Assignment – 2

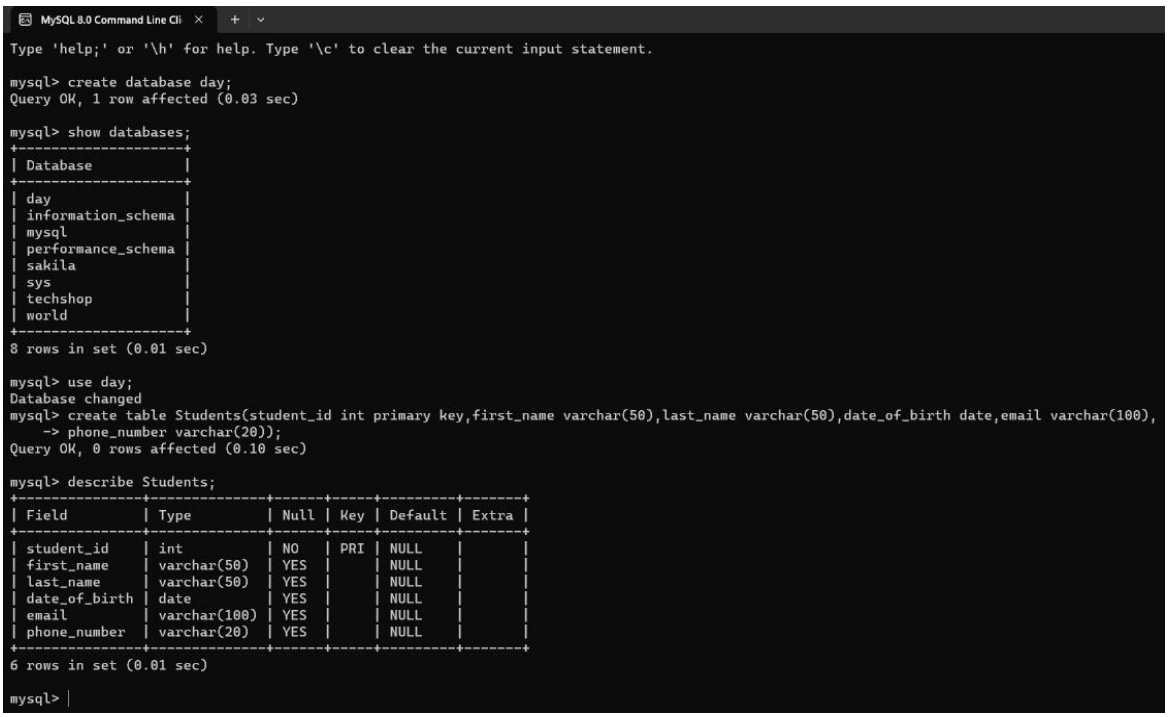
Madiha Aimon Tappal

Database Tables:

The SIS data base consists of the following tables:

1. **Students**

- Students_id (Primary Key)
- First_name
- Last_name
- Date_of_birth
- Email
- Phone_number

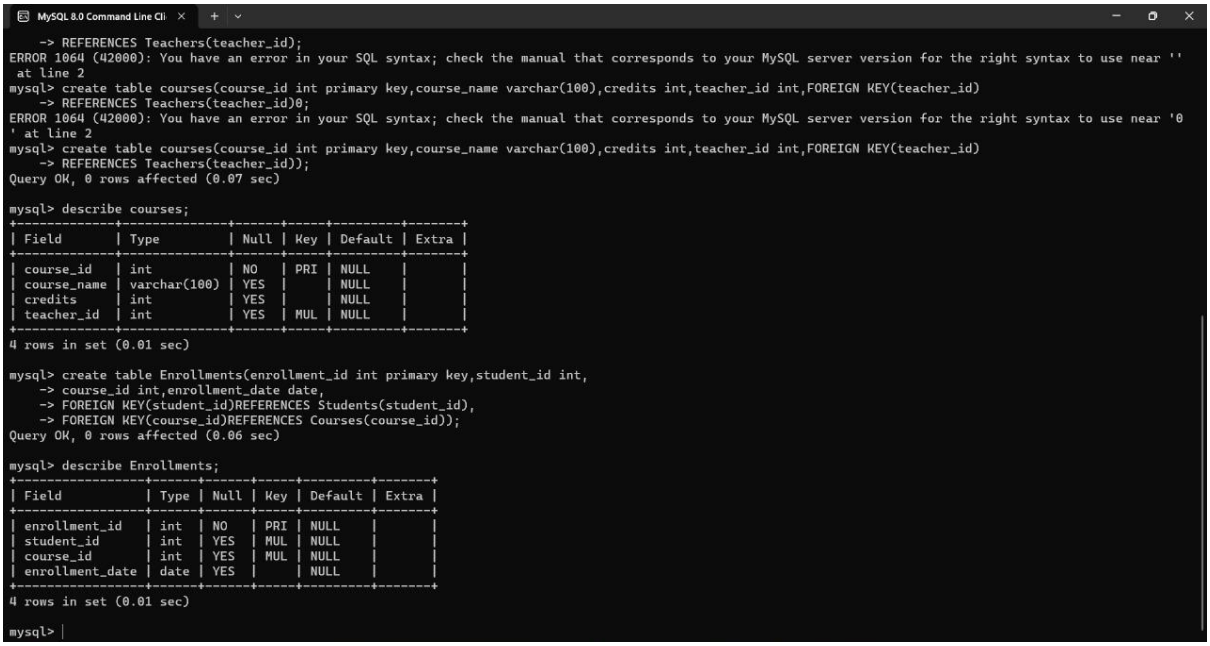


2. **Courses:**

- Course_id(Primary Key)
- Course_name
- Credits
- Teacher_id(Foreign Key)

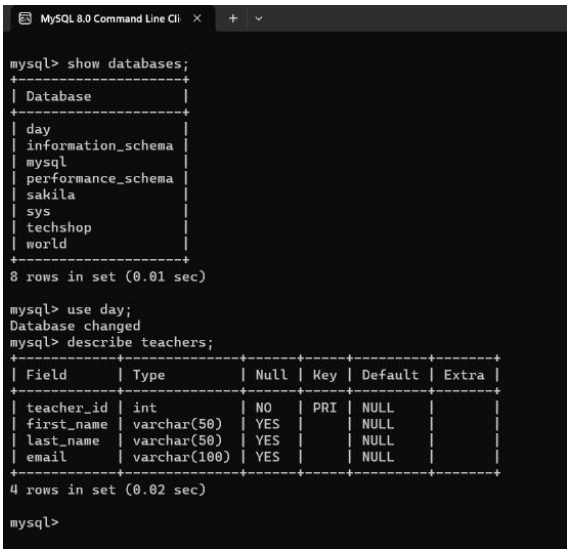
3. **Enrollments:**

- Enrollments_id(Primary Key)
- Students_id (Foreign Key)
- Course_id (Foreign Key)
- Enrollment_date



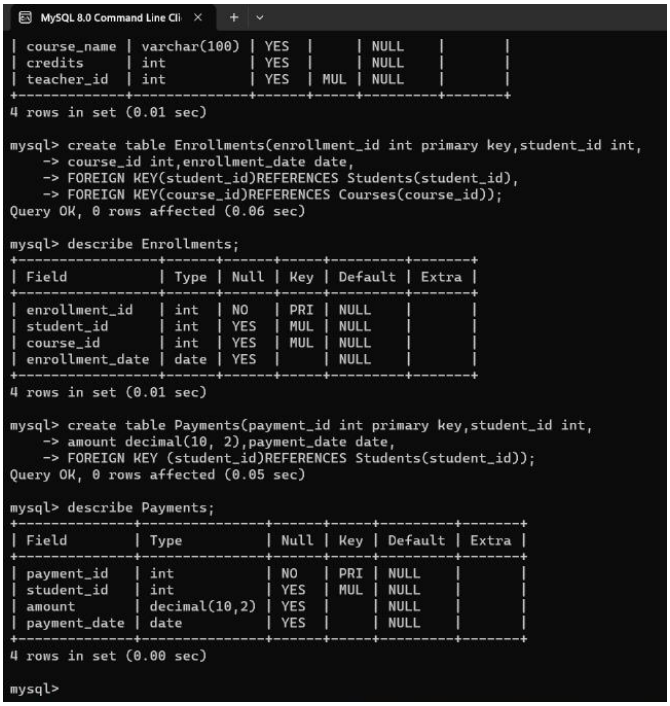
4. **Teachers:**

- Teacher_id (Primary key)
- First_name
- Last_name
- Email



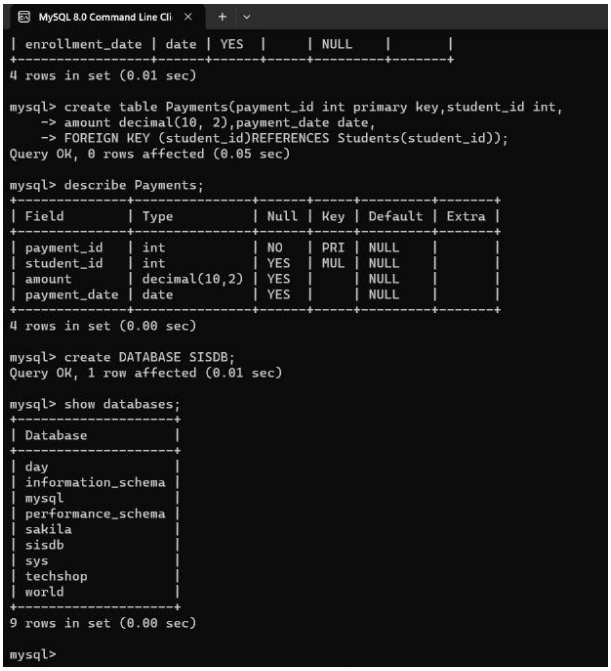
5. **Payments:**

- Payment_id (Primary Key)
- Student_id(Foreign key)
- Amount
- Payment_date



Task 1. Database Design:

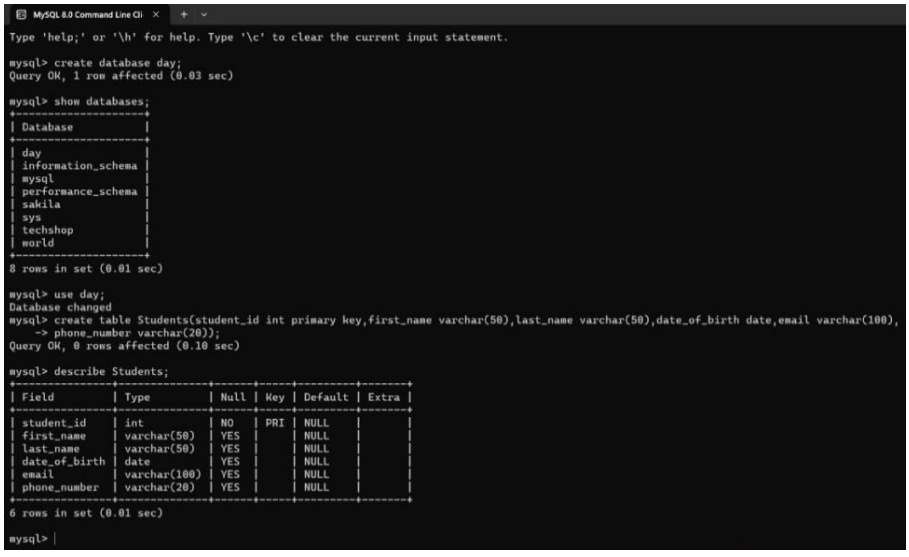
1. Create the Database named “SISDB”?



```
MySQL 8.0 Command Line Cli

| enrollment_date | date | YES  |     | NULL    |       |
+-----------------+------+------+-----+---------+-------+
4 rows in set (0.01 sec)

mysql> create table Payments(payment_id int primary key,student_id int,
    -> amount decimal(10, 2),payment_date date,
    -> FOREIGN KEY (student_id)REFERENCES Students(student_id));
Query OK, 0 rows affected (0.05 sec)

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

mysql> create DATABASE SISDB;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| day                |
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sisdb              |
| sys                |
| techshop           |
| world              |
+--------------------+
9 rows in set (0.00 sec)

mysql>
```

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
    - Students
    - Cources
    - Enrollments
    - Teachers
    - Payments

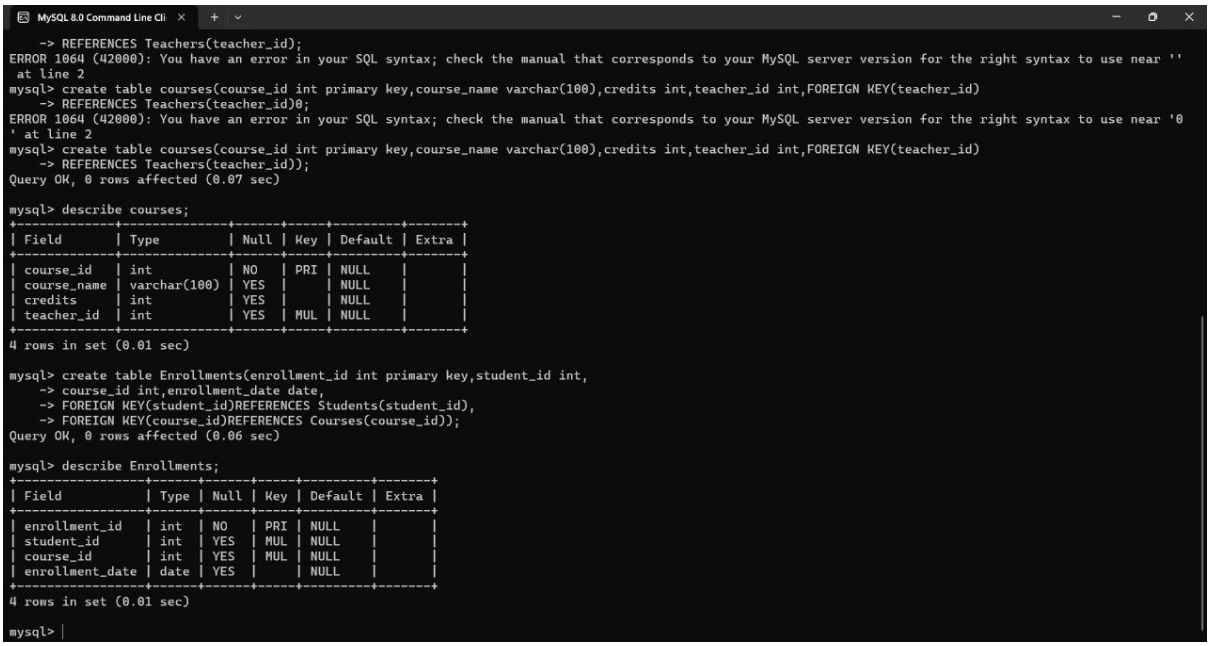
1. Students



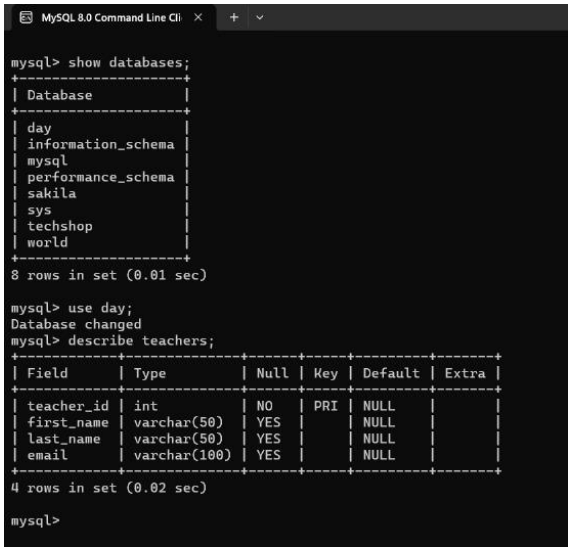
```
MySQL 8.0 Command Line Cli

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

mysql> create database day;
Query OK, 1 row affected (0.03 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| day                |
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| techshop           |
| world              |
+--------------------+
8 rows in set (0.01 sec)

mysql> use day;
Database changed
mysql> create table Students(student_id int primary key,first_name varchar(50),last_name varchar(50),date_of_birth date,email varchar(100),
    -> phone_number varchar(20));
Query OK, 0 rows affected (0.10 sec)

mysql> describe Students;
+---------------+--------------+------+-----+---------+-------+
| 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(100) | YES  |     | NULL    |       |
| phone_number  | varchar(20)  | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
6 rows in set (0.01 sec)

mysql>
```

2. Cources

3. Enrollments

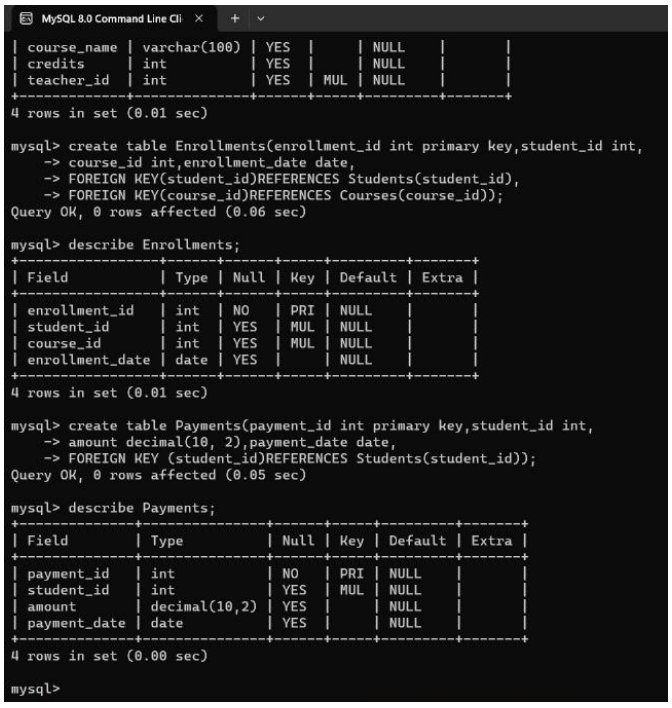


4. Teachers



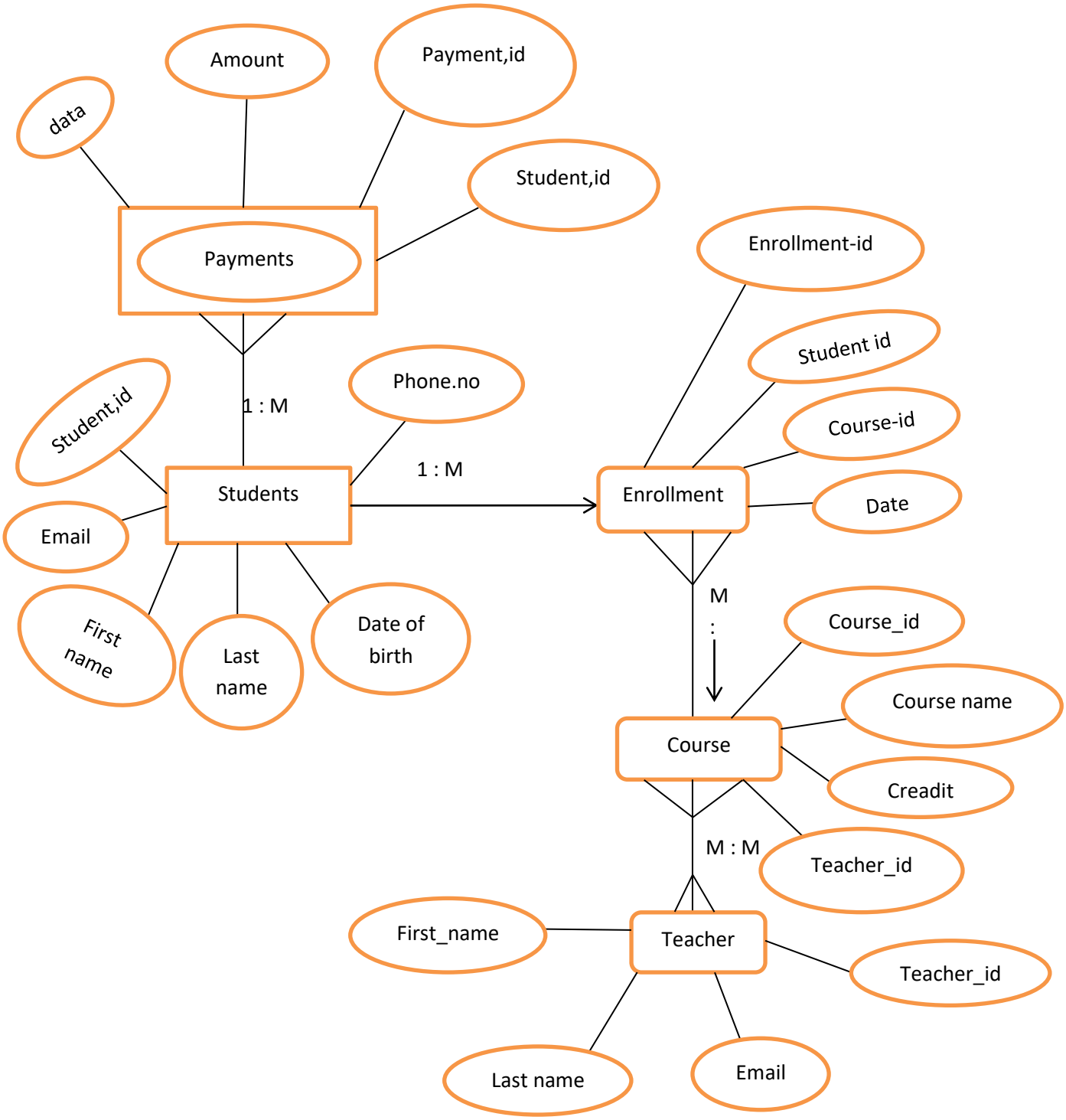
```
MySQL 8.0 Command Line Cli

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| day                |
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| techshop           |
| world              |
+--------------------+
8 rows in set (0.01 sec)

mysql> use day;
Database changed
mysql> describe 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(100) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
4 rows in set (0.02 sec)

mysql>
```

5. Payments



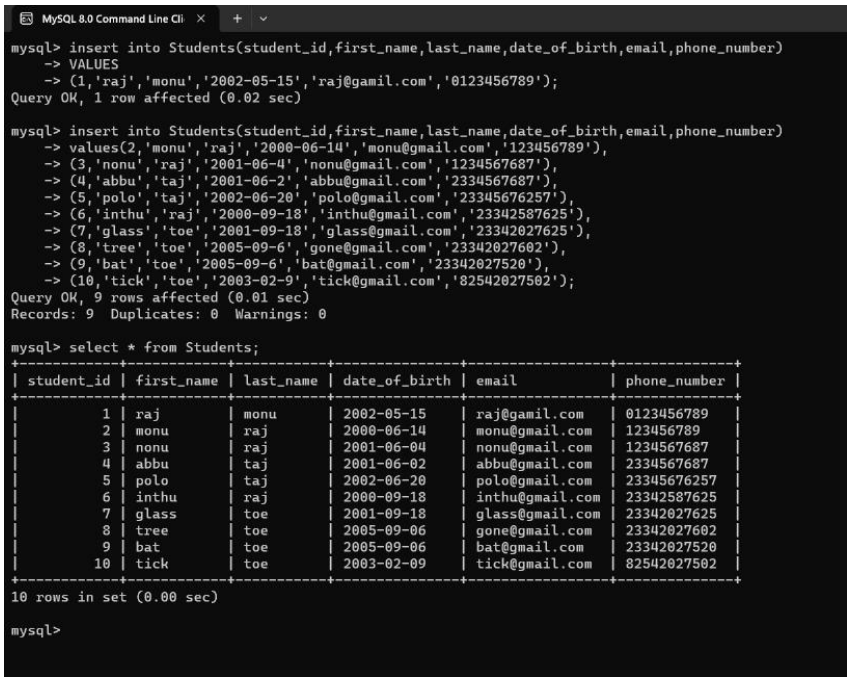
```
MySQL 8.0 Command Line Cli

| course_name | varchar(100) | YES  |     | NULL    |       |
| credits     | int          | YES  |     | NULL    |       |
| teacher_id  | int          | YES  | MUL | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
4 rows in set (0.01 sec)

mysql> create table Enrollments(enrollment_id int primary key,student_id int,
    -> course_id int,enrollment_date date,
    -> FOREIGN KEY(student_id)REFERENCES Students(student_id),
    -> FOREIGN KEY(course_id)REFERENCES Courses(course_id));
Query OK, 0 rows affected (0.06 sec)

mysql> describe 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 (0.01 sec)

mysql> create table Payments(payment_id int primary key,student_id int,
    -> amount decimal(10, 2),payment_date date,
    -> FOREIGN KEY (student_id)REFERENCES Students(student_id));
Query OK, 0 rows affected (0.05 sec)

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

mysql>
```

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


Amount
Payment,id
data
Student,id
Payments
Enrollment-id
Student id
Course-id
Date
Enrollment
1 : M
Phone.no
Student,id
1 : M
Students
Email
First name
Last name
Date of birth
M
:
Course_id
Course name
Course
Creadit
M : M
Teacher_id
First_name
Teacher
Teacher_id
Last name
Email

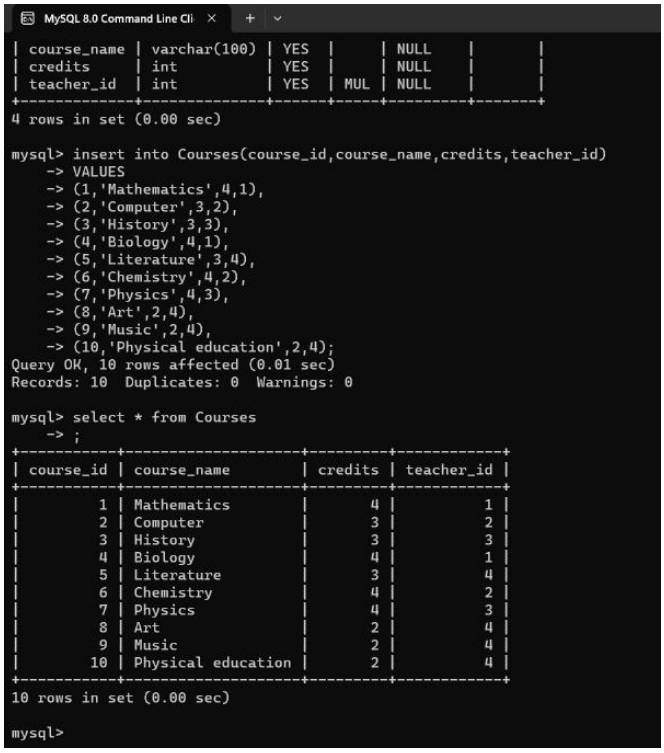
4 . Create appropriate Primary key and Foreign Key constraints referential integrity.

1. **Students:**



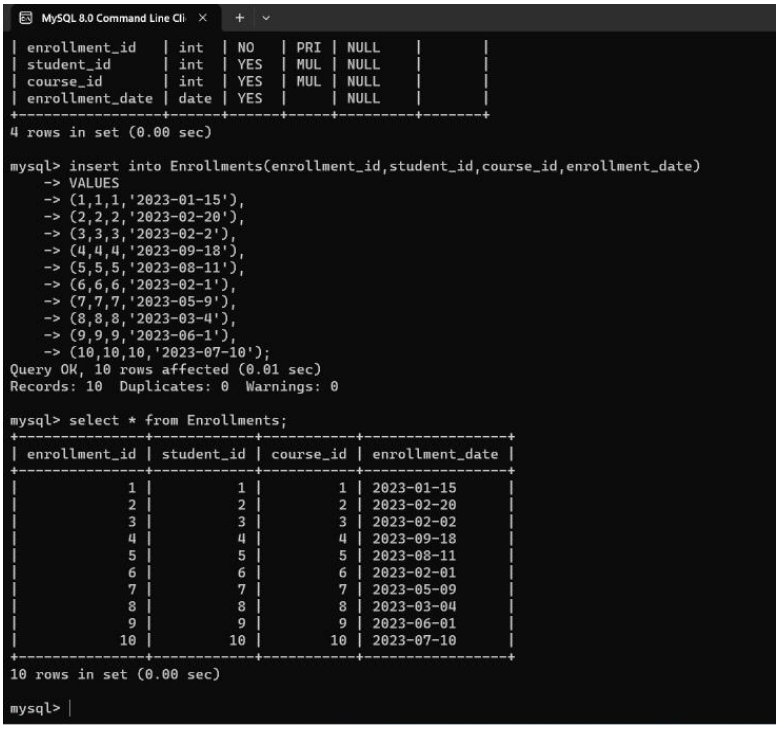
```
MySQL 8.0 Command Line Cli

mysql> insert into Students(student_id,first_name,last_name,date_of_birth,email,phone_number)
    -> VALUES
    -> (1,'raj','monu','2002-05-15','raj@gamil.com','0123456789');
Query OK, 1 row affected (0.02 sec)

mysql> insert into Students(student_id,first_name,last_name,date_of_birth,email,phone_number)
    -> values(2,'monu','raj','2000-06-14','monu@gmail.com','123456789'),
    -> (3,'nonu','raj','2001-06-4','nonu@gmail.com','1234567687'),
    -> (4,'abbu','taj','2001-06-2','abbu@gmail.com','2334567687'),
    -> (5,'polo','taj','2002-06-20','polo@gmail.com','23345676257'),
    -> (6,'inthu','raj','2000-09-18','inthu@gmail.com','23342587625'),
    -> (7,'glass','toe','2001-09-18','glass@gmail.com','23342027625'),
    -> (8,'tree','toe','2005-09-6','gone@gmail.com','23342027602'),
    -> (9,'bat','toe','2005-09-6','bat@gmail.com','23342027520'),
    -> (10,'tick','toe','2003-02-9','tick@gmail.com','82542027502');
Query OK, 9 rows affected (0.01 sec)
Records: 9  Duplicates: 0  Warnings: 0

mysql> select * from Students;
+------------+------------+-----------+---------------+-----------------+--------------+
| student_id | first_name | last_name | date_of_birth | email           | phone_number |
+------------+------------+-----------+---------------+-----------------+--------------+
|          1 | raj        | monu      | 2002-05-15    | raj@gamil.com   | 0123456789   |
|          2 | monu       | raj       | 2000-06-14    | monu@gmail.com  | 123456789    |
|          3 | nonu       | raj       | 2001-06-04    | nonu@gmail.com  | 1234567687   |
|          4 | abbu       | taj       | 2001-06-02    | abbu@gmail.com  | 2334567687   |
|          5 | polo       | taj       | 2002-06-20    | polo@gmail.com  | 23345676257  |
|          6 | inthu      | raj       | 2000-09-18    | inthu@gmail.com | 23342587625  |
|          7 | glass      | toe       | 2001-09-18    | glass@gmail.com | 23342027625  |
|          8 | tree       | toe       | 2005-09-06    | gone@gmail.com  | 23342027602  |
|          9 | bat        | toe       | 2005-09-06    | bat@gmail.com   | 23342027520  |
|         10 | tick       | toe       | 2003-02-09    | tick@gmail.com  | 82542027502  |
+------------+------------+-----------+---------------+-----------------+--------------+
10 rows in set (0.00 sec)

mysql>
```

2. **Cources:**



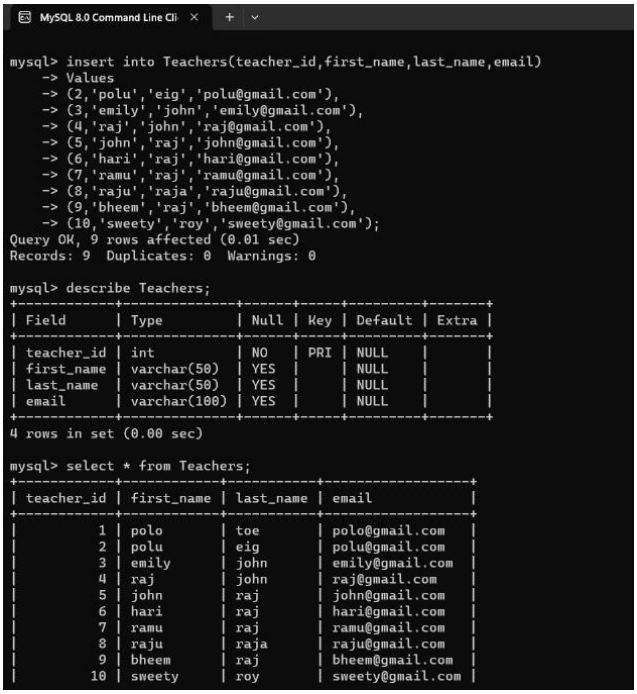
```
MySQL 8.0 Command Line Cli

| course_name | varchar(100) | YES  |     | NULL    |       |
| credits     | int          | YES  |     | NULL    |       |
| teacher_id  | int          | YES  | MUL | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> insert into Courses(course_id,course_name,credits,teacher_id)
    -> VALUES
    -> (1,'Mathematics',4,1),
    -> (2,'Computer',3,2),
    -> (3,'History',3,3),
    -> (4,'Biology',4,1),
    -> (5,'Literature',3,4),
    -> (6,'Chemistry',4,2),
    -> (7,'Physics',4,3),
    -> (8,'Art',2,4),
    -> (9,'Music',2,4),
    -> (10,'Physical education',2,4);
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 | Mathematics        |       4 |          1 |
|         2 | Computer           |       3 |          2 |
|         3 | History            |       3 |          3 |
|         4 | Biology            |       4 |          1 |
|         5 | Literature         |       3 |          4 |
|         6 | Chemistry          |       4 |          2 |
|         7 | Physics            |       4 |          3 |
|         8 | Art                |       2 |          4 |
|         9 | Music              |       2 |          4 |
|        10 | Physical education |       2 |          4 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)

mysql>
```

3. Enrollment:



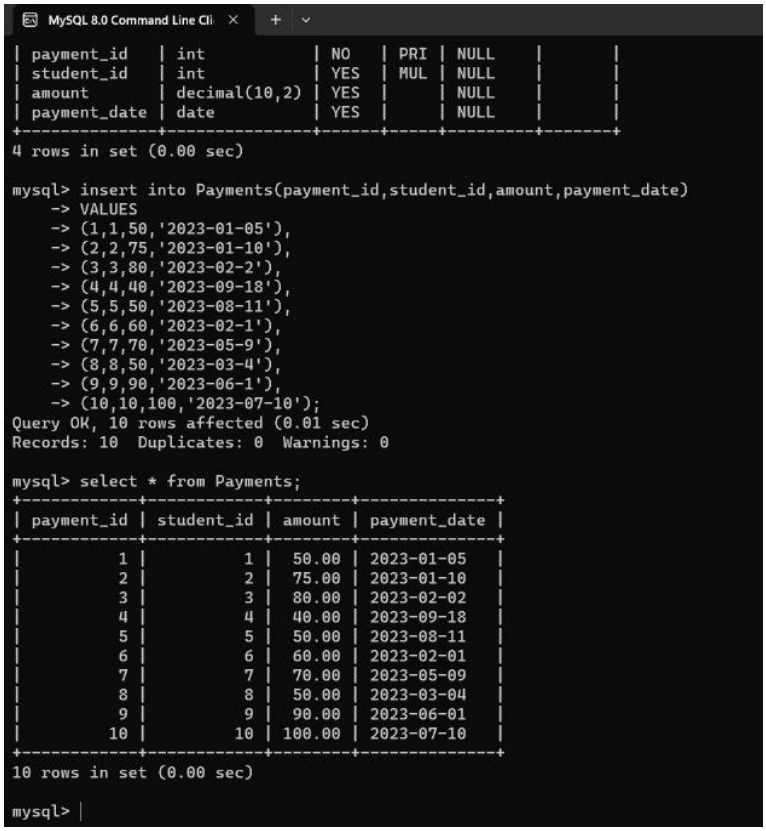
```
MySQL 8.0 Command Line Cli

| 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> insert into Enrollments(enrollment_id,student_id,course_id,enrollment_date)
    -> VALUES
    -> (1,1,1,'2023-01-15'),
    -> (2,2,2,'2023-02-20'),
    -> (3,3,3,'2023-02-2'),
    -> (4,4,4,'2023-09-18'),
    -> (5,5,5,'2023-08-11'),
    -> (6,6,6,'2023-02-1'),
    -> (7,7,7,'2023-05-9'),
    -> (8,8,8,'2023-03-4'),
    -> (9,9,9,'2023-06-1'),
    -> (10,10,10,'2023-07-10');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2023-01-15      |
|             2 |          2 |         2 | 2023-02-20      |
|             3 |          3 |         3 | 2023-02-02      |
|             4 |          4 |         4 | 2023-09-18      |
|             5 |          5 |         5 | 2023-08-11      |
|             6 |          6 |         6 | 2023-02-01      |
|             7 |          7 |         7 | 2023-05-09      |
|             8 |          8 |         8 | 2023-03-04      |
|             9 |          9 |         9 | 2023-06-01      |
|            10 |         10 |        10 | 2023-07-10      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)

mysql>
```

4. Teacher:



```
MySQL 8.0 Command Line Cli

mysql> insert into Teachers(teacher_id,first_name,last_name,email)
    -> Values
    -> (2,'polu','eig','polu@gmail.com'),
    -> (3,'emily','john','emily@gmail.com'),
    -> (4,'raj','john','raj@gmail.com'),
    -> (5,'john','raj','john@gmail.com'),
    -> (6,'hari','raj','hari@gmail.com'),
    -> (7,'ramu','raj','ramu@gmail.com'),
    -> (8,'raju','raja','raju@gmail.com'),
    -> (9,'bheem','raj','bheem@gmail.com'),
    -> (10,'sweety','roy','sweety@gmail.com');
Query OK, 9 rows affected (0.01 sec)
Records: 9  Duplicates: 0  Warnings: 0

mysql> describe 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(100) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> select * from Teachers;
+------------+------------+-----------+------------------+
| teacher_id | first_name | last_name | email            |
+------------+------------+-----------+------------------+
|          1 | polo       | toe       | polo@gmail.com   |
|          2 | polu       | eig       | polu@gmail.com   |
|          3 | emily      | john      | emily@gmail.com  |
|          4 | raj        | john      | raj@gmail.com    |
|          5 | john       | raj       | john@gmail.com   |
|          6 | hari       | raj       | hari@gmail.com   |
|          7 | ramu       | raj       | ramu@gmail.com   |
|          8 | raju       | raja      | raju@gmail.com   |
|          9 | bheem      | raj       | bheem@gmail.com  |
|         10 | sweety     | roy       | sweety@gmail.com |
```

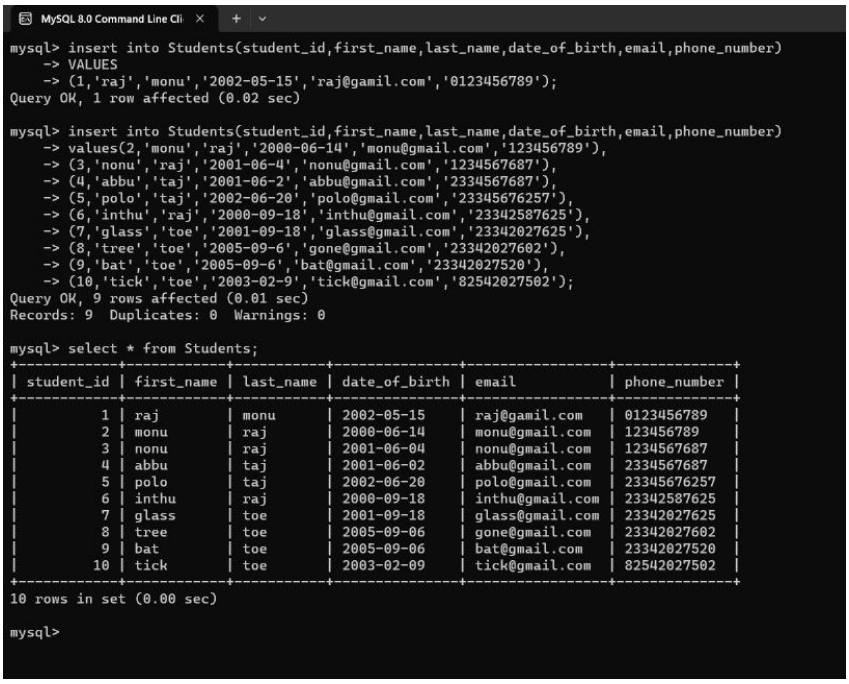
5. Payments:


MySQL 8.0 Command Line Cli
payment_id	int	NO	PRI	NULL	
student_id	int	YES	MUL	NULL	
amount	decimal(10,2)	YES		NULL	
payment_date	date	YES		NULL	
4 rows in set (0.00 sec)
mysql> insert into Payments(payment_id,student_id,amount,payment_date)
-> VALUES
-> (1,1,50,'2023-01-05'),
-> (2,2,75,'2023-01-10'),
-> (3,3,80,'2023-02-2'),
-> (4,4,40,'2023-09-18'),
-> (5,5,50,'2023-08-11'),
-> (6,6,60,'2023-02-1'),
-> (7,7,70,'2023-05-9'),
-> (8,8,50,'2023-03-4'),
-> (9,9,90,'2023-06-1'),
-> (10,10,100,'2023-07-10');
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
1	1	50.00	2023-01-05
2	2	75.00	2023-01-10
3	3	80.00	2023-02-02
4	4	40.00	2023-09-18
5	5	50.00	2023-08-11
6	6	60.00	2023-02-01
7	7	70.00	2023-05-09
8	8	50.00	2023-03-04
9	9	90.00	2023-06-01
10	10	100.00	2023-07-10
10 rows in set (0.00 sec)
mysql>

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

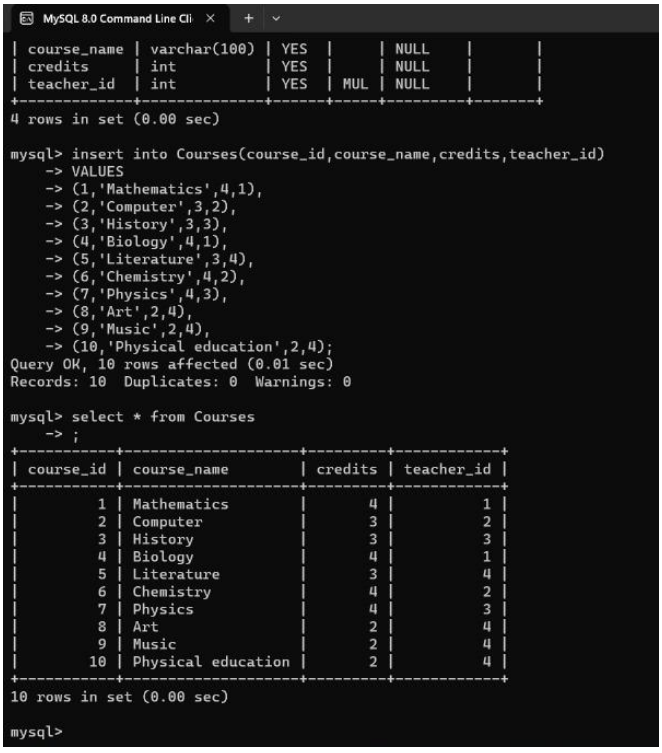
- Students
- Cource
- Enrollment
- Teacher
- Payments

1. **Students:**



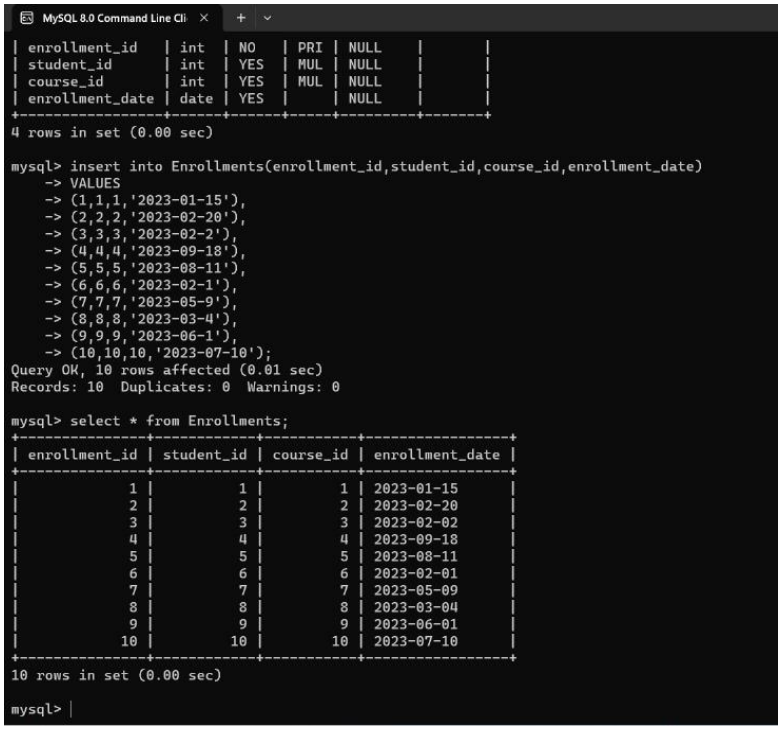
```
MySQL 8.0 Command Line Cli

mysql> insert into Students(student_id,first_name,last_name,date_of_birth,email,phone_number)
    -> VALUES
    -> (1,'raj','monu','2002-05-15','raj@gamil.com','0123456789');
Query OK, 1 row affected (0.02 sec)

mysql> insert into Students(student_id,first_name,last_name,date_of_birth,email,phone_number)
    -> values(2,'monu','raj','2000-06-14','monu@gmail.com','123456789'),
    -> (3,'nonu','raj','2001-06-4','nonu@gmail.com','1234567687'),
    -> (4,'abbu','taj','2001-06-2','abbu@gmail.com','2334567687'),
    -> (5,'polo','taj','2002-06-20','polo@gmail.com','23345676257'),
    -> (6,'inthu','raj','2000-09-18','inthu@gmail.com','23342587625'),
    -> (7,'glass','toe','2001-09-18','glass@gmail.com','23342027625'),
    -> (8,'tree','toe','2005-09-6','gone@gmail.com','23342027602'),
    -> (9,'bat','toe','2005-09-6','bat@gmail.com','23342027520'),
    -> (10,'tick','toe','2003-02-9','tick@gmail.com','82542027502');
Query OK, 9 rows affected (0.01 sec)
Records: 9  Duplicates: 0  Warnings: 0

mysql> select * from Students;
+------------+------------+-----------+---------------+-----------------+--------------+
| student_id | first_name | last_name | date_of_birth | email           | phone_number |
+------------+------------+-----------+---------------+-----------------+--------------+
|          1 | raj        | monu      | 2002-05-15    | raj@gamil.com   | 0123456789   |
|          2 | monu       | raj       | 2000-06-14    | monu@gmail.com  | 123456789    |
|          3 | nonu       | raj       | 2001-06-04    | nonu@gmail.com  | 1234567687   |
|          4 | abbu       | taj       | 2001-06-02    | abbu@gmail.com  | 2334567687   |
|          5 | polo       | taj       | 2002-06-20    | polo@gmail.com  | 23345676257  |
|          6 | inthu      | raj       | 2000-09-18    | inthu@gmail.com | 23342587625  |
|          7 | glass      | toe       | 2001-09-18    | glass@gmail.com | 23342027625  |
|          8 | tree       | toe       | 2005-09-06    | gone@gmail.com  | 23342027602  |
|          9 | bat        | toe       | 2005-09-06    | bat@gmail.com   | 23342027520  |
|         10 | tick       | toe       | 2003-02-09    | tick@gmail.com  | 82542027502  |
+------------+------------+-----------+---------------+-----------------+--------------+
10 rows in set (0.00 sec)

mysql>
```

2. **Cources:**



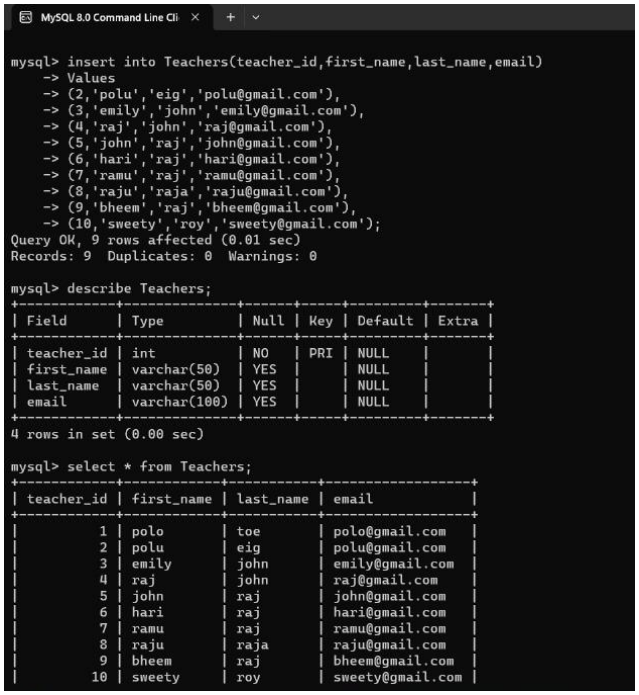
```
MySQL 8.0 Command Line Cli

| course_name | varchar(100) | YES  |     | NULL    |       |
| credits     | int          | YES  |     | NULL    |       |
| teacher_id  | int          | YES  | MUL | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> insert into Courses(course_id,course_name,credits,teacher_id)
    -> VALUES
    -> (1,'Mathematics',4,1),
    -> (2,'Computer',3,2),
    -> (3,'History',3,3),
    -> (4,'Biology',4,1),
    -> (5,'Literature',3,4),
    -> (6,'Chemistry',4,2),
    -> (7,'Physics',4,3),
    -> (8,'Art',2,4),
    -> (9,'Music',2,4),
    -> (10,'Physical education',2,4);
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 | Mathematics        |       4 |          1 |
|         2 | Computer           |       3 |          2 |
|         3 | History            |       3 |          3 |
|         4 | Biology            |       4 |          1 |
|         5 | Literature         |       3 |          4 |
|         6 | Chemistry          |       4 |          2 |
|         7 | Physics            |       4 |          3 |
|         8 | Art                |       2 |          4 |
|         9 | Music              |       2 |          4 |
|        10 | Physical education |       2 |          4 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)

mysql>
```

3. Enrollment:



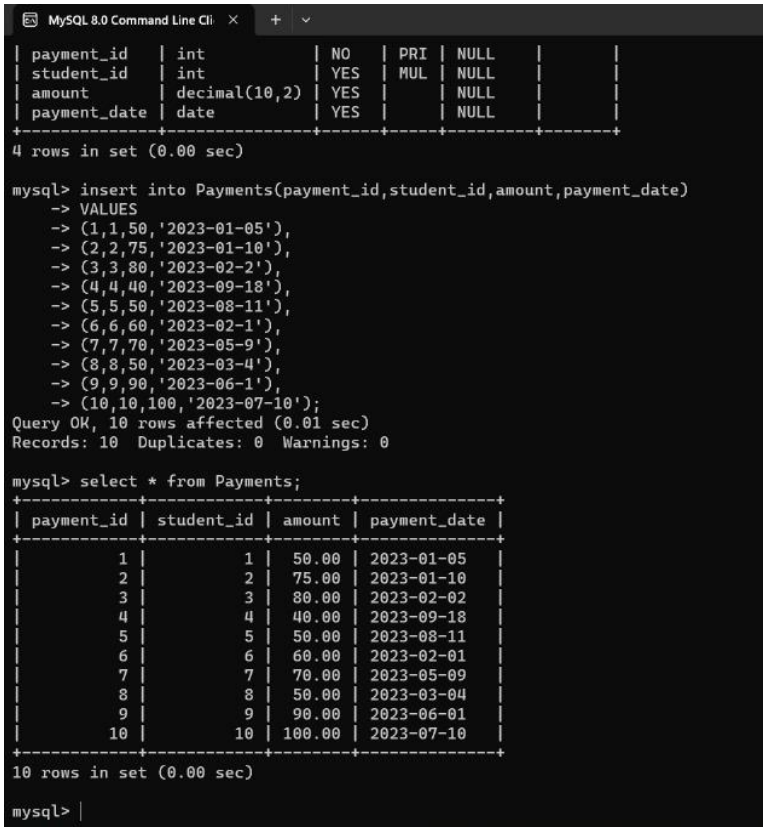
```
MySQL 8.0 Command Line Cli

| 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> insert into Enrollments(enrollment_id,student_id,course_id,enrollment_date)
    -> VALUES
    -> (1,1,1,'2023-01-15'),
    -> (2,2,2,'2023-02-20'),
    -> (3,3,3,'2023-02-2'),
    -> (4,4,4,'2023-09-18'),
    -> (5,5,5,'2023-08-11'),
    -> (6,6,6,'2023-02-1'),
    -> (7,7,7,'2023-05-9'),
    -> (8,8,8,'2023-03-4'),
    -> (9,9,9,'2023-06-1'),
    -> (10,10,10,'2023-07-10');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2023-01-15      |
|             2 |          2 |         2 | 2023-02-20      |
|             3 |          3 |         3 | 2023-02-02      |
|             4 |          4 |         4 | 2023-09-18      |
|             5 |          5 |         5 | 2023-08-11      |
|             6 |          6 |         6 | 2023-02-01      |
|             7 |          7 |         7 | 2023-05-09      |
|             8 |          8 |         8 | 2023-03-04      |
|             9 |          9 |         9 | 2023-06-01      |
|            10 |         10 |        10 | 2023-07-10      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)

mysql>
```

4. Teacher:



```
MySQL 8.0 Command Line Cli

mysql> insert into Teachers(teacher_id,first_name,last_name,email)
    -> Values
    -> (2,'polu','eig','polu@gmail.com'),
    -> (3,'emily','john','emily@gmail.com'),
    -> (4,'raj','john','raj@gmail.com'),
    -> (5,'john','raj','john@gmail.com'),
    -> (6,'hari','raj','hari@gmail.com'),
    -> (7,'ramu','raj','ramu@gmail.com'),
    -> (8,'raju','raja','raju@gmail.com'),
    -> (9,'bheem','raj','bheem@gmail.com'),
    -> (10,'sweety','roy','sweety@gmail.com');
Query OK, 9 rows affected (0.01 sec)
Records: 9  Duplicates: 0  Warnings: 0

mysql> describe 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(100) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> select * from Teachers;
+------------+------------+-----------+------------------+
| teacher_id | first_name | last_name | email            |
+------------+------------+-----------+------------------+
|          1 | polo       | toe       | polo@gmail.com   |
|          2 | polu       | eig       | polu@gmail.com   |
|          3 | emily      | john      | emily@gmail.com  |
|          4 | raj        | john      | raj@gmail.com    |
|          5 | john       | raj       | john@gmail.com   |
|          6 | hari       | raj       | hari@gmail.com   |
|          7 | ramu       | raj       | ramu@gmail.com   |
|          8 | raju       | raja      | raju@gmail.com   |
|          9 | bheem      | raj       | bheem@gmail.com  |
|         10 | sweety     | roy       | sweety@gmail.com |
```

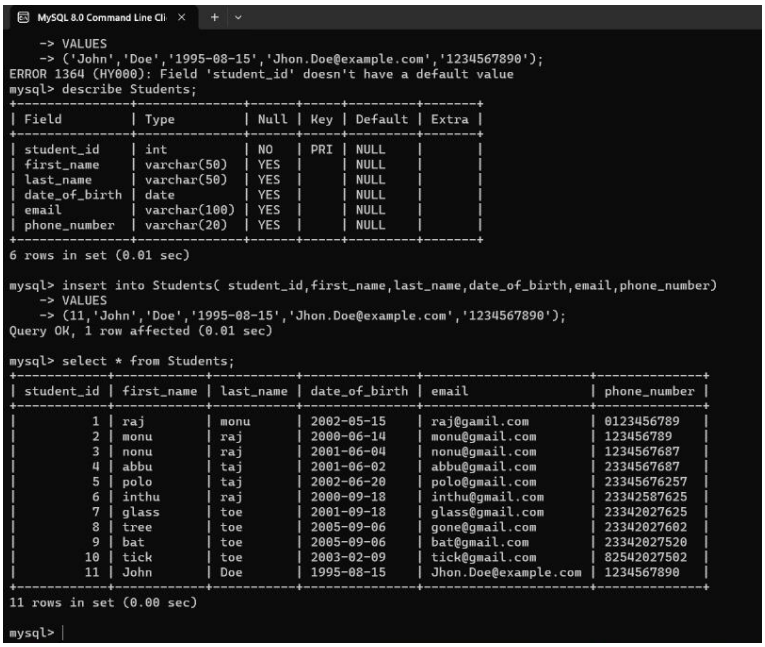
5. Payments:


MySQL 8.0 Command Line Cli
payment_id	int	NO	PRI	NULL	
student_id	int	YES	MUL	NULL	
amount	decimal(10,2)	YES		NULL	
payment_date	date	YES		NULL	
4 rows in set (0.00 sec)
mysql> insert into Payments(payment_id,student_id,amount,payment_date)
-> VALUES
-> (1,1,50,'2023-01-05'),
-> (2,2,75,'2023-01-10'),
-> (3,3,80,'2023-02-2'),
-> (4,4,40,'2023-09-18'),
-> (5,5,50,'2023-08-11'),
-> (6,6,60,'2023-02-1'),
-> (7,7,70,'2023-05-9'),
-> (8,8,50,'2023-03-4'),
-> (9,9,90,'2023-06-1'),
-> (10,10,100,'2023-07-10');
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
1	1	50.00	2023-01-05
2	2	75.00	2023-01-10
3	3	80.00	2023-02-02
4	4	40.00	2023-09-18
5	5	50.00	2023-08-11
6	6	60.00	2023-02-01
7	7	70.00	2023-05-09
8	8	50.00	2023-03-04
9	9	90.00	2023-06-01
10	10	100.00	2023-07-10
10 rows in set (0.00 sec)
mysql>

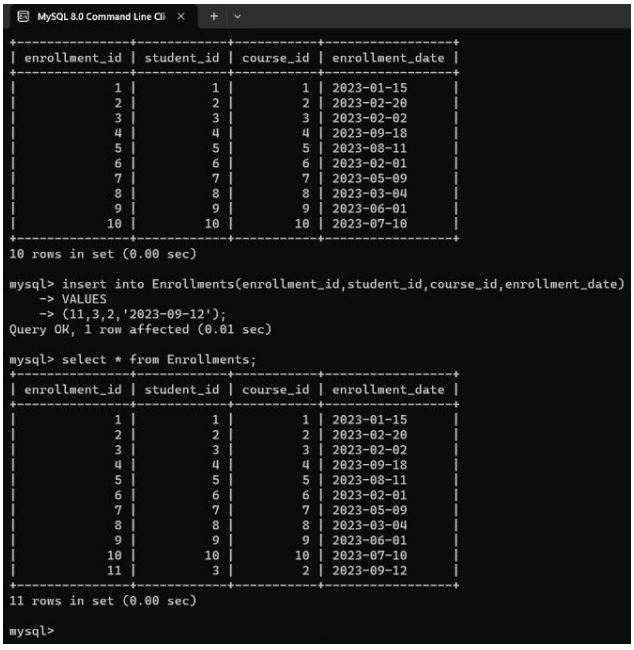
Task 2:

Select, Where, Between, And,Like

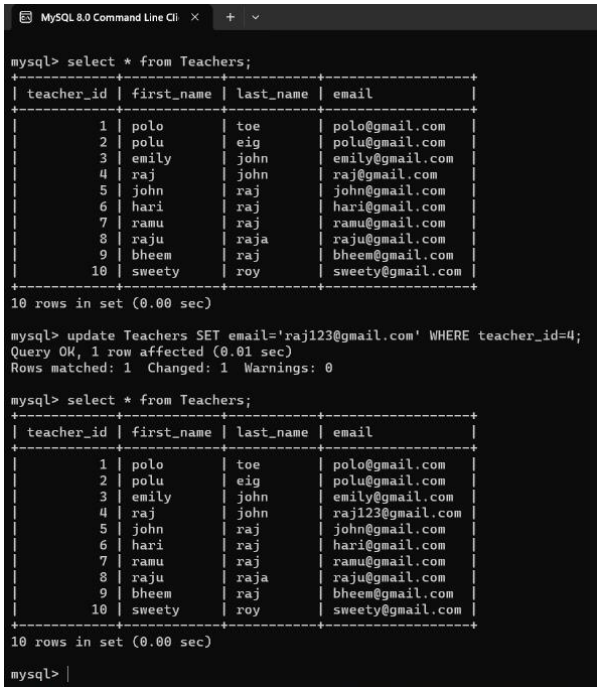
1. Write an SQL query to insert a new “Student” table with the following details
   a. First Name: John
   b. Last name: doe
   c. Date of Birth: 1995-08-15
   d. Phone number:123456790



2. Write an SQL quary 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.?

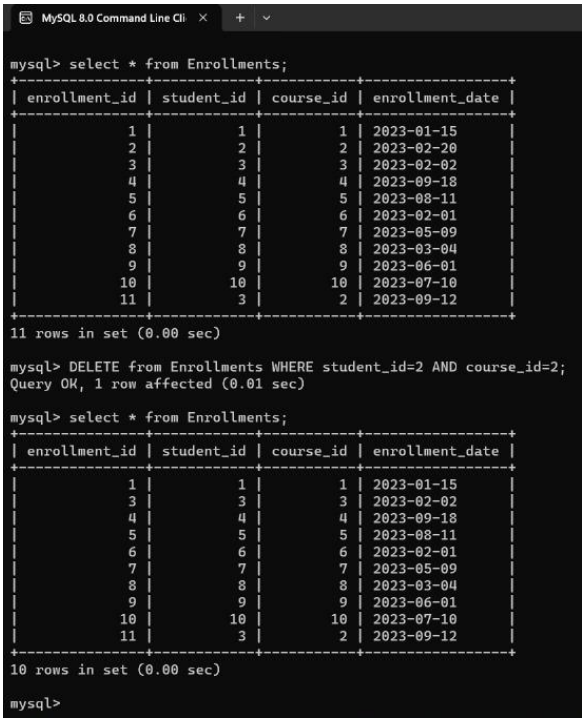


3. Update the email address of a specific teacher in the “Teacher” table.Choose any teacher and modify their email address.?



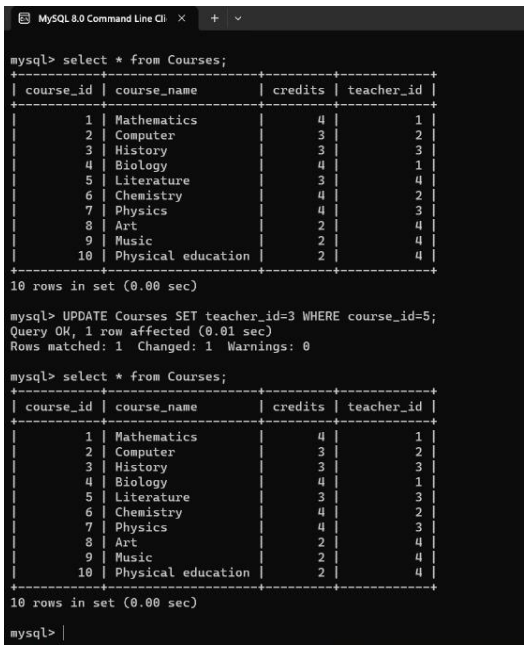
```
MySQL 8.0 Command Line Cli

mysql> select * from Teachers;
+------------+------------+-----------+------------------+
| teacher_id | first_name | last_name | email            |
+------------+------------+-----------+------------------+
|          1 | polo       | toe       | polo@gmail.com   |
|          2 | polu       | eig       | polu@gmail.com   |
|          3 | emily      | john      | emily@gmail.com  |
|          4 | raj        | john      | raj@gmail.com    |
|          5 | john       | raj       | john@gmail.com   |
|          6 | hari       | raj       | hari@gmail.com   |
|          7 | ramu       | raj       | ramu@gmail.com   |
|          8 | raju       | raja      | raju@gmail.com   |
|          9 | bheem      | raj       | bheem@gmail.com  |
|         10 | sweety     | roy       | sweety@gmail.com |
+------------+------------+-----------+------------------+
10 rows in set (0.00 sec)

mysql> update Teachers SET email='raj123@gmail.com' WHERE teacher_id=4;
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 | polo       | toe       | polo@gmail.com   |
|          2 | polu       | eig       | polu@gmail.com   |
|          3 | emily      | john      | emily@gmail.com  |
|          4 | raj        | john      | raj123@gmail.com |
|          5 | john       | raj       | john@gmail.com   |
|          6 | hari       | raj       | hari@gmail.com   |
|          7 | ramu       | raj       | ramu@gmail.com   |
|          8 | raju       | raja      | raju@gmail.com   |
|          9 | bheem      | raj       | bheem@gmail.com  |
|         10 | sweety     | roy       | sweety@gmail.com |
+------------+------------+-----------+------------------+
10 rows in set (0.00 sec)

mysql>
```

4. Write an SQL query to delete a specific enrollment record from the “Enrollment” table. Select an enrollment record based on the student and cource.



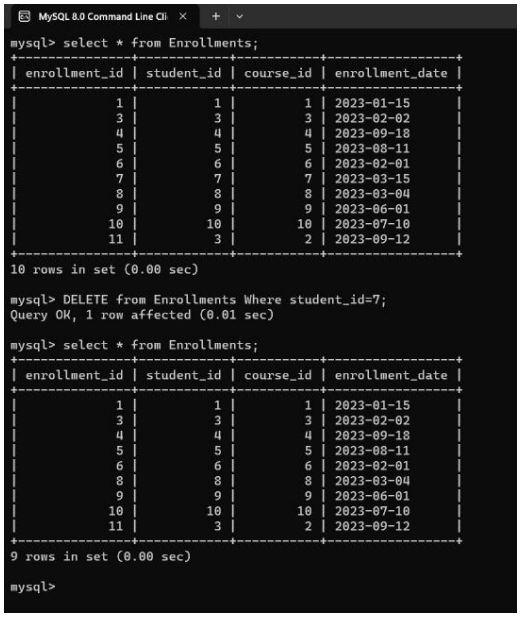
```
MySQL 8.0 Command Line Cli

mysql> select * from Enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2023-01-15      |
|             2 |          2 |         2 | 2023-02-20      |
|             3 |          3 |         3 | 2023-02-02      |
|             4 |          4 |         4 | 2023-09-18      |
|             5 |          5 |         5 | 2023-08-11      |
|             6 |          6 |         6 | 2023-02-01      |
|             7 |          7 |         7 | 2023-05-09      |
|             8 |          8 |         8 | 2023-03-04      |
|             9 |          9 |         9 | 2023-06-01      |
|            10 |         10 |        10 | 2023-07-10      |
|            11 |          3 |         2 | 2023-09-12      |
+---------------+------------+-----------+-----------------+
11 rows in set (0.00 sec)

mysql> DELETE from Enrollments WHERE student_id=2 AND course_id=2;
Query OK, 1 row affected (0.01 sec)

mysql> select * from Enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2023-01-15      |
|             3 |          3 |         3 | 2023-02-02      |
|             4 |          4 |         4 | 2023-09-18      |
|             5 |          5 |         5 | 2023-08-11      |
|             6 |          6 |         6 | 2023-02-01      |
|             7 |          7 |         7 | 2023-05-09      |
|             8 |          8 |         8 | 2023-03-04      |
|             9 |          9 |         9 | 2023-06-01      |
|            10 |         10 |        10 | 2023-07-10      |
|            11 |          3 |         2 | 2023-09-12      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)

mysql>
```

5. Update the “Cource” table to assign a specific teacher to a cource. Choose any course teacher from the respective tables.



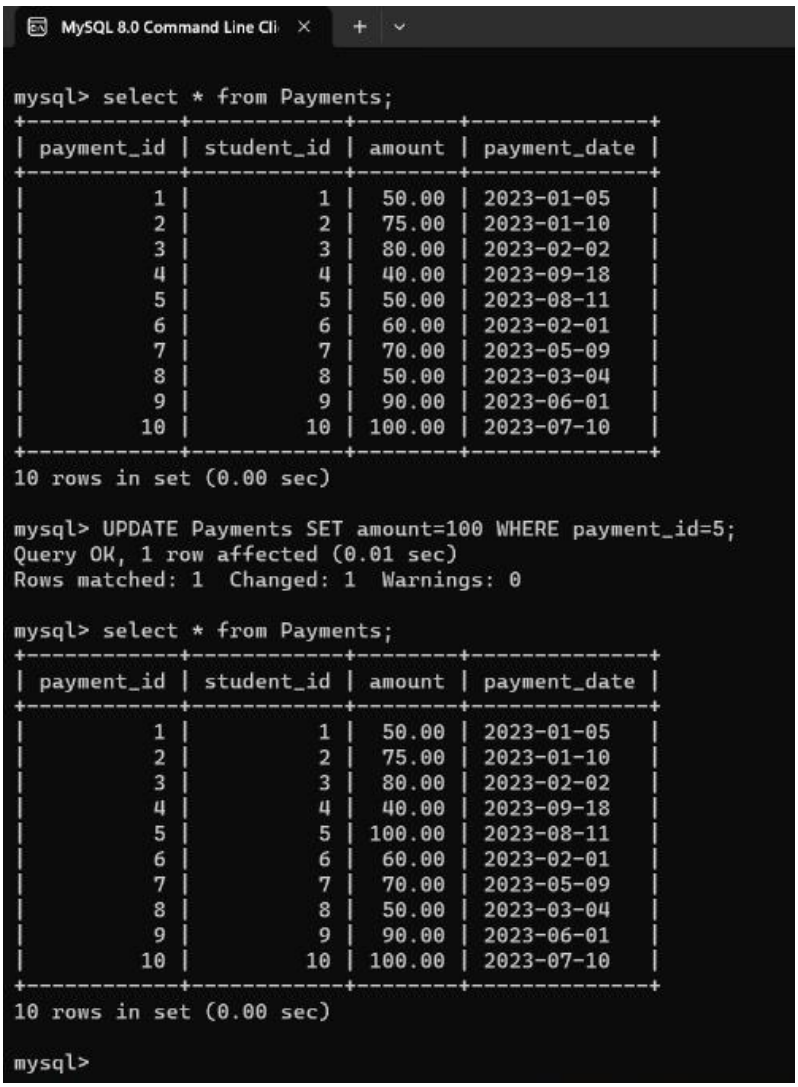
```
MySQL 8.0 Command Line Cli

mysql> select * from Courses;
+-----------+--------------------+---------+------------+
| course_id | course_name        | credits | teacher_id |
+-----------+--------------------+---------+------------+
|         1 | Mathematics        |       4 |          1 |
|         2 | Computer           |       3 |          2 |
|         3 | History            |       3 |          3 |
|         4 | Biology            |       4 |          1 |
|         5 | Literature         |       3 |          4 |
|         6 | Chemistry          |       4 |          2 |
|         7 | Physics            |       4 |          3 |
|         8 | Art                |       2 |          4 |
|         9 | Music              |       2 |          4 |
|        10 | Physical education |       2 |          4 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)

mysql> UPDATE Courses SET teacher_id=3 WHERE course_id=5;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Courses;
+-----------+--------------------+---------+------------+
| course_id | course_name        | credits | teacher_id |
+-----------+--------------------+---------+------------+
|         1 | Mathematics        |       4 |          1 |
|         2 | Computer           |       3 |          2 |
|         3 | History            |       3 |          3 |
|         4 | Biology            |       4 |          1 |
|         5 | Literature         |       3 |          3 |
|         6 | Chemistry          |       4 |          2 |
|         7 | Physics            |       4 |          3 |
|         8 | Art                |       2 |          4 |
|         9 | Music              |       2 |          4 |
|        10 | Physical education |       2 |          4 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)

mysql>
```

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.



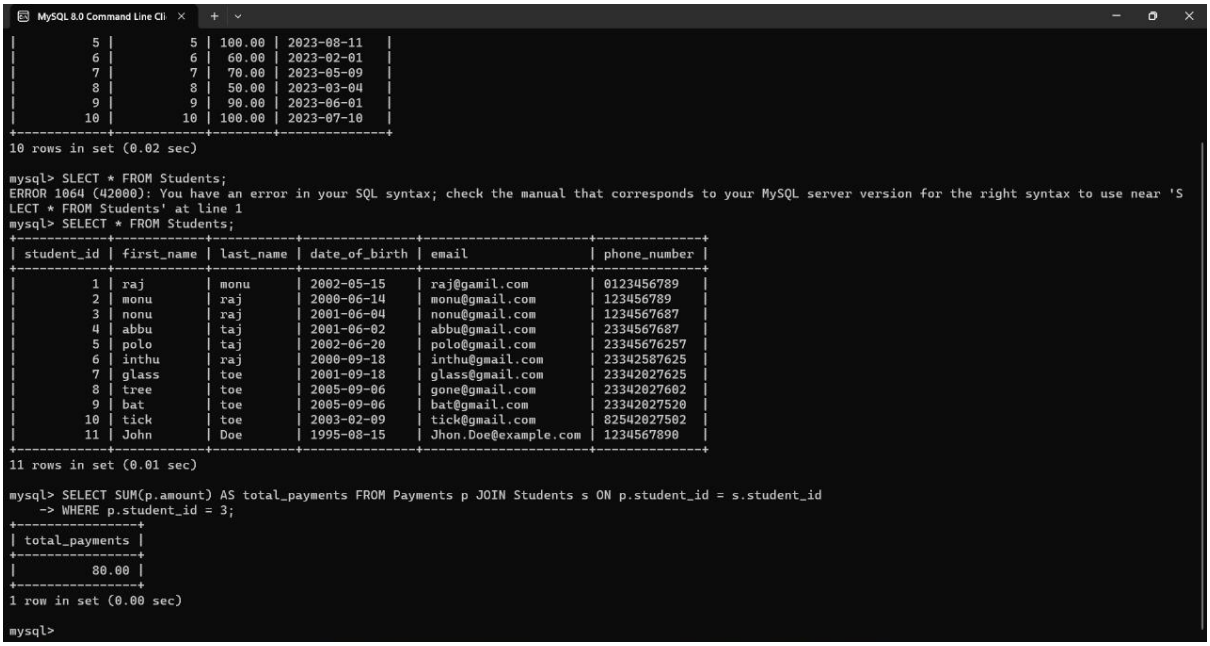
```
MySQL 8.0 Command Line Cli

mysql> select * from Enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2023-01-15      |
|             3 |          3 |         3 | 2023-02-02      |
|             4 |          4 |         4 | 2023-09-18      |
|             5 |          5 |         5 | 2023-08-11      |
|             6 |          6 |         6 | 2023-02-01      |
|             7 |          7 |         7 | 2023-03-15      |
|             8 |          8 |         8 | 2023-03-04      |
|             9 |          9 |         9 | 2023-06-01      |
|            10 |         10 |        10 | 2023-07-10      |
|            11 |          3 |         2 | 2023-09-12      |
+---------------+------------+-----------+-----------------+
10 rows in set (0.00 sec)

mysql> DELETE from Enrollments Where student_id=7;
Query OK, 1 row affected (0.01 sec)

mysql> select * from Enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2023-01-15      |
|             3 |          3 |         3 | 2023-02-02      |
|             4 |          4 |         4 | 2023-09-18      |
|             5 |          5 |         5 | 2023-08-11      |
|             6 |          6 |         6 | 2023-02-01      |
|             8 |          8 |         8 | 2023-03-04      |
|             9 |          9 |         9 | 2023-06-01      |
|            10 |         10 |        10 | 2023-07-10      |
|            11 |          3 |         2 | 2023-09-12      |
+---------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)

mysql>
```

7. Update the payment amount for specific payment record in the “Payment” table. Choose any payment record and modify the payment amount.

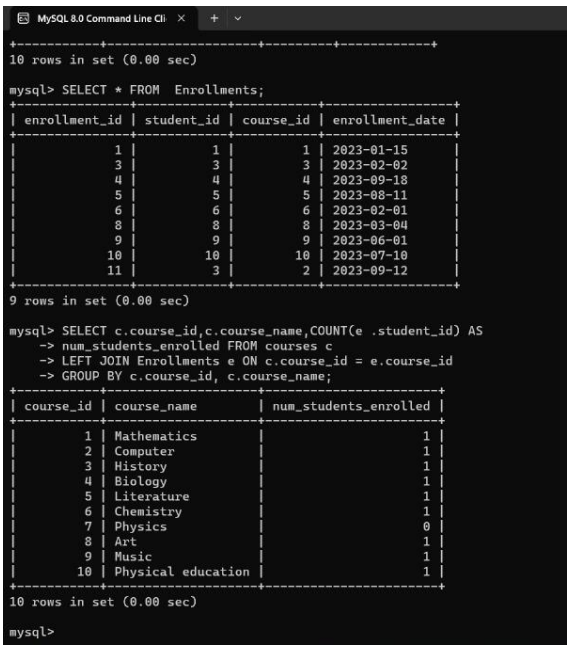


Task-3

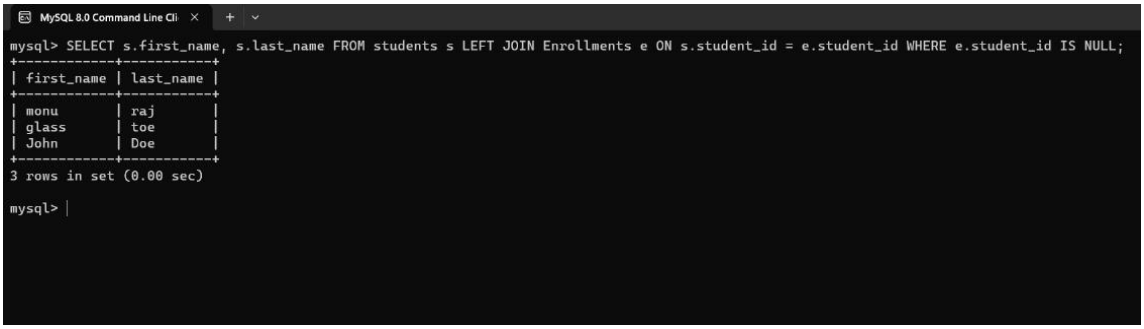
1. Write an SQL query to calculate the total payments made by a specific student. You will need to join the “payments” table with “Students” table based on the students’s ID.



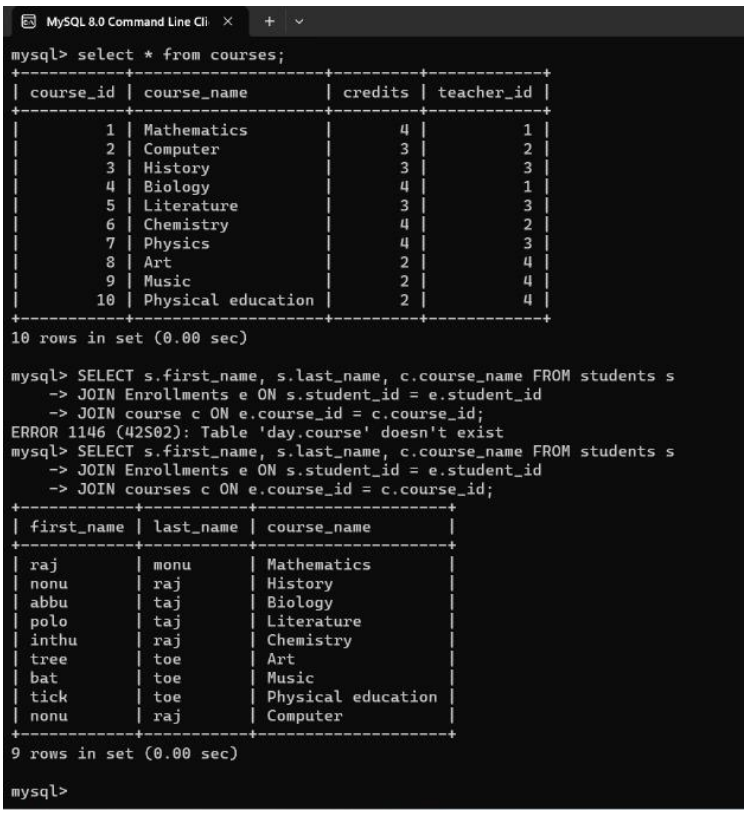
2. Write an SQL query to retrieve a list of courses along with the count of students enrolled in each course. Use a JOIN operation between the "Courses" table and the "Enrollments" table.



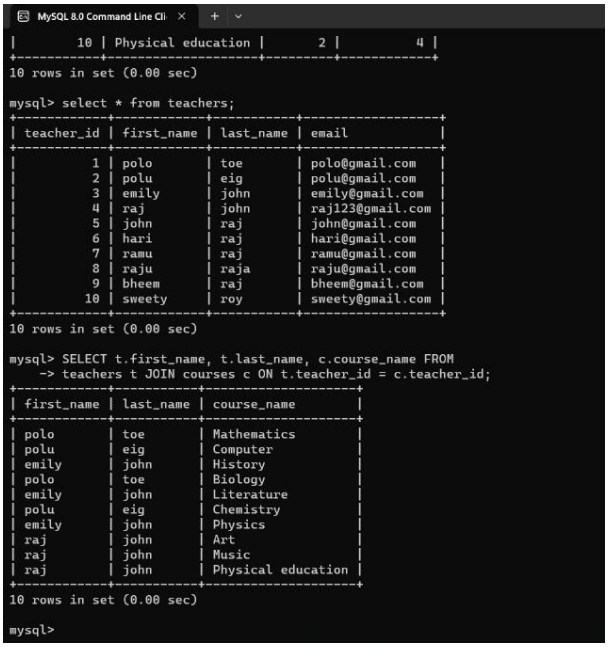
3. Write an SQL query to find the names of students who have not enrolled in any course. Use a LEFT JOIN between the "Students" table and the "Enrollments" table to identify students without enrollments.



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.

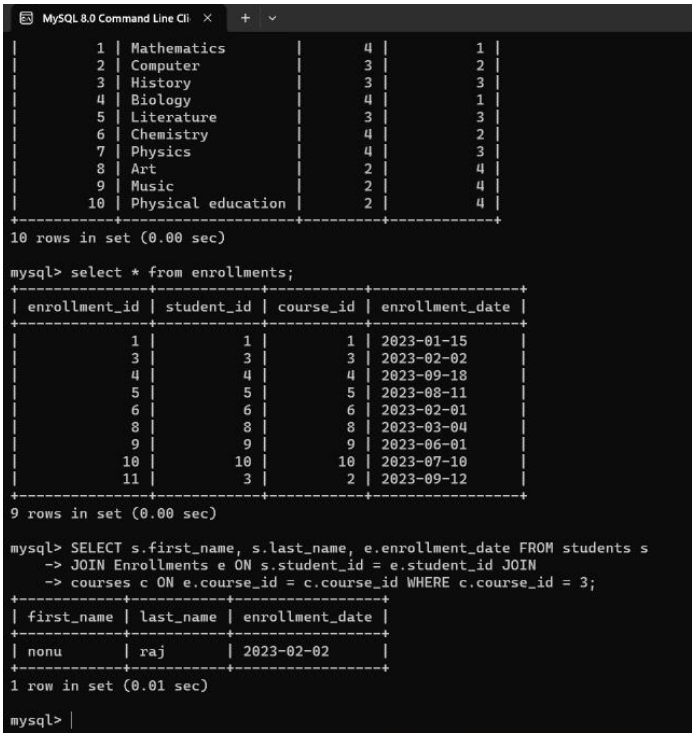


5. Create a query to list the names of teachers and the courses they are assigned to. Join the "Teacher" table with the "Courses" table.



```
MySQL 8.0 Command Line Cli
|        10 | Physical education |         2 |          4 |
+-----------+--------------------+-----------+------------+
10 rows in set (0.00 sec)

mysql> select * from teachers;
+------------+------------+-----------+------------------+
| teacher_id | first_name | last_name | email            |
+------------+------------+-----------+------------------+
|          1 | polo       | toe       | polo@gmail.com   |
|          2 | polu       | eig       | polu@gmail.com   |
|          3 | emily      | john      | emily@gmail.com  |
|          4 | raj        | john      | raj123@gmail.com |
|          5 | john       | raj       | john@gmail.com   |
|          6 | hari       | raj       | hari@gmail.com   |
|          7 | ramu       | raj       | ramu@gmail.com   |
|          8 | raju       | raja      | raju@gmail.com   |
|          9 | bheem      | raj       | bheem@gmail.com  |
|         10 | sweety     | roy       | sweety@gmail.com |
+------------+------------+-----------+------------------+
10 rows in set (0.00 sec)

mysql> SELECT t.first_name, t.last_name, c.course_name FROM
    -> teachers t JOIN courses c ON t.teacher_id = c.teacher_id;
+------------+-----------+--------------------+
| first_name | last_name | course_name        |
+------------+-----------+--------------------+
| polo       | toe       | Mathematics        |
| polu       | eig       | Computer           |
| emily      | john      | History            |
| polo       | toe       | Biology            |
| emily      | john      | Literature         |
| polu       | eig       | Chemistry          |
| emily      | john      | Physics            |
| raj        | john      | Art                |
| raj        | john      | Music              |
| raj        | john      | Physical education |
+------------+-----------+--------------------+
10 rows in set (0.00 sec)

mysql>
```

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.

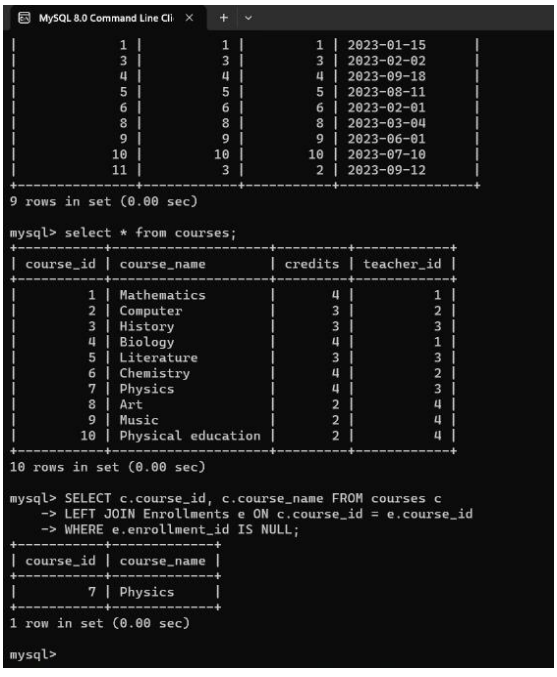


```
MySQL 8.0 Command Line Cli
|         1 | Mathematics        |         4 |          1 |
|         2 | Computer           |         3 |          2 |
|         3 | History            |         3 |          3 |
|         4 | Biology            |         4 |          1 |
|         5 | Literature         |         3 |          3 |
|         6 | Chemistry          |         4 |          2 |
|         7 | Physics            |         4 |          3 |
|         8 | Art                |         2 |          4 |
|         9 | Music              |         2 |          4 |
|        10 | Physical education |         2 |          4 |
+-----------+--------------------+-----------+------------+
10 rows in set (0.00 sec)

mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2023-01-15      |
|             3 |          3 |         3 | 2023-02-02      |
|             4 |          4 |         4 | 2023-09-18      |
|             5 |          5 |         5 | 2023-08-11      |
|             6 |          6 |         6 | 2023-02-01      |
|             8 |          8 |         8 | 2023-03-04      |
|             9 |          9 |         9 | 2023-06-01      |
|            10 |         10 |        10 | 2023-07-10      |
|            11 |          3 |         2 | 2023-09-12      |
+---------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)

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 = 3;
+------------+-----------+-----------------+
| first_name | last_name | enrollment_date |
+------------+-----------+-----------------+
| nonu       | raj       | 2023-02-02      |
+------------+-----------+-----------------+
1 row in set (0.01 sec)

mysql>
```

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.

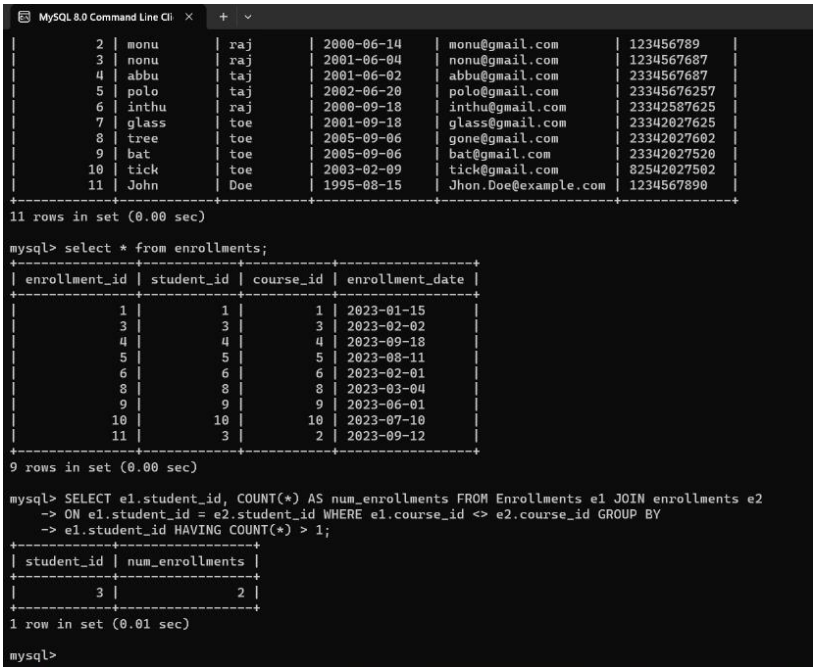
```
MySQL 8.0 Command Line Cli
+---------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)

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 = 3;
+------------+-----------+-----------------+
| first_name | last_name | enrollment_date |
+------------+-----------+-----------------+
| nonu       | raj       | 2023-02-02      |
+------------+-----------+-----------------+
1 row in set (0.01 sec)

mysql> select * from payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          1 |          1 |  50.00 | 2023-01-05   |
|          2 |          2 |  75.00 | 2023-01-10   |
|          3 |          3 |  80.00 | 2023-02-02   |
|          4 |          4 |  40.00 | 2023-09-18   |
|          5 |          5 | 100.00 | 2023-08-11   |
|          6 |          6 |  60.00 | 2023-02-01   |
|          7 |          7 |  70.00 | 2023-05-09   |
|          8 |          8 |  50.00 | 2023-03-04   |
|          9 |          9 |  90.00 | 2023-06-01   |
|         10 |         10 | 100.00 | 2023-07-10   |
+------------+------------+--------+--------------+
10 rows in set (0.01 sec)

mysql> SELECT s.first_name, s.last_name FROM students s LEFT JOIN payments p
    -> ON s.student_id = p.student_id WHERE p.payment_id IS NULL;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| John       | Doe       |
+------------+-----------+
1 row in set (0.00 sec)

mysql>
```

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.



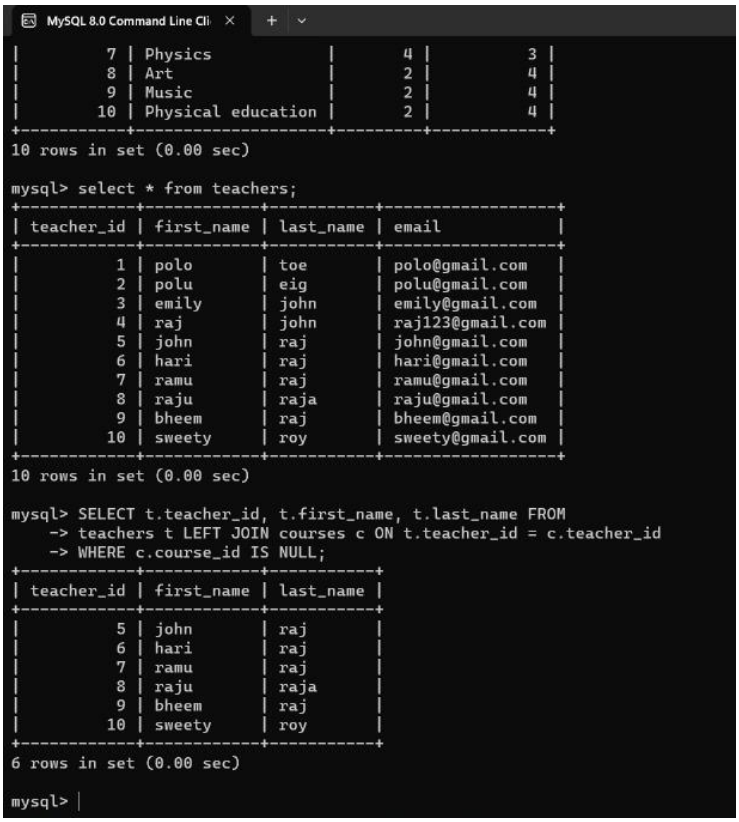
```
MySQL 8.0 Command Line Cli
|            1 |          1 |         1 | 2023-01-15      |
|            3 |          3 |         3 | 2023-02-02      |
|            4 |          4 |         4 | 2023-09-18      |
|            5 |          5 |         5 | 2023-08-11      |
|            6 |          6 |         6 | 2023-02-01      |
|            8 |          8 |         8 | 2023-03-04      |
|            9 |          9 |         9 | 2023-06-01      |
|           10 |         10 |        10 | 2023-07-10      |
|           11 |          3 |         2 | 2023-09-12      |
+--------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)

mysql> select * from courses;
+-----------+--------------------+---------+------------+
| course_id | course_name        | credits | teacher_id |
+-----------+--------------------+---------+------------+
|         1 | Mathematics        |       4 |          1 |
|         2 | Computer           |       3 |          2 |
|         3 | History            |       3 |          3 |
|         4 | Biology            |       4 |          1 |
|         5 | Literature         |       3 |          3 |
|         6 | Chemistry          |       4 |          2 |
|         7 | Physics            |       4 |          3 |
|         8 | Art                |       2 |          4 |
|         9 | Music              |       2 |          4 |
|        10 | Physical education |       2 |          4 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)

mysql> SELECT c.course_id, c.course_name FROM courses c
    -> LEFT JOIN Enrollments e ON c.course_id = e.course_id
    -> WHERE e.enrollment_id IS NULL;
+-----------+-------------+
| course_id | course_name |
+-----------+-------------+
|         7 | Physics     |
+-----------+-------------+
1 row in set (0.00 sec)

mysql>
```

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 8.0 Command Line Cli

|          2 | monu       | raj       | 2000-06-14    | monu@gmail.com       | 1234567890    |
|          3 | nonu       | raj       | 2001-06-04    | nonu@gmail.com       | 1234567687    |
|          4 | abbu       | taj       | 2001-06-02    | abbu@gmail.com       | 2334567687    |
|          5 | polo       | taj       | 2002-06-20    | polo@gmail.com       | 23345676257   |
|          6 | inthu      | raj       | 2000-09-18    | inthu@gmail.com      | 23342587625   |
|          7 | glass      | toe       | 2001-09-18    | glass@gmail.com      | 23342027625   |
|          8 | tree       | toe       | 2005-09-06    | gone@gmail.com       | 23342027602   |
|          9 | bat        | toe       | 2005-09-06    | bat@gmail.com        | 23342027520   |
|         10 | tick       | toe       | 2003-02-09    | tick@gmail.com       | 82542027502   |
|         11 | John       | Doe       | 1995-08-15    | Jhon.Doe@example.com | 1234567890    |
+------------+------------+-----------+---------------+----------------------+---------------+
11 rows in set (0.00 sec)

mysql> select * from enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2023-01-15      |
|             3 |          3 |         3 | 2023-02-02      |
|             4 |          4 |         4 | 2023-09-18      |
|             5 |          5 |         5 | 2023-08-11      |
|             6 |          6 |         6 | 2023-02-01      |
|             8 |          8 |         8 | 2023-03-04      |
|             9 |          9 |         9 | 2023-06-01      |
|            10 |         10 |        10 | 2023-07-10      |
|            11 |          3 |         2 | 2023-09-12      |
+---------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)

mysql> SELECT e1.student_id, COUNT(*) AS num_enrollments FROM Enrollments e1 JOIN enrollments e2
    -> ON e1.student_id = e2.student_id WHERE e1.course_id <> e2.course_id GROUP BY
    -> e1.student_id HAVING COUNT(*) > 1;
+------------+-----------------+
| student_id | num_enrollments |
+------------+-----------------+
|          3 |               2 |
+------------+-----------------+
1 row in set (0.01 sec)

mysql>
```

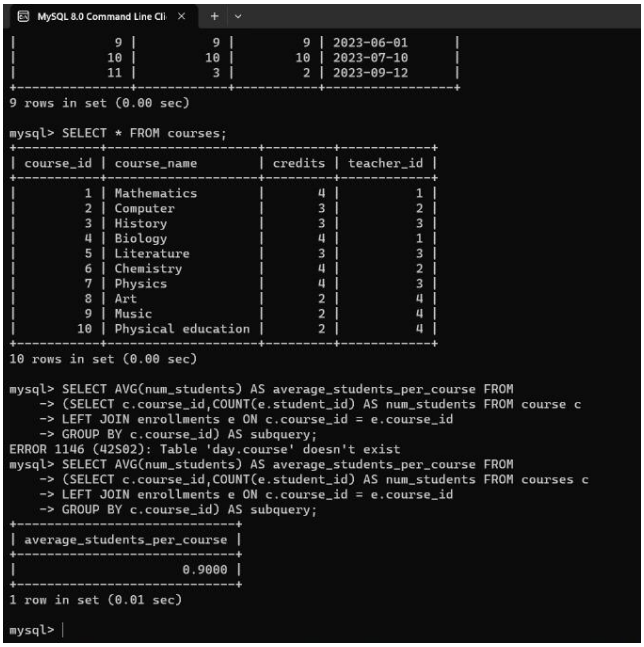
10. Find teachers who are not assigned to any courses. Use a LEFT JOIN between the "Teacher" table and the "Courses" table and filter for teachers with NULL course assignments.



```
MySQL 8.0 Command Line Cli

|          7 | Physics            |         4 |           3 |
|          8 | Art                |         2 |           4 |
|          9 | Music              |         2 |           4 |
|         10 | Physical education |         2 |           4 |
+------------+--------------------+-----------+-------------+
10 rows in set (0.00 sec)

mysql> select * from teachers;
+------------+------------+-----------+-------------------+
| teacher_id | first_name | last_name | email             |
+------------+------------+-----------+-------------------+
|          1 | polo       | toe       | polo@gmail.com    |
|          2 | polu       | eig       | polu@gmail.com    |
|          3 | emily      | john      | emily@gmail.com   |
|          4 | raj        | john      | raj123@gmail.com  |
|          5 | john       | raj       | john@gmail.com    |
|          6 | hari       | raj       | hari@gmail.com    |
|          7 | ramu       | raj       | ramu@gmail.com    |
|          8 | raju       | raja      | raju@gmail.com    |
|          9 | bheem      | raj       | bheem@gmail.com   |
|         10 | sweety     | roy       | sweety@gmail.com  |
+------------+------------+-----------+-------------------+
10 rows in set (0.00 sec)

mysql> SELECT t.teacher_id, t.first_name, t.last_name FROM
    -> teachers t LEFT JOIN courses c ON t.teacher_id = c.teacher_id
    -> WHERE c.course_id IS NULL;
+------------+------------+-----------+
| teacher_id | first_name | last_name |
+------------+------------+-----------+
|          5 | john       | raj       |
|          6 | hari       | raj       |
|          7 | ramu       | raj       |
|          8 | raju       | raja      |
|          9 | bheem      | raj       |
|         10 | sweety     | roy       |
+------------+------------+-----------+
6 rows in set (0.00 sec)

mysql>
```

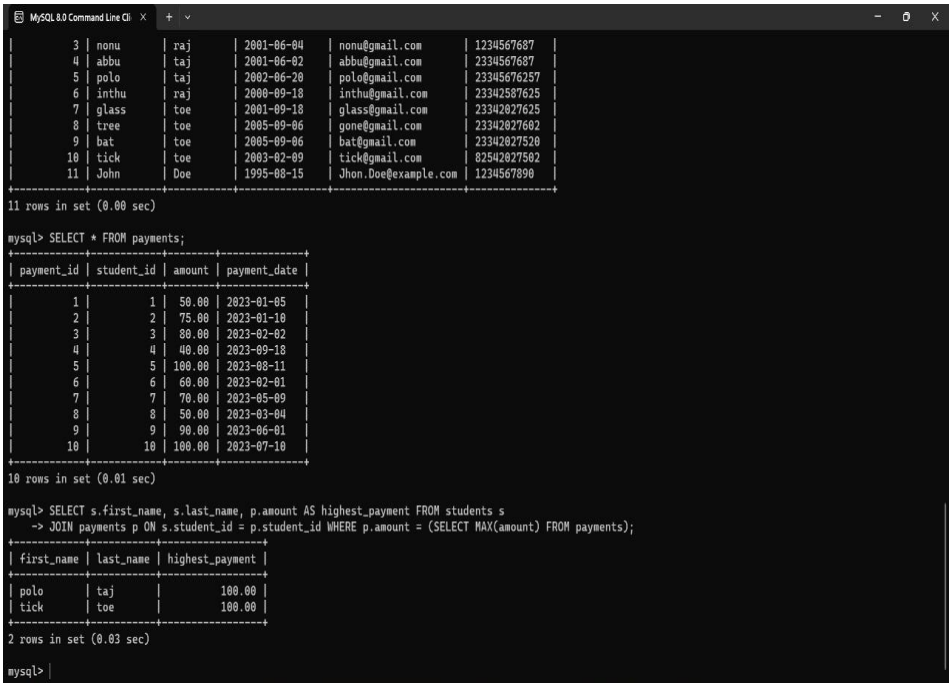
Task 4:

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



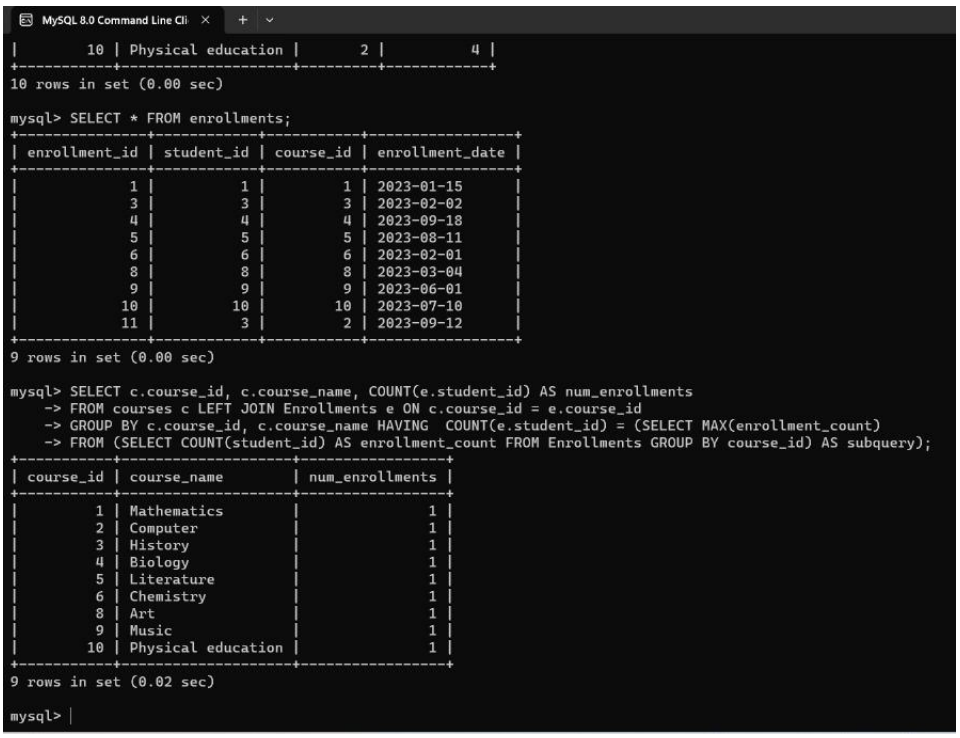
```
MySQL 8.0 Command Line Cli

|            9 |          9 |         9 | 2023-06-01      |
|           10 |         10 |        10 | 2023-07-10      |
|           11 |          3 |         2 | 2023-09-12      |
+--------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)

mysql> SELECT * FROM courses;
+-----------+--------------------+---------+------------+
| course_id | course_name        | credits | teacher_id |
+-----------+--------------------+---------+------------+
|         1 | Mathematics        |       4 |          1 |
|         2 | Computer           |       3 |          2 |
|         3 | History            |       3 |          3 |
|         4 | Biology            |       4 |          1 |
|         5 | Literature         |       3 |          3 |
|         6 | Chemistry          |       4 |          2 |
|         7 | Physics            |       4 |          3 |
|         8 | Art                |       2 |          4 |
|         9 | Music              |       2 |          4 |
|        10 | Physical education |       2 |          4 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)

mysql> SELECT AVG(num_students) AS average_students_per_course FROM
    -> (SELECT c.course_id,COUNT(e.student_id) AS num_students FROM course c
    -> LEFT JOIN enrollments e ON c.course_id = e.course_id
    -> GROUP BY c.course_id) AS subquery;
ERROR 1146 (42S02): Table 'day.course' doesn't exist
mysql> SELECT AVG(num_students) AS average_students_per_course FROM
    -> (SELECT c.course_id,COUNT(e.student_id) AS num_students FROM courses c
    -> LEFT JOIN enrollments e ON c.course_id = e.course_id
    -> GROUP BY c.course_id) AS subquery;
+------------------------------+
| average_students_per_course |
+------------------------------+
|                       0.9000 |
+------------------------------+
1 row in set (0.01 sec)

mysql>
```

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.



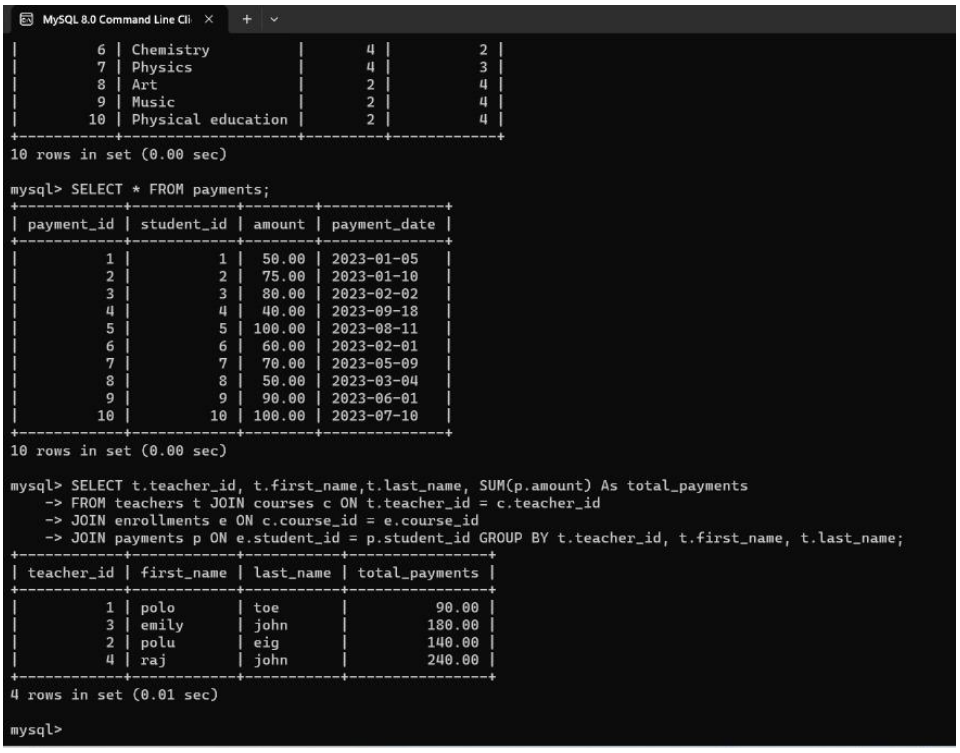
```
MySQL 8.0 Command Line Cli

|         3 | nonu       | raj       | 2001-06-04    | nonu@gmail.com       | 1234567687   |
|         4 | abbu       | taj       | 2001-06-02    | abbu@gmail.com       | 2334567687   |
|         5 | polo       | taj       | 2002-06-20    | polo@gmail.com       | 23345676257  |
|         6 | inthu      | raj       | 2000-09-18    | inthu@gmail.com      | 23342587625  |
|         7 | glass      | toe       | 2001-09-18    | glass@gmail.com      | 23342027625  |
|         8 | tree       | toe       | 2005-09-06    | gone@gmail.com       | 23342027602  |
|         9 | bat        | toe       | 2005-09-06    | bat@gmail.com        | 23342027520  |
|        10 | tick       | toe       | 2003-02-09    | tick@gmail.com       | 82542027502  |
|        11 | John       | Doe       | 1995-08-15    | Jhon.Doe@example.com | 1234567890   |
+-----------+------------+-----------+---------------+----------------------+--------------+
11 rows in set (0.00 sec)

mysql> SELECT * FROM payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          1 |          1 |  50.00 | 2023-01-05   |
|          2 |          2 |  75.00 | 2023-01-10   |
|          3 |          3 |  80.00 | 2023-02-02   |
|          4 |          4 |  40.00 | 2023-09-18   |
|          5 |          5 | 100.00 | 2023-08-11   |
|          6 |          6 |  60.00 | 2023-02-01   |
|          7 |          7 |  70.00 | 2023-05-09   |
|          8 |          8 |  50.00 | 2023-03-04   |
|          9 |          9 |  90.00 | 2023-06-01   |
|         10 |         10 | 100.00 | 2023-07-10   |
+------------+------------+--------+--------------+
10 rows in set (0.01 sec)

mysql> SELECT s.first_name, s.last_name, p.amount AS highest_payment FROM students s
    -> JOIN payments p ON s.student_id = p.student_id WHERE p.amount = (SELECT MAX(amount) FROM payments);
+------------+-----------+-----------------+
| first_name | last_name | highest_payment |
+------------+-----------+-----------------+
| polo       | taj       |          100.00 |
| tick       | toe       |          100.00 |
+------------+-----------+-----------------+
2 rows in set (0.03 sec)

mysql>
```

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



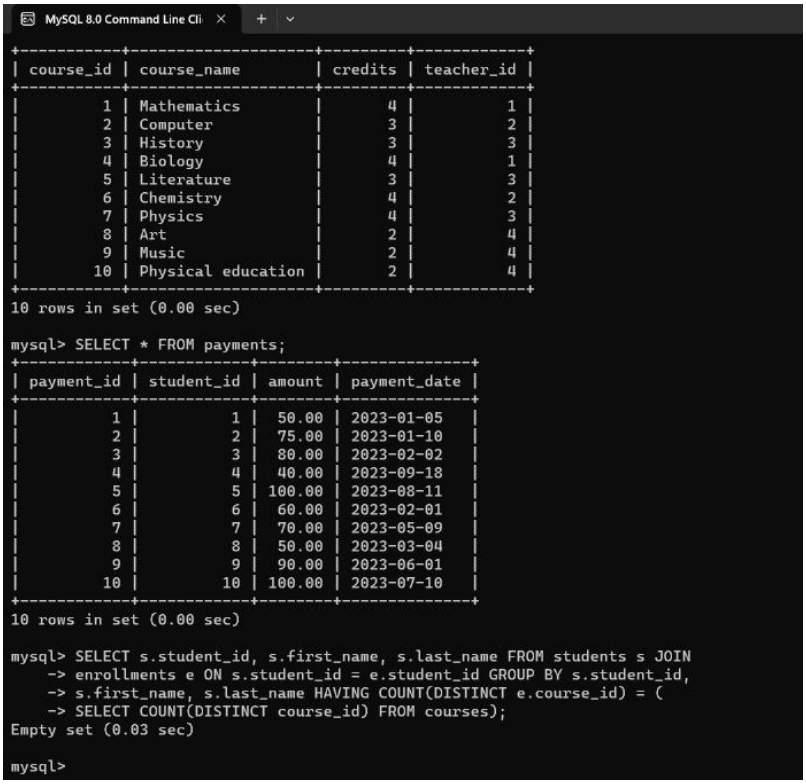
```
MySQL 8.0 Command Line Cli

|        10 | Physical education |       2 |          4 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2023-01-15      |
|             3 |          3 |         3 | 2023-02-02      |
|             4 |          4 |         4 | 2023-09-18      |
|             5 |          5 |         5 | 2023-08-11      |
|             6 |          6 |         6 | 2023-02-01      |
|             8 |          8 |         8 | 2023-03-04      |
|             9 |          9 |         9 | 2023-06-01      |
|            10 |         10 |        10 | 2023-07-10      |
|            11 |          3 |         2 | 2023-09-12      |
+---------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)

mysql> SELECT c.course_id, c.course_name, COUNT(e.student_id) AS num_enrollments
    -> FROM courses c LEFT JOIN Enrollments e ON c.course_id = e.course_id
    -> GROUP BY c.course_id, c.course_name HAVING  COUNT(e.student_id) = (SELECT MAX(enrollment_count)
    -> FROM (SELECT COUNT(student_id) AS enrollment_count FROM Enrollments GROUP BY course_id) AS subquery);
+-----------+--------------------+-----------------+
| course_id | course_name        | num_enrollments |
+-----------+--------------------+-----------------+
|         1 | Mathematics        |               1 |
|         2 | Computer           |               1 |
|         3 | History            |               1 |
|         4 | Biology            |               1 |
|         5 | Literature         |               1 |
|         6 | Chemistry          |               1 |
|         8 | Art                |               1 |
|         9 | Music              |               1 |
|        10 | Physical education |               1 |
+-----------+--------------------+-----------------+
9 rows in set (0.02 sec)

mysql>
```

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



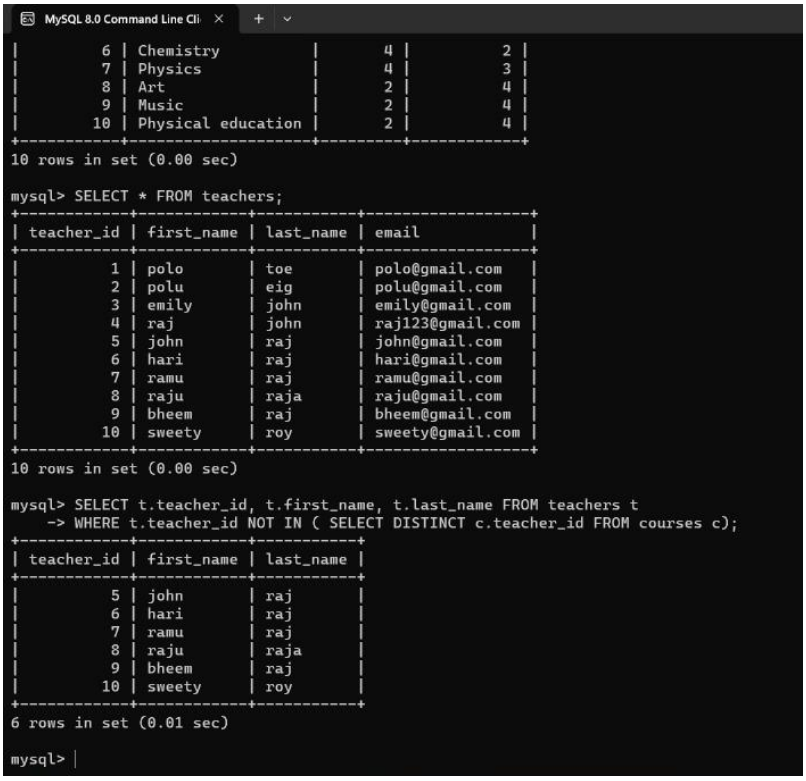
```
MySQL 8.0 Command Line Cli

|         6 | Chemistry          |       4 |          2 |
|         7 | Physics            |       4 |          3 |
|         8 | Art                |       2 |          4 |
|         9 | Music              |       2 |          4 |
|        10 | Physical education |       2 |          4 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          1 |          1 |  50.00 | 2023-01-05   |
|          2 |          2 |  75.00 | 2023-01-10   |
|          3 |          3 |  80.00 | 2023-02-02   |
|          4 |          4 |  40.00 | 2023-09-18   |
|          5 |          5 | 100.00 | 2023-08-11   |
|          6 |          6 |  60.00 | 2023-02-01   |
|          7 |          7 |  70.00 | 2023-05-09   |
|          8 |          8 |  50.00 | 2023-03-04   |
|          9 |          9 |  90.00 | 2023-06-01   |
|         10 |         10 | 100.00 | 2023-07-10   |
+------------+------------+--------+--------------+
10 rows in set (0.00 sec)

mysql> SELECT t.teacher_id, t.first_name,t.last_name, SUM(p.amount) As total_payments
    -> FROM teachers t JOIN courses c ON t.teacher_id = c.teacher_id
    -> JOIN enrollments e ON c.course_id = e.course_id
    -> JOIN payments p ON e.student_id = p.student_id GROUP BY t.teacher_id, t.first_name, t.last_name;
+------------+------------+-----------+----------------+
| teacher_id | first_name | last_name | total_payments |
+------------+------------+-----------+----------------+
|          1 | polo       | toe       |          90.00 |
|          3 | emily      | john      |         180.00 |
|          2 | polu       | eig       |         140.00 |
|          4 | raj        | john      |         240.00 |
+------------+------------+-----------+----------------+
4 rows in set (0.01 sec)

mysql>
```

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



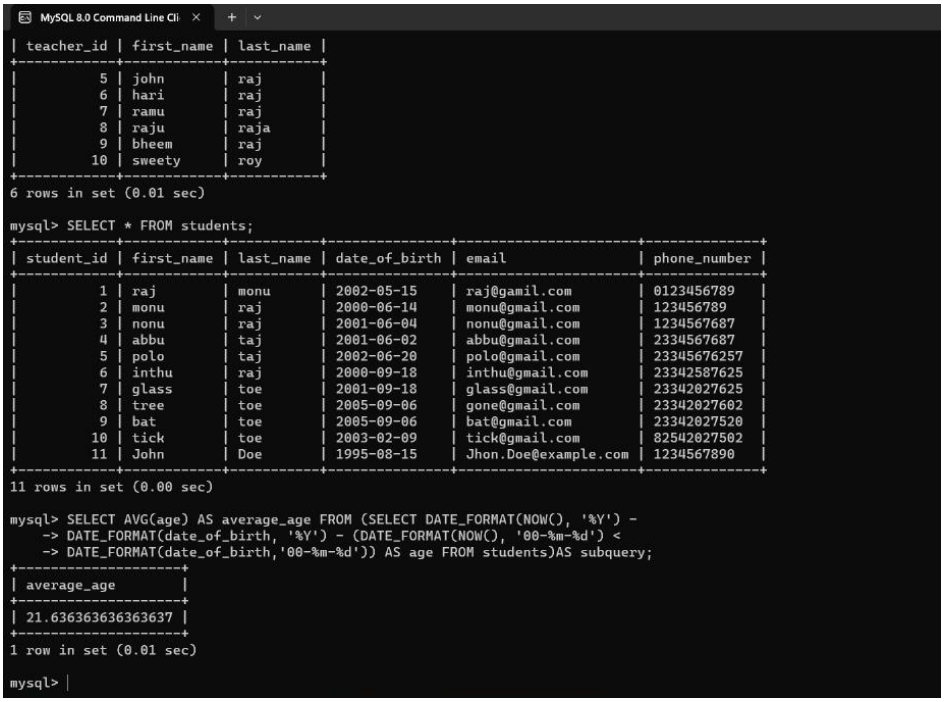
```
MySQL 8.0 Command Line Cli

+-----------+--------------------+---------+------------+
| course_id | course_name        | credits | teacher_id |
+-----------+--------------------+---------+------------+
|         1 | Mathematics        |       4 |          1 |
|         2 | Computer           |       3 |          2 |
|         3 | History            |       3 |          3 |
|         4 | Biology            |       4 |          1 |
|         5 | Literature         |       3 |          3 |
|         6 | Chemistry          |       4 |          2 |
|         7 | Physics            |       4 |          3 |
|         8 | Art                |       2 |          4 |
|         9 | Music              |       2 |          4 |
|        10 | Physical education |       2 |          4 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          1 |          1 |  50.00 | 2023-01-05   |
|          2 |          2 |  75.00 | 2023-01-10   |
|          3 |          3 |  80.00 | 2023-02-02   |
|          4 |          4 |  40.00 | 2023-09-18   |
|          5 |          5 | 100.00 | 2023-08-11   |
|          6 |          6 |  60.00 | 2023-02-01   |
|          7 |          7 |  70.00 | 2023-05-09   |
|          8 |          8 |  50.00 | 2023-03-04   |
|          9 |          9 |  90.00 | 2023-06-01   |
|         10 |         10 | 100.00 | 2023-07-10   |
+------------+------------+--------+--------------+
10 rows in set (0.00 sec)

mysql> SELECT s.student_id, s.first_name, s.last_name FROM students s JOIN
    -> enrollments e ON s.student_id = e.student_id GROUP BY s.student_id,
    -> s.first_name, s.last_name HAVING COUNT(DISTINCT e.course_id) = (
    -> SELECT COUNT(DISTINCT course_id) FROM courses);
Empty set (0.03 sec)

mysql>
```

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



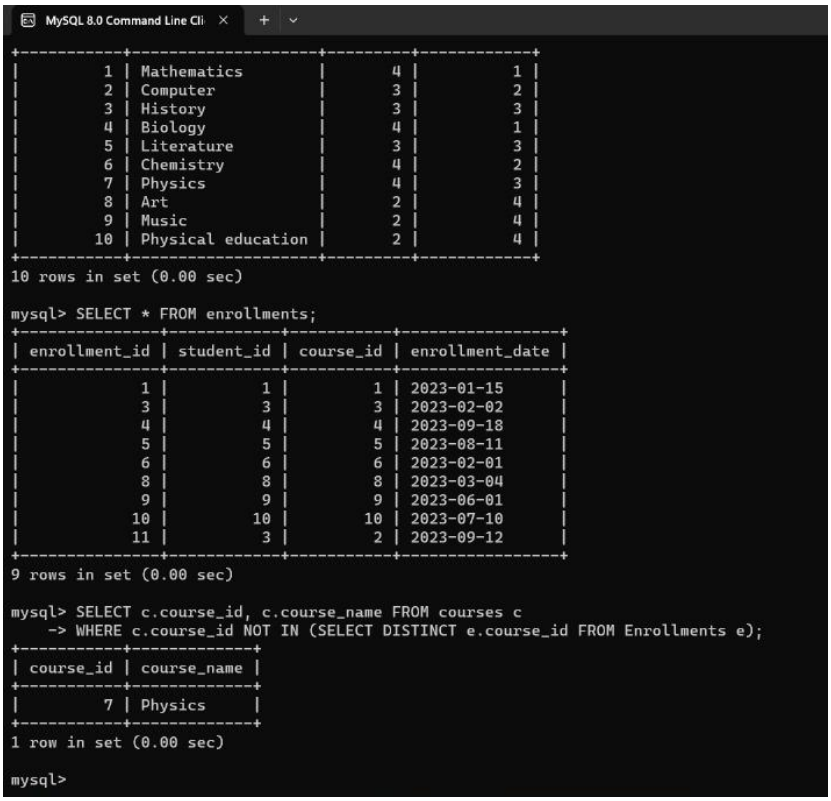
```
MySQL 8.0 Command Line Cli

|         6 | Chemistry          |       4 |          2 |
|         7 | Physics            |       4 |          3 |
|         8 | Art                |       2 |          4 |
|         9 | Music              |       2 |          4 |
|        10 | Physical education |       2 |          4 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM teachers;
+------------+------------+-----------+-------------------+
| teacher_id | first_name | last_name | email             |
+------------+------------+-----------+-------------------+
|          1 | polo       | toe       | polo@gmail.com    |
|          2 | polu       | eig       | polu@gmail.com    |
|          3 | emily      | john      | emily@gmail.com   |
|          4 | raj        | john      | raj123@gmail.com  |
|          5 | john       | raj       | john@gmail.com    |
|          6 | hari       | raj       | hari@gmail.com    |
|          7 | ramu       | raj       | ramu@gmail.com    |
|          8 | raju       | raja      | raju@gmail.com    |
|          9 | bheem      | raj       | bheem@gmail.com   |
|         10 | sweety     | roy       | sweety@gmail.com  |
+------------+------------+-----------+-------------------+
10 rows in set (0.00 sec)

mysql> SELECT t.teacher_id, t.first_name, t.last_name FROM teachers t
    -> WHERE t.teacher_id NOT IN ( SELECT DISTINCT c.teacher_id FROM courses c);
+------------+------------+-----------+
| teacher_id | first_name | last_name |
+------------+------------+-----------+
|          5 | john       | raj       |
|          6 | hari       | raj       |
|          7 | ramu       | raj       |
|          8 | raju       | raja      |
|          9 | bheem      | raj       |
|         10 | sweety     | roy       |
+------------+------------+-----------+
6 rows in set (0.01 sec)

mysql>
```

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



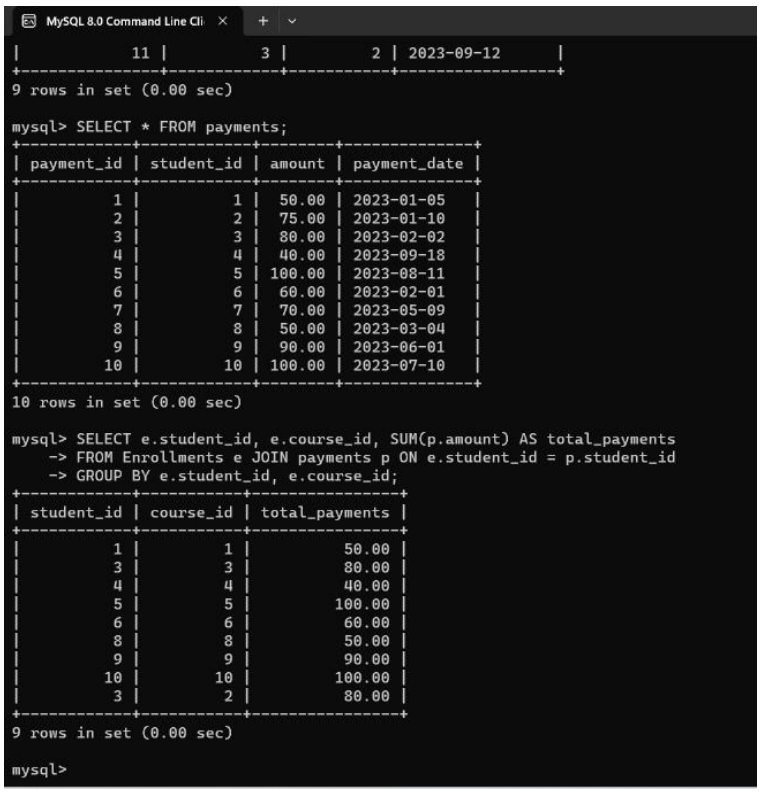
```
MySQL 8.0 Command Line Cli

| teacher_id | first_name | last_name |
+------------+------------+-----------+
|          5 | john       | raj       |
|          6 | hari       | raj       |
|          7 | ramu       | raj       |
|          8 | raju       | raja      |
|          9 | bheem      | raj       |
|         10 | sweety     | roy       |
+------------+------------+-----------+
6 rows in set (0.01 sec)

mysql> SELECT * FROM students;
+------------+------------+-----------+---------------+---------------------+--------------+
| student_id | first_name | last_name | date_of_birth | email               | phone_number |
+------------+------------+-----------+---------------+---------------------+--------------+
|          1 | raj        | monu      | 2002-05-15    | raj@gamil.com       | 0123456789   |
|          2 | monu       | raj       | 2000-06-14    | monu@gmail.com      | 123456789    |
|          3 | nonu       | raj       | 2001-06-04    | nonu@gmail.com      | 1234567687   |
|          4 | abbu       | taj       | 2001-06-02    | abbu@gmail.com      | 2334567687   |
|          5 | polo       | taj       | 2002-06-20    | polo@gmail.com      | 23345676257  |
|          6 | inthu      | raj       | 2000-09-18    | inthu@gmail.com     | 23342587625  |
|          7 | glass      | toe       | 2001-09-18    | glass@gmail.com     | 23342027625  |
|          8 | tree       | toe       | 2005-09-06    | gone@gmail.com      | 23342027602  |
|          9 | bat        | toe       | 2005-09-06    | bat@gmail.com       | 23342027520  |
|         10 | tick       | toe       | 2003-02-09    | tick@gmail.com      | 82542027502  |
|         11 | John       | Doe       | 1995-08-15    | Jhon.Doe@example.com | 1234567890   |
+------------+------------+-----------+---------------+---------------------+--------------+
11 rows in set (0.00 sec)

mysql> SELECT AVG(age) AS average_age FROM (SELECT DATE_FORMAT(NOW(), '%Y') -
    -> DATE_FORMAT(date_of_birth, '%Y') - (DATE_FORMAT(NOW(), '00-%m-%d') <
    -> DATE_FORMAT(date_of_birth,'00-%m-%d')) AS age FROM students)AS subquery;
+--------------------+
| average_age        |
+--------------------+
| 21.636363636363637 |
+--------------------+
1 row in set (0.01 sec)

mysql>
```

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



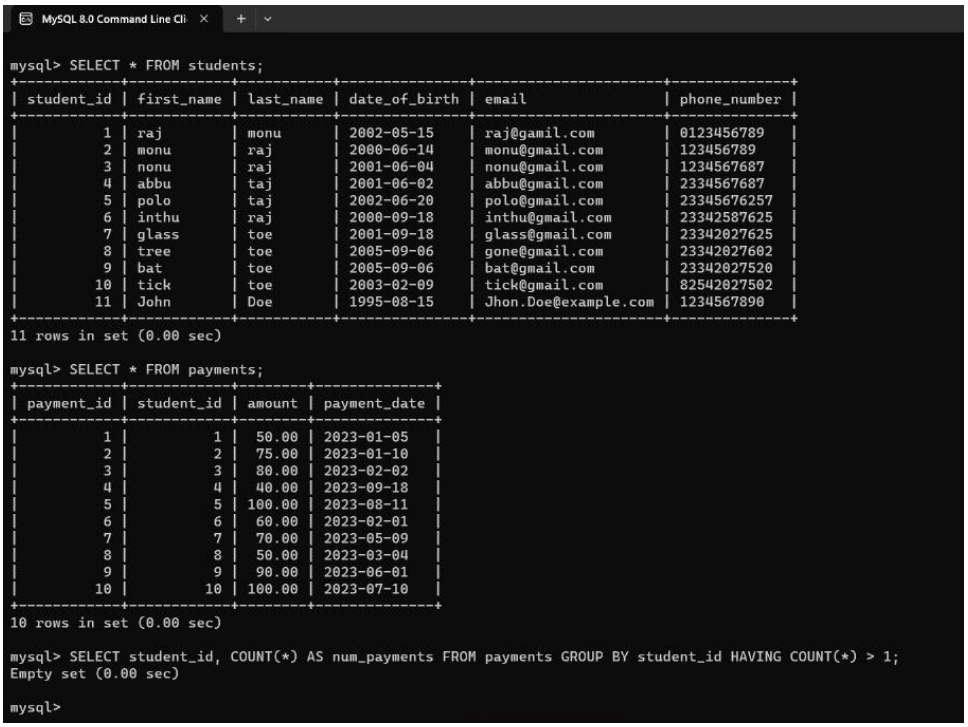
```
MySQL 8.0 Command Line Cli

+-----------+--------------------+---------+------------+
|         1 | Mathematics        |       4 |          1 |
|         2 | Computer           |       3 |          2 |
|         3 | History            |       3 |          3 |
|         4 | Biology            |       4 |          1 |
|         5 | Literature         |       3 |          3 |
|         6 | Chemistry          |       4 |          2 |
|         7 | Physics            |       4 |          3 |
|         8 | Art                |       2 |          4 |
|         9 | Music              |       2 |          4 |
|        10 | Physical education |       2 |          4 |
+-----------+--------------------+---------+------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2023-01-15      |
|             3 |          3 |         3 | 2023-02-02      |
|             4 |          4 |         4 | 2023-09-18      |
|             5 |          5 |         5 | 2023-08-11      |
|             6 |          6 |         6 | 2023-02-01      |
|             8 |          8 |         8 | 2023-03-04      |
|             9 |          9 |         9 | 2023-06-01      |
|            10 |         10 |        10 | 2023-07-10      |
|            11 |          3 |         2 | 2023-09-12      |
+---------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)

mysql> SELECT c.course_id, c.course_name FROM courses c
    -> WHERE c.course_id NOT IN (SELECT DISTINCT e.course_id FROM Enrollments e);
+-----------+-------------+
| course_id | course_name |
+-----------+-------------+
|         7 | Physics     |
+-----------+-------------+
1 row 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.



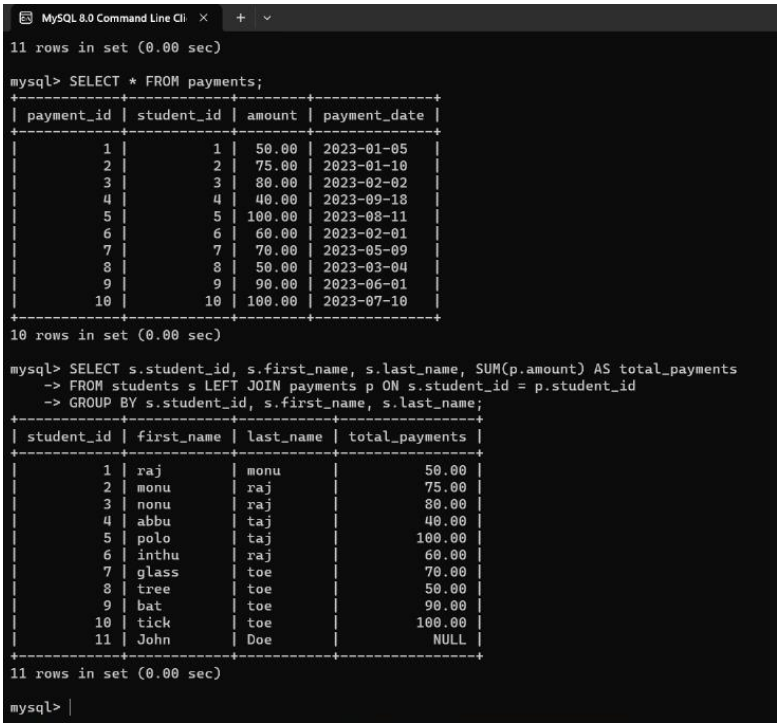
```
MySQL 8.0 Command Line Cli

|           11 |          3 |         2 | 2023-09-12      |
+--------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)

mysql> SELECT * FROM payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          1 |          1 |  50.00 | 2023-01-05   |
|          2 |          2 |  75.00 | 2023-01-10   |
|          3 |          3 |  80.00 | 2023-02-02   |
|          4 |          4 |  40.00 | 2023-09-18   |
|          5 |          5 | 100.00 | 2023-08-11   |
|          6 |          6 |  60.00 | 2023-02-01   |
|          7 |          7 |  70.00 | 2023-05-09   |
|          8 |          8 |  50.00 | 2023-03-04   |
|          9 |          9 |  90.00 | 2023-06-01   |
|         10 |         10 | 100.00 | 2023-07-10   |
+------------+------------+--------+--------------+
10 rows in set (0.00 sec)

mysql> SELECT e.student_id, e.course_id, SUM(p.amount) AS total_payments
    -> FROM Enrollments e JOIN payments p ON e.student_id = p.student_id
    -> GROUP BY e.student_id, e.course_id;
+------------+-----------+----------------+
| student_id | course_id | total_payments |
+------------+-----------+----------------+
|          1 |         1 |          50.00 |
|          3 |         3 |          80.00 |
|          4 |         4 |          40.00 |
|          5 |         5 |         100.00 |
|          6 |         6 |          60.00 |
|          8 |         8 |          50.00 |
|          9 |         9 |          90.00 |
|         10 |        10 |         100.00 |
|          3 |         2 |          80.00 |
+------------+-----------+----------------+
9 rows in set (0.00 sec)

mysql>
```

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.



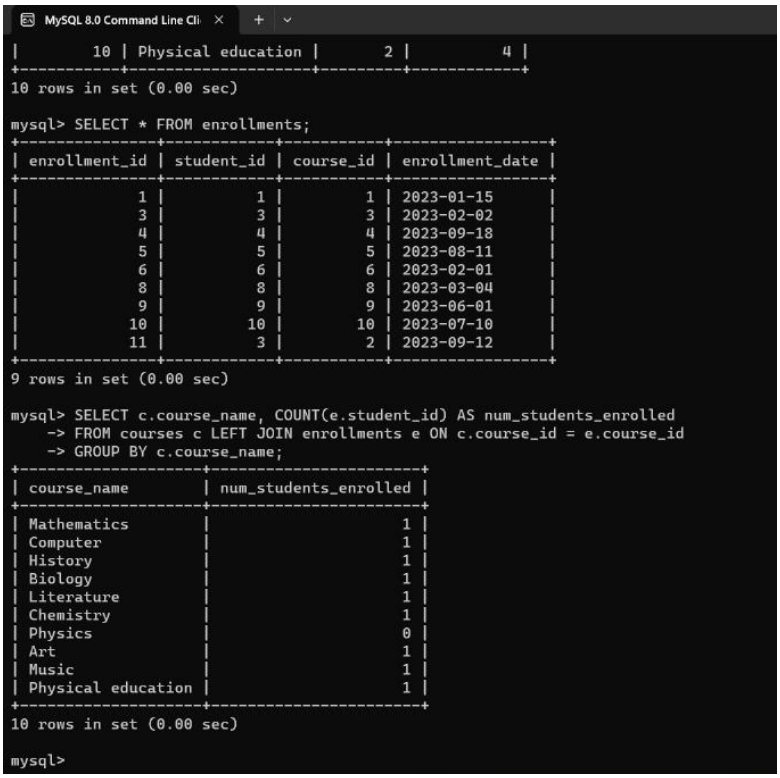
```
MySQL 8.0 Command Line Cli

mysql> SELECT * FROM students;
+------------+------------+-----------+---------------+---------------------+--------------+
| student_id | first_name | last_name | date_of_birth | email               | phone_number |
+------------+------------+-----------+---------------+---------------------+--------------+
|          1 | raj        | monu      | 2002-05-15    | raj@gamil.com       | 0123456789   |
|          2 | monu       | raj       | 2000-06-14    | monu@gmail.com      | 123456789    |
|          3 | nonu       | raj       | 2001-06-04    | nonu@gmail.com      | 1234567687   |
|          4 | abbu       | taj       | 2001-06-02    | abbu@gmail.com      | 2334567687   |
|          5 | polo       | taj       | 2002-06-20    | polo@gmail.com      | 23345676257  |
|          6 | inthu      | raj       | 2000-09-18    | inthu@gmail.com     | 23342587625  |
|          7 | glass      | toe       | 2001-09-18    | glass@gmail.com     | 23342027625  |
|          8 | tree       | toe       | 2005-09-06    | gone@gmail.com      | 23342027602  |
|          9 | bat        | toe       | 2005-09-06    | bat@gmail.com       | 23342027520  |
|         10 | tick       | toe       | 2003-02-09    | tick@gmail.com      | 82542027502  |
|         11 | John       | Doe       | 1995-08-15    | Jhon.Doe@example.com | 1234567890   |
+------------+------------+-----------+---------------+---------------------+--------------+
11 rows in set (0.00 sec)

mysql> SELECT * FROM payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          1 |          1 |  50.00 | 2023-01-05   |
|          2 |          2 |  75.00 | 2023-01-10   |
|          3 |          3 |  80.00 | 2023-02-02   |
|          4 |          4 |  40.00 | 2023-09-18   |
|          5 |          5 | 100.00 | 2023-08-11   |
|          6 |          6 |  60.00 | 2023-02-01   |
|          7 |          7 |  70.00 | 2023-05-09   |
|          8 |          8 |  50.00 | 2023-03-04   |
|          9 |          9 |  90.00 | 2023-06-01   |
|         10 |         10 | 100.00 | 2023-07-10   |
+------------+------------+--------+--------------+
10 rows in set (0.00 sec)

mysql> SELECT student_id, COUNT(*) AS num_payments FROM payments GROUP BY student_id HAVING COUNT(*) > 1;
Empty set (0.00 sec)

mysql>
```

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.



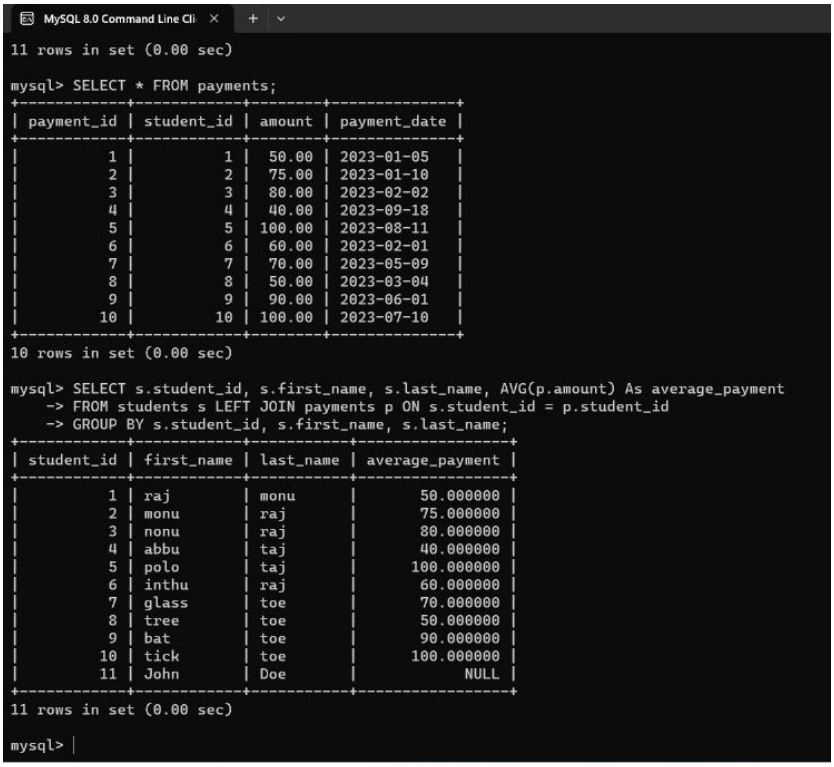
```
MySQL 8.0 Command Line Cli

11 rows in set (0.00 sec)

mysql> SELECT * FROM payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          1 |          1 |  50.00 | 2023-01-05   |
|          2 |          2 |  75.00 | 2023-01-10   |
|          3 |          3 |  80.00 | 2023-02-02   |
|          4 |          4 |  40.00 | 2023-09-18   |
|          5 |          5 | 100.00 | 2023-08-11   |
|          6 |          6 |  60.00 | 2023-02-01   |
|          7 |          7 |  70.00 | 2023-05-09   |
|          8 |          8 |  50.00 | 2023-03-04   |
|          9 |          9 |  90.00 | 2023-06-01   |
|         10 |         10 | 100.00 | 2023-07-10   |
+------------+------------+--------+--------------+
10 rows in set (0.00 sec)

mysql> SELECT s.student_id, s.first_name, s.last_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, s.first_name, s.last_name;
+------------+------------+-----------+----------------+
| student_id | first_name | last_name | total_payments |
+------------+------------+-----------+----------------+
|          1 | raj        | monu      |          50.00 |
|          2 | monu       | raj       |          75.00 |
|          3 | nonu       | raj       |          80.00 |
|          4 | abbu       | taj       |          40.00 |
|          5 | polo       | taj       |         100.00 |
|          6 | inthu      | raj       |          60.00 |
|          7 | glass      | toe       |          70.00 |
|          8 | tree       | toe       |          50.00 |
|          9 | bat        | toe       |          90.00 |
|         10 | tick       | toe       |         100.00 |
|         11 | John       | Doe       |           NULL |
+------------+------------+-----------+----------------+
11 rows in set (0.00 sec)

mysql>
```

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 8.0 Command Line Cli

|         10 | Physical education |         2 |           4 |
+------------+--------------------+-----------+-------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM enrollments;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2023-01-15      |
|             3 |          3 |         3 | 2023-02-02      |
|             4 |          4 |         4 | 2023-09-18      |
|             5 |          5 |         5 | 2023-08-11      |
|             6 |          6 |         6 | 2023-02-01      |
|             8 |          8 |         8 | 2023-03-04      |
|             9 |          9 |         9 | 2023-06-01      |
|            10 |         10 |        10 | 2023-07-10      |
|            11 |          3 |         2 | 2023-09-12      |
+---------------+------------+-----------+-----------------+
9 rows in set (0.00 sec)

mysql> SELECT c.course_name, COUNT(e.student_id) AS num_students_enrolled
    -> FROM courses c LEFT JOIN enrollments e ON c.course_id = e.course_id
    -> GROUP BY c.course_name;
+--------------------+------------------------+
| course_name        | num_students_enrolled  |
+--------------------+------------------------+
| Mathematics        |                      1 |
| Computer           |                      1 |
| History            |                      1 |
| Biology            |                      1 |
| Literature         |                      1 |
| Chemistry          |                      1 |
| Physics            |                      0 |
| Art                |                      1 |
| Music              |                      1 |
| Physical education |                      1 |
+--------------------+------------------------+
10 rows in set (0.00 sec)

mysql>
```

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 8.0 Command Line Cli
11 rows in set (0.00 sec)

mysql> SELECT * FROM payments;
+------------+------------+--------+--------------+
| payment_id | student_id | amount | payment_date |
+------------+------------+--------+--------------+
|          1 |          1 |  50.00 | 2023-01-05   |
|          2 |          2 |  75.00 | 2023-01-10   |
|          3 |          3 |  80.00 | 2023-02-02   |
|          4 |          4 |  40.00 | 2023-09-18   |
|          5 |          5 | 100.00 | 2023-08-11   |
|          6 |          6 |  60.00 | 2023-02-01   |
|          7 |          7 |  70.00 | 2023-05-09   |
|          8 |          8 |  50.00 | 2023-03-04   |
|          9 |          9 |  90.00 | 2023-06-01   |
|         10 |         10 | 100.00 | 2023-07-10   |
+------------+------------+--------+--------------+
10 rows in set (0.00 sec)

mysql> SELECT s.student_id, s.first_name, s.last_name, AVG(p.amount) As average_payment
    -> 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 | average_payment |
+------------+------------+-----------+-----------------+
|          1 | raj        | monu      |       50.000000 |
|          2 | monu       | raj       |       75.000000 |
|          3 | nonu       | raj       |       80.000000 |
|          4 | abbu       | taj       |       40.000000 |
|          5 | polo       | taj       |      100.000000 |
|          6 | inthu      | raj       |       60.000000 |
|          7 | glass      | toe       |       70.000000 |
|          8 | tree       | toe       |       50.000000 |
|          9 | bat        | toe       |       90.000000 |
|         10 | tick       | toe       |      100.000000 |
|         11 | John       | Doe       |            NULL |
+------------+------------+-----------+-----------------+
11 rows in set (0.00 sec)

mysql>
```