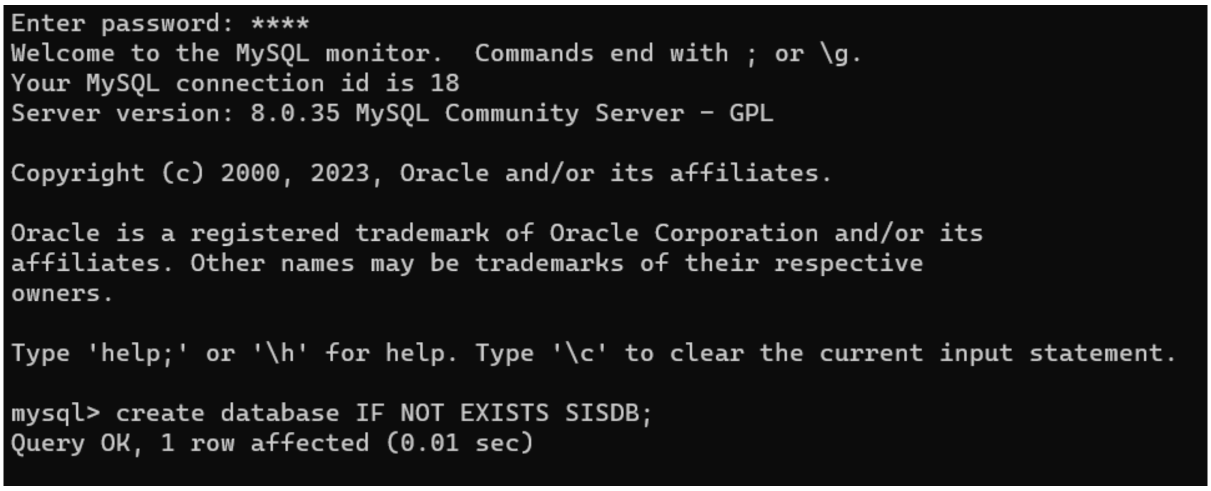
Name :- Sneha Das

Assignment 2 (SQL)

Task 1: Database Design:

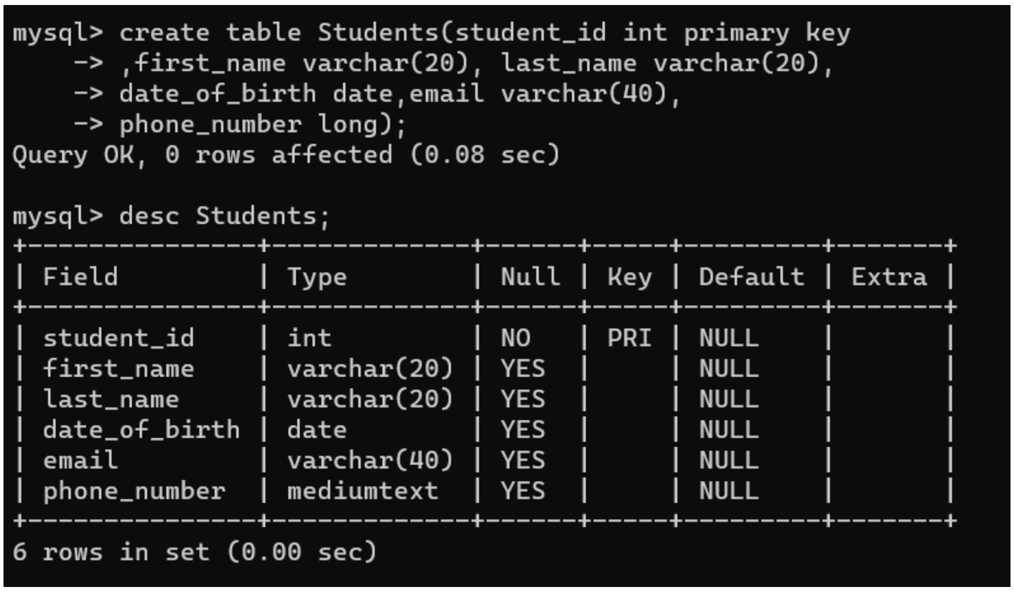
Question 1:- Create the database named SISDB.



```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.35 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database IF NOT EXISTS SISDB;
Query OK, 1 row affected (0.01 sec)
```

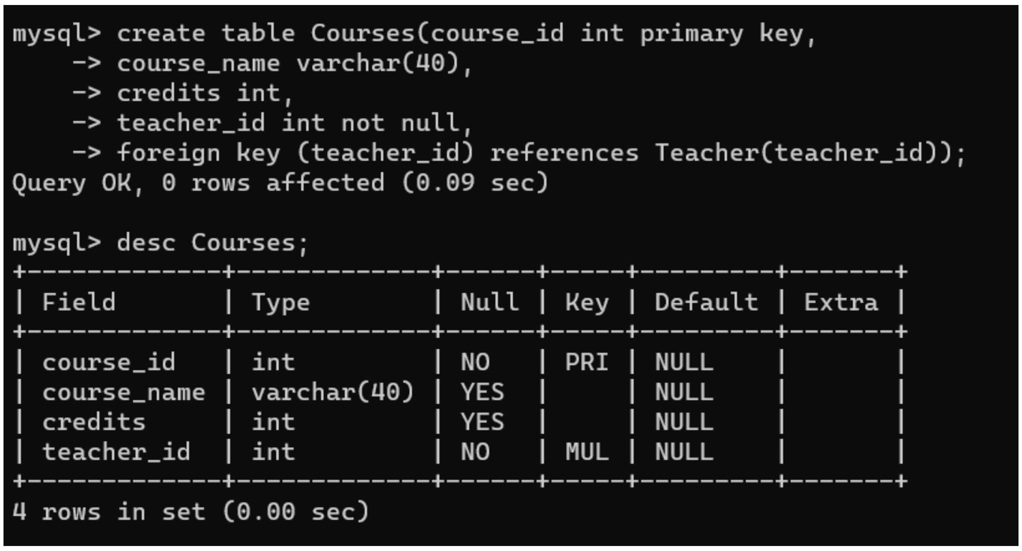
Question 2:- Define the schema for the Students, Courses, Enrollments, Teacher, and Payments tables based on the provided schema. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.

A. Students



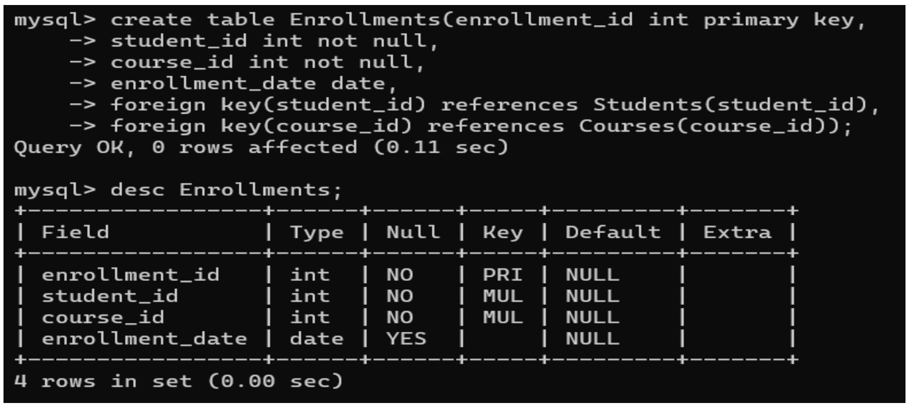
```
mysql> create table Students(student_id int primary key
    -> ,first_name varchar(20), last_name varchar(20),
    -> date_of_birth date,email varchar(40),
    -> phone_number long);
Query OK, 0 rows affected (0.08 sec)

mysql> desc Students;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| student_id    | int         | NO   | PRI | NULL    |       |
| first_name    | varchar(20) | YES  |     | NULL    |       |
| last_name     | varchar(20) | YES  |     | NULL    |       |
| date_of_birth | date        | YES  |     | NULL    |       |
| email         | varchar(40) | YES  |     | NULL    |       |
| phone_number  | mediumtext  | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

B. Courses



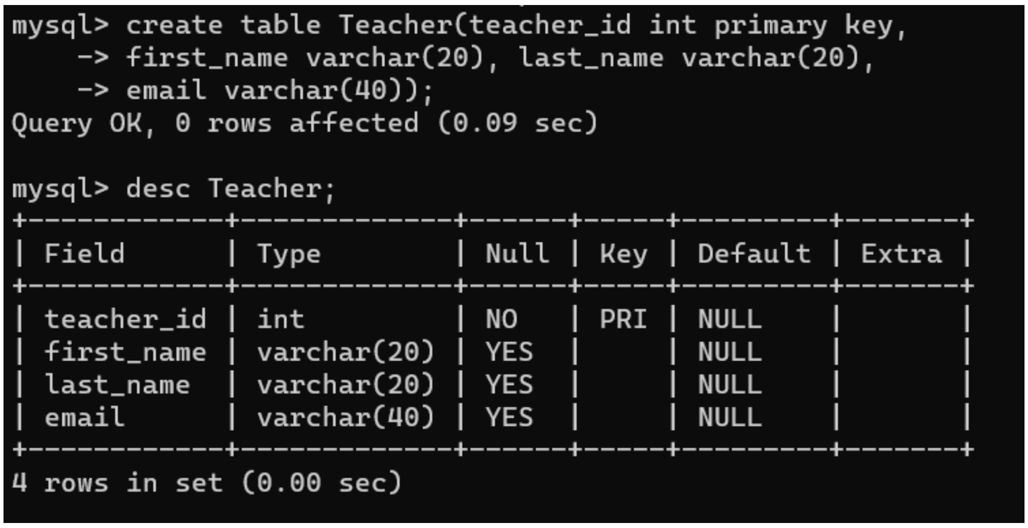
```
mysql> create table Courses(course_id int primary key,
    -> course_name varchar(40),
    -> credits int,
    -> teacher_id int not null,
    -> foreign key (teacher_id) references Teacher(teacher_id));
Query OK, 0 rows affected (0.09 sec)

mysql> desc Courses;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| course_id   | int         | NO   | PRI | NULL    |       |
| course_name | varchar(40) | YES  |     | NULL    |       |
| credits     | int         | YES  |     | NULL    |       |
| teacher_id  | int         | NO   | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

C. Enrollments



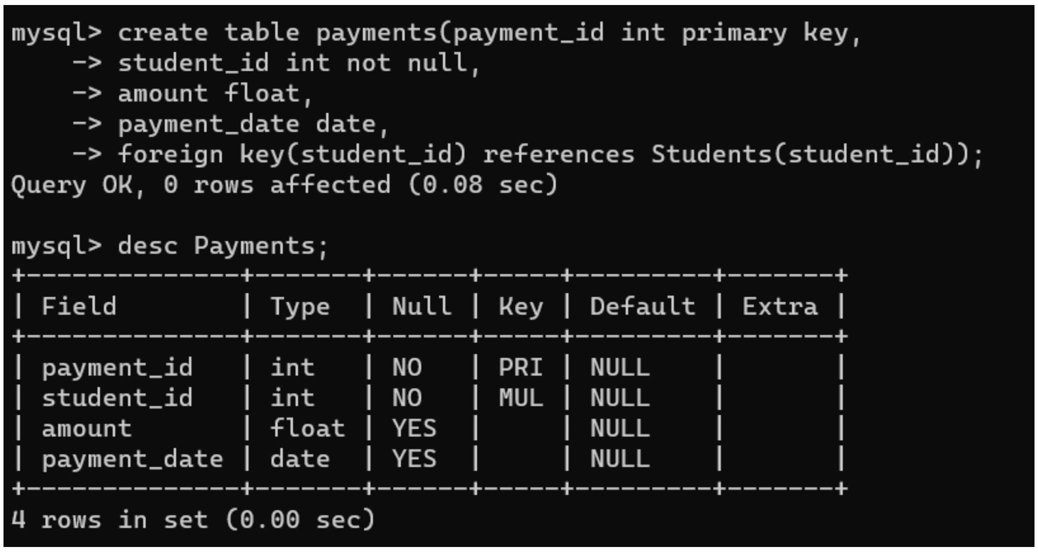
```
mysql> create table Enrollments(enrollment_id int primary key,
    -> student_id int not null,
    -> course_id int not null,
    -> enrollment_date date,
    -> foreign key(student_id) references Students(student_id),
    -> foreign key(course_id) references Courses(course_id));
Query OK, 0 rows affected (0.11 sec)

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

D. Teacher



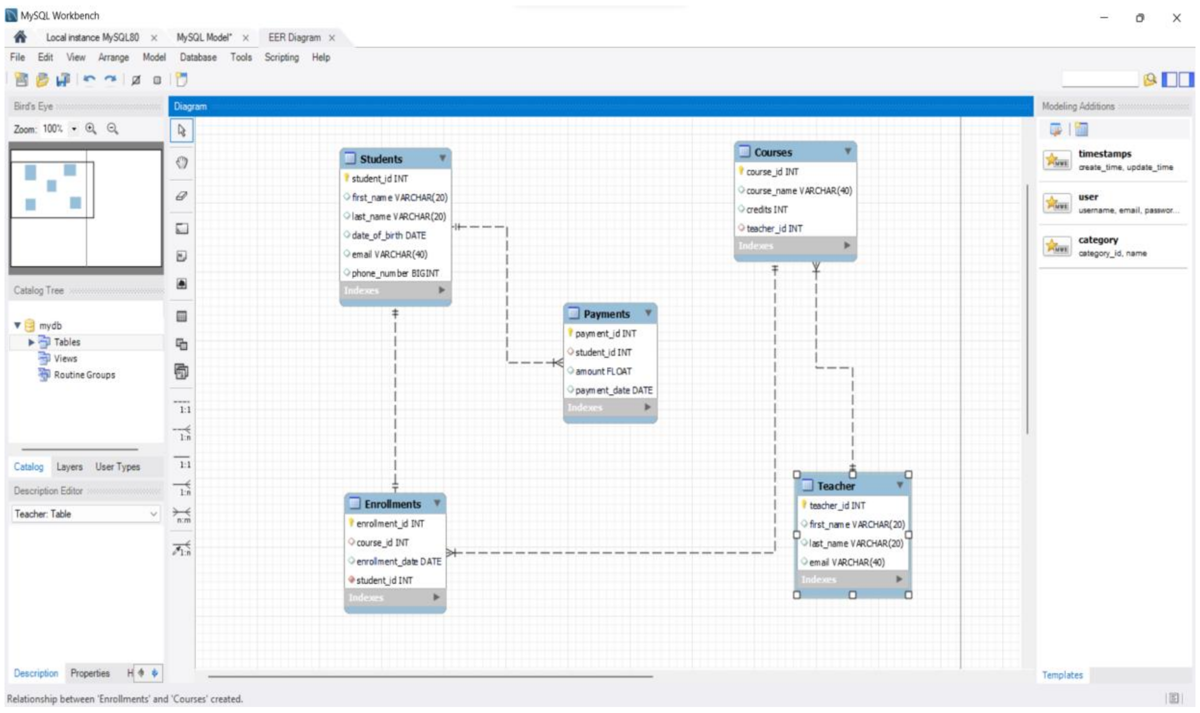
```
mysql> create table Teacher(teacher_id int primary key,
    -> first_name varchar(20), last_name varchar(20),
    -> email varchar(40));
Query OK, 0 rows affected (0.09 sec)

mysql> desc Teacher;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| teacher_id | int         | NO   | PRI | NULL    |       |
| first_name | varchar(20) | YES  |     | NULL    |       |
| last_name  | varchar(20) | YES  |     | NULL    |       |
| email      | varchar(40) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

E. Payments



```
mysql> create table payments(payment_id int primary key,
    -> student_id int not null,
    -> amount float,
    -> payment_date date,
    -> foreign key(student_id) references Students(student_id));
Query OK, 0 rows affected (0.08 sec)

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

Question 3:- Create an ERD for the database.

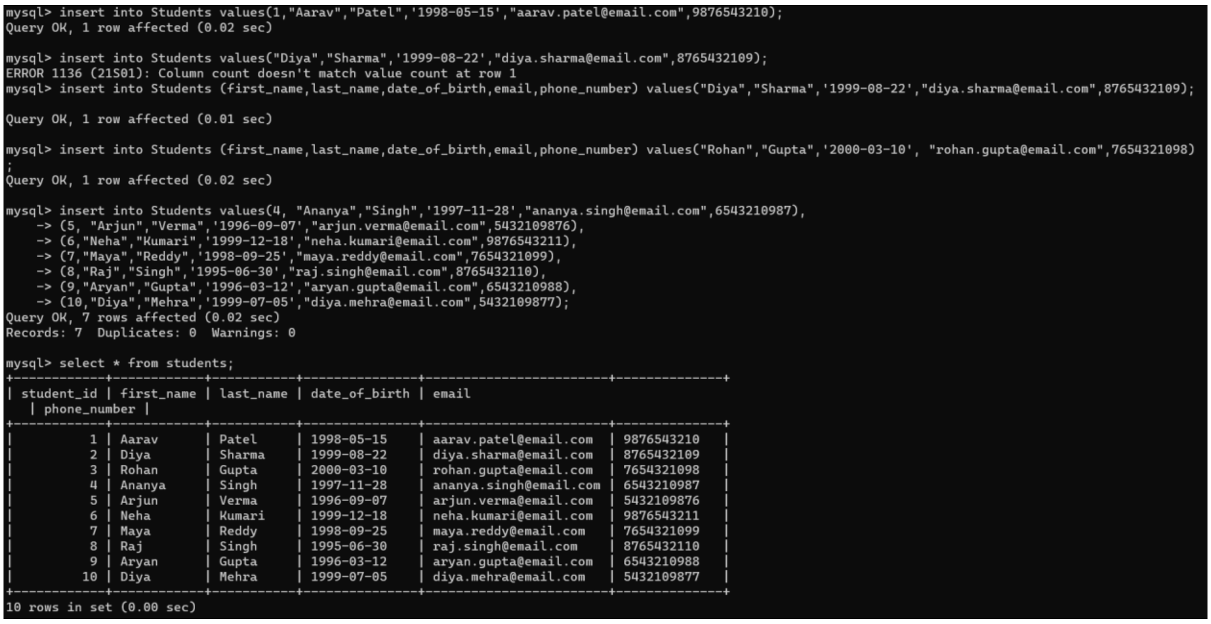


Relationship between 'Enrollments' and 'Courses' created.

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

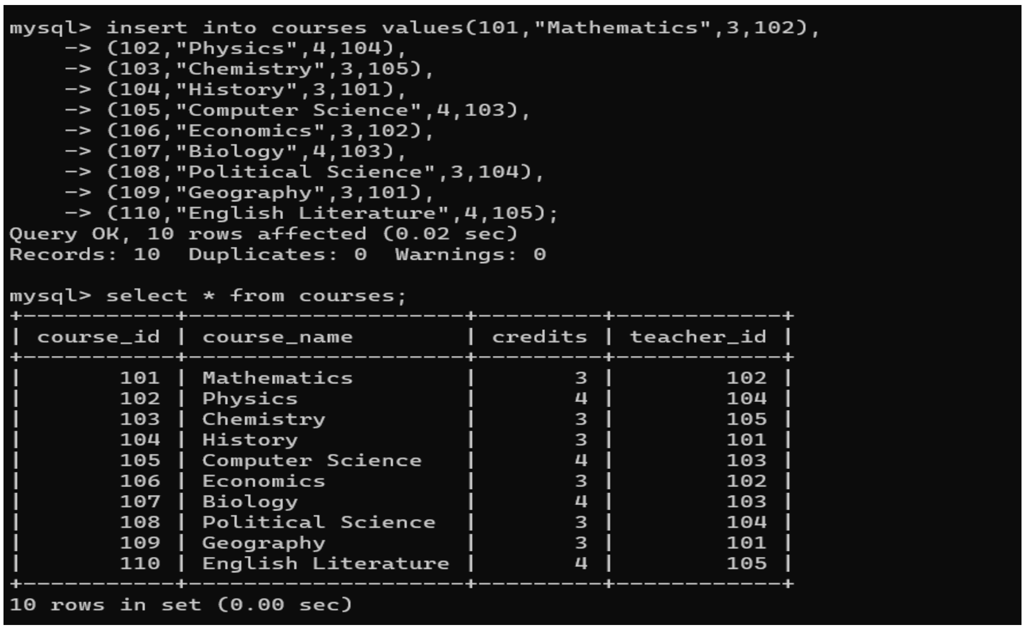
i. Students

Note:- In second query you might see I've tried to insert a tuple without any student_id i.e. primary key. That's because I've set student_id as auto increment.



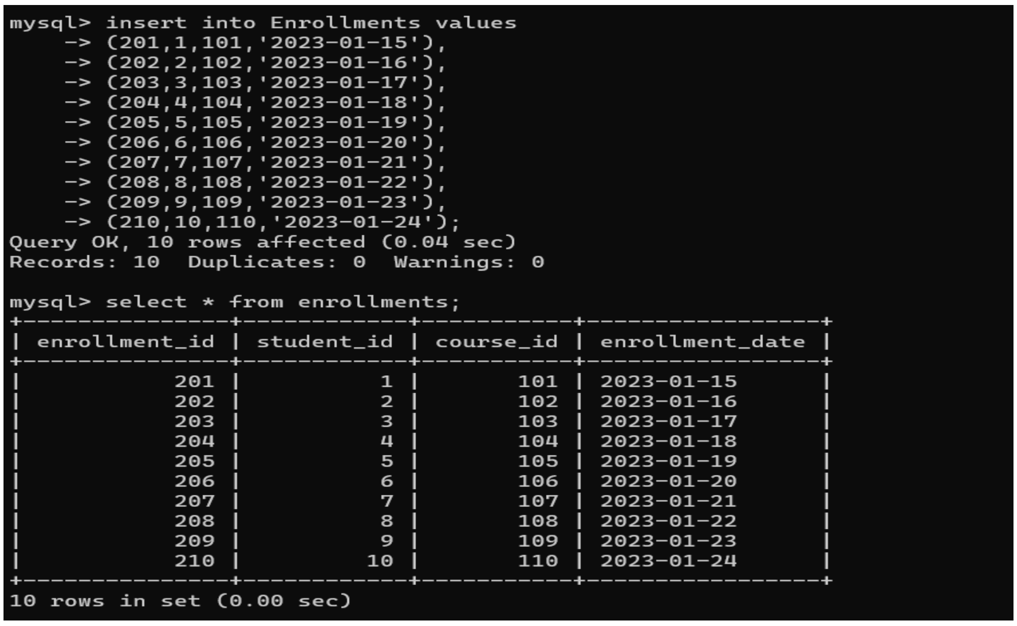
```
mysql> insert into Students values(1,"Aarav","Patel",'1998-05-15',"aarav.patel@email.com",9876543210);
Query OK, 1 row affected (0.02 sec)

mysql> insert into Students values("Diya","Sharma",'1999-08-22',"diya.sharma@email.com",8765432109);
ERROR 1136 (21S01): Column count doesn't match value count at row 1
mysql> insert into Students (first_name,last_name,date_of_birth,email,phone_number) values("Diya","Sharma",'1999-08-22',"diya.sharma@email.com",8765432109);

Query OK, 1 row affected (0.01 sec)

mysql> insert into Students (first_name,last_name,date_of_birth,email,phone_number) values("Rohan","Gupta",'2000-03-10', "rohan.gupta@email.com",7654321098)
;
Query OK, 1 row affected (0.02 sec)

mysql> insert into Students values(4, "Ananya","Singh",'1997-11-28',"ananya.singh@email.com",6543210987),
    -> (5, "Arjun","Verma",'1996-09-07',"arjun.verma@email.com",5432109876),
    -> (6,"Neha","Kumari",'1999-12-18',"neha.kumari@email.com",9876543211),
    -> (7,"Maya","Reddy",'1998-09-25',"maya.reddy@email.com",7654321099),
    -> (8,"Raj","Singh",'1995-06-30',"raj.singh@email.com",8765432110),
    -> (9,"Aryan","Gupta",'1996-03-12',"aryan.gupta@email.com",6543210988),
    -> (10,"Diya","Mehra",'1999-07-05',"diya.mehra@email.com",5432109877);
Query OK, 7 rows affected (0.02 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> select * from students;
+------------+------------+-----------+---------------+------------------------+--------------+
| student_id | first_name | last_name | date_of_birth | email
   | phone_number |
+------------+------------+-----------+---------------+------------------------+--------------+
|          1 | Aarav      | Patel     | 1998-05-15    | aarav.patel@email.com  | 9876543210   |
|          2 | Diya       | Sharma    | 1999-08-22    | diya.sharma@email.com  | 8765432109   |
|          3 | Rohan      | Gupta     | 2000-03-10    | rohan.gupta@email.com  | 7654321098   |
|          4 | Ananya     | Singh     | 1997-11-28    | ananya.singh@email.com | 6543210987   |
|          5 | Arjun      | Verma     | 1996-09-07    | arjun.verma@email.com  | 5432109876   |
|          6 | Neha       | Kumari    | 1999-12-18    | neha.kumari@email.com  | 9876543211   |
|          7 | Maya       | Reddy     | 1998-09-25    | maya.reddy@email.com   | 7654321099   |
|          8 | Raj        | Singh     | 1995-06-30    | raj.singh@email.com    | 8765432110   |
|          9 | Aryan      | Gupta     | 1996-03-12    | aryan.gupta@email.com  | 6543210988   |
|         10 | Diya       | Mehra     | 1999-07-05    | diya.mehra@email.com   | 5432109877   |
+------------+------------+-----------+---------------+------------------------+--------------+
10 rows in set (0.00 sec)
```

ii. Courses



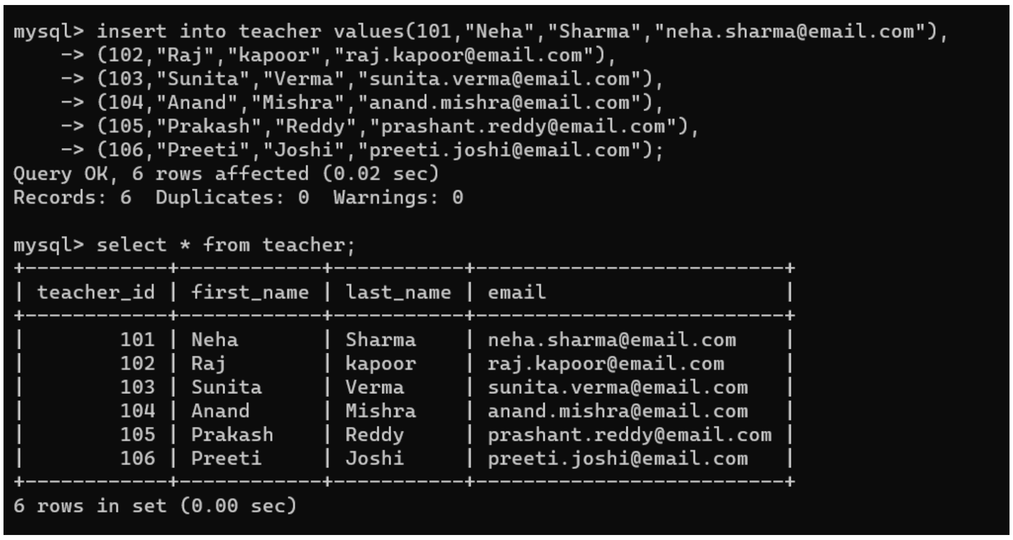
```
mysql> insert into courses values(101,"Mathematics",3,102),
    -> (102,"Physics",4,104),
    -> (103,"Chemistry",3,105),
    -> (104,"History",3,101),
    -> (105,"Computer Science",4,103),
    -> (106,"Economics",3,102),
    -> (107,"Biology",4,103),
    -> (108,"Political Science",3,104),
    -> (109,"Geography",3,101),
    -> (110,"English Literature",4,105);
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from courses;
+-----------+--------------------+---------+------------+
| course_id | course_name        | credits | teacher_id |
+-----------+--------------------+---------+------------+
|       101 | Mathematics        |       3 |        102 |
|       102 | Physics            |       4 |        104 |
|       103 | Chemistry          |       3 |        105 |
|       104 | History            |       3 |        101 |
|       105 | Computer Science   |       4 |        103 |
|       106 | Economics          |       3 |        102 |
|       107 | Biology            |       4 |        103 |
|       108 | Political Science  |       3 |        104 |
|       109 | Geography          |       3 |        101 |
|       110 | English Literature |       4 |        105 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)
```

iii. Enrollments



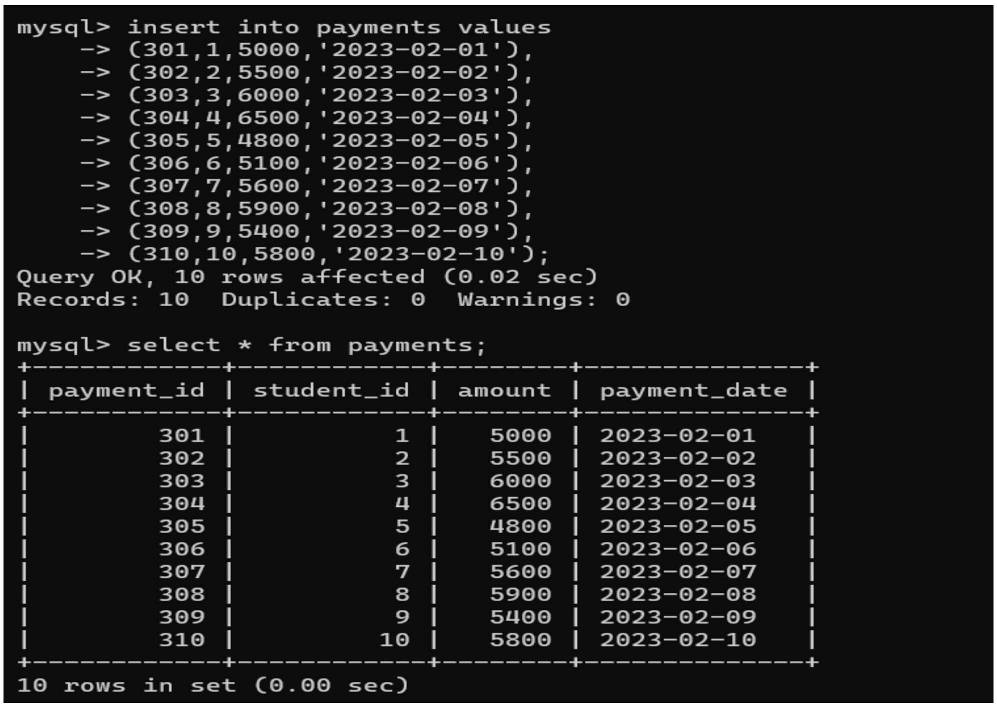
```
mysql> insert into Enrollments values
    -> (201,1,101,'2023-01-15'),
    -> (202,2,102,'2023-01-16'),
    -> (203,3,103,'2023-01-17'),
    -> (204,4,104,'2023-01-18'),
    -> (205,5,105,'2023-01-19'),
    -> (206,6,106,'2023-01-20'),
    -> (207,7,107,'2023-01-21'),
    -> (208,8,108,'2023-01-22'),
    -> (209,9,109,'2023-01-23'),
    -> (210,10,110,'2023-01-24');
Query OK, 10 rows affected (0.04 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|           201 |          1 |       101 | 2023-01-15      |
|           202 |          2 |       102 | 2023-01-16      |
|           203 |          3 |       103 | 2023-01-17      |
|           204 |          4 |       104 | 2023-01-18      |
|           205 |          5 |       105 | 2023-01-19      |
|           206 |          6 |       106 | 2023-01-20      |
|           207 |          7 |       107 | 2023-01-21      |
|           208 |          8 |       108 | 2023-01-22      |
|           209 |          9 |       109 | 2023-01-23      |
|           210 |         10 |       110 | 2023-01-24      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)
```

iv. Teacher



```
mysql> insert into teacher values(101,"Neha","Sharma","neha.sharma@email.com"),
    -> (102,"Raj","kapoor","raj.kapoor@email.com"),
    -> (103,"Sunita","Verma","sunita.verma@email.com"),
    -> (104,"Anand","Mishra","anand.mishra@email.com"),
    -> (105,"Prakash","Reddy","prashant.reddy@email.com"),
    -> (106,"Preeti","Joshi","preeti.joshi@email.com");
Query OK, 6 rows affected (0.02 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> select * from teacher;
+------------+------------+-----------+--------------------------+
| teacher_id | first_name | last_name | email                    |
+------------+------------+-----------+--------------------------+
|        101 | Neha       | Sharma    | neha.sharma@email.com    |
|        102 | Raj        | kapoor    | raj.kapoor@email.com     |
|        103 | Sunita     | Verma     | sunita.verma@email.com   |
|        104 | Anand      | Mishra    | anand.mishra@email.com   |
|        105 | Prakash    | Reddy     | prashant.reddy@email.com |
|        106 | Preeti     | Joshi     | preeti.joshi@email.com   |
+------------+------------+-----------+--------------------------+
6 rows in set (0.00 sec)
```

v. Payments

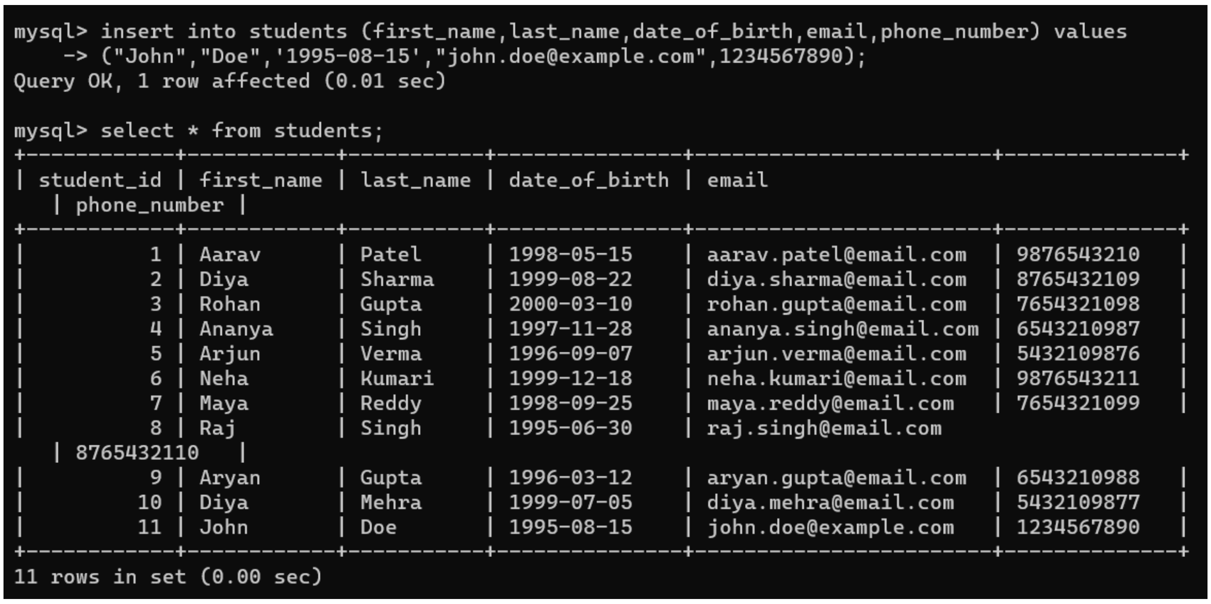


```
mysql> insert into payments values
    -> (301,1,5000,'2023-02-01'),
    -> (302,2,5500,'2023-02-02'),
    -> (303,3,6000,'2023-02-03'),
    -> (304,4,6500,'2023-02-04'),
    -> (305,5,4800,'2023-02-05'),
    -> (306,6,5100,'2023-02-06'),
    -> (307,7,5600,'2023-02-07'),
    -> (308,8,5900,'2023-02-08'),
    -> (309,9,5400,'2023-02-09'),
    -> (310,10,5800,'2023-02-10');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|        301 |          1 |   5000 | 2023-02-01   |
|        302 |          2 |   5500 | 2023-02-02   |
|        303 |          3 |   6000 | 2023-02-03   |
|        304 |          4 |   6500 | 2023-02-04   |
|        305 |          5 |   4800 | 2023-02-05   |
|        306 |          6 |   5100 | 2023-02-06   |
|        307 |          7 |   5600 | 2023-02-07   |
|        308 |          8 |   5900 | 2023-02-08   |
|        309 |          9 |   5400 | 2023-02-09   |
|        310 |         10 |   5800 | 2023-02-10   |
+------------+------------+--------+--------------+
10 rows in set (0.00 sec)
```

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

Question 1:- Write an SQL query to insert a new student into the "Students" table with the following details:

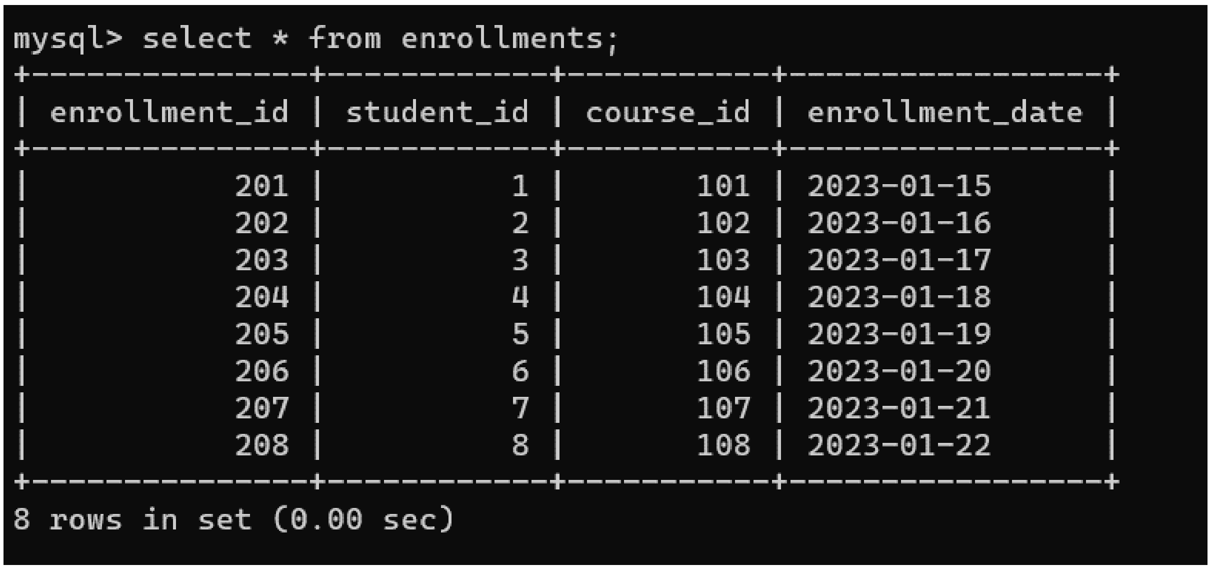
a. First Name: John
b. Last Name: Doe
c. Date of Birth: 1995-08-15
d. Email: john.doe@example.com
e. Phone Number: 1234567890



```
mysql> insert into students (first_name,last_name,date_of_birth,email,phone_number) values
    -> ("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
   | phone_number |
+------------+------------+-----------+---------------+-------------------------+--------------+
|          1 | Aarav      | Patel     | 1998-05-15    | aarav.patel@email.com   | 9876543210   |
|          2 | Diya       | Sharma    | 1999-08-22    | diya.sharma@email.com   | 8765432109   |
|          3 | Rohan      | Gupta     | 2000-03-10    | rohan.gupta@email.com   | 7654321098   |
|          4 | Ananya     | Singh     | 1997-11-28    | ananya.singh@email.com  | 6543210987   |
|          5 | Arjun      | Verma     | 1996-09-07    | arjun.verma@email.com   | 5432109876   |
|          6 | Neha       | Kumari    | 1999-12-18    | neha.kumari@email.com   | 9876543211   |
|          7 | Maya       | Reddy     | 1998-09-25    | maya.reddy@email.com    | 7654321099   |
|          8 | Raj        | Singh     | 1995-06-30    | raj.singh@email.com
   | 8765432110   |
|          9 | Aryan      | Gupta     | 1996-03-12    | aryan.gupta@email.com   | 6543210988   |
|         10 | Diya       | Mehra     | 1999-07-05    | diya.mehra@email.com    | 5432109877   |
|         11 | John       | Doe       | 1995-08-15    | john.doe@example.com    | 1234567890   |
+------------+------------+-----------+---------------+-------------------------+--------------+
11 rows in set (0.00 sec)
```

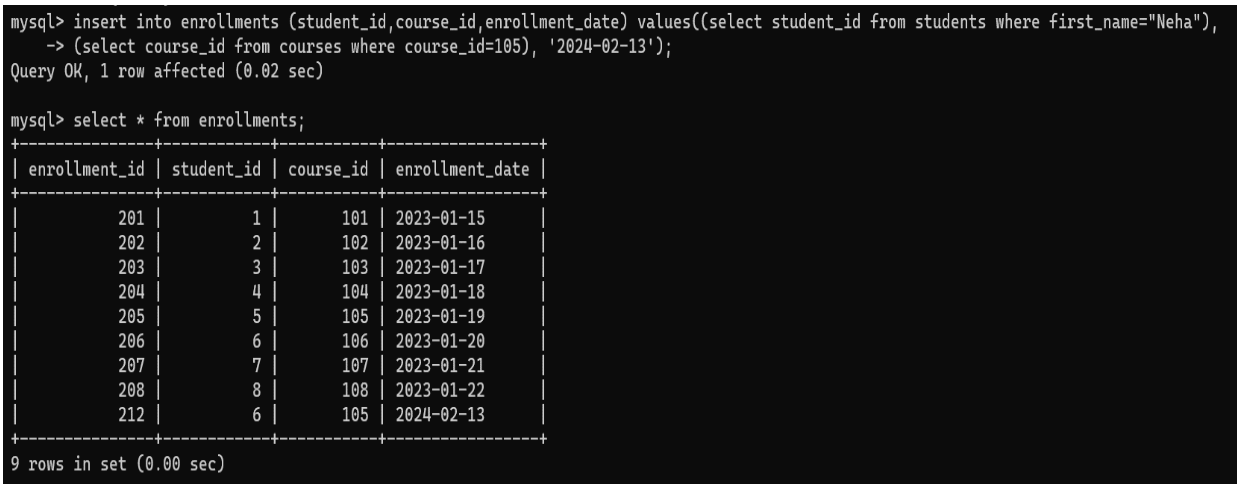
Question 2:- Write an SQL query to enroll a student in a course. Choose an existing student and course and insert a record into the "Enrollments" table with the enrollment date.

Before:-



```
mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|           201 |          1 |       101 | 2023-01-15      |
|           202 |          2 |       102 | 2023-01-16      |
|           203 |          3 |       103 | 2023-01-17      |
|           204 |          4 |       104 | 2023-01-18      |
|           205 |          5 |       105 | 2023-01-19      |
|           206 |          6 |       106 | 2023-01-20      |
|           207 |          7 |       107 | 2023-01-21      |
|           208 |          8 |       108 | 2023-01-22      |
+---------------+------------+-----------+-----------------+
8 rows in set (0.00 sec)
```

After:-

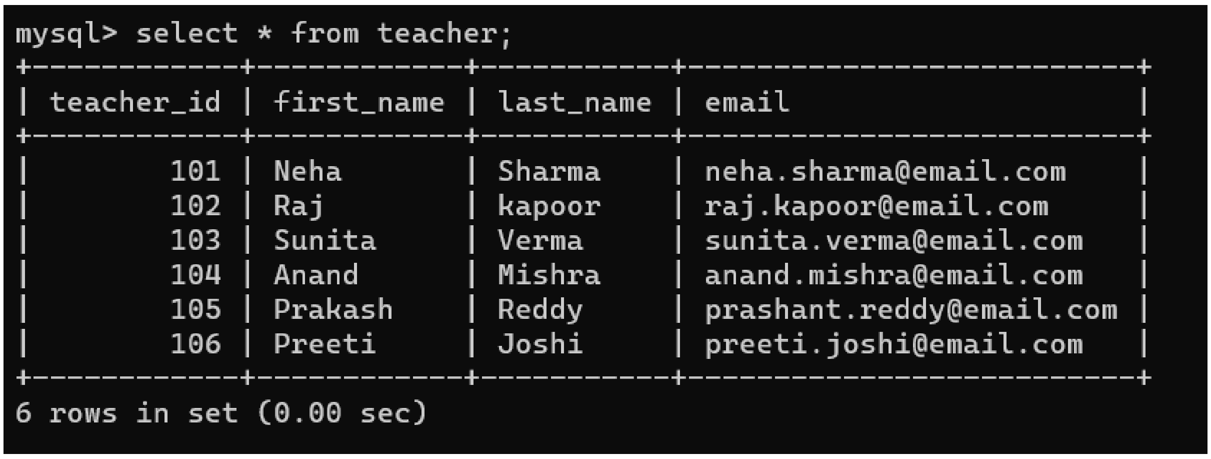


```
mysql> insert into enrollments (student_id,course_id,enrollment_date) values((select student_id from students where first_name="Neha"),
    -> (select course_id from courses where course_id=105), '2024-02-13');
Query OK, 1 row affected (0.02 sec)

mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|           201 |          1 |       101 | 2023-01-15      |
|           202 |          2 |       102 | 2023-01-16      |
|           203 |          3 |       103 | 2023-01-17      |
|           204 |          4 |       104 | 2023-01-18      |
|           205 |          5 |       105 | 2023-01-19      |
|           206 |          6 |       106 | 2023-01-20      |
|           207 |          7 |       107 | 2023-01-21      |
|           208 |          8 |       108 | 2023-01-22      |
|           212 |          6 |       105 | 2024-02-13      |
+---------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)
```

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

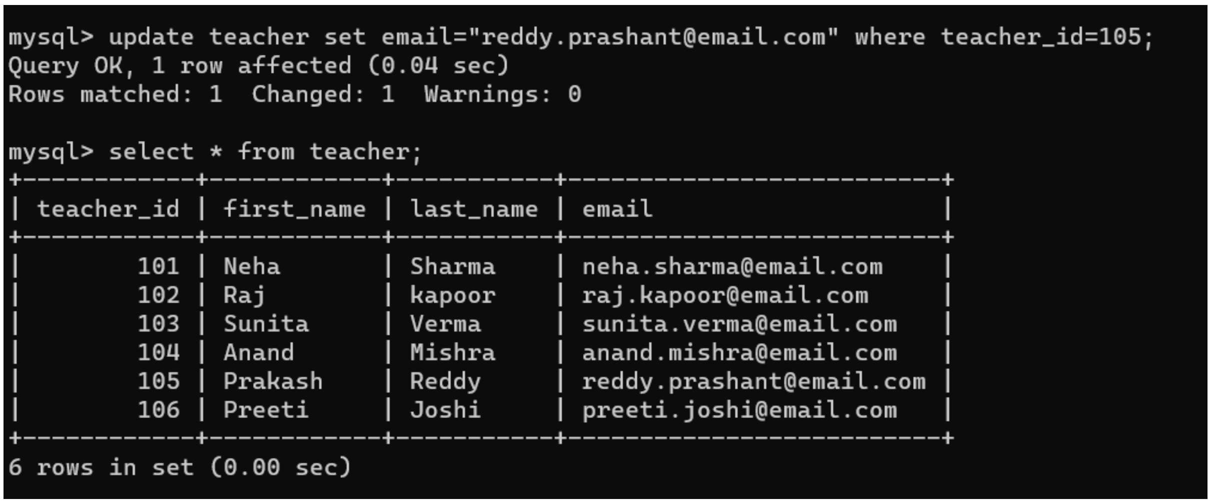
Note:- I’ve changed the address of Prakash Reddy teacher_id=105

Before:-



```
mysql> select * from teacher;
+------------+------------+-----------+---------------------------+
| teacher_id | first_name | last_name | email                     |
+------------+------------+-----------+---------------------------+
|        101 | Neha       | Sharma    | neha.sharma@email.com     |
|        102 | Raj        | kapoor    | raj.kapoor@email.com      |
|        103 | Sunita     | Verma     | sunita.verma@email.com    |
|        104 | Anand      | Mishra    | anand.mishra@email.com    |
|        105 | Prakash    | Reddy     | prashant.reddy@email.com  |
|        106 | Preeti     | Joshi     | preeti.joshi@email.com    |
+------------+------------+-----------+---------------------------+
6 rows in set (0.00 sec)
```

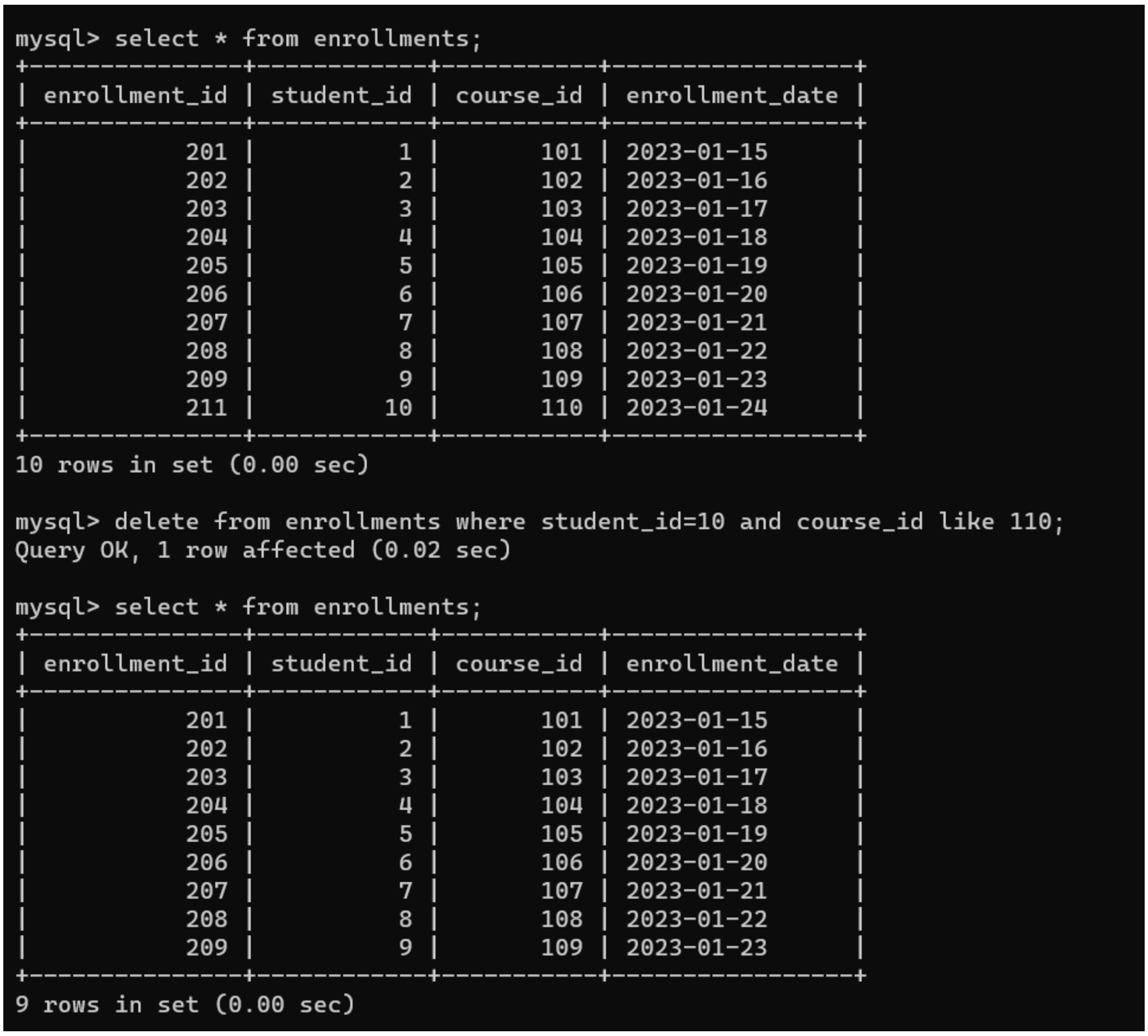
After :-



```
mysql> update teacher set email="reddy.prashant@email.com" where teacher_id=105;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from teacher;
+------------+------------+-----------+---------------------------+
| teacher_id | first_name | last_name | email                     |
+------------+------------+-----------+---------------------------+
|        101 | Neha       | Sharma    | neha.sharma@email.com     |
|        102 | Raj        | kapoor    | raj.kapoor@email.com      |
|        103 | Sunita     | Verma     | sunita.verma@email.com    |
|        104 | Anand      | Mishra    | anand.mishra@email.com    |
|        105 | Prakash    | Reddy     | reddy.prashant@email.com  |
|        106 | Preeti     | Joshi     | preeti.joshi@email.com    |
+------------+------------+-----------+---------------------------+
6 rows in set (0.00 sec)
```

Question 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.

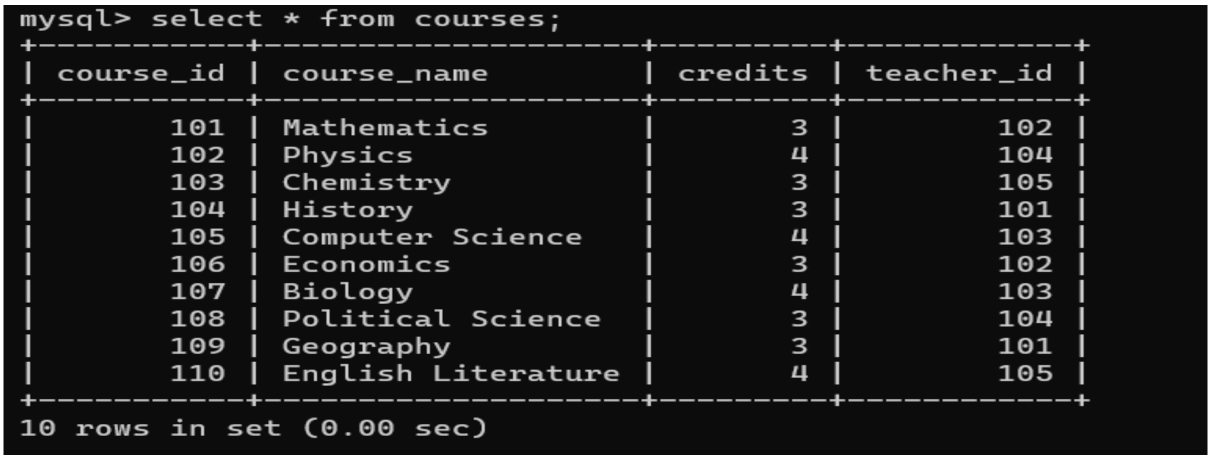
Note: In this query I've delete from enrollments where student_id = 10 and course_id=110. Here is the before and after of table.



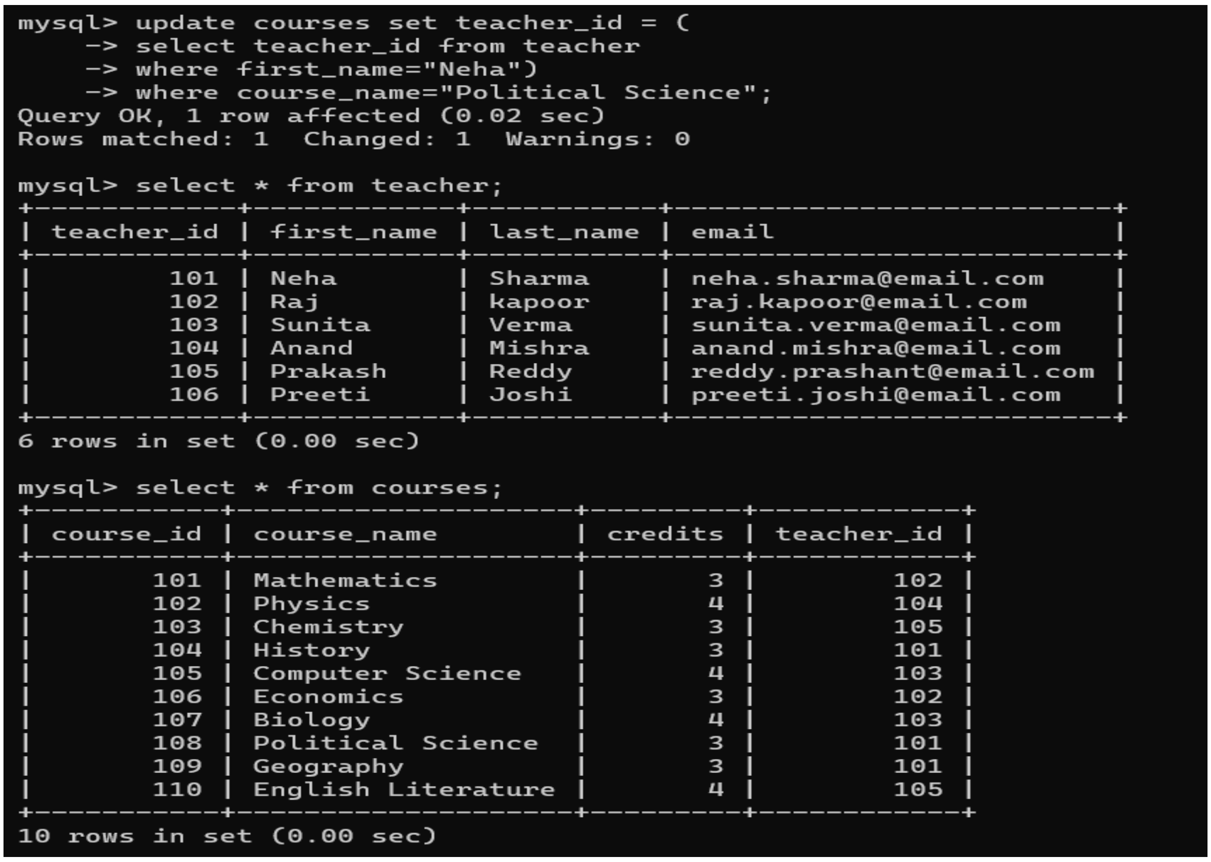
```
mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|           201 |          1 |       101 | 2023-01-15      |
|           202 |          2 |       102 | 2023-01-16      |
|           203 |          3 |       103 | 2023-01-17      |
|           204 |          4 |       104 | 2023-01-18      |
|           205 |          5 |       105 | 2023-01-19      |
|           206 |          6 |       106 | 2023-01-20      |
|           207 |          7 |       107 | 2023-01-21      |
|           208 |          8 |       108 | 2023-01-22      |
|           209 |          9 |       109 | 2023-01-23      |
|           211 |         10 |       110 | 2023-01-24      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)

mysql> delete from enrollments where student_id=10 and course_id like 110;
Query OK, 1 row affected (0.02 sec)

mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|           201 |          1 |       101 | 2023-01-15      |
|           202 |          2 |       102 | 2023-01-16      |
|           203 |          3 |       103 | 2023-01-17      |
|           204 |          4 |       104 | 2023-01-18      |
|           205 |          5 |       105 | 2023-01-19      |
|           206 |          6 |       106 | 2023-01-20      |
|           207 |          7 |       107 | 2023-01-21      |
|           208 |          8 |       108 | 2023-01-22      |
|           209 |          9 |       109 | 2023-01-23      |
+---------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)
```

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

Note: Here is the before and after table of courses where I've set Neha as Political Science teacher.



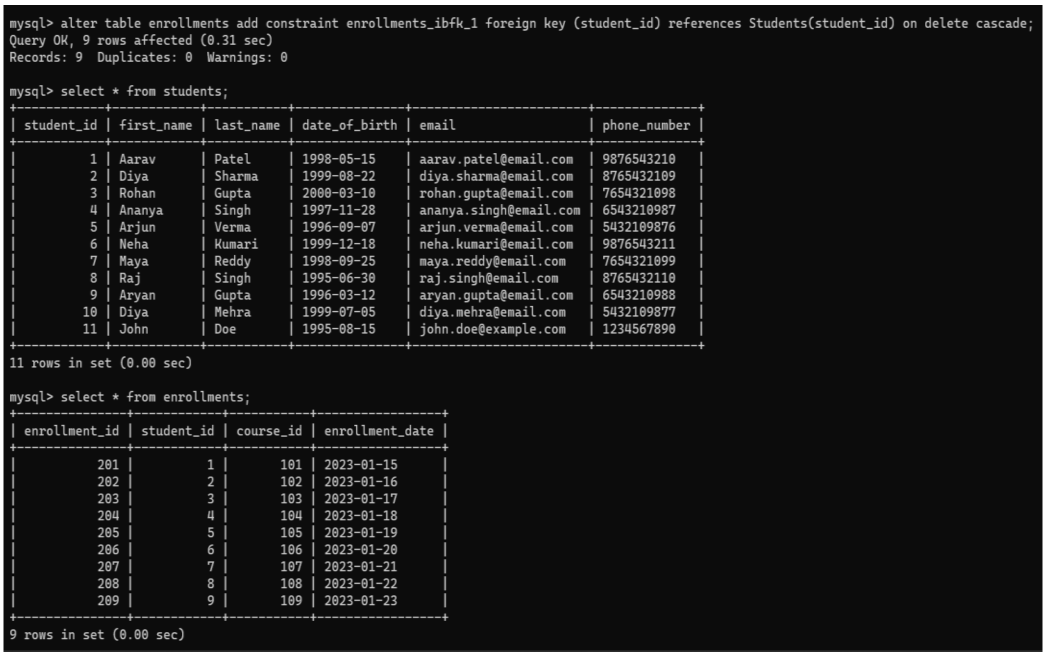
```
mysql> select * from courses;
+-----------+--------------------+---------+------------+
| course_id | course_name        | credits | teacher_id |
+-----------+--------------------+---------+------------+
|       101 | Mathematics        |       3 |        102 |
|       102 | Physics            |       4 |        104 |
|       103 | Chemistry          |       3 |        105 |
|       104 | History            |       3 |        101 |
|       105 | Computer Science   |       4 |        103 |
|       106 | Economics          |       3 |        102 |
|       107 | Biology            |       4 |        103 |
|       108 | Political Science  |       3 |        104 |
|       109 | Geography          |       3 |        101 |
|       110 | English Literature |       4 |        105 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)
```



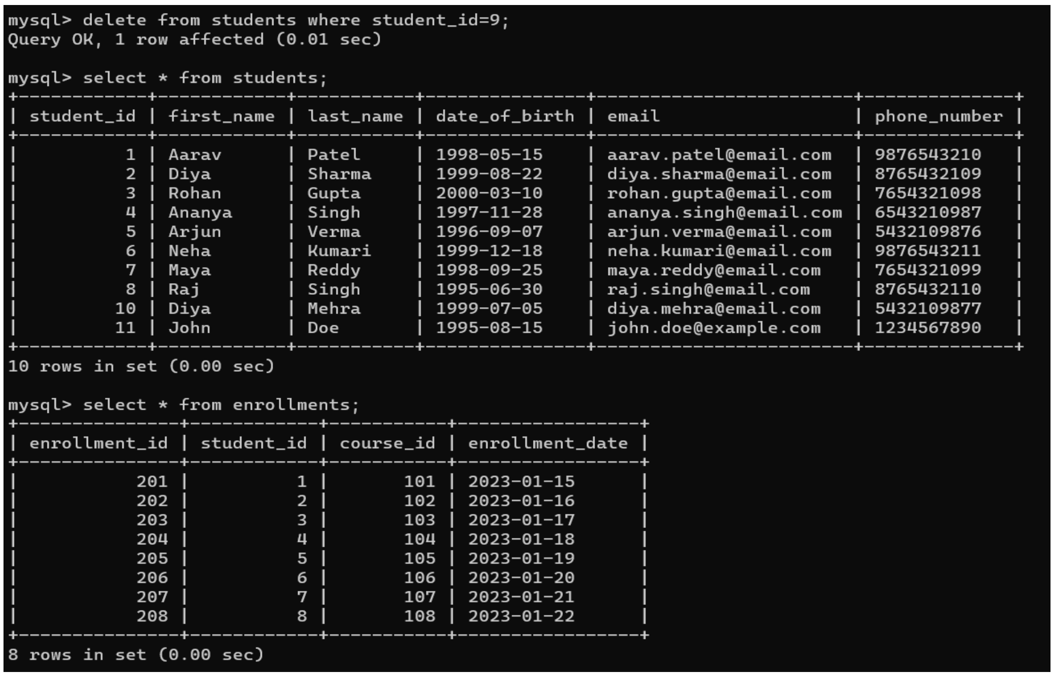
```
mysql> update courses set teacher_id = (
    -> select teacher_id from teacher
    -> where first_name="Neha")
    -> where course_name="Political Science";
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from teacher;
+------------+------------+-----------+---------------------------+
| teacher_id | first_name | last_name | email                     |
+------------+------------+-----------+---------------------------+
|        101 | Neha       | Sharma    | neha.sharma@email.com     |
|        102 | Raj        | kapoor    | raj.kapoor@email.com      |
|        103 | Sunita     | Verma     | sunita.verma@email.com    |
|        104 | Anand      | Mishra    | anand.mishra@email.com    |
|        105 | Prakash    | Reddy     | reddy.prashant@email.com  |
|        106 | Preeti     | Joshi     | preeti.joshi@email.com    |
+------------+------------+-----------+---------------------------+
6 rows in set (0.00 sec)

mysql> select * from courses;
+-----------+--------------------+---------+------------+
| course_id | course_name        | credits | teacher_id |
+-----------+--------------------+---------+------------+
|       101 | Mathematics        |       3 |        102 |
|       102 | Physics            |       4 |        104 |
|       103 | Chemistry          |       3 |        105 |
|       104 | History            |       3 |        101 |
|       105 | Computer Science   |       4 |        103 |
|       106 | Economics          |       3 |        102 |
|       107 | Biology            |       4 |        103 |
|       108 | Political Science  |       3 |        101 |
|       109 | Geography          |       3 |        101 |
|       110 | English Literature |       4 |        105 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)
```

Question 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.

Note:- I’ve change the referential integrity and added on delete cascade so that if there is a deletion in parent table it should reflect in child table.

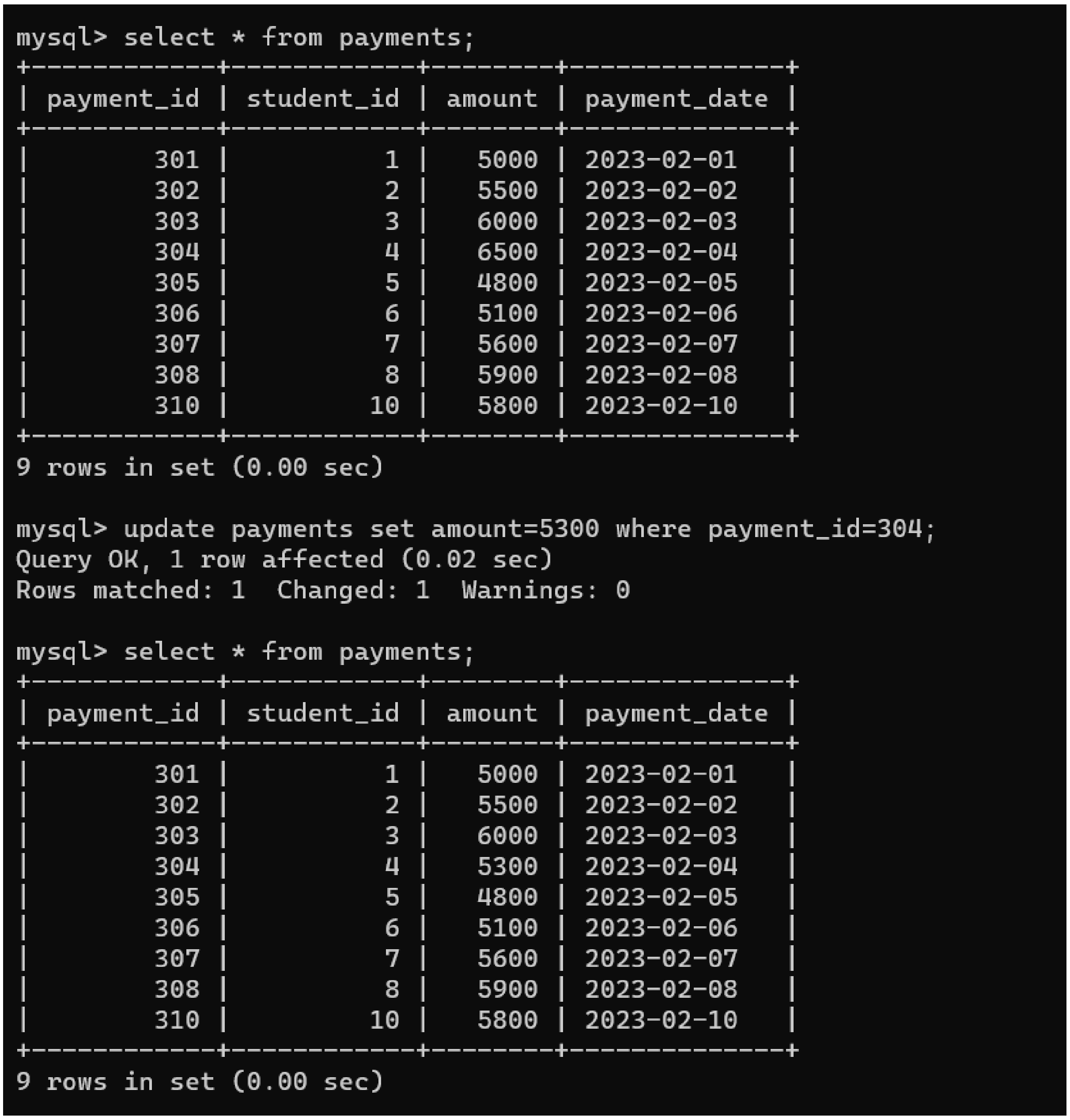


```
mysql> alter table enrollments add constraint enrollments_ibfk_1 foreign key (student_id) references Students(student_id) on delete cascade;
Query OK, 9 rows affected (0.31 sec)
Records: 9  Duplicates: 0  Warnings: 0

mysql> select * from students;
+------------+------------+-----------+---------------+-------------------------+--------------+
| student_id | first_name | last_name | date_of_birth | email                   | phone_number |
+------------+------------+-----------+---------------+-------------------------+--------------+
|          1 | Aarav      | Patel     | 1998-05-15    | aarav.patel@email.com   | 9876543210   |
|          2 | Diya       | Sharma    | 1999-08-22    | diya.sharma@email.com   | 8765432109   |
|          3 | Rohan      | Gupta     | 2000-03-10    | rohan.gupta@email.com   | 7654321098   |
|          4 | Ananya     | Singh     | 1997-11-28    | ananya.singh@email.com  | 6543210987   |
|          5 | Arjun      | Verma     | 1996-09-07    | arjun.verma@email.com   | 5432109876   |
|          6 | Neha       | Kumari    | 1999-12-18    | neha.kumari@email.com   | 9876543211   |
|          7 | Maya       | Reddy     | 1998-09-25    | maya.reddy@email.com    | 7654321099   |
|          8 | Raj        | Singh     | 1995-06-30    | raj.singh@email.com     | 8765432110   |
|          9 | Aryan      | Gupta     | 1996-03-12    | aryan.gupta@email.com   | 6543210988   |
|         10 | Diya       | Mehra     | 1999-07-05    | diya.mehra@email.com    | 5432109877   |
|         11 | John       | Doe       | 1995-08-15    | john.doe@example.com    | 1234567890   |
+------------+------------+-----------+---------------+-------------------------+--------------+
11 rows in set (0.00 sec)

mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|           201 |          1 |       101 | 2023-01-15      |
|           202 |          2 |       102 | 2023-01-16      |
|           203 |          3 |       103 | 2023-01-17      |
|           204 |          4 |       104 | 2023-01-18      |
|           205 |          5 |       105 | 2023-01-19      |
|           206 |          6 |       106 | 2023-01-20      |
|           207 |          7 |       107 | 2023-01-21      |
|           208 |          8 |       108 | 2023-01-22      |
|           209 |          9 |       109 | 2023-01-23      |
+---------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)
```



```
mysql> delete from students where student_id=9;
Query OK, 1 row affected (0.01 sec)

mysql> select * from students;
+------------+------------+-----------+---------------+-------------------------+--------------+
| student_id | first_name | last_name | date_of_birth | email                   | phone_number |
+------------+------------+-----------+---------------+-------------------------+--------------+
|          1 | Aarav      | Patel     | 1998-05-15    | aarav.patel@email.com   | 9876543210   |
|          2 | Diya       | Sharma    | 1999-08-22    | diya.sharma@email.com   | 8765432109   |
|          3 | Rohan      | Gupta     | 2000-03-10    | rohan.gupta@email.com   | 7654321098   |
|          4 | Ananya     | Singh     | 1997-11-28    | ananya.singh@email.com  | 6543210987   |
|          5 | Arjun      | Verma     | 1996-09-07    | arjun.verma@email.com   | 5432109876   |
|          6 | Neha       | Kumari    | 1999-12-18    | neha.kumari@email.com   | 9876543211   |
|          7 | Maya       | Reddy     | 1998-09-25    | maya.reddy@email.com    | 7654321099   |
|          8 | Raj        | Singh     | 1995-06-30    | raj.singh@email.com     | 8765432110   |
|         10 | Diya       | Mehra     | 1999-07-05    | diya.mehra@email.com    | 5432109877   |
|         11 | John       | Doe       | 1995-08-15    | john.doe@example.com    | 1234567890   |
+------------+------------+-----------+---------------+-------------------------+--------------+
10 rows in set (0.00 sec)

mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|           201 |          1 |       101 | 2023-01-15      |
|           202 |          2 |       102 | 2023-01-16      |
|           203 |          3 |       103 | 2023-01-17      |
|           204 |          4 |       104 | 2023-01-18      |
|           205 |          5 |       105 | 2023-01-19      |
|           206 |          6 |       106 | 2023-01-20      |
|           207 |          7 |       107 | 2023-01-21      |
|           208 |          8 |       108 | 2023-01-22      |
+---------------+------------+-----------+-----------------+
8 rows in set (0.00 sec)
```

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

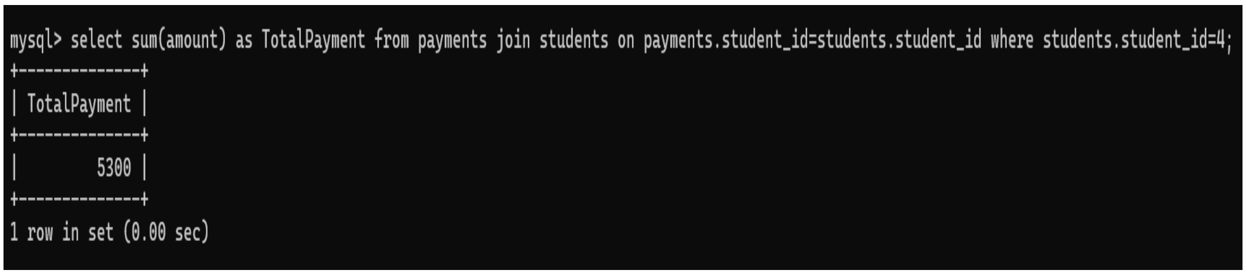


```
mysql> select * from payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|        301 |          1 |   5000 | 2023-02-01   |
|        302 |          2 |   5500 | 2023-02-02   |
|        303 |          3 |   6000 | 2023-02-03   |
|        304 |          4 |   6500 | 2023-02-04   |
|        305 |          5 |   4800 | 2023-02-05   |
|        306 |          6 |   5100 | 2023-02-06   |
|        307 |          7 |   5600 | 2023-02-07   |
|        308 |          8 |   5900 | 2023-02-08   |
|        310 |         10 |   5800 | 2023-02-10   |
+------------+------------+--------+--------------+
9 rows in set (0.00 sec)

mysql> update payments set amount=5300 where payment_id=304;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|        301 |          1 |   5000 | 2023-02-01   |
|        302 |          2 |   5500 | 2023-02-02   |
|        303 |          3 |   6000 | 2023-02-03   |
|        304 |          4 |   5300 | 2023-02-04   |
|        305 |          5 |   4800 | 2023-02-05   |
|        306 |          6 |   5100 | 2023-02-06   |
|        307 |          7 |   5600 | 2023-02-07   |
|        308 |          8 |   5900 | 2023-02-08   |
|        310 |         10 |   5800 | 2023-02-10   |
+------------+------------+--------+--------------+
9 rows in set (0.00 sec)
```

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

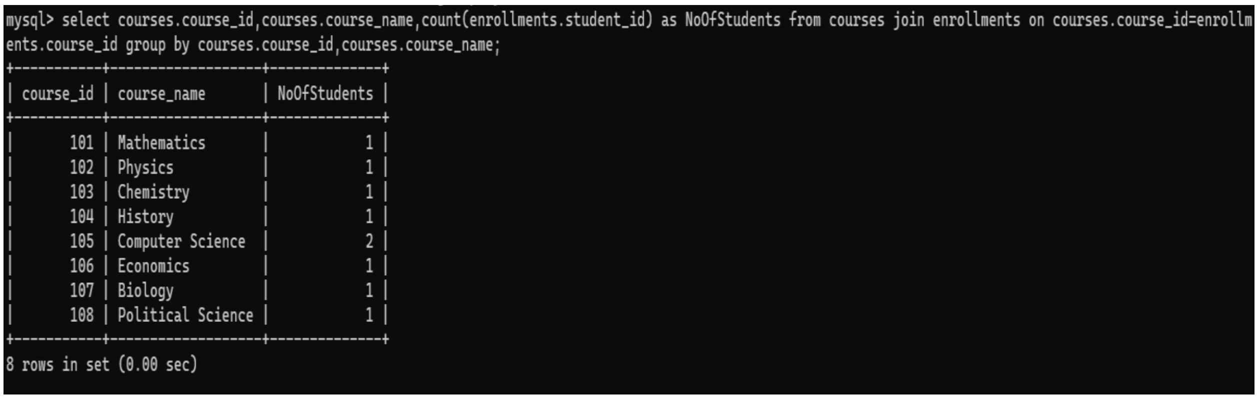
1. Write an SQL query to calculate the total payments made by a specific student. You will need to join the "Payments" table with the "Students" table based on the student's ID.

Note:- I've taken student_id=4



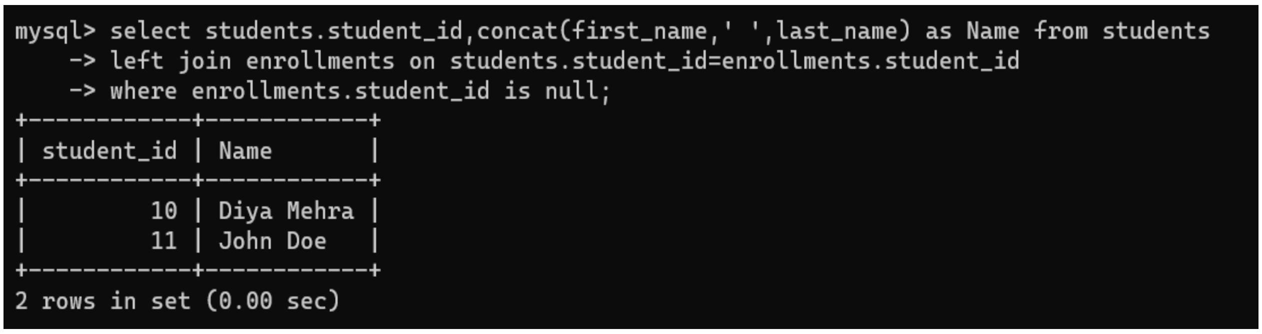
```
mysql> select sum(amount) as TotalPayment from payments join students on payments.student_id=students.student_id where students.student_id=4;
+--------------+
| TotalPayment |
+--------------+
|         5300 |
+--------------+
1 row in set (0.00 sec)
```

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.



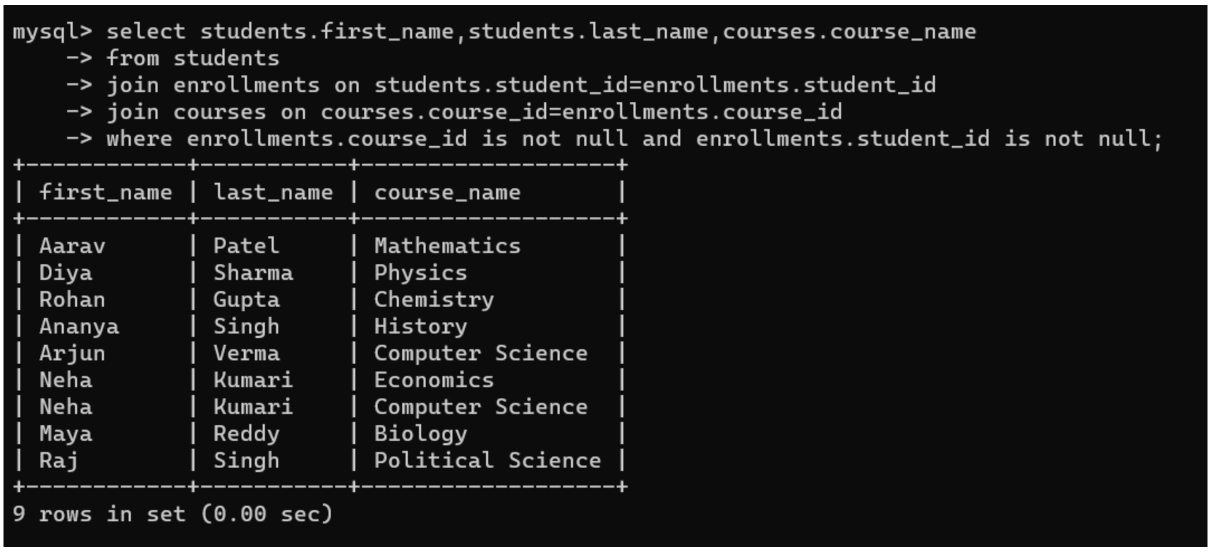
```
mysql> select courses.course_id,courses.course_name,count(enrollments.student_id) as NoOfStudents from courses join enrollments on courses.course_id=enrollm
ents.course_id group by courses.course_id,courses.course_name;
+-----------+-------------------+--------------+
| course_id | course_name       | NoOfStudents |
+-----------+-------------------+--------------+
|       101 | Mathematics       |            1 |
|       102 | Physics           |            1 |
|       103 | Chemistry         |            1 |
|       104 | History           |            1 |
|       105 | Computer Science  |            2 |
|       106 | Economics         |            1 |
|       107 | Biology           |            1 |
|       108 | Political Science |            1 |
+-----------+-------------------+--------------+
8 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.



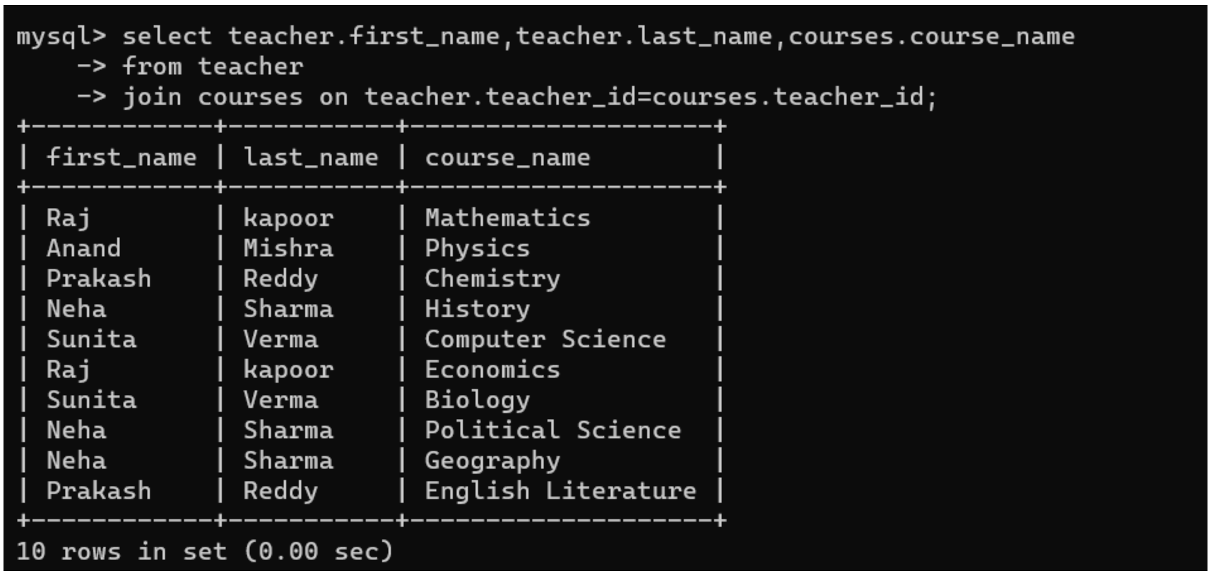
```
mysql> select students.student_id,concat(first_name,' ',last_name) as Name from students
    -> left join enrollments on students.student_id=enrollments.student_id
    -> where enrollments.student_id is null;
+------------+------------+
| student_id | Name       |
+------------+------------+
|         10 | Diya Mehra |
|         11 | John Doe   |
+------------+------------+
2 rows 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.



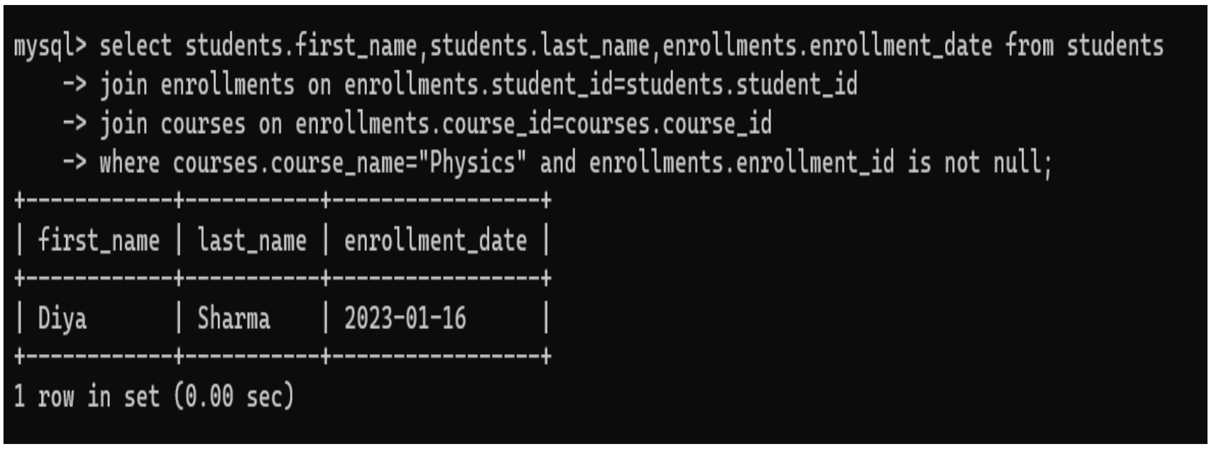
```
mysql> select students.first_name,students.last_name,courses.course_name
    -> from students
    -> join enrollments on students.student_id=enrollments.student_id
    -> join courses on courses.course_id=enrollments.course_id
    -> where enrollments.course_id is not null and enrollments.student_id is not null;
+------------+-----------+-------------------+
| first_name | last_name | course_name       |
+------------+-----------+-------------------+
| Aarav      | Patel     | Mathematics       |
| Diya       | Sharma    | Physics           |
| Rohan      | Gupta     | Chemistry         |
| Ananya     | Singh     | History           |
| Arjun      | Verma     | Computer Science  |
| Neha       | Kumari    | Economics         |
| Neha       | Kumari    | Computer Science  |
| Maya       | Reddy     | Biology           |
| Raj        | Singh     | Political Science |
+------------+-----------+-------------------+
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.



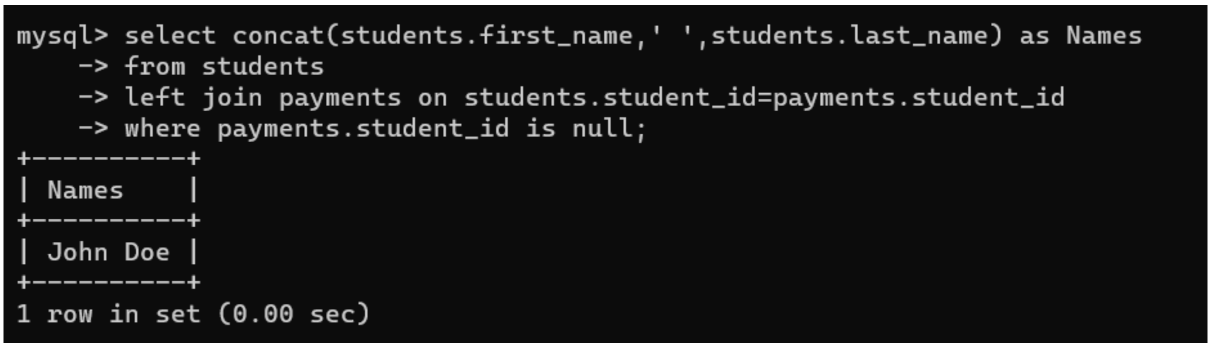
```
mysql> select teacher.first_name,teacher.last_name,courses.course_name
    -> from teacher
    -> join courses on teacher.teacher_id=courses.teacher_id;
+------------+-----------+---------------------+
| first_name | last_name | course_name         |
+------------+-----------+---------------------+
| Raj        | kapoor    | Mathematics         |
| Anand      | Mishra    | Physics             |
| Prakash    | Reddy     | Chemistry           |
| Neha       | Sharma    | History             |
| Sunita     | Verma     | Computer Science    |
| Raj        | kapoor    | Economics           |
| Sunita     | Verma     | Biology             |
| Neha       | Sharma    | Political Science   |
| Neha       | Sharma    | Geography           |
| Prakash    | Reddy     | English Literature  |
+------------+-----------+---------------------+
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.



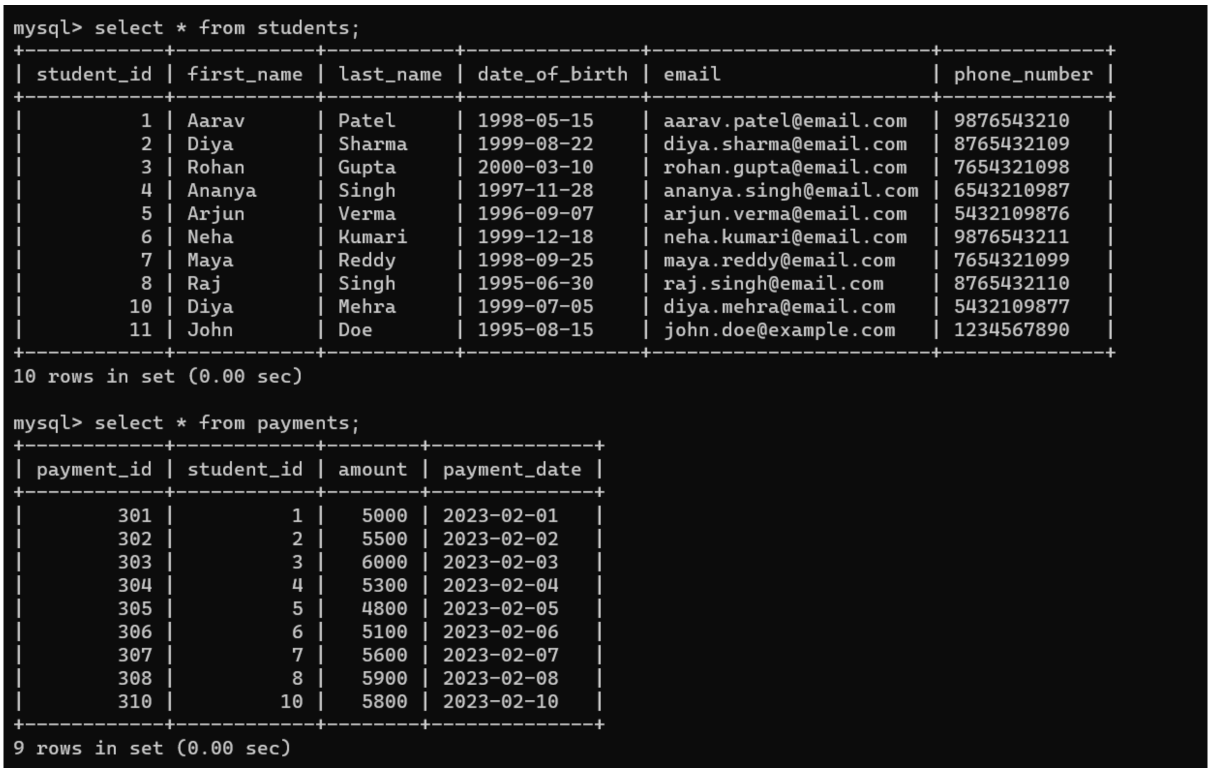
```
mysql> select students.first_name,students.last_name,enrollments.enrollment_date from students
    -> join enrollments on enrollments.student_id=students.student_id
    -> join courses on enrollments.course_id=courses.course_id
    -> where courses.course_name="Physics" and enrollments.enrollment_id is not null;
+------------+-----------+-----------------+
| first_name | last_name | enrollment_date |
+------------+-----------+-----------------+
| Diya       | Sharma    | 2023-01-16      |
+------------+-----------+-----------------+
1 row 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.



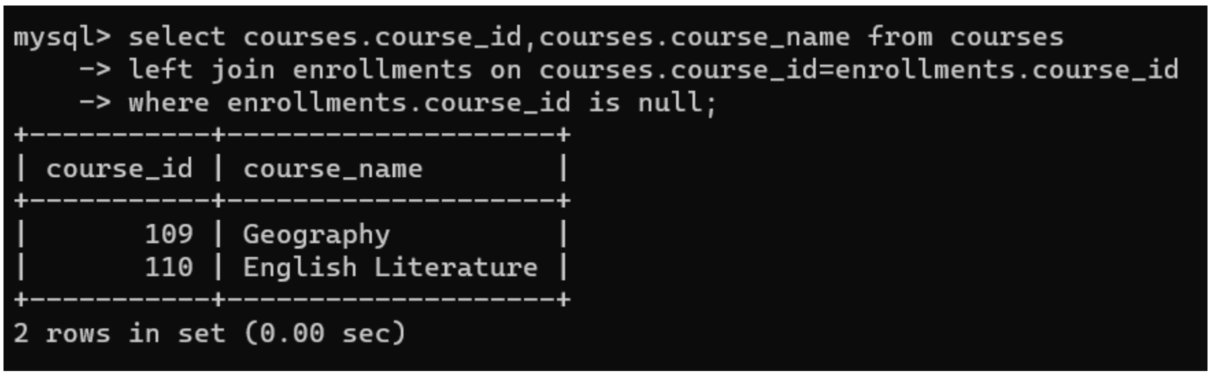
```
mysql> select concat(students.first_name,' ',students.last_name) as Names
    -> from students
    -> left join payments on students.student_id=payments.student_id
    -> where payments.student_id is null;
+----------+
| Names    |
+----------+
| John Doe |
+----------+
1 row in set (0.00 sec)
```

You can see students and payments details below



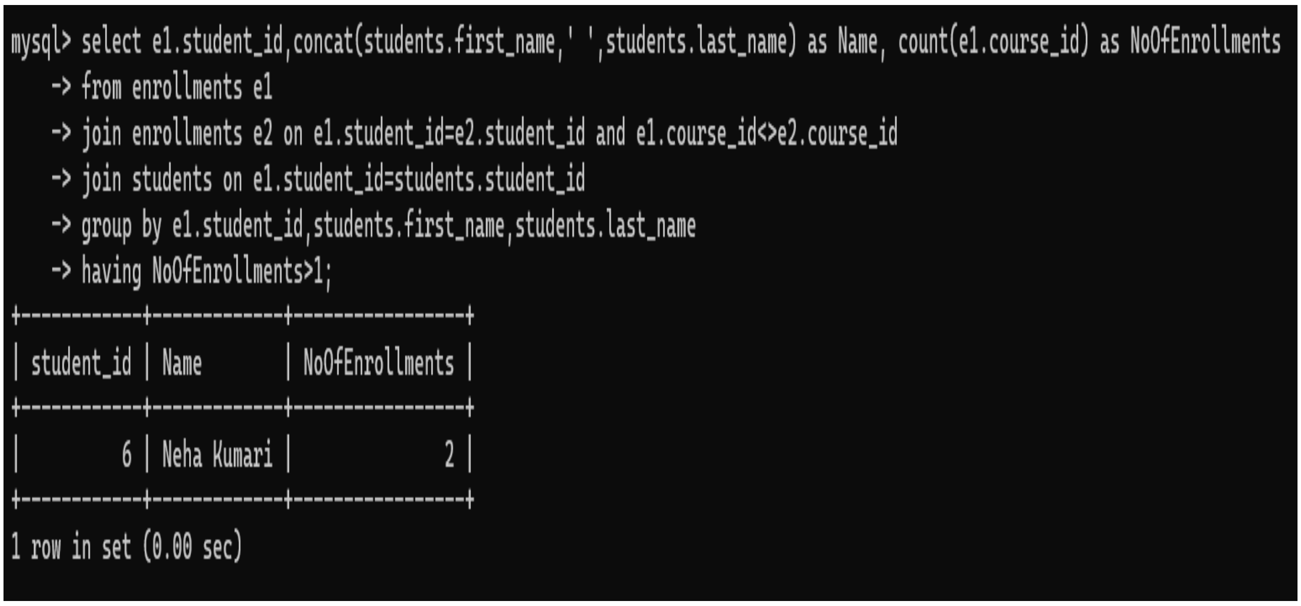
```
mysql> select * from students;
+------------+------------+-----------+---------------+------------------------+--------------+
| student_id | first_name | last_name | date_of_birth | email                  | phone_number |
+------------+------------+-----------+---------------+------------------------+--------------+
|          1 | Aarav      | Patel     | 1998-05-15    | aarav.patel@email.com  | 9876543210   |
|          2 | Diya       | Sharma    | 1999-08-22    | diya.sharma@email.com  | 8765432109   |
|          3 | Rohan      | Gupta     | 2000-03-10    | rohan.gupta@email.com  | 7654321098   |
|          4 | Ananya     | Singh     | 1997-11-28    | ananya.singh@email.com | 6543210987   |
|          5 | Arjun      | Verma     | 1996-09-07    | arjun.verma@email.com  | 5432109876   |
|          6 | Neha       | Kumari    | 1999-12-18    | neha.kumari@email.com  | 9876543211   |
|          7 | Maya       | Reddy     | 1998-09-25    | maya.reddy@email.com   | 7654321099   |
|          8 | Raj        | Singh     | 1995-06-30    | raj.singh@email.com    | 8765432110   |
|         10 | Diya       | Mehra     | 1999-07-05    | diya.mehra@email.com   | 5432109877   |
|         11 | John       | Doe       | 1995-08-15    | john.doe@example.com   | 1234567890   |
+------------+------------+-----------+---------------+------------------------+--------------+
10 rows in set (0.00 sec)

mysql> select * from payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|        301 |          1 |   5000 | 2023-02-01   |
|        302 |          2 |   5500 | 2023-02-02   |
|        303 |          3 |   6000 | 2023-02-03   |
|        304 |          4 |   5300 | 2023-02-04   |
|        305 |          5 |   4800 | 2023-02-05   |
|        306 |          6 |   5100 | 2023-02-06   |
|        307 |          7 |   5600 | 2023-02-07   |
|        308 |          8 |   5900 | 2023-02-08   |
|        310 |         10 |   5800 | 2023-02-10   |
+------------+------------+--------+--------------+
9 rows 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.



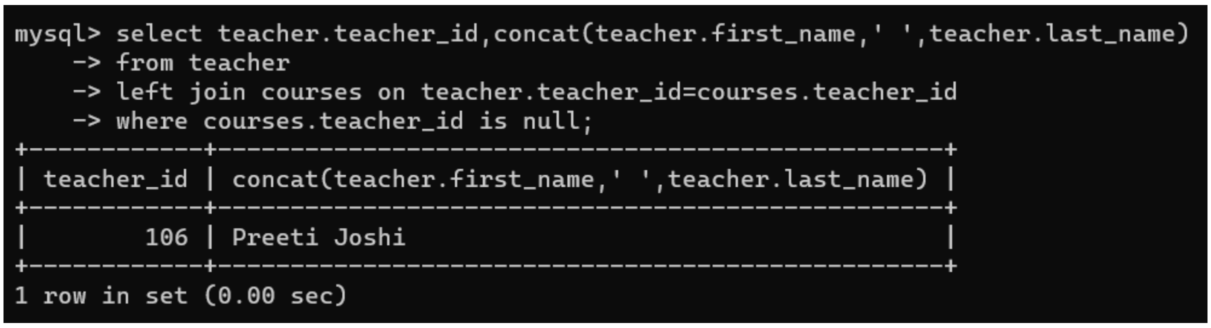
```
mysql> select courses.course_id,courses.course_name from courses
    -> left join enrollments on courses.course_id=enrollments.course_id
    -> where enrollments.course_id is null;
+-----------+--------------------+
| course_id | course_name        |
+-----------+--------------------+
|       109 | Geography          |
|       110 | English Literature |
+-----------+--------------------+
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 e1.student_id,concat(students.first_name,' ',students.last_name) as Name, count(e1.course_id) as NoOfEnrollments
    -> from enrollments e1
    -> join enrollments e2 on e1.student_id=e2.student_id and e1.course_id<>e2.course_id
    -> join students on e1.student_id=students.student_id
    -> group by e1.student_id,students.first_name,students.last_name
    -> having NoOfEnrollments>1;
+------------+------------+-----------------+
| student_id | Name       | NoOfEnrollments |
+------------+------------+-----------------+
|          6 | Neha Kumari |              2 |
+------------+------------+-----------------+
1 row in set (0.00 sec)
```

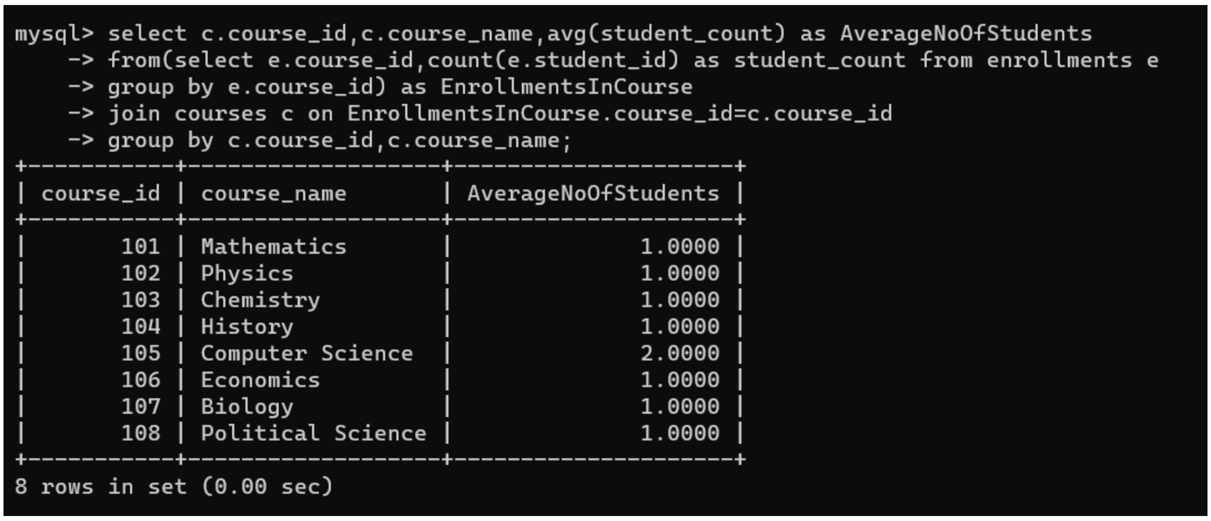
10. Find teachers who are not assigned to any courses. Use a LEFT JOIN between the "Teacher" table and the "Courses" table and filter for teachers with NULL course assignments.



```
mysql> select teacher.teacher_id,concat(teacher.first_name,' ',teacher.last_name)
    -> from teacher
    -> left join courses on teacher.teacher_id=courses.teacher_id
    -> where courses.teacher_id is null;
+------------+--------------------------------------------------+
| teacher_id | concat(teacher.first_name,' ',teacher.last_name) |
+------------+--------------------------------------------------+
|        106 | Preeti Joshi                                     |
+------------+--------------------------------------------------+
1 row in set (0.00 sec)
```

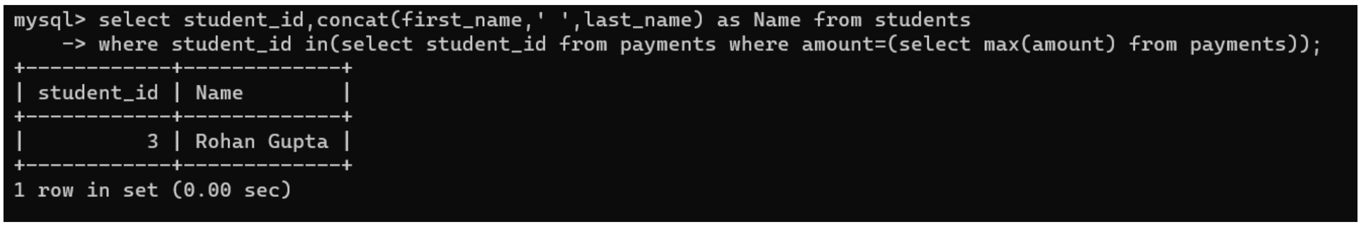
Task 4. Subquery and its type:

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



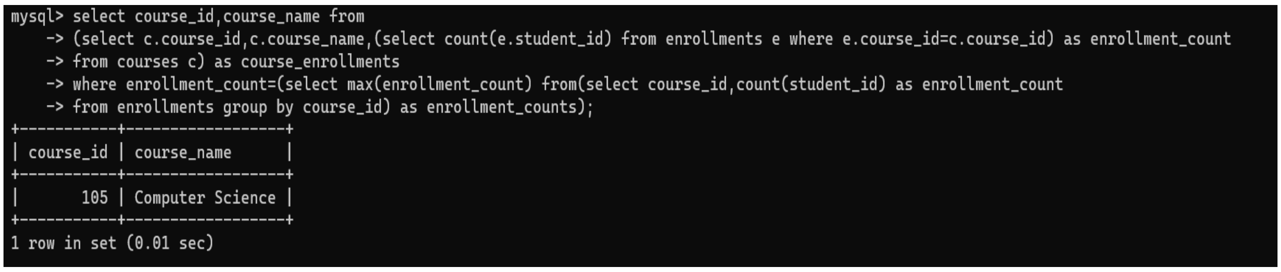
```
mysql> select c.course_id,c.course_name,avg(student_count) as AverageNoOfStudents
    -> from(select e.course_id,count(e.student_id) as student_count from enrollments e
    -> group by e.course_id) as EnrollmentsInCourse
    -> join courses c on EnrollmentsInCourse.course_id=c.course_id
    -> group by c.course_id,c.course_name;
+-----------+-------------------+---------------------+
| course_id | course_name       | AverageNoOfStudents |
+-----------+-------------------+---------------------+
|       101 | Mathematics       |              1.0000 |
|       102 | Physics           |              1.0000 |
|       103 | Chemistry         |              1.0000 |
|       104 | History           |              1.0000 |
|       105 | Computer Science  |              2.0000 |
|       106 | Economics         |              1.0000 |
|       107 | Biology           |              1.0000 |
|       108 | Political Science |              1.0000 |
+-----------+-------------------+---------------------+
8 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.



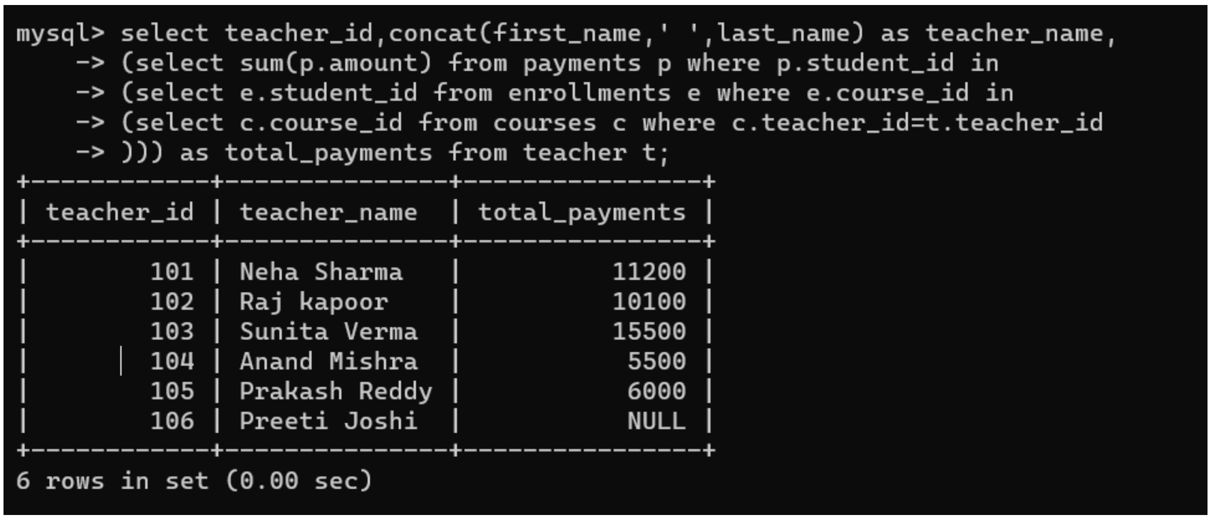
```
mysql> select student_id,concat(first_name,' ',last_name) as Name from students
    -> where student_id in(select student_id from payments where amount=(select max(amount) from payments));
+------------+-------------+
| student_id | Name        |
+------------+-------------+
|          3 | Rohan Gupta |
+------------+-------------+
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.



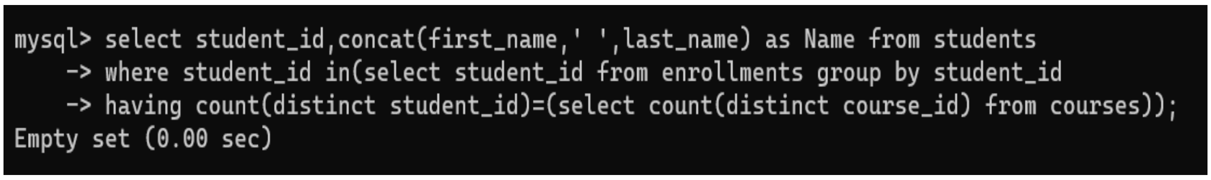
```
mysql> select course_id,course_name from
    -> (select c.course_id,c.course_name,(select count(e.student_id) from enrollments e where e.course_id=c.course_id) as enrollment_count
    -> from courses c) as course_enrollments
    -> where enrollment_count=(select max(enrollment_count) from(select course_id,count(student_id) as enrollment_count
    -> from enrollments group by course_id) as enrollment_counts);
+-----------+------------------+
| course_id | course_name      |
+-----------+------------------+
|       105 | Computer Science |
+-----------+------------------+
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.



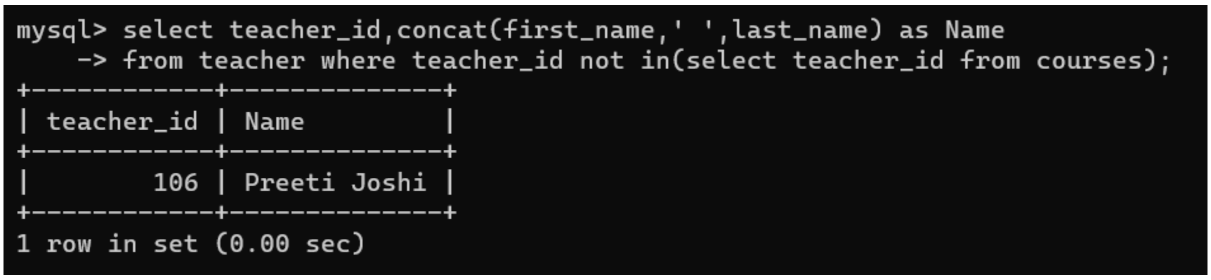
```
mysql> select teacher_id,concat(first_name,' ',last_name) as teacher_name,
    -> (select sum(p.amount) from payments p where p.student_id in
    -> (select e.student_id from enrollments e where e.course_id in
    -> (select c.course_id from courses c where c.teacher_id=t.teacher_id
    -> ))) as total_payments from teacher t;
+------------+---------------+----------------+
| teacher_id | teacher_name  | total_payments |
+------------+---------------+----------------+
|        101 | Neha Sharma   |          11200 |
|        102 | Raj kapoor    |          10100 |
|        103 | Sunita Verma  |          15500 |
|        104 | Anand Mishra  |           5500 |
|        105 | Prakash Reddy |           6000 |
|        106 | Preeti Joshi  |           NULL |
+------------+---------------+----------------+
6 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.



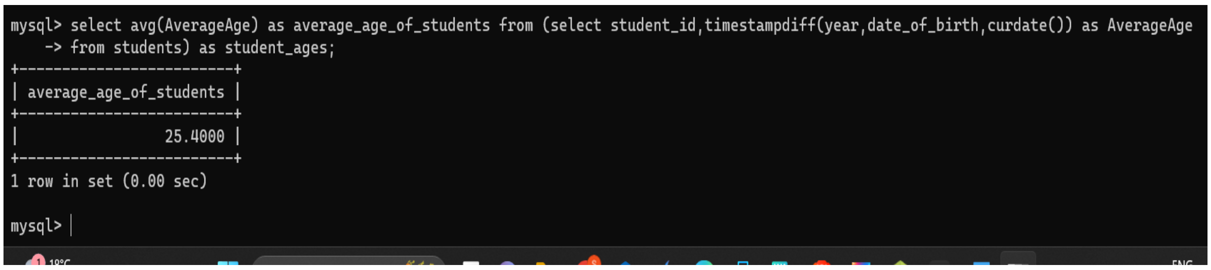
```
mysql> select student_id,concat(first_name,' ',last_name) as Name from students
    -> where student_id in(select student_id from enrollments group by student_id
    -> having count(distinct student_id)=(select count(distinct course_id) from courses));
Empty set (0.00 sec)
```

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



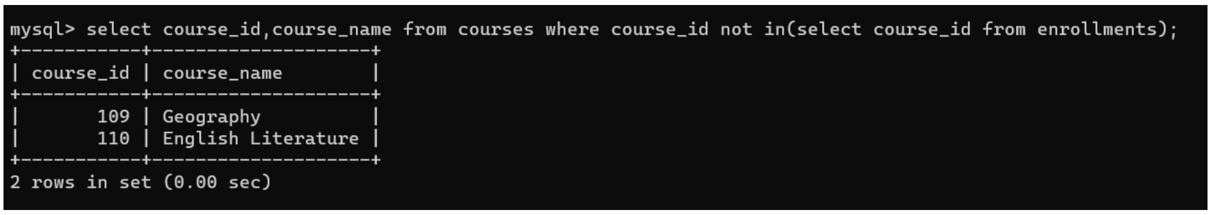
```
mysql> select teacher_id,concat(first_name,' ',last_name) as Name
    -> from teacher where teacher_id not in(select teacher_id from courses);
+------------+--------------+
| teacher_id | Name         |
+------------+--------------+
|        106 | Preeti Joshi |
+------------+--------------+
1 row 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.



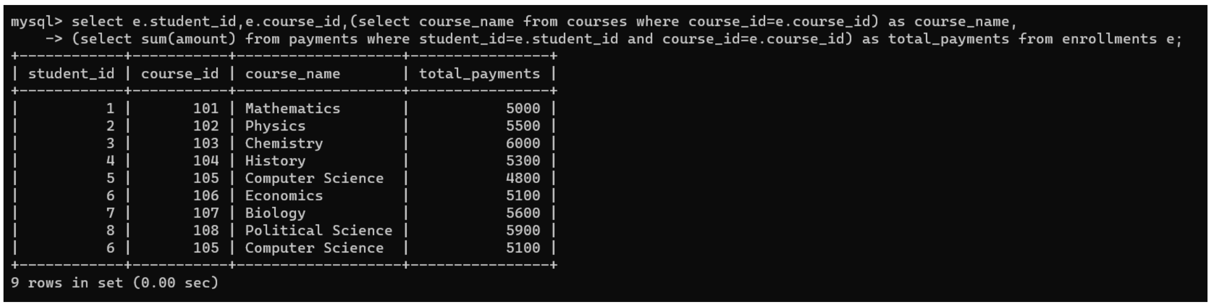
```
mysql> select avg(AverageAge) as average_age_of_students from (select student_id,timestampdiff(year,date_of_birth,curdate()) as AverageAge
    -> from students) as student_ages;
+-------------------------+
| average_age_of_students |
+-------------------------+
|                 25.4000 |
+-------------------------+
1 row in set (0.00 sec)

mysql> |
```

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



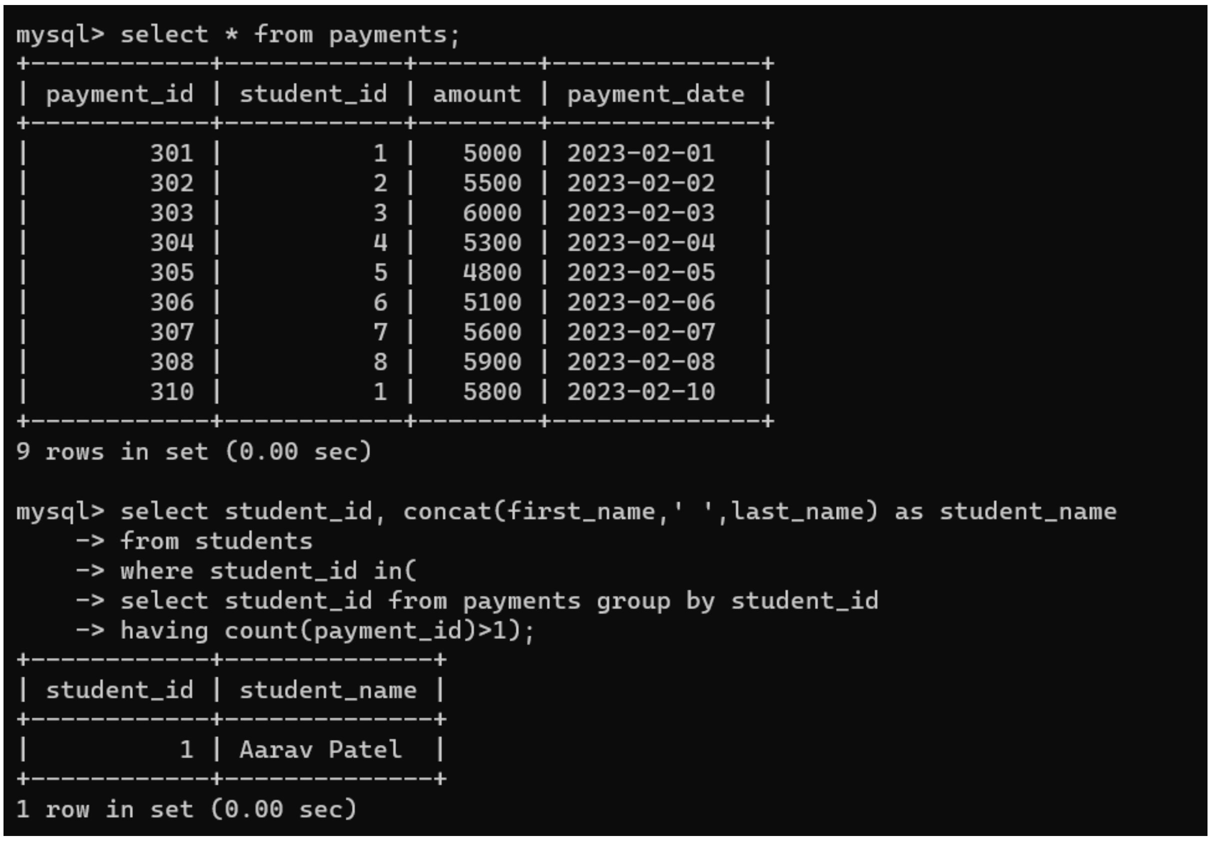
```
mysql> select course_id,course_name from courses where course_id not in(select course_id from enrollments);
+-----------+--------------------+
| course_id | course_name        |
+-----------+--------------------+
|       109 | Geography          |
|       110 | English Literature |
+-----------+--------------------+
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.



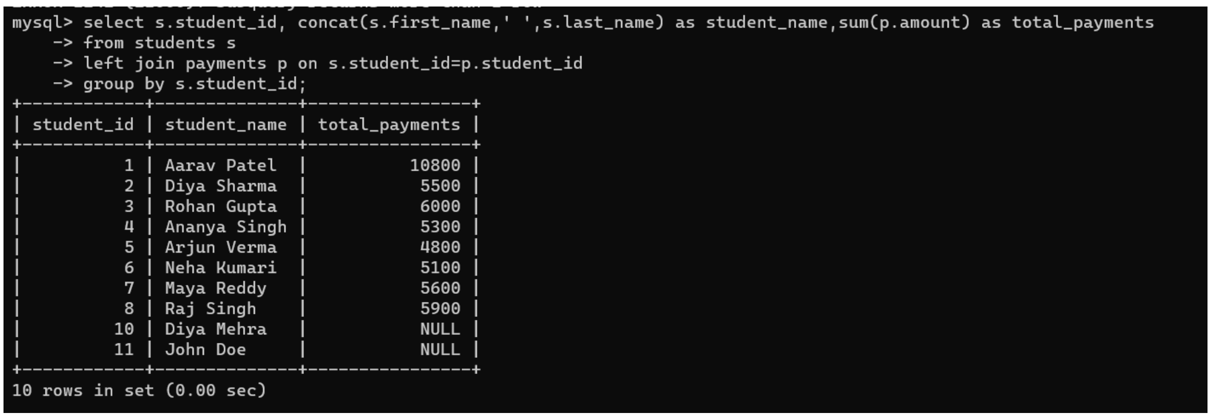
```
mysql> select e.student_id,e.course_id,(select course_name from courses where course_id=e.course_id) as course_name,
    -> (select sum(amount) from payments where student_id=e.student_id and course_id=e.course_id) as total_payments from enrollments e;
+------------+-----------+-------------------+----------------+
| student_id | course_id | course_name       | total_payments |
+------------+-----------+-------------------+----------------+
|          1 |       101 | Mathematics       |           5000 |
|          2 |       102 | Physics           |           5500 |
|          3 |       103 | Chemistry         |           6000 |
|          4 |       104 | History           |           5300 |
|          5 |       105 | Computer Science  |           4800 |
|          6 |       106 | Economics         |           5100 |
|          7 |       107 | Biology           |           5600 |
|          8 |       108 | Political Science |           5900 |
|          6 |       105 | Computer Science  |           5100 |
+------------+-----------+-------------------+----------------+
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.



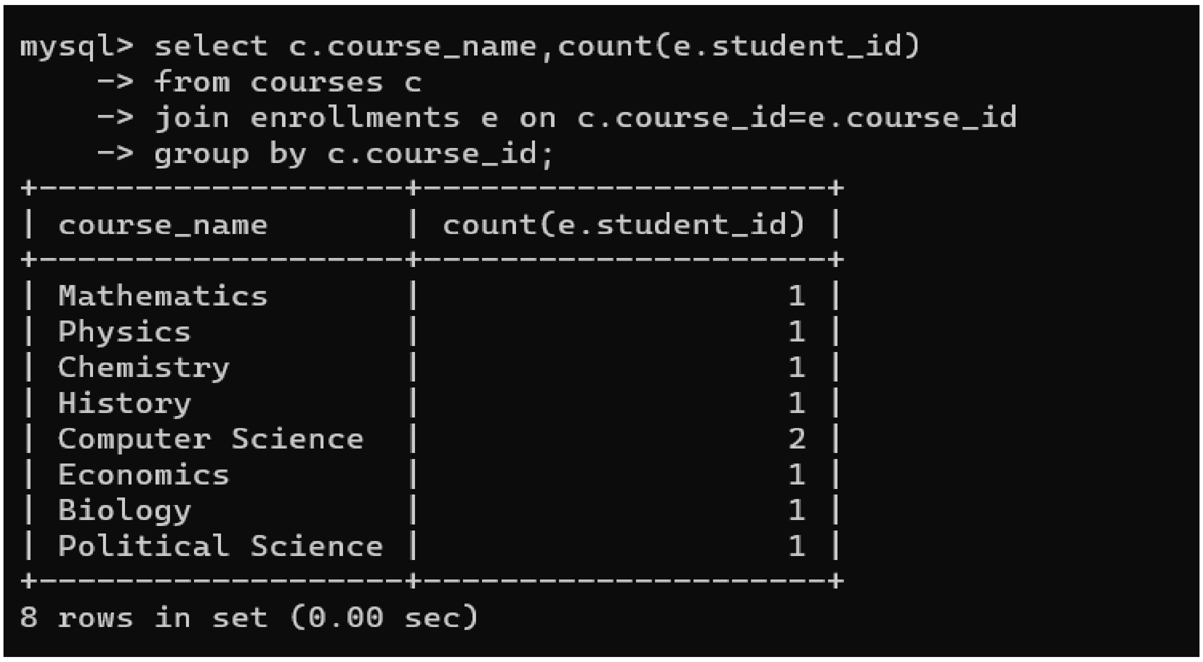
```
mysql> select * from payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|        301 |          1 |   5000 | 2023-02-01   |
|        302 |          2 |   5500 | 2023-02-02   |
|        303 |          3 |   6000 | 2023-02-03   |
|        304 |          4 |   5300 | 2023-02-04   |
|        305 |          5 |   4800 | 2023-02-05   |
|        306 |          6 |   5100 | 2023-02-06   |
|        307 |          7 |   5600 | 2023-02-07   |
|        308 |          8 |   5900 | 2023-02-08   |
|        310 |          1 |   5800 | 2023-02-10   |
+------------+------------+--------+--------------+
9 rows in set (0.00 sec)

mysql> select student_id, concat(first_name,' ',last_name) as student_name
    -> from students
    -> where student_id in(
    -> select student_id from payments group by student_id
    -> having count(payment_id)>1);
+------------+--------------+
| student_id | student_name |
+------------+--------------+
|          1 | Aarav Patel  |
+------------+--------------+
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.



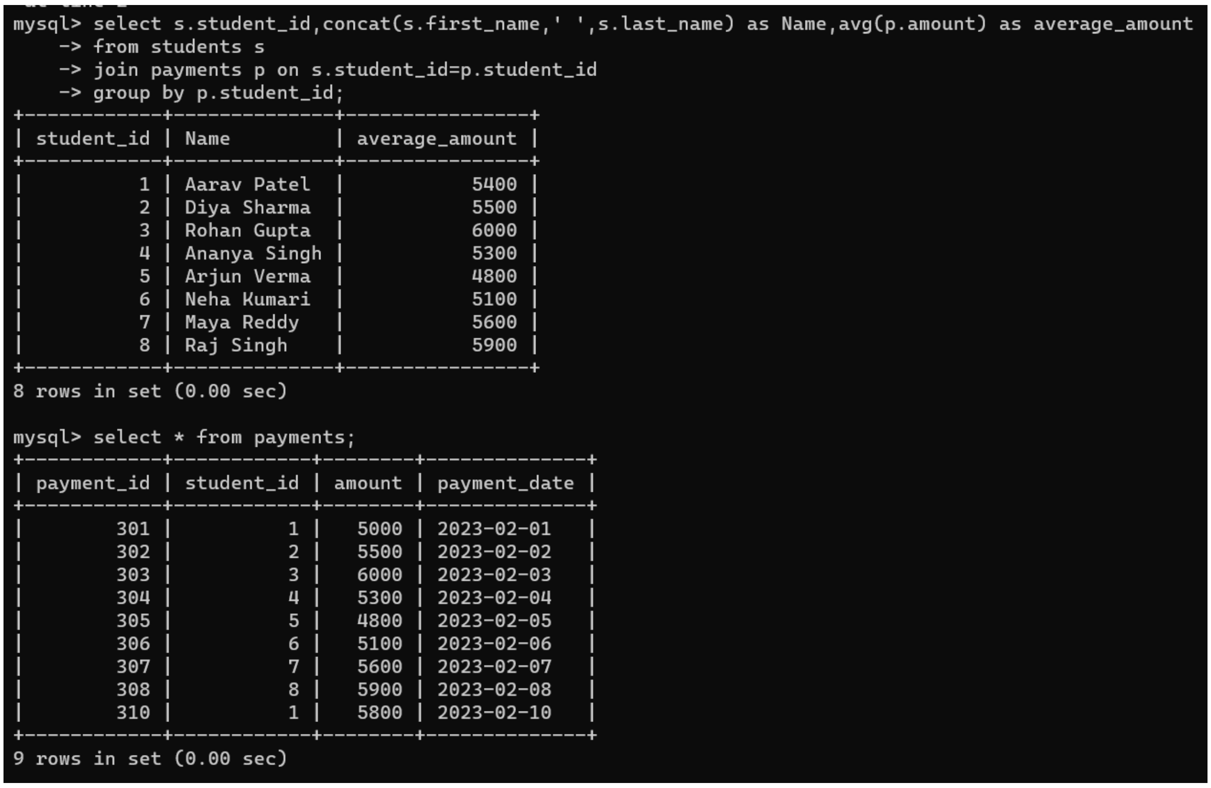
```
mysql> select s.student_id, concat(s.first_name,' ',s.last_name) as student_name,sum(p.amount) as total_payments
    -> from students s
    -> left join payments p on s.student_id=p.student_id
    -> group by s.student_id;
+------------+--------------+----------------+
| student_id | student_name | total_payments |
+------------+--------------+----------------+
|          1 | Aarav Patel  |          10800 |
|          2 | Diya Sharma  |           5500 |
|          3 | Rohan Gupta  |           6000 |
|          4 | Ananya Singh |           5300 |
|          5 | Arjun Verma  |           4800 |
|          6 | Neha Kumari  |           5100 |
|          7 | Maya Reddy   |           5600 |
|          8 | Raj Singh    |           5900 |
|         10 | Diya Mehra   |           NULL |
|         11 | John Doe     |           NULL |
+------------+--------------+----------------+
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)
    -> from courses c
    -> join enrollments e on c.course_id=e.course_id
    -> group by c.course_id;
+-------------------+---------------------+
| course_name       | count(e.student_id) |
+-------------------+---------------------+
| Mathematics       |                   1 |
| Physics           |                   1 |
| Chemistry         |                   1 |
| History           |                   1 |
| Computer Science  |                   2 |
| Economics         |                   1 |
| Biology           |                   1 |
| Political Science |                   1 |
+-------------------+---------------------+
8 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 s.student_id,concat(s.first_name,' ',s.last_name) as Name,avg(p.amount) as average_amount
    -> from students s
    -> join payments p on s.student_id=p.student_id
    -> group by p.student_id;
+------------+--------------+----------------+
| student_id | Name         | average_amount |
+------------+--------------+----------------+
|          1 | Aarav Patel  |           5400 |
|          2 | Diya Sharma  |           5500 |
|          3 | Rohan Gupta  |           6000 |
|          4 | Ananya Singh |           5300 |
|          5 | Arjun Verma  |           4800 |
|          6 | Neha Kumari  |           5100 |
|          7 | Maya Reddy   |           5600 |
|          8 | Raj Singh    |           5900 |
+------------+--------------+----------------+
8 rows in set (0.00 sec)

mysql> select * from payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|        301 |          1 |   5000 | 2023-02-01   |
|        302 |          2 |   5500 | 2023-02-02   |
|        303 |          3 |   6000 | 2023-02-03   |
|        304 |          4 |   5300 | 2023-02-04   |
|        305 |          5 |   4800 | 2023-02-05   |
|        306 |          6 |   5100 | 2023-02-06   |
|        307 |          7 |   5600 | 2023-02-07   |
|        308 |          8 |   5900 | 2023-02-08   |
|        310 |          1 |   5800 | 2023-02-10   |
+------------+------------+--------+--------------+
9 rows in set (0.00 sec)
```