Assignment : Student Information System

Task 1. Database Design:

1. Create the database named "SISDB"

A black background with white text

Description automatically generated

2. Define the schema for the Students, Courses, Enrollments, Teacher, and Payments tables based

on the provided schema. Write SQL scripts to create the mentioned tables with appropriate data

types, constraints, and relationships.

a. Students

b. Courses

c. Enrollments

d. Teacher

e. Payments



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

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

The above scripts already include PRIMARY KEY and FOREIGN KEY constraints. For example:

* **Enrollments** references **Students** and **Courses**
* **Courses** references **Teacher**
* **Payments** references **Students**

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

i. Students

ii. Courses

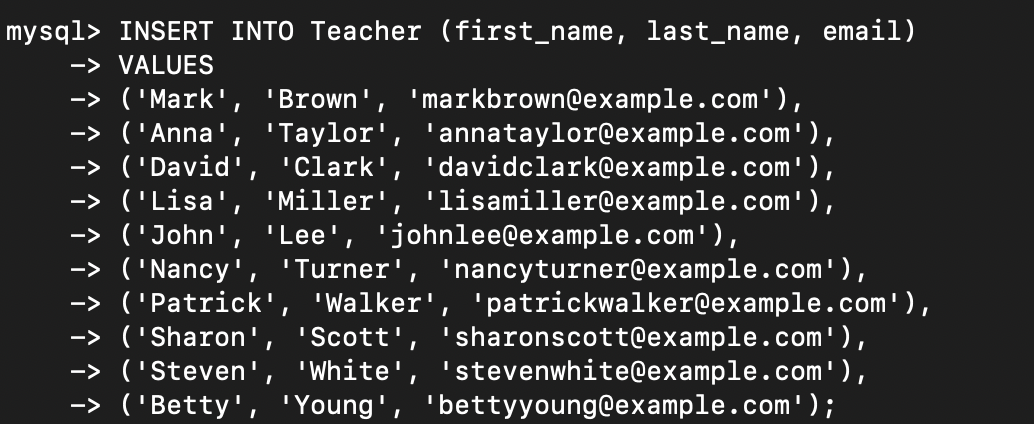
iii. Enrollments

iv. Teacher

v. Payments

A screen shot of a computer

Description automatically generated



A computer screen shot of a black screen

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

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

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

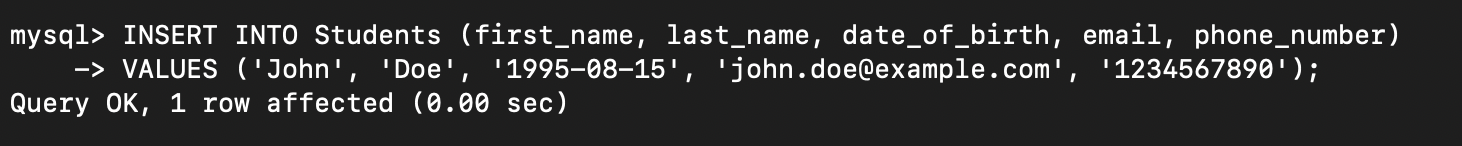
a. First Name: John

b. Last Name: Doe

c. Date of Birth: 1995-08-15

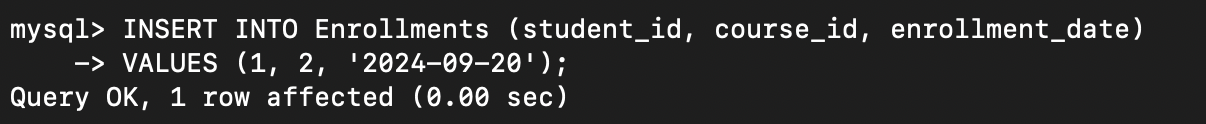
d. Email: john.doe@example.com

e. Phone Number: 1234567890



2. Write an SQL query to enroll a student in a course. Choose an existing student and course and

insert a record into the "Enrollments" table with the enrollment date.



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

modify their email address.

A black screen with white text

Description automatically generated

4. Write an SQL query to delete a specific enrollment record from the "Enrollments" table. Select

an enrollment record based on the student and course.

A black background with white text

Description automatically generated

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

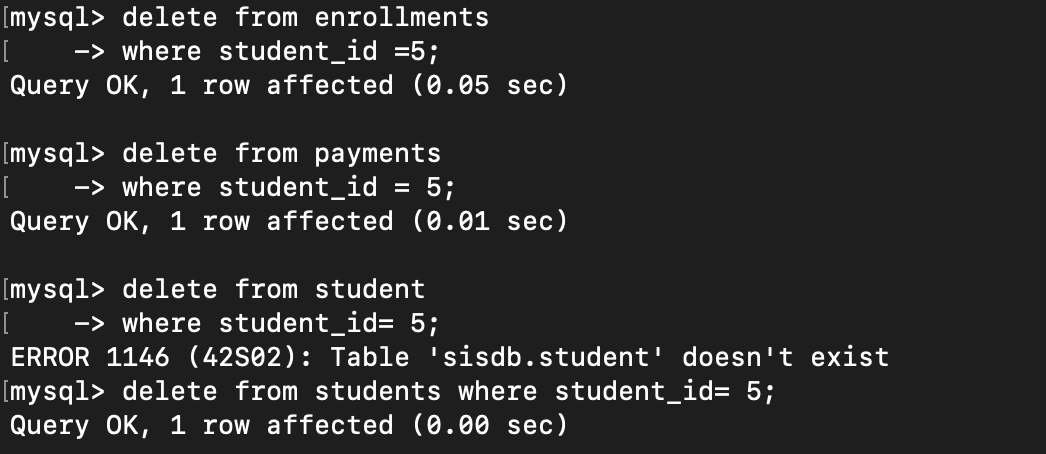
teacher from the respective tables.

A black background with white text

Description automatically generated

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.



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

payment record and modify the payment amount.

A black background with white text

Description automatically generated

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

1. Write an SQL query to calculate the total payments made by a specific student. You will need to

join the "Payments" table with the "Students" table based on the student's ID.

A computer screen with white text

Description automatically generated

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.

A computer screen with white text

Description automatically generated

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.

A screen shot of a computer

Description automatically generated

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.

A black screen with white text

Description automatically generated

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.

A screen shot of a computer

Description automatically generated

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.

A black screen with white text

Description automatically generated

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.

A computer screen shot of a black screen

Description automatically generated

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.

A screen shot of a computer

Description automatically generated

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.

A computer screen with white text

Description automatically generated

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.

A computer screen with white text

Description automatically generated