Uptake Database Task

Create Table Student:-

```
MySQL>  INSERT INTO
Student(First_Name,Last_Name,GPA,Enrollment_Date,Major) VALUES
    -> ("Shivansh","Mahajan",8.79,now(),"Computer Science"),
    -> ("Umesh" ,"Sharma" ,8.44 ,now(),"Mathematics"),
    -> ("Rakesh", "Kumar",5.60,now() ,"Biology"),
    -> ("Radha", "Sharma",9.20,now(),"Chemistry"),
    -> ("Kush", "Kumar",7.85,now(),"Physics"),
    -> ("Prem","opra",9.56,now(),"History"),
    -> ("Pankaj","Vats",9.78,now(),"English"),
    -> ("Navleen","Kau",7.00,now(),"Mathematic");
```

```
Query OK, 8 rows affected (0.01 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

OutPut:-

```
mysql> select * from Student;
+------------+------------+-----------+------+---------------------+------------------+
| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     |
Major            |
+------------+------------+-----------+------+---------------------+---
---------------+
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 14:43:37 |
Computer Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 14:43:37 |
Mathematics      |
|          3 | Rakesh     | Kumar     |  5.6 | 2024-03-20 14:43:37 | Biology
|
|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | Chemistry
|
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 14:43:37 | Physics
|
|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History
|
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | English
|
|          8 | Navleen    | Kau       |    7 | 2024-03-20 14:43:37 | Mathematic
|
+------------+------------+-----------+------+---------------------+------------------+
8 rows in set (0.00 sec)
```

Create Table Program:-

```
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 (0.00 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

Output:-

```
MySQL> select * from Program;
+----+------------+------------------+---------------------+
| ID | Student_ID | PROGRAM_NAME     | PROGRAM_START_DATE  |
+----+------------+------------------+---------------------+
|  1 |          1 | Computer Science | 2024-03-20 14:43:55 |
|  2 |          1 | Computer Science | 2024-03-20 14:44:16 |
|  3 |          2 | Mathematics      | 2024-03-20 14:44:16 |
|  4 |          8 | Mathematics      | 2024-03-20 14:44:16 |
|  5 |          5 | Physics          | 2024-03-20 14:44:16 |
|  6 |          4 | Chemistry        | 2024-03-20 14:44:16 |
|  7 |          7 | Psychology       | 2024-03-20 14:44:16 |
|  8 |          6 | History          | 2024-03-20 14:44:16 |
|  9 |          3 | Biology          | 2024-03-20 14:44:16 |
+----+------------+------------------+---------------------+
9 rows in set (0.00 sec)
```

Create table Scholarship:-

```
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 (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

OUTPUT:-

```
mysql> select * from Scholarship;
+----+------------+--------------------+---------------------+
| ID | Student_ID | SCHOLARSHIP_AMOUNT | SCHOLARSHIP_DATE    |
+----+------------+--------------------+---------------------+
|  1 |          1 |               5000 | 2024-03-20 14:50:00 |
|  2 |          2 |               4500 | 2024-03-20 14:50:00 |
|  3 |          3 |               3000 | 2024-03-20 14:50:00 |
|  4 |          1 |               4000 | 2024-03-20 14:50:00 |
+----+------------+--------------------+---------------------+
4 rows in set (0.00 sec)
```

SQL QUERIES:-

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

   OUTPUT:-

   ```
   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.
   OUTPUT:-

```
mysql> SELECT DISTINCT major
    -> FROM student;
+------------------+
| major            |
+------------------+
| Computer Science |
| Mathematics      |
| Biology          |
| Chemistry        |
| Physics          |
| History          |
| English          |
| Mathematic       |
+------------------+
8 rows in set (0.00 sec)
```

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

```
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.

   OUTPUT:-

```
mysql> SELECT INSTR(FIRST_NAME, 'a') FROM Student WHERE FIRST_NAME = 'Shivansh';
+------------------------+
| INSTR(FIRST_NAME, 'a') |
+------------------------+
|                      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.

OUTPUT:-

```
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 |

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

5. rows in set (0.00 sec)

6. Write a SQL query to print FIRST_NAME from the Student table after replacing ‘a’ with ‘A’.
   OUTPUT:-

```
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.
   OUTPUT:-

```
mysql> SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS COMPLETE_NAME FROM Student;

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

| COMPLETE_NAME    |

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

| Shivansh Mahajan |
| Umesh Sharma     |
| Rakesh Kumar     |
| Radha Sharma     |
| Kush Kumar       |
| Prem opra        |
| Pankaj Vats      |
| Navleen Kau      |
+------------------+
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 .
   OUTPUT:-

```
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 14:43:37 | Physics          |
|          8 | Navleen    | Kau       |    7 | 2024-03-20 14:43:37 | Mathematic       |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | English          |
|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History          |
|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | Chemistry        |
|          3 | Rakesh     | Kumar     |  5.6 | 2024-03-20 14:43:37 | Biology          |
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 14:43:37 | Computer Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 14:43:37 | Mathematic
```

8 rows in set (0.0

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

OUTPUT:-

```
mysql> SELECT *
    -> FROM Student
    -> WHERE FIRST_NAME IN ('Prem', 'Shivansh');
+------------+------------+-----------+------+---------------------+------------------+
| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major            |
+------------+------------+-----------+------+---------------------+------------------+
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 14:43:37 | Computer Science |
|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History          |
+------------+------------+-----------+------+---------------------+------------------+
2 rows in set (0.00 sec)
```

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

```
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 14:43:37 | Mathematic |
|          3 | Rakesh     | Kumar     |  5.6 | 2024-03-20 14:43:37 | Biology    |
|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | Chemistry  |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 14:43:37 | Physics    |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | English    |
|          8 | Navleen    | Kau       |    7 | 2024-03-20 14:43:37 | 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’.

OUTPUT:-

```
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 14:43:37 | Chemistry |
+------------+------------+-----------+------+---------------------+-----------+
1 row in set (0.01 sec)
```

12. Write an SQL query to print details of the Students whose FIRST_NAME ends with ‘a’ and contains six alphabets.

OUTPUT:-

```
mysql> SELECT * FROM Student WHERE First_Name LIKE '______%a';
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.

OUTPUT:-

```
mysql> SELECT * From Student WHERE GPA BETWEEN 9 AND 9.99;
+------------+------------+-----------+------+---------------------+-----------+
| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major     |
+------------+------------+-----------+------+---------------------+-----------+
|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | Chemistry |
|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History   |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | 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';
```

OUTPUT:-

```
+----------+
| COUNT(*) |
+----------+
|        1 |
+----------+
1 row in set (0.01 sec)
```

15. Write an SQL query to fetch Students full names with GPA >= 8.5 and <= 9.5.

```
mysql> SELECT * FROM Student WHERE GPA >= 8.5 AND 9.5;
```

OUTPUT:-

```
+------------+------------+-----------+------+---------------------+------------------+
| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major            |
+------------+------------+-----------+------+---------------------+------------------+
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 14:43:37 | Computer Science |
|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | Chemistry        |
|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History          |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | English          |
+------------+------------+-----------+------+---------------------+------------------+
```

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

```
mysql> SELECT major, COUNT(student_id) AS num_students
    -> FROM student
    -> GROUP BY major
    -> ORDER BY num_students DESC;
+------------------+--------------+
| major            | num_students |
+------------------+--------------+
| Mathematic       |            2 |
| Computer Science |            1 |
| Biology          |            1 |
| Chemistry        |            1 |
| Physics          |            1 |
| History          |            1 |
| English          |            1 |
+------------------+--------------+
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 First_Name ,Last_Name, SCHOLARSHIP_AMOUNT , SCHOLARSHIP_DATE From Student INNER JOIN Scholarship ON Student.Student_ID = Scholarsh
ip.Student_ID;
```

OUTPUT:-

```
+------------+-----------+--------------------+---------------------+
| First_Name | Last_Name | SCHOLARSHIP_AMOUNT | SCHOLARSHIP_DATE    |
+------------+-----------+--------------------+---------------------+
| Shivansh   | Mahajan   |               5000 | 2024-03-20 14:50:00 |
| Umesh      | Sharma    |               4500 | 2024-03-20 14:50:00 |
| Rakesh     | Kumar     |               3000 | 2024-03-20 14:50:00 |
| Shivansh   | Mahajan   |               4000 | 2024-03-20 14:50:00 |
+------------+-----------+--------------------+---------------------+
4 rows in set (0.00 sec)
```

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

```
mysql> Select * from Student where (Student_ID % 2) = 1;
```

OUTPUT:-

```
+------------+------------+-----------+------+---------------------+------------------+
| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major            |
+------------+------------+-----------+------+---------------------+------------------+
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 14:43:37 | Computer Science |
|          3 | Rakesh     | Kumar     |  5.6 | 2024-03-20 14:43:37 | Biology          |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 14:43:37 | Physics          |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | English          |

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

4 rows in set (0.01 sec)
```

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

```
mysql> Select * from Student where (Student_ID % 2) = 0;

OUTPUT:-

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

| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major      |

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

|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 14:43:37 | Mathematic |

|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | Chemistry  |

|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History    |

|          8 | Navleen    | Kau       |    7 | 2024-03-20 14:43:37 | 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.

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;
```

OUTPUT:-

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

| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major            |

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

|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | English          |

|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History          |
|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | Chemistry        |
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 14:43:37 | Computer Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 14:43:37 | Mathematic       |
+------------+------------+-----------+------+---------------------+------------------+
5 rows in set (0.00 sec)
```

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

OUTPUT:-

```
mysql> select * from Student ORDER BY GPA LIMIT 4,1;
+------------+------------+-----------+------+---------------------+------------+
| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major      |
+------------+------------+-----------+------+---------------------+------------+
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 14:43:37 | 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;
```

OUTPUT:-

```
+---------+------+
| RANKING | GPA  |
+---------+------+
|       5 | 7.85 |
+---------+------+

1 row in set (0.01 sec)
```

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

```
mysql> SELECT s1. *

    -> FROM Student s1, Student s2

    -> WHERE s1. GPA = s2. GPA

    -> AND s1. Student_id != s2. Student_id;
```

OUTPUT:-

```
Empty set (0.01 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 NOT IN (SELECT MAX(GPA) FROM Student);
```

Second Way:-

```
select *from Student
group by GPA
order by  GPA DESC limit 1,1
```

```
+----------+
| 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
    -> Order By Student_ID;
```

OUTPUT:-

```
+------------+------------+-----------+------+---------------------+------------------+
| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major            |
+------------+------------+-----------+------+---------------------+------------------+
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 14:43:37 | Computer Science |
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 14:43:37 | Computer Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 14:43:37 | Mathematic       |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 14:43:37 | Mathematic       |
|          3 | Rakesh     | Kumar     |  5.6 | 2024-03-20 14:43:37 | Biology          |
|          3 | Rakesh     | Kumar     |  5.6 | 2024-03-20 14:43:37 | Biology          |
|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | Chemistry        |
|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | Chemistry        |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 14:43:37 | Physics          |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 14:43:37 | Physics          |
|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History          |
|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History          |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | English          |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | English          |
|          8 | Navleen    | Kau       |    7 | 2024-03-20 14:43:37 | Mathematic       |
|          8 | Navleen    | Kau       |    7 | 2024-03-20 14:43:37 | 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);
```

OUTPUT:-

```
+------------+
| 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 14:43:37 | Computer Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 14:43:37 | Mathematic       |
|          3 | Rakesh     | Kumar     |  5.6 | 2024-03-20 14:43:37 | Biology          |
|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | 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 ;
OUTPUT:-
+------------------+-------------+
| 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(MAJOR) AS ALL_MAJOR FROM Student GROUP BY MAJOR;
```

OUTPUT:-

```
+------------------+-----------+
| MAJOR            | ALL_MAJOR |
+------------------+-----------+
| Computer Science |         1 |
| Mathematic       |         2 |
| Biology          |         1 |
| Chemistry        |         1 |
| Physics          |         1 |
| History          |         1 |
| English          |         1 |
+------------------+-----------+
7 rows in set (0.01 sec)
```

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

```
mysql> SELECT * FROM Student WHERE Student_ID=(SELECT max(Student_ID) FROM Student);
OUTPUT:-
+------------+------------+-----------+------+---------------------+------------+
| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major      |
+------------+------------+-----------+------+---------------------+------------+
|          8 | Navleen    | Kau       |    7 | 2024-03-20 14:43:37 | 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 WHERE Student_ID=(SELECT min(Student_ID) FROM Student);
OUTPUT:-
+------------+------------+-----------+------+---------------------+------------------+
| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major            |
+------------+------------+-----------+------+---------------------+------------------+
|          1 | Shivansh   | Mahajan   | 8.79 | 2024-03-20 14:43:37 | 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);
```

OUTPUT:-

```
+------------+------------+-----------+------+---------------------+------------+
| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major      |
+------------+------------+-----------+------+---------------------+------------+
|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | Chemistry  |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 14:43:37 | Physics    |
|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History    |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | English    |
|          8 | Navleen    | Kau       |    7 | 2024-03-20 14:43:37 | 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 First_Name,GPA
    -> from Student S1
    -> where 3 >= (SELECT COUNT(DISTINCT GPA)
    -> from Student S2
    ->              WHERE S2.GPA >= S1.GPA)
    -> ORDER BY GPA DESC
    -> LIMIT 3;
```

OUTPUT:-

```
+------------+------+
| First_Name | GPA  |
+------------+------+
| Pankaj     | 9.78 |
| Prem       | 9.56 |
| Radha      |  9.2 |
+------------+------+
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 First_Name,GPA from Student S1 where 3 >= (SELECT COUNT(DISTINCT GPA) from
Student S2 WHERE S2.GPA <= S1.G
PA) ORDER BY GPA DESC LIMIT 3;
```

OUTPUT:- DESC

```
+------------+------+
| First_Name | GPA  |
+------------+------+
| Kush       | 7.85 |
| Navleen    |    7 |
| Rakesh     |  5.6 |
+------------+------+
3 rows in set (0.00 sec)
```

```
mysql> select First_Name,GPA from Student S1 where 3 >= (SELECT COUNT(DISTINCT GPA) from
Student S2 WHERE S2.GPA <= S1.GPA) ORDER BY GPA ASC LIMIT 3;
```

OUTPUT:- Answer of 35 with ASC

```
+------------+------+
| First_Name | GPA  |
+------------+------+
| Rakesh     |  5.6 |
| Navleen    |    7 |
| Kush       | 7.85 |
+------------+------+
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) as FULL_NAME,GPA FROM Student ORDER BY GPA
LIMIT 2,1;
```

OUTPUT:-

```
+-----------------+------+
| FULL_NAME       | GPA  |
+-----------------+------+
| ShivanshMahajan |  7.5 |
+-----------------+------+
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;
```

OUTPUT:-

```
+------------------+----------+
| Major            | MAX(GPA) |
+------------------+----------+
| Computer Science |      7.5 |
| Mathematic       |     8.44 |
| Biology          |      5.6 |
| Chemistry        |      9.2 |
| Physics          |      7.5 |
| History          |     9.56 |
| English          |     9.78 |
+------------------+----------+
7 rows in set (0.00 sec)
```

38. Write an SQL query to fetch the names of Students who has highest GPA.

```
mysql> SELECT FIRST_NAME, GPA
    -> FROM Student
    -> WHERE GPA = (SELECT MAX(GPA) FROM Student);
```

OUTPUT:-

```
+------------+------+
| FIRST_NAME | GPA  |
+------------+------+
| Pankaj     | 9.78 |
+------------+------+
1 row in set (0.01 sec)
```

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

```
mysql> SELECT now();
+---------------------+
| now()               |
+---------------------+
| 2024-03-21 10:50:03 |
+---------------------+
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 (0.05 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   |  7.5 | 2024-03-20 14:43:37 | Computer Science |
|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 14:43:37 | Mathematic       |
|          3 | Rakesh     | Kumar     |  5.6 | 2024-03-20 14:43:37 | Biology          |
|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | Chemistry        |
|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 14:43:37 | Physics          |
|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History          |
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | English          |
|          8 | Navleen    | Kau       |    7 | 2024-03-20 14:43:37 | 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 (0.03 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 14:43:37 | Computer Science |

|          2 | Umesh      | Sharma    | 8.44 | 2024-03-20 14:43:37 | Mathematic       |

|          3 | Rakesh     | Kumar     |  5.6 | 2024-03-20 14:43:37 | Biology          |

|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | Chemistry        |

|          5 | Kush       | Kumar     | 7.85 | 2024-03-20 14:43:37 | Physics          |

|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History          |

|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | English          |

|          8 | Navleen    | Kau       |    7 | 2024-03-20 14:43:37 | Mathematic       |

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

8 rows in set (0.00 sec)
```

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

```
mysql> SELECT MAJOR, AVG(GPA) AS AVERAGE_GPA
    -> FROM STUDENT
    -> GROUP BY MAJOR;
```

OUTPUT:-

```
+------------------+-------------------+
| MAJOR            | AVERAGE_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 (0.01 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;
```

OUTPUT:-

```
+------------+------------+-----------+------+---------------------+-----------+
| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major     |
+------------+------------+-----------+------+---------------------+-----------+
|          7 | Pankaj     | Vats      | 9.78 | 2024-03-20 14:43:37 | English   |
|          6 | Prem       | opra      | 9.56 | 2024-03-20 14:43:37 | History   |
|          4 | Radha      | Sharma    |  9.2 | 2024-03-20 14:43:37 | 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 HIGH_GPA

    -> FROM Student

    -> WHERE GPA > 7.5

    -> GROUP BY MAJOR;
```

OUTPUT:-

```
+------------+----------------+
| MAJOR      | HIGH_GPA |
+------------+----------------+
| Mathematic |            1 |
| Chemistry  |            1 |
| Physics    |            1 |
| History    |            1 |
| English    |            1 |
+------------+----------------+
5 rows in set (0.01 sec)
```

45. Write an SQL query to find the students who have the same GPA as ‘Shivansh Mahajan’.

OUTPUT:-

```
mysql> select * from Student WHERE GPA = (SELECT GPA FROM Student WHERE First_Name = "Shivansh" OR Last_Name = "Mahajan");

+------------+------------+-----------+------+---------------------+------------------+
| Student_ID | First_Name | Last_Name | GPA  | Enrollment_Date     | Major            |
+------------+------------+-----------+------+---------------------+------------------+
|          1 | Shivansh   | Mahajan   |  7.5 | 2024-03-20 14:43:37 | Computer Science |
|          5 | Kush       | Kumar     |  7.5 | 2024-03-20 14:43:37 | Physics          |
+------------+------------+-----------+------+---------------------+------------------+
2 rows in set (0.00 sec)
```