Database Task: -

CREATING TABLES: -

1. STUDENT table

- CREATE:
  Mysql> create table student (STUDENT_ID int primary key auto_increment, FIRST_NAME varchar (255), LAST_NAME varchar (255), GPA float (20),
  -> ENROLLMENT_DATE datetime, MAJOR Varchar (100));
  Query OK, 0 rows affected (0.07 sec)

- INSERT:

  Mysql> insert into student (FIRST_NAME, LAST_NAME, GPA, ENROLLMENT_DATE, MAJOR) values

  -> ("Rakesh", "Kumar", 5.60, now (),"Biology"),

  -> ("Radha", "Sharma", 9.20, now (), "Chemistry"),

  -> ("Kush", "Kumar", 7.85, now (), "Physics"),

  -> ("Prem", "Chopra”, 9.56, now (), "History"),

  -> ("Pankaj", "Vats”, 9.78, now (), "English"),

  -> ("Navleen", "Kaur", 7.00, now (), "Mathematic");

  Query OK, 6 rows affected (0.01 sec)

  Records: 6 Duplicates: 0 Warnings: 0

Mysql> select * from student;

```
+------------+------------+-----------+------+--------------------+---------------
--+
| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA |
ENROLLMENT_DATE     | MAJOR            |
+------------+------------+-----------+------+--------------------+---------------
--+
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 12:19:09 | Computer
Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 12:22:07 |
Mathematics      |
|          3 | Rakesh     | Kumar     | 5.6 | 2024-03-20 12:25:20 | Biology
|
|          4 | Radha      | Sharma    | 9.2 | 2024-03-20 12:25:20 | Chemistry
|
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 12:25:20 | Physics
|
|          6 | Prem       | Chopra    | 9.56 | 2024-03-20 12:25:20 | History
|
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 12:25:20 | English          |
|          8 | Navleen    | Kaur      |    7 | 2024-03-20 12:25:20 | Mathematic
|
+------------+------------+-----------+------+--------------------+---------------
--+
```

8 rows in set (0.06 sec)

2. PROGRAM table

- CREATE

```
Mysql> CREATE TABLE Program (
    ->     id int AUTO_INCREMENT,
    ->     STUDENT_ID int,
    ->     PROGRAM_NAME VARCHAR (255),
    ->     PROGRAM_START_DATE datetime,
    ->     PRIMARY KEY (id),
    ->     FOREIGN KEY (STUDENT_ID) REFERENCES student
(STUDENT_ID)
    -> );
Query OK, 0 rows affected (0.08 sec)
```

- INSERT

```
Mysql> insert into Program (STUDENT_ID, PROGRAM_NAME,
PROGRAM_START_DATE) values
    -> (1,"Computer Science”, now ()),
    -> (2,"Mathematics", now ()),
    -> (8,"Mathematics”, now ()),
    -> (5,"Physics”, now ()),
    -> (4,"Chemistry", now ()),
    -> (7,"Psychology", now ()),
    -> (6,"History", now ()),
    -> (3,"Biology", now ())
    -> ;
Query OK, 8 rows affected (0.01 sec)
Records: 8 Duplicates: 0 Warnings: 0
```

Mysql> select * from program;

```
+----+------------+-----------------+--------------------+
| Id | STUDENT_ID | PROGRAM_NAME    | PROGRAM_START_DATE |
+----+------------+-----------------+--------------------+
| 1 |          1 | Computer Science | 2024-03-20 13:26:56 |
| 2 |          2 | Mathematics      | 2024-03-20 13:26:56 |
| 3 |          8 | Mathematics      | 2024-03-20 13:26:56 |
| 4 |          5 | Physics          | 2024-03-20 13:26:56 |
| 5 |          4 | Chemistry        | 2024-03-20 13:26:56 |
| 6 |          7 | Psychology       | 2024-03-20 13:26:56 |
| 7 |          6 | History          | 2024-03-20 13:26:56 |
| 8 |          3 | Biology          | 2024-03-20 13:26:56 |
+----+------------+-----------------+--------------------+
```

8 rows in set (0.00 sec)

3. SCHOLARSHIP table

- CREATE
  Mysql> CREATE TABLE Scholarship (id int primary key auto_increment, STUDENT_ID int, SCHOLARSHIP_AMOUNT int,
  -> SCHOLARSHIP_DATE datetime,
  -> FOREIGN KEY (STUDENT_ID) REFERENCES student (STUDENT_ID))
  -> ;
  Query OK, 0 rows affected (0.13 sec)

- INSERT
  Mysql> insert into Scholarship (STUDENT_ID, SCHOLARSHIP_AMOUNT, SCHOLARSHIP_DATE) values
  -> (1, 5000, now ()),
  -> (2, 4500, now ()),
  -> (3, 3000, now ()),
  -> (1, 4000, now ());

Query OK, 4 rows affected (0.04 sec)
Records: 4 Duplicates: 0 Warnings: 0

Mysql> select * from scholarship;

Id	STUDENT_ID	SCHOLARSHIP_AMOUNT	SCHOLARSHIP_DATE
1	1	5000	2024-03-20 14:21:20
2	2	4500	2024-03-20 14:21:20
3	3	3000	2024-03-20 14:21:20
4	1	4000	2024-03-20 14:21:20

4 rows in set (0.00 sec)

QUERIES

1. Write a SQL query to fetch “FIRST_NAME” from the Student table in upper Case and use ALIAS name as STUDENT_NAME.

```
Mysql> SELECT UPPER (FIRST_NAME) AS STUDENT_NAME FROM STUDENT;
+--------------+
| STUDENT_NAME |
+--------------+
| SHIVANSH     |
| UMESH        |
| RAKESH       |
| RADHA        |
| KUSH         |
| PREM         |
| PANKAJ       |
| NAVLEEN      |
+--------------+
8    rows in set (0.01 sec)
```

2. Write a SQL query to fetch unique values of MAJOR Subjects from Student Table.

```
Mysql> SELECT DISTINCT MAJOR FROM STUDENT;
+------------------+
| MAJOR            |
+------------------+
| Computer Science |
| Mathematic       |
| Biology          |
| Chemistry        |
| Physics          |
| History          |
| English          |
+------------------+
7 rows in set (0.00 sec)
```

3. Write a SQL query to print the first 3 characters of FIRST_NAME from Student table.

```
Mysql> SELECT SUBSTRING (FIRST_NAME, 1, 3)
   -> FROM STUDENT;
+--------------------------+
| SUBSTRING (FIRST_NAME, 1, 3) |
+--------------------------+
| Shi                      |
| Ume                      |
| Rak                      |
| Rad                      |
| Kus                      |
| Pre                      |
| Pan                      |
| Nav                      |
+--------------------------+
8 rows in set (0.00 sec)
```

4. Write a SQL query to find the position of alphabet (‘a’) int the first name column ‘Shivansh’ from Student table

```
Mysql> SELECT FIRST_NAME, POSITION ("a" IN FIRST_NAME) FROM
STUDENT WHERE FIRST_NAME="Shivansh";
+------------+-----------------------------+
| FIRST_NAME | POSITION ("a" IN FIRST_NAME) |
+------------+-----------------------------+
| Shivansh   |                           5 |
+------------+-----------------------------+
1 row in set (0.00 sec)
```

5. Write a SQL query that fetches the unique values of MAJOR Subjects from Student table and print its length.

```
Mysql> SELECT DISTINCT MAJOR, length (MAJOR) FROM STUDENT;
+-----------------+---------------+
| MAJOR           | length (MAJOR) |
+-----------------+---------------+
| Computer Science |           16 |
| Mathematic      |           10 |
| Biology         |            7 |
| Chemistry       |            9 |
| Physics         |            7 |
| History         |            7 |
| English         |            7 |
+-----------------+---------------+
7 rows in set (0.00 sec)
```

6. Write a SQL query to print FIRST_NAME from the Student table after Replacing ‘a’ with ‘A’.

```
Mysql> SELECT REPLACE (FIRST_NAME,"a","A") FROM STUDENT;
+-----------------------------+
| REPLACE (FIRST_NAME,"a","A") |
+-----------------------------+
| ShivAnsh                    |
| Umesh                       |
| RAkesh                      |
| RAdhA                       |
| Kush                        |
| Prem                        |
| PAnkAj                      |
| NAvleen                     |
```

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

8 rows in set (0.00 sec)

7. Write a SQL query to print the FIRST_NAME and LAST_NAME from Student Table into single column COMPLETE_NAME.

```
Mysql> SELECT CONCAT (FIRST_NAME," ", LAST_NAME) FROM STUDENT;
+---------------------------------+
| CONCAT (FIRST_NAME," ", LAST_NAME) |
+---------------------------------+
| Shivansh Mahajan                |
| Umesh Sharma                    |
| Rakesh Kumar                    |
| Radha Sharma                    |
| Kush Kumar                      |
| Prem Chopra                     |
| Pankaj Vats                     |
| Navleen Kaur                    |
+---------------------------------+
8 rows in set (0.00 sec)
```

8. Write a SQL query to print all Student details from Student table order by FIRST_NAME Ascending and MAJOR Subject descending.

```
Mysql> select * from student ORDER BY FIRST_NAME ASC, MAJOR DESC;
+------------+------------+-----------+------+---------------------+------------------+
| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE     | MAJOR            |
+------------+------------+-----------+------+---------------------+------------------+
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 12:25:20 | Physics          |
|          8 | Navleen    | Kaur      |    7 | 2024-03-20 12:25:20 | Mathematic       |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 12:25:20 | English          |
|          6 | Prem       | Chopra    | 9.56 | 2024-03-20 12:25:20 | History          |
```

| 4 | Radha | Sharma | 9.2 | 2024-03-20 12:25:20 | Chemistry |

| 3 | Rakesh | Kumar | 5.6 | 2024-03-20 12:25:20 | Biology |

| 1 | Shivansh | Mahajan | 8.79 | 2024-03-20 12:19:09 | Computer Science |

| 2 | Umesh | Sharma | 8.44 | 2024-03-20 12:22:07 | Mathematic |

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

8 rows in set (0.01 sec)

9. Write a SQL query to print details of the Students with the FIRST_NAME as ‘Prem’ and ‘Shivansh’ from Student table.

```
Mysql> SELECT * FROM STUDENT
   -> Where FIRST_NAME="Prem" or first_name="Shivansh";
```

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

| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE | MAJOR |

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

| 1 | Shivansh | Mahajan | 8.79 | 2024-03-20 12:19:09 | Computer Science |

| 6 | Prem | Chopra | 9.56 | 2024-03-20 12:25:20 | History |

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

2 rows in set (0.00 sec)

10. Write a SQL query to print details of the Students excluding FIRST_NAME as ‘Prem’ and ‘Shivansh’ from Student table.

```
Mysql> SELECT * FROM STUDENT WHERE FIRST_NAME not in ("Prem","Shivansh");
```

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

| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE | MAJOR |

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

| 2 | Umesh | Sharma | 8.44 | 2024-03-20 12:22:07 | Mathematic |

| 3 | Rakesh | Kumar | 5.6 | 2024-03-20 12:25:20 | Biology |

| 4 | Radha | Sharma | 9.2 | 2024-03-20 12:25:20 | Chemistry |

| 5 | Kush | Kumar | 7.85 | 2024-03-20 12:25:20 | Physics |

| 7 | Pankaj | Vats | 9.78 | 2024-03-20 12:25:20 | English |

| 8 | Navleen | Kaur | 7 | 2024-03-20 12:25:20 | Mathematic |

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

6 rows in set (0.00 sec)

11. Write a SQL query to print details of the Students whose FIRST_NAME

Ends with ‘a’.

Mysql> SELECT * FROM STUDENT

-> WHERE FIRST_NAME LIKE '%a';

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

| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE | MAJOR |

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

| 4 | Radha | Sharma | 9.2 | 2024-03-20 12:25:20 | Chemistry |

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

1 row in set (0.00 sec)

12. Write an SQL query to print details of the Students whose FIRST_NAME

Ends with ‘a’ and contains six alphabets.

Mysql> SELECT * FROM STUDENT WHERE FIRST_NAME LIKE '%a' AND LENGTH (FIRST_NAME)>6;

Empty set (0.00 sec)

13. Write an SQL query to print details of the Students whose GPA lies

Between 9.00 and 9.99.

Mysql> SELECT * FROM STUDENT

-> WHERE GPA BETWEEN 9.00 and 9.99;

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

| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE | MAJOR |

```
+------------+------------+-----------+------+---------------------+-----------+
|          4 | Radha      | Sharma    | 9.2  | 2024-03-20 12:25:20 | Chemistry |
|          6 | Prem       | Chopra    | 9.56 | 2024-03-20 12:25:20 | History   |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 12:25:20 | English   |
+------------+------------+-----------+------+---------------------+-----------+
```

3 rows in set (0.00 sec)

14. Write an SQL query to fetch the count of Students having Major Subject

'Computer Science'.

```
Mysql> SELECT COUNT (*) FROM STUDENT WHERE MAJOR = "Computer Science";
+----------+
| COUNT (*) |
+----------+
|        1 |
+----------+
```

1 row in set (0.00 sec)

15. Write an SQL query to fetch Students full names with GPA >= 8.5 and <=

9.5.

```
Mysql> SELECT CONCAT (FIRST_NAME," ", LAST_NAME), GPA FROM STUDENT WHERE GPA >= 8.5 AND 9.5;
+----------------------------------+------+
| CONCAT (FIRST_NAME," ", LAST_NAME) | GPA |
+----------------------------------+------+
| Shivansh Mahajan                 | 8.79 |
| Radha Sharma                     | 9.2  |
| Prem Chopra                      | 9.56 |
| Pankaj Vats                      | 9.78 |
+----------------------------------+------+
```

4 rows in set (0.00 sec)

16. Write an SQL query to fetch the no. of Students for each MAJOR subject in the descending order.

```
Mysql> SELECT COUNT (*) AS COUNT, MAJOR
    -> FROM STUDENT
    -> GROUP BY MAJOR
    -> ORDER BY MAJOR DESC;
+-------+------------------+
| COUNT | MAJOR            |
+-------+------------------+
|     1 | Physics          |
|     2 | Mathematic       |
|     1 | History          |
|     1 | English          |
|     1 | Computer Science |
|     1 | Chemistry        |
|     1 | Biology          |
+-------+------------------+
7 rows in set (0.00 sec)
```

17. Display the details of students who have received scholarships, including their names, scholarship amounts, and scholarship dates.

```
Mysql> SELECT CONCAT (FIRST_NAME," ", LAST_NAME) AS FULL_NAME, SUM (SCHOLARSHIP_AMOUNT)
    -> FROM STUDENT AS ST
    -> INNER JOIN SCHOLARSHIP AS SCH
    -> ON ST.STUDENT_ID=SCH.STUDENT_ID
    -> GROUP BY FULL_NAME;
+------------------+-------------------------+
| FULL_NAME        | SUM (SCHOLARSHIP_AMOUNT) |
```

```
+-----------------+------------------------+
| Shivansh Mahajan |                   9000 |
| Umesh Sharma     |                   4500 |
| Rakesh Kumar     |                   3000 |
+-----------------+------------------------+
```

3 rows in set (0.00 sec)

18. Write an SQL query to show only odd rows from Student table.

```
Mysql> SELECT * FROM STUDENT WHERE MOD (STUDENT_ID, 2) <> 0;
+------------+------------+-----------+------+---------------------+-----------------+
| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE     | MAJOR           |
+------------+------------+-----------+------+---------------------+-----------------+
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 12:19:09 | Computer Science |
|          3 | Rakesh     | Kumar     | 5.6 | 2024-03-20 12:25:20 | Biology          |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 12:25:20 | Physics          |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 12:25:20 | English          |
+------------+------------+-----------+------+---------------------+-----------------+
```

4 rows in set (0.00 sec)

19. Write an SQL query to show only even rows from Student table.

```
Mysql> SELECT * FROM STUDENT WHERE MOD (STUDENT_ID, 2) = 0;
+------------+------------+-----------+------+---------------------+------------+
| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE     | MAJOR      |
+------------+------------+-----------+------+---------------------+------------+
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 12:22:07 | Mathematic |
|          4 | Radha      | Sharma    | 9.2 | 2024-03-20 12:25:20 | Chemistry |
|          6 | Prem       | Chopra    | 9.56 | 2024-03-20 12:25:20 | History    |
|          8 | Navleen    | Kaur      |    7 | 2024-03-20 12:25:20 | Mathematic |
+------------+------------+-----------+------+---------------------+------------+
```

4 rows in set (0.00 sec)

20. List all students and their scholarship amounts if they have received any.
If a student has not received a scholarship, display NULL for the scholarship details.

FULL OUTER JOIN ERROR

21. Write an SQL query to show the top n (say 5) records of Student table order by descending GPA.

```
Mysql> SELECT * FROM STUDENT
   -> ORDER BY GPA DESC
   -> LIMIT 5;
+------------+------------+-----------+------+--------------------+-----------------+
| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE    | MAJOR           |
+------------+------------+-----------+------+--------------------+-----------------+
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 12:25:20 | English         |
|          6 | Prem       | Chopra    | 9.56 | 2024-03-20 12:25:20 | History         |
|          4 | Radha      | Sharma    | 9.2 | 2024-03-20 12:25:20 | Chemistry       |
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 12:19:09 | Computer Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 12:22:07 | Mathematic      |
+------------+------------+-----------+------+--------------------+-----------------+
5 rows in set (0.01 sec)
```

22. Write an SQL query to determine the nth (say n=5) highest GPA from a Table.

```
Mysql> SELECT * FROM STUDENT
   -> ORDER BY GPA
   -> DESC LIMIT 4, 1;
+------------+------------+-----------+------+--------------------+------------+
```

| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE | MAJOR |

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

| 2 | Umesh | Sharma | 8.44 | 2024-03-20 12:22:07 | Mathematic |

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

1 row in set (0.00 sec)

23. Write an SQL query to determine the 5th highest GPA without using LIMIT

Keyword.

```
Mysql> SELECT * FROM (
   -> SELECT ROW_NUMBER () OVER (ORDER BY GPA DESC) AS RANKING, GPA
   -> FROM STUDENT)
   -> AS FIFTHGPA
   -> WHERE RANKING = 5;
```

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

| RANKING | GPA |

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

| 5 | 8.44 |

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

1 row in set (0.00 sec)

24. Write an SQL query to fetch the list of Students with the same GPA.

```
Mysql> select * from STUDENT where GPA in (
   ->    select GPA from STUDENT
   ->    group by GPA having count (*) > 1);
```

Empty set (0.03 sec)

25. Write an SQL query to show the second highest GPA from a Student table

using sub-query.

```
Mysql> SELECT MAX (GPA) FROM STUDENT
   -> WHERE GPA< (SELECT MAX (GPA) FROM STUDENT);
```

```
+----------+
| MAX(GPA) |
+----------+
|     9.56 |
+----------+
1 row in set (0.00 sec)
```

26. Write an SQL query to show one row twice in results from a table.

```
Mysql> SELECT * FROM STUDENT
   -> UNION ALL
   -> SELECT * FROM STUDENT;
+------------+------------+-----------+------+---------------------+-----------------+
| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE     | MAJOR           |
+------------+------------+-----------+------+---------------------+-----------------+
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 12:19:09 | Computer Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 12:22:07 | Mathematic       |
|          3 | Rakesh     | Kumar     | 5.6 | 2024-03-20 12:25:20 | Biology          |
|          4 | Radha      | Sharma    | 9.2 | 2024-03-20 12:25:20 | Chemistry        |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 12:25:20 | Physics          |
|          6 | Prem       | Chopra    | 9.56 | 2024-03-20 12:25:20 | History          |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 12:25:20 | English          |
|          8 | Navleen    | Kaur      |    7 | 2024-03-20 12:25:20 | Mathematic       |
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 12:19:09 | Computer Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 12:22:07 | Mathematic       |
|          3 | Rakesh     | Kumar     | 5.6 | 2024-03-20 12:25:20 | Biology          |
|          4 | Radha      | Sharma    | 9.2 | 2024-03-20 12:25:20 | Chemistry        |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 12:25:20 | Physics          |
|          6 | Prem       | Chopra    | 9.56 | 2024-03-20 12:25:20 | History          |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 12:25:20 | English          |
|          8 | Navleen    | Kaur      |    7 | 2024-03-20 12:25:20 | Mathematic       |
```

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

16 rows in set (0.00 sec)

27. Write an SQL query to list STUDENT_ID who does not get Scholarship.

```
Mysql> SELECT STUDENT_ID FROM STUDENT
    -> WHERE STUDENT_ID NOT IN (SELECT STUDENT_ID FROM SCHOLARSHIP);
+------------+
| STUDENT_ID |
+------------+
|          4 |
|          5 |
|          6 |
|          7 |
|          8 |
+------------+
```

5 rows in set (0.01 sec)

28. Write an SQL query to fetch the first 50% records from a table.

```
Mysql> SELECT * FROM STUDENT
    -> WHERE STUDENT_ID <= (SELECT COUNT (STUDENT_ID)/2 FROM
STUDENT);
+------------+------------+-----------+------+--------------------+------------------+
| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE     |
MAJOR           |
+------------+------------+-----------+------+--------------------+------------------+
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 12:19:09 | Computer Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 12:22:07 | Mathematic       |
|          3 | Rakesh     | Kumar     | 5.6 | 2024-03-20 12:25:20 | Biology          |
|          4 | Radha      | Sharma    | 9.2 | 2024-03-20 12:25:20 | Chemistry        |
+------------+------------+-----------+------+--------------------+------------------+
```

4 rows in set (0.00 sec)

29. Write an SQL query to fetch the MAJOR subject that have less than 4 People in it.

```
Mysql> SELECT MAJOR, COUNT (MAJOR) AS MAJOR_COUNT
    -> FROM Student
    -> GROUP BY MAJOR
    -> HAVING COUNT (MAJOR) < 4;
+------------------+-------------+
| MAJOR            | MAJOR_COUNT |
+------------------+-------------+
| Computer Science |           1 |
| Mathematic       |           2 |
| Biology          |           1 |
| Chemistry        |           1 |
| Physics          |           1 |
| History          |           1 |
| English          |           1 |
+------------------+-------------+
7 rows in set (0.00 sec)
```

30. Write an SQL query to show all MAJOR subject along with the number of People in there.

```
Mysql> SELECT MAJOR, COUNT (STUDENT_ID) AS STUDENTS
    -> FROM STUDENT
    -> GROUP BY MAJOR;
+------------------+----------+
| MAJOR            | STUDENTS |
+------------------+----------+
| Computer Science |        1 |
| Mathematic       |        2 |
```

| Biology | 1 |

| Chemistry | 1 |

| Physics | 1 |

| History | 1 |

| English | 1 |

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

7 rows in set (0.00 sec)

31. Write an SQL query to show the last record from a table.

Mysql> SELECT * FROM STUDENT

-> ORDER BY STUDENT_ID DESC

-> LIMIT 1;

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

| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE | MAJOR |

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

| 8 | Navleen | Kaur | 7 | 2024-03-20 12:25:20 | Mathematic |

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

1 row in set (0.00 sec)

32. Write an SQL query to fetch the first row of a table.

Mysql> SELECT * FROM STUDENT

-> ORDER BY STUDENT_ID ASC

-> LIMIT 1;

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

| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE | MAJOR |

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

| 1 | Shivansh | Mahajan | 8.79 | 2024-03-20 12:19:09 | Computer Science |

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

1 row in set (0.00 sec)

33. Write an SQL query to fetch the last five records from a table.

```
Mysql> SELECT * FROM STUDENT
   -> WHERE STUDENT_ID>= (SELECT COUNT (STUDENT_ID)/2 FROM STUDENT);
+------------+------------+-----------+------+---------------------+------------+
| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE     | MAJOR      |
+------------+------------+-----------+------+---------------------+------------+
|          4 | Radha      | Sharma    | 9.2 | 2024-03-20 12:25:20 | Chemistry |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 12:25:20 | Physics    |
|          6 | Prem       | Chopra    | 9.56 | 2024-03-20 12:25:20 | History    |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 12:25:20 | English    |
|          8 | Navleen    | Kaur      |    7 | 2024-03-20 12:25:20 | Mathematic |
+------------+------------+-----------+------+---------------------+------------+
5 rows in set (0.00 sec)
```

34. Write an SQL query to fetch three max GPA from a table using co-related

Subquery.

```
Mysql> SELECT * FROM STUDENT AS S1
   -> WHERE 3>= (SELECT COUNT (DISTINCT GPA) FROM STUDENT S2
   -> WHERE S2.GPA >= S1.GPA)
   -> ORDER BY GPA DESC
   -> LIMIT 3;
+------------+------------+-----------+------+---------------------+-----------+
| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE     | MAJOR     |
+------------+------------+-----------+------+---------------------+-----------+
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 12:25:20 | English   |
|          6 | Prem       | Chopra    | 9.56 | 2024-03-20 12:25:20 | History   |
|          4 | Radha      | Sharma    | 9.2 | 2024-03-20 12:25:20 | Chemistry |
```

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

3 rows in set (0.00 sec)

35. Write an SQL query to fetch three min GPA from a table using co-related Subquery.

```
Mysql> SELECT * FROM STUDENT AS S1
   -> WHERE 3>= (SELECT COUNT (DISTINCT GPA) FROM STUDENT S2
   -> WHERE S2.GPA <= S1.GPA)
   -> ORDER BY GPA ASC
   -> LIMIT 3;
+------------+------------+-----------+------+---------------------+------------+
| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE     | MAJOR      |
+------------+------------+-----------+------+---------------------+------------+
|          3 | Rakesh     | Kumar     | 5.6 | 2024-03-20 12:25:20 | Biology    |
|          8 | Navleen    | Kaur      |   7 | 2024-03-20 12:25:20 | Mathematic |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 12:25:20 | Physics    |
+------------+------------+-----------+------+---------------------+------------+
```

3 rows in set (0.00 sec)

36. Write an SQL query to fetch nth max GPA from a table.

```
Mysql> SELECT CONCAT (FIRST_NAME," ", LAST_NAME), GPA FROM STUDENT
   -> ORDER BY GPA
   -> LIMIT 2, 1;
+---------------------------------+------+
| CONCAT (FIRST_NAME," ", LAST_NAME) | GPA |
+---------------------------------+------+
| Kush Kumar                      | 7.85 |
+---------------------------------+------+
```

1 row in set (0.00 sec)

37. Write an SQL query to fetch MAJOR subjects along with the max GPA in

Each of these MAJOR subjects.

```
MySQL> SELECT DISTINCT (MAJOR), MAX (GPA) FROM STUDENT
    -> GROUP BY MAJOR;
+------------------+----------+
| MAJOR            | MAX (GPA) |
+------------------+----------+
| Computer Science |     8.79 |
| Mathematic       |     8.44 |
| Biology          |      5.6 |
| Chemistry        |      9.2 |
| Physics          |     7.85 |
| History          |     9.56 |
| English          |     9.78 |
+------------------+----------+
7 rows in set (0.00 sec)
```

38. Write an SQL query to fetch the name of Student who has highest GPA.

```
MySQL> SELECT CONCAT (FIRST_NAME," ", LAST_NAME) AS FULL_NAME, GPA
FROM STUDENT
    -> ORDER BY GPA DESC
    -> LIMIT 1;
+-------------+------+
| FULL_NAME   | GPA |
+-------------+------+
| Pankaj Vats | 9.78 |
+-------------+------+
1 row in set (0.00 sec)
```

39. Write an SQL query to show the current date and time.

```
MySQL> SELECT NOW () AS CURRENT_DATE_TIME;
+---------------------+
| CURRENT_DATE_TIME   |
+---------------------+
| 2024-03-21 11:55:40 |
+---------------------+
1 row in set (0.00 sec)
```

40. Write a query to create a new table which consists of data and structure

Copied from the other table (say Student) or clone the table named Student.

```
MySQL> CREATE TABLE CLONE_STUDENT AS SELECT * FROM STUDENT;
Query OK, 8 rows affected (0.17 sec)
Records: 8 Duplicates: 0 Warnings: 0

MySQL> SELECT * FROM CLONE_STUDENT;
+------------+------------+-----------+------+---------------------+------------------+
| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE     | MAJOR            |
+------------+------------+-----------+------+---------------------+------------------+
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 12:19:09 | Computer Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 12:22:07 | Mathematic       |
|          3 | Rakesh     | Kumar     | 5.6 | 2024-03-20 12:25:20 | Biology          |
|          4 | Radha      | Sharma    | 9.2 | 2024-03-20 12:25:20 | Chemistry        |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 12:25:20 | Physics          |
|          6 | Prem       | Chopra    | 9.56 | 2024-03-20 12:25:20 | History          |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 12:25:20 | English          |
|          8 | Navleen    | Kaur      |    7 | 2024-03-20 12:25:20 | Mathematic       |
+------------+------------+-----------+------+---------------------+------------------+
8 rows in set (0.00 sec)
```

41. Write an SQL query to update the GPA of all the students in ‘Computer Science’ MAJOR subject to 7.5.

```
MySQL> UPDATE STUDENT
    -> SET GPA=7.5
    -> WHERE MAJOR = "Computer Science";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MySQL> SELECT * FROM STUDENT;
+------------+------------+-----------+------+---------------------+-----------------+
| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE     | MAJOR           |
+------------+------------+-----------+------+---------------------+-----------------+
|          1 | Shivansh   | Mahajan   | 7.5 | 2024-03-20 12:19:09 | Computer Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 12:22:07 | Mathematic       |
|          3 | Rakesh     | Kumar     | 5.6 | 2024-03-20 12:25:20 | Biology          |
|          4 | Radha      | Sharma    | 9.2 | 2024-03-20 12:25:20 | Chemistry        |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 12:25:20 | Physics          |
|          6 | Prem       | Chopra    | 9.56 | 2024-03-20 12:25:20 | History          |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 12:25:20 | English          |
|          8 | Navleen    | Kaur      | 7 | 2024-03-20 12:25:20 | Mathematic       |
+------------+------------+-----------+------+---------------------+-----------------+
8 rows in set (0.00 sec)
```

42. Write an SQL query to find the average GPA for each major.

```
Mysql> SELECT DISTINCT (MAJOR), AVG (GPA) FROM STUDENT
    -> GROUP BY MAJOR;
+-----------------+------------------+
| MAJOR           | AVG (GPA)        |
+-----------------+------------------+
| Computer Science |              7.5 |
```

| Mathematic | 7.71999979019165 |

| Biology | 5.599999904632568 |

| Chemistry | 9.199999809265137 |

| Physics | 7.849999904632568 |

| History | 9.5600004196167 |

| English | 9.779999732971191 |

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

7 rows in set (0.00 sec)

43. Write an SQL query to show the top 3 students with the highest GPA.

Mysql> SELECT * FROM STUDENT

-> ORDER BY GPA DESC

-> LIMIT 3;

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

| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE | MAJOR |

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

| 7 | Pankaj | Vats | 9.78 | 2024-03-20 12:25:20 | English |

| 6 | Prem | Chopra | 9.56 | 2024-03-20 12:25:20 | History |

| 4 | Radha | Sharma | 9.2 | 2024-03-20 12:25:20 | Chemistry |

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

3 rows in set (0.00 sec)

44. Write an SQL query to find the number of students in each major who have a

GPA greater than 7.5.

Mysql> SELECT MAJOR, COUNT (STUDENT_ID) AS STUDENTS

-> FROM STUDENT

-> WHERE GPA>=7.5

-> GROUP BY MAJOR;

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

| MAJOR | STUDENTS |

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

| Computer Science | 1 |

| Mathematic | 1 |

| Chemistry | 1 |

| Physics | 1 |

| History | 1 |

| English | 1 |

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

6 rows in set (0.00 sec)

45. Write an SQL query to find the students who have the same GPA as

‘Shivansh Mahajan’.

Mysql> SELECT * FROM STUDENT

   -> WHERE GPA=(SELECT GPA FROM STUDENT

   -> WHERE FIRST_NAME="Shivansh" AND LAST_NAME="Mahajan");

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

| STUDENT_ID | FIRST_NAME | LAST_NAME | GPA | ENROLLMENT_DATE | MAJOR |

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

| 1 | Shivansh | Mahajan | 7.5 | 2024-03-20 12:19:09 | Computer Science |

| 8 | Navleen | Kaur | 7.5 | 2024-03-20 12:25:20 | Mathematic |

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

2 rows in set (0.00 sec)