DBMS Practical

Create and use the following student-society database schema for a college to answer the given (sample) queries using the standalone SQL editor. STUDENT Roll No StudentName Course DOB Char(6) Varchar(20) Varchar(10) Date SOCIETY SocID SocName MentorName TotalSeats Char(6) Varchar(20) Varchar(15) Unsigned int ENROLLMENT Roll No Char(6) SID Char(6) DateOfEnrollment Date Here Rollno (ENROLLMENT) and SID (ENROLLMENT) are foreign keys.

mysql> show databases;

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| ramanujan          |
+--------------------+
4 rows in set (0.07 sec)
```

mysql> use ramanujan;

```
Database changed
```

mysql> show tables ;

```
+---------------------+
| Tables_in_ramanujan |
+---------------------+
| enrollment          |
| society             |
| student             |
+---------------------+
3 rows in set (0.00 sec)
```

mysql> desc student;

```
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| RollNo      | char(6)     | NO   | PRI | NULL    |       |
| StudentName | varchar(20) | NO   |     | NULL    |       |
| Course      | varchar(10) | NO   |     | NULL    |       |
| DOB         | date        | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
mysql> select * from student;
+--------+-------------+-----------+------------+
| RollNo | StudentName | Course    | DOB        |
+--------+-------------+-----------+------------+
| 11     | Aman        | CS        | 2004-05-12 |
| 12     | Raghav      | Chemistry | 2006-11-08 |
| 13     | Yubraj      | Physics   | 2005-09-11 |
| 14     | Kishan      | CS        | 2007-10-25 |
| 15     | Roshan      | Chemistry | 2006-08-14 |
| 16     | Nikhil      | CS        | 2004-08-19 |
| 17     | Sashank     | Chemistry | 2006-01-14 |
| 18     | Dinesh      | B.Voc     | 2003-06-15 |
| 19     | Rima        | B.Voc     | 2004-06-15 |
| 20     | Kesari      | Physics   | 2006-06-04 |
+--------+-------------+-----------+------------+
10 rows in set (0.00 sec)
mysql> select * from society;
+-------+-------------+------------+------------+
| SocID | SocName     | MentorName | TotalSeats |
+-------+-------------+------------+------------+
| 21    | NSS         | Gupta      |         20 |
| 22    | Dance       | Mehata     |         23 |
| 23    | Jazba       | Jain       |         24 |
| 24    | Debating    | Rima       |         30 |
| 25    | Sashakt     | Dhaka      |         35 |
| 26    | Shivranjini | Rawat      |         26 |
| 27    | DNA         | Bairav     |         50 |
+-------+-------------+------------+------------+
7 rows in set (0.00 sec)
mysql> select * from enrollment;
```

```
+--------+-----+------------------+

| RollNo | SID | DateOfEnrollment |

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

| 11     | 21  | 2021-05-06       |

| 12     | 22  | 2022-09-15       |

| 13     | 23  | 2023-02-24       |

| 14     | 24  | 2023-03-13       |

| 15     | 25  | 2023-04-15       |

| 16     | 26  | 2022-05-19       |

| 17     | 27  | 2022-09-25       |

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

7 rows in set (0.00 sec)
```

Q.1—Retrieve names of students enrolled in any society.

```
mysql> Select  StudentName from student where RollNo in (select RollNo from enrollment);

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

| StudentName |

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

| Aman        |

| Raghav      |

| Yubraj      |

| Kishan      |

| Roshan      |

| Nikhil      |

| Sashank     |

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

7 rows in set (0.00 sec)
```

Q.2—Retrieve all society names

```
Mysql>select SocName from society;

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

| socName     |

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

| NSS         |

| Dance       |
| Jazba       |
| Debating    |
| Sashakt     |
| Shivranjini |
| DNA         |
+-------------+
7 rows in set (0.00 sec)
```

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

```
mysql> Select StudentName from student where studentName like 'a%';
+-------------+
| StudentName |
+-------------+
| Aman        |
+-------------+
1 row in set (0.00 sec)
```

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

```
mysql> Select * from student where Course in ('CS','Chemistry');
+--------+-------------+-----------+------------+
| RollNo | StudentName | Course    | DOB        |
+--------+-------------+-----------+------------+
| 11     | Aman        | CS        | 2004-05-12 |
| 12     | Raghav      | Chemistry | 2006-11-08 |
| 14     | Kishan      | CS        | 2007-10-25 |
| 15     | Roshan      | Chemistry | 2006-08-14 |
| 16     | Nikhil      | CS        | 2004-08-19 |
| 17     | Sashank     | Chemistry | 2006-01-14 |
+--------+-------------+-----------+------------+
6 rows in set (0.00 sec)
```

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

```
mysql> select studentName from student where RollNo like '1%3' or RollNo like '1%8';
+-------------+
| studentName |
+-------------+
| Yubraj      |
| Dinesh      |
+-------------+
2 rows in set (0.03 sec)
```

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

```
Mysql>SELECT *FROM society WHERE TotalSeats > 30;
+-------+---------+------------+------------+
| SocID | SocName | MentorName | TotalSeats |
+-------+---------+------------+------------+
| 25    | Sashakt | Shyam      |         35 |
| 27    | DNA     | Bairav     |         50 |
+-------+---------+------------+------------+
```

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

```
mysql> UPDATE society SET MentorName = 'Shyam' WHERE SocName = 'Sashakt';
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0
mysql> select * from society;
+-------+-------------+------------+------------+
| SocID | SocName     | MentorName | TotalSeats |
+-------+-------------+------------+------------+
| 21    | NSS         | Gupta      |         20 |
| 22    | Dance       | Mehata     |         23 |
| 23    | Jazba       | Jain       |         24 |
| 24    | Debating    | Rima       |         30 |
| 25    | Sashakt     | Shyam      |         35 |
| 26    | Shivranjini | Rawat      |         26 |
| 27    | DNA         | Bairav     |         50 |
+-------+-------------+------------+------------+
7 rows in set (0.00 sec)
```

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

```
mysql> select SocName from society  where SocID IN (select SocID from enrollment GROUP BY SocID HAVING COUNT(RollNo) > 5);
+-------------+
| SocName     |
+-------------+
| NSS         |
| Dance       |
| Jazba       |
| Debating    |
| Sashakt     |
| Shivranjini |
| DNA         |
+-------------+
7 rows in set (0.08 sec)
```

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

```
mysql>SELECT StudentName FROM student WHERE RollNo IN(SELECT RollNo FROM enrollment WHERE SID = '21') ORDER BY DOB ASC LIMIT 1;
+-------------+
| StudentName |
+-------------+
| Aman        |
+-------------+
```

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

```
mysql>SELECT SocName FROM society JOIN enrollment ON SocID = SID GROUP BY SocID, SocName ORDER BY COUNT(RollNo) DESC LIMIT 1;
+---------+
| SocName |
+---------+
| NSS     |
+---------+
1 row in set (0.03 sec)
```

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

```
Mysql>SELECT SocName FROM society JOIN enrollment ON SocID = SID GROUP BY SocID, SocName ORDER BY COUNT(RollNo) ASC LIMIT 2;
+---------+
| SocName |
+---------+
| NSS     |

| Dance   |

+---------+

2 rows in set (0.00 sec)
```

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

```
Mysql>SELECT StudentName from  student  where RollNo NOT IN(select RollNo from enrollment);

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

| StudentName |

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

| Dinesh      |

| Rima        |

| Kesari      |

+-------------+
```

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

```
Mysql>select s.StudentName FROM student s JOIN enrollment e ON s.RollNo = e.RollNo GROUP BY
s.StudentName HAVING COUNT(*) >=1;

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

| StudentName |

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

| Aman        |

| Raghav      |

| Yubraj      |

| Kishan      |

| Roshan      |

| Nikhil      |

| Sashank     |

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

7 rows in set (0.00 sec)
```

Q.14—Find society names in which maximum students are enrolled

```
Mysql>SELECT MAX(RollNo) FROM enrollment;

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

| MAX(RollNo) |

+-------------+
| 17          |

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

1 row in set (0.00 sec)
```

Q.15- Find names of all students who have enrolled in any society and society names in which at least one 4 student has enrolled

```
Mysql>SELECT StudentName from student where RollNo IN (select RollNo from enrollment);

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

| StudentName |

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

| Aman        |

| Raghav      |

| Yubraj      |

| Kishan      |

| Roshan      |

| Nikhil      |

| Sashank     |

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

7 rows in set (0.00 sec)
```

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

```
Msql>SELECT StudentName FROM student WHERE RollNo IN(SELECT RollNo FROM enrollment WHERE SID
IN(SELECT SocID FROM society WHERE SocName IN('Debating','Dance','Sashakt'))) ;

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

| StudentName |

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

| Raghav      |

| Kishan      |

| Roshan      |

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

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

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

```
mysql> Select SocName from society where MentorName = 'Gupta';

+---------+

| SocName |
+---------+
| NSS     |
+---------+
1 row in set (0.00 sec
```

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

```
Mysql>SELECT SocName FROM society WHERE SocID IN(SELECT SID FROM enrollment GROUP BY SID HAVING COUNT(*) <=(0.1*TotalSeats)) ;
```

```
+-------------+
| SocName     |
+-------------+
| NSS         |
| Dance       |
| Jazba       |
| Debating    |
| Sashakt     |
| Shivranjini |
| DNA         |
+-------------+
```

Q.19—Display the vacant seats for each society.

```
Mysql>SELECT s.SocName, (s.TotalSeats - e.enrolled_count) AS vacant_seats FROM society s LEFT JOIN (SELECT SID, COUNT(*) AS enrolled_count FROM enrollment GROUP BY SID) e ON s.SocID = e.SID;
```

```
+-------------+--------------+
| SocName     | vacant_seats |
+-------------+--------------+
| NSS         |           21 |
| Dance       |           24 |
| Jazba       |           25 |
| Debating    |           32 |
| Sashakt     |           38 |
| Shivranjini |           28 |
| DNA         |           54 |
+-------------+--------------+
7 rows in set (0.03 sec)
```

Q.20—Increment Total Seats of each society by 10%

```
Mysql>UPDATE society SET TotalSeats = TotalSeats * 1.1;

Query OK, 7 rows affected (0.18 sec)

Rows matched: 7  Changed: 7  Warnings: 0

Mysql>select * from society;

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

| SocID | SocName     | MentorName | TotalSeats |

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

| 21    | NSS         | Gupta      |         22 |

| 22    | Dance       | Mehata     |         25 |

| 23    | Jazba       | Jain       |         26 |

| 24    | Debating    | Rima       |         33 |

| 25    | Sashakt     | Shyam      |         39 |

| 26    | Shivranjini | Rawat      |         29 |

| 27    | DNA         | Bairav     |         55 |

+-------+-------------+------------+------------
```

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

```
Mysql>ALTER TABLE enrollment ADD COLUMN fees_paid VARCHAR(3);

Query OK, 0 rows affected (0.60 sec)

Mysql>SELECT * from enrollment;

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

| RollNo | SID | DateOfEnrollment | fees_paid |

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

| 11     | 21  | 2021-05-06       | yes       |

| 12     | 22  | 2022-09-15       | no        |

| 13     | 23  | 2023-02-24       | no        |

| 14     | 24  | 2023-03-13       | yes       |

| 15     | 25  | 2023-04-15       | yes       |

| 16     | 26  | 2022-05-19       | yes       |

| 17     | 27  | 2022-09-25       | no        |

+--------+-----+------------------+-----------+
```

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

```
Mysql>UPDATE enrollment SET DateOfEnrollment = CASE SID WHEN 21 THEN '2021-06-17' WHEN 22 THEN
CURDATE() WHEN 24 THEN '2023-11-18' END WHERE SID IN(21, 22, 24);
```

```
Query OK, 3 rows affected (0.04 sec)

select * from enrollment;

+--------+-----+------------------+-----------+s

| RollNo | SID | DateOfEnrollment | fees_paid |

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

| 11     | 21  | 2021-06-17       | yes       |

| 12     | 22  | 2023-06-06       | no        |

| 13     | 23  | 2023-02-24       | no        |

| 14     | 24  | 2023-11-18       | yes       |

| 15     | 25  | 2023-04-15       | yes       |

| 16     | 26  | 2022-05-19       | yes       |

| 17     | 27  | 2022-09-25       | no        |

+--------+-----+------------------+-----------+
```

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

Mysql>CREATE VIEW SocietyEnrollmentView AS SELECT SocName, COUNT(SocID) AS TotalENrollment FROM society LEFT JOIN enrollment ON SocID = SID GROUP BY SocName;

```
Dance         1
Debating      1
DNA           1
Jazba         1
NSS           1
Sashakt       1
Shivranjini   1
Query OK, 0 rows affected (0.28 sec)
```

Q.24—Find student names enrolled in all the societies.

Mysql>SELECT s.StudentName FROM student s JOIN enrollment e ON s.RollNo = e.RollNo GROUP BY s.StudentName HAVING COUNT(DISTINCT e.SID) = (SELECT COUNT(*) FROM society);

```
Empty set (0.09 sec)
```

Q.25—Count number of societies with more than 5 student enrolled in it

Mysql>SELECT COUNT(*) AS societyCOUNT FROM societyenrollmentview WHERE Totalenrollment >5;

```
+--------------+

| societyCOUNT |

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

|            0 |

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

1 row in set (0.00 sec)
```

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

```
Mysql>ALTER TABLE student ADD MobileNumber VARCHAR(10) DEFAULT '9999999999';

Query OK, 0 rows affected (0.21 sec)

Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM student;

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

| RollNo | StudentName | Course    | DOB        | MobileNumber |

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

| 11     | Aman        | CS        | 2004-05-12 | 9999999999   |

| 12     | Raghav      | Chemistry | 2006-11-08 | 9999999999   |

| 13     | Yubraj      | Physics   | 2005-09-11 | 9999999999   |

| 14     | Kishan      | CS        | 2007-10-25 | 9999999999   |

| 15     | Roshan      | Chemistry | 2006-08-14 | 9999999999   |

| 16     | Nikhil      | CS        | 2004-08-19 | 9999999999   |

| 17     | Sashank     | Chemistry | 2006-01-14 | 9999999999   |

| 18     | Dinesh      | B.Voc     | 2003-06-15 | 9999999999   |

| 19     | Rima        | B.Voc     | 2004-06-15 | 9999999999   |

| 20     | Kesari      | Physics   | 2006-06-04 | 9999999999   |

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

10 rows in set (0.00 sec)
```

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

```
Mysql>SELECT COUNT(*) AS TotalStudent FROM student WHERE DOB > 20;

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

| TotalStudent |

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

|           10 |

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

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

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

```
SELECT DISTINCT s.StudentName FROM student s INNER JOIN enrollment e ON s.RollNo = e.RollNo WHERE
YEAR(s.DOB) = 2004;

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

| StudentName |
+-------------+
| Aman        |
| Nikhil      |
+-------------+
2 rows in set (0.01 sec)
```

28-- **SELECT StudentName FROM student WHERE YEAR(DOB) = 2004 AND RollNO IN(SELECT DISTINCT RollNo FROM enrollment);**

```
+-------------+
| StudentName |
+-------------+
| Aman        |
| Nikhil      |
+-------------+
2 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?

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

mysql> Select s.SocName, s.MentorName, s.TotalSeats AS TotalCapcity, COUNT(e.RollNo) AS TotalEnrolled, s.TotalSeats - COUNT(e.RollNo) AS UnfilledSeats From society s Left join enrollment e on s.SocID = e.SID group by s.SocID;

```
+-------------+------------+--------------+---------------+---------------+
| SocName     | MentorName | TotalCapcity | TotalEnrolled | UnfilledSeats |
+-------------+------------+--------------+---------------+---------------+
| NSS         | Gupta      |           22 |             1 |            21 |
| Dance       | Mehata     |           25 |             1 |            24 |
| Jazba       | Jain       |           26 |             1 |            25 |
| Debating    | Rima       |           33 |             1 |            32 |
| Sashakt     | Shyam      |           39 |             1 |            38 |
| Shivranjini | Rawat      |           29 |             1 |            28 |
| DNA         | Bairav     |           55 |             1 |            54 |
+-------------+------------+--------------+---------------+---------------+
7 rows in set (0.03 sec)
```