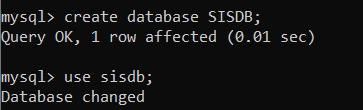
**ASSIGNMENT-2**

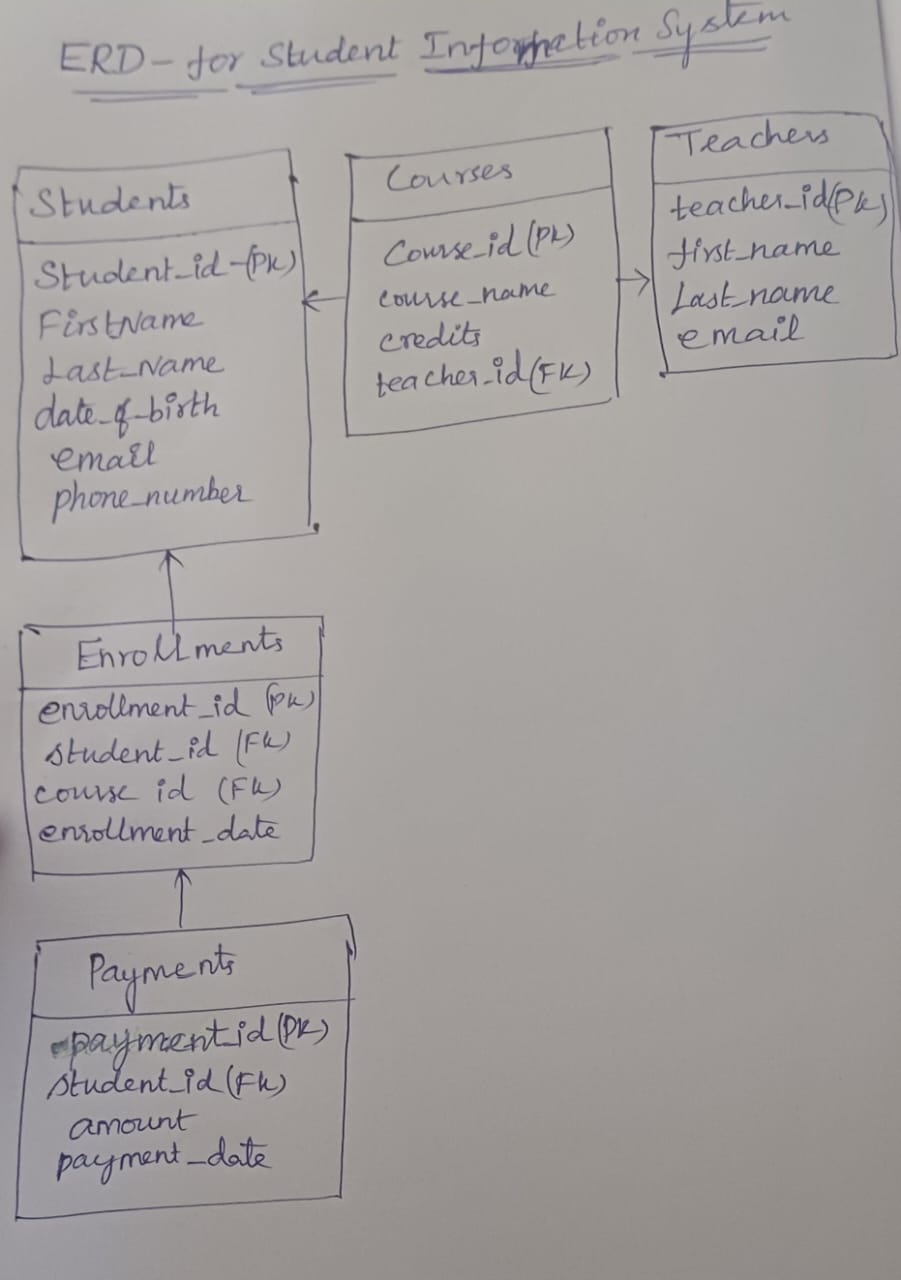
**STUDENT INFORMATION SYSTEM**

**TASK-1**

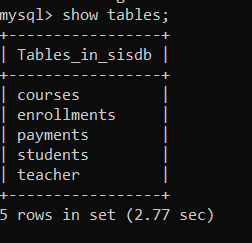
**1.Creation of database**



**3.ERD diagram**



**2.Define schema for students, Courses, enrollments, Teacher and payments**

****

**4.Createappropriate primary and foreign key**

Student\_Id int primary key

teacher\_id int primary key

course\_id int primary key

enrollment\_id int primary key

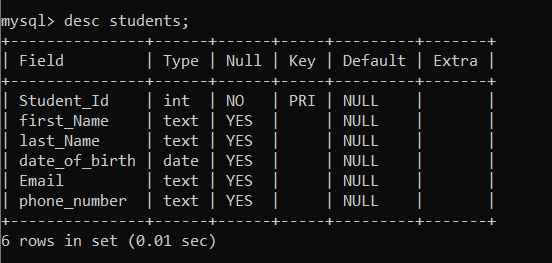
foreign key(student\_id) references Students(student\_id)

foreign key(student\_id) references Students(student\_id),

foreign key (course\_id) references Courses(course\_id));

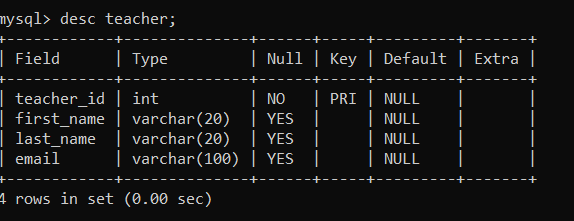
Create student table

create table Students (Student\_Id int primary key,first\_Name text ,last\_Name text,date\_of\_birth date,Email text,phone\_number text);



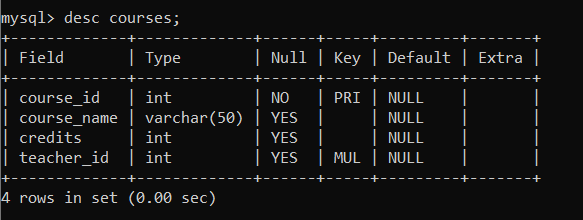
Create the Teacher table

create table Teacher (teacher\_id int primary key, first\_name varchar(20) ,last\_name varchar(20) ,email varchar(100));



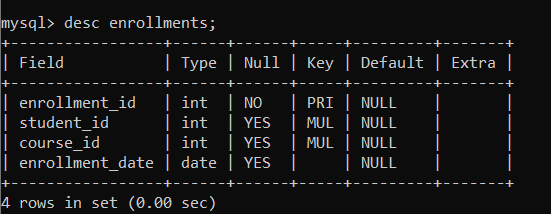
Create the Courses table

create table Courses (course\_id int primary key,course\_name varchar(50),credits int,teacher\_id int,foreign key (teacher\_id) references Teacher(teacher\_id));



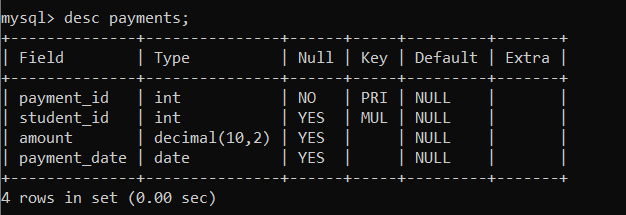
Create the Enrollments table

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));



Create the Payments table

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));



**5.Insert values into the tables**

insert into Students table

insert into Students values

(1, 'Aarav',' Patel','1999-07-02', 'aarav.patel@example.com', '9876543210'),

(2, 'Isha',' Singh','2000-05-09', 'isha.singh@example.com', '9786543210'),

(3, 'Aryan ','Reddy','2001-09-08', 'aryan.reddy@example.com', '9087654321'),

(4, 'Ananya',' Kumar','2000-02-07', 'ananya.kumar@example.com', '8907654321'),

(5, 'Sneha', ' Sharma','2000-11-04', 'sneha.sharma@example.com', '9776543210'),

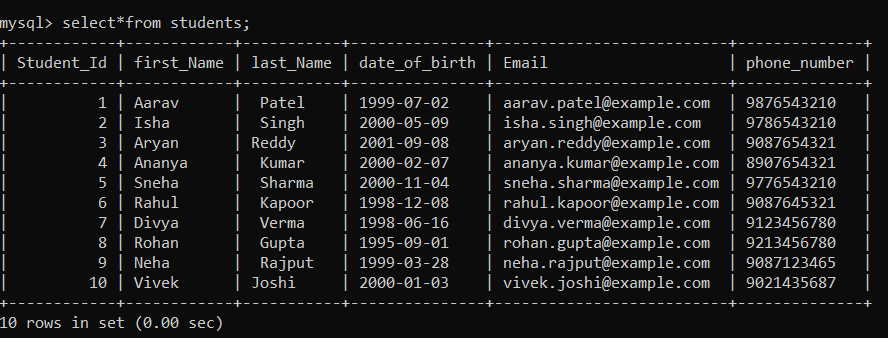
(6, 'Rahul',' Kapoor','1998-12-08','rahul.kapoor@example.com', '9087645321'),

(7, 'Divya',' Verma','1998-06-16', 'divya.verma@example.com', '9123456780'),

(8, 'Rohan',' Gupta','1995-09-01', 'rohan.gupta@example.com', '9213456780'),

(9, 'Neha',' Rajput','1999-03-28','neha.rajput@example.com', '9087123465'),

(10, 'Vivek ','Joshi','2000-01-03', 'vivek.joshi@example.com', '9021435687');



insert into Teacher values

(1, 'Anjali', 'Mishra', 'anjali.mishra@example.com'),

(2, 'Vikram', 'Singh', 'vikram.singh@example.com'),

(3, 'Sneha', 'Gupta', 'sneha.gupta@example.com'),

(4, 'Amit', 'Sharma', 'amit.sharma@example.com'),

(5, 'Pooja', 'Verma', 'pooja.verma@example.com'),

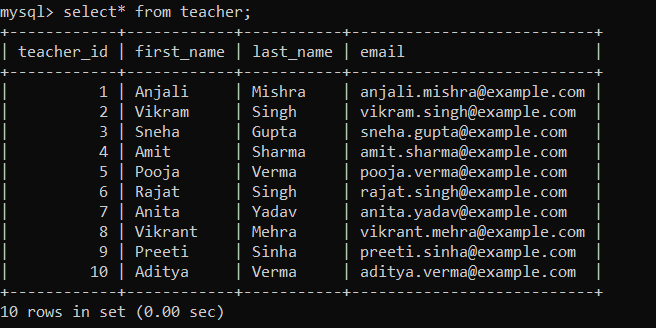
(6, 'Rajat', 'Singh', 'rajat.singh@example.com'),

(7, 'Anita', 'Yadav', 'anita.yadav@example.com'),

(8, 'Vikrant', 'Mehra', 'vikrant.mehra@example.com'),

(9, 'Preeti', 'Sinha', 'preeti.sinha@example.com'),

(10, 'Aditya', 'Verma', 'aditya.verma@example.com');



insert into Courses values

(101, 'Computer Science', 4, 1),

(102, 'Mathematics', 3, 2),

(103, 'Physics', 3, 3),

(104, 'Chemistry', 4, 4),

(105, 'Biology', 3, 5),

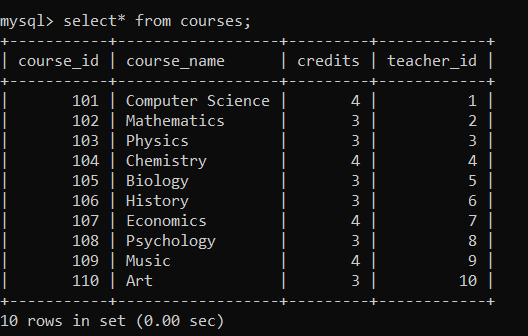
(106, 'History', 3, 6),

(107, 'Economics', 4, 7),

(108, 'Psychology', 3, 8),

(109, 'Music', 4, 9),

(110, 'Art', 3, 10);



insert into Enrollments values

(201, 1, 101, '2023-01-15'),

(202, 2, 102, '2023-01-20'),

(203, 3, 103, '2023-01-25'),

(204, 4, 104, '2023-03-15'),

(205, 5, 105, '2023-03-20'),

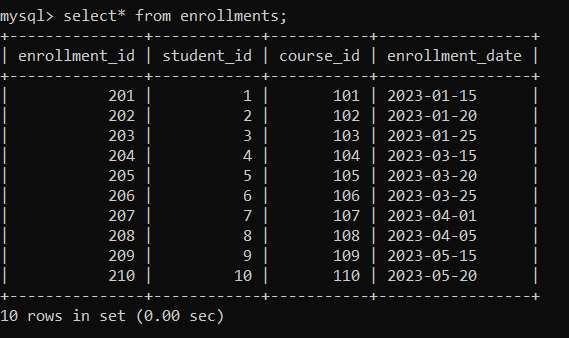
(206, 6, 106, '2023-03-25'),

(207, 7, 107, '2023-04-01'),

(208, 8, 108, '2023-04-05'),

(209, 9, 109, '2023-05-15'),

(210, 10, 110, '2023-05-20');



insert into Payments values

(301, 1, 1000.00, '2023-02-01'),

(302, 2, 1200.00, '2023-02-05'),

(303, 3, 800.00, '2023-02-10'),

(304, 4, 1200.00, '2023-04-10'),

(305, 5, 1000.00, '2023-04-15'),

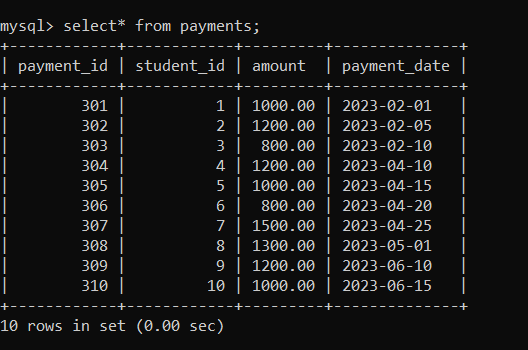
(306, 6, 800.00, '2023-04-20'),

(307, 7, 1500.00, '2023-04-25'),

(308, 8, 1300.00, '2023-05-01'),

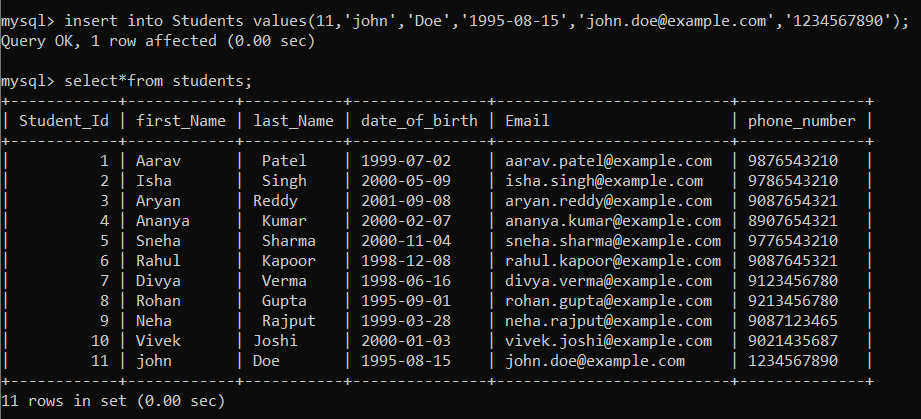
(309, 9, 1200.00, '2023-06-10'),

(310, 10, 1000.00, '2023-06-15');

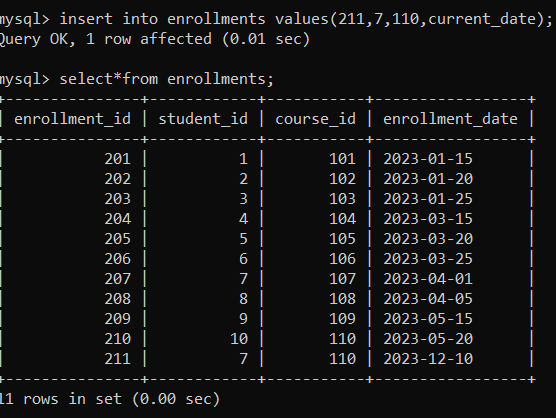


**TASK-2**

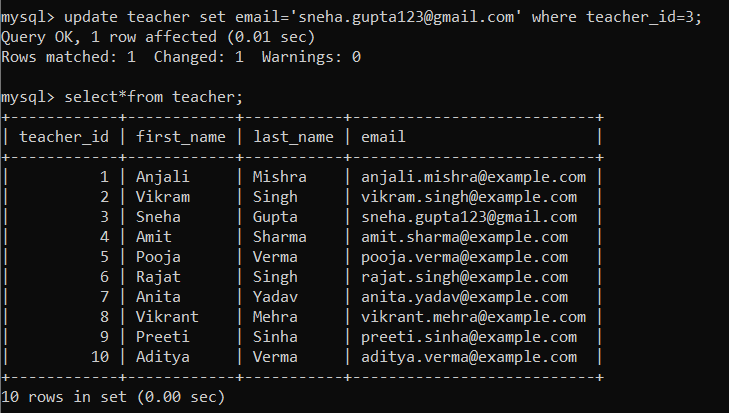
1.Insert a new student



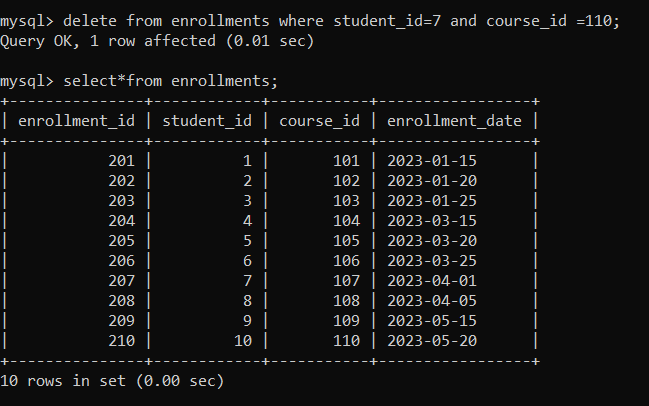
2.Enroll a student in a course



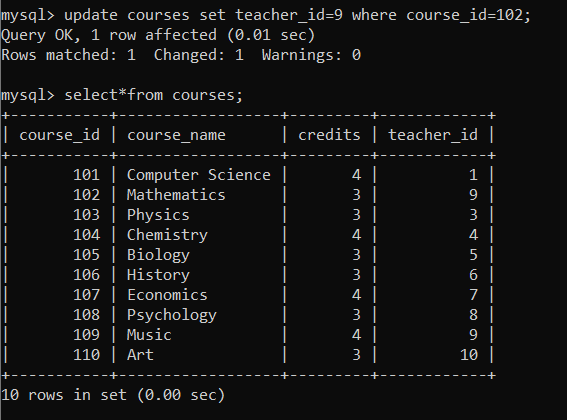
3.update email id of a specific teacher



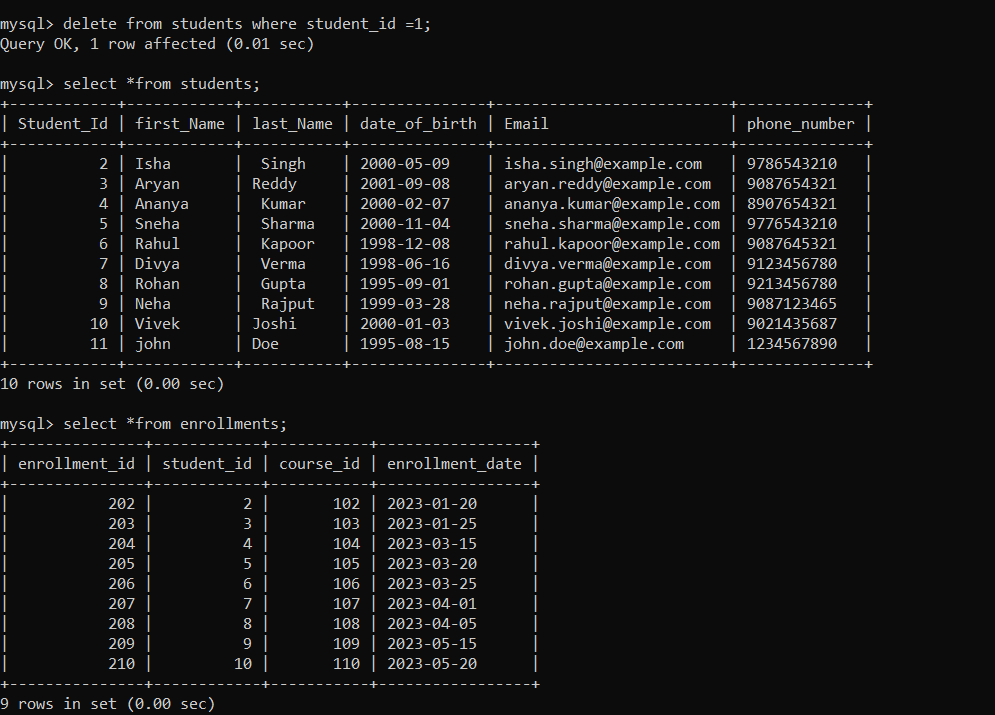
4.delete a specific enrollment record



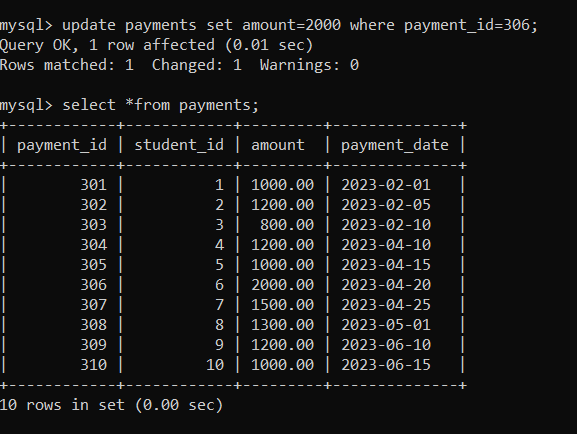
5.update course table to assign a specific teacher



6. Delete a specific student from the "Students" table and remove all their enrollment records

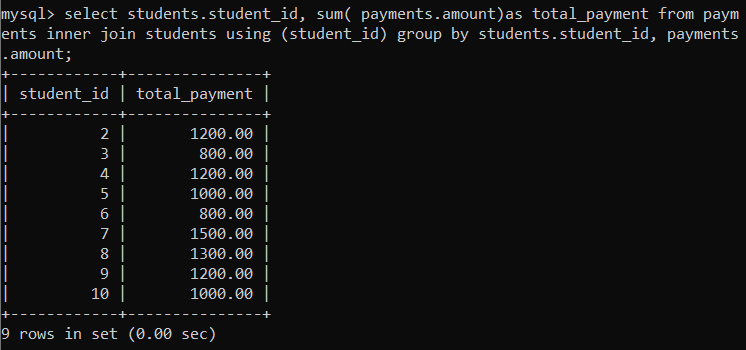


7.update the payment amount in payment table

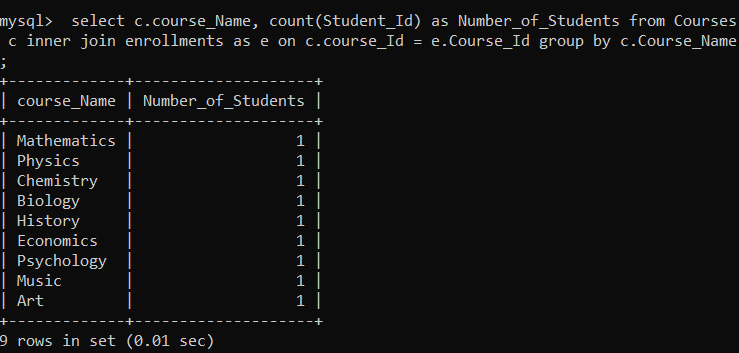


TASK 3

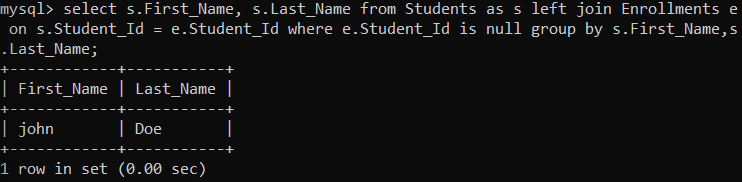
1. Calculate the total payments made by a specific student. You will need to join the "Payments" table with the "Students" table based on the student's ID.



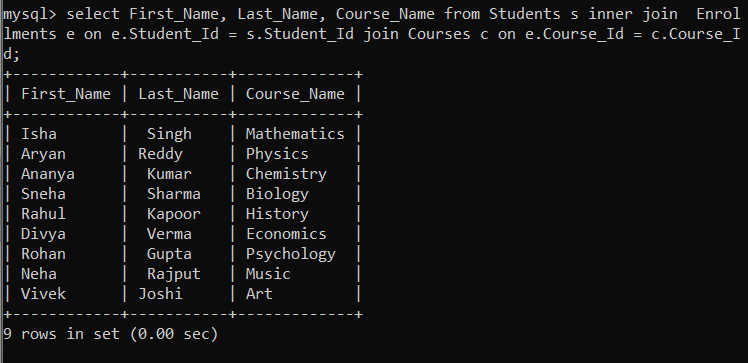
2. retrieve a list of courses along with the count of students enrolled in each course



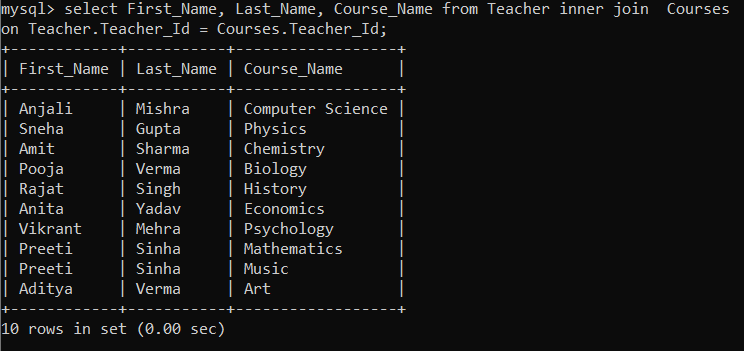
3. find the names of students who have not enrolled in any course.



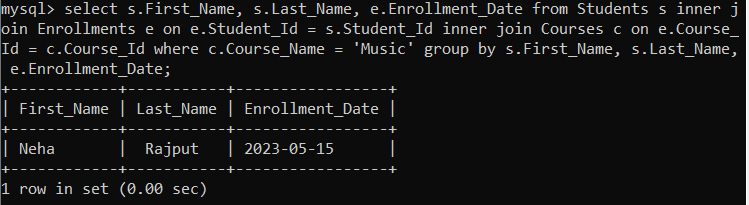
4. retrieve the first name, last name of students, and the names of the courses they are enrolled



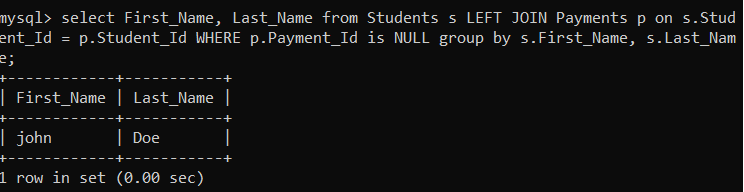
5. list the names of teachers and the courses they are assigned



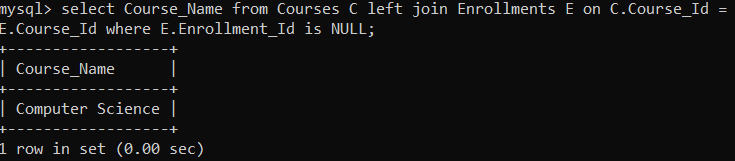
6. list of students and their enrollment dates for a specific course



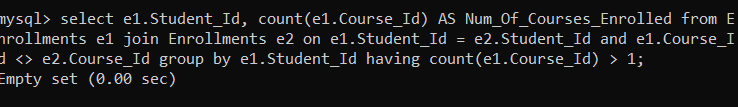
7. the names of students who have not made any payments



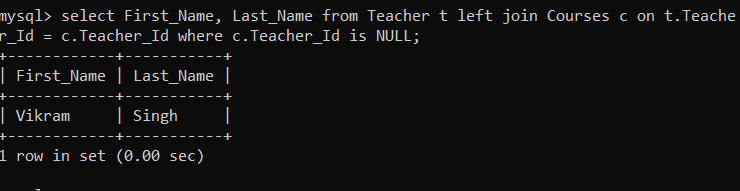
8. identify courses that have no enrollments.



9. students who are enrolled in more than one course

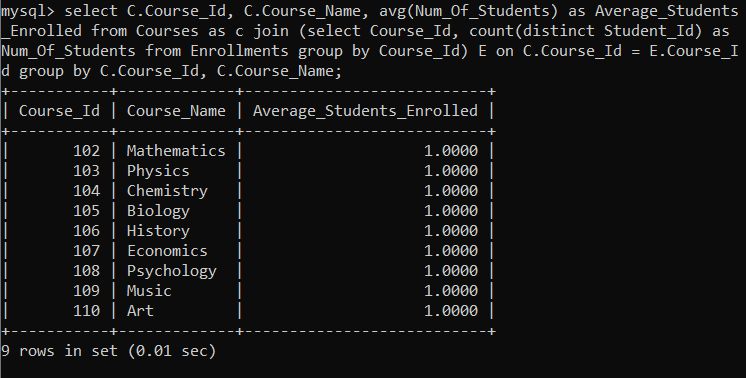


10. teachers who are not assigned to any courses. Use a LEFT JOIN between the "Teacher" table and the "Courses" table and filter for teachers with NULL course assignments

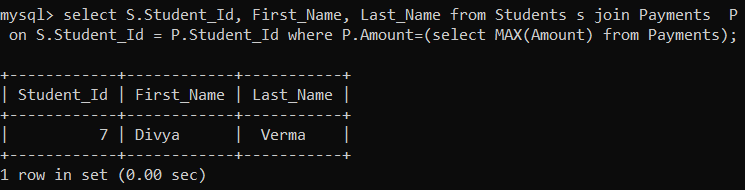


TASK 4

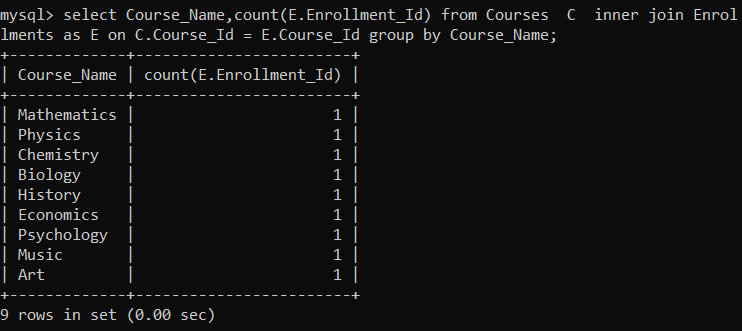
1. calculate the average number of students enrolled in each course.



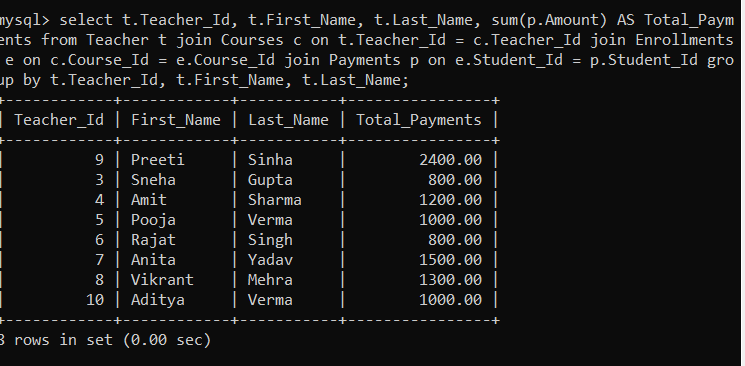
2. the student(s) who made the highest payment.



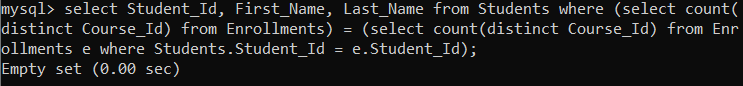
3. list of courses with the highest number of enrollments.



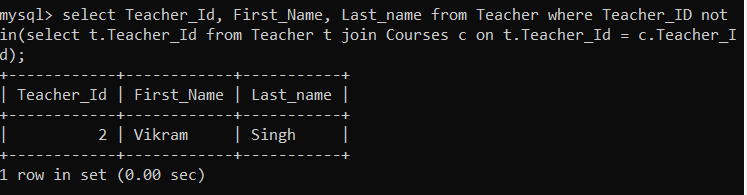
4. the total payments made to courses taught by each teacher



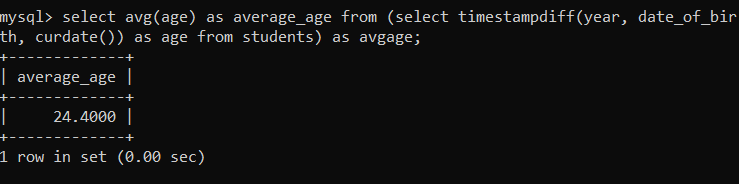
5. students who are enrolled in all available courses



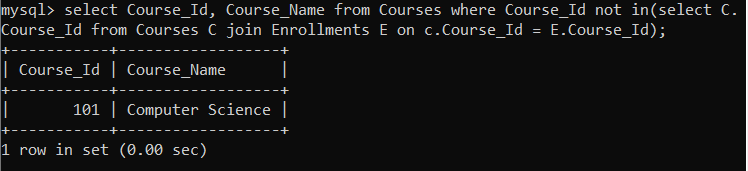
6. the names of teachers who have not been assigned to any courses



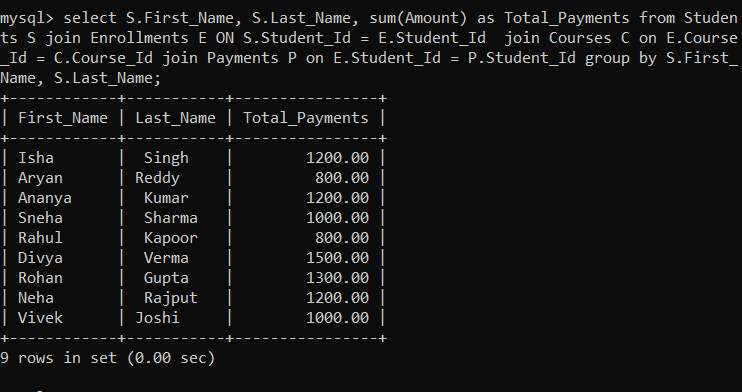
7. the average age of all students



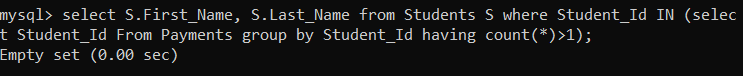
8. Identify courses with no enrollments. Use subqueries to find courses without enrollment records



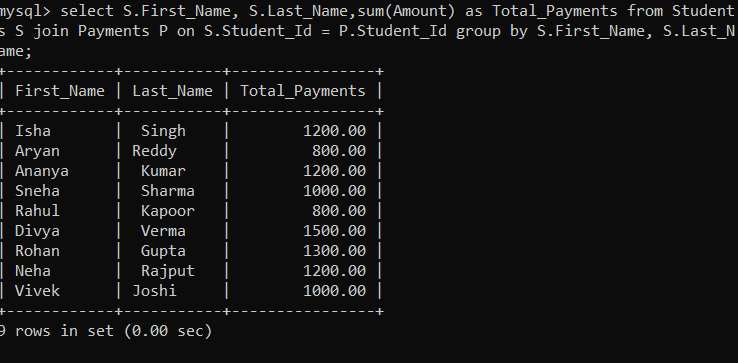
9. the total payments made by each student for each course they are enrolled



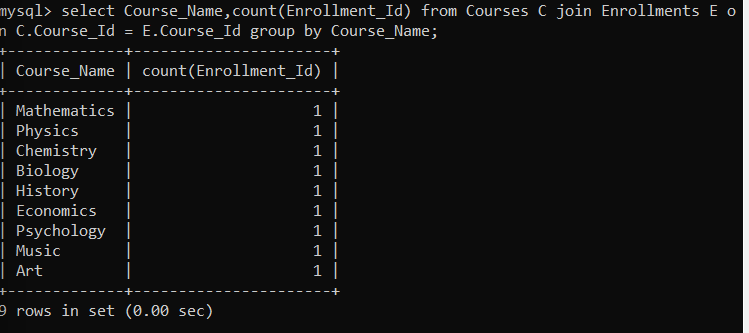
10. students who have made more than one payment.



11. SQL query to calculate the total payments made by each student



12. a list of course names along with the count of students enrolled in each course



13. the average payment amount made by students.

