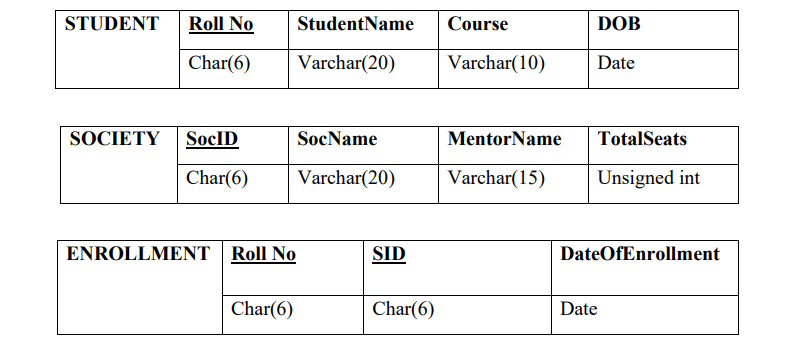
**Creating Student-Society database schema for a college and Querying it**



For Student Table

create table student(RollNo char(6) primary key, studentdName varchar(20), course varchar(10), dob date);

For Society Table

create table society(socID char(6) primary key, socName varchar(20), mentorName varchar(15), totalSeats int unsigned);

For Enrollment Table

create table enrollment(RollNo char(6), socId char(6),dateOfEnrollment date,foreign key (RollNo) references student(RollNo), foreign key(socId) references society(socID));

Queries :

1.Retrieve names of students enrolled in any society.

select studentName from student;

2. Retrieve all society names.

select socName from society;

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

select studentName from student where studentName like "A%";

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

select \* from student where (course = "computer science" or course = "chemistry");

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

select studentName from student where (RollNo like "X%9" or RollNo like "Z%9");

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

set @N = 10;

select \* from society where totalSeats > @N;

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

UPDATE society set mentorName = "Ditto" where socID = "1";

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

select society.socName from society join enrollment on society.socID = enrollment.socId group by society.socID, society.socName having count(enrollment.RollNo)>5;

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

select studentName from student s join enrollment e on s.RollNo = e.RollNo join society soc on soc.socID = e.socId where socName = "NSS" order by s.dob desc limit 1;

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

select socName from society s join enrollment e on s.socID = e.socId group by s.socID, s.socName order by count(e.RollNo) desc limit 1;

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

select socName from society s join enrollment e on s.socID = e.socId group by s.socID, s.socName order by count(e.RollNo) asc limit 2;

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

select studentName from student s where s.RollNo not in (select RollNo from enrollment);

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

select studentName from student where student.RollNo in (select RollNo from enrollment group by RollNo having count(RollNo)>2);

14. Find society names in which maximum students are enrolled

select s.socName from society s join enrollment e on s.socID = e.socId group by s.socID, s.socName order by count(e.RollNo) desc limit 1;

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.studentName,society.socName

-> from student join enrollment

-> on enrollment.RollNo = student.RollNo

-> join society on enrollment.socId = society.socID;

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

select studentName from student join enrollment on student.RollNo = enrollment.RollNo where enrollment.socId in (select socID from society where (socName ="Debating" or socName="Dancing" or socName="Sashakt"));

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

select socName from society where mentorName like "%Gupta";

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

select s.socName from society s join enrollment e on s.socID = e.socId group by s.socID, s.totalSeats having count(e.RollNo) = round((s.totalSeats\*0.1));

19. Display the vacant seats for each society.

select s.socName, (s.totalSeats - count(e.RollNo)) as vacant\_seats from society s join enrollment e on s.socID = e.socId group by s.socID;

20. Increment Total Seats of each society by 10%

update society set totalSeats =totalSeats+ totalSeats\*0.1;

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

alter table enrollment add enrollmentFees char(3);

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 =

-> case

-> when socId = "s1" then "2018-10-15"

-> when socId = "s2" then "2024-04-16"

-> when socId = "s3" then "2018-01-02"

-> end

-> where socId in ("3","7","8");

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

create view Track as select s.socID, s.socName, count(e.RollNo) as totalEnrolled from society s left join enrollment e on s.socID = e.socId group by s.socID, s.socName;

24. Find student names enrolled in all the societies.

select s.studentName from student s join enrollment e on s.RollNo = e.RollNo group by s.RollNo, s.studentName having count(distinct e.socId) = (select count(\*)from society);

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

select count(\*) as noOfSocOver5Member from society where socID in (select socId from enrollment group by socId having count(socId)>5);

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

alter table student add mobileNumber char(10) default "9999999999";

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

select count(\*) as over\_20\_student from student where datediff(current\_date(),dob)/365.25>20;

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

select studentName from student join enrollment on student.RollNo = enrollment.RollNo where year(dob)=2001;

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(e.RollNo) from enrollment e join society s on e.socId = s.socID where s.socName like "S%t" group by s.socId, s.socName having count(e.RollNo>5);

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

select s.socName as "Society Name", s.mentorName as "Mentor Name", count(e.RollNo) as "Total Enrolled", (s.totalSeats - count(e.RollNO)) as "Unfilled Seats" from society s left join enrollment e on s.socID = e.socId group by s.socID;