RAMANUJAN COLLEGE

UNIVERSITY OF DELHI

D.B.M.S

MY SQL Practicals

NAME- HARSHIT SRIVASTAVA

COURSE- B.SC(HONS.) COMPUTER SCIENCE

SEMESTER- 4

COLLEGE ROLL NO. - 20221464

CREATING THE DATABASE

```
CREATE DATABASE College;
USE College;
CREATE TABLE Student (
Roll_No CHAR(6) PRIMARY KEY,
StudentName VARCHAR(20),
Course VARCHAR(10),
DOB DATE );
CREATE TABLE Society (
SocID CHAR(6) PRIMARY KEY,
SocName VARCHAR(20),
MentorName VARCHAR(15),
TotalSeats INT UNSIGNED );
CREATE TABLE Enrollment (
Roll_No CHAR(6),
SID CHAR(6),
DateOfEnrollment DATE,
FOREIGN KEY(Roll_No) REFERENCES Student(Roll_No),
FOREIGN KEY(SID) REFERENCES Society(SocID)
ON UPDATE CASCADE
ON DELETE CASCADE );
INSERT INTO Student VALUES
("BY2M1G", "Aarav Patel", "Chemistry", "2004-08-15"),
("E41246", "Aisha Shah", "Computer", "2011-02-18"),
("H7I8H9", "Advik Singh", "Physics", "2008-06-21"),
("K1L2H3", "Ananya Mishra", "Chemistry", "2010-10-04"),
("G4O5P6", "Ishaan Kumar", "History", "2009-04-17"),
("Z7R8S9", "Kavya Gupta", "Computer", "2013-01-20"),
("T1F2V3", "Diya Sharma", "Biology", "2007-07-10"),
("W4XFY6", "Vihaan Joshi", "Maths", "2006-11-23"),
("Z7A869", "Riya Patel", "English", "2012-05-06"),
("X1D2S9", "Shaurya Reddy", "Chemistry", "2005-09-28"),
("F4G5F6", "Myra Kumar", "Computer", "2011-03-12"),
("I7J8KB", "Rudra Malhotra", "Physics", "2008-07-25"),
("L1M2NW", "Zoya Khan", "History", "2009-01-18"),
("Z4P5Q9", "Aryan Sharma", "Biology", "2010-09-11"),
("R7SST9", "Anika Singh", "Maths", "2007-03-24"),
("U1VDW3", "Ishan Patel", "English", "2013-08-07"),
("X4YNZ6", "Ananya Gupta", "Chemistry", "2005-12-15"),
("A7B3C9", "Arjun Singh", "Computer", "2011-06-28"),
("D1R2F3", "Anaya Verma", "Physics", "2008-10-31"),
("G4Y5I6", "Saanvi Kapoor", "History", "2009-04-04"),
("Z7I8L9", "Kabir Sharma", "Biology", "2010-11-27"),
("M1H2O3", "Pari Mishra", "Maths", "2002-06-10"),
("P455R6", "Ahaan Patel", "English", "2013-11-23"),
("B224D6", "Aditi Singhania", "Chemistry", "2004-08-25"),
("E345G7", "Aryan Kapoor", "Computer", "2011-02-28"),
("H4IRJ8", "Diya Mehra", "Physics", "2008-06-12"),
("K5LRM9", "Ishaan Gupta", "Chemistry", "2010-10-24"),
("N6O4P1", "Kavya Reddy", "History", "2009-04-07"),
("Q7R9S2", "Rohan Sharma", "Computer", "2013-01-10"),
("T8U1F3", "Ananya Rajput", "Biology", "2007-07-15"),("W9XJY4", "Arnav Khanna",
"Maths", "2006-11-28"),
("Z1AGB9", "Ishita Mehta", "English", "2012-05-11"),
("C6DEE1", "Riya Verma", "Chemistry", "2000-09-30"),
("F7F9H2", "Samarth Jain", "Computer", "2011-03-22"),
("I8F1K3", "Tanvi Agrawal", "Physics", "2001-07-05"),
```

```
("C8R3E9", "Ria Verma", "Chemistry", "2005-09-28"),
("F2I7H4", "Samaira Choudhary", "Computer", "2011-03-21"),
("I5J1K8", "Tanisha Singh", "Physics", "2008-07-04"),
("O6Q2Q7", "Zara Gupta", "Biology", "2010-09-20"),
("R1A1T3", "Aaradhya Malhotra", "Maths", "2007-04-13"),
("U8H4W9", "Aarav Verma", "English", "2013-08-26"),
("X3Z8Z1", "Ishita Sharma", "Chemistry", "2005-12-31"),
("A7V2C6", "Ritvik Singh", "Computer", "2011-06-14"),
("D3S9F4", "Shreya Malhotra", "Physics", "2008-10-07"),
("G7H2I6", "Vivaan Verma", "History", "2003-05-01"),
("J4Q9L3", "Zara Gupta", "Biology", "2010-11-24"),
("X136O9", "Aaradhya Malhotra", "Maths", "2007-06-17"),
("P5R1R7", "Ayaan Singh", "English", "2000-12-06");

INSERT INTO Society VALUES
("S1", "Prakritik Sangathan", "Rohan Gupta", 31),
("S2", "Dancing", "Aditya Sharma", 26),
("S3", "Cinema Samaj", "Aaradhya Patel", 35),
("S4", "Coding Samiti", "Gupta Singh", 28),
("S5", "NSS", "Sneha Gupta", 17),
("S6", "Rasoi Mandal", "Manvi Singh", 29),
("S7", "Vigyan Kalp", "Aryan Sharma", 36),
("S8", "Debating", "Ananya Gupta", 28),
("S9", "Sashakt", "Ishaan Joshi", 27),
("S10", "Robotics Parishad", "Aarav Singh", 19);
INSERT INTO Enrollment (Roll_No, SID, DateOfEnrollment) VALUES
("BY2M1G", "S1", "2024-05-03"),
("BY2M1G", "S2", "2024-05-03"),
("E41246", "S2", "2024-05-03"),
("E41246", "S3", "2024-05-03"),
("H7I8H9", "S4", "2024-05-03"),
("H7I8H9", "S5", "2024-05-03"),
("K1L2H3", "S6", "2024-05-03"),
("G4O5P6", "S8", "2024-05-03"),
("G4O5P6", "S9", "2024-05-03"),
("Z7R8S9", "S10", "2024-05-03"),
("Z7R8S9", "S1", "2024-05-03"),
("T1F2V3", "S2", "2024-05-03"),
("T1F2V3", "S3", "2024-05-03"),
("W4XFY6", "S4", "2024-05-03"),
("W4XFY6", "S5", "2024-05-03"),
("Z7A869", "S6", "2024-05-03"),
("Z7A869", "S7", "2024-05-03"),
("X1D2S9", "S8", "2024-05-03"),
("X1D2S9", "S9", "2024-05-03"),
("F4G5F6", "S10", "2024-05-03"),
("F4G5F6", "S1", "2024-05-03"),
("I7J8KB", "S2", "2024-05-03"),
("I7J8KB", "S3", "2024-05-03"),
("L1M2NW", "S4", "2024-05-03"),
("L1M2NW", "S5", "2024-05-03"),
("U7V9F1", "S7", "2024-05-03"),
("X2Y4Z9", "S8", "2024-05-03"),
("X2Y4Z9", "S9", "2024-05-03"),
("A3E5C7", "S10", "2024-05-03"),
("A3E5C7", "S1", "2024-05-03"),
("D6T8F1", "S2", "2024-05-03"),
("D6T8F1", "S3", "2024-05-03"),
("G7T9I2", "S4", "2024-05-03"),
("G7T9I2", "S5", "2024-05-03"),
("X8W1L9", "S6", "2024-05-03"),
("B4C7E2", "S2", "2024-05-03"),
```

("U8H4W9", "S3", "2024-05-03"),
("U8H4W9", "S4", "2024-05-03"),
("U8H4W9", "S5", "2024-05-03"),
("U8H4W9", "S7", "2024-05-03"),
("B4C7E2", "S3", "2024-05-03"),
("E8F3S9", "S4", "2024-05-03"),
("E8F3S9", "S5", "2024-05-03"),
("H2I5JH", "S6", "2024-05-03"),
("H2I5JH", "S7", "2024-05-03"),
("K7L3MD", "S8", "2024-05-03"),
("K7L3MD", "S9", "2024-05-03"),
("X1O6P9", "S10", "2024-05-03"),("X1O6P9", "S1", "2024-05-03"),
("Q5S9S3", "S2", "2024-05-03"),
("Q5S9S3", "S3", "2024-05-03"),
("T2A7V4", "S4", "2024-05-03"),
("T2A7V4", "S5", "2024-05-03"),
("W5XGY8", "S6", "2024-05-03"),
("W5XGY8", "S7", "2024-05-03"),
("Z9A4G6", "S8", "2024-05-03"),
("Z9A4G6", "S9", "2024-05-03"),
("C8D3EG", "S10", "2024-05-03"),
("F2G722", "S3", "2024-05-03"),
("I5J12W", "S4", "2024-05-03"),
("I5J12W", "S5", "2024-05-03"),
("Z3M9N9", "S6", "2024-05-03"),
("Z3M9N9", "S7", "2024-05-03"),
("O6QFQ7", "S8", "2024-05-03"),
("X3YAZ1", "S5", "2024-05-03"),
("U8H4W9", "S1", "2024-05-03"),
("U8H4W9", "S2", "2024-05-03"),
("O6QFQ7", "S9", "2024-05-03"),
("X3YAZ1", "S4", "2024-05-03"),
("U8H4W9", "S8", "2024-05-03"),
("U8H4W9", "S9", "2024-05-03"),
("A7B2CN", "S6", "2024-05-03"),
("A7B2CN", "S7", "2024-05-03"),
("D3E9FS", "S8", "2024-05-03"),
("D3E9FS", "S9", "2024-05-03"),
("G7H9ID", "S10", "2024-05-03"),
("G7H9ID", "S1", "2024-05-03"),
("J4K9LE", "S2", "2024-05-03"),
("J4K9LE", "S3", "2024-05-03"),
("X1N6O9", "S4", "2024-05-03"),
("X1N6O9", "S5", "2024-05-03"),
("B3Q8D2", "S8", "2024-05-03"),
("B3Q8D2", "S9", "2024-05-03"),
("T3S8V5", "S1", "2024-05-03"),
("W6C1Y9", "S2", "2024-05-03"),
("W6C1Y9", "S3", "2024-05-03"),
("C8R3E9", "S6", "2024-05-03"),
("C8R3E9", "S7", "2024-05-03"),
("F2I7H4", "S8", "2024-05-03"),
("F2I7H4", "S9", "2024-05-03"),
("I5J1K8", "S10", "2024-05-03"),
("I5J1K8", "S1", "2024-05-03"),
("O6Q2Q7", "S2", "2024-05-03"),
("O6Q2Q7", "S3", "2024-05-03"),
("R1A1T3", "S4", "2024-05-03"),
("R1A1T3", "S5", "2024-05-03"),
("U8H4W9", "S6", "2024-05-03"),
("U8H4W9", "S7", "2024-05-03"),("X3Z8Z1", "S8", "2024-05-03"),
("X3Z8Z1", "S9", "2024-05-03"),

```
("A7V2C6", "S10", "2024-05-03"),
("A7V2C6", "S1", "2024-05-03"),
("D3S9F4", "S2", "2024-05-03"),
("D3S9F4", "S3", "2024-05-03"),
("G7H2I6", "S4", "2024-05-03"),
("G7H2I6", "S5", "2024-05-03"),
("J4Q9L3", "S6", "2024-05-03"),
("J4Q9L3", "S7", "2024-05-03"),
("X136O9", "S8", "2024-05-03"),
("X136O9", "S9", "2024-05-03"),
("P5R1R7", "S10", "2024-05-03"),
("P5R1R7", "S1", "2024-05-03"),
("U8H4W9", "S10", "2024-05-03");
```

```
mysql>
mysql> INSERT INTO Student VALUES
    -> ("W77QBT", "Roman", "Chemistry", "2013-12-04"),
    -> ("K00JKG", "David", "Computer", "2012-06-08"),
    -> ("L21I3S", "Mike", "Physics", "2012-01-04"),
    -> ("XPVU29", "Lily", "Chemistry", "2007-07-12"),
    -> ("9REUV2", "Rose", "History", "2006-11-01"),
    -> ("X09UK5", "James", "Computer", "2010-05-03"),
    -> ("R4F2MQ", "Nicholas", "Physics", "2009-02-09"),
    -> ("BZSBEH", "Mike", "English", "2006-07-11"),
    -> ("ZR3IA9", "Brad", "History", "2011-08-10"),
    -> ("QNG1PR", "Bob", "Computer", "2004-09-07"),
    -> ("CJ18KU", "Emily", "Biology", "2008-04-15"),
    -> ("FJ52LM", "Sophia", "Math", "2009-09-22"),
    -> ("PK91ZV", "Daniel", "Chemistry", "2010-11-30"),
    -> ("MV73QX", "Emma", "Computer", "2007-03-18"),
    -> ("RJ29PY", "Olivia", "Physics", "2011-07-25"),
    -> ("HS67FG", "Ethan", "History", "2013-05-12"),
    -> ("LW82NJ", "Ava", "English", "2005-10-29"),
    -> ("GT46HB", "William", "Computer", "2006-12-08"),
    -> ("ZV95BC", "Isabella", "Physics", "2014-02-14"),
    -> ("QW36DS", "Alexander", "Chemistry", "2008-08-01"),
    -> ("DN24PL", "Mia", "History", "2012-01-19");
ERROR 1136 (21S01): Column count doesn't match value count at row 1
mysql>
mysql> INSERT INTO Society VALUES
    -> ("S1", "NSS", "Raj", 8),
    -> ("S2", "Debating", "Gupta", 4),
    -> ("S3", "Dancing", "Malik", 3),
    -> ("S4", "Sashakt", "Roy", 6),
    -> ("S5", "Music", "Singh", 5),
    -> ("S6", "Photography", "Sharma", 7),
    -> ("S7", "Literature", "Khan", 4),
    -> ("S8", "Drama", "Patel", 6),
    -> ("S9", "Art", "Chowdhury", 5),
    -> ("S10", "Sports", "Verma", 8);
ERROR 1062 (23000): Duplicate entry 'S1' for key 'society.PRIMARY'
mysql>
mysql> INSERT INTO Enrollment VALUES
    -> ("W77QBT", "S1", "2020-06-05"),
    -> ("K00JKG", "S2", "2020-07-10"),
    -> ("L21I3S", "S3", "2020-02-12"),
    -> ("XPVU29", "S1", "2020-04-01"),
    -> ("9REUV2", "S5", "2021-01-15"),
    -> ("X09UK5", "S1", "2022-08-26"),
    -> ("R4F2MQ", "S2", "2023-03-12"),
    -> ("BZSBEH", "S4", "2021-10-11"),
    -> ("ZR3IA9", "S1", "2021-07-30"),
    -> ("QNG1PR", "S2", "2023-04-09"),
    -> ("CJ18KU", "S6", "2022-11-20"),
    -> ("FJ52LM", "S1", "2021-09-08"),
    -> ("PK91ZV", "S8", "2022-03-25"),
```

```
    -> ("PK91ZV", "S8", "2022-03-25"),
    -> ("MV73QX", "S9", "2020-08-14"),
    -> ("RJ29PY", "S1", "2021-06-30"),
    -> ("HS67FG", "S4", "2022-02-19");
ERROR 1136 (21S01): Column count doesn't match value count at row 1
mysql>
mysql> SELECT * FROM Student;
+---------+-------------+-----------+------------+------------+
| Roll_No | StudentName | Course    | DOB        | MobileNo   |
+---------+-------------+-----------+------------+------------+
| 9REUV2  | Rose        | History   | 2006-11-01 | 9999999999 |
| BZSBEH  | Mike        | English   | 2006-07-11 | 9999999999 |
| CJ18KU  | Emily       | Biology   | 2008-04-15 | 9999999999 |
| DN24PL  | Mia         | History   | 2012-01-19 | 9999999999 |
| FJ52LM  | Sophia      | Math      | 2009-09-22 | 9999999999 |
| GT46HB  | William     | Computer  | 2006-12-08 | 9999999999 |
| HS67FG  | Ethan       | History   | 2013-05-12 | 9999999999 |
| K00JKG  | David       | Computer  | 2012-06-08 | 9999999999 |
| L21I3S  | Mike        | Physics   | 2012-01-04 | 9999999999 |
| LW82NJ  | Ava         | English   | 2005-10-29 | 9999999999 |
| MV73QX  | Emma        | Computer  | 2007-03-18 | 9999999999 |
| PK91ZV  | Daniel      | Chemistry | 2010-11-30 | 9999999999 |
| QNG1PR  | Bob         | Computer  | 2004-09-07 | 9999999999 |
| QW36DS  | Alexander   | Chemistry | 2008-08-01 | 9999999999 |
| R4F2MQ  | Nicholas    | Physics   | 2009-02-09 | 9999999999 |
| RJ29PY  | Olivia      | Physics   | 2011-07-25 | 9999999999 |
| W77QBT  | Roman       | Chemistry | 2013-12-04 | 9999999999 |
| X09UK5  | James       | Computer  | 2010-05-03 | 9999999999 |
| XPVU29  | Lily        | Chemistry | 2007-07-12 | 9999999999 |
| ZR3IA9  | Brad        | History   | 2011-08-10 | 9999999999 |
| ZV95BC  | Isabella    | Physics   | 2014-02-14 | 9999999999 |
+---------+-------------+-----------+------------+------------+
21 rows in set (0.00 sec)

mysql> SELECT * FROM Society;
+-------+-------------+--------------+------------+
| SocID | SocName     | MentorName   | TotalSeats |
+-------+-------------+--------------+------------+
| S1    | NSS         | Raj          |          9 |
| S10   | Sports      | Verma        |          9 |
| S2    | Debating    | Gupta        |          4 |
| S3    | Dancing     | Chirag Mittal |          3 |
| S4    | Sashakt     | Roy          |          7 |
| S5    | Music       | Singh        |          6 |
| S6    | Photography | Sharma       |          8 |
| S7    | Literature  | Khan         |          4 |
| S8    | Drama       | Patel        |          7 |
| S9    | Art         | Chowdhury    |          6 |
+-------+-------------+--------------+------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM Enrollment;
+---------+------+----------------+----------+
```

```
mysql> SELECT * FROM Enrollment;
+---------+------+------------------+-----------+
| Roll_No | SID  | DateOfEnrollment | FeesPaid  |
+---------+------+------------------+-----------+
| W77QBT  | S1   | 2018-01-15       | NULL      |
| K00JKG  | S2   | 2024-04-29       | NULL      |
| L21I3S  | S3   | 2018-01-02       | NULL      |
| XPVU29  | S1   | 2018-01-15       | NULL      |
| 9REUV2  | S5   | 2021-01-15       | NULL      |
| X09UK5  | S1   | 2018-01-15       | NULL      |
| R4F2MQ  | S2   | 2024-04-29       | NULL      |
| BZSBEH  | S4   | 2021-10-11       | NULL      |
| ZR3IA9  | S1   | 2018-01-15       | NULL      |
| QNG1PR  | S2   | 2024-04-29       | NULL      |
| CJ18KU  | S6   | 2022-11-20       | NULL      |
| FJ52LM  | S1   | 2018-01-15       | NULL      |
| PK91ZV  | S8   | 2022-03-25       | NULL      |
| MV73QX  | S9   | 2020-08-14       | NULL      |
| RJ29PY  | S1   | 2018-01-15       | NULL      |
| HS67FG  | S4   | 2022-02-19       | NULL      |
+---------+------+------------------+-----------+
16 rows in set (0.00 sec)

mysql>
mysql>
mysql> SELECT * FROM Student;
+---------+-------------+-----------+------------+------------+
| Roll_No | StudentName | Course    | DOB        | MobileNo   |
+---------+-------------+-----------+------------+------------+
| 9REUV2  | Rose        | History   | 2006-11-01 | 9999999999 |
| BZSBEH  | Mike        | English   | 2006-07-11 | 9999999999 |
| CJ18KU  | Emily       | Biology   | 2008-04-15 | 9999999999 |
| DN24PL  | Mia         | History   | 2012-01-19 | 9999999999 |
| FJ52LM  | Sophia      | Math      | 2009-09-22 | 9999999999 |
| GT46HB  | William     | Computer  | 2006-12-08 | 9999999999 |
| HS67FG  | Ethan       | History   | 2013-05-12 | 9999999999 |
| K00JKG  | David       | Computer  | 2012-06-08 | 9999999999 |
| L21I3S  | Mike        | Physics   | 2012-01-04 | 9999999999 |
| LW82NJ  | Ava         | English   | 2005-10-29 | 9999999999 |
| MV73QX  | Emma        | Computer  | 2007-03-18 | 9999999999 |
| PK91ZV  | Daniel      | Chemistry | 2010-11-30 | 9999999999 |
| QNG1PR  | Bob         | Computer  | 2004-09-07 | 9999999999 |
| QW36DS  | Alexander   | Chemistry | 2008-08-01 | 9999999999 |
| R4F2MQ  | Nicholas    | Physics   | 2009-02-09 | 9999999999 |
| RJ29PY  | Olivia      | Physics   | 2011-07-25 | 9999999999 |
| W77QBT  | Roman       | Chemistry | 2013-12-04 | 9999999999 |
| X09UK5  | James       | Computer  | 2010-05-03 | 9999999999 |
| XPVU29  | Lily        | Chemistry | 2007-07-12 | 9999999999 |
| ZR3IA9  | Brad        | History   | 2011-08-10 | 9999999999 |
| ZV95BC  | Isabella    | Physics   | 2014-02-14 | 9999999999 |
+---------+-------------+-----------+------------+------------+
21 rows in set (0.00 sec)
```

```
21 rows in set (0.00 sec)

mysql> SELECT * FROM Society;
+-------+------------+--------------+-----------+
| SocID | SocName    | MentorName   | TotalSeats |
+-------+------------+--------------+-----------+
| S1    | NSS        | Raj          |          9 |
| S10   | Sports     | Verma        |          9 |
| S2    | Debating   | Gupta        |          4 |
| S3    | Dancing    | Chirag Mittal |          3 |
| S4    | Sashakt    | Roy          |          7 |
| S5    | Music      | Singh        |          6 |
| S6    | Photography | Sharma       |          8 |
| S7    | Literature | Khan         |          4 |
| S8    | Drama      | Patel        |          7 |
| S9    | Art        | Chowdhury    |          6 |
+-------+------------+--------------+-----------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM Enrollment;
+---------+------+-----------------+----------+
| Roll_No | SID  | DateOfEnrollment | FeesPaid |
+---------+------+-----------------+----------+
| W77QBT  | S1   | 2018-01-15      | NULL     |
| K00JKG  | S2   | 2024-04-29      | NULL     |
| L21I3S  | S3   | 2018-01-02      | NULL     |
| XPVU29  | S1   | 2018-01-15      | NULL     |
| 9REUV2  | S5   | 2021-01-15      | NULL     |
| X09UK5  | S1   | 2018-01-15      | NULL     |
| R4F2MQ  | S2   | 2024-04-29      | NULL     |
| BZSBEH  | S4   | 2021-10-11      | NULL     |
| ZR3IA9  | S1   | 2018-01-15      | NULL     |
| QNG1PR  | S2   | 2024-04-29      | NULL     |
| CJ18KU  | S6   | 2022-11-20      | NULL     |
| FJ52LM  | S1   | 2018-01-15      | NULL     |
| PK91ZV  | S8   | 2022-03-25      | NULL     |
| MV73QX  | S9   | 2020-08-14      | NULL     |
| RJ29PY  | S1   | 2018-01-15      | NULL     |
| HS67FG  | S4   | 2022-02-19      | NULL     |
+---------+------+-----------------+----------+
16 rows in set (0.00 sec)
```

1. Retrieve names of students enrolled in any society.

```
SELECT DISTINCT StudentName FROM Student AS Stu INNER JOIN Enrollment AS Enr ON
Stu.Roll_No =    Enr.Roll_No;
```

```
mysql>
mysql> SELECT DISTINCT StudentName FROM Student AS Stu
    -> INNER JOIN Enrollment AS Enr
    -> ON Stu.Roll_No = Enr.Roll_No;
+-------------+
| StudentName |
+-------------+
| Rose        |
| Mike        |
| Emily       |
| Sophia      |
| Ethan       |
| David       |
| Emma        |
| Daniel      |
| Bob         |
| Nicholas    |
| Olivia      |
| Roman       |
| James       |
| Lily        |
| Brad        |
+-------------+
15 rows in set (0.00 sec)
```

2. Retrieve all society names.

```
SELECT SocName FROM Society;
```

```
mysql> SELECT SocName FROM Society;
+-------------+
| SocName     |
+-------------+
| NSS         |
| Sports      |
| Debating    |
| Dancing     |
| Sashakt     |
| Music       |
| Photography |
| Literature  |
| Drama       |
| Art         |
+-------------+
10 rows in set (0.00 sec)
```

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

```
SELECT StudentName FROM Student WHERE StudentName LIKE "A%";
```

```
mysql> SELECT StudentName FROM Student WHERE StudentName LIKE "A%";
+-------------+
| StudentName |
+-------------+
| Ava         |
| Alexander   |
+-------------+
2 rows in set (0.00 sec)
```

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

```
SELECT * FROM Student AS Stu INNER JOIN Enrollment AS Enr ON Stu.Roll_No =
Enr.Roll_No WHERE
Course = "Computer" OR Course = "Chemistry
```

```
mysql>
mysql> SELECT * FROM Student AS Stu
    -> INNER JOIN Enrollment AS Enr ON Stu.Roll_No = Enr.Roll_No
    -> WHERE Course = "Computer" OR Course = "Chemistry";
+---------+-------------+-----------+------------+------------+---------+------+-----------------+----------+
| Roll_No | StudentName | Course    | DOB        | MobileNo   | Roll_No | SID  | DateOfEnrollment | FeesPaid |
+---------+-------------+-----------+------------+------------+---------+------+-----------------+----------+
| K00JKG  | David       | Computer  | 2012-06-08 | 9999999999 | K00JKG  | S2   | 2024-04-29      | NULL     |
| MV73QX  | Emma        | Computer  | 2007-03-18 | 9999999999 | MV73QX  | S9   | 2020-08-14      | NULL     |
| PK91ZV  | Daniel      | Chemistry | 2010-11-30 | 9999999999 | PK91ZV  | S8   | 2022-03-25      | NULL     |
| QNG1PR  | Bob         | Computer  | 2004-09-07 | 9999999999 | QNG1PR  | S2   | 2024-04-29      | NULL     |
| W77QBT  | Roman       | Chemistry | 2013-12-04 | 9999999999 | W77QBT  | S1   | 2018-01-15      | NULL     |
| X09UK5  | James       | Computer  | 2010-05-03 | 9999999999 | X09UK5  | S1   | 2018-01-15      | NULL     |
| XPVU29  | Lily        | Chemistry | 2007-07-12 | 9999999999 | XPVU29  | S1   | 2018-01-15      | NULL     |
+---------+-------------+-----------+------------+------------+---------+------+-----------------+----------+
7 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 FROM Student WHERE Roll_No LIKE "X%9" OR Roll_No LIKE "Z%9";
```

```
mysql>
mysql> SELECT StudentName FROM Student
    -> WHERE Roll_No LIKE "X%9" OR Roll_No LIKE "Z%9";
+-------------+
| StudentName |
+-------------+
| Lily        |
| Brad        |
+-------------+
2 rows in set (0.01 sec)

mysql>
mysql> SET @N = 7;
Query OK, 0 rows affected (0.00 sec)
```

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

```
SET @N = 7;
SELECT * FROM Society WHERE TotalSeats > @N;
```

```
mysql>
mysql> SET @N = 7;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM Society WHERE TotalSeats > @N;
+-------+-------------+------------+------------+
| SocID | SocName     | MentorName | TotalSeats |
+-------+-------------+------------+------------+
| S1    | NSS         | Raj        |          9 |
| S10   | Sports      | Verma      |          9 |
| S6    | Photography | Sharma     |          8 |
+-------+-------------+------------+------------+
3 rows in set (0.00 sec)
```

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

```
UPDATE Society SET MentorName = "Chirag Mittal" WHERE SocName = "Dancing";
```

```
mysql>
mysql> UPDATE Society SET MentorName = "Chirag Mittal" WHERE SocName = "Dancing";
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0
```

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

```
SELECT SocName FROM Society AS Soc INNER JOIN Enrollment as Enr ON Soc.SocID =
Enr.SID GROUP BY SID HAVING COUNT(Enr.SID) >5;
```

```
mysql>
mysql> SELECT SocName FROM Society AS Soc
    -> INNER JOIN Enrollment as Enr ON Soc.SocID = Enr.SID
    -> GROUP BY SID HAVING COUNT(Enr.SID) >5;
+---------+
| SocName |
+---------+
| NSS     |
+---------+
1 row in set (0.00 sec)
```

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

```
SELECT StudentName FROM Student AS Stu INNER JOIN Enrollment AS Enr ON
Stu.Roll_No =
Enr.Roll_No INNER JOIN Society as Soc ON Enr.SID = Soc.SocID WHERE SocName =
"NSS" ORDER BY DOB
DESC LIMIT 1;
```

```
mysql>
mysql> SELECT StudentName FROM Student AS Stu
    -> INNER JOIN Enrollment AS Enr ON Stu.Roll_No = Enr.Roll_No
    -> INNER JOIN Society as Soc ON Enr.SID = Soc.SocID
    -> WHERE SocName = "NSS" ORDER BY DOB DESC LIMIT 1;
+-------------+
| StudentName |
+-------------+
| Roman       |
+-------------+
1 row in set (0.00 sec)
```

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

```
SELECT SocName FROM Society AS Soc INNER JOIN Enrollment as ENR ON Soc.SocID =
Enr.SID GROUP BY SocName ORDER BY COUNT(*) DESC LIMIT 1;
```

```
mysql>
mysql> SELECT SocName FROM Society AS Soc
    -> INNER JOIN Enrollment as ENR ON Soc.SocID = Enr.SID
    -> GROUP BY SocName ORDER BY COUNT(*) DESC LIMIT 1;
+---------+
| SocName |
+---------+
| NSS     |
+---------+
1 row in set (0.00 sec)
```

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

```
SELECT SocName FROM Society AS Soc INNER JOIN Enrollment as ENR ON Soc.SocID =
Enr.SID GROUP BY SocName ORDER BY COUNT(*) ASC LIMIT 2;
```

```
mysql>
mysql> SELECT SocName FROM Society AS Soc
    -> INNER JOIN Enrollment as ENR ON Soc.SocID = Enr.SID
    -> GROUP BY SocName ORDER BY COUNT(*) ASC LIMIT 2;
+---------+
| SocName |
+---------+
| Art     |
| Music   |
+---------+
2 rows in set (0.00 sec)
```

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

```
SELECT StudentName FROM Student AS Stu LEFT JOIN Enrollment AS Enr ON
Stu.Roll_No = Enr.Roll_No
```

```
mysql>
mysql> SELECT StudentName FROM Student AS Stu
    -> LEFT JOIN Enrollment AS Enr ON Stu.Roll_No = Enr.Roll_No
    -> WHERE SID IS NULL;
+-------------+
| StudentName |
+-------------+
| Mia         |
| William     |
| Ava         |
| Alexander   |
| Isabella    |
+-------------+
5 rows in set (0.00 sec)
```

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

```
SELECT StudentName FROM Student AS Stu INNER JOIN Enrollment AS Enr ON
Stu.Roll_No = Enr.Roll_No GROUP BY StudentName HAVING COUNT(*) >2;
```

```
mysql> SELECT StudentName FROM Student AS Stu
    -> INNER JOIN Enrollment AS Enr ON Stu.Roll_No = Enr.Roll_No
    -> GROUP BY StudentName HAVING COUNT(*) >2;
Empty set (0.00 sec)
```

14. Find society names in which maximum students are enrolled

```
SELECT SocName FROM Society AS Soc INNER JOIN Enrollment AS Enr ON Soc.SocID =
Enr.SID GROUP BY SocName ORDER BY COUNT(*) DESC LIMIT 1;
```

```
mysql> SELECT SocName FROM Society AS Soc
    -> INNER JOIN Enrollment AS Enr ON Soc.SocID = Enr.SID
    -> GROUP BY SocName ORDER BY COUNT(*) DESC LIMIT 1;
+---------+
| SocName |
+---------+
| NSS     |
+---------+
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 StudentName, 'Student' AS Designation FROM Student AS Stu INNER JOIN
Enrollment AS Enr
ON Stu.Roll_no = Enr.Roll_no UNION SELECT SocName, 'Society' AS Designation
FROM Society AS Soc INNER JOIN Enrollment AS Enr ON Soc.SocID = Enr.SID;
```

```
mysql> SELECT StudentName, 'Student' AS Designation FROM Student AS Stu
    -> INNER JOIN Enrollment AS Enr ON Stu.Roll_no = Enr.Roll_no
    -> UNION SELECT SocName, 'Society' AS Designation FROM Society AS Soc
    -> INNER JOIN Enrollment AS Enr ON Soc.SocID = Enr.SID;
+-------------+-------------+
| StudentName | Designation |
+-------------+-------------+
| Rose        | Student     |
| Mike        | Student     |
| Emily       | Student     |
| Sophia      | Student     |
| Ethan       | Student     |
| David       | Student     |
| Emma        | Student     |
| Daniel      | Student     |
| Bob         | Student     |
| Nicholas    | Student     |
| Olivia      | Student     |
| Roman       | Student     |
| James       | Student     |
| Lily        | Student     |
| Brad        | Student     |
| NSS         | Society     |
| Debating    | Society     |
| Dancing     | Society     |
| Sashakt     | Society     |
| Music       | Society     |
| Photography | Society     |
| Drama       | Society     |
| Art         | Society     |
+-------------+-------------+
23 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 DISTINCT StudentName FROM (SELECT StudentName, SID FROM Student AS Stu
INNER JOIN Enrollment AS Enr ON Stu.Roll_no = Enr.Roll_no) AS TEMP INNER JOIN
Society AS Soc ON TEMP.SID = Soc.SocID WHERE SocName = "Debating" OR SocName =
"Dancing" OR SocName = "Sashakt";
```

```
mysql>
mysql> SELECT DISTINCT StudentName FROM
    -> (SELECT StudentName, SID FROM Student AS Stu
    -> INNER JOIN Enrollment AS Enr ON Stu.Roll_no = Enr.Roll_no)
    -> AS TEMP INNER JOIN Society AS Soc ON TEMP.SID = Soc.SocID
    -> WHERE SocName = "Debating" OR SocName = "Dancing" OR SocName = "Sashakt";
+-------------+
| StudentName |
+-------------+
| David       |
| Nicholas    |
| Bob         |
| Mike        |
| Ethan       |
+-------------+
5 rows in set (0.00 sec)
```

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

```
SELECT SocName, MentorName FROM Society WHERE MentorName LIKE "%Gupta%";
```

```
mysql> SELECT SocName, MentorName FROM Society WHERE MentorName LIKE "%Gupta%";
+----------+------------+
| SocName  | MentorName |
+----------+------------+
| Debating | Gupta      |
+----------+------------+
1 row in set (0.00 sec)
```

18. Find the society names in which the number of enrolled students is only 12% of its capacity.

```
SELECT SocName, COUNT(*)*100/TotalSeats AS EnrollmentPercentage FROM (SELECT SID, COUNT(*) FROM Enrollment GROUP BY SID) AS Temp INNER JOIN Society AS Soc ON Temp.SID = Soc.SocID GROUP BY SocName, TotalSeats HAVING EnrollmentPercentage <12;
```

```
mysql>
mysql> SELECT SocName, COUNT(*)*100/TotalSeats AS EnrollmentPercentage FROM
    -> (SELECT SID, COUNT(*) FROM Enrollment GROUP BY SID)
    -> AS Temp INNER JOIN Society AS Soc ON Temp.SID = Soc.SocID
    -> GROUP BY SocName, TotalSeats HAVING EnrollmentPercentage <12;
+---------+----------------------+
| SocName | EnrollmentPercentage |
+---------+----------------------+
| NSS     |              11.1111 |
+---------+----------------------+
1 row in set (0.00 sec)
```

19. Display the vacant seats for each society.

```
SELECT SocName, TotalSeats - COUNT(*) AS VacantSeats FROM (SELECT SID, COUNT(*) FROM Enrollment GROUP BY SID) AS Temp INNER JOIN Society AS Soc ON Temp.SID = Soc.SocID GROUP BY SocName, TotalSeats;
```

```
mysql>
mysql> SELECT SocName, TotalSeats - COUNT(*) AS VacantSeats FROM
    -> (SELECT SID, COUNT(*) FROM Enrollment GROUP BY SID) AS Temp
    -> INNER JOIN Society AS Soc ON Temp.SID = Soc.SocID
    -> GROUP BY SocName, TotalSeats;
+-------------+-------------+
| SocName     | VacantSeats |
+-------------+-------------+
| NSS         |           8 |
| Debating    |           3 |
| Dancing     |           2 |
| Sashakt     |           6 |
| Music       |           5 |
| Photography |           7 |
| Drama       |           6 |
| Art         |           5 |
+-------------+-------------+
8 rows in set (0.00 sec)
```

20. Increment Total Seats of each society by 10%

```
UPDATE Society SET TotalSeats = 1.1*TotalSeats;
```

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

```
ALTER TABLE Enrollment ADD COLUMN FeesPaid 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 = "2018-01-15" WHERE SID = "S1";
UPDATE Enrollment SET DateOfEnrollment = "2024-04-29" WHERE SID = "S2";
UPDATE Enrollment SET DateOfEnrollment = "2018-01-02" WHERE SID = "S3";
```

```
mysql>
mysql> UPDATE Society SET TotalSeats = 1.1*TotalSeats;
Query OK, 7 rows affected (0.01 sec)
Rows matched: 10  Changed: 7  Warnings: 0

mysql>
mysql> ALTER TABLE Enrollment ADD COLUMN FeesPaid char(3);
ERROR 1060 (42S21): Duplicate column name 'FeesPaid'
mysql>
mysql> UPDATE Enrollment SET DateOfEnrollment = "2018-01-15" WHERE SID = "S1";
Query OK, 0 rows affected (0.00 sec)
Rows matched: 6  Changed: 0  Warnings: 0

mysql>
mysql> UPDATE Enrollment SET DateOfEnrollment = "2024-04-29" WHERE SID = "S2";
Query OK, 0 rows affected (0.00 sec)
Rows matched: 3  Changed: 0  Warnings: 0

mysql>
mysql> UPDATE Enrollment SET DateOfEnrollment = "2018-01-02" WHERE SID = "S3";
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0
```

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

```
CREATE VIEW View1 AS (SELECT SocName, Count(*) AS TotalStudents FROM Society AS Soc INNER JOIN Enrollment AS Enr ON Soc.SocID = Enr.SID GROUP BY SocName);
```

24. Find student names enrolled in all the societies.

```
SELECT StudentName FROM (SELECT StudentName, COUNT(DISTINCT SID) AS SocEnrolled FROM Student AS STU INNER JOIN Enrollment AS Enr ON Stu.Roll_no = Enr.Roll_No GROUP BY StudentName) AS Temp WHERE SocEnrolled = (SELECT COUNT(*) FROM Society);
```

```
mysql>
mysql> CREATE VIEW View1 AS
    -> (SELECT SocName, Count(*) AS TotalStudents
    -> FROM Society AS Soc INNER JOIN Enrollment AS Enr
    -> ON Soc.SocID = Enr.SID GROUP BY SocName);
ERROR 1050 (42S01): Table 'View1' already exists
mysql>
mysql> SELECT StudentName FROM
    -> (SELECT StudentName, COUNT(DISTINCT SID) AS SocEnrolled FROM Student AS STU
    -> INNER JOIN Enrollment AS Enr ON Stu.Roll_no = Enr.Roll_No GROUP BY StudentName)
    -> AS Temp WHERE SocEnrolled = (SELECT COUNT(*) FROM Society);
Empty set (0.00 sec)
```

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

```
SELECT COUNT(*) AS NumSocieties FROM (SELECT SocID FROM Society as Soc JOIN Enrollment as Enr ON Soc.SocId = Enr.SIDGROUP BY SocIDHAVING COUNT(*) > 5) AS Temp;
```

```
mysql> SELECT COUNT(*) AS NumSocieties FROM
    -> (SELECT SocID FROM Society as Soc JOIN Enrollment as Enr ON Soc.SocId = Enr.SID
    -> GROUP BY SocID HAVING COUNT(*) > 5) AS Temp;
+--------------+
| NumSocieties |
+--------------+
|            1 |
+--------------+
1 row in set (0.00 sec)
```

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

```
ALTER TABLE Student ADD COLUMN MobileNo BIGINT DEFAULT 9999999999;
```

```
mysql>
    -> ALTER TABLE Student ADD COLUMN MobileNo BIGINT DEFAULT 9999999999;
ERROR 1060 (42S21): Duplicate column name 'MobileNo'
```

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

```
SELECT COUNT(*) AS TotalStudents FROM Student WHERE
YEAR(CURDATE())YEAR(DOB) >20;
```

```
mysql>
mysql> SELECT COUNT(*) AS TotalStudents FROM Student WHERE YEAR(CURDATE()) - YEAR(DOB) >20;
+---------------+
| TotalStudents |
+---------------+
|             0 |
+---------------+
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 DISTINCT StudentName FROM Student AS Stu INNER JOIN Enrollment AS Enr ON
Stu.Roll_No = Enr.Roll_No WHERE YEAR(DOB) = 2001;
```

```
mysql>
    -> SELECT DISTINCT StudentName FROM Student AS Stu
    -> INNER JOIN Enrollment AS Enr ON Stu.Roll_No = Enr.Roll_No
    -> WHERE YEAR(DOB) = 2001;
Empty set (0.00 sec)

mysql>
    ->
```

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

```
SELECT SocName FROM Society AS Soc INNER JOIN Enrollment AS Enr ON Soc.SocID =
Enr.SID GROUP BY
SID HAVING SocName LIKE "S%t" AND COUNT(SID) > 5;
```

```
mysql>
    ->
    -> SELECT SocName FROM Society AS Soc
    -> INNER JOIN Enrollment AS Enr ON Soc.SocID = Enr.SID
    -> GROUP BY SID HAVING SocName LIKE "S%t" AND COUNT(SID) > 5;
Empty set (0.00 sec)
```

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

```
SELECT SocName AS SocietyName, MentorName, TotalSeats AS
TotalCapacity,TotalSeats - COUNT(*) AS
TotalEnrolledUnfilledSeat FROM Society AS Soc INNER JOIN Enrollment
AS Enr ON Soc.SocID = Enr.SID GROUP BY SocName, MentorName, TotalSeats;
```

```
mysql>
mysql> SELECT SocName AS SocietyName, MentorName, TotalSeats AS TotalCapacity,
    -> TotalSeats - COUNT(*) AS TotalEnrolledUnfilledSeat
    -> FROM Society AS Soc INNER JOIN Enrollment AS Enr ON Soc.SocID = Enr.SID
    -> GROUP BY SocName, MentorName, TotalSeats;
+-------------+---------------+---------------+---------------------------+
| SocietyName | MentorName    | TotalCapacity | TotalEnrolledUnfilledSeat |
+-------------+---------------+---------------+---------------------------+
| NSS         | Raj           |            10 |                         4 |
| Debating    | Gupta         |             4 |                         1 |
| Dancing     | Chirag Mittal |             3 |                         2 |
| Sashakt     | Roy           |             8 |                         6 |
| Music       | Singh         |             7 |                         6 |
| Photography | Sharma        |             9 |                         8 |
| Drama       | Patel         |             8 |                         7 |
| Art         | Chowdhury     |             7 |                         6 |
+-------------+---------------+---------------+---------------------------+
8 rows in set (0.00 sec)
```