ASSIGNMENT 2 SQL

Task 1. Database Design:

1. **Create the database named "SISDB**

```
mysql> create database SISDB;
Query OK, 1 row affected (0.04 sec)
```

2. Define the schema for the Students, Courses, Enrollments, Teacher, and Payments tables based on the provided schema. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.

a. Students b. Courses c. Enrollments d. Teacher e. Payments

```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| student_id    | int         | NO   | PRI | NULL    |       |
| first_name    | varchar(50) | YES  |     | NULL    |       |
| Last_name     | varchar(50) | YES  |     | NULL    |       |
| date_of_birth | date        | YES  |     | NULL    |       |
| email         | varchar(25) | YES  |     | NULL    |       |
| phone_number  | varchar(20) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)

mysql> desc teachers;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| teacher_id | int         | NO   | PRI | NULL    |       |
| first_name | varchar(50) | YES  |     | NULL    |       |
| last_name  | varchar(50) | YES  |     | NULL    |       |
| email      | varchar(50) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> desc courses;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| course_id   | int         | NO   | PRI | NULL    |       |
| course_name | varchar(50) | YES  |     | NULL    |       |
| credits     | int         | YES  |     | NULL    |       |
| teacher_id  | int         | YES  | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> desc enrollments;
+-----------------+------+------+-----+---------+-------+
| Field           | Type | Null | Key | Default | Extra |
+-----------------+------+------+-----+---------+-------+
| enrollment_id   | int  | NO   | PRI | NULL    |       |
| student_id      | int  | YES  | MUL | NULL    |       |
| course_id       | int  | YES  | MUL | NULL    |       |
| enrollment_date | date | YES  |     | NULL    |       |
+-----------------+------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> desc payments;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| payment_id   | int           | NO   | PRI | NULL    |       |
| student_id   | int           | YES  |     | NULL    |       |
| amount       | decimal(10,2) | YES  |     | NULL    |       |
| payment_date | date          | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

3.Create an ERD (Entity Relationship Diagram) for the databas

4. Create appropriate Primary Key and Foreign Key constraints for referential integrity

```
mysql> create table students(
    -> student_id INT PRIMARY KEY,
    -> first_name VARCHAR(50),
    -> Last_name  VARCHAR(50),
    -> date_of_birth DATE,
    -> email VARCHAR(25),
    -> phone_number VARCHAR(20)
    -> );
```

```
mysql> create table teachers(
    -> teacher_id INT PRIMARY KEY,
    -> first_name VARCHAR(50),
    -> last_name VARCHAR(50),
    -> email VARCHAR(50));
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> create table courses(
    -> course_id INT PRIMARY KEY,
    -> course_name VARCHAR(50),
    -> credits INT,
    -> teacher_id INT,
    -> FOREIGN KEY(teacher_id) REFERENCES teachers(teacher_id)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> create table Enrollments(
    -> enrollment_id INT PRIMARY KEY,
    -> student_id INT,
    -> course_id INT,
    -> enrollment_date DATE,
    -> FOREIGN KEY(student_id) REFERENCE
    -> FOREIGN KEY(course_id) REFERENCES
    -> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> create table payments(
    -> payment_id INT PRIMARY KEY,
    -> student_id INT,
    -> amount DECIMAL(10,2),
    -> payment_date DATE
    -> );
Query OK, 0 rows affected (0.01 sec)
```

5. Insert at least 10 sample records into each of the following tables.

i. Students ii. Courses iii. Enrollment iv. Teacher v. Payments

students table

```
mysql> insert into students values
    ->   (1,"Sourav","kumar",'2000-12-27','srv@gmail.com',11111111),
    ->     (2,"Rishav","kumar",'2000-08-06',"riv@gmail.com",11111112),
    ->     (3,"Subham","gupta",'2000-08-28',"sub@gmail.com",11111113),
    ->     (4,"mayank","mohak",'2000-01-08',"may@gmail.com",11111114),
    ->      (5,"abhishek","jha",'2000-04-12',"abhi@gmail.com",11111115),
    ->     (6,"rahul","kumar",'2000-04-27',"rah@gmail.com",11111116),
    ->      (7,"Satyam","kumar",'2000-09-13',"sat@gmail.com",11111117),
    ->      (8,"rocky","gupta",'2000-01-18',"rok@gmail.com",11111118),
    ->       (9,"purusottam","kumar",'2000-01-09',"puru@gmail.com",11111119),
    ->       (10,"anurag","singh",'2000-10-26',"anu@gmail.com",111111110);
Query OK, 10 rows affected (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

ASSIGNMENT 2 SQL

```
mysql> select *from students;
+------------+------------+-----------+--------------+----------------+--------------+
| student_id | first_name | Last_name | date_of_birth | email          | phone_number |
+------------+------------+-----------+--------------+----------------+--------------+
|          1 | Sourav     | kumar     | 2000-12-27    | srv@gmail.com  | 11111111     |
|          2 | Rishav     | kumar     | 2000-08-06    | riv@gmail.com  | 11111112     |
|          3 | Subham     | gupta     | 2000-08-28    | sub@gmail.com  | 11111113     |
|          4 | mayank     | mohak     | 2000-01-08    | may@gmail.com  | 11111114     |
|          5 | abhishek   | jha       | 2000-04-12    | abhi@gmail.com | 11111115     |
|          6 | rahul      | kumar     | 2000-04-27    | rah@gmail.com  | 11111116     |
|          7 | Satyam     | kumar     | 2000-09-13    | sat@gmail.com  | 11111117     |
|          8 | rocky      | gupta     | 2000-01-18    | rok@gmail.com  | 11111118     |
|          9 | purusottam | kumar     | 2000-01-09    | puru@gmail.com | 11111119     |
|         10 | anurag     | singh     | 2000-10-26    | anu@gmail.com  | 111111110    |
+------------+------------+-----------+--------------+----------------+--------------+
10 rows in set (0.00 sec)
```

Teachers table

```
mysql>  insert into teachers values(
    -> 001,"ram","singh","ram@giet.edu"),
    -> (002,"sam","ji","sam@giet.edu"),
    -> (003,"tam","yadav","tam@giet.edu"),
    -> (004,"jam","kumar","jam@giet.edu"),
    ->  (005,"aam","nayak","aam@giet.edu"),
    -> (006,"bam","sahu","bam@giet.edu"),
    ->  (007,"cam","mahanty","cam@giet.edu"),
    ->  (008,"dam","pandit","dam@giet.edu"),
    -> (009,"eam","singh","eam@giet.edu"),
    ->  (010,"sourav","aahir","fam@giet.edu");
Query OK, 10 rows affected (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from teachers;
+------------+------------+-----------+--------------+
| teacher_id | first_name | last_name | email        |
+------------+------------+-----------+--------------+
|          1 | ram        | singh     | ram@giet.edu |
|          2 | sam        | ji        | sam@giet.edu |
|          3 | tam        | yadav     | tam@giet.edu |
|          4 | jam        | kumar     | jam@giet.edu |
|          5 | aam        | nayak     | aam@giet.edu |
|          6 | bam        | sahu      | bam@giet.edu |
|          7 | cam        | mahanty   | cam@giet.edu |
|          8 | dam        | pandit    | dam@giet.edu |
|          9 | eam        | singh     | eam@giet.edu |
|         10 | sourav     | aahir     | fam@giet.edu |
+------------+------------+-----------+--------------+
10 rows in set (0.00 sec)
```

ASSIGNMENT 2 SQL

Course table

```
mysql> insert into courses values(
    ->  1,"physics",4,001),
    ->  (2,"chemistry",3,002),
    ->  (3,"biology",4,003),
    -> (4,"mathematics",5,004),
    -> (5,"history",4,005),
    ->  (6,"civics",5,006),
    ->  (7,"economics",2,007),
    ->  (8,"geography",3,008),
    ->  (9,"computer",5,009),
    ->  (10,"physical edu",4,010);
Query OK, 10 rows affected (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select *from courses;
+-----------+--------------+---------+------------+
| course_id | course_name  | credits | teacher_id |
+-----------+--------------+---------+------------+
|         1 | physics      |       4 |          1 |
|         2 | chemistry    |       3 |          2 |
|         3 | biology      |       4 |          3 |
|         4 | mathematics  |       5 |          4 |
|         5 | history      |       4 |          5 |
|         6 | civics       |       5 |          6 |
|         7 | economics    |       2 |          7 |
|         8 | geography    |       3 |          8 |
|         9 | computer     |       5 |          9 |
|        10 | physical edu |       4 |         10 |
+-----------+--------------+---------+------------+
10 rows in set (0.00 sec)
```

Enrollments table;-

```
mysql> insert into enrollments values
    -> (01,1,1,'2024-01-10'),
    ->  (02,2,2,'2024-01-11'),
    ->  (03,3,3,'2024-01-12'),
    ->  (04,4,4,'2024-01-13'),
    ->  (05,5,5,'2024-01-14'),
    ->  (06,6,6,'2024-01-15'),
    ->  (07,7,7,'2024-01-16'),
    ->  (08,8,8,'2024-01-17'),
    ->  (09,9,9,'2024-01-18'),
    ->  (10,10,10,'2024-01-20');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

ASSIGNMENT 2 SQL

```
mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2024-01-10      |
|             2 |          2 |         2 | 2024-01-11      |
|             3 |          3 |         3 | 2024-01-12      |
|             4 |          4 |         4 | 2024-01-13      |
|             5 |          5 |         5 | 2024-01-14      |
|             6 |          6 |         6 | 2024-01-15      |
|             7 |          7 |         7 | 2024-01-16      |
|             8 |          8 |         8 | 2024-01-17      |
|             9 |          9 |         9 | 2024-01-18      |
|            10 |         10 |        10 | 2024-01-20      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)
```

Payments table:-

```
mysql> insert into payments values
    -> (5,1,50,'2024-01-05'),
    ->  (10,2,55,'2024-01-10'),
    ->  (15,3,60,'2024-01-15'),
    ->  (16,4,65,'2024-01-20'),
    ->  (20,5,70,'2024-01-25'),
    ->  (21,6,75,'2024-01-30'),
    ->  (22,7,80,'2024-02-02'),
    ->  (24,8,90,'2024-02-05'),
    ->  (17,9,95,'2024-02-09'),
    ->  (29,10,100,'2024-02-10');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          5 |          1 |  50.00 | 2024-01-05   |
|         10 |          2 |  55.00 | 2024-01-10   |
|         15 |          3 |  60.00 | 2024-01-15   |
|         16 |          4 |  65.00 | 2024-01-20   |
|         17 |          9 |  95.00 | 2024-02-09   |
|         20 |          5 |  70.00 | 2024-01-25   |
|         21 |          6 |  75.00 | 2024-01-30   |
|         22 |          7 |  80.00 | 2024-02-02   |
|         24 |          8 |  90.00 | 2024-02-05   |
|         29 |         10 | 100.00 | 2024-02-10   |
+------------+------------+--------+--------------+
10 rows in set (0.00 sec)
```

ASSIGNMENT 2 SQL

Tasks 2: Select, Where, Between, AND, LIKE:

1. **Write an SQL query to insert a new student into the "Students" table with the following details: a. First Name: John b. Last Name: Doe c. Date of Birth: 1995-08-15 d. Email: john.doe@example.com e. Phone Number: 1234567890**

```
mysql> insert into students(student_id,first_name,last_name,date_of_birth,email)
    ->  values(11,'John','Doe','1995-08-15','john.doe@example.com');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select *from students;
+------------+------------+-----------+---------------+---------------------+--------------+
| student_id | first_name | Last_name | date_of_birth | email               | phone_number |
+------------+------------+-----------+---------------+---------------------+--------------+
|          1 | Sourav     | kumar     | 2000-12-27    | srv@gmail.com       | 11111111     |
|          2 | Rishav     | kumar     | 2000-08-06    | riv@gmail.com       | 11111112     |
|          3 | Subham     | gupta     | 2000-08-28    | sub@gmail.com       | 11111113     |
|          4 | mayank     | mohak     | 2000-01-08    | may@gmail.com       | 11111114     |
|          5 | abhishek   | jha       | 2000-04-12    | abhi@gmail.com      | 11111115     |
|          6 | rahul      | kumar     | 2000-04-27    | rah@gmail.com       | 11111116     |
|          7 | Satyam     | kumar     | 2000-09-13    | sat@gmail.com       | 11111117     |
|          8 | rocky      | gupta     | 2000-01-18    | rok@gmail.com       | 11111118     |
|          9 | purusottam | kumar     | 2000-01-09    | puru@gmail.com      | 11111119     |
|         10 | anurag     | singh     | 2000-10-26    | anu@gmail.com       | 111111110    |
|         11 | John       | Doe       | 1995-08-15    | john.doe@example.com | NULL         |
+------------+------------+-----------+---------------+---------------------+--------------+
11 rows in set (0.00 sec)

mysql>
```

2. **Write an SQL query to enroll a student in a course. Choose an existing student and course and insert a record into the "Enrollments" table with the enrollment date**

```
mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2024-01-10      |
|             2 |          2 |         2 | 2024-01-11      |
|             3 |          3 |         3 | 2024-01-12      |
|             4 |          4 |         4 | 2024-01-13      |
|             5 |          5 |         5 | 2024-01-14      |
|             6 |          6 |         6 | 2024-01-15      |
|             7 |          7 |         7 | 2024-01-16      |
|             8 |          8 |         8 | 2024-01-17      |
|            11 |          9 |         9 | 2024-01-18      |
+---------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)

mysql> INSERT INTO Enrollments (student_id, course_id, enrollment_date)
    -> VALUES ('1', '4', '2024-01-19');
Query OK, 1 row affected (0.00 sec)

mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2024-01-10      |
|             2 |          2 |         2 | 2024-01-11      |
|             3 |          3 |         3 | 2024-01-12      |
|             4 |          4 |         4 | 2024-01-13      |
|             5 |          5 |         5 | 2024-01-14      |
|             6 |          6 |         6 | 2024-01-15      |
|             7 |          7 |         7 | 2024-01-16      |
|             8 |          8 |         8 | 2024-01-17      |
|            11 |          9 |         9 | 2024-01-18      |
|            12 |          1 |         4 | 2024-01-19      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)
```

3. **Update the email address of a specific teacher in the "Teacher" table. Choose any teacher and modify their email address.**

```
mysql> select * from teachers;
+------------+------------+-----------+--------------+
| teacher_id | first_name | last_name | email        |
+------------+------------+-----------+--------------+
|          1 | ram        | singh     | ram@giet.edu |
|          2 | sam        | ji        | sam@giet.edu |
|          3 | tam        | yadav     | tam@giet.edu |
|          4 | jam        | kumar     | jam@giet.edu |
|          5 | aam        | nayak     | aam@giet.edu |
|          6 | bam        | sahu      | bam@giet.edu |
|          7 | cam        | mahanty   | cam@giet.edu |
|          8 | dam        | pandit    | dam@giet.edu |
|          9 | eam        | singh     | eam@giet.edu |
|         10 | sourav     | aahir     | fam@giet.edu |
+------------+------------+-----------+--------------+
10 rows in set (0.00 sec)
```

```
mysql> update teachers
    -> set email='ramayan@gmail.com'
    -> where teacher_id=1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from teachers;
+------------+------------+-----------+-------------------+
| teacher_id | first_name | last_name | email             |
+------------+------------+-----------+-------------------+
|          1 | ram        | singh     | ramayan@gmail.com |
|          2 | sam        | ji        | sam@giet.edu      |
|          3 | tam        | yadav     | tam@giet.edu      |
|          4 | jam        | kumar     | jam@giet.edu      |
|          5 | aam        | nayak     | aam@giet.edu      |
|          6 | bam        | sahu      | bam@giet.edu      |
|          7 | cam        | mahanty   | cam@giet.edu      |
|          8 | dam        | pandit    | dam@giet.edu      |
|          9 | eam        | singh     | eam@giet.edu      |
|         10 | sourav     | aahir     | fam@giet.edu      |
+------------+------------+-----------+-------------------+
10 rows in set (0.00 sec)
```

4.Write an SQL query to delete a specific enrollment record from the "Enrollments" table. Select an enrollment record based on the student and course

ASSIGNMENT 2 SQL

```
mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2024-01-10      |
|             2 |          2 |         2 | 2024-01-11      |
|             3 |          3 |         3 | 2024-01-12      |
|             4 |          4 |         4 | 2024-01-13      |
|             5 |          5 |         5 | 2024-01-14      |
|             6 |          6 |         6 | 2024-01-15      |
|             7 |          7 |         7 | 2024-01-16      |
|             8 |          8 |         8 | 2024-01-17      |
|             9 |          9 |         9 | 2024-01-18      |
|            10 |         10 |        10 | 2024-01-20      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)
```

```
mysql> delete from enrollments
    -> where student_id=10 and course_id=10;
Query OK, 1 row affected (0.00 sec)

mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2024-01-10      |
|             2 |          2 |         2 | 2024-01-11      |
|             3 |          3 |         3 | 2024-01-12      |
|             4 |          4 |         4 | 2024-01-13      |
|             5 |          5 |         5 | 2024-01-14      |
|             6 |          6 |         6 | 2024-01-15      |
|             7 |          7 |         7 | 2024-01-16      |
|             8 |          8 |         8 | 2024-01-17      |
|             9 |          9 |         9 | 2024-01-18      |
+---------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)
```

5.Update the "Courses" table to assign a specific teacher to a course. Choose any course and teacher from the respective tables

```
mysql> select * from courses;
+-----------+--------------+---------+------------+
| course_id | course_name  | credits | teacher_id |
+-----------+--------------+---------+------------+
|         1 | physics      |       4 |          1 |
|         2 | chemistry    |       3 |          2 |
|         3 | biology      |       4 |          3 |
|         4 | mathematics  |       5 |          4 |
|         5 | history      |       4 |          5 |
|         6 | civics       |       5 |          6 |
|         7 | economics    |       2 |          7 |
|         8 | geography    |       3 |          8 |
|         9 | computer     |       5 |          9 |
|        10 | physical edu |       4 |         10 |
+-----------+--------------+---------+------------+
10 rows in set (0.00 sec)
```

```
mysql> update courses
    -> set course_name='dbms'
    -> where teacher_id=10;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from courses;
+-----------+-------------+---------+------------+
| course_id | course_name | credits | teacher_id |
+-----------+-------------+---------+------------+
|         1 | physics     |       4 |          1 |
|         2 | chemistry   |       3 |          2 |
|         3 | biology     |       4 |          3 |
|         4 | mathematics |       5 |          4 |
|         5 | history     |       4 |          5 |
|         6 | civics      |       5 |          6 |
|         7 | economics   |       2 |          7 |
|         8 | geography   |       3 |          8 |
|         9 | computer    |       5 |          9 |
|        10 | dbms        |       4 |         10 |
+-----------+-------------+---------+------------+
10 rows in set (0.00 sec)
```

6. Delete a specific student from the "Students" table and remove all their enrollment records from the "Enrollments" table. Be sure to maintain referential integrity.

```
+------------+------------+-----------+---------------+---------------------+--------------+
| student_id | first_name | Last_name | date_of_birth | email               | phone_number |
+------------+------------+-----------+---------------+---------------------+--------------+
|          1 | Sourav     | kumar     | 2000-12-27    | srv@gmail.com       | 11111111     |
|          2 | Rishav     | kumar     | 2000-08-06    | riv@gmail.com       | 11111112     |
|          3 | Subham     | gupta     | 2000-08-28    | sub@gmail.com       | 11111113     |
|          4 | mayank     | mohak     | 2000-01-08    | may@gmail.com       | 11111114     |
|          5 | abhishek   | jha       | 2000-04-12    | abhi@gmail.com      | 11111115     |
|          6 | rahul      | kumar     | 2000-04-27    | rah@gmail.com       | 11111116     |
|          7 | Satyam     | kumar     | 2000-09-13    | sat@gmail.com       | 11111117     |
|          8 | rocky      | gupta     | 2000-01-18    | rok@gmail.com       | 11111118     |
|          9 | purusottam | kumar     | 2000-01-09    | puru@gmail.com      | 11111119     |
|         10 | anurag     | singh     | 2000-10-26    | anu@gmail.com       | 111111110    |
|         11 | John       | Doe       | 1995-08-15    | john.doe@example.com | NULL         |
+------------+------------+-----------+---------------+---------------------+--------------+
```

```
mysql> delete from enrollments
    -> where student_id=11;
Query OK, 0 rows affected (0.00 sec)

mysql> delete from students
    -> where student_id=11;
Query OK, 1 row affected (0.00 sec)
```

ASSIGNMENT 2 SQL

```
mysql> select* from students;
+------------+------------+-----------+--------------+---------------+--------------+
| student_id | first_name | Last_name | date_of_birth | email         | phone_number |
+------------+------------+-----------+--------------+---------------+--------------+
|          1 | Sourav     | kumar     | 2000-12-27    | srv@gmail.com | 11111111     |
|          2 | Rishav     | kumar     | 2000-08-06    | riv@gmail.com | 11111112     |
|          3 | Subham     | gupta     | 2000-08-28    | sub@gmail.com | 11111113     |
|          4 | mayank     | mohak     | 2000-01-08    | may@gmail.com | 11111114     |
|          5 | abhishek   | jha       | 2000-04-12    | abhi@gmail.com| 11111115     |
|          6 | rahul      | kumar     | 2000-04-27    | rah@gmail.com | 11111116     |
|          7 | Satyam     | kumar     | 2000-09-13    | sat@gmail.com | 11111117     |
|          8 | rocky      | gupta     | 2000-01-18    | rok@gmail.com | 11111118     |
|          9 | purusottam | kumar     | 2000-01-09    | puru@gmail.com| 11111119     |
|         10 | anurag     | singh     | 2000-10-26    | anu@gmail.com | 111111110    |
+------------+------------+-----------+--------------+---------------+--------------+
10 rows in set (0.00 sec)
```

```
mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2024-01-10      |
|             2 |          2 |         2 | 2024-01-11      |
|             3 |          3 |         3 | 2024-01-12      |
|             4 |          4 |         4 | 2024-01-13      |
|             5 |          5 |         5 | 2024-01-14      |
|             6 |          6 |         6 | 2024-01-15      |
|             7 |          7 |         7 | 2024-01-16      |
|             8 |          8 |         8 | 2024-01-17      |
|            11 |          9 |         9 | 2024-01-18      |
+---------------+------------+-----------+-----------------+
```

7. Update the payment amount for a specific payment record in the "Payments" table. Choose any payment record and modify the payment amount.

```
mysql> select * from payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          5 |          1 |  50.00 | 2024-01-05   |
|         10 |          2 |  55.00 | 2024-01-10   |
|         15 |          3 |  60.00 | 2024-01-15   |
|         16 |          4 |  65.00 | 2024-01-20   |
|         17 |          9 |  95.00 | 2024-02-09   |
|         20 |          5 |  70.00 | 2024-01-25   |
|         21 |          6 |  75.00 | 2024-01-30   |
|         22 |          7 |  80.00 | 2024-02-02   |
|         24 |          8 |  90.00 | 2024-02-05   |
|         29 |         10 | 100.00 | 2024-02-10   |
+------------+------------+--------+--------------+
10 rows in set (0.00 sec)
```

```
mysql> update payments
    -> set amount=999
    -> where payment_id=24;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          5 |          1 |  50.00 | 2024-01-05   |
|         10 |          2 |  55.00 | 2024-01-10   |
|         15 |          3 |  60.00 | 2024-01-15   |
|         16 |          4 |  65.00 | 2024-01-20   |
|         17 |          9 |  95.00 | 2024-02-09   |
|         20 |          5 |  70.00 | 2024-01-25   |
|         21 |          6 |  75.00 | 2024-01-30   |
|         22 |          7 |  80.00 | 2024-02-02   |
|         24 |          8 | 999.00 | 2024-02-05   |
|         29 |         10 | 100.00 | 2024-02-10   |
+------------+------------+--------+--------------+
10 rows in set (0.00 sec)
```

Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

1. **Write an SQL query to calculate the total payments made by a specific student. You will need to join the "Payments" table with the "Students" table based on the student's ID**

```
mysql> select * from payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          5 |          1 |  50.00 | 2024-01-05   |
|         10 |          2 |  55.00 | 2024-01-10   |
|         15 |          3 |  60.00 | 2024-01-15   |
|         16 |          4 |  65.00 | 2024-01-20   |
|         17 |          9 |  95.00 | 2024-02-09   |
|         20 |          5 |  70.00 | 2024-01-25   |
|         21 |          6 |  75.00 | 2024-01-30   |
|         22 |          7 |  80.00 | 2024-02-02   |
|         24 |          8 | 999.00 | 2024-02-05   |
|         29 |         10 | 100.00 | 2024-02-10   |
|         30 |          1 | 100.00 | 2024-01-19   |
|         31 |          1 | 100.00 | 2024-01-19   |
|         32 |          1 | 350.00 | 2024-02-12   |
+------------+------------+--------+--------------+
13 rows in set (0.00 sec)
```

```
mysql>  select students.student_id,students.first_name,SUM(payments.amount)
AS total_payments from students
    ->  JOIN payments ON students.student_id=payments.student_id
    ->  where students.student_id=1
    ->  group by students.student_id,students.first_name;
+------------+------------+----------------+
| student_id | first_name | total_payments |
+------------+------------+----------------+
|          1 | Sourav     |         600.00 |
+------------+------------+----------------+
1 row in set (0.00 sec)
```

ASSIGNMENT 2 SQL

2. **Write an SQL query to retrieve a list of courses along with the count of students enrolled in each course. Use a JOIN operation between the "Courses" table and the "Enrollments" table.**

```
mysql> select * from courses;
+-----------+-------------+---------+------------+
| course_id | course_name | credits | teacher_id |
+-----------+-------------+---------+------------+
|         1 | physics     |       4 |          1 |
|         2 | chemistry   |       3 |          2 |
|         3 | biology     |       4 |          3 |
|         4 | mathematics |       5 |          4 |
|         5 | history     |       4 |          5 |
|         6 | civics      |       5 |          6 |
|         7 | economics   |       2 |          7 |
|         8 | geography   |       3 |          8 |
|         9 | computer    |       5 |          9 |
|        10 | dbms        |       4 |         10 |
+-----------+-------------+---------+------------+
10 rows in set (0.00 sec)
```

```
mysql> select* from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2024-01-10      |
|             2 |          2 |         2 | 2024-01-11      |
|             3 |          3 |         3 | 2024-01-12      |
|             4 |          4 |         4 | 2024-01-13      |
|             5 |          5 |         5 | 2024-01-14      |
|             6 |          6 |         6 | 2024-01-15      |
|             7 |          7 |         7 | 2024-01-16      |
|             8 |          8 |         8 | 2024-01-17      |
|            11 |          9 |         9 | 2024-01-18      |
|            12 |          1 |         4 | 2024-01-19      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)
```

```
mysql> select courses.course_id,courses.course_name,COUNT(enrollments.student_id) AS enroll_studden_count from courses
    -> JOIN enrollments ON courses.course_id=enrollments.course_id
    -> group by courses.course_id,courses.course_name;
+-----------+-------------+---------------------+
| course_id | course_name | enroll_studden_count |
+-----------+-------------+---------------------+
|         1 | physics     |                   1 |
|         2 | chemistry   |                   1 |
|         3 | biology     |                   1 |
|         4 | mathematics |                   2 |
|         5 | history     |                   1 |
|         6 | civics      |                   1 |
|         7 | economics   |                   1 |
|         8 | geography   |                   1 |
|         9 | computer    |                   1 |
+-----------+-------------+---------------------+
9 rows in set (0.00 sec)
```

ASSIGNMENT 2 SQL

3. **Write an SQL query to find the names of students who have not enrolled in any course. Use a LEFT JOIN between the "Students" table and the "Enrollments" table to identify students without enrollments**

```
mysql> select * from students;
+------------+------------+-----------+---------------+----------------+--------------+
| student_id | first_name | Last_name | date_of_birth | email          | phone_number |
+------------+------------+-----------+---------------+----------------+--------------+
|          1 | Sourav     | kumar     | 2000-12-27    | srv@gmail.com  | 11111111     |
|          2 | Rishav     | kumar     | 2000-08-06    | riv@gmail.com  | 11111112     |
|          3 | Subham     | gupta     | 2000-08-28    | sub@gmail.com  | 11111113     |
|          4 | mayank     | mohak     | 2000-01-08    | may@gmail.com  | 11111114     |
|          5 | abhishek   | jha       | 2000-04-12    | abhi@gmail.com | 11111115     |
|          6 | rahul      | kumar     | 2000-04-27    | rah@gmail.com  | 11111116     |
|          7 | Satyam     | kumar     | 2000-09-13    | sat@gmail.com  | 11111117     |
|          8 | rocky      | gupta     | 2000-01-18    | rok@gmail.com  | 11111118     |
|          9 | purusottam | kumar     | 2000-01-09    | puru@gmail.com | 11111119     |
|         10 | anurag     | singh     | 2000-10-26    | anu@gmail.com  | 111111110    |
+------------+------------+-----------+---------------+----------------+--------------+
10 rows in set (0.00 sec)
```

```
mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2024-01-10      |
|             2 |          2 |         2 | 2024-01-11      |
|             3 |          3 |         3 | 2024-01-12      |
|             4 |          4 |         4 | 2024-01-13      |
|             5 |          5 |         5 | 2024-01-14      |
|             6 |          6 |         6 | 2024-01-15      |
|             7 |          7 |         7 | 2024-01-16      |
|             8 |          8 |         8 | 2024-01-17      |
|            11 |          9 |         9 | 2024-01-18      |
|            12 |          1 |         4 | 2024-01-19      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)
```

```
mysql> select students.student_id,students.first_name from students
    -> LEFT JOIN enrollments ON  students.student_id=enrollments.student_id
    -> where enrollments.student_id IS NULL;
+------------+------------+
| student_id | first_name |
+------------+------------+
|         10 | anurag     |
+------------+------------+
1 row in set (0.00 sec)
```

4. **Write an SQL query to retrieve the first name, last name of students, and the names of the courses they are enrolled in. Use JOIN operations between the "Students" table and the "Enrollments" and "Courses" tables.**

```
mysql> select students.student_id,students.first_name,students.last_name,courses.course_name from students
    -> JOIN enrollments ON students.student_id=enrollments.student_id
    -> JOIN courses ON enrollments.course_id=courses.course_id;
+------------+------------+-----------+-------------+
| student_id | first_name | last_name | course_name |
+------------+------------+-----------+-------------+
|          1 | Sourav     | kumar     | physics     |
|          2 | Rishav     | kumar     | chemistry   |
|          3 | Subham     | gupta     | biology     |
|          4 | mayank     | mohak     | mathematics |
|          1 | Sourav     | kumar     | mathematics |
|          5 | abhishek   | jha       | history     |
|          6 | rahul      | kumar     | civics      |
|          7 | Satyam     | kumar     | economics   |
|          8 | rocky      | gupta     | geography   |
|          9 | purusottam | kumar     | computer    |
+------------+------------+-----------+-------------+
10 rows in set (0.00 sec)
```

5. **Create a query to list the names of teachers and the courses they are assigned to. Join the "Teacher" table with the "Courses" table.**

```
mysql> select courses.course_id,courses.course_name,teachers.first_name from courses
    -> JOIN teachers ON courses.teacher_id=teachers.teacher_id;
+-----------+-------------+------------+
| course_id | course_name | first_name |
+-----------+-------------+------------+
|         1 | physics     | ram        |
|         2 | chemistry   | sam        |
|         3 | biology     | tam        |
|         4 | mathematics | jam        |
|         5 | history     | aam        |
|         6 | civics      | bam        |
|         7 | economics   | cam        |
|         8 | geography   | dam        |
|         9 | computer    | eam        |
|        10 | dbms        | sourav     |
+-----------+-------------+------------+
10 rows in set (0.00 sec)
```

6. **Retrieve a list of students and their enrollment dates for a specific course. You'll need to join the "Students" table with the "Enrollments" and "Courses" tables.**

```
mysql> select students.first_name,enrollments.enrollment_date,courses.course_name from students
    -> JOIN enrollments ON students.student_id=enrollments.student_id
    -> JOIN courses ON enrollments.course_id=courses.course_id;
+------------+-----------------+-------------+
| first_name | enrollment_date | course_name |
+------------+-----------------+-------------+
| Sourav     | 2024-01-10      | physics     |
| Rishav     | 2024-01-11      | chemistry   |
| Subham     | 2024-01-12      | biology     |
| mayank     | 2024-01-13      | mathematics |
| abhishek   | 2024-01-14      | history     |
| rahul      | 2024-01-15      | civics      |
| Satyam     | 2024-01-16      | economics   |
| rocky      | 2024-01-17      | geography   |
| purusottam | 2024-01-18      | computer    |
| Sourav     | 2024-01-19      | mathematics |
+------------+-----------------+-------------+
10 rows in set (0.00 sec)
```

7. **Find the names of students who have not made any payments. Use a LEFT JOIN between the "Students" table and the "Payments" table and filter for students with NULL payment records.**

```
mysql> select * from payments;
+------------+------------+---------+--------------+
| payment_id | student_id | amount  | payment_date |
+------------+------------+---------+--------------+
|          5 |          1 |   50.00 | 2024-01-05   |
|         10 |          2 |   55.00 | 2024-01-10   |
|         15 |          3 |   60.00 | 2024-01-15   |
|         16 |          4 |   65.00 | 2024-01-20   |
|         17 |          9 |   95.00 | 2024-02-09   |
|         20 |          5 |   70.00 | 2024-01-25   |
|         21 |          6 |   75.00 | 2024-01-30   |
|         22 |          7 |   80.00 | 2024-02-02   |
|         24 |          8 |  999.00 | 2024-02-05   |
|         29 |         10 |  100.00 | 2024-02-10   |
|         30 |          1 |  100.00 | 2024-01-19   |
|         31 |          1 |  100.00 | 2024-01-19   |
|         32 |          1 |  350.00 | 2024-02-12   |
+------------+------------+---------+--------------+
13 rows in set (0.00 sec)
```

```
mysql> select students.student_id,students.first_name from students
    -> LEFT JOIN payments ON students.student_id=payments.student_id
    -> where payments.amount IS NULL;
+------------+------------+
| student_id | first_name |
+------------+------------+
|         11 | sanjeev    |
+------------+------------+
1 row in set (0.00 sec)
```

ASSIGNMENT 2 SQL

8. **Write a query to identify courses that have no enrollments. You'll need to use a LEFT JOIN between the "Courses" table and the "Enrollments" table and filter for courses with NULL enrollment records**

```
mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2024-01-10      |
|             2 |          2 |         2 | 2024-01-11      |
|             3 |          3 |         3 | 2024-01-12      |
|             4 |          4 |         4 | 2024-01-13      |
|             5 |          5 |         5 | 2024-01-14      |
|             6 |          6 |         6 | 2024-01-15      |
|             7 |          7 |         7 | 2024-01-16      |
|             8 |          8 |         8 | 2024-01-17      |
|            11 |          9 |         9 | 2024-01-18      |
|            12 |          1 |         4 | 2024-01-19      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)
```

```
mysql>  select courses.course_id,courses.course_name from courses
    -> LEFT JOIN enrollments ON courses.course_id=enrollments.course_id
    -> where
    -> courses.course_id NOT IN(select course_id from enrollments);
+-----------+-------------+
| course_id | course_name |
+-----------+-------------+
|        10 | dbms        |
+-----------+-------------+
1 row in set (0.00 sec)
```

9. **Identify students who are enrolled in more than one course. Use a self-join on the "Enrollments" table to find students with multiple enrollment records.**

```
mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2024-01-10      |
|             2 |          2 |         2 | 2024-01-11      |
|             3 |          3 |         3 | 2024-01-12      |
|             4 |          4 |         4 | 2024-01-13      |
|             5 |          5 |         5 | 2024-01-14      |
|             6 |          6 |         6 | 2024-01-15      |
|             7 |          7 |         7 | 2024-01-16      |
|             8 |          8 |         8 | 2024-01-17      |
|            11 |          9 |         9 | 2024-01-18      |
|            12 |          1 |         4 | 2024-01-19      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)
```

```
mysql> SELECT
    ->     e1.student_id,
    ->     s.first_name,
    ->     s.last_name,
    ->     COUNT(e1.course_id) AS enrolled_courses_count
    -> FROM
    ->     Enrollments e1
    -> JOIN
    ->     Enrollments e2 ON e1.student_id = e2.student_id AND e1.course_id <> e2.course_id
    -> JOIN
    ->     Students s ON e1.student_id = s.student_id
    -> GROUP BY
    ->     e1.student_id, s.first_name, s.last_name
    -> HAVING
    ->     COUNT(e1.course_id) > 1;
+------------+------------+-----------+-----------------------+
| student_id | first_name | last_name | enrolled_courses_count |
+------------+------------+-----------+-----------------------+
|          1 | Sourav     | kumar     |                     2 |
+------------+------------+-----------+-----------------------+
1 row in set (0.00 sec)
```

10. **Find teachers who are not assigned to any courses. Use a LEFT JOIN between the "Teacher" table and the "Courses" table and filter for teachers with NULL course assignments.**

```
mysql> select * from teachers;
+------------+------------+-----------+-------------------+
| teacher_id | first_name | last_name | email             |
+------------+------------+-----------+-------------------+
|          1 | ram        | singh     | ramayan@gmail.com |
|          2 | sam        | ji        | sam@giet.edu      |
|          3 | tam        | yadav     | tam@giet.edu      |
|          4 | jam        | kumar     | jam@giet.edu      |
|          5 | aam        | nayak     | aam@giet.edu      |
|          6 | bam        | sahu      | bam@giet.edu      |
|          7 | cam        | mahanty   | cam@giet.edu      |
|          8 | dam        | pandit    | dam@giet.edu      |
|          9 | eam        | singh     | eam@giet.edu      |
|         10 | sourav     | aahir     | fam@giet.edu      |
+------------+------------+-----------+-------------------+
10 rows in set (0.00 sec)
```

```
mysql> select * from courses;
+------------+--------------+---------+------------+
| course_id  | course_name  | credits | teacher_id |
+------------+--------------+---------+------------+
|          1 | physics      |       4 |          1 |
|          2 | chemistry    |       3 |          2 |
|          3 | biology      |       4 |          3 |
|          4 | mathematics  |       5 |          4 |
|          5 | history      |       4 |          5 |
|          6 | civics       |       5 |          6 |
|          7 | economics    |       2 |          7 |
|          8 | geography    |       3 |          8 |
|          9 | computer     |       5 |          9 |
|         10 | dbms         |       4 |         10 |
+------------+--------------+---------+------------+
10 rows in set (0.00 sec)
```

```
ysql> select teachers.teacher_id,teachers.first_name,teachers.last_name fro
 teachers
    -> LEFT JOIN courses ON teachers.teacher_id=courses.teacher_id
    -> where courses.teacher_id is NULL;
mpty set (0.00 sec)
```

ASSIGNMENT 2 SQL

```
ysql> alter table teachers
   -> modify column teacher_id INT auto_increment;
ERROR 1833 (HY000): Cannot change column 'teacher_id': used in a foreign key
constraint 'courses_ibfk_1' of table 'sisdb.courses'
```

```
mysql> alter table courses
    -> drop foreign key courses_ibfk_1;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
ysql> alter table teachers
    -> modify column teacher_id INT auto_increment;
Query OK, 10 rows affected (0.04 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> alter table courses
    -> add foreign key(teacher_id) References teachers(teacher_id);
Query OK, 10 rows affected (0.06 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> insert into teachers(first_name,last_name,email)values('kumod','kumar
','kumod@gmail.com');
Query OK, 1 row affected (0.00 sec)

mysql> insert into teachers(first_name,last_name,email)values('subash','yada
v','subash@gmail.com');
Query OK, 1 row affected (0.00 sec)

mysql> select teachers.teacher_id,teachers.first_name from teachers
    -> LEFT JOIN courses ON teachers.teacher_id=courses.teacher_id
    -> where courses.teacher_id is NULL;
+------------+------------+
| teacher_id | first_name |
+------------+------------+
|         11 | kumod      |
|         12 | subash     |
+------------+------------+
2 rows in set (0.00 sec)
```

ASSIGNMENT 2 SQL

Task 4. Subquery and its type:

1. **Write an SQL query to calculate the average number of students enrolled in each course. Use aggregate functions and subqueries to achieve this**

```
ysql> SELECT
    ->     course_id,
    ->     course_name,
    ->     AVG(enrollment_count) AS average_students_enrolled
    -> FROM (
    ->     SELECT
    ->         Courses.course_id,
    ->         Courses.course_name,
    ->         COUNT(Enrollments.student_id) AS enrollment_count
    ->     FROM
    ->         Courses
    ->     LEFT JOIN
    ->         Enrollments ON Courses.course_id = Enrollments.course_id
    ->     GROUP BY
    ->         Courses.course_id, Courses.course_name
    -> ) AS CourseEnrollmentCounts
    -> GROUP BY
    ->     course_id, course_name;
+-----------+-------------+---------------------------+
| course_id | course_name | average_students_enrolled |
+-----------+-------------+---------------------------+
|         1 | physics     |                    1.0000 |
|         2 | chemistry   |                    1.0000 |
|         3 | biology     |                    1.0000 |
|         4 | mathematics |                    2.0000 |
|         5 | history     |                    1.0000 |
|         6 | civics      |                    1.0000 |
|         7 | economics   |                    1.0000 |
|         8 | geography   |                    1.0000 |
|         9 | computer    |                    1.0000 |
|        10 | dbms        |                    0.0000 |
+-----------+-------------+---------------------------+
10 rows in set (0.01 sec)
```

2. **Identify the student(s) who made the highest payment. Use a subquery to find the maximum payment amount and then retrieve the student(s) associated with that amount.**

```
mysql> select students.student_id,students.first_name,payments.amount from students
    -> join payments on students.student_id=payments.student_id
    -> where payments.amount=(select  max(amount) from payments);
+------------+------------+--------+
| student_id | first_name | amount |
+------------+------------+--------+
|          8 | rocky      | 999.00 |
+------------+------------+--------+
1 row in set (0.00 sec)
```

3. **Retrieve a list of courses with the highest number of enrollments. Use subqueries to find the course(s) with the maximum enrollment count**

```
mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2024-01-10      |
|             2 |          2 |         2 | 2024-01-11      |
|             3 |          3 |         3 | 2024-01-12      |
|             4 |          4 |         4 | 2024-01-13      |
|             5 |          5 |         5 | 2024-01-14      |
|             6 |          6 |         6 | 2024-01-15      |
|             7 |          7 |         7 | 2024-01-16      |
|             8 |          8 |         8 | 2024-01-17      |
|            11 |          9 |         9 | 2024-01-18      |
|            12 |          1 |         4 | 2024-01-19      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)
```

```
mysql> SELECT
    ->     Courses.course_id,
    ->     Courses.course_name,
    ->     enrollment_count
    -> FROM
    ->     Courses
    -> JOIN (
    ->     SELECT
    ->         course_id,
    ->         COUNT(student_id) AS enrollment_count
    ->     FROM
    ->         Enrollments
    ->     GROUP BY
    ->         course_id
    -> ) AS CourseEnrollmentCounts ON Courses.course_id = CourseEnrollmentCounts.course_id
    -> WHERE
    ->     enrollment_count = (
    ->         SELECT
    ->             MAX(enrollment_count)
    ->         FROM
    ->             (
    ->                 SELECT
    ->                     course_id,
    ->                     COUNT(student_id) AS enrollment_count
    ->                 FROM
    ->                     Enrollments
    ->                 GROUP BY
    ->                     course_id
    ->             ) AS MaxEnrollmentCounts
    ->     );
+-----------+-------------+-----------------+
| course_id | course_name | enrollment_count |
+-----------+-------------+-----------------+
|         4 | mathematics |               2 |
+-----------+-------------+-----------------+
1 row in set (0.00 sec)
```

4. **Calculate the total payments made to courses taught by each teacher. Use subqueries to sum payments for each teacher's courses.**

```
mysql> SELECT
    ->     teachers.teacher_id,
    ->     teachers.first_name,
    ->     SUM(Payments.amount) AS total_payments
    -> FROM
    ->     teachers
    -> LEFT JOIN
    ->     courses ON teachers.teacher_id = courses.teacher_id
    -> LEFT JOIN
    ->     enrollments ON courses.course_id = enrollments.course_id
    -> LEFT JOIN
    ->     payments ON enrollments.student_id = payments.student_id
    -> GROUP BY
    ->     teachers.teacher_id, teachers.first_name;
+------------+------------+----------------+
| teacher_id | first_name | total_payments |
+------------+------------+----------------+
|          1 | ram        |         600.00 |
|          2 | sam        |          55.00 |
|          3 | tam        |          60.00 |
|          4 | jam        |         665.00 |
|          5 | aam        |          70.00 |
|          6 | bam        |          75.00 |
|          7 | cam        |          80.00 |
|          8 | dam        |         999.00 |
|          9 | eam        |          95.00 |
|         10 | sourav     |           NULL |
|         11 | kumod      |           NULL |
|         12 | subash     |           NULL |
+------------+------------+----------------+
12 rows in set (0.00 sec)
```

5. **Identify students who are enrolled in all available courses. Use subqueries to compare a student's enrollments with the total number of courses.**

```
mysql> SELECT
    ->     student_id,
    ->     first_name,
    ->     last_name
    -> FROM
    ->     Students
    -> WHERE
    ->     (
    ->         SELECT COUNT(DISTINCT course_id)
    ->         FROM Enrollments
    ->     ) = (
    ->         SELECT COUNT(DISTINCT course_id)
    ->         FROM Enrollments AS e2
    ->         WHERE Students.student_id = e2.student_id
    ->     );
Empty set (0.00 sec)
```

6. **Retrieve the names of teachers who have not been assigned to any courses. Use subqueries to find teachers with no course assignments**

```
mysql> select teacher_id,first_name,last_name from teachers
    -> where teacher_id NOT IN (select distinct teacher_id from courses);
+------------+------------+-----------+
| teacher_id | first_name | last_name |
+------------+------------+-----------+
|         11 | kumod      | kumar     |
|         12 | subash     | yadav     |
+------------+------------+-----------+
2 rows in set (0.00 sec)
```

7. Calculate the average age of all students. Use subqueries to calculate the age of each student based on their date of birth.

```
mysql> SELECT
    ->     AVG(student_age) AS average_age
    -> FROM (
    ->     SELECT
    ->         student_id,
    ->         TIMESTAMPDIFF(YEAR, date_of_birth, CURDATE()) AS student_age
    ->     FROM
    ->         Students
    -> ) AS StudentAges;
+-------------+
| average_age |
+-------------+
|     23.2727 |
+-------------+
1 row in set (0.00 sec)
```

8. **Identify courses with no enrollments. Use subqueries to find courses without enrollment records.**

```
mysql> select course_id,course_name from courses
    -> where course_id NOT IN (select distinct course_id from enrollments);
+-----------+-------------+
| course_id | course_name |
+-----------+-------------+
|        10 | dbms        |
|        11 | mysql       |
|        12 | python      |
+-----------+-------------+
3 rows in set (0.00 sec)

mysql>
```

9. **Calculate the total payments made by each student for each course they are enrolled in. Use subqueries and aggregate functions to sum payments.**

```
ysql> SELECT
    ->     Enrollments.student_id,
    ->     Students.first_name,
    ->     Students.last_name,
    ->     Enrollments.course_id,
    ->     Courses.course_name,
    ->     SUM(Payments.amount) AS total_payments
    -> FROM
    ->     Enrollments
    -> JOIN
    ->     Students ON Enrollments.student_id = Students.student_id
    -> JOIN
    ->     Courses ON Enrollments.course_id = Courses.course_id
    -> LEFT JOIN
    ->     Payments ON Enrollments.student_id = Payments.student_id
    ->
    -> group by
    ->     Enrollments.student_id,
    ->     Students.first_name,
    ->     Students.last_name,
    ->     Enrollments.course_id,
    ->     Courses.course_name;
```

```
    ->     Courses.course_name;
+------------+------------+-----------+-----------+-------------+----------------+
| student_id | first_name | last_name | course_id | course_name | total_payments |
+------------+------------+-----------+-----------+-------------+----------------+
|          1 | Sourav     | kumar     |         1 | physics     |         600.00 |
|          1 | Sourav     | kumar     |         4 | mathematics |         600.00 |
|          2 | Rishav     | kumar     |         2 | chemistry   |          55.00 |
|          3 | Subham     | gupta     |         3 | biology     |          60.00 |
|          4 | mayank     | mohak     |         4 | mathematics |          65.00 |
|          5 | abhishek   | jha       |         5 | history     |          70.00 |
|          6 | rahul      | kumar     |         6 | civics      |          75.00 |
|          7 | Satyam     | kumar     |         7 | economics   |          80.00 |
|          8 | rocky      | gupta     |         8 | geography   |         999.00 |
|          9 | purusottam | kumar     |         9 | computer    |          95.00 |
+------------+------------+-----------+-----------+-------------+----------------+
10 rows in set (0.00 sec)
```

10. **Identify students who have made more than one payment. Use subqueries and aggregate functions to count payments per student and filter for those with counts greater than one**

```
mysql> SELECT
    ->     student_id,
    ->     first_name,
    ->     last_name
    -> FROM
    ->     Students
    -> WHERE
    ->     student_id IN (
    ->         SELECT
    ->             student_id
    ->         FROM
    ->             Payments
    ->         GROUP BY
    ->             student_id
    ->         HAVING
    ->             COUNT(*) > 1
    ->     );
+------------+------------+-----------+
| student_id | first_name | last_name |
+------------+------------+-----------+
|          1 | Sourav     | kumar     |
+------------+------------+-----------+
1 row in set (0.00 sec)
```

11. **Write an SQL query to calculate the total payments made by each student. Join the "Students" table with the "Payments" table and use GROUP BY to calculate the sum of payments for each student**

```
mysql> SELECT
    ->     Students.student_id,
    ->     Students.first_name,
    ->     Students.last_name,
    ->     SUM(Payments.amount)AS total_payments
    -> FROM
    ->     Students
    -> LEFT JOIN
    ->     Payments ON Students.student_id = Payments.student_id
    -> GROUP BY
    ->     Students.student_id, Students.first_name, Students.last_name;
+------------+------------+-----------+----------------+
| student_id | first_name | last_name | total_payments |
+------------+------------+-----------+----------------+
|          1 | Sourav     | kumar     |         600.00 |
|          2 | Rishav     | kumar     |          55.00 |
|          3 | Subham     | gupta     |          60.00 |
|          4 | mayank     | mohak     |          65.00 |
|          5 | abhishek   | jha       |          70.00 |
|          6 | rahul      | kumar     |          75.00 |
|          7 | Satyam     | kumar     |          80.00 |
|          8 | rocky      | gupta     |         999.00 |
|          9 | purusottam | kumar     |          95.00 |
|         10 | anurag     | singh     |         100.00 |
|         11 | sanjeev    | kumar     |           NULL |
+------------+------------+-----------+----------------+
11 rows in set (0.00 sec)
```

12. **Retrieve a list of course names along with the count of students enrolled in each course. Use JOIN operations between the "Courses" table and the "Enrollments" table and GROUP BY to count enrollments**

```
mysql> SELECT
    ->     Courses.course_id,
    ->     Courses.course_name,
    ->     COUNT(Enrollments.student_id) AS enrolled_students_count
    -> FROM
    ->     Courses
    -> LEFT JOIN
    ->     Enrollments ON Courses.course_id = Enrollments.course_id
    -> GROUP BY
    ->     Courses.course_id, Courses.course_name;
+-----------+-------------+--------------------------+
| course_id | course_name | enrolled_students_count  |
+-----------+-------------+--------------------------+
|         1 | physics     |                        1 |
|         2 | chemistry   |                        1 |
|         3 | biology     |                        1 |
|         4 | mathematics |                        2 |
|         5 | history     |                        1 |
|         6 | civics      |                        1 |
|         7 | economics   |                        1 |
|         8 | geography   |                        1 |
|         9 | computer    |                        1 |
|        10 | dbms        |                        0 |
|        11 | mysql       |                        0 |
|        12 | python      |                        0 |
+-----------+-------------+--------------------------+
12 rows in set (0.00 sec)
```

13.

13. **Calculate the average payment amount made by students. Use JOIN operations between the "Students" table and the "Payments" table and GROUP BY to calculate the average**

```
mysql> SELECT
    ->     Students.student_id,
    ->     Students.first_name,
    ->     Students.last_name,
    ->     AVG(Payments.amount) AS average_payment_amount
    -> FROM
    ->     Students
    -> LEFT JOIN
    ->     Payments ON Students.student_id = Payments.student_id
    -> GROUP BY
    ->     Students.student_id, Students.first_name, Students.last_name;
+------------+------------+-----------+-----------------------+
| student_id | first_name | last_name | average_payment_amount |
+------------+------------+-----------+-----------------------+
|          1 | Sourav     | kumar     |            150.000000 |
|          2 | Rishav     | kumar     |             55.000000 |
|          3 | Subham     | gupta     |             60.000000 |
|          4 | mayank     | mohak     |             65.000000 |
|          5 | abhishek   | jha       |             70.000000 |
|          6 | rahul      | kumar     |             75.000000 |
|          7 | Satyam     | kumar     |             80.000000 |
|          8 | rocky      | gupta     |            999.000000 |
|          9 | purusottam | kumar     |             95.000000 |
|         10 | anurag     | singh     |            100.000000 |
|         11 | sanjeev    | kumar     |                  NULL |
+------------+------------+-----------+-----------------------+
11 rows in set (0.00 sec)
```