UNIVERSITY OF DELHI

Ramanujan College

DSC 11: Database Management Systems

Semester-04

(2024-25)

Submitted By:

~Ratnesh Kumar

20231432(23020570034)

Submitted To:

~Dr. Kamlesh Kumar Raghuvanshi

Acknowledgement

I would like to take this opportunity to acknowledge everyone who has helped us in every stage of this project.

I am deeply indebted to my computer science Professor, Dr. Kamlesh Kumar Raghuvanshi for his guidance and suggestions in completing this practical. The completion of this practical was possible under his guidance and support.

I am also very thankful to my parents and my friends who have boosted me up morally with their continuous support.

At last but not least, I am very thankful to God almighty for showering his blessings upon me.

Index

21.	**Add the enrollment fees paid ('yes'/'No') field in the enrollment table.**	
22.	**Update date of enrollment of society id 's1' to '2018-01-15', 's2' to the current date, and 's3' to '2018-01-02'.**	
23.	**Create a view to keep track of society names with the total number of students enrolled in it.**	
24.	**Find student names enrolled in all the societies.**	
25.	**Count the number of societies with more than 5 students enrolled in it.**	
26.	**Add column **Mobile number** in student table with default value **'9999999999'**.**	
27.	**Find the total number of students whose age is > 20 years.**	
28.	**Find names of students who were born in 2001 and are enrolled in at least one society.**	
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**	
II.	***Do the following database administration commands:***	
1.	**Create user,**	
2.	**Create role**	
3.	**Grant privileges to a role**	
4.	**Revoke privileges from a role**	
5.	**Create index**	

1. Retrieve names of students enrolled in any society.

```
mysql> SELECT StudentName FROM STUDENT
    -> INNER JOIN ENROLLMENT
    -> ON STUDENT.RollNo = ENROLLMENT.RollNo;
+---------------------+
| StudentName         |
+---------------------+
| ISHIKA PURAKAYASTHA |
| Nitisha Agrawal     |
| PIYUSH KUSHWAHA     |
| RATNESH KUMAR       |
| SHIV KUMAR          |
| LOVISH JAIN         |
| YOGITA GUPTA        |
| VIKASH KUMAR        |
| Aditya Chaturvedi   |
| Ajay                |
| Sumit Kumar         |
| AKSHAT GUPTA        |
| YOGESH JAISWAL      |
```

2. Retrieve all society names.

```
mysql> SELECT SocName FROM SOCIETY;
+--------------+
| SocName      |
+--------------+
| NCC          |
| NSS          |
| Debating     |
| Dancing      |
| Sashakt      |
| Tech Club    |
| Photography  |
| Neev         |
+--------------+
8 rows in set (0.00 sec)
```

3. Retrieve student's names starting with the letter ‘A’.

```
mysql> SELECT StudentName FROM STUDENT
    -> WHERE StudentName LIKE "A%";
+--------------------+
| StudentName        |
+--------------------+
| Arun Kumar         |
| AYUSH TIWARI       |
| Amit Pal           |
| Aditya Chaturvedi  |
| Ajay               |
| Avnish Rana        |
| ANANYA             |
| AHSANA FATHIMA     |
| AKSHAT GUPTA       |
| Aditya Bisht       |
| Arjun Pahariya     |
| ANSHUL KASANA      |
| Avantika           |
+--------------------+
13 rows in set (0.00 sec)
```

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

```
mysql> SELECT * FROM STUDENT
    -> WHERE Course IN ('Comp Sci','Chemistry');
+--------+--------------------+-----------+------------+
| RollNo | StudentName        | Course    | DOB        |
+--------+--------------------+-----------+------------+
| S1005  | Geetanshu          | Chemistry | 2000-11-27 |
| S1006  | Kohana Bhalla      | Chemistry | 2001-01-19 |
| S1007  | Nitisha Agrawal    | Chemistry | 2001-03-15 |
| S1008  | KSHITIZ SADH       | Comp Sci  | 2002-04-22 |
| S1009  | PIYUSH KUSHWAHA    | Comp Sci  | 2002-06-15 |
| S1010  | Priyanshu Sisodiya | Comp Sci  | 2002-08-30 |
| S1011  | RAHUL BANSAL       | Comp Sci  | 2002-11-02 |
| S1012  | RATNESH KUMAR      | Comp Sci  | 2003-01-28 |
| S1013  | SHIV KUMAR         | Comp Sci  | 2003-03-19 |
| S1014  | SOURABH KUMAR      | Comp Sci  | 2003-05-14 |
| X1003  | Rashmi Bisht       | Chemistry | 2001-05-25 |
| X1004  | Ajay               | Chemistry | 2001-07-21 |
| X1005  | Sujal Singh        | Comp Sci  | 2003-07-29 |
| X1006  | Sumit Kumar        | Comp Sci  | 2003-09-10 |
| X1007  | Avnish Rana        | Comp Sci  | 2004-02-14 |
| Z1004  | YOGESH JAISWAL     | Chemistry | 2001-10-10 |
| Z1005  | Aditya Bisht       | Chemistry | 2002-02-05 |
| Z1006  | Shubham Kapoor     | Comp Sci  | 2004-04-09 |
| Z1007  | Arjun Pahariya     | Comp Sci  | 2004-06-23 |
| Z1008  | RAJ SINGH          | Comp Sci  | 2004-08-05 |
| Z1009  | VISHNU             | Comp Sci  | 2004-10-31 |
+--------+--------------------+-----------+------------+
21 rows in set (0.00 sec)
```

5. Retrieve student's names whose roll number either starts with ‘X’ or ‘Z’ and ends with ‘9’.

```
mysql> SELECT StudentName FROM STUDENT
    -> WHERE RollNo LIKE 'X%9' OR RollNo LIKE 'Z%9';
+-----------------+
| StudentName     |
+-----------------+
| PARTH BUDHIRAJA |
| VISHNU          |
+-----------------+
2 rows in set (0.00 sec)
```

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

```
mysql> SET @N =15;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM SOCIETY WHERE Totalseats > @N;
+-------+-----------+-------------+-------------+
| SocID | SocName   | MentorName  | TotalSeats  |
+-------+-----------+-------------+-------------+
| S1    | NCC       | Dr. Sharma  |          20 |
| S4    | Dancing   | Ms. Singh   |          20 |
| S6    | Tech Club | Ms. Verma   |          20 |
| S8    | Neev      | Dr. Kumar   |          20 |
+-------+-----------+-------------+-------------+
4 rows in set (0.00 sec)
```

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

```
mysql> UPDATE SOCIETY SET MentorName = 'Mr. Sonu'
    -> WHERE SocName = 'NCC';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

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

```
mysql> SELECT SocName FROM SOCIETY
    -> INNER JOIN ENROLLMENT
    -> ON SOCIETY.SocID = ENROLLMENT.SID
    -> GROUP BY ENROLLMENT.SID
    -> HAVING COUNT(ENROLLMENT.SID)>5;
+------------+
| SocName    |
+------------+
| NCC        |
| NSS        |
| Debating   |
| Sashakt    |
| Tech Club  |
+------------+
5 rows in set (0.00 sec)
```

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

```
mysql> SELECT StudentName FROM STUDENT
    -> INNER JOIN ENROLLMENT
    -> ON STUDENT.RollNo = ENROLLMENT.RollNo
    -> INNER JOIN SOCIETY
    -> ON SOCIETY.SocID = ENROLLMENT.SID
    -> WHERE SOCIETY.SocName = 'NSS'
    -> ORDER BY STUDENT.DOB DESC
    -> LIMIT 1;
+---------------+
| StudentName   |
+---------------+
| RISHABH GUPTA |
+---------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT SocName FROM SOCIETY
    -> INNER JOIN ENROLLMENT
    -> ON SOCIETY.SocID = ENROLLMENT.SID
    -> GROUP BY SOCIETY.SocID
    -> ORDER BY COUNT(SOCIETY.SocID) DESC
    -> LIMIT 1;
+---------+
| SocName |
+---------+
| NCC     |
+---------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT SocName FROM SOCIETY
    -> INNER JOIN ENROLLMENT
    -> ON SOCIETY.SocID = ENROLLMENT.SID
    -> GROUP BY SOCIETY.SocID
    -> ORDER BY COUNT(SOCIETY.SocID) ASC
    -> LIMIT 2;
+---------+
| SocName |
+---------+
| Neev    |
| Dancing |
+---------+
2 rows in set (0.00 sec)
```

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

```
mysql> SELECT StudentName FROM STUDENT
    -> LEFT JOIN ENROLLMENT
    -> ON STUDENT.RollNo = ENROLLMENT.RollNo
    -> WHERE ENROLLMENT.RollNo IS NULL;
+--------------------+
| StudentName        |
+--------------------+
| RENUKA             |
| RAHUL BANSAL       |
| SOURABH KUMAR      |
| JAGRIT GUPTA       |
| Kanishka Saini     |
| Vansh Banderwal    |
| RUPESH KUMAR       |
| AYUSH TIWARI       |
| Amit Pal           |
| Deepa              |
| ANANYA             |
| SUNNY              |
| SUJIT KUMAR YADAV  |
| AHSANA FATHIMA     |
| Bhumika Yadav      |
| Aditya Bisht       |
| ANSHUL KASANA      |
| Prachi Kumari      |
| Vishal             |
| Prayag Kaushik     |
| Avantika           |
+--------------------+
21 rows in set (0.00 sec)
```

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

```
mysql> SELECT StudentName FROM STUDENT
    -> LEFT JOIN ENROLLMENT ON STUDENT.RollNo=ENROLLMENT.RollNo
    -> GROUP BY STUDENT.StudentName
    -> HAVING COUNT(STUDENT.RollNo) >= 2;
+--------------------+
| StudentName        |
+--------------------+
| ISHIKA PURAKAYASTHA |
| Arun Kumar         |
+--------------------+
2 rows in set (0.00 sec)
```

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

```
mysql> SELECT SocName FROM SOCIETY
    -> LEFT JOIN ENROLLMENT ON SOCIETY.SocID=ENROLLMENT.SID
    -> GROUP BY SOCIETY.SocID
    -> order BY COUNT(SOCIETY.SocID) DESC
    -> LIMIT 1;
+---------+
| SocName |
+---------+
| NCC     |
+---------+
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.

```
mysql> (SELECT DISTINCT STUDENT.StudentName AS StudentSocietyName
    -> FROM STUDENT
    -> INNER JOIN ENROLLMENT ON STUDENT.RollNo = ENROLLMENT.RollNo)
    -> UNION
    -> (SELECT DISTINCT SOCIETY.SocName AS StudentSocietyName
    -> FROM SOCIETY
    -> INNER JOIN ENROLLMENT ON SOCIETY.SocID = ENROLLMENT.SID);
+--------------------+
| StudentSocietyName |
+--------------------+
| ISHIKA PURAKAYASTHA |
| Nitisha Agrawal    |
| PIYUSH KUSHWAHA    |
| RATNESH KUMAR      |
| SHIV KUMAR         |
| LOVISH JAIN        |
| YOGITA GUPTA       |
| VIKASH KUMAR       |
| Aditya Chaturvedi  |
```

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

```
mysql> SELECT DISTINCT STUDENT.StudentName FROM STUDENT
    -> INNER JOIN ENROLLMENT ON ENROLLMENT.RollNo = STUDENT.RollNo
    -> INNER JOIN SOCIETY ON SOCIETY.SocID = ENROLLMENT.SID
    -> WHERE SOCIETY.SocName IN ('Debating','Dancing','Sashakt');
+----------------------+
| StudentName          |
+----------------------+
| ISHIKA PURAKAYASTHA  |
| NISHANT SINGH        |
| Geetanshu            |
| KSHITIZ SADH         |
| Panchika Agrawal     |
| Suryash Vaibhav      |
| Sujal Singh          |
| PARTH BUDHIRAJA      |
| Mr. Rahul            |
| Priyanka Saini       |
| Avnish Rana          |
| Ujjwal Dhama         |
| KESHAV KHANDELWAL    |
| HARSH DUBEY          |
+----------------------+
14 rows in set (0.00 sec)
```

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

```
mysql> SELECT SocName FROM SOCIETY WHERE MentorName LIKE '%Gupta%';
+---------+
| SocName |
+---------+
| NSS     |
+---------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT SocName FROM SOCIETY
    -> INNER JOIN ENROLLMENT ON SOCIETY.SocID = ENROLLMENT.SID
    -> GROUP BY SOCIETY.SocID, SOCIETY.SocName, SOCIETY.TotalSeats
    -> HAVING COUNT(ENROLLMENT.RollNo) = (SOCIETY.TotalSeats * 0.1);
+---------+
| SocName |
+---------+
| Dancing |
+---------+
1 row in set (0.00 sec)
```

19. Display the vacant seats for each society.

```
mysql> SELECT SocName,TotalSeats - COUNT(ENROLLMENT.SID) AS VacantSeats
    -> FROM SOCIETY
    -> LEFT JOIN ENROLLMENT ON SOCIETY.SocID = ENROLLMENT.SID
    -> GROUP BY SOCIETY.SocID, SOCIETY.SocName, SOCIETY.TotalSeats;
+-------------+--------------+
| SocName     | VacantSeats  |
+-------------+--------------+
| NCC         |            6 |
| NSS         |            8 |
| Debating    |            2 |
| Dancing     |           18 |
| Sashakt     |            4 |
| Tech Club   |           14 |
| Photography |            0 |
| Neev        |           19 |
+-------------+--------------+
8 rows in set (0.00 sec)
```

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

```
mysql> UPDATE SOCIETY SET Totalseats = CEIL(Totalseats * 1.1);
Query OK, 8 rows affected (0.02 sec)
Rows matched: 8  Changed: 8  Warnings: 0

mysql> SELECT SocName AS 'Society Name',Totalseats AS 'Total Seat' FROM SOCIETY;
+--------------+-------------+
| Society Name | Total Seat  |
+--------------+-------------+
| NCC          |          22 |
| NSS          |          17 |
| Debating     |          11 |
| Dancing      |          22 |
| Sashakt      |          11 |
| Tech Club    |          22 |
| Photography  |           6 |
| Neev         |          22 |
+--------------+-------------+
8 rows in set (0.00 sec)
```

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

```
mysql> ALTER TABLE ENROLLMENT ADD COLUMN FeesPaid ENUM('Yes', 'No') DEFAULT 'No';
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC ENROLLMENT;
+------------------+------------------+------+-----+---------+-------+
| Field            | Type             | Null | Key | Default | Extra |
+------------------+------------------+------+-----+---------+-------+
| RollNo           | char(6)          | NO   | PRI | NULL    |       |
| SID              | char(6)          | NO   | PRI | NULL    |       |
| DateOfEnrollment | date             | YES  |     | NULL    |       |
| FeesPaid         | enum('Yes','No') | YES  |     | No      |       |
+------------------+------------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

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

```
mysql> UPDATE ENROLLMENT
    -> SET DateOfEnrollment = CASE
    ->     WHEN SID = 's1' THEN '2018-01-15'
    ->     WHEN SID = 's2' THEN CURDATE()
    ->     WHEN SID = 's3' THEN '2018-01-02'
    -> END
    -> WHERE SID IN ('s1', 's2', 's3');
Query OK, 0 rows affected (0.00 sec)
Rows matched: 29  Changed: 0  Warnings: 0

mysql> SELECT SID,DateOfEnrollment FROM ENROLLMENT
    -> WHERE SID IN ('S1','S2','S3')
    -> GROUP BY SID,DateOfEnrollment;
+-----+------------------+
| SID | DateOfEnrollment |
+-----+------------------+
| S1  | 2018-01-15       |
| S2  | 2025-02-09       |
| S3  | 2018-01-02       |
+-----+------------------+
3 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.

```
mysql> CREATE VIEW EnrollmentOfSociety AS
    -> SELECT SOCIETY.SocName, COUNT(ENROLLMENT.RollNo) AS TotalStudents FROM SOCIETY
    -> LEFT JOIN ENROLLMENT ON SOCIETY.SocID = ENROLLMENT.SID
    -> GROUP BY SOCIETY.SocID;
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT * FROM EnrollmentOfSociety;
+-------------+---------------+
| SocName     | TotalStudents |
+-------------+---------------+
| NCC         |            14 |
| NSS         |             7 |
| Debating    |             8 |
| Dancing     |             2 |
| Sashakt     |             6 |
| Tech Club   |             6 |
| Photography |             5 |
| Neev        |             1 |
+-------------+---------------+
8 rows in set (0.00 sec)
```

24. Find student names enrolled in all the societies.

```
mysql> SELECT StudentName FROM STUDENT
    -> INNER JOIN ENROLLMENT ON STUDENT.RollNo = ENROLLMENT.RollNo
    -> GROUP BY ENROLLMENT.RollNo
    -> HAVING COUNT(ENROLLMENT.SID) > (SELECT COUNT(*) FROM SOCIETY);
Empty set (0.00 sec)
```

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

```
mysql> SELECT COUNT(*)  AS 'Number Of Society' FROM (SELECT SID FROM ENROLLMENT
    -> GROUP BY SID HAVING COUNT(RollNo)> 5) AS SOCIETY;
+------------------+
| Number Of Society |
+------------------+
|                5 |
+------------------+
1 row in set (0.00 sec)
```

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

```
mysql> ALTER TABLE STUDENT ADD COLUMN MobileNumber VARCHAR(10) DEFAULT '9999999999';
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

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       |       |
| MobileNumber | varchar(10) | YES  |     | 9999999999 |       |
+--------------+-------------+------+-----+------------+-------+
5 rows in set (0.00 sec)
```

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

```
mysql> SELECT COUNT(*) AS Students FROM STUDENT
    -> WHERE TIMESTAMPDIFF(YEAR, DOB, CURDATE()) > 20;
+----------+
| Students |
+----------+
|       26 |
+----------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT DISTINCT StudentName FROM STUDENT
    -> JOIN ENROLLMENT ON STUDENT.RollNo = ENROLLMENT.RollNo
    -> WHERE YEAR(STUDENT.DOB) = 2001;
+-----------------+
| StudentName     |
+-----------------+
| Nitisha Agrawal |
| Ajay            |
| YOGESH JAISWAL  |
| Kohana Bhalla   |
| Rashmi Bisht    |
+-----------------+
5 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.

```
mysql> SELECT SocName FROM SOCIETY
    -> INNER JOIN ENROLLMENT ON SOCIETY.SocID = ENROLLMENT.SID
    -> WHERE SocName LIKE 'S%T'
    -> GROUP BY ENROLLMENT.SID HAVING COUNT(ENROLLMENT.SID)>= 5;
+---------+
| SocName |
+---------+
| Sashakt |
+---------+
1 row in set (0.00 sec)
```

30. Display the following information:

- **Society name**
- **Mentor name**
- **Total Capacity**
- **Total Enrolled**
- **Unfilled Seats**

```
mysql> SELECT SocName AS "Society Name",
    -> MentorName AS "Mentor Name",
    -> Totalseats AS "Total Capacity",
    -> COUNT(ENROLLMENT.RollNo) AS "Total Enrolled",
    -> TotalSeats - COUNT(ENROLLMENT.RollNo) AS "Unfilled Seats" FROM SOCIETY
    -> LEFT JOIN ENROLLMENT ON SOCIETY.SocID = ENROLLMENT.SID
    -> GROUP BY SOCIETY.SocID;
+--------------+-------------+----------------+----------------+----------------+
| Society Name | Mentor Name | Total Capacity | Total Enrolled | Unfilled Seats |
+--------------+-------------+----------------+----------------+----------------+
| NCC          | Mr. Sonu    |             22 |             14 |              8 |
| NSS          | Dr. Gupta   |             17 |              7 |             10 |
| Debating     | Mr. Mehta   |             11 |              8 |              3 |
| Dancing      | Ms. Singh   |             22 |              2 |             20 |
| Sashakt      | Dr. Yadav   |             11 |              6 |              5 |
| Tech Club    | Ms. Verma   |             22 |              6 |             16 |
| Photography  | Ms. Kapoor  |              6 |              5 |              1 |
| Neev         | Dr. Kumar   |             22 |              1 |             21 |
+--------------+-------------+----------------+----------------+----------------+
8 rows in set (0.00 sec)
```

II. Database Administration Commands

- **Create user**

```
mysql> CREATE USER 'Ratnesh'@'localhost' IDENTIFIED BY 'Ratnesh@123';
Query OK, 0 rows affected (0.05 sec)

mysql> SELECT User, Host FROM mysql.user;
+------------------+-----------+
| User             | Host      |
+------------------+-----------+
| Ratnesh          | localhost |
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| root             | localhost |
+------------------+-----------+
5 rows in set (0.00 sec)
```

- **Create role**

```
mysql> CREATE ROLE 'developer';
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW GRANTS FOR 'developer';
+----------------------------------------+
| Grants for developer@%                 |
+----------------------------------------+
| GRANT USAGE ON *.* TO `developer`@`%`  |
+----------------------------------------+
1 row in set (0.00 sec)
```

- **Grant privileges to a role**

```
mysql> GRANT SELECT, INSERT, UPDATE ON Student_Society.* TO 'developer';
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW GRANTS FOR 'developer';
+-----------------------------------------------------------------------+
| Grants for developer@%                                                |
+-----------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `developer`@`%`                                 |
| GRANT SELECT, INSERT, UPDATE ON `student_society`.* TO `developer`@`%` |
+-----------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

- **Revoke privileges from a role**

```
mysql> REVOKE UPDATE ON Student_Society.* FROM 'developer';
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW GRANTS FOR 'developer';
+-----------------------------------------------------------------+
| Grants for developer@%                                          |
+-----------------------------------------------------------------+
| GRANT USAGE ON *.* TO `developer`@`%`                           |
| GRANT SELECT, INSERT ON `student_society`.* TO `developer`@`%` |
+-----------------------------------------------------------------+
2 rows in set (0.00 sec)
```

- **Create index**

```
mysql> CREATE UNIQUE INDEX Ratnesh ON Student_Society.society (SocID);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT INDEX_NAME, COLUMN_NAME, NON_UNIQUE
    -> FROM INFORMATION_SCHEMA.STATISTICS
    -> WHERE TABLE_NAME = 'society'
    -> AND TABLE_SCHEMA = 'Student_Society';
+------------+-------------+------------+
| INDEX_NAME | COLUMN_NAME | NON_UNIQUE |
+------------+-------------+------------+
| PRIMARY    | SocID       |          0 |
| Ratnesh    | SocID       |          0 |
+------------+-------------+------------+
2 rows in set (0.00 sec)
```