

## ASSIGNMENT -2

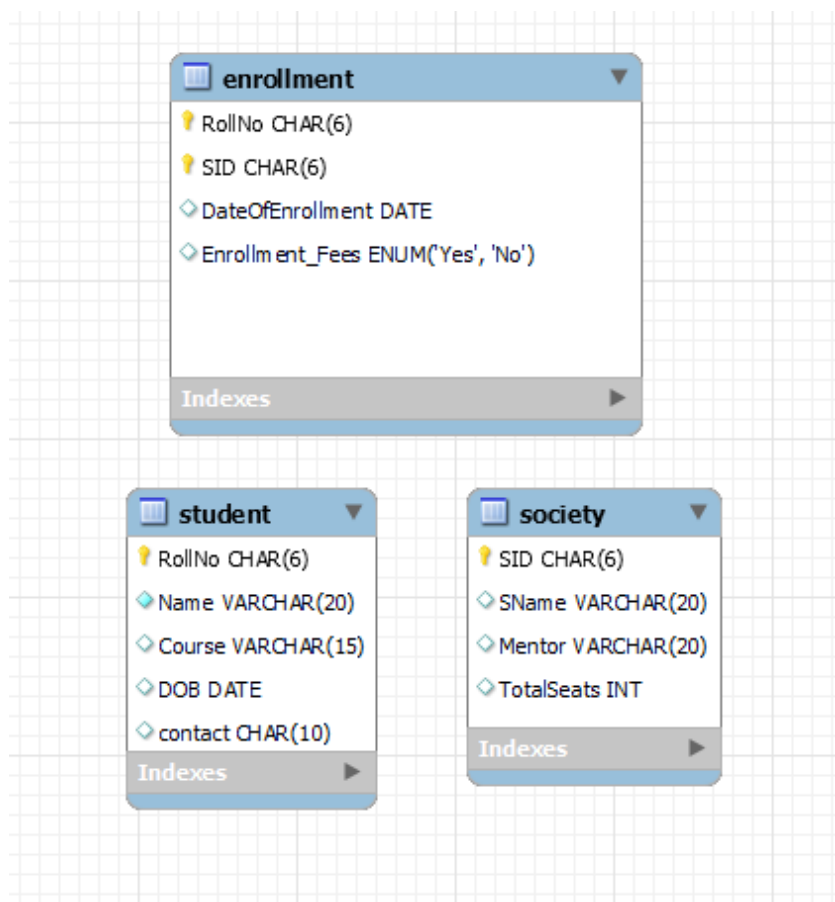
Submitted By : Avishkaar Pawar

Roll No : AD-1224

Course : B.Sc. Hons Computer Science

Semester : 4

Date of Submission : 21-Nov-2023



```
CREATE TABLE studentsociety.student (  
  `RollNo` CHAR(6) NOT NULL,  
  `Name` VARCHAR(20) NOT NULL,  
  `Course` VARCHAR(15) NULL DEFAULT NULL,  
  `DOB` DATE NULL DEFAULT NULL,  
  `contact` CHAR(10) NULL DEFAULT NULL,
```

PRIMARY KEY (`RollNo`)

)

---

```
CREATE TABLE IF NOT EXISTS `studentsociety`.`society` (  
  `SID` CHAR(6) NOT NULL,  
  `SName` VARCHAR(20) NULL DEFAULT NULL,  
  `Mentor` VARCHAR(20) NULL DEFAULT NULL,  
  `TotalSeats` INT UNSIGNED NULL DEFAULT NULL,  
  PRIMARY KEY (`SID`)
```

)

---

```
CREATE TABLE IF NOT EXISTS `studentsociety`.`enrollment` (  
  `RollNo` CHAR(6) NOT NULL,  
  `SID` CHAR(6) NOT NULL,  
  `DateOfEnrollment` DATE NULL DEFAULT '2023-01-01',  
  `Enrollment_Fees` ENUM('Yes', 'No') NULL DEFAULT 'No',  
  PRIMARY KEY (`RollNo`, `SID`)
```

)

---

- 1) Retrieve students name enrolled in any society.

```
mysql> SELECT DISTINCT Name FROM student WHERE rollno IN (SELECT rollno FROM enrollment);  
+-----+  
| Name |  
+-----+  
| Elija |  
| Farukh |  
| Ajit |  
| Ameer |  
| chaitanya |  
| Deshmukh |  
+-----+  
6 rows in set (0.00 sec)
```

- 2) Retrieve all society names.

```
mysql> select Sname from Society;
+-----+
| Sname |
+-----+
| Dhun  |
| Hackers |
| Nark  |
| Samaaj |
| Tahalka |
| Tark  |
+-----+
6 rows in set (0.00 sec)
```

- 3) Retrieve Student Name whose name starts with letter "A".

```
mysql> SELECT Name FROM Student WHERE Name LIKE 'A%';
+-----+
| Name |
+-----+
| Ajit |
| Ameer |
+-----+
2 rows in set (0.00 sec)
```

- 4) Retrieve Students studying in course CS(H) or LS

```
mysql> SELECT * FROM student WHERE course IN ("CS(H)", "LS");
+-----+-----+-----+-----+
| RollNo | Name      | Course | DOB      |
+-----+-----+-----+-----+
| AD0814 | Elija     | LS     | 2004-02-02 |
| AD0874 | Farukh    | LS     | 2003-12-02 |
| AD1221 | Ajit      | CS(H)  | 2003-12-04 |
| AD1230 | chaitanya | CS(H)  | 2003-11-28 |
| AD1232 | Deshmukh  | CS(H)  | 2003-11-02 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

- 5) Retrieve Students whose roll no starts with letter "C" or "A".

```
mysql> SELECT * FROM student WHERE rollno LIKE "A%" OR rollno LIKE 'C%';
```

RollNo	Name	Course	DOB
AD0814	Elija	LS	2004-02-02
AD0874	Farukh	LS	2003-12-02
AD1221	Ajit	CS(H)	2003-12-04
AD1222	Ameer	CS	2003-12-21
AD1230	chaitanya	CS(H)	2003-11-28
AD1232	Deshmukh	CS(H)	2003-11-02
AD1234	Elija	CS	2003-01-02
CX1225	Ranjeet	B.Com	2002-04-09

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

- 6) Retrieve Society whose capacity is more than 15.

```
mysql> SELECT sname FROM society WHERE TotalSeats > 15;
```

sname
Samaaj
Tahalka
Tark

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

- 7) Update Society Table for mentor name for a specific society.

```
mysql> UPDATE society
-> SET
-> Mentor = 'Mr Manoj Ghai '
-> WHERE
-> SName LIKE 'Tark';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0
```

```
mysql> select Sname,Mentor from society;
+-----+-----+
| Sname  | Mentor                |
+-----+-----+
| Dhun   | Mrs Suneeta           |
| Hackers| Mr X                  |
| Nark   | Mr Devil              |
| Samaaj | Mrs Rohini Kukreja    |
| Tahalka| Dr Titu               |
| Tark   | Mr Atul Vasant       |
+-----+-----+
6 rows in set (0.00 sec)
```

8) Remove details for a student who is not enrolled in any society.

```
mysql> DELETE FROM student
-> WHERE
->     rollno NOT IN (SELECT
->         rollno
->     FROM
->         enrollment);
Query OK, 1 row affected (0.01 sec)
```

9) Increment Capacity of society by 10%.

```
mysql> SELECT
->      *
-> FROM
->      society;
+-----+-----+-----+-----+
| SID   | SName  | Mentor          | TotalSeats |
+-----+-----+-----+-----+
| Soc001 | Dhun   | Mrs Suneeta     | 17         |
| Soc002 | Hackers | Mr X            | 13         |
| Soc003 | Nark   | Mr Devil        | 17         |
| Soc004 | Samaaj | Mrs Rohini Kukreja | 21         |
| Soc005 | Tahalka | Dr Titu         | 26         |
| Soc006 | Tark   | Mr Atul Vasant  | 26         |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> UPDATE society SET TotalSeats = TotalSeats*1.1;
Query OK, 6 rows affected (0.00 sec)
Rows matched: 6  Changed: 6  Warnings: 0
```

```
mysql> SELECT * FROM Society;
+-----+-----+-----+-----+
| SID   | SName  | Mentor          | TotalSeats |
+-----+-----+-----+-----+
| Soc001 | Dhun   | Mrs Suneeta     | 19         |
| Soc002 | Hackers | Mr X            | 14         |
| Soc003 | Nark   | Mr Devil        | 19         |
| Soc004 | Samaaj | Mrs Rohini Kukreja | 23         |
| Soc005 | Tahalka | Dr Titu         | 29         |
| Soc006 | Tark   | Mr Atul Vasant  | 29         |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

10) Find the students name who are not enrolled in any society.

```
mysql> SELECT DISTINCT
->      Name
-> FROM
->      student
->      RIGHT JOIN
->      enrollment ON student.rollno = enrollment.rollno;
+-----+
| Name   |
+-----+
| Elija  |
| chaitanya |
| Ajit   |
| Ameer  |
| Deshmukh |
| Farukh |
+-----+
6 rows in set (0.00 sec)
```

11) Find total number of Students whose age is greater than 19

```
mysql> select count(*) from student where ( (datediff(curdate(),dob))/365 >19);
+-----+
| count(*) |
+-----+
|          8 |
+-----+
1 row in set (0.01 sec)
```

12) Find the student names enrolled in atleast two societies

```
mysql> Select name from student natural join enrollment group by rollno having count(*)>1;
+-----+
| name   |
+-----+
| Ameer  |
| Elija  |
| Deshmukh |
+-----+
3 rows in set (0.00 sec)
```

13) Find society names in which any student is enrolled

```
mysql> select DISTINCT SName from society S,enrollment E where S.SID=E.SID ;
+-----+
| SName |
+-----+
| Dhun  |
| Hackers |
| Nark  |
+-----+
3 rows in set (0.01 sec)
```

14) Find names of all students enrolled in any society and society names in which any student is enrolled

```
mysql> select distinct name "Student Enrolled in any society" from student S where S.rollno in (select rollno from enrollment);
+-----+
| Student Enrolled in any society |
+-----+
| Elija |
| chaitanya |
| Ajit |
| Ameer |
| Deshmukh |
| Farukh |
+-----+
6 rows in set (0.00 sec)

mysql>
mysql> select DISTINCT SName "society names in which any student is enrolled" from society S,enrollment E where S.SID=E.SID ;
+-----+
| society names in which any student is enrolled |
+-----+
| Dhun |
| Hackers |
| Nark |
+-----+
3 rows in set (0.00 sec)
```

15) Find names of students who are enrolled in all three societies "Dhun","Hackers"and"Nark".

```
mysql> select name from (select name,S.rollno from student S, Enrollment E where S.rollno=E.rollno and sid in (select sid from society where sname in ("Dhun","Hackers","Nark"))) as X group by X.rollno having count(*)=3 ;
+-----+
| name |
+-----+
| Ameer |
+-----+
1 row in set (0.00 sec)
```

16) Find society names that has 'abc' as mentor or 'abc' as the name of enrolled student.



```
mysql> SELECT DISTINCT
->     Sname
-> FROM
->     Society,
->     student,
->     enrollment
-> WHERE
->     (student.rollno = enrollment.rollno
->         AND society.sid = enrollment.sid)
->         AND (society.mentor = 'Mr X'
->             OR student.name = 'Ameer');
```

Sname
Hackers
Dhun
Nark
Tahalka

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

17) Find society names whose mentor name is same as that of any enrolled student in it.

```
mysql> update society set Mentor="Ameer" where Sid="soc001";
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from society;
```

SID	SName	Mentor	TotalSeats
Soc001	Dhun	Ameer	19
Soc002	Hackers	Mr X	14
Soc003	Nark	Mr Devil	19
Soc004	Samaaj	Mrs Rohini Kukreja	23
Soc005	Tahalka	Dr Titu	29
Soc006	Tark	Mr Atul Vasant	29
Soc007	Zero	Shunya Prasad	13

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

```
mysql> select distinct Sname from Society,student,enrollment where (student.rollno=enrollment.rollno and society.sid=enrollment.sid) and (society.mentor=student.name);
```

Sname
Dhun

```
1 row in set (0.00 sec)
```

18) Find the society names in which number of enrolled students are less than its capacity.

```
mysql> create view vacant as
    -> select E.sname,TotalSeats,enrolled from society S , enroll_count E where S.sname=E.sname;
ERROR 1050 (42S01): Table 'vacant' already exists
mysql> select sname from vacant where totaleseats>enrolled;
+-----+
| sname |
+-----+
| Hackers |
| Nark |
| Samaaj |
| Tahalka |
| Tark |
| Zero |
+-----+
6 rows in set (0.00 sec)
```

19) Display the vacant seats for each society.

```
mysql> select sname,totalseats-enrolled as vacant from vacant;
+-----+-----+
| sname | vacant |
+-----+-----+
| Dhun | 0 |
| Hackers | 11 |
| Nark | 16 |
| Samaaj | 23 |
| Tahalka | 28 |
| Tark | 29 |
| Zero | 13 |
+-----+-----+
7 rows in set (0.00 sec)
```

20) Find society names in which more than one students have enrolled in the given year.

```
mysql> select * from enrollment;
+-----+-----+-----+-----+
| RollNo | SID   | DateOfEnrollment | Enrollment_Fees |
+-----+-----+-----+-----+
| AD1234 | Soc001 | 2018-01-15       | NULL            |
| AD1230 | Soc002 | 2023-02-21       | NULL            |
| AD1221 | Soc003 | 2018-01-02       | NULL            |
| AD1222 | Soc002 | 2023-02-21       | NULL            |
| AD1222 | Soc001 | 2018-01-15       | NULL            |
| AD0814 | Soc001 | 2018-01-15       | NULL            |
| AD0814 | Soc003 | 2018-01-02       | NULL            |
| AD1232 | Soc001 | 2018-01-15       | NULL            |
| AD1232 | Soc002 | 2023-02-21       | NULL            |
| AD0874 | Soc001 | 2018-01-15       | NULL            |
| AD1222 | Soc003 | 2018-01-02       | NULL            |
| AD1222 | Soc005 | 2022-08-12       | NULL            |
+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> set @year:=#year;
-> 2018;
Query OK, 0 rows affected (0.00 sec)

mysql> select sname from society S where sid in (select sid from (select sid from enrollment where year(DateOfEnrollment)=@year ) as E group by sid
having count(*)>1);
+-----+
| sname |
+-----+
| Dhun  |
| Nark  |
+-----+
2 rows in set (0.00 sec)
```

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

```
mysql> Alter table enrollment drop column Enrollment_Fees;
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> Alter table enrollment add column Enrollment_Fees Enum("Yes","No");
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

22) Update date of enrolment of society s1 to '2018-01-15', s2 to current date and s3 to '2018-01-02'.

```
mysql> update enrollment set dateofenrollment="2018-01-15" where sid="Soc001";
Query OK, 3 rows affected (0.01 sec)
Rows matched: 5  Changed: 3  Warnings: 0

mysql> update enrollment set dateofenrollment=curdate() where sid="Soc002";
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> update enrollment set dateofenrollment="2018-01-02" where sid="Soc003";
Query OK, 2 rows affected (0.01 sec)
Rows matched: 3  Changed: 2  Warnings: 0
```

23) Find society names whose enrolment is over.

```
mysql> create view vacant as
-> select E.sname,TotalSeats,enrolled from society S , enroll_count E where S.sname=E.sname;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from vacant;
+-----+-----+-----+
| sname | TotalSeats | enrolled |
+-----+-----+-----+
| Dhun  | 5          | 5        |
| Hackers | 14        | 3        |
| Nark  | 19        | 3        |
| Samaaj | 23        | 0        |
| Tahalka | 29       | 1        |
| Tark  | 29        | 0        |
| Zero  | 13        | 0        |
+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> select totalseats-enrolled as vacant from vacant;
+-----+
| vacant |
+-----+
| 0      |
| 11     |
| 16     |
| 23     |
| 28     |
| 29     |
| 13     |
+-----+
7 rows in set (0.00 sec)
```

```
mysql> select sname from vacant where totalseats-enrolled=0;
+-----+
| sname |
+-----+
| Dhun  |
+-----+
1 row in set (0.00 sec)
```

24) Find common societies of students of courses 'cs(h)' and 'cs'

```
mysql> select sname from society where sid in (select sid from enrollment E where E.rollno in (select rollno from Student St where (St.course like "
cs(h)")))) INTERSECT
-> select sname from society where sid in (select sid from enrollment E where E.rollno in (select rollno from Student St where (St.course like "
cs"))));
+-----+
| sname |
+-----+
| Hackers |
| Nark   |
| Dhun   |
+-----+
3 rows in set (0.00 sec)
```

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

```
mysql> create view enroll_count as
-> select sname , count(*) from (select * from society S natural join enrollment E ) as X group by sid;
Query OK, 0 rows affected (0.01 sec)
```

26) Find student names enrolled in all societies.

```
mysql> set @maxSociety:=(select count(*) from society);
Query OK, 0 rows affected (0.00 sec)

mysql> select name from student where rollno in (select rollno from enrollment group by rollno having count(*)=@maxSociety);
Empty set (0.01 sec)
```

27) Count societies with student enrolled > 3.

```
mysql> select count(*) from (select sid,count(*) from enrollment group by sid having count(*)>2) as X;
+-----+
| count(*) |
+-----+
|          3 |
+-----+
1 row in set (0.00 sec)
```

28) add column contact in student with default value.

```
mysql> alter table student add column contact char(10) default '1234567890';
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

29) Find the name of oldest and youngest student in class along with their age and DOB

```
mysql> set @maxdiff:= (select max(datediff(curdate(),dob)) from student);
Query OK, 0 rows affected (0.00 sec)

mysql> set @mindiff:= (select min(datediff(curdate(),dob)) from student);
Query OK, 0 rows affected (0.00 sec)

mysql> select name,datediff(curdate(),dob)/365 Age,dob from student where datediff(curdate(),dob)=@maxdiff union select name,datediff(curdate(),dob)/365 Age,dob from student where datediff(curdate(),dob)=@mindiff ;
+-----+-----+-----+
| name      | Age   | dob      |
+-----+-----+-----+
| Akshay Kumar | 20.9726 | 2002-03-08 |
| Rahul Dua   | 18.2192 | 2004-12-07 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

30) Find the most popular and least popular society name (on the basis of enrolled students)

```
mysql> set @maxCount:=(select max(x.count) from (select (count(rollno)) as Count from Society S left join Enrollment E on E.sid=S.sid group by S.sid) as X);
Query OK, 0 rows affected (0.00 sec)

mysql> set @minCount:=(select min(x.count) from (select (count(rollno)) as Count from Society S left join Enrollment E on E.sid=S.sid group by S.sid) as X);
Query OK, 0 rows affected (0.00 sec)

mysql> select sname Society_Name,x.count as Frequency from (select sname,count(rollno) Count from society S left join enrollment E on E.sid=S.sid group by S.sid) as X where x.count=@maxCount
-> union
-> select sname,x.count from (select sname,count(rollno) Count from society S left join enrollment E on E.sid=S.sid group by S.sid) as X where x.count=@minCount;
+-----+-----+
| Society_Name | Frequency |
+-----+-----+
| Dhun         | 5        |
| Samaaj       | 0        |
| Tark         | 0        |
| Zero         | 0        |
+-----+-----+
4 rows in set (0.00 sec)
```

31) Find names of students born in year 2003 and enrolled in atleast one society

```
mysql> Select distinct name from student,enrollment where student.rollno=enrollment.rollno and year(dob)='2003';
+-----+
| name      |
+-----+
| Elija     |
| chaitanya |
| Ajit      |
| Ameer     |
| Deshmukh  |
| Farukh    |
+-----+
6 rows in set (0.00 sec)
```

32) Remove default value of any field.

```
mysql> alter table student modify column contact char(10) ;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

33) Find society names where students have enrolled in month Jan/Feb

```
mysql> select distinct sname from society S natural join enrollment E where month(DateOfEnrollment)=2 or month(DateOfEnrollment)=1;
+-----+
| sname |
+-----+
| Dhun  |
| Hackers |
| Nark  |
+-----+
3 rows in set (0.00 sec)
```

34) Display all students details alongwith society name if they are enrolled in any society

```
mysql> select S.rollno,name,course,dob,contact,sname from student S natural join enrollment natural join society;
+-----+-----+-----+-----+-----+-----+
| rollno | name      | course | dob      | contact | sname |
+-----+-----+-----+-----+-----+-----+
| AD1234 | Elija     | CS      | 2003-01-02 | 1234567890 | Dhun |
| AD1230 | chaitanya | CS(H)    | 2003-11-28 | 1234567890 | Hackers |
| AD1221 | Ajit      | CS(H)    | 2003-12-04 | 1234567890 | Nark |
| AD1222 | Ameer     | CS      | 2003-12-21 | 1234567890 | Hackers |
| AD1222 | Ameer     | CS      | 2003-12-21 | 1234567890 | Dhun |
| AD0814 | Elija     | LS      | 2004-02-02 | 1234567890 | Dhun |
| AD0814 | Elija     | LS      | 2004-02-02 | 1234567890 | Nark |
| AD1232 | Deshmukh  | CS(H)    | 2003-11-02 | 1234567890 | Dhun |
| AD1232 | Deshmukh  | CS(H)    | 2003-11-02 | 1234567890 | Hackers |
| AD0874 | Farukh    | LS      | 2003-12-02 | 1234567890 | Dhun |
| AD1222 | Ameer     | CS      | 2003-12-21 | 1234567890 | Nark |
| AD1222 | Ameer     | CS      | 2003-12-21 | 1234567890 | Tahaalka |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

35) Display society names in uppercase and padded with character \* to get a length of 10 characters which are mentored by mentors whose names start with 'S' and ends with 'd' and capacity is between 10 to 20.

```
mysql> select * from society;
+-----+-----+-----+-----+
| SID   | SName   | Mentor      | TotalSeats |
+-----+-----+-----+-----+
| Soc001 | Dhun    | Ameer       | 19 |
| Soc002 | Hackers | Mr X        | 14 |
| Soc003 | Nark    | Mr Devil    | 19 |
| Soc004 | Samaaj  | Mrs Rohini Kukreja | 23 |
| Soc005 | Tahaalka | Dr Titu     | 29 |
| Soc006 | Tark    | Mr Atul Vasant | 29 |
| Soc007 | Zero    | Shunya Prasad | 13 |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

```
mysql> select lpad(upper(sname),10,"*") Society_Name from society where mentor like "S%d" and 10<totalSeats<20;;
+-----+
| Society_Name |
+-----+
| *****ZERO |
+-----+
1 row in set (0.00 sec)
```

