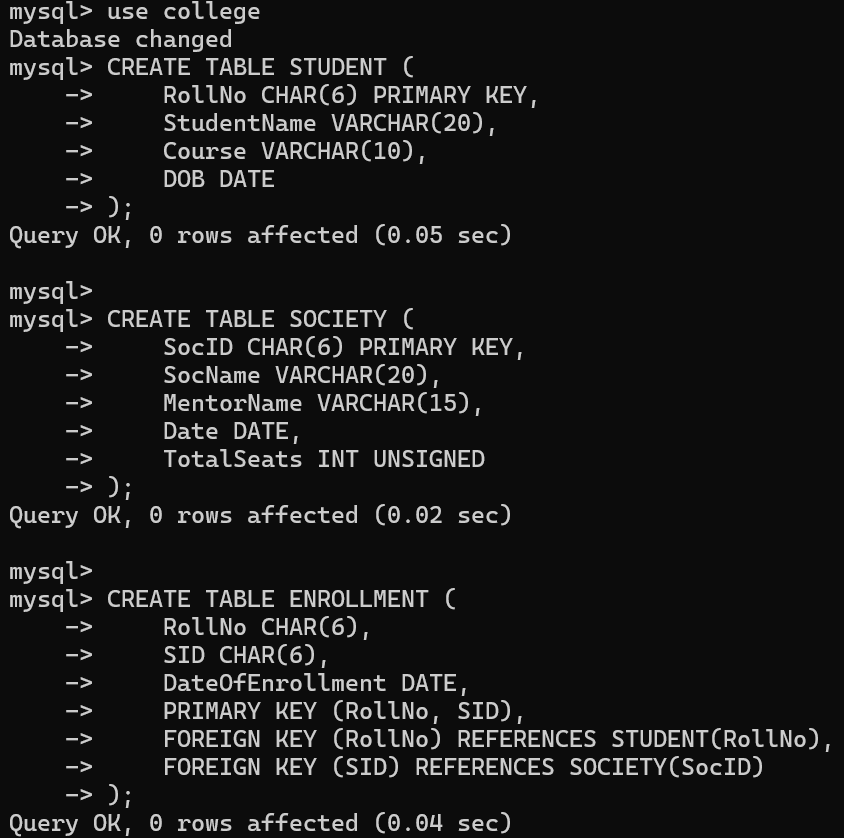
DBMS Practical file

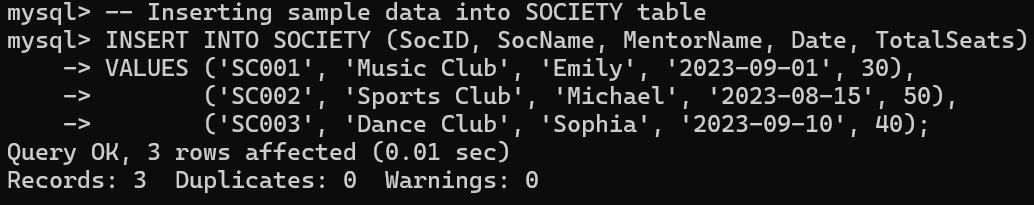
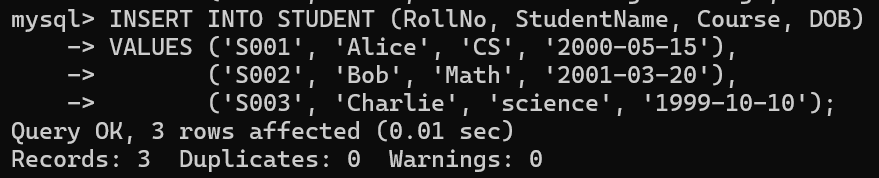
Name: KUR ATHUAI ACHUIL KUOL

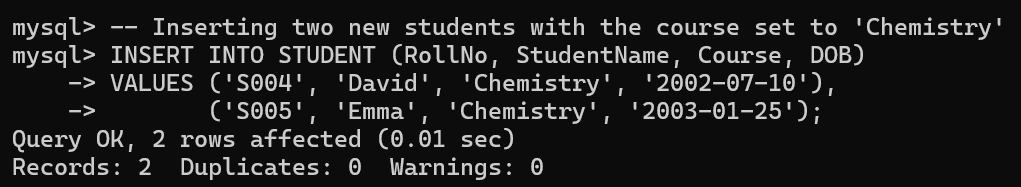
Roll no:20221423

Course: B.sc Computer science

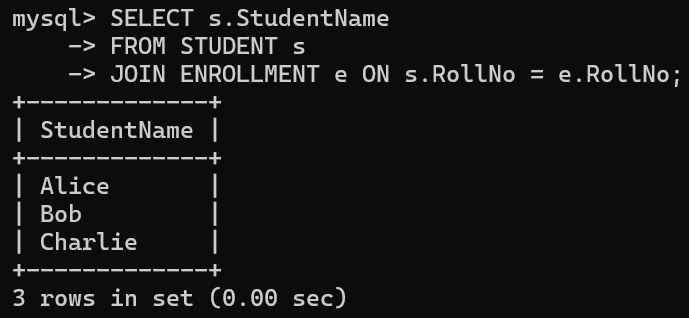
Q1: I. Create and use the following student-society database schema for a college to answer the given (sample) queries using the standalone SQL editor.



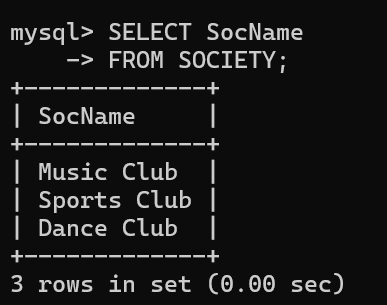




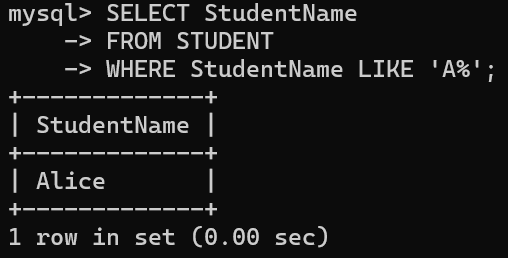
Query 1: Retrieve names of students enrolled in any society.



Query 2: Retrieve all society names.

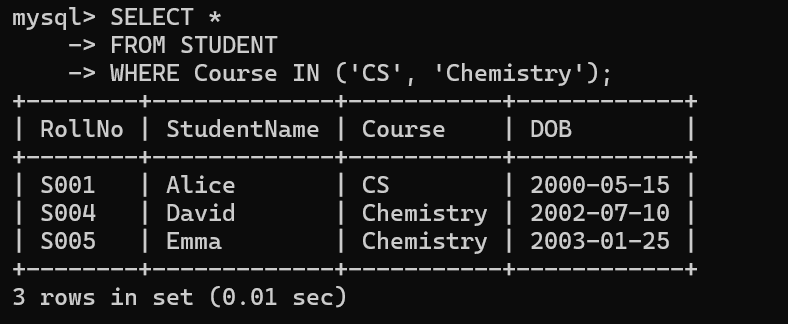


Query 3: Retrieve students' names starting with letter ‘A’



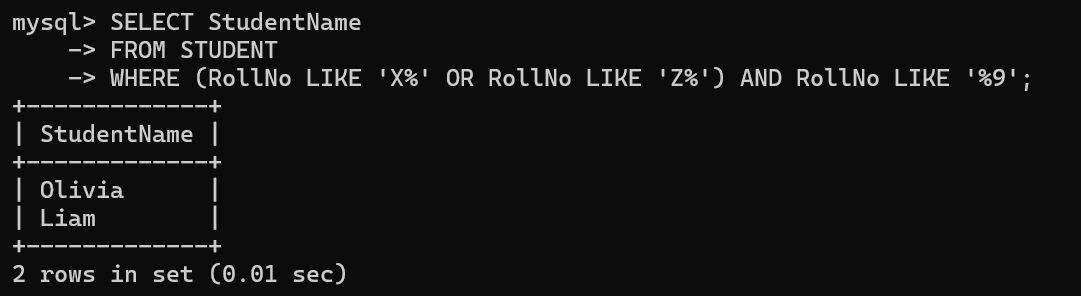
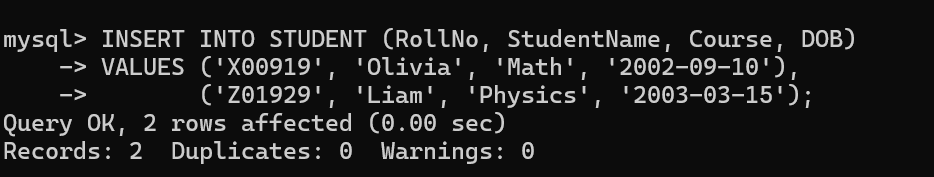
Query 4: Retrieve students' details studying in courses

‘computer science’ or ‘chemistry’.

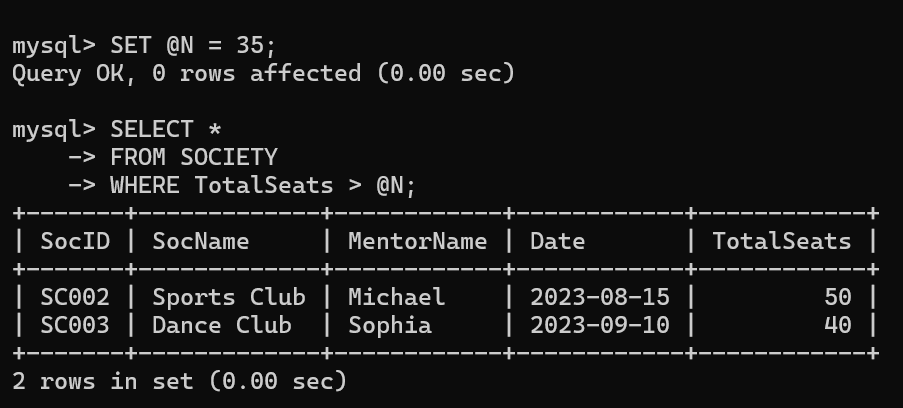


Query 5:

Retrieve students’ names whose roll no either starts with ‘X’ or ‘Z’ and ends with ‘9’.



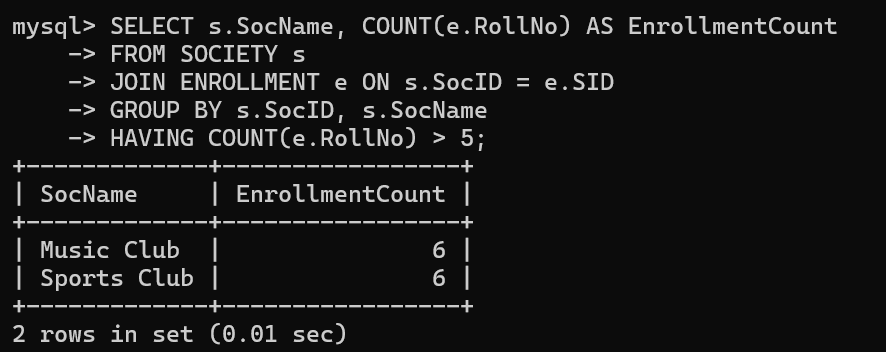
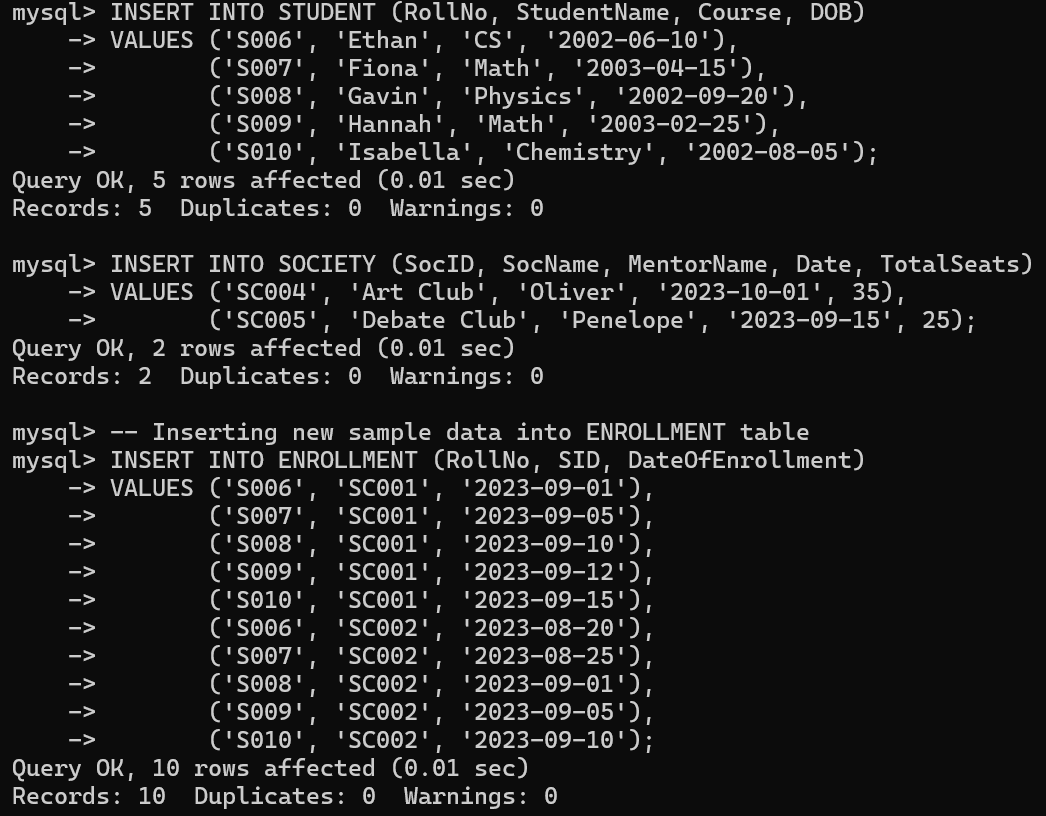
Query 6: Find society details with more than N TotalSeats where N is to be input by the user.



Query 7: Update society table for mentor name of a specific society

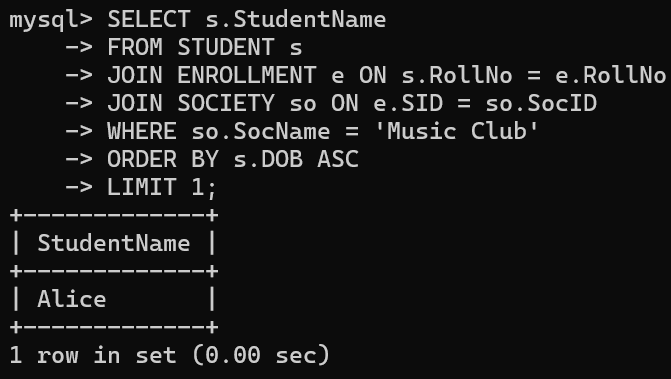


Query 8: Find society names in which more than five students have enrolled.

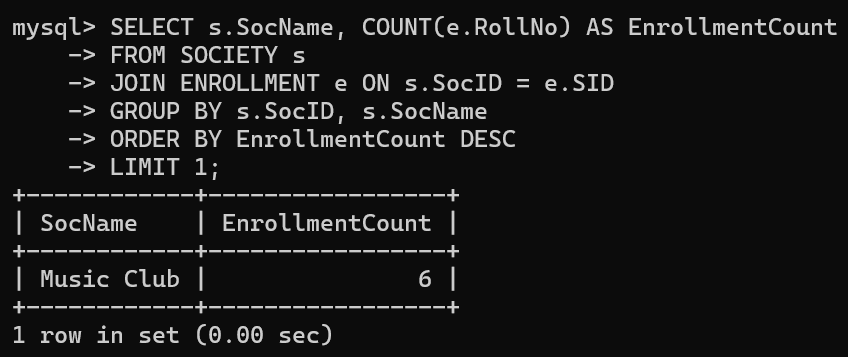


Query 9: Find the name of youngest student enrolled in

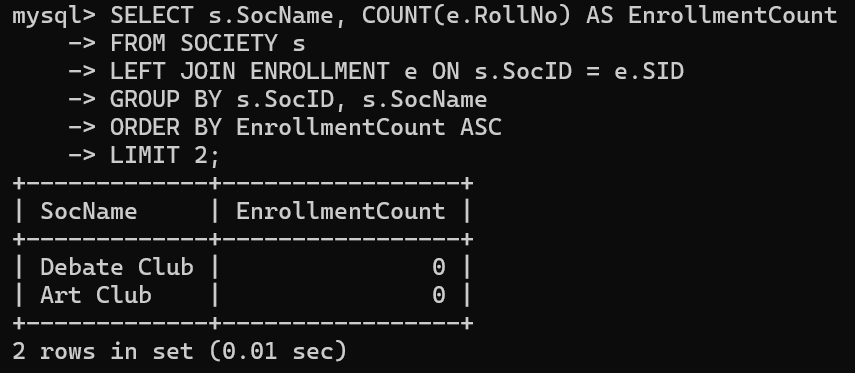
society ‘music club’



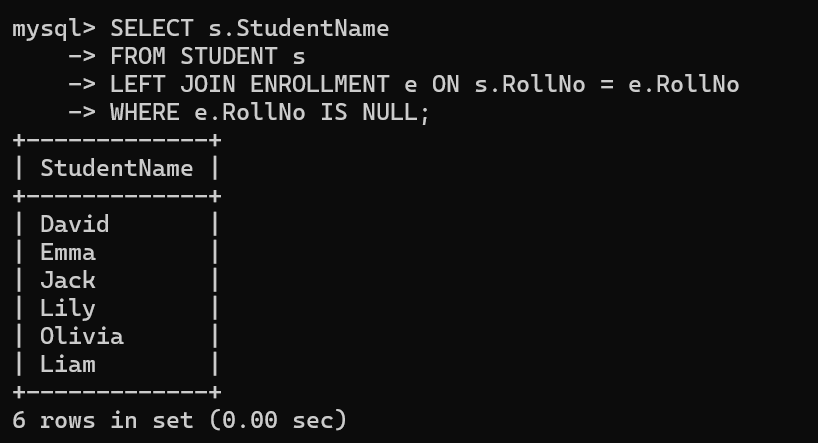
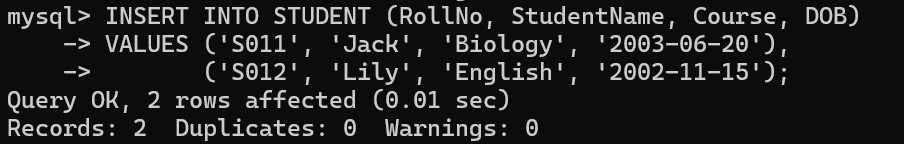
Query 10: Find the name of most popular society (on the basis of enrolled students)



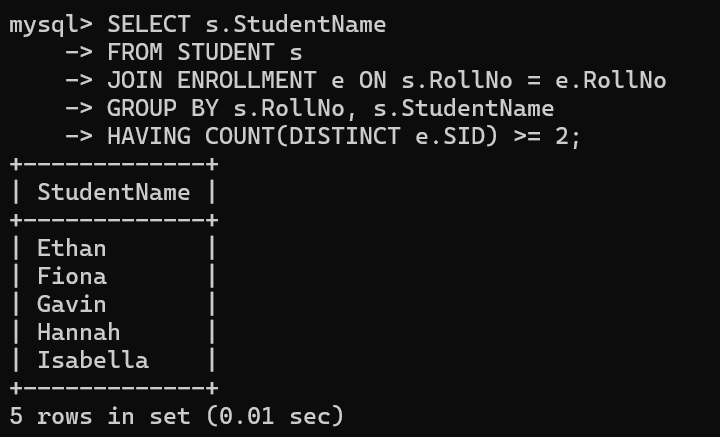
Query 11: Find the name of two least popular societies (on the basis of enrolled students)



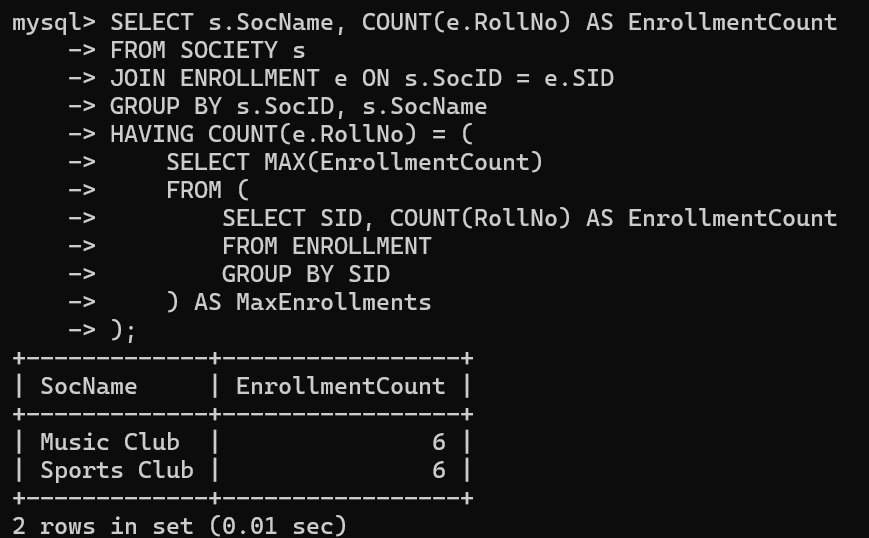
Query 12: Find the student names who are not enrolled in any society



Query 13: Find the student names enrolled in at least two societies



Query 14: Find society names in which maximum students are enrolled

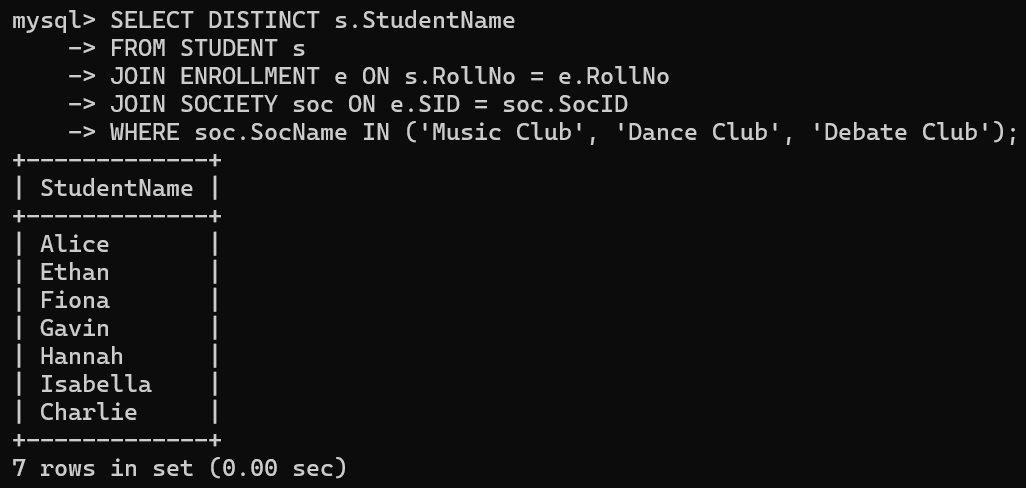


Query 15: Find names of all students who have enrolled in any society and society names in which at least one student has enrolled



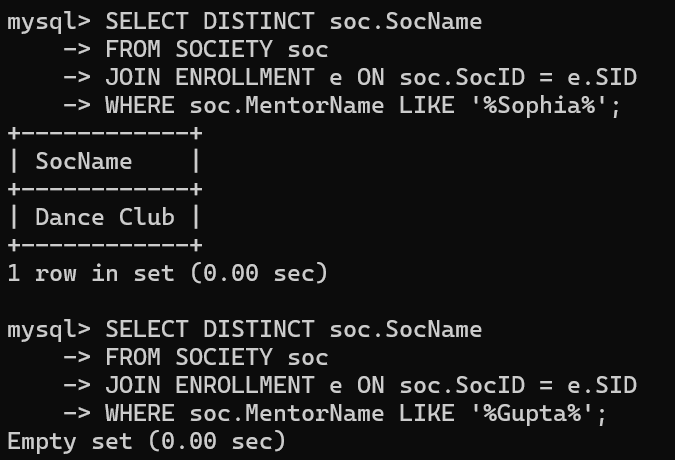
Query 16: Find names of students who are enrolled in any of

the three societies ‘Debating’, ‘Dancing’ and ‘music’

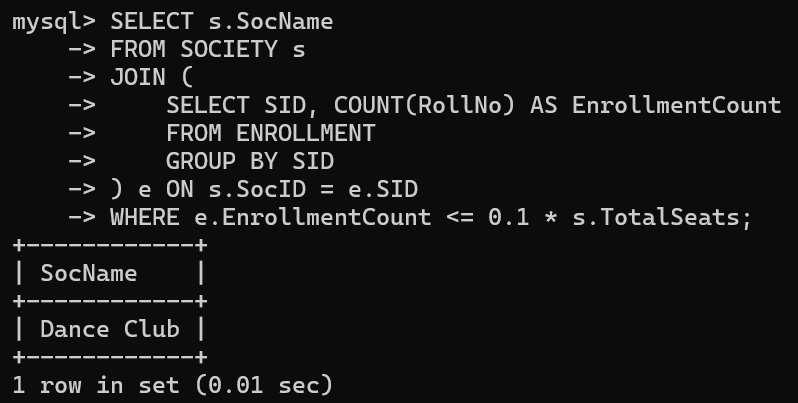


Query 17: Find society names such that its mentor has a

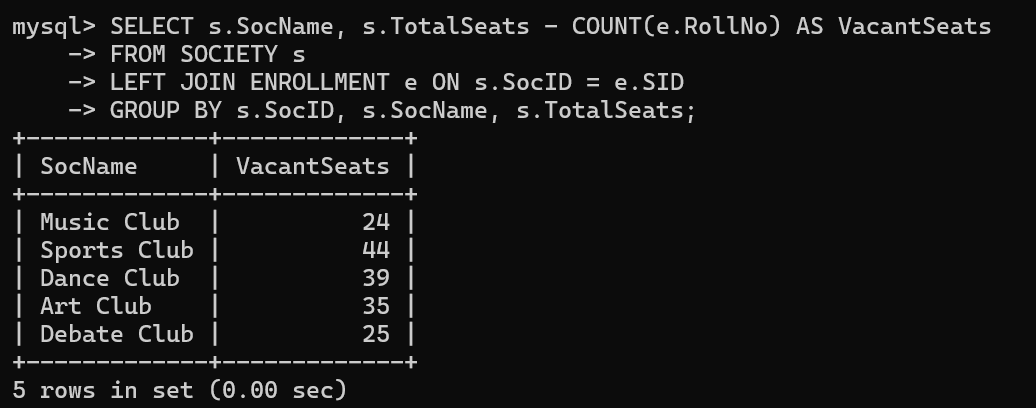
name with ‘’ in it.



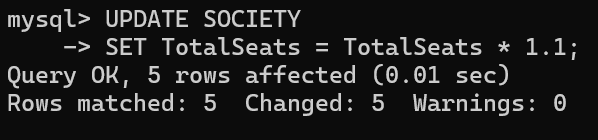
Query 18: Find the society names in which the number of enrolled students is only 10% of its capacity.

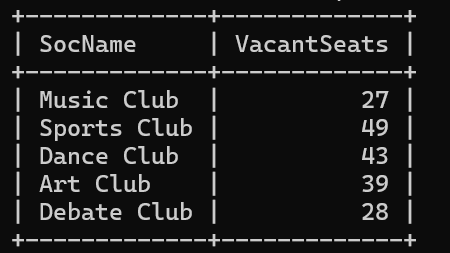


Query 19: Display the vacant seats for each society.



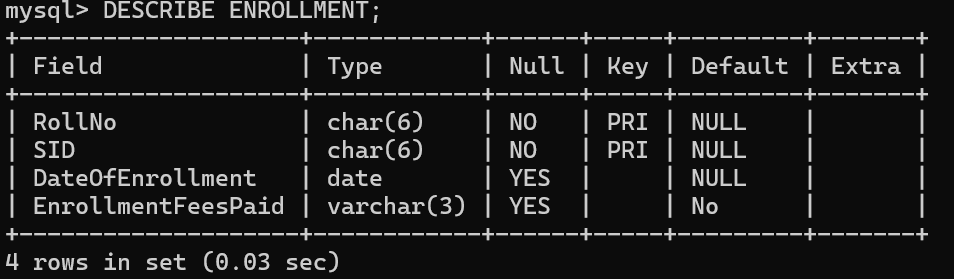
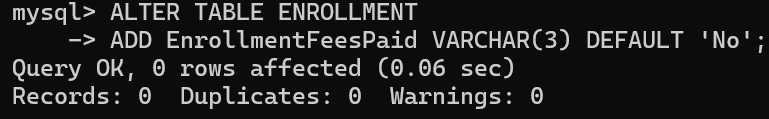
Query 20: Increment Total Seats of each society by 10%



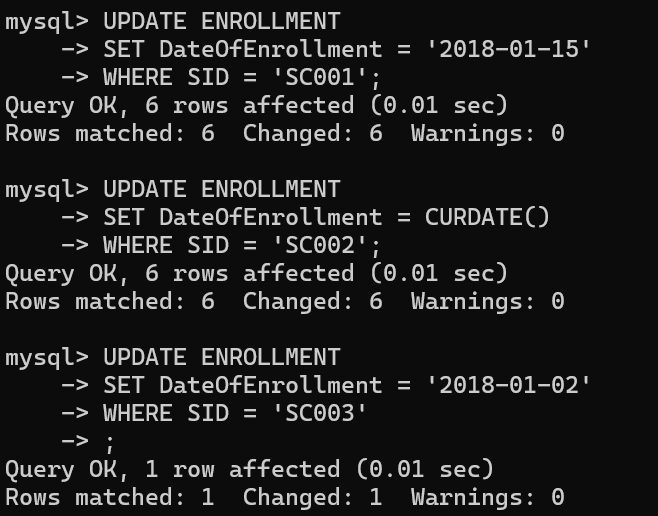


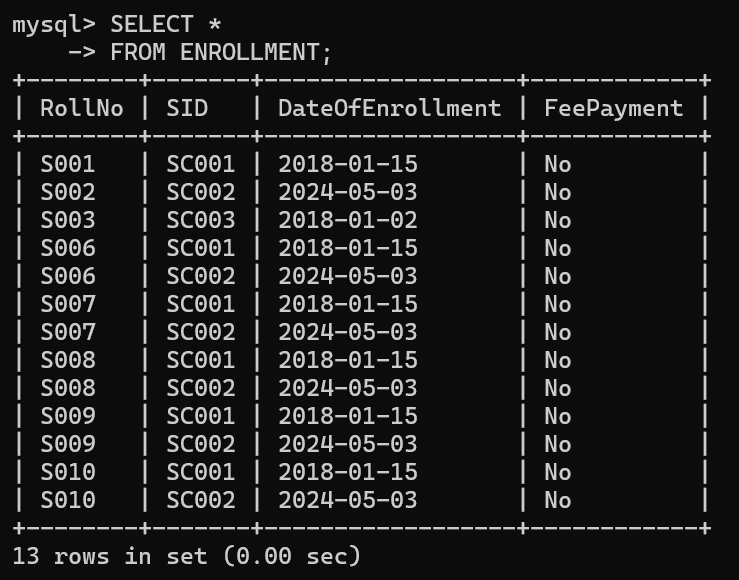
Query 21: Add the enrollment fees paid (‘yes’/’No’) field in

the enrollment table.

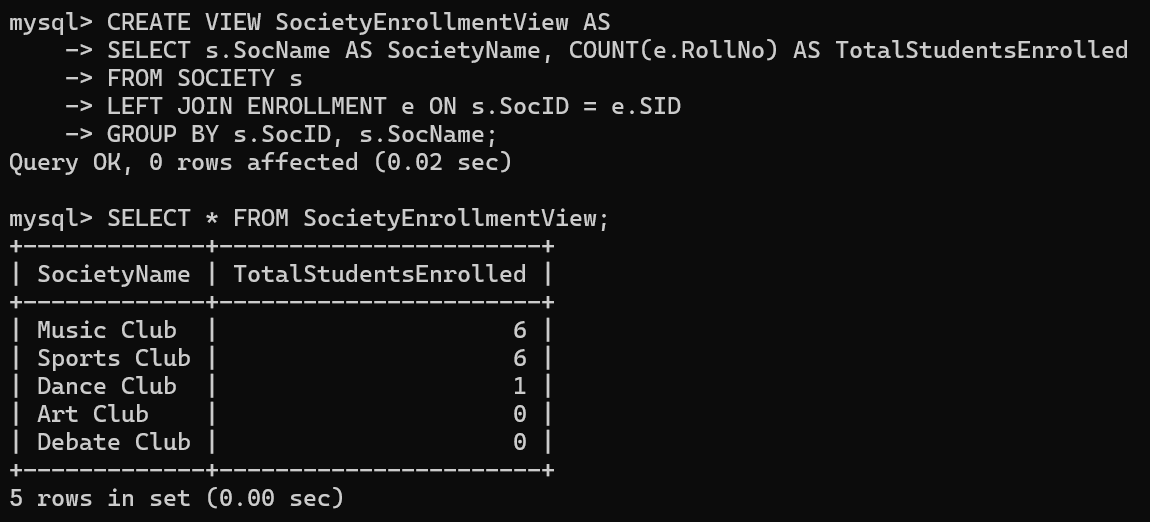


Query 22: Update date of enrollment of society id ‘s1’ to ‘2018-01-15’, ‘s2’ to current date and ‘s3’ to ‘2018-01-02’.

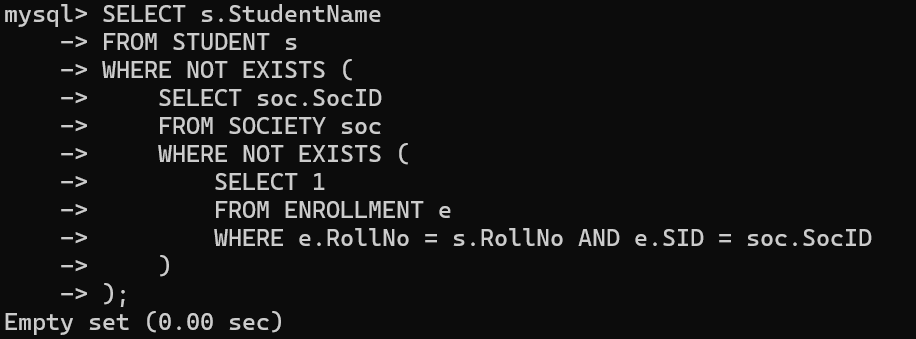


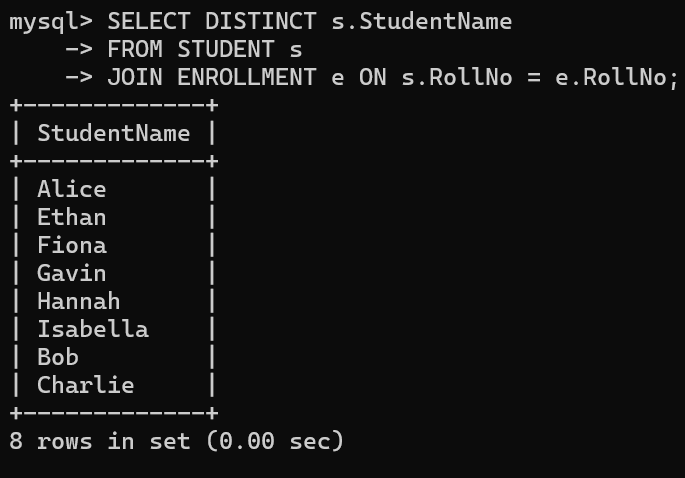


Query 23: Create a view to keep track of society names with the total number of students enrolled in it.

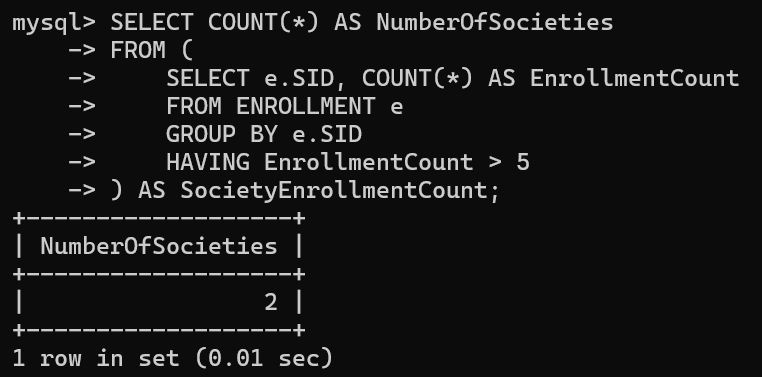


Query 24: Find student names enrolled in all the societies.



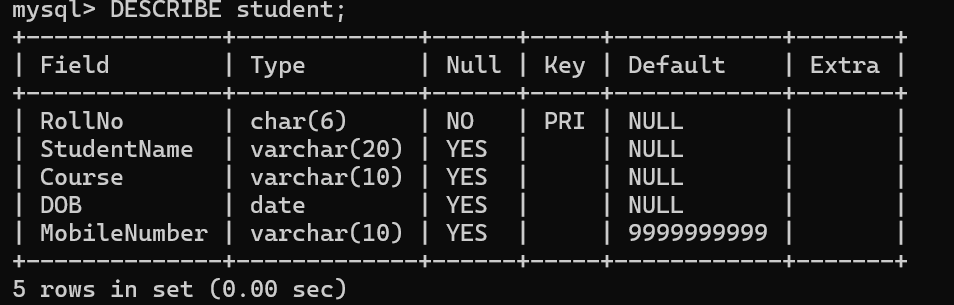


Query 25: Count the number of societies with more than 5 students enrolled in it

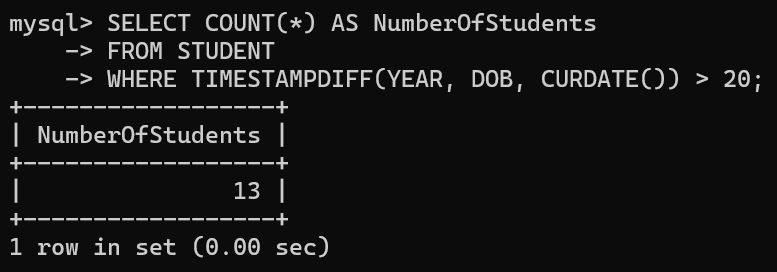


Query 26: Add column Mobile number in student table with

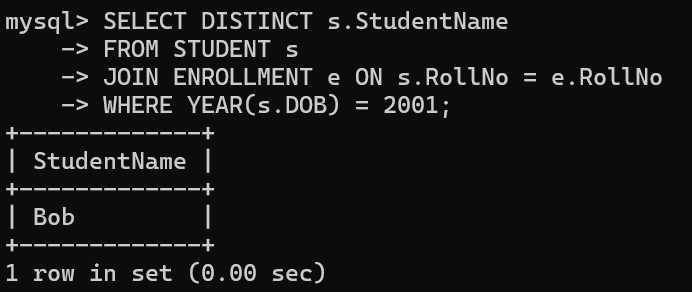
default value ‘9999999999’



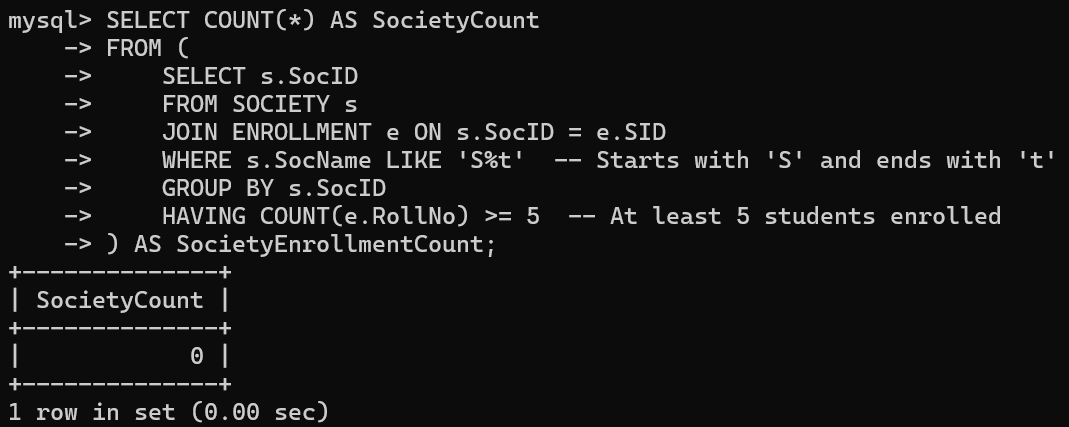
Query 27: Find the total number of students whose age is > 20 years.



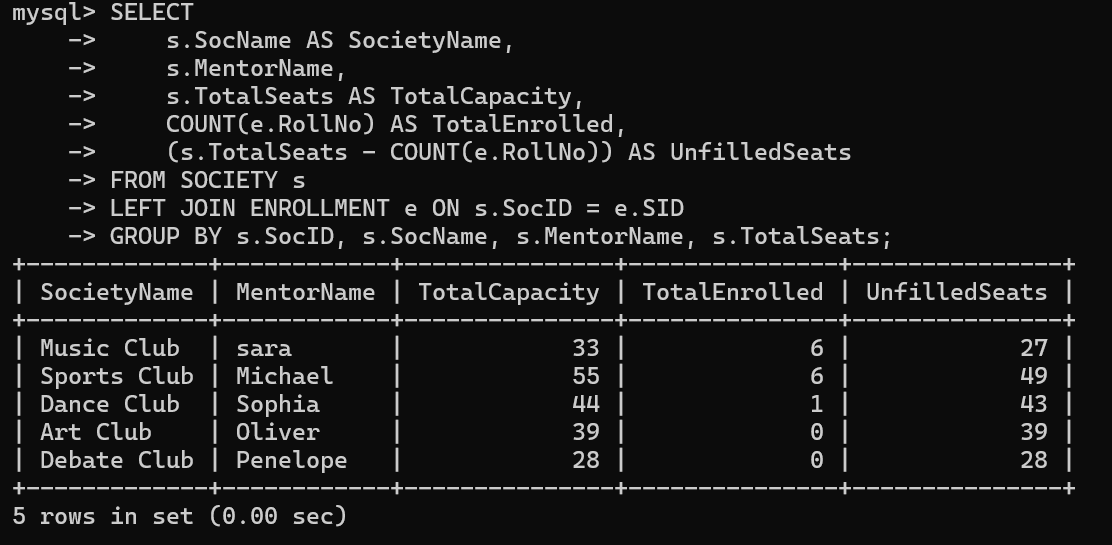
Query 28: Find names of students who are born in 2001 and are enrolled in at least one society.



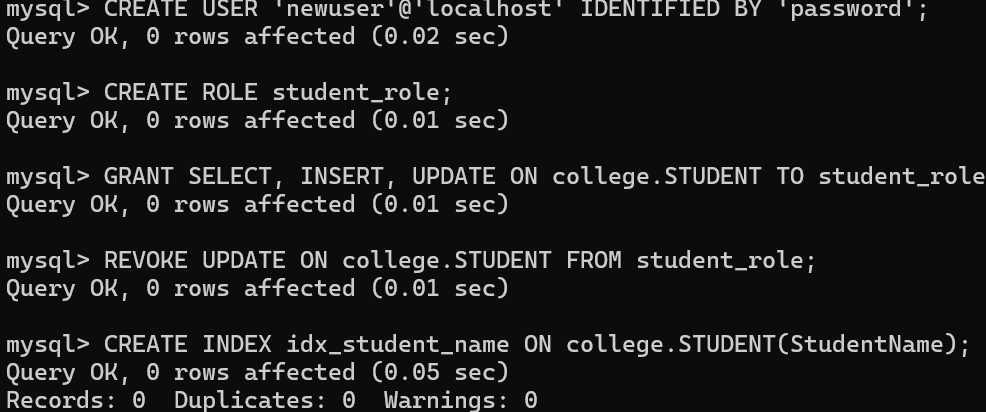
Query 29: Count all societies whose name starts with ‘S’ and ends with ‘t’ and at least 5 students are enrolled in the society.



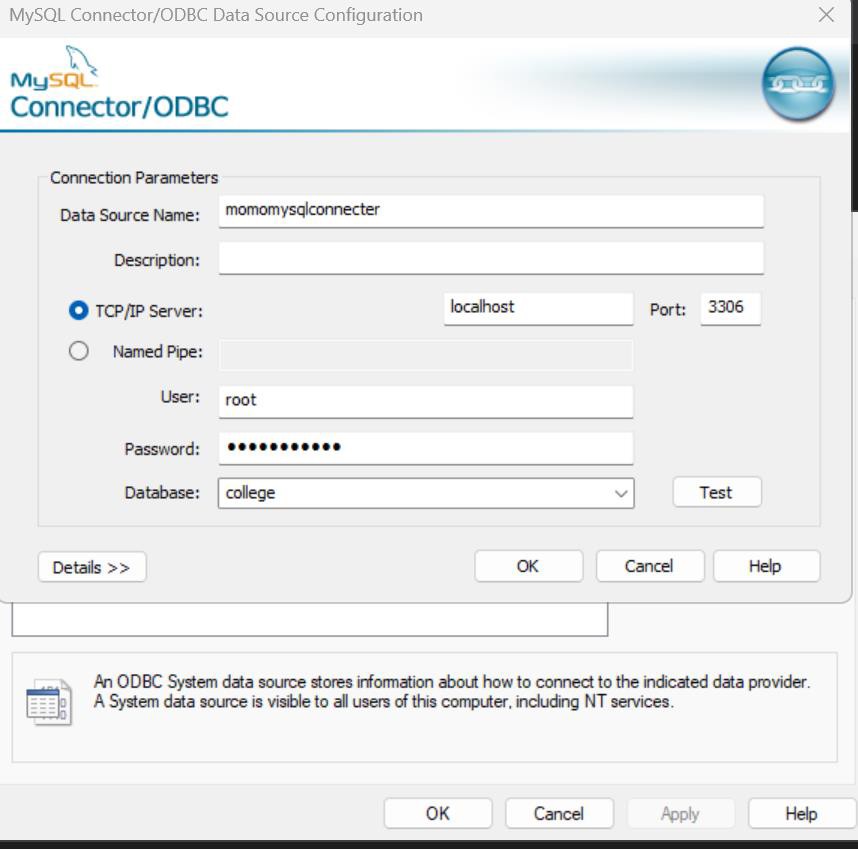
Query 30: Display the following information: Society name Mentor name Total Capacity Total Enrolled Unfilled Seats



Q2: Do the following database administration commands: create user, create role, grant privileges to a role, revoke privileges from a role, create index



Q3: Execute queries given in part I through a high-level language using ODBC connection.



## The code:

impcrt pyodbc

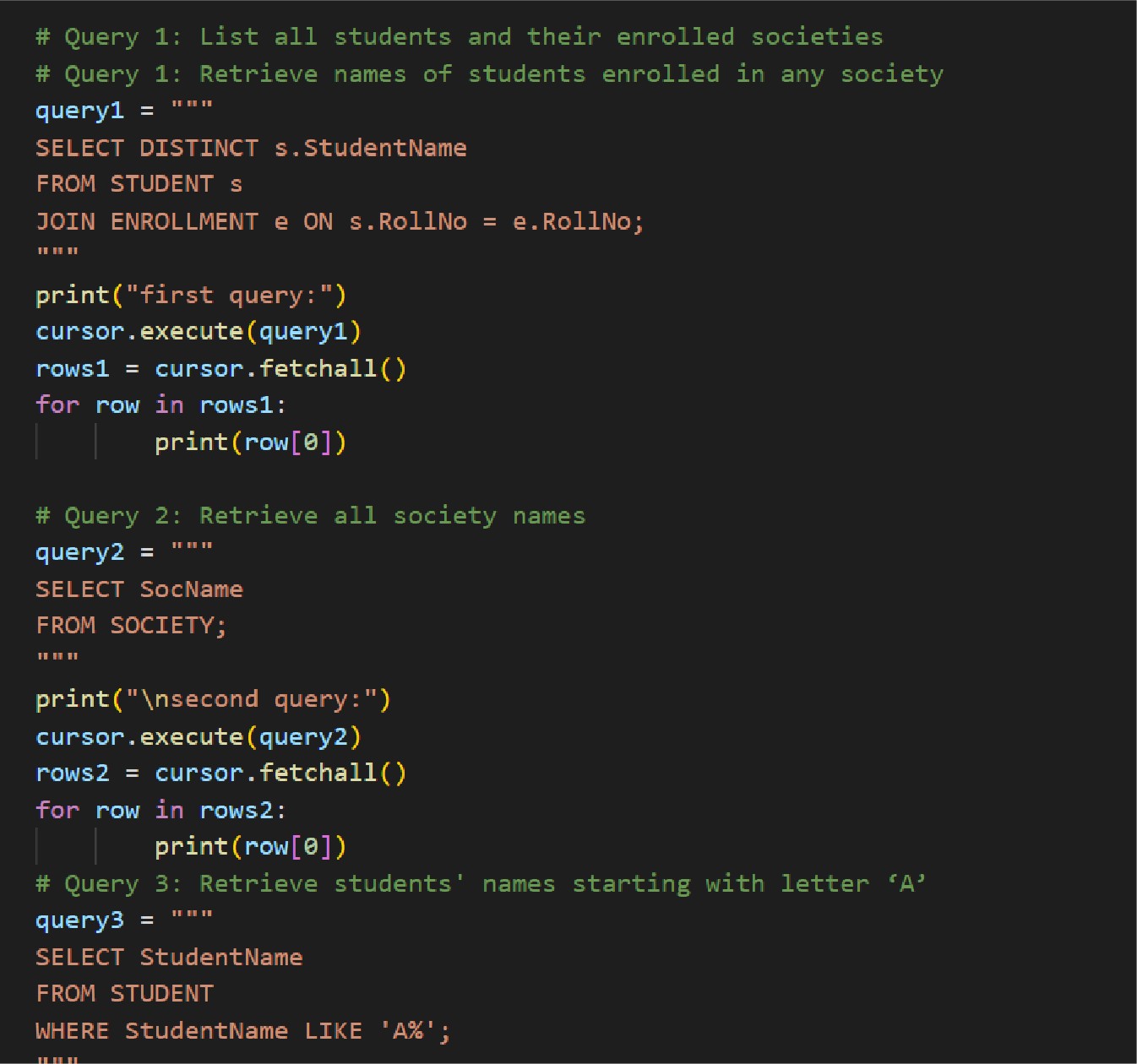
**conn= pyodbc.connect(**'JSN=momonysqlcon ec:er;U:J=root; WJ=· ' )

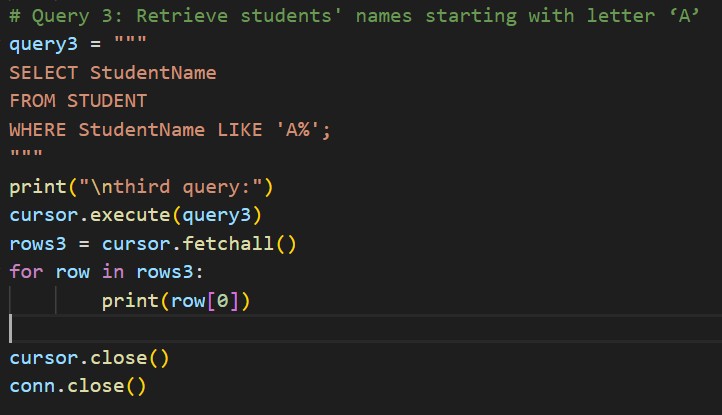
**cursor= conn.cursor()**

**queryl** = "··"

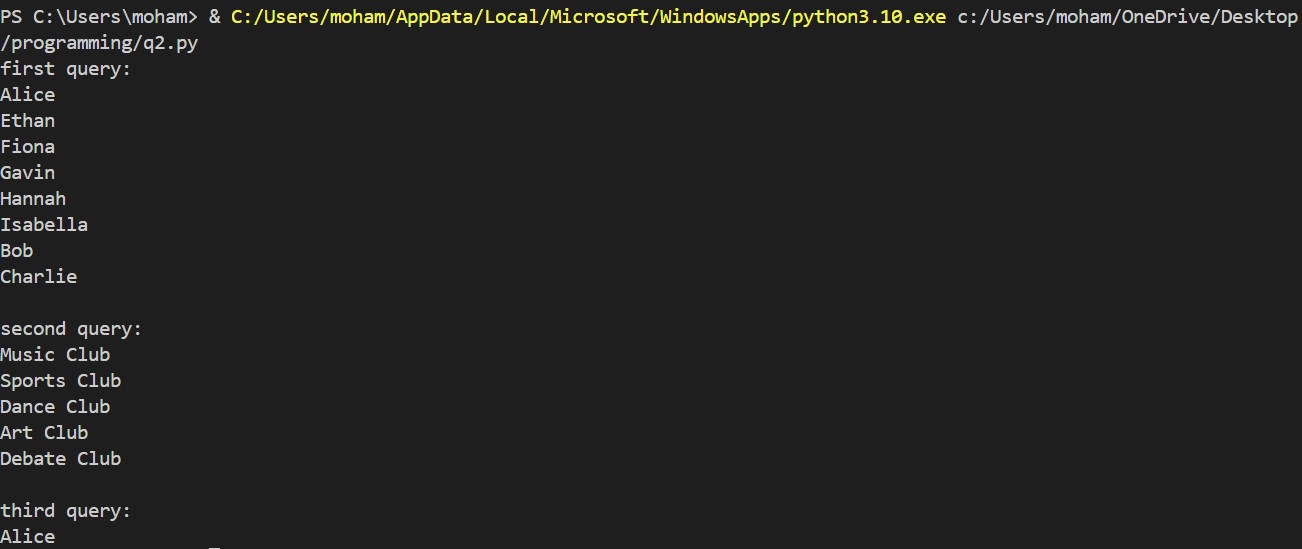
SELECT DISTI CT s.StudertName FRO,:.STUDEIH s

JOH. ErlFOLLf',EIH e ml s. FolHlo *=.* RolHI:;





## The output:



Q4: Students should implement the COMPANY database schema from Chapter 3 [1] and execute the solved queries of Chapter 7 [1].

**sec)**

**mysq > create ,ata ase Company; Query OK, 1 row affected (0.02**

**mysql> use Company;**

**Database changed**

**mysql> -- Create the company database if it doesn't exist mysql> CREATE DATABASE IF NOT EXISTS company;**

**Query OH, 1 row affected, 1 warning (0.00 sec)**

**mysql> USE company; Database changed mysql>**

**mysql> -- Create the DEPARTMENT table**

**mysql> CREATE TABLE IF NOT EXISTS DEPARTMENT** (

-> **Number INT PRIMARY HEY,**

-> **Name VARCHAR(255) NOT NULL,**

-> **Locations VARCHAR(255),**

-> **Manager\_start\_date DATE**

->

);

**Query OH, 0 rows affected (0.0ij sec)**

mysql> -- Create the PROJECT table

mysql> CREATE TABLE IF NOT EXISTS PROJECT (

-> Number INT PRIMARY KEY,

-> Name VARCHAR(255) NOT NULL,

-> Location VARCHAR(255),

-> Controlling\_department INT,

-> CONSTRAINT fk\_department FOREIGN KEY (Controlling\_department) REFERENCES DEPARTMENT(Number)

-> );

Query OK, 0 rows affected (0.07 sec)

|  |
| --- |
| -> |
| -> |
| ->  ->  ->  ->  ->  ->  ->  -> |
| -> |

mysql> -- Create the DEPENDENT table

mysql> -- Create the EMPLOYEE table

mysql> CREATE TABLE IF NOT EXISTS EMPLOYEE ( Ssn INT PRIMARY KEY,

Fname VARCHAR(255) NOT NULL,

Minit CHAR(l),

Lname VARCHAR(255) NOT NULL,

Birth\_date DATE, Sex CHAR(l),

Address VARCHAR(255), Supervisor INT,

Salary DECIMAL(10, 2),

Works\_on INT, Department INT,

->

->

->

->

->

->

->

Manager INT, Project INT,

Hours DECIMAL(10, 2),

CONSTRAINT fk\_supervisor FOREIGN KEY (Supervisor) REFERENCES EMPLOYEE(Ssn), CONSTRAINT fk\_department\_emp FOREIGN KEY (Department) REFERENCES DEPARTMENT(Number), CONSTRAINT fk\_manager FOREIGN KEY (Manager) REFERENCES EMPLOYEE(Ssn),

CONSTRAINT fk\_project FOREIGN KEY (Project) REFERENCES PROJECT(Number)

-> );

Query OK, 0 rows affected (0.07 sec)

mysql> CREATE TABLE IF NOT EXISTS DEPENDENT (

-> Dependent\_name VARCHAR(255) NOT NULL,

-> Relationship VARCHAR(255),

-> Birth\_date DATE,

-> Sex CHAR(l),

-> Address VARCHAR(255),

-> Employee INT,

-> CONSTRAINT fk\_emp\_dependent FOREIGN KEY (Employee) REFERENCES EMPLOYEE(Ssn)

-> );

Query OK, 0 rows affected (0.06 sec)

**m**

|  |  |
| --- | --- |
| **ysql> show tables;**  - + | |
| **I Tables\_in\_company** | **I** |
| - +  **department dependent employee project**  - +  **q rows in set (0.00 sec** | |

+

+

+

**)**

de

-> **Ac**

mysql> describe deparment;

ERROR llij6 (ij2S02): Table 1company.deparment1 doesn't exist mysql> describe department;

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

mysql> scribe

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

I Field I Type I Null I ey I Default I Extra I

Number Name Locations

int NO

varchar(255) NO varchar(255) YES

PRI NULL NULL NULL

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

ij rows in set (0.00 sec)

Manager\_start\_date date YES NULL

mysql> describe dependent;

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

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

I Field I Type I Null I ey I Default I Extra I

Dependent\_name varchar(255) NO Relationship varchar(255) YES Birth\_date date YES

Sex char(l) YES

Address varchar(255) YES

NULL NULL NULL NULL NULL

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

6 rows in set (0.00 sec)

Employee int YES MUL NULL

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

mysql> describe employee;

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

I Field

I Type

I Null I ey I Default I Extra I

Ssn int

Fname varchar(255)

Minit char(l)

Lname varchar(255) Birth date date

Sex char(l)

Address varchar(255) Supervisor int

Salary decimal(l0,2)

Works\_on int Department int Manager int

Project int

NO NO YES NO YES YES YES YES YES YES YES YES YES

PRI

MUL

MUL MUL MUL

NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL

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

1q rows in set (0.00 sec)

Hours decimal(l0,2) YES NULL

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

mysql> describe project;

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

I Field

I Type

I Null I ey I Default I Extra I

Number Name

int varchar(255)

Location varchar(255)

NO NO YES

PRI NULL NULL NULL

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

*q* rows in set (0.00 sec)

Controlling\_department int

YES

MUL NULL

**Super\_ssn IS NULL' at line 2 mysql> SELECT Fname, Lname**

-> **FROM EMPLOYEE**

-> **WHERE Super\_ssn IS NULL;**

mysql> SELECT DISTINCT Pnumber

-> FROM PROJECT

-> WHERE Pnumber IN

-> ( SELECT Pnumber

-> FROM PROJECT, DEPARTMENT, EMPLOYEE

-> WHERE Dnum = Dnumber AND

-> Mgr\_ssn = Ssn AND Lname = <Smith1 )

-> OR

-> Pnumber IN

-> ( SELECT Pno

-> FROM WOR S\_ON, EMPLOYEE

-> WHERE Essn = Ssn AND Lname - 'Smith1 )·

mysql> SELECT DISTINCT Essn

-> FROM WOR S\_ON

-> WHERE (Pno, Hours) IN ( SELECT Pno, Hours

-> FROM WOR S\_ON

-> WHERE Essn = '123ij567891 );

mysql> SELECT Lname, Fname

# -> FROM EMPLOYEE

-> WHERE Salary> ALL ( SELECT Salary

# -> FROM EMPLOYEE

-> WHERE Dno = 5)

mysql> SELECT StudentName

# -> FROM STUDENT

-> WHERE StudentName LI E 1A%1 •

*I*

+ +

I StudentName I

+ +

I Alice

+ +

1 row in set (0.00 sec)