

## Assignment sql-2

### Task-1: Database design

#### 1. Create database sisdb

```
mysql> create database SISDB;  
Query OK, 1 row affected (0.02 sec)
```

```
mysql> Show databases;
```

Database
d1
hexaware
information_schema
mysql
performance_schema
petpals
sakila
sisdb
sys
techshop
world

```
11 rows in set (0.00 sec)
```

```
mysql> use sisdb;
```

```
Database changed
```

```
mysql> |
```

#### 2. Create tables

##### 1.Students

```
Database changed  
mysql> create table students (student_id int primary key,first_name text,last_name text,date_of_birth date,email varchar(20),phonenummer bigint);  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> desc students;
```

Field	Type	Null	Key	Default	Extra
student_id	int	NO	PRI	NULL	
first_name	text	YES		NULL	
last_name	text	YES		NULL	
date_of_birth	date	YES		NULL	
email	varchar(20)	YES		NULL	
phonenummer	bigint	YES		NULL	

```
6 rows in set (0.00 sec)
```

## 2.teacher

```
mysql> create table teacher (teacher_id int primary key,first_name text,last_name text,email varchar(20));
Query OK, 0 rows affected (0.04 sec)

mysql> desc teacher;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| teacher_id | int       | NO   | PRI | NULL    |       |
| first_name  | text     | YES  |     | NULL    |       |
| last_name   | text     | YES  |     | NULL    |       |
| email       | varchar(20) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## 3.courses

```
mysql> create table courses (course_id int primary key,course_name text,credits int,teacher_id int,foreign key(teacher_id) references teacher(teacher_id));
Query OK, 0 rows affected (0.06 sec)

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

## 4. Enrollments

```
mysql> create table enrollments (enrollment_id int primary key,student_id int, foreign key(student_id) references students(student_id),course_id int,foreign key(course_id) references courses(course_id),enrollment_date date);
Query OK, 0 rows affected (0.11 sec)

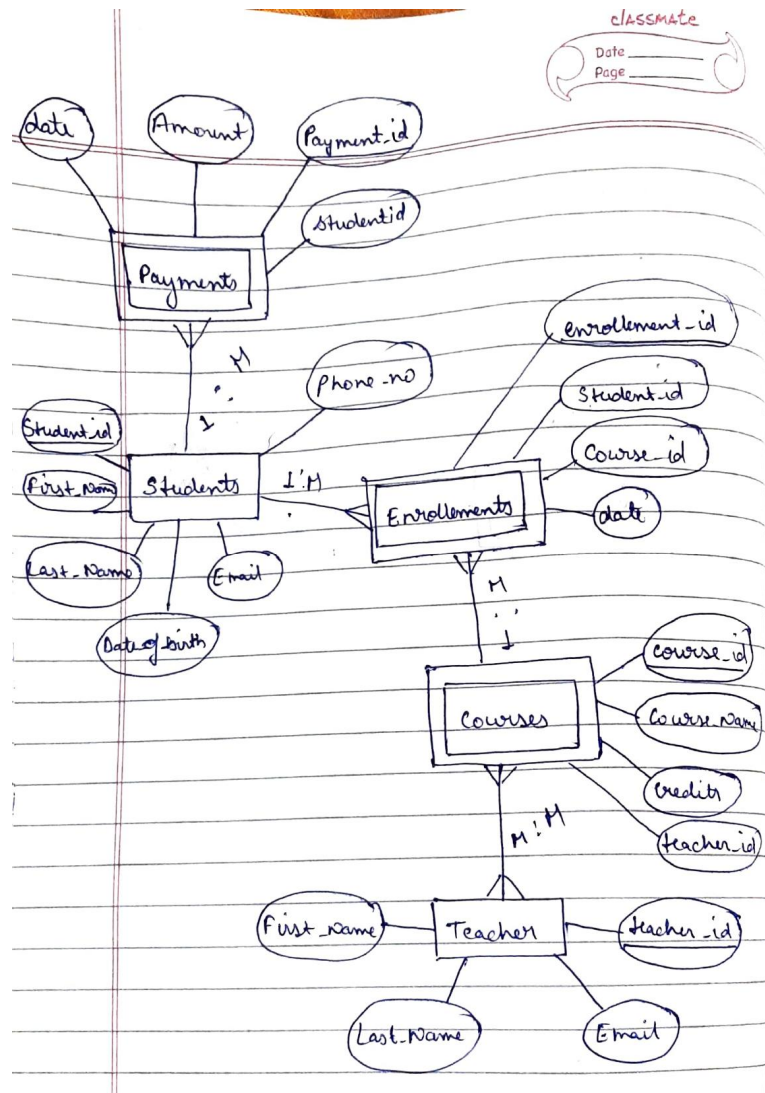
mysql> desc enrollments;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| enrollment_id | int       | NO   | PRI | NULL    |       |
| student_id    | int       | YES  | MUL | NULL    |       |
| course_id     | int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

## 5.payments

```
mysql> create table payments (payment_id int primary key,student_id int, foreign key(student_id) references students(student_id),amount bigint,enrollment_date date);
Query OK, 0 rows affected (0.07 sec)

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

## 3. Create entity relationship diagram



4. Create primary and foreign key

Already done above

5. Insert 10 sample data in each table

1. Students

```
mysql> insert into students values(1,'Asmita','Porwal','2001-04-04','asmita@gmail.com','7876765432'),(2,'Bhavya','Khurana','2001-10-24','bhav@gmail.com','9376765434'),(3,'Apoorva','Porwal','1993-11-17','apoorva@gmail.com','7877755532'),(4,'Neha','Porwal','2003-02-02','Neha@gmail.com','7876005439'),(5,'Rahul','Dusaj','2001-01-05','rahul@gmail.com','7854765489'),(6,'Riya','Porwal','2000-04-02','riya@gmail.com','7076765432'),(7,'Raman','Sharma','2001-04-09','raman@gmail.com','9826765431'),(8,'Radha','Singh','2005-03-03','radha@gmail.com','7872785432'),(9,'Shyam','Porwal','2002-05-04','shyam@gmail.com','9876738431'),(10,'Ram','Gupta','2001-04-14','ram@gmail.com','8987645679');
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
mysql> select * from students;
```

student_id	first_name	last_name	date_of_birth	email	phonenum
1	Asmita	Porwal	2001-04-04	asmita@gmail.com	7876765432
2	Bhavya	Khurana	2001-10-24	bhav@gmail.com	9376765434
3	Apoorva	Porwal	1993-11-17	apoorva@gmail.com	7877755532
4	Neha	Porwal	2003-02-02	Neha@gmail.com	7876005439
5	Rahul	Dusaj	2001-01-05	rahul@gmail.com	7854765489
6	Riya	Porwal	2000-04-02	riya@gmail.com	7076765432
7	Raman	Sharma	2001-04-09	raman@gmail.com	9826765431
8	Radha	Singh	2005-03-03	radha@gmail.com	7872785432
9	Shyam	Porwal	2002-05-04	shyam@gmail.com	9876738431
10	Ram	Gupta	2001-04-14	ram@gmail.com	8987645679

```
10 rows in set (0.00 sec)
```

## 2. Teacher

```
mysql> insert into teacher values(1,'Grace','Singh','grace@gmail.com'),(2,'Janvi','Singh','janvi@gmail.com'),(3,'Tanu','taneja','tanu@gmail.com'),(4,'Tushar','Khurana','tushar@gmail.com'),(5,'Shilpa','Singh','shilpa@gmail.com'),(6,'Naresh','Gupta','naresh@gmail.com'),(7,'Mamta','Gupta','mamta@gmail.com'),(8,'Paritosh','Porwal','paritosh@gmail.com'),(9,'Suman','Sharma','suman@gmail.com'),(10,'Ramesh','Sood','ramesh@gmail.com');
Query OK, 10 rows affected (0.04 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
mysql> select * from teacher;
```

teacher_id	first_name	last_name	email
1	Grace	Singh	grace@gmail.com
2	Janvi	Singh	janvi@gmail.com
3	Tanu	taneja	tanu@gmail.com
4	Tushar	Khurana	tushar@gmail.com
5	Shilpa	Singh	shilpa@gmail.com
6	Naresh	Gupta	naresh@gmail.com
7	Mamta	Gupta	mamta@gmail.com
8	Paritosh	Porwal	paritosh@gmail.com
9	Suman	Sharma	suman@gmail.com
10	Ramesh	Sood	ramesh@gmail.com

```
10 rows in set (0.00 sec)
```

## 3.Courses

```
mysql> insert into courses values(1,'General Knowledge',4,9),(2,'Arts',3,10),(3,'Commerce',4,8),(4,'Moral Science',3,5),(5,'Sanskrit',3,4),(6,'Social Science',4,3),(7,'Science',5,7),(8,'Maths',4,6),(9,'Hindi',3,1),(10,'English',4,2);
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0

mysql> select * from courses;
```

course_id	course_name	credits	teacher_id
1	General Knowledge	4	9
2	Arts	3	10
3	Commerce	4	8
4	Moral Science	3	5
5	Sanskrit	3	4
6	Social Science	4	3
7	Science	5	7
8	Maths	4	6
9	Hindi	3	1
10	English	4	2

```
10 rows in set (0.00 sec)
```

## 4.Enrollments

```
mysql> insert into enrollments values(1,9,8,'2022-09-02'),(2,7,10,'2019-02-01'),(3,10,9,'2019-11-19'),(4,8,7,'2023-02-14'),(5,6,6,'2021-11-11'),(6,4,5,'2021-09-07'),(7,3,2,'2023-01-02'),(8,5,1,'2021-04-19'),(9,2,3,'2020-04-10'),(10,1,4,'2020-03-01');
Query OK, 10 rows affected (0.02 sec)
Records: 10 Duplicates: 0 Warnings: 0

mysql> select * from enrollments;
```

enrollment_id	student_id	course_id	enrollment_date
1	9	8	2022-09-02
2	7	10	2019-02-01
3	10	9	2019-11-19
4	8	7	2023-02-14
5	6	6	2021-11-11
6	4	5	2021-09-07
7	3	2	2023-01-02
8	5	1	2021-04-19
9	2	3	2020-04-10
10	1	4	2020-03-01

```
10 rows in set (0.00 sec)
```

## 5.Payments

```
mysql> insert into payments values(101,7,200,'2019-02-01'),(102,7,200,'2019-04-01'),(103,1,400,'2020-03-01'),(104,2,300,'2020-04-01'),(105,2,100,'2020-06-01'),(106,10,400,'2019-11-19'),(107,4,400,'2021-09-07'),(108,6,400,'2021-11-11'),(109,5,400,'2021-04-19'),(110,9,400,'2022-09-02'),(111,8,400,'2023-02-14'),(112,3,200,'2023-01-02');
Query OK, 12 rows affected (0.02 sec)
Records: 12 Duplicates: 0 Warnings: 0

mysql> select * from payments;
```

payment_id	student_id	amount	payment_date
101	7	200	2019-02-01
102	7	200	2019-04-01
103	1	400	2020-03-01
104	2	300	2020-04-01
105	2	100	2020-06-01
106	10	400	2019-11-19
107	4	400	2021-09-07
108	6	400	2021-11-11
109	5	400	2021-04-19
110	9	400	2022-09-02
111	8	400	2023-02-14
112	3	200	2023-01-02

```
12 rows in set (0.00 sec)
```

## Task-2

### 1.insert a new student

```
mysql> insert into students values(11,'John','Doe','1995-08-15','john.doe@example.com',1234567890);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from students;
```

student_id	first_name	last_name	date_of_birth	email	phonenumber
1	Asmita	Porwal	2001-04-04	asmita@gmail.com	7876765432
2	Bhavya	Khurana	2001-10-24	bhav@gmail.com	9376765434
3	Apoorva	Porwal	1993-11-17	apoorva@gmail.com	7877755532
4	Neha	Porwal	2003-02-02	Neha@gmail.com	7876005439
5	Rahul	Dusaj	2001-01-05	rahul@gmail.com	7854765489
6	Riya	Porwal	2000-04-02	riya@gmail.com	7076765432
7	Raman	Sharma	2001-04-09	raman@gmail.com	9826765431
8	Radha	Singh	2005-03-03	radha@gmail.com	7872785432
9	Shyam	Porwal	2002-05-04	shyam@gmail.com	9876738431
10	Ram	Gupta	2001-04-14	ram@gmail.com	8987645679
11	John	Doe	1995-08-15	john.doe@example.com	1234567890

```
11 rows in set (0.01 sec)
```

2. Write an sql query to enroll a student in a course

```
mysql> insert into enrollments values(11,9,10,'2023-01-01');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from enrollments;
```

enrollment_id	student_id	course_id	enrollment_date
1	9	8	2022-09-02
2	7	10	2019-02-01
3	10	9	2019-11-19
4	8	7	2023-02-14
5	6	6	2021-11-11
6	4	5	2021-09-07
7	3	2	2023-01-02
8	5	1	2021-04-19
9	2	3	2020-04-10
10	1	4	2020-03-01
11	9	10	2023-01-01

```
11 rows in set (0.00 sec)
```

3. Update the email address of a specific teacher

```
mysql> update teacher set email="grace.singh@yahoo.com" where teacher_id=1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from teacher;
```

teacher_id	first_name	last_name	email
1	Grace	Singh	grace.singh@yahoo.com
2	Janvi	Singh	janvi@gmail.com
3	Tanu	taneja	tanu@gmail.com
4	Tushar	Khurana	tushar@gmail.com
5	Shilpa	Singh	shilpa@gmail.com
6	Naresh	Gupta	naresh@gmail.com
7	Mamta	Gupta	mamta@gmail.com
8	Paritosh	Porwal	paritosh@gmail.com
9	Suman	Sharma	suman@gmail.com
10	Ramesh	Sood	ramesh@gmail.com

```
10 rows in set (0.00 sec)
```

#### 4. Delete a specific enrollment record

```
mysql> delete from enrollments where student_id=10 and course_id=9;
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from enrollments;
```

enrollment_id	student_id	course_id	enrollment_date
1	9	8	2022-09-02
2	7	10	2019-02-01
4	8	7	2023-02-14
5	6	6	2021-11-11
6	4	5	2021-09-07
7	3	2	2023-01-02
8	5	1	2021-04-19
9	2	3	2020-04-10
10	1	4	2020-03-01
11	9	10	2023-01-01

```
10 rows in set (0.00 sec)
```

5.Update the courses table to assign a specific teacher to a course.

```
mysql> select * from courses;
```

course_id	course_name	credits	teacher_id
1	General Knowledge	4	9
2	Arts	3	10
3	Commerce	4	8
4	Moral Science	3	5
5	Sanskrit	3	4
6	Social Science	4	3
7	Science	5	7
8	Maths	4	6
9	Hindi	3	1
10	English	4	2

```
10 rows in set (0.01 sec)
```

```
mysql> update courses set teacher_id=8 where course_id=1;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from courses;
```

course_id	course_name	credits	teacher_id
1	General Knowledge	4	8
2	Arts	3	10
3	Commerce	4	8
4	Moral Science	3	5
5	Sanskrit	3	4
6	Social Science	4	3
7	Science	5	7
8	Maths	4	6
9	Hindi	3	1
10	English	4	2

```
10 rows in set (0.00 sec)
```

6.Delete a student and their enrollment records



```
mysql> delete from enrollments where student_id=2;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> delete from payments where student_id=2;
Query OK, 2 rows affected (0.01 sec)
```

```
mysql> delete from students where student_id=2;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from students;
```

student_id	first_name	last_name	date_of_birth	email	phonenummer
1	Asmita	Porwal	2001-04-04	asmita@gmail.com	7876765432
3	Apoorva	Porwal	1993-11-17	apoorva@gmail.com	7877755532
4	Neha	Porwal	2003-02-02	Neha@gmail.com	7876005439
5	Rahul	Dusaj	2001-01-05	rahul@gmail.com	7854765489
6	Riya	Porwal	2000-04-02	riya@gmail.com	7076765432
7	Raman	Sharma	2001-04-09	raman@gmail.com	9826765431
8	Radha	Singh	2005-03-03	radha@gmail.com	7872785432
9	Shyam	Porwal	2002-05-04	shyam@gmail.com	9876738431
10	Ram	Gupta	2001-04-14	ram@gmail.com	8987645679
11	John	Doe	1995-08-15	john.doe@example.com	1234567890

10 rows in set (0.00 sec)

7. Update the payment amount from payments table.

```
mysql> select * from payments;
```

payment_id	student_id	amount	payment_date
101	7	200	2019-02-01
102	7	200	2019-04-01
103	1	400	2020-03-01
106	10	400	2019-11-19
107	4	400	2021-09-07
108	6	400	2021-11-11
109	5	400	2021-04-19
110	9	400	2022-09-02
111	8	400	2023-02-14
112	3	200	2023-01-02

```
10 rows in set (0.00 sec)
```

```
mysql> update payments set amount=400 where payment_id=112;
```

```
Query OK, 1 row affected (0.01 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from payments;
```

payment_id	student_id	amount	payment_date
101	7	200	2019-02-01
102	7	200	2019-04-01
103	1	400	2020-03-01
106	10	400	2019-11-19
107	4	400	2021-09-07
108	6	400	2021-11-11
109	5	400	2021-04-19
110	9	400	2022-09-02
111	8	400	2023-02-14
112	3	400	2023-01-02

```
10 rows in set (0.00 sec)
```

### Task -3

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 s.student_id, concat(s.first_name,' ',last_name) as Student_Name, SUM(p.amount) AS total_payments_made
-> FROM Students s
-> inner join Payments p ON s.student_id = p.student_id
-> WHERE s.student_id = '7'
-> GROUP BY s.student_id;
```

student_id	Student_Name	total_payments_made
7	Raman Sharma	400

1 row in set (0.01 sec)

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 c.course_id, c.course_name, COUNT(e.student_id) AS enrolled_students_count
-> FROM Courses c
-> LEFT JOIN Enrollments e ON c.course_id = e.course_id
-> GROUP BY c.course_id, c.course_name;
```

course_id	course_name	enrolled_students_count
1	General Knowledge	1
2	Arts	1
3	Commerce	0
4	Moral Science	1
5	Sanskrit	1
6	Social Science	1
7	Science	1
8	Maths	1
9	Hindi	0
10	English	2

10 rows in set (0.00 sec)

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 s.student_id, concat(s.first_name,' ',s.last_name) as student_name
-> FROM Students s
-> LEFT JOIN Enrollments e ON s.student_id = e.student_id
-> WHERE e.student_id IS NULL;
+-----+-----+
| student_id | student_name |
+-----+-----+
|          10 | Ram Gupta    |
|          11 | John Doe     |
+-----+-----+
2 rows in set (0.01 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 s.first_name, s.last_name, c.course_name
-> FROM Students s
-> JOIN Enrollments e ON s.student_id = e.student_id
-> JOIN Courses c ON e.course_id = c.course_id;
+-----+-----+-----+
| first_name | last_name | course_name |
+-----+-----+-----+
| Shyam      | Porwal    | Maths       |
| Raman      | Sharma    | English     |
| Radha      | Singh     | Science     |
| Riya       | Porwal    | Social Science |
| Neha       | Porwal    | Sanskrit    |
| Apoorva    | Porwal    | Arts        |
| Rahul      | Dusaj     | General Knowledge |
| Asmita     | Porwal    | Moral Science |
| Shyam      | Porwal    | English     |
+-----+-----+-----+
9 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 concat(t.first_name,' ',t.last_name)as teacher_name, c.course_name
-> FROM Teacher t
-> JOIN Courses c ON t.teacher_id = c.teacher_id;
```

teacher_name	course_name
Grace Singh	Hindi
Janvi Singh	English
Tanu taneja	Social Science
Tushar Khurana	Sanskrit
Shilpa Singh	Moral Science
Naresh Gupta	Maths
Mamta Gupta	Science
Paritosh Porwal	General Knowledge
Paritosh Porwal	Commerce
Ramesh Sood	Arts

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 s.first_name, s.last_name, e.enrollment_date
-> FROM Students s
-> JOIN Enrollments e ON s.student_id = e.student_id
-> JOIN Courses c ON e.course_id = c.course_id
-> WHERE c.course_id = 10;
```

first_name	last_name	enrollment_date
Raman	Sharma	2019-02-01
Shyam	Porwal	2023-01-01

2 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 s.student_id, s.first_name, s.last_name
-> FROM Students s
-> LEFT JOIN Payments p ON s.student_id = p.student_id
-> WHERE p.student_id IS NULL;
+-----+-----+-----+
| student_id | first_name | last_name |
+-----+-----+-----+
|          11 | John      | Doe       |
+-----+-----+-----+
1 row in set (0.00 sec)
```

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 c.course_id, c.course_name
-> FROM Courses c
-> LEFT JOIN Enrollments e ON c.course_id = e.course_id
-> WHERE e.course_id IS NULL;
+-----+-----+
| course_id | course_name |
+-----+-----+
|          3 | Commerce    |
|          9 | Hindi       |
+-----+-----+
2 rows 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 DISTINCT e1.student_id, concat(s.first_name, ' ', last_name)
-> 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;
+-----+-----+
| student_id | concat(s.first_name, ' ', last_name) |
+-----+-----+
|          9 | Shyam Porwal                          |
+-----+-----+
1 row in set (0.01 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 t.teacher_id, concat(t.first_name,' ',t.last_name) as teacher_name
-> FROM Teacher t
-> LEFT JOIN Courses c ON t.teacher_id = c.teacher_id
-> WHERE c.course_id IS NULL;
+-----+-----+
| teacher_id | teacher_name |
+-----+-----+
|          9 | Suman Sharma |
+-----+-----+
1 row in set (0.00 sec)
```

#### Task-4

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

```
mysql> SELECT c.course_id, c.course_name, AVG(num_students) AS average_students_per_course
-> FROM courses c
-> LEFT JOIN (
->   SELECT course_id, COUNT(DISTINCT student_id) AS num_students
->   FROM enrollments
->   GROUP BY course_id
-> ) AS enrollment_counts ON c.course_id = enrollment_counts.course_id
-> GROUP BY c.course_id, c.course_name;
+-----+-----+-----+
| course_id | course_name | average_students_per_course |
+-----+-----+-----+
|          1 | General Knowledge | 1.0000 |
|          2 | Arts | 1.0000 |
|          3 | Commerce | NULL |
|          4 | Moral Science | 1.0000 |
|          5 | Sanskrit | 1.0000 |
|          6 | Social Science | 1.0000 |
|          7 | Science | 1.0000 |
|          8 | Maths | 1.0000 |
|          9 | Hindi | NULL |
|         10 | English | 2.0000 |
+-----+-----+-----+
10 rows in set (0.00 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 s.student_id, concat(s.first_name,' ',last_name) as student_name, p.amount
-> FROM Students s
-> JOIN Payments p ON s.student_id = p.student_id
-> WHERE p.amount = (
->     SELECT MAX(amount)
->     FROM Payments
-> );
```

student_id	student_name	amount
1	Asmita Porwal	400
10	Ram Gupta	400
4	Neha Porwal	400
6	Riya Porwal	400
5	Rahul Dusaj	400
9	Shyam Porwal	400
8	Radha Singh	400
3	Apoorva Porwal	400

8 rows 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 c.course_id, c.course_name, COUNT(*) AS enrollment_count
-> FROM courses c
-> JOIN enrollments e ON c.course_id = e.course_id
-> GROUP BY c.course_id, c.course_name
-> HAVING COUNT(*) = (
->     SELECT MAX(enrollment_count)
->     FROM (
->         SELECT COUNT(*) AS enrollment_count
->         FROM enrollments
->         GROUP BY course_id
->     ) AS subquery
-> );
```

course_id	course_name	enrollment_count
10	English	2

1 row in set (0.01 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 t.teacher_id, CONCAT(t.first_name, ' ', t.last_name) AS teacher_name, COALESCE(SUM(p.amount), 0) AS total_payments
-> FROM Teacher t
-> LEFT JOIN Courses c ON t.teacher_id = c.teacher_id
-> LEFT JOIN Enrollments e ON c.course_id = e.course_id
-> LEFT JOIN Payments p ON e.student_id = p.student_id
-> GROUP BY t.teacher_id, teacher_name;
+-----+-----+-----+
| teacher_id | teacher_name | total_payments |
+-----+-----+-----+
| 1 | Grace Singh | 0 |
| 2 | Janvi Singh | 800 |
| 3 | Tanu taneja | 400 |
| 4 | Tushar Khurana | 400 |
| 5 | Shilpa Singh | 400 |
| 6 | Naresh Gupta | 400 |
| 7 | Mamta Gupta | 400 |
| 8 | Paritosh Porwal | 400 |
| 9 | Suman Sharma | 0 |
| 10 | Ramesh Sood | 400 |
+-----+-----+-----+
10 rows in set (0.01 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
-> FROM Enrollments
-> GROUP BY student_id
-> HAVING COUNT(DISTINCT course_id) = (
->     SELECT COUNT(DISTINCT course_id)
->     FROM Courses
-> );
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, CONCAT(first_name, ' ', last_name) AS teacher_name
-> FROM Teacher
-> WHERE teacher_id NOT IN (
->     SELECT DISTINCT teacher_id
->     FROM Courses
-> );
+-----+-----+
| teacher_id | teacher_name |
+-----+-----+
| 9 | Suman Sharma |
+-----+-----+
1 row in set (0.01 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 TIMESTAMPDIFF(YEAR, date_of_birth, CURDATE()) AS student_age
->     FROM students
-> ) AS student_ages;
+-----+
| average_age |
+-----+
|      22.8000 |
+-----+
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 |
+-----+-----+
|          3 | Commerce   |
|          9 | Hindi      |
+-----+-----+
2 rows in set (0.00 sec)
```

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

```
mysql> SELECT e.course_id,
-> COALESCE((
-> SELECT SUM(p.amount)
-> FROM Payments p
-> WHERE p.student_id = e.student_id
-> ), 0) AS total_payments
-> FROM Enrollments e
-> order by course_id;
```

course_id	total_payments
1	400
2	400
4	400
5	400
6	400
7	400
8	400
10	400
10	400

```
9 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 *
-> FROM students
-> WHERE student_id IN (
-> SELECT student_id
-> FROM Payments
-> GROUP BY student_id
-> HAVING COUNT(payment_id) > 1
-> );
```

student_id	first_name	last_name	date_of_birth	email	phonenumber
7	Raman	Sharma	2001-04-09	raman@gmail.com	9826765431

```
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 s.student_id, s.first_name, s.last_name, COALESCE(SUM(p.amount), 0) AS total_payments
-> FROM Students s
-> LEFT JOIN Payments p ON s.student_id = p.student_id
-> GROUP BY s.student_id, s.first_name, s.last_name;
```

student_id	first_name	last_name	total_payments
1	Asmita	Porwal	400
3	Apoorva	Porwal	400
4	Neha	Porwal	400
5	Rahul	Dusaj	400
6	Riya	Porwal	400
7	Raman	Sharma	400
8	Radha	Singh	400
9	Shyam	Porwal	400
10	Ram	Gupta	400
11	John	Doe	0

10 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 c.course_name, COUNT(e.student_id) AS enrolled_students
-> FROM Courses c
-> LEFT JOIN Enrollments e ON c.course_id = e.course_id
-> GROUP BY c.course_name;
```

course_name	enrolled_students
General Knowledge	1
Arts	1
Commerce	0
Moral Science	1
Sanskrit	1
Social Science	1
Science	1
Maths	1
Hindi	0
English	2

10 rows in set (0.00 sec)

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 AVG(p.amount) AS average_payment  
-> FROM Payments p  
-> JOIN Students s ON p.student_id = s.student_id;
```

average_payment
360.0000

1 row in set (0.00 sec)