

COMPONENT 1

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

STUDENT	<u>Roll No</u>	StudentName	Course	DOB
	Char(6)	Varchar(20)	Varchar(10)	Date

SOCIETY	<u>SocID</u>	SocName	MentorName	TotalSeats
	Char(6)	Varchar(20)	Varchar(15)	Unsigned int

ENROLLMENT	<u>Roll No</u>	<u>SID</u>	DateOfEnrollment
	Char(6)	Char(6)	Date

Here Rollno (ENROLLMENT) and SID (ENROLLMENT) are foreign keys.

MySQL

- SHOW DATABASES;

```
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)
```

- CREATE DATABASE student_society;
- SHOW DATABASES;

```
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| student_society |
| sys |
+-----+
5 rows in set (0.00 sec)
```

- USE student_society;
- CREATE TABLE STUDENT (
 - ➔ Roll_No CHAR(6),
 - ➔ StudentName VARCHAR(20) NOT NULL,
 - ➔ Course VARCHAR(10) NOT NULL,
 - ➔ DOB DATE NOT NULL,
 - ➔ PRIMARY KEY (Roll_No)
 - ➔);
- CREATE TABLE SOCIETY (
 - ➔ SocID CHAR(6),
 - ➔ SocName VARCHAR(20) NOT NULL,
 - ➔ MentorName VARCHAR(15) NOT NULL,
 - ➔ TotalSeats INT UNSIGNED NOT NULL,
 - ➔ PRIMARY KEY (SocID)
 - ➔);
- CREATE TABLE ENROLLMENT (
 - ➔ Roll_No CHAR(6),
 - ➔ SID CHAR(6),
 - ➔ DateOfEnrollment DATE,

- ➔ CONSTRAINT FK_Student FOREIGN KEY (Roll_No)
REFERENCES STUDENT(Roll_No),
- ➔ CONSTRAINT FK_Society FOREIGN KEY (SID)
REFERENCES SOCIETY(SocID)
- ➔);

- DESCRIBE STUDENT;

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Roll_No    | char(6)       | NO   | PRI | NULL    |       |
| StudentName | varchar(20)   | NO   |     | NULL    |       |
| Course     | varchar(10)   | NO   |     | NULL    |       |
| DOB        | date          | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.08 sec)

```

- DESCRIBE SOCIETY;

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| SocID      | char(6)       | NO   | PRI | NULL    |       |
| SocName    | varchar(20)   | NO   |     | NULL    |       |
| MentorName | varchar(15)   | NO   |     | NULL    |       |
| TotalSeats | int unsigned  | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

```

- DESCRIBE ENROLLMENT;

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Roll_No    | char(6)       | YES  | MUL | NULL    |       |
| SID        | char(6)       | YES  | MUL | NULL    |       |
| DateOfEnrollment | date        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

- INSERT INTO STUDENT VALUES ('YCS014','Adah Sharma','comp. sci.','2001-07-03'),('XCH002','Shiv Prakash','chemistry','2003-03-24'),('ZPY096','Kushal Verma','physics','2002-12-27'),('ZMT049','Yami Singh','maths','2004-11-14'),('XEL025','Ram Kumar','elec.','2002-06-10'),('ZEL066','Manoj Yadav','elec.','2004-01-04'),('YCS016','Isha Gupta','comp. sci.','2004-12-09'),('ZCS063','Rahul','comp. sci.','2001-01-04'),('YPY010','Amit Kumar','physics','2001-09-09'),('YCH094','Dinesh','chemistry','2001-05-04'),('XMT005','Rajeev','maths','2001-04-01'),('ZCS031','Anita Mishra','comp. sci.','2002-12-02'),('YMT072','Sumit','maths','2004-02-24'),('YCS100','Kriti','comp. sci.','2004-09-21'),('XPY019','Aman Singh','physics','2001-01-30'),('ZPY098','Yash Mittal','physics','2004-05-26'),('YCS093','Hemant Singh','comp. sci.','2003-05-29'),('YCS076','Trisha','comp. sci.','2003-10-18'),('ZEL064','Khushi','elec.','2002-01-31'),('YCS099','Deepak Kumar','comp. sci.','2004-10-13'),('ZMT069','Kabir','maths','2001-05-14'),('YEL068','Rivaba','elec.','2003-02-10'),('XCH052','Harsh Dagar','chemistry','2003-11-28'),('XCS046','Priya','comp. sci.','2003-02-14'),('YCS075','Avni','comp. sci.','2002-10-24'),('ZCH091','Shivam','chemistry','2003-12-22'),('YCS090','Arjun Chaudhary','comp. sci.','2002-07-15'),('ZPY044','Vikram Singh','physics','2003-03-07'),('XPY013','Sanjay Singh','physics','2004-09-26'),('XMT058','Abhishek','maths','2001-12-29');
- SELECT * FROM STUDENT;

Roll_No	StudentName	Course	DOB
XCH002	Shiv Prakash	chemistry	2003-03-24
XCH052	Harsh Dagar	chemistry	2003-11-28
XCS046	Priya	comp. sci.	2003-02-14
XEL025	Ram Kumar	elec.	2002-06-10
XMT005	Rajeev	maths	2001-04-01
XMT058	Abhishek	maths	2001-12-29
XPY013	Sanjay Singh	physics	2004-09-26
XPY019	Aman Singh	physics	2001-01-30
YCH094	Dinesh	chemistry	2001-05-04
YCS014	Adah Sharma	comp. sci.	2001-07-03
YCS016	Isha Gupta	comp. sci.	2004-12-09
YCS075	Avni	comp. sci.	2002-10-24
YCS076	Trisha	comp. sci.	2003-10-18
YCS090	Arjun Chaudhary	comp. sci.	2002-07-15
YCS093	Hemant Singh	comp. sci.	2003-05-29
YCS099	Deepak Kumar	comp. sci.	2004-10-13
YCS100	Kriti	comp. sci.	2004-09-21
YEL068	Rivaba	elec.	2003-02-10
YMT072	Sumit	maths	2004-02-24
YPY010	Amit Kumar	physics	2001-09-09
ZCH091	Shivam	chemistry	2003-12-22
ZCS031	Anita Mishra	comp. sci.	2002-12-02
ZCS063	Rahul	comp. sci.	2001-01-04
ZEL064	Khushi	elec.	2002-01-31
ZEL066	Manoj Yadav	elec.	2004-01-04
ZMT049	Yami Singh	maths	2004-11-14
ZMT069	Kabir	maths	2001-05-14
ZPY044	Vikram Singh	physics	2003-03-07
ZPY096	Kushal Verma	physics	2002-12-27
ZPY098	Yash Mittal	physics	2004-05-26

30 rows in set (0.01 sec)

- INSERT INTO SOCIETY VALUES ('s1','Sashakt','Dr. Anita',15),('s2','Debating','Dr. Yasheshwar',10),('s3','Music','Mr. Aman Gupta',20),('s4','Dancing','Mr. Abhimanyu',25),('s5','NSS','Mr. Narander',30);
- SELECT * FROM SOCIETY;

SocID	SocName	MentorName	TotalSeats
s1	Sashakt	Dr. Anita	15
s2	Debating	Dr. Yasheshwar	10
s3	Music	Mr. Aman Gupta	20
s4	Dancing	Mr. Abhimanyu	25
s5	NSS	Mr. Narander	30

5 rows in set (0.00 sec)

- SET foreign_key_checks=0; /*DISABLE FOREIGN KEY CHECKS*/
- INSERT INTO ENROLLMENT (Roll_No, SID) VALUES ('XCH002','s2'),('ZPY096','s2'),('ZMT049','s1'),('XEL025','s2'),('ZEL066','s4'),('YCS016','s4'),('ZCS063','s2'),('YPY010','s3'),('YCH094','s4'),('XMT005','s4'),('ZCS031','s1'),('ZCS031','s2'),('YMT072','s2'),('YCS100','s1'),('YCS100','s2'),('YCS100','s3'),('YCS100','s4'),('YCS100','s5'),('XPY019','s3'),('ZPY098','s3'),('YCS093','s2'),('ZEL064','s3'),('YCS099','s4'),('ZMT069','s1'),('ZMT069','s2'),('ZMT069','s3'),('ZMT069','s4'),('ZMT069','s5'),('YEL068','s1'),('XCH052','s4'),('XCS046','s3'),('YCS075','s1'),('ZCH091','s3'),('ZPY044','s1'),('XPY013','s2'),('XMT058','s3'),('XMT058','s5');
- UPDATE ENROLLMENT SET DateOfEnrollment='2019-01-15' WHERE SID='s1';
- UPDATE ENROLLMENT SET DateOfEnrollment='2024-01-22' WHERE SID='s2';
- UPDATE ENROLLMENT SET DateOfEnrollment='2019-01-02' WHERE SID='s3';
- UPDATE ENROLLMENT SET DateOfEnrollment='2022-02-05' WHERE SID='s4';
- UPDATE ENROLLMENT SET DateOfEnrollment='2023-03-12' WHERE SID='s5';
- SELECT * FROM ENROLLMENT;

```

+-----+-----+-----+
| Roll_No | SID   | DateOfEnrollment |
+-----+-----+-----+
| XCH002   | s2    | 2024-01-22        |
| ZPY096   | s2    | 2024-01-22        |
| ZMT049   | s1    | 2019-01-15        |
| XEL025   | s2    | 2024-01-22        |
| ZEL066   | s4    | 2022-02-05        |
| YCS016   | s4    | 2022-02-05        |
| ZCS063   | s2    | 2024-01-22        |
| YPY010   | s3    | 2019-01-02        |
| YCH094   | s4    | 2022-02-05        |
| XMT005   | s4    | 2022-02-05        |
| ZCS031   | s1    | 2019-01-15        |
| ZCS031   | s2    | 2024-01-22        |
| YMT072   | s2    | 2024-01-22        |
| YCS100   | s1    | 2019-01-15        |
| YCS100   | s2    | 2024-01-22        |
| YCS100   | s3    | 2019-01-02        |
| YCS100   | s4    | 2022-02-05        |
| YCS100   | s5    | 2023-03-12        |
| XPY019   | s3    | 2019-01-02        |
| ZPY098   | s3    | 2019-01-02        |
| YCS093   | s2    | 2024-01-22        |
| ZEL064   | s3    | 2019-01-02        |
| YCS099   | s4    | 2022-02-05        |
| ZMT069   | s1    | 2019-01-15        |
| ZMT069   | s2    | 2024-01-22        |
| ZMT069   | s3    | 2019-01-02        |
| ZMT069   | s4    | 2022-02-05        |
| ZMT069   | s5    | 2023-03-12        |
| YEL068   | s1    | 2019-01-15        |
| XCH052   | s4    | 2022-02-05        |
| XCS046   | s3    | 2019-01-02        |
| YCS075   | s1    | 2019-01-15        |
| ZCH091   | s3    | 2019-01-02        |
| ZPY044   | s1    | 2019-01-15        |
| XPY013   | s2    | 2024-01-22        |
| XMT058   | s3    | 2019-01-02        |
| XMT058   | s5    | 2023-03-12        |
+-----+-----+-----+
37 rows in set (0.00 sec)

```

- SET foreign_key_checks=1; /*ENABLE FOREIGN KEY CHECKS*/

QUERIES

1. Retrieve names of students enrolled in any society.

- SELECT DISTINCT STUDENT.Roll_No AS 'ROLL NUMBERS',
STUDENT.StudentName AS 'NAMES'
FROM STUDENT
INNER JOIN ENROLLMENT
ON STUDENT.Roll_No=ENROLLMENT.Roll_No
ORDER BY STUDENT.StudentName;

ROLL NUMBERS	NAMES
XMT058	Abhishek
XPY019	Aman Singh
YPY010	Amit Kumar
ZCS031	Anita Mishra
YCS075	Avni
YCS099	Deepak Kumar
YCH094	Dinesh
XCH052	Harsh Dagar
YCS093	Hemant Singh
YCS016	Isha Gupta
ZMT069	Kabir
ZEL064	Khushi
YCS100	Kriti
ZPY096	Kushal Verma
ZEL066	Manoj Yadav
XCS046	Priya
ZCS063	Rahul
XMT005	Rajeev
XEL025	Ram Kumar
YEL068	Rivaba
XPY013	Sanjay Singh
XCH002	Shiv Prakash
ZCH091	Shivam
YMT072	Sumit
ZPY044	Vikram Singh
ZMT049	Yami Singh
ZPY098	Yash Mittal

27 rows in set (0.00 sec)

OR

- SELECT STUDENT.Roll_No, STUDENT.StudentName
FROM STUDENT
INNER JOIN ENROLLMENT
ON STUDENT.Roll_No=ENROLLMENT.Roll_No
ORDER BY STUDENT.StudentName;

Roll_No	StudentName
XMT058	Abhishek
XMT058	Abhishek
XPY019	Aman Singh
YPY010	Amit Kumar
ZCS031	Anita Mishra
ZCS031	Anita Mishra
YCS075	Avni
YCS099	Deepak Kumar
YCH094	Dinesh
XCH052	Harsh Dagar
YCS093	Hemant Singh
YCS016	Isha Gupta
ZMT069	Kabir
ZMT069	Kabir
ZMT069	Kabir
ZMT069	Kabir
ZMT069	Kabir
ZEL064	Khushi
YCS100	Kriti
YCS100	Kriti
YCS100	Kriti
YCS100	Kriti
YCS100	Kriti
ZPY096	Kushal Verma
ZEL066	Manoj Yadav
XCS046	Priya
ZCS063	Rahul
XMT005	Rajeev
XEL025	Ram Kumar
YEL068	Rivaba
XPY013	Sanjay Singh
XCH002	Shiv Prakash
ZCH091	Shivam
YMT072	Sumit
ZPY044	Vikram Singh
ZMT049	Yami Singh
ZPY098	Yash Mittal

37 rows in set (0.00 sec)

2. Retrieve all society names.

- `SELECT SocName AS 'SOCIETY NAMES'`
`FROM SOCIETY`
`ORDER BY SocName;`

```
+-----+
| SOCIETY NAMES |
+-----+
| Dancing       |
| Debating      |
| Music         |
| NSS           |
| Sashakt       |
+-----+
5 rows in set (0.00 sec)
```

3. Retrieve students' names starting with letter 'A'.

- `SELECT StudentName AS "STUDENTS' NAMES STARTING WITH 'A'"`
`FROM STUDENT`
`WHERE StudentName LIKE 'a%';`

```
+-----+
| STUDENTS' NAMES STARTING WITH 'A' |
+-----+
| Abhishek      |
| Aman Singh    |
| Adah Sharma   |
| Avni          |
| Arjun Chaudhary |
| Amit Kumar    |
| Anita Mishra  |
+-----+
7 rows in set (0.00 sec)
```

4. Retrieve students' details studying in courses 'computer science' or 'chemistry'.

- SELECT * FROM STUDENT
WHERE Course='comp. sci.' OR Course='chemistry'
ORDER BY StudentName;

Roll_No	StudentName	Course	DOB
YCS014	Adah Sharma	comp. sci.	2001-07-03
ZCS031	Anita Mishra	comp. sci.	2002-12-02
YCS090	Arjun Chaudhary	comp. sci.	2002-07-15
YCS075	Avni	comp. sci.	2002-10-24
YCS099	Deepak Kumar	comp. sci.	2004-10-13
YCH094	Dinesh	chemistry	2001-05-04
XCH052	Harsh Dagar	chemistry	2003-11-28
YCS093	Hemant Singh	comp. sci.	2003-05-29
YCS016	Isha Gupta	comp. sci.	2004-12-09
YCS100	Kriti	comp. sci.	2004-09-21
XCS046	Priya	comp. sci.	2003-02-14
ZCS063	Rahul	comp. sci.	2001-01-04
XCH002	Shiv Prakash	chemistry	2003-03-24
ZCH091	Shivam	chemistry	2003-12-22
YCS076	Trisha	comp. sci.	2003-10-18

15 rows in set (0.00 sec)

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

- SELECT StudentName, Roll_No
FROM STUDENT
WHERE Roll_No LIKE 'x%9' OR Roll_No LIKE 'z%9';

StudentName	Roll_No
Aman Singh	XPY019
Yami Singh	ZMT049
Kabir	ZMT069

3 rows in set (0.00 sec)

OR

- SELECT StudentName
FROM STUDENT
WHERE Roll_No LIKE 'x%9' OR Roll_No LIKE 'z%9'
ORDER BY StudentName;

```
+-----+
| StudentName |
+-----+
| Aman Singh  |
| Kabir       |
| Yami Singh  |
+-----+
3 rows in set (0.00 sec)
```

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

- set @N:=15;
- SELECT *
FROM SOCIETY
WHERE TotalSeats>@N;

```
+-----+-----+-----+-----+
| SocID | SocName | MentorName | TotalSeats |
+-----+-----+-----+-----+
| s3    | Music   | Mr. Aman Gupta | 20 |
| s4    | Dancing | Mr. Abhimanyu  | 25 |
| s5    | NSS     | Mr. Narander   | 30 |
+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

7. Update society table for mentor name of a specific society

- UPDATE SOCIETY
SET MentorName='Dr. Yash'
WHERE SocID='s2';
- SELECT * FROM SOCIETY;

SocID	SocName	MentorName	TotalSeats
s1	Sashakt	Dr. Anita	15
s2	Debating	Dr. Yash	10
s3	Music	Mr. Aman Gupta	20
s4	Dancing	Mr. Abhimanyu	25
s5	NSS	Mr. Narander	30

5 rows in set (0.00 sec)

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

- SELECT s.SocName, COUNT(e.SID) AS TOTAL
FROM SOCIETY AS s
INNER JOIN ENROLLMENT AS e
ON s.SocID=e.SID
GROUP BY s.SocName
HAVING COUNT(e.SID)>5
ORDER BY s.SocName;

SocName	TOTAL
Dancing	8
Debating	10
Music	9
Sashakt	7

4 rows in set (0.01 sec)

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

- ```

SELECT STUDENT.Roll_No, STUDENT.StudentName,
TIMESTAMPDIFF(YEAR, STUDENT.DOB, CURDATE()) AS AGE
FROM STUDENT
INNER JOIN ENROLLMENT
ON STUDENT.Roll_No=ENROLLMENT.Roll_No
INNER JOIN SOCIETY
ON ENROLLMENT.SID=SOCIETY.SocID
WHERE SOCIETY.SocName='NSS' AND TIMESTAMPDIFF(YEAR,
STUDENT.DOB, CURDATE())=(SELECT MIN(TIMESTAMPDIFF(YEAR,
STUDENT.DOB, CURDATE())) FROM STUDENT);

```

```

+-----+-----+-----+
| Roll_No | StudentName | AGE |
+-----+-----+-----+
| YCS100 | Kriti | 19 |
+-----+-----+-----+
1 row in set (0.00 sec)

```

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

- ```

SELECT SOCIETY.SocName, COUNT(ENROLLMENT.SID) AS 'TOTAL
STUDENTS'
FROM SOCIETY
INNER JOIN ENROLLMENT
ON SOCIETY.SocID=ENROLLMENT.SID
GROUP BY SOCIETY.SocName
HAVING COUNT(ENROLLMENT.SID)=(SELECT MAX(counts) FROM (SELECT
SOCIETY.SocName, COUNT(ENROLLMENT.SID) AS counts FROM SOCIETY
INNER JOIN ENROLLMENT ON SOCIETY.SocID=ENROLLMENT.SID GROUP
BY SOCIETY.SocName) as temp);

```

```

+-----+-----+
| SocName | TOTAL STUDENTS |
+-----+-----+
| Debating |          10 |
+-----+-----+
1 row in set (0.00 sec)

```

OR

- SELECT SOCIETY.SocName, COUNT(ENROLLMENT.SID) AS 'TOTAL STUDENTS'
FROM SOCIETY
INNER JOIN ENROLLMENT
ON SOCIETY.SocID=ENROLLMENT.SID
GROUP BY SOCIETY.SocName
ORDER BY COUNT(ENROLLMENT.SID) DESC;

```

+-----+-----+
| SocName | TOTAL STUDENTS |
+-----+-----+
| Debating |          10 |
| Music    |           9 |
| Dancing  |           8 |
| Sashakt  |           7 |
| NSS      |           3 |
+-----+-----+
5 rows in set (0.00 sec)

```

- SELECT SOCIETY.SocName, COUNT(ENROLLMENT.SID) AS 'TOTAL STUDENTS'
FROM SOCIETY
INNER JOIN ENROLLMENT
ON SOCIETY.SocID=ENROLLMENT.SID
GROUP BY SOCIETY.SocName
ORDER BY COUNT(ENROLLMENT.SID) DESC LIMIT 1;


```

+-----+-----+
| SocName | TOTAL STUDENTS |
+-----+-----+
| Debating |          10 |
+-----+-----+
1 row in set (0.00 sec)

```

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

- ```

SELECT SOCIETY.SocID, SOCIETY.SocName, COUNT(ENROLLMENT.SID)
AS 'ENROLLED STUDENTS'
FROM SOCIETY
INNER JOIN ENROLLMENT
ON SOCIETY.SocID=ENROLLMENT.SID
GROUP BY SOCIETY.SocID, SOCIETY.SocName
ORDER BY COUNT(ENROLLMENT.SID) ASC LIMIT 2;

```

```

+-----+-----+-----+
| SocID | SocName | ENROLLED STUDENTS |
+-----+-----+-----+
| s5 | NSS | 3 |
| s1 | Sashakt | 7 |
+-----+-----+-----+
2 rows in set (0.00 sec)

```

**12. Find the student names who are not enrolled in any society**

- ```

SELECT STUDENT.Roll_No, STUDENT.StudentName
FROM STUDENT
LEFT JOIN ENROLLMENT
ON STUDENT.Roll_No=ENROLLMENT.Roll_No
WHERE ENROLLMENT.Roll_No IS NULL;

```

```

+-----+-----+
| Roll_No | StudentName |
+-----+-----+
| YCS014  | Adah Sharma |
| YCS076  | Trisha      |
| YCS090  | Arjun Chaudhary |
+-----+-----+
3 rows in set (0.00 sec)

```

13. Find the student names enrolled in at least two societies

- SELECT STUDENT.Roll_No, STUDENT.StudentName
 FROM STUDENT
 INNER JOIN ENROLLMENT
 ON STUDENT.Roll_No=ENROLLMENT.Roll_No
 GROUP BY STUDENT.Roll_No, STUDENT.StudentName
 HAVING COUNT(ENROLLMENT.SID)>=2;

```

+-----+-----+
| Roll_No | StudentName |
+-----+-----+
| ZCS031  | Anita Mishra |
| YCS100  | Kriti       |
| ZMT069  | Kabir       |
| XMT058  | Abhishek    |
+-----+-----+
4 rows in set (0.00 sec)

```

14. Find society names in which maximum students are enrolled

- SELECT SOCIETY.SocName, COUNT(ENROLLMENT.SID) AS 'TOTAL STUDENTS'
 FROM SOCIETY
 INNER JOIN ENROLLMENT
 ON SOCIETY.SocID=ENROLLMENT.SID
 GROUP BY SOCIETY.SocName
 HAVING COUNT(ENROLLMENT.SID)=(SELECT MAX(counts) FROM (SELECT SOCIETY.SocName, COUNT(ENROLLMENT.SID) AS counts FROM SOCIETY INNER JOIN ENROLLMENT ON SOCIETY.SocID=ENROLLMENT.SID GROUP BY SOCIETY.SocName) as temp);

```

+-----+-----+
| SocName | TOTAL STUDENTS |
+-----+-----+
| Debating |          10 |
+-----+-----+
1 row in set (0.00 sec)

```

OR

- SELECT SOCIETY.SocName, COUNT(ENROLLMENT.SID) AS 'TOTAL STUDENTS'
 FROM SOCIETY
 INNER JOIN ENROLLMENT
 ON SOCIETY.SocID=ENROLLMENT.SID
 GROUP BY SOCIETY.SocName
 ORDER BY COUNT(ENROLLMENT.SID) DESC LIMIT 1;

```

+-----+-----+
| SocName | TOTAL STUDENTS |
+-----+-----+
| Debating |          10 |
+-----+-----+
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

- SELECT STUDENT.Roll_No, STUDENT.StudentName, SOCIETY.SocName
FROM STUDENT
INNER JOIN ENROLLMENT
ON STUDENT.Roll_No=ENROLLMENT.Roll_No
INNER JOIN SOCIETY
ON SOCIETY.SocID=ENROLLMENT.SID
GROUP BY STUDENT.Roll_No, STUDENT.StudentName, SOCIETY.SocName
HAVING COUNT(ENROLLMENT.SID)>=1
ORDER BY SOCIETY.SocName;

Roll_No	StudentName	SocName
XMT005	Rajeev	Dancing
YCH094	Dinesh	Dancing
XCH052	Harsh Dagar	Dancing
ZMT069	Kabir	Dancing
YCS099	Deepak Kumar	Dancing
YCS100	Kriti	Dancing
YCS016	Isha Gupta	Dancing
ZEL066	Manoj Yadav	Dancing
ZMT069	Kabir	Debating
XPY013	Sanjay Singh	Debating
YCS093	Hemant Singh	Debating
YCS100	Kriti	Debating
ZCS031	Anita Mishra	Debating
ZCS063	Rahul	Debating
XCH002	Shiv Prakash	Debating
XEL025	Ram Kumar	Debating
ZPY096	Kushal Verma	Debating
YMT072	Sumit	Debating
ZEL064	Khushi	Music
XMT058	Abhishek	Music
ZCH091	Shivam	Music
XCS046	Priya	Music
ZMT069	Kabir	Music
ZPY098	Yash Mittal	Music
XPY019	Aman Singh	Music
YCS100	Kriti	Music
YPY010	Amit Kumar	Music
XMT058	Abhishek	NSS
YCS100	Kriti	NSS
ZMT069	Kabir	NSS
ZPY044	Vikram Singh	Sashakt
YCS075	Avni	Sashakt
YEL068	Rivaba	Sashakt
ZMT069	Kabir	Sashakt
YCS100	Kriti	Sashakt
ZCS031	Anita Mishra	Sashakt
ZMT049	Yami Singh	Sashakt

37 rows in set (0.00 sec)

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

- SELECT STUDENT.Roll_No, STUDENT.StudentName
FROM STUDENT
INNER JOIN ENROLLMENT
ON STUDENT.Roll_No=ENROLLMENT.Roll_No
INNER JOIN SOCIETY
ON ENROLLMENT.SID=SOCIETY.SocID
WHERE SOCIETY.SocName IN ('Debating', 'Dancing', 'Sashakt');

Roll_No	StudentName
ZMT049	Yami Singh
ZCS031	Anita Mishra
YCS100	Kriti
ZMT069	Kabir
YEL068	Rivaba
YCS075	Avni
ZPY044	Vikram Singh
XCH002	Shiv Prakash
ZPY096	Kushal Verma
XEL025	Ram Kumar
ZCS063	Rahul
ZCS031	Anita Mishra
YMT072	Sumit
YCS100	Kriti
YCS093	Hemant Singh
ZMT069	Kabir
XPY013	Sanjay Singh
ZEL066	Manoj Yadav
YCS016	Isha Gupta
YCH094	Dinesh
XMT005	Rajeev
YCS100	Kriti
YCS099	Deepak Kumar
ZMT069	Kabir
XCH052	Harsh Dagar

25 rows in set (0.01 sec)

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

- SELECT SocName
FROM SOCIETY
WHERE MentorName LIKE '%Gupta%';

```
+-----+
| SocName |
+-----+
| Music   |
+-----+
1 row in set (0.01 sec)
```

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

- SELECT SOCIETY.SocName, SOCIETY.TotalSeats,
COUNT(ENROLLMENT.SID) AS 'Total Students'
FROM SOCIETY
INNER JOIN ENROLLMENT
ON SOCIETY.SocID=ENROLLMENT.SID
GROUP BY SOCIETY.SocName, SOCIETY.TotalSeats
HAVING COUNT('Total Students')=0.1*SOCIETY.TotalSeats;

```
+-----+-----+-----+
| SocName | TotalSeats | Total Students |
+-----+-----+-----+
| NSS     |          30 |                3 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

19. Display the vacant seats for each society.

- SELECT SOCIETY.SocName, (SOCIETY.TotalSeats-COUNT(ENROLLMENT.SID)) AS 'Vacant Seats'
FROM SOCIETY
LEFT JOIN ENROLLMENT
ON SOCIETY.SocID=ENROLLMENT.SID
GROUP BY SOCIETY.SocName, SOCIETY.TotalSeats;

SocName	Vacant Seats
Sashakt	8
Debating	0
Music	11
Dancing	17
NSS	27

5 rows in set (0.00 sec)

20. Increment Total Seats of each society by 10%

- SELECT * FROM SOCIETY;

SocID	SocName	MentorName	TotalSeats
s1	Sashakt	Dr. Anita	15
s2	Debating	Dr. Yash	10
s3	Music	Mr. Aman Gupta	20
s4	Dancing	Mr. Abhimanyu	25
s5	NSS	Mr. Narander	30

5 rows in set (0.00 sec)

- UPDATE SOCIETY
SET TotalSeats=1.1*TotalSeats;

```
Query OK, 5 rows affected (0.12 sec)
Rows matched: 5  Changed: 5  Warnings: 0
```

- SELECT * FROM SOCIETY;

```
+-----+-----+-----+-----+
| SocID | SocName | MentorName | TotalSeats |
+-----+-----+-----+-----+
| s1    | Sashakt | Dr. Anita  | 17         |
| s2    | Debating | Dr. Yash   | 11         |
| s3    | Music   | Mr. Aman Gupta | 22         |
| s4    | Dancing | Mr. Abhimanyu | 28         |
| s5    | NSS     | Mr. Narander | 33         |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

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

- ALTER TABLE ENROLLMENT
ADD Fees_Paid ENUM('yes','No');

```
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

- UPDATE ENROLLMENT
SET Fees_Paid='yes'
WHERE SID='s1';

- UPDATE ENROLLMENT
SET Fees_Paid='yes'
WHERE SID='s2';
- UPDATE ENROLLMENT
SET Fees_Paid='yes'
WHERE SID='s3';
- UPDATE ENROLLMENT
SET Fees_Paid='yes'
WHERE SID='s4';
- UPDATE ENROLLMENT
SET Fees_Paid='No'
WHERE SID='s5';
- SELECT * FROM ENROLLMENT;

```
mysql> SELECT * FROM ENROLLMENT;
```

Roll_No	SID	DateOfEnrollment	Fees_Paid
XCH002	s2	2024-01-22	yes
ZPY096	s2	2024-01-22	yes
ZMT049	s1	2019-01-15	yes
XEL025	s2	2024-01-22	yes
ZEL066	s4	2022-02-05	yes
YCS016	s4	2022-02-05	yes
ZCS063	s2	2024-01-22	yes
YPY010	s3	2019-01-02	yes
YCH094	s4	2022-02-05	yes
XMT005	s4	2022-02-05	yes
ZCS031	s1	2019-01-15	yes
ZCS031	s2	2024-01-22	yes
YMT072	s2	2024-01-22	yes
YCS100	s1	2019-01-15	yes
YCS100	s2	2024-01-22	yes
YCS100	s3	2019-01-02	yes
YCS100	s4	2022-02-05	yes
YCS100	s5	2023-03-12	No
XPY019	s3	2019-01-02	yes
ZPY098	s3	2019-01-02	yes
YCS093	s2	2024-01-22	yes
ZEL064	s3	2019-01-02	yes
YCS099	s4	2022-02-05	yes
ZMT069	s1	2019-01-15	yes
ZMT069	s2	2024-01-22	yes
ZMT069	s3	2019-01-02	yes
ZMT069	s4	2022-02-05	yes
ZMT069	s5	2023-03-12	No
YEL068	s1	2019-01-15	yes
XCH052	s4	2022-02-05	yes
XCS046	s3	2019-01-02	yes
YCS075	s1	2019-01-15	yes
ZCH091	s3	2019-01-02	yes
ZPY044	s1	2019-01-15	yes
XPY013	s2	2024-01-22	yes
XMT058	s3	2019-01-02	yes
XMT058	s5	2023-03-12	No

```
37 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'.

- UPDATE ENROLLMENT
SET DateOfEnrollment='2018-01-15'
WHERE SID='s1';
- UPDATE ENROLLMENT
SET DateOfEnrollment=CURDATE()
WHERE SID='s2';
- UPDATE ENROLLMENT
SET DateOfEnrollment='2018-01-02'
WHERE SID='s3';
- SELECT * FROM ENROLLMENT;

```
mysql> SELECT * FROM ENROLLMENT;
```

Roll_No	SID	DateOfEnrollment	Fees_Paid
XCH002	s2	2024-04-24	yes
ZPY096	s2	2024-04-24	yes
ZMT049	s1	2018-01-15	yes
XEL025	s2	2024-04-24	yes
ZEL066	s4	2022-02-05	yes
YCS016	s4	2022-02-05	yes
ZCS063	s2	2024-04-24	yes
YPY010	s3	2018-01-02	yes
YCH094	s4	2022-02-05	yes
XMT005	s4	2022-02-05	yes
ZCS031	s1	2018-01-15	yes
ZCS031	s2	2024-04-24	yes
YMT072	s2	2024-04-24	yes
YCS100	s1	2018-01-15	yes
YCS100	s2	2024-04-24	yes
YCS100	s3	2018-01-02	yes
YCS100	s4	2022-02-05	yes
YCS100	s5	2023-03-12	No
XPY019	s3	2018-01-02	yes
ZPY098	s3	2018-01-02	yes
YCS093	s2	2024-04-24	yes
ZEL064	s3	2018-01-02	yes
YCS099	s4	2022-02-05	yes
ZMT069	s1	2018-01-15	yes
ZMT069	s2	2024-04-24	yes
ZMT069	s3	2018-01-02	yes
ZMT069	s4	2022-02-05	yes
ZMT069	s5	2023-03-12	No
YEL068	s1	2018-01-15	yes
XCH052	s4	2022-02-05	yes
XCS046	s3	2018-01-02	yes
YCS075	s1	2018-01-15	yes
ZCH091	s3	2018-01-02	yes
ZPY044	s1	2018-01-15	yes
XPY013	s2	2024-04-24	yes
XMT058	s3	2018-01-02	yes
XMT058	s5	2023-03-12	No

```
37 rows in set (0.00 sec)
```

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

- CREATE VIEW society_names AS
SELECT SOCIETY.SocName, COUNT(ENROLLMENT.SID) AS 'Total Students'
FROM SOCIETY
INNER JOIN ENROLLMENT
ON SOCIETY.SocID=ENROLLMENT.SID
GROUP BY SOCIETY.SocName;
- SELECT * FROM society_names;

```
+-----+-----+
| SocName | Total Students |
+-----+-----+
| Sashakt |          7 |
| Debating |         10 |
| Music |          9 |
| Dancing |          8 |
| NSS |          3 |
+-----+-----+
5 rows in set (0.07 sec)
```

24. Find student names enrolled in all the societies.

- SELECT STUDENT.Roll_No, STUDENT.StudentName
FROM STUDENT
INNER JOIN ENROLLMENT
ON STUDENT.Roll_No=ENROLLMENT.Roll_No
GROUP BY STUDENT.Roll_No, STUDENT.StudentName
HAVING COUNT(ENROLLMENT.Roll_No)=(SELECT COUNT(*) FROM SOCIETY);

```
+-----+-----+
| Roll_No | StudentName |
+-----+-----+
| YCS100 | Kriti |
| ZMT069 | Kabir |
+-----+-----+
2 rows in set (0.08 sec)
```

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

- SELECT COUNT(*) AS Total
FROM (SELECT SOCIETY.SocName
FROM SOCIETY
INNER JOIN ENROLLMENT
ON SOCIETY.SocID=ENROLLMENT.SID
GROUP BY SOCIETY.SocName
HAVING COUNT(ENROLLMENT.SID)>5) AS TEMP;

```
+-----+  
| Total |  
+-----+  
|      4 |  
+-----+  
1 row in set (0.00 sec)
```

26. Add column Mobile number in student table with default value '9999999999'

- ALTER TABLE STUDENT
ADD Mobile_number varchar(20) DEFAULT '9999999999';

```
Query OK, 0 rows affected (0.57 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

- SELECT * FROM STUDENT;

Roll_No	StudentName	Course	DOB	Mobile_number
XCH002	Shiv Prakash	chemistry	2003-03-24	9999999999
XCH052	Harsh Dagar	chemistry	2003-11-28	9999999999
XCS046	Priya	comp. sci.	2003-02-14	9999999999
XEL025	Ram Kumar	elec.	2002-06-10	9999999999
XMT005	Rajeev	maths	2001-04-01	9999999999
XMT058	Abhishek	maths	2001-12-29	9999999999
XPY013	Sanjay Singh	physics	2004-09-26	9999999999
XPY019	Aman Singh	physics	2001-01-30	9999999999
YCH094	Dinesh	chemistry	2001-05-04	9999999999
YCS014	Adah Sharma	comp. sci.	2001-07-03	9999999999
YCS016	Isha Gupta	comp. sci.	2004-12-09	9999999999
YCS075	Avni	comp. sci.	2002-10-24	9999999999
YCS076	Trisha	comp. sci.	2003-10-18	9999999999
YCS090	Arjun Chaudhary	comp. sci.	2002-07-15	9999999999
YCS093	Hemant Singh	comp. sci.	2003-05-29	9999999999
YCS099	Deepak Kumar	comp. sci.	2004-10-13	9999999999
YCS100	Kriti	comp. sci.	2004-09-21	9999999999
YEL068	Rivaba	elec.	2003-02-10	9999999999
YMT072	Sumit	maths	2004-02-24	9999999999
YPY010	Amit Kumar	physics	2001-09-09	9999999999
ZCH091	Shivam	chemistry	2003-12-22	9999999999
ZCS031	Anita Mishra	comp. sci.	2002-12-02	9999999999
ZCS063	Rahul	comp. sci.	2001-01-04	9999999999
ZEL064	Khushi	elec.	2002-01-31	9999999999
ZEL066	Manoj Yadav	elec.	2004-01-04	9999999999
ZMT049	Yami Singh	maths	2004-11-14	9999999999
ZMT069	Kabir	maths	2001-05-14	9999999999
ZPY044	Vikram Singh	physics	2003-03-07	9999999999
ZPY096	Kushal Verma	physics	2002-12-27	9999999999
ZPY098	Yash Mittal	physics	2004-05-26	9999999999

30 rows in set (0.00 sec)

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

- SELECT COUNT(*) AS 'TOTAL STUDENTS'
FROM (SELECT Roll_No, StudentName
FROM STUDENT
WHERE TIMESTAMPDIFF(YEAR, DOB, CURDATE()) > 20) AS Temp;

```

+-----+
| TOTAL STUDENTS |
+-----+
|              18 |
+-----+
1 row in set (0.00 sec)

```

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

- `SELECT Roll_No, StudentName
FROM STUDENT
WHERE EXISTS (SELECT *
FROM ENROLLMENT
WHERE STUDENT.Roll_No=ENROLLMENT.Roll_No) AND DOB LIKE
'%2001%';`

```

+-----+-----+
| Roll_No | StudentName |
+-----+-----+
| XMT005  | Rajeev      |
| XMT058  | Abhishek    |
| XPY019  | Aman Singh  |
| YCH094  | Dinesh      |
| YPY010  | Amit Kumar  |
| ZCS063  | Rahul       |
| ZMT069  | Kabir       |
+-----+-----+
7 rows in set (0.00 sec)

```


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

- SELECT COUNT(*) AS 'TOTAL SOCIETIES'
FROM (SELECT SOCIETY.SocName, COUNT(ENROLLMENT.SID)
FROM SOCIETY
INNER JOIN ENROLLMENT
ON SOCIETY.SocID=ENROLLMENT.SID
WHERE SOCIETY.SocName LIKE 's%t'
GROUP BY SOCIETY.SocName
HAVING COUNT(ENROLLMENT.SID)>=5) AS TEMP;

```
+-----+
| TOTAL SOCIETIES |
+-----+
|                1 |
+-----+
1 row in set (0.00 sec)
```

30. Display the following information:

Society name Mentor name Total Capacity Total Enrolled Unfilled Seats

- SELECT SOCIETY.SocName AS 'Society name', SOCIETY.MentorName AS 'Mentor name', SOCIETY.TotalSeats AS 'Total Capacity',
COUNT(ENROLLMENT.SID) AS 'Total Enrolled', (SOCIETY.TotalSeats-
COUNT(ENROLLMENT.SID)) AS 'Unfilled Seats'
FROM SOCIETY
LEFT JOIN ENROLLMENT
ON SOCIETY.SocID=ENROLLMENT.SID
GROUP BY SOCIETY.SocName, SOCIETY.MentorName, SOCIETY.TotalSeats;

```
+-----+-----+-----+-----+-----+
| Society name | Mentor name | Total Capacity | Total Enrolled | Unfilled Seats |
+-----+-----+-----+-----+-----+
| Sashakt      | Dr. Anita   | 17             | 7              | 10             |
| Debating     | Dr. Yash    | 11             | 10             | 1              |
| Music        | Mr. Aman Gupta | 22            | 9              | 13             |
| Dancing      | Mr. Abhimanyu | 28            | 8              | 20             |
| NSS          | Mr. Narander | 33            | 3              | 30             |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```