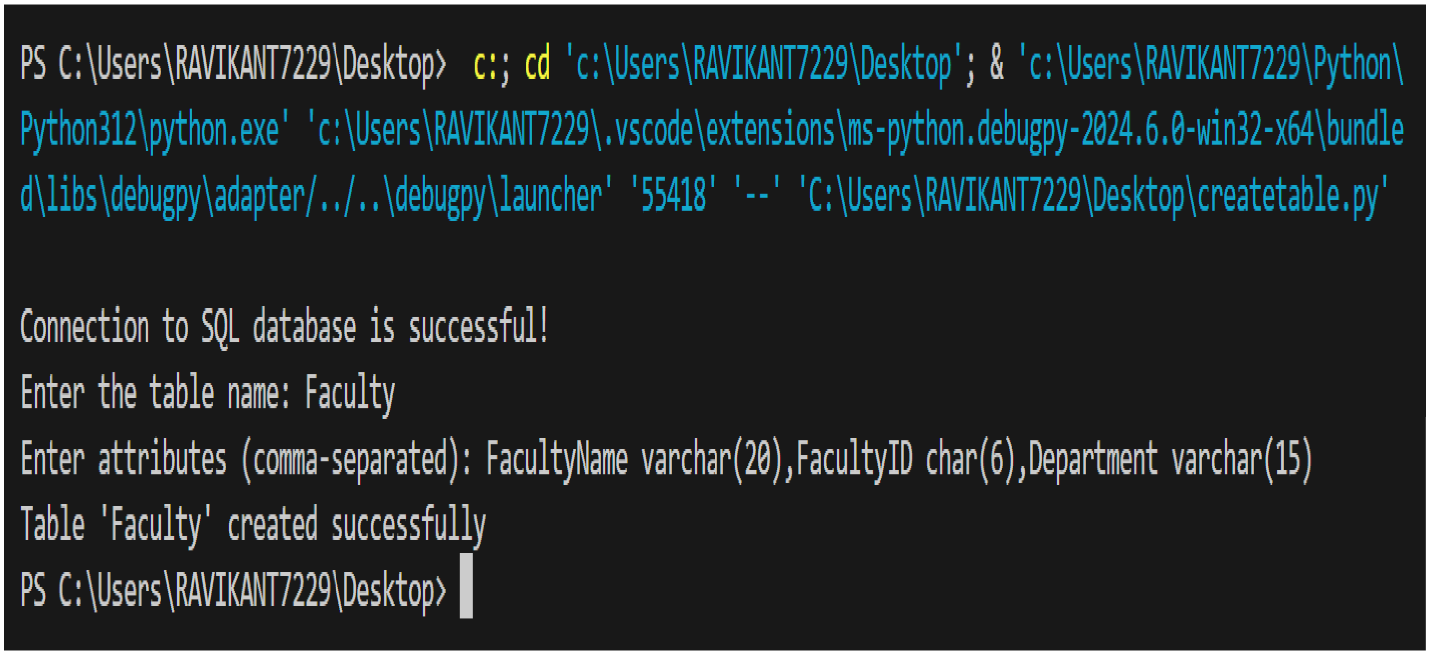
***CREATING A TABLE:***

***Query:***



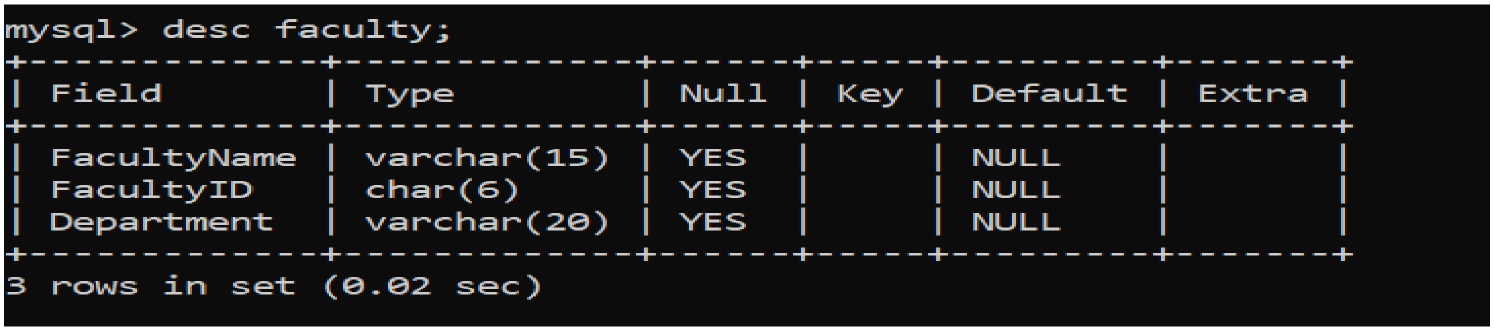
```
import mysql.connector as con
con =con.connect(host="localhost",user='root',passwd='Ravi@1234' ,database='studentsociety')
if con.is_connected():
    print("Connection to SQL database is successful!")
else:
    print("Connection failed or has been closed.")
def createtable():

    c = con.cursor()
    table_name = input("Enter the table name: ")
    attributes = input("Enter attributes (comma-separated): ")
    sql = f"""
    CREATE TABLE IF NOT EXISTS {table_name} (
        {attributes}
    )
    """

    con.commit()
    print(f"Table '{table_name}' created successfully")
# Call the function to create the table
```

***Qutput:***



```
PS C:\Users\RAVIKANT7229\Desktop>  c:; cd 'c:\Users\RAVIKANT7229\Desktop'; & 'c:\Users\RAVIKANT7229\Python\Python312\python.exe' 'c:\Users\RAVIKANT7229\.vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '55418' '--' 'C:\Users\RAVIKANT7229\Desktop\createtable.py'

Connection to SQL database is successful!
Enter the table name: Faculty
Enter attributes (comma-separated): FacultyName varchar(20),FacultyID char(6),Department varchar(15)
Table 'Faculty' created successfully
PS C:\Users\RAVIKANT7229\Desktop>
```

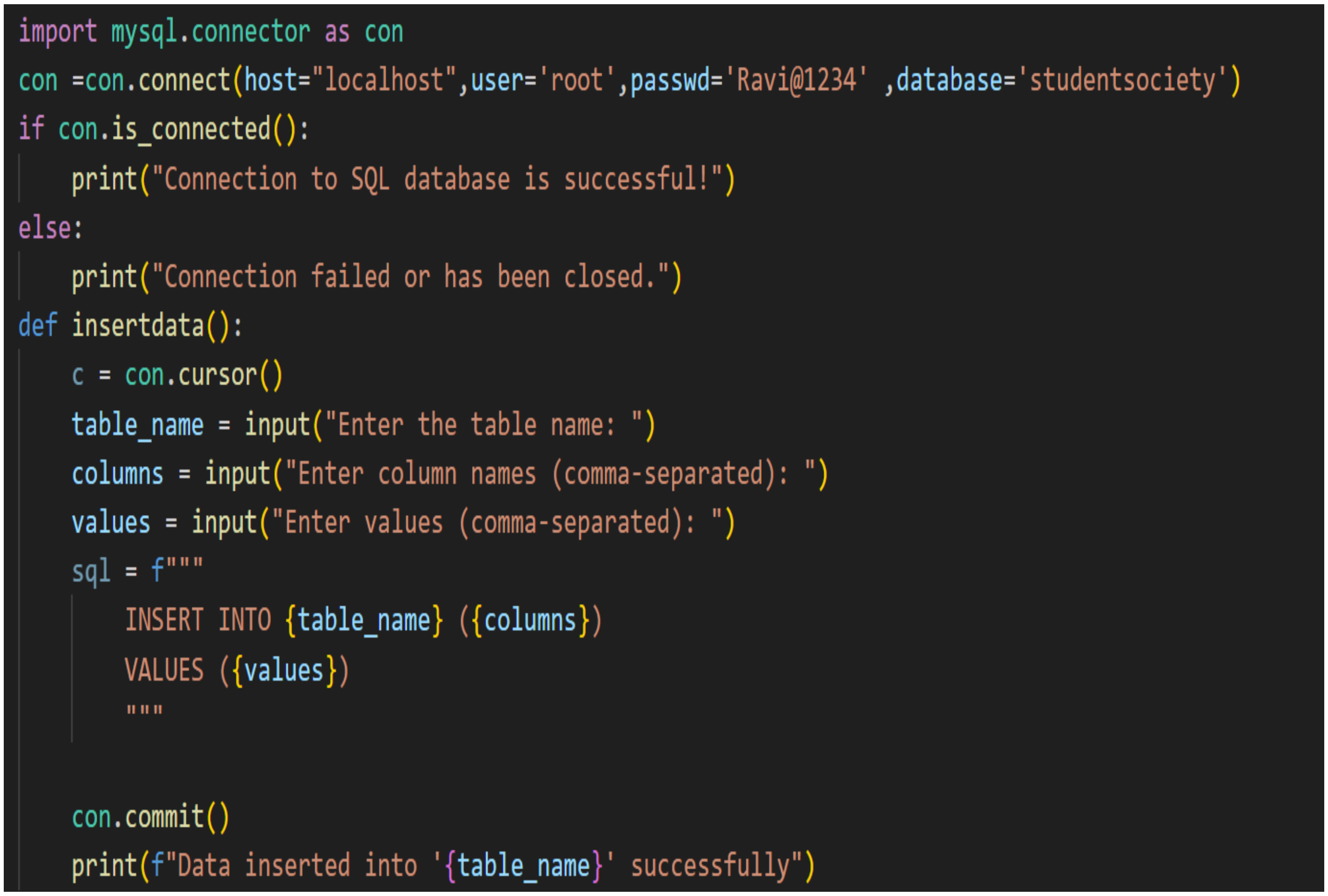
***Verfying:***



```
mysql> desc faculty;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| FacultyName | varchar(15) | YES  |     | NULL    |       |
| FacultyID   | char(6)     | YES  |     | NULL    |       |
| Department  | varchar(20) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.02 sec)
```

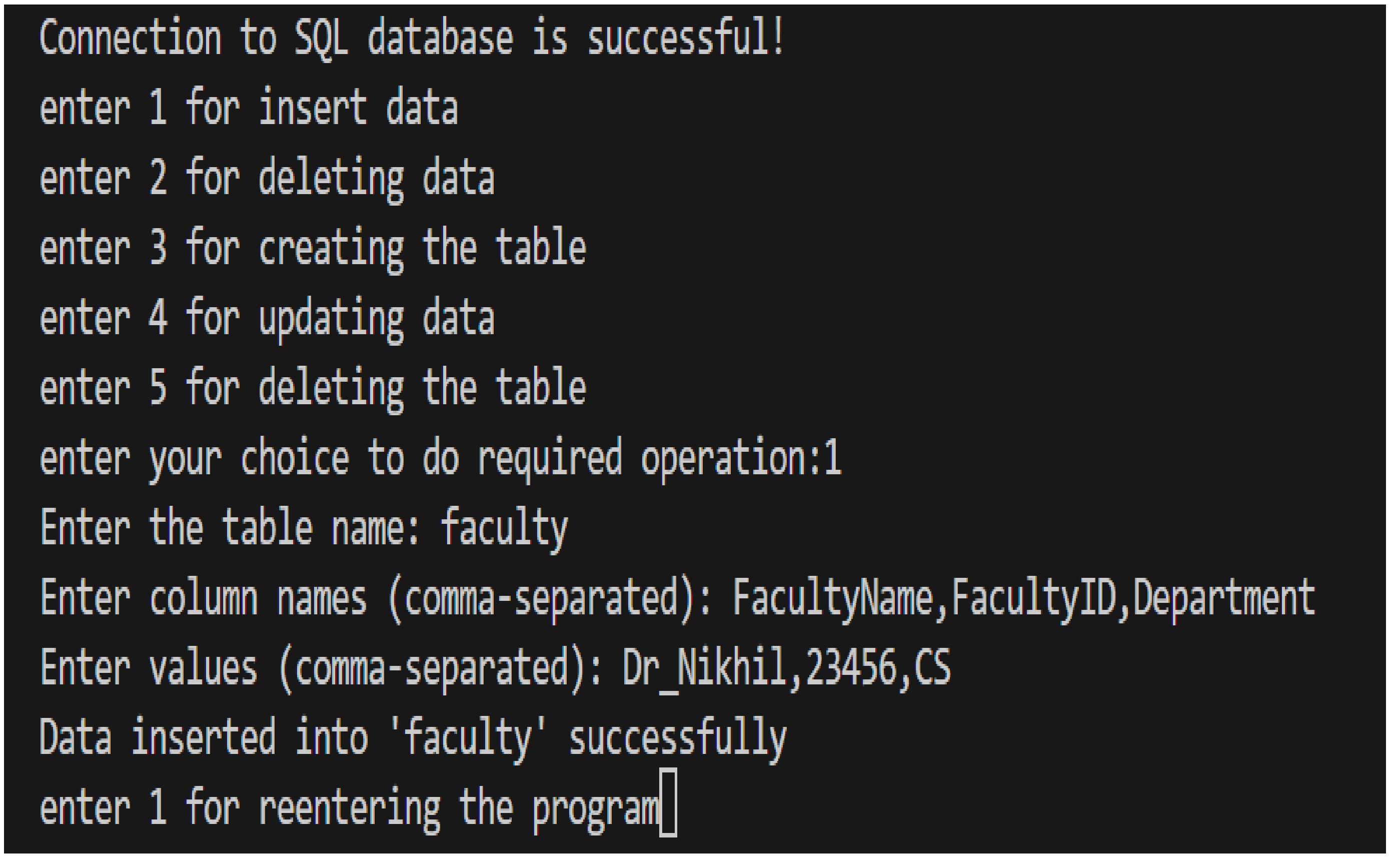
## ***INSERTING INTO  TABLE:***

## ***Query:***



```python
import mysql.connector as con
con =con.connect(host="localhost",user='root',passwd='Ravi@1234' ,database='studentsociety')
if con.is_connected():
    print("Connection to SQL database is successful!")
else:
    print("Connection failed or has been closed.")
def insertdata():
    c = con.cursor()
    table_name = input("Enter the table name: ")
    columns = input("Enter column names (comma-separated): ")
    values = input("Enter values (comma-separated): ")
    sql = f"""
        INSERT INTO {table_name} ({columns})
        VALUES ({values})
        """

    con.commit()
    print(f"Data inserted into '{table_name}' successfully")
```

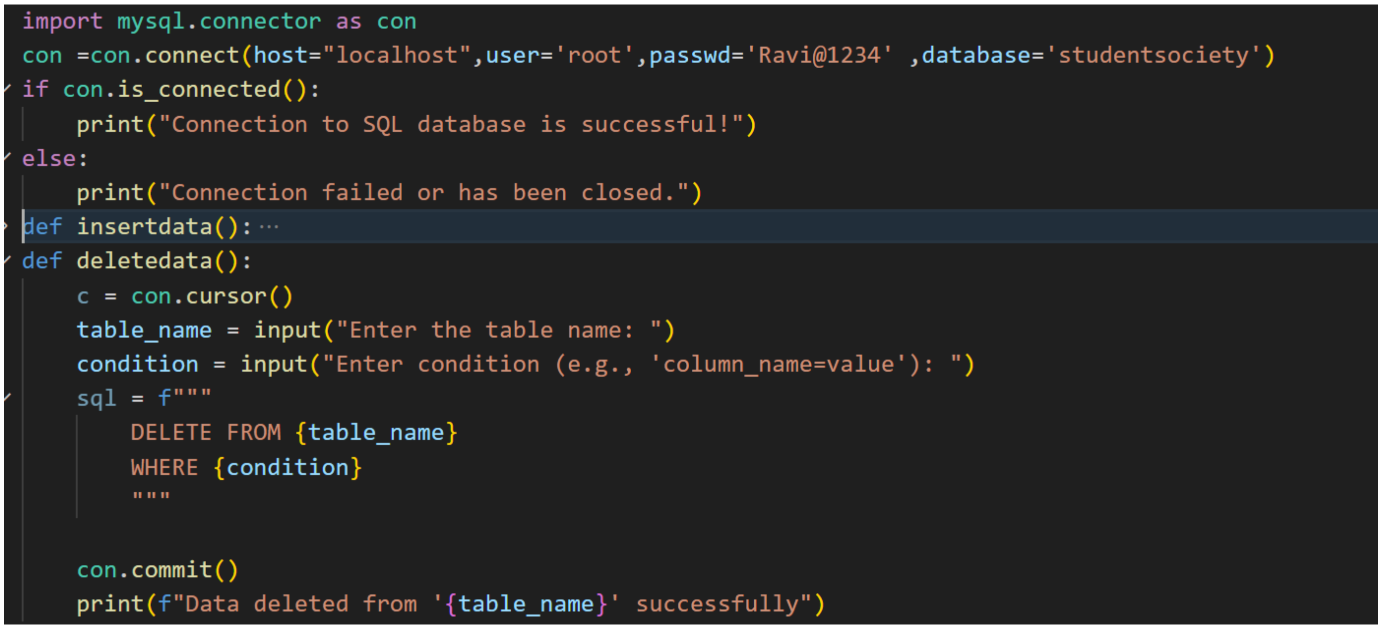
## ***Output:-***



```
Connection to SQL database is successful!
enter 1 for insert data
enter 2 for deleting data
enter 3 for creating the table
enter 4 for updating data
enter 5 for deleting the table
enter your choice to do required operation:1
Enter the table name: faculty
Enter column names (comma-separated): FacultyName,FacultyID,Department
Enter values (comma-separated): Dr_Nikhil,23456,CS
Data inserted into 'faculty' successfully
enter 1 for reentering the program
```

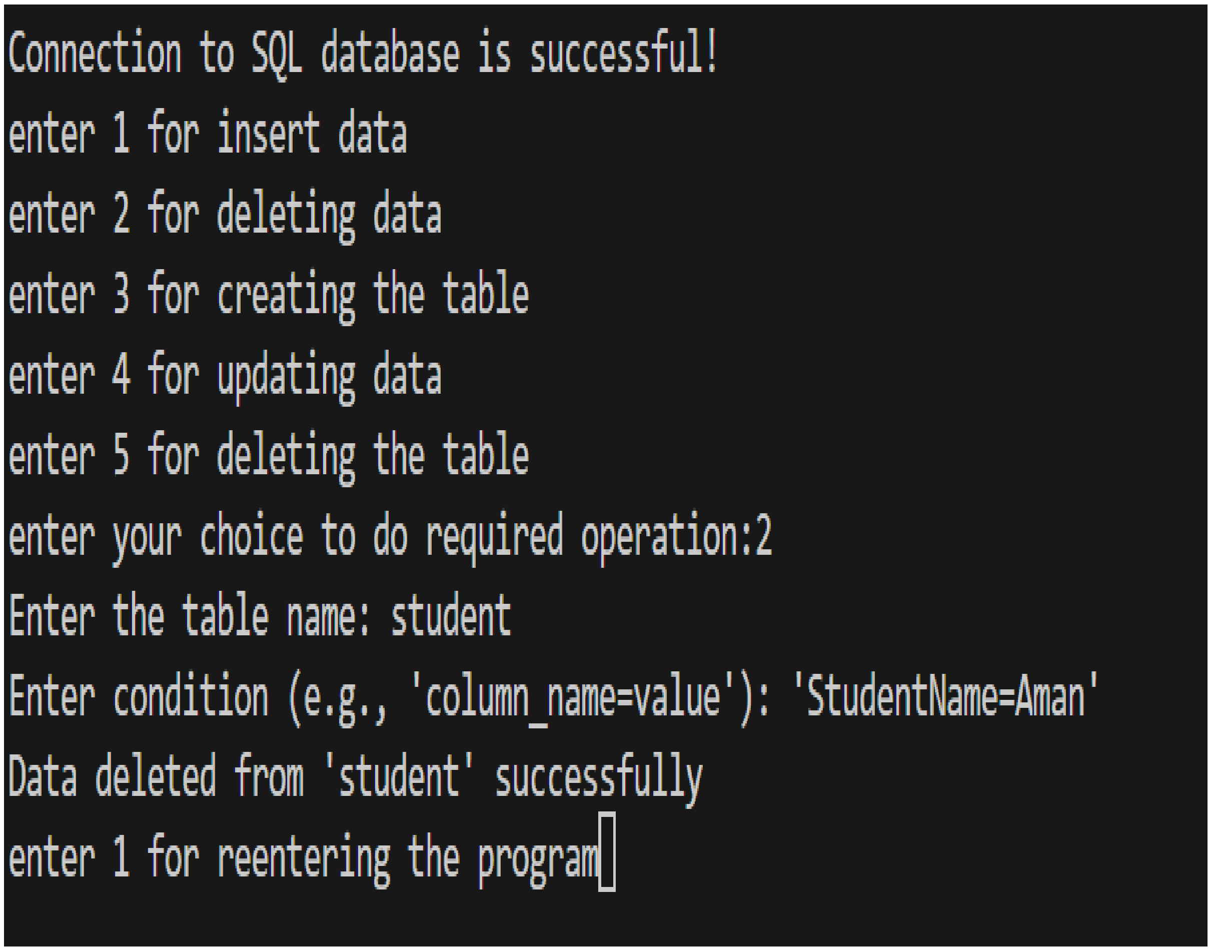
## ***Deleting Data:***

### ***Query:***



```
import mysql.connector as con
con =con.connect(host="localhost",user='root',passwd='Ravi@1234' ,database='studentsociety')
if con.is_connected():
    print("Connection to SQL database is successful!")
else:
    print("Connection failed or has been closed.")
def insertdata():...
def deletedata():
    c = con.cursor()
    table_name = input("Enter the table name: ")
    condition = input("Enter condition (e.g., 'column_name=value'): ")
    sql = f"""
        DELETE FROM {table_name}
        WHERE {condition}
        """

    con.commit()
    print(f"Data deleted from '{table_name}' successfully")
```

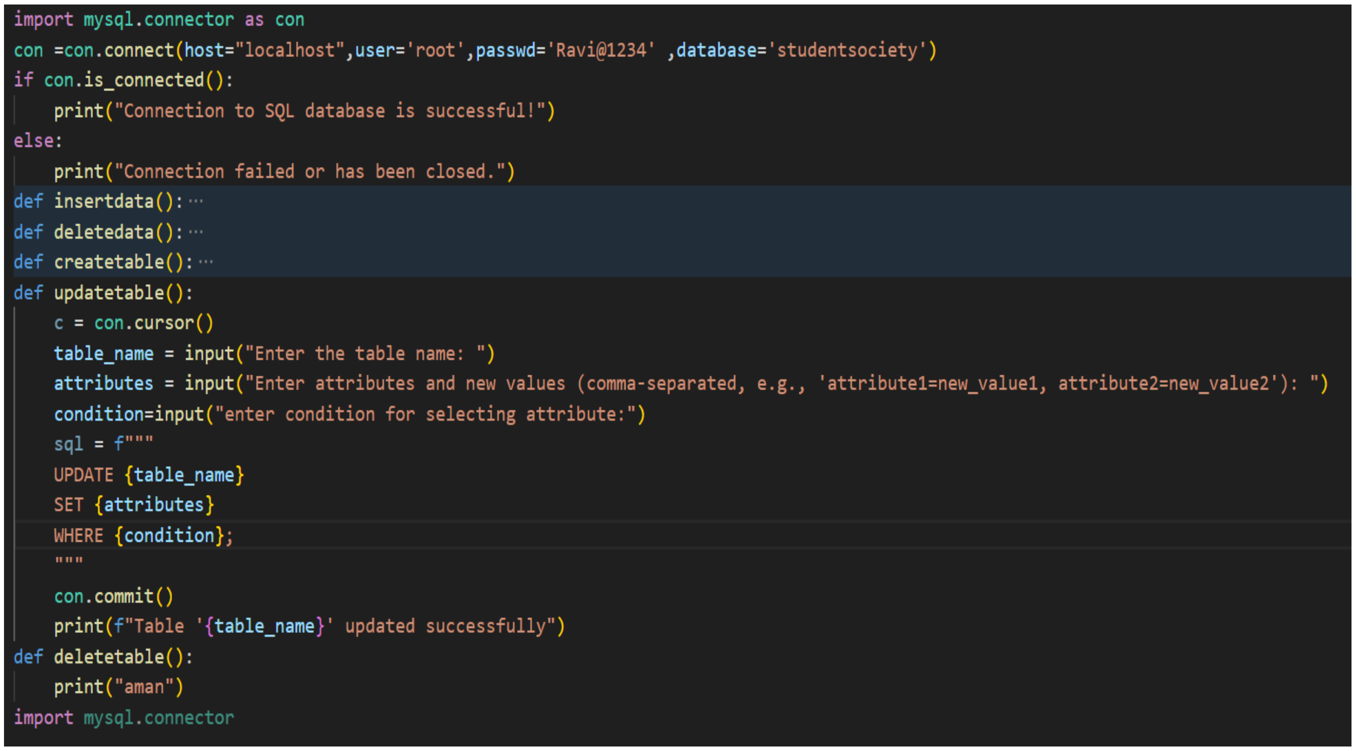
***Output:-***



```
Connection to SQL database is successful!
enter 1 for insert data
enter 2 for deleting data
enter 3 for creating the table
enter 4 for updating data
enter 5 for deleting the table
enter your choice to do required operation:2
Enter the table name: student
Enter condition (e.g., 'column_name=value'): 'StudentName=Aman'
Data deleted from 'student' successfully
enter 1 for reentering the program
```

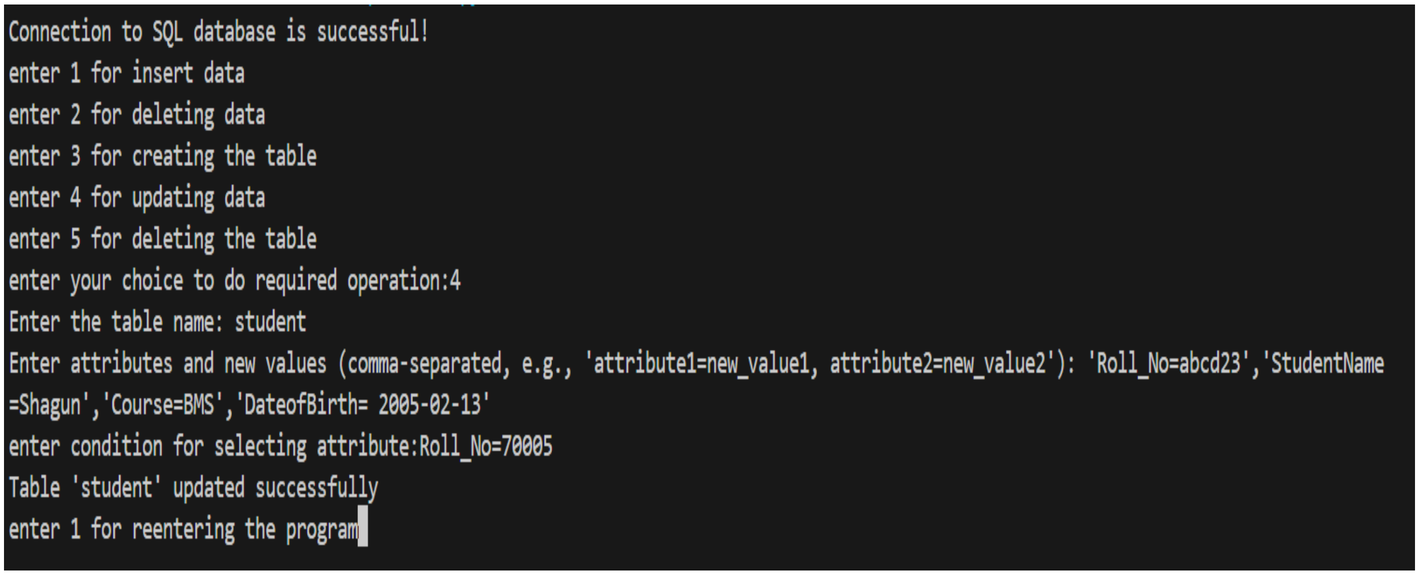
## ***Updating Data:***

## ***Query:***



```
import mysql.connector as con
con =con.connect(host="localhost",user='root',passwd='Ravi@1234' ,database='studentsociety')
if con.is_connected():
    print("Connection to SQL database is successful!")
else:
    print("Connection failed or has been closed.")
def insertdata():⋯
def deletedata():⋯
def createtable():⋯
def updatetable():
    c = con.cursor()
    table_name = input("Enter the table name: ")
    attributes = input("Enter attributes and new values (comma-separated, e.g., 'attribute1=new_value1, attribute2=new_value2'): ")
    condition=input("enter condition for selecting attribute:")
    sql = f"""
    UPDATE {table_name}
    SET {attributes}
    WHERE {condition};
    """
    con.commit()
    print(f"Table '{table_name}' updated successfully")
def deletetable():
    print("aman")
import mysql.connector
```

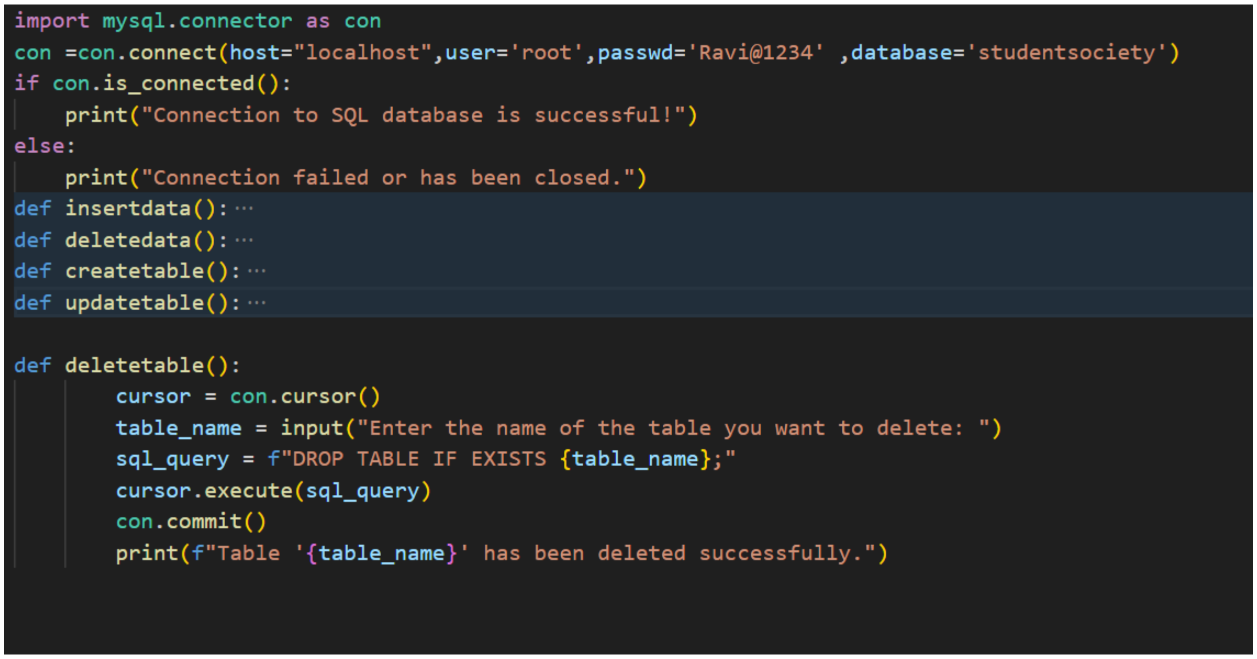
## ***Output:***



```
Connection to SQL database is successful!
enter 1 for insert data
enter 2 for deleting data
enter 3 for creating the table
enter 4 for updating data
enter 5 for deleting the table
enter your choice to do required operation:4
Enter the table name: student
Enter attributes and new values (comma-separated, e.g., 'attribute1=new_value1, attribute2=new_value2'): 'Roll_No=abcd23','StudentName
=Shagun','Course=BMS','DateofBirth= 2005-02-13'
enter condition for selecting attribute:Roll_No=70005
Table 'student' updated successfully
enter 1 for reentering the program
```

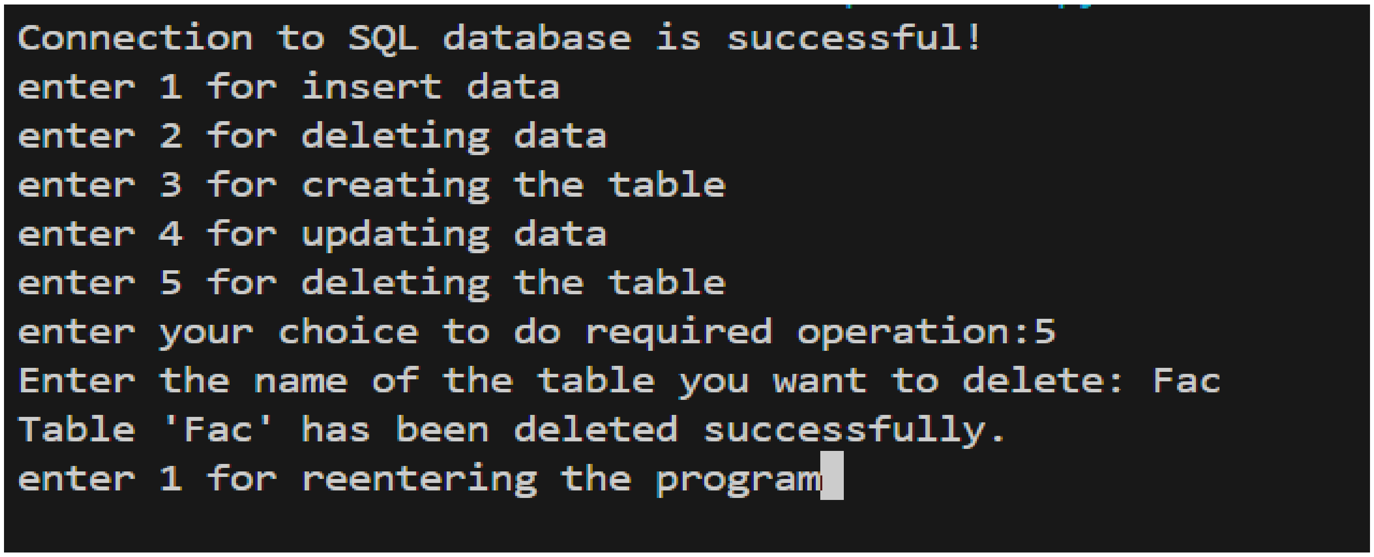
## ***Deleting Table:***

## ***Query:***



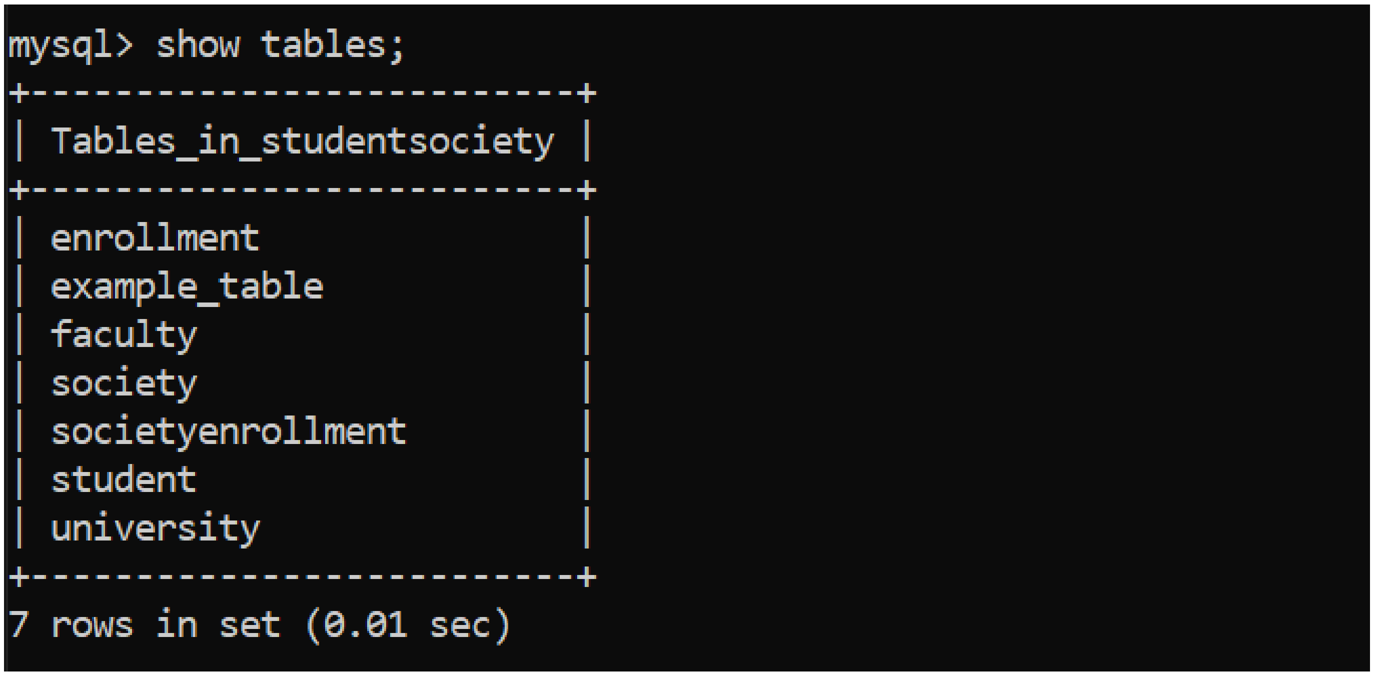
```
import mysql.connector as con
con =con.connect(host="localhost",user='root',passwd='Ravi@1234' ,database='studentsociety')
if con.is_connected():
    print("Connection to SQL database is successful!")
else:
    print("Connection failed or has been closed.")
def insertdata():···
def deletedata():···
def createtable():···
def updatetable():···

def deletetable():
        cursor = con.cursor()
        table_name = input("Enter the name of the table you want to delete: ")
        sql_query = f"DROP TABLE IF EXISTS {table_name};"
        cursor.execute(sql_query)
        con.commit()
        print(f"Table '{table_name}' has been deleted successfully.")
```

## ***Output:***



```
Connection to SQL database is successful!
enter 1 for insert data
enter 2 for deleting data
enter 3 for creating the table
enter 4 for updating data
enter 5 for deleting the table
enter your choice to do required operation:5
Enter the name of the table you want to delete: Fac
Table 'Fac' has been deleted successfully.
enter 1 for reentering the program
```

## ***Verfying:***



```
mysql> show tables;
+---------------------------+
| Tables_in_studentsociety |
+---------------------------+
| enrollment                |
| example_table             |
| faculty                   |
| society                   |
| societyenrollment         |
| student                   |
| university                |
+---------------------------+
7 rows in set (0.01 sec)
```