

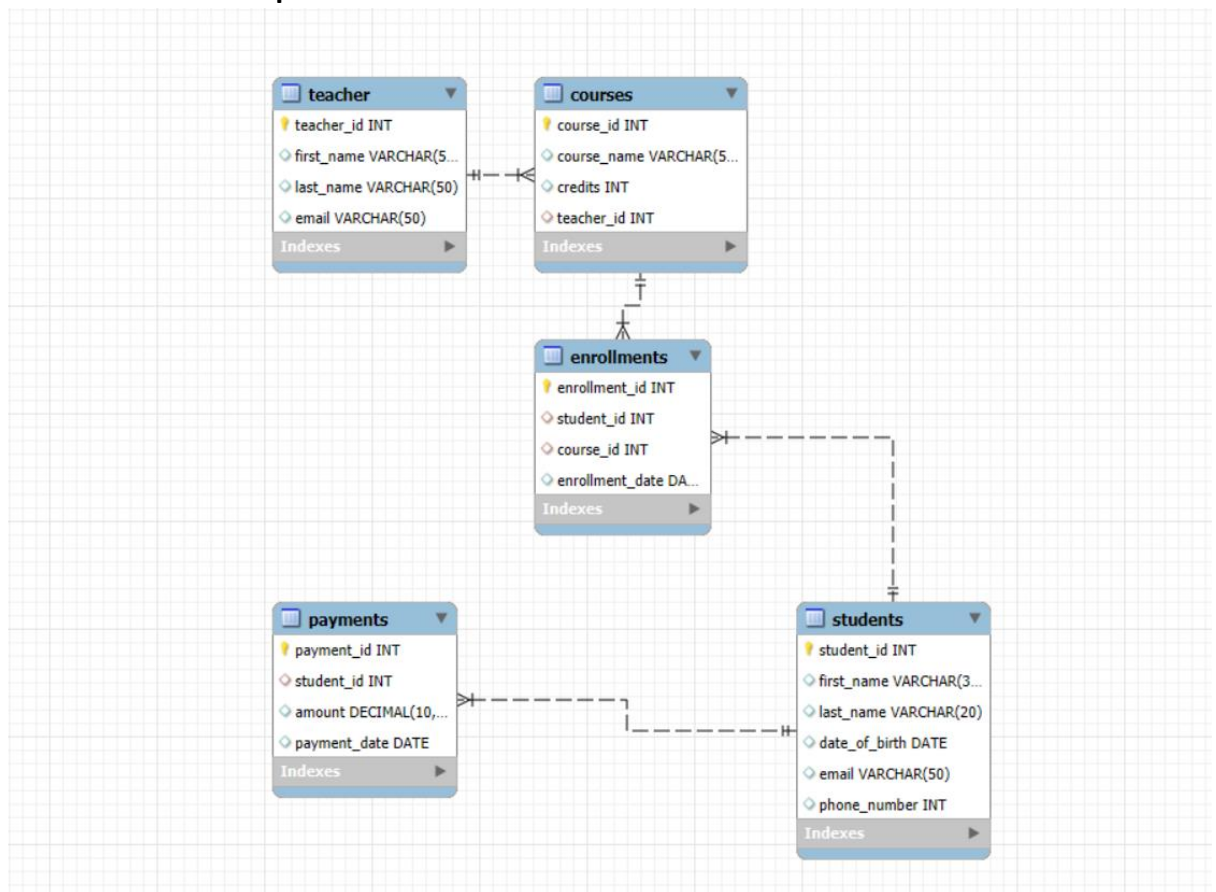
STUDENT INFORMATION SYSTEM

Task 1. Database Design:

1. Create the database named "SISDB":

```
mysql> create database Student_Information_System;  
Query OK, 1 row affected (0.06 sec)
```

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



3. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.

Students

```
mysql> Create table Students(student_id int primary key auto_increment , first_name varchar(30),last_name varchar(20),date_of_birth date,email varchar(50),phone_number int);
Query OK, 0 rows affected (0.09 sec)
```

Courses

```
mysql> create table courses(course_id int primary key auto_increment,course_name varchar(50),credits int,teacher_id int
, foreign key(teacher_id) references teacher(teacher_id) on delete cascade on update cascade);
Query OK, 0 rows affected (0.10 sec)
```

Enrollments

```
mysql> create table enrollments(
-> enrollment_id int primary key auto_increment,
-> student_id int,
-> course_id int,
-> enrollment_date date,
-> foreign key(student_id) references students(student_id)
-> on delete cascade
-> on update cascade
-> , foreign key(course_id) references courses(course_id)
-> on delete cascade
-> on update cascade
-> );
Query OK, 0 rows affected (0.07 sec)
```

Teacher

```
mysql> create table Teacher(teacher_id int primary key auto_increment, first_name varchar(50),last_name varchar(50),email varchar(50));
Query OK, 0 rows affected (0.08 sec)
```

Payments

```
mysql> create table payments(
-> payment_id int primary key auto_increment,
-> student_id int,
-> amount decimal(10,2),
-> payment_date date,
-> foreign key(student_id) references students(student_id)
-> on delete cascade
-> on update cascade);
Query OK, 0 rows affected (0.11 sec)
```

4. Create an ERD (Entity Relationship Diagram) for the database.
5. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

FOREIGN KEY (student_id) REFERENCES Students(student_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (course_id) REFERENCES Courses(course_id) ON DELETE CASCADE ON UPDATE CASCADE

FOREIGN KEY (teacher_id) REFERENCES Teachers(teacher_id) ON DELETE SET NULL

FOREIGN KEY (teacher_id) REFERENCES Teachers(teacher_id) ON DELETE SET NULL

payment_id INT PRIMARY KEY AUTO_INCREMENT,
teacher_id INT PRIMARY KEY AUTO_INCREMENT,
student_id INT PRIMARY KEY AUTO_INCREMENT,

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

Students

```
mysql> INSERT INTO Students (student_id, first_name, last_name, date_of_birth, email, phone_number) VALUES
-> (1, 'Alice', 'Johnson', '2001-05-14', 'alice.johnson@example.com', '1234567890'),
-> (2, 'Bob', 'Smith', '2000-08-22', 'bob.smith@example.com', '1234567891'),
-> (3, 'Charlie', 'Brown', '1999-12-10', 'charlie.brown@example.com', '1234567892'),
-> (4, 'David', 'Wilson', '2002-04-18', 'david.wilson@example.com', '1234567893'),
-> (5, 'Emma', 'Davis', '2001-07-30', 'emma.davis@example.com', '1234567894'),
-> (6, 'Frank', 'Miller', '2000-11-15', 'frank.miller@example.com', '1234567895'),
-> (7, 'Grace', 'Taylor', '1999-09-25', 'grace.taylor@example.com', '1234567896'),
-> (8, 'Hannah', 'Anderson', '2002-06-05', 'hannah.anderson@example.com', '1234567897'),
-> (9, 'Ian', 'Thomas', '2001-03-12', 'ian.thomas@example.com', '1234567898'),
-> (10, 'Jack', 'White', '2000-10-20', 'jack.white@example.com', '1234567899');
Query OK, 10 rows affected (0.02 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
mysql> select * from students;
+-----+-----+-----+-----+-----+-----+
| student_id | first_name | last_name | date_of_birth | email | phone_number |
+-----+-----+-----+-----+-----+-----+
| 1 | Alice | Johnson | 2001-05-14 | alice.johnson@example.com | 1234567890 |
| 2 | Bob | Smith | 2000-08-22 | bob.smith@example.com | 1234567891 |
| 3 | Charlie | Brown | 1999-12-10 | charlie.brown@example.com | 1234567892 |
| 4 | David | Wilson | 2002-04-18 | david.wilson@example.com | 1234567893 |
| 5 | Emma | Davis | 2001-07-30 | emma.davis@example.com | 1234567894 |
| 6 | Frank | Miller | 2000-11-15 | frank.miller@example.com | 1234567895 |
| 7 | Grace | Taylor | 1999-09-25 | grace.taylor@example.com | 1234567896 |
| 8 | Hannah | Anderson | 2002-06-05 | hannah.anderson@example.com | 1234567897 |
| 9 | Ian | Thomas | 2001-03-12 | ian.thomas@example.com | 1234567898 |
| 10 | Jack | White | 2000-10-20 | jack.white@example.com | 1234567899 |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.04 sec)
```

Courses

```
mysql> INSERT INTO Courses (course_id, course_name, credits, teacher_id) VALUES
-> (1, 'Mathematics', 3, 1),
-> (2, 'Physics', 4, 2),
-> (3, 'Chemistry', 3, 3),
-> (4, 'Biology', 4, 4),
-> (5, 'Computer Science', 5, 5),
-> (6, 'History', 3, 6),
-> (7, 'Economics', 3, 7),
-> (8, 'Philosophy', 2, 8),
-> (9, 'Political Science', 3, 9),
-> (10, 'Psychology', 4, 10);
Query OK, 10 rows affected (0.05 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

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

course_id	course_name	credits	teacher_id
1	Mathematics	3	1
2	Physics	4	2
3	Chemistry	3	3
4	Biology	4	4
5	Computer Science	5	5
6	History	3	6
7	Economics	3	7
8	Philosophy	2	8
9	Political Science	3	9
10	Psychology	4	10

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

Enrollments

```
mysql> INSERT INTO Courses (course_id, course_name, credits, teacher_id) VALUES
-> (1, 'Mathematics', 3, 1),
-> (2, 'Physics', 4, 2),
-> (3, 'Chemistry', 3, 3),
-> (4, 'Biology', 4, 4),
-> (5, 'Computer Science', 5, 5),
-> (6, 'History', 3, 6),
-> (7, 'Economics', 3, 7),
-> (8, 'Philosophy', 2, 8),
-> (9, 'Political Science', 3, 9),
-> (10, 'Psychology', 4, 10);
Query OK, 10 rows affected (0.05 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

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

enrollment_id	student_id	course_id	enrollment_date
1	1	1	2024-01-15
2	2	2	2024-02-10
3	3	3	2024-03-05
4	4	4	2024-01-20
5	5	5	2024-02-15
6	6	6	2024-03-01
7	7	7	2024-01-25
8	8	8	2024-02-20
9	9	9	2024-03-10
10	10	10	2024-01-30

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

Teacher

```
mysql> INSERT INTO Teacher (teacher_id, first_name, last_name, email) VALUES
-> (1, 'Mark', 'Johnson', 'mark.johnson@example.com'),
-> (2, 'Sarah', 'Williams', 'sarah.williams@example.com'),
-> (3, 'John', 'Brown', 'john.brown@example.com'),
-> (4, 'Emily', 'Davis', 'emily.davis@example.com'),
-> (5, 'Michael', 'Wilson', 'michael.wilson@example.com'),
-> (6, 'Laura', 'Miller', 'laura.miller@example.com'),
-> (7, 'Daniel', 'Taylor', 'daniel.taylor@example.com'),
-> (8, 'Sophia', 'Anderson', 'sophia.anderson@example.com'),
-> (9, 'James', 'Thomas', 'james.thomas@example.com'),
-> (10, 'Olivia', 'White', 'olivia.white@example.com');
Query OK, 10 rows affected (0.04 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
mysql> select * from teacher;
+-----+-----+-----+-----+
| teacher_id | first_name | last_name | email |
+-----+-----+-----+-----+
|          1 | Mark      | Johnson   | mark.johnson@example.com |
|          2 | Sarah     | Williams  | sarah.williams@example.com |
|          3 | John      | Brown     | john.brown@example.com |
|          4 | Emily     | Davis     | emily.davis@example.com |
|          5 | Michael   | Wilson    | michael.wilson@example.com |
|          6 | Laura     | Miller    | laura.miller@example.com |
|          7 | Daniel    | Taylor    | daniel.taylor@example.com |
|          8 | Sophia    | Anderson  | sophia.anderson@example.com |
|          9 | James     | Thomas    | james.thomas@example.com |
|         10 | Olivia    | White     | olivia.white@example.com |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

Payments

```
mysql> INSERT INTO Payments (payment_id, student_id, amount, payment_date) VALUES
-> (1, 1, 500, '2024-02-01'),
-> (2, 2, 700, '2024-02-10'),
-> (3, 3, 600, '2024-02-15'),
-> (4, 4, 800, '2024-03-01'),
-> (5, 5, 550, '2024-03-10'),
-> (6, 6, 750, '2024-03-15'),
-> (7, 7, 900, '2024-04-01'),
-> (8, 8, 650, '2024-04-10'),
-> (9, 9, 720, '2024-04-15'),
-> (10, 10, 580, '2024-05-01');
Query OK, 10 rows affected (0.05 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
mysql> select * from payments;
+-----+-----+-----+-----+
| payment_id | student_id | amount | payment_date |
+-----+-----+-----+-----+
|          1 |          1 | 500.00 | 2024-02-01 |
|          2 |          2 | 700.00 | 2024-02-10 |
|          3 |          3 | 600.00 | 2024-02-15 |
|          4 |          4 | 800.00 | 2024-03-01 |
|          5 |          5 | 550.00 | 2024-03-10 |
|          6 |          6 | 750.00 | 2024-03-15 |
|          7 |          7 | 900.00 | 2024-04-01 |
|          8 |          8 | 650.00 | 2024-04-10 |
|          9 |          9 | 720.00 | 2024-04-15 |
|         10 |         10 | 580.00 | 2024-05-01 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

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

```
mysql> INSERT INTO Students (first_name, last_name, date_of_birth, email, phone_number)
-> VALUES ('John', 'Durairaj', '1995-08-15', 'john.durai@example.com', '1234567890');
Query OK, 1 row affected (0.05 sec)
```

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.

```
mysql> insert into enrollments(student_id, course_id, enrollment_date) values(4,5,'2025-03-19');
Query OK, 1 row affected (0.01 sec)
```

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

```
mysql> update teacher set email="michael.hussey@gmail.com" where teacher_id = 5;
Query OK, 1 row affected (0.05 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

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.

```
mysql> delete from enrollments where student_id=6 and course_id = 6;  
Query OK, 1 row affected (0.05 sec)
```

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

```
mysql> update courses set teacher_id = 6 where course_id=3;  
Query OK, 1 row affected (0.05 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```

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> delete from students where student_id = 3;  
Query OK, 1 row affected (0.02 sec)
```

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

```
mysql> update payments set amount = 1000.00 where payment_id = 6;  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```