**Student Information System (SIS)**

**Task 1. Database Design:**

**1. Create the database named "SISDB"**

mysql> create database sisdb;

Query OK, 1 row affected (0.28 sec)

mysql> show databases;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

| sisdb |

| sys |

| testdb |

+--------------------+

6 rows in set (0.00 sec)

**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. Payment**

mysql> use sisdb;

Database changed

mysql> create table Students(student\_id integer(3) primary key,

-> first\_name varchar(20),

-> last\_name varchar(20),

-> date\_of\_birth date,

-> email varchar(25),

-> phone\_number integer(10)

-> );

Query OK, 0 rows affected, 2 warnings (0.92 sec)

mysql> desc Students;

+---------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+-------------+------+-----+---------+-------+

| student\_id | int | NO | PRI | NULL | |

| first\_name | varchar(20) | YES | | NULL | |

| last\_name | varchar(20) | YES | | NULL | |

| date\_of\_birth | date | YES | | NULL | |

| email | varchar(25) | YES | | NULL | |

| phone\_number | int | YES | | NULL | |

+---------------+-------------+------+-----+---------+-------+

6 rows in set (0.00 sec)

mysql> create table Teacher(teacher\_id int(3) primary key,

-> first\_name varchar(20),

-> last\_name varchar(20),

-> email varchar(25)

-> );

Query OK, 0 rows affected, 1 warning (1.18 sec)

mysql> desc Teacher;

+------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+-------------+------+-----+---------+-------+

| teacher\_id | int | NO | PRI | NULL | |

| first\_name | varchar(20) | YES | | NULL | |

| last\_name | varchar(20) | YES | | NULL | |

| email | varchar(25) | YES | | NULL | |

+------------+-------------+------+-----+---------+-------+

4 rows in set (0.00 sec)

mysql> create table Courses(course\_id int(3) primary key,

-> course\_name varchar(20),

-> credits int(3),

-> teacher\_id int(3) references Teacher(teacher\_id)

-> );

Query OK, 0 rows affected, 3 warnings (0.46 sec)

mysql> desc Courses;

+-------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+-------+

| course\_id | int | NO | PRI | NULL | |

| course\_name | varchar(20) | YES | | NULL | |

| credits | int | YES | | NULL | |

| teacher\_id | int | YES | | NULL | |

+-------------+-------------+------+-----+---------+-------+

4 rows in set (0.00 sec)

mysql> create table Enrollments(enrollment\_id int (4) primary key,

-> student\_id int (3) references Students(student\_id),

-> course\_id int(3) references Courses(course\_id),

-> enrollment\_date date

-> );

Query OK, 0 rows affected, 3 warnings (0.53 sec)

mysql> desc Enrollments;

+-----------------+------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------------+------+------+-----+---------+-------+

| enrollment\_id | int | NO | PRI | NULL | |

| student\_id | int | YES | | NULL | |

| course\_id | int | YES | | NULL | |

| enrollment\_date | date | YES | | NULL | |

+-----------------+------+------+-----+---------+-------+

4 rows in set (0.00 sec)

mysql> create table Payments(payment\_id int(4) primary key,

-> student\_id int(3) references Students(student\_id),

-> amount int(5),

-> payment\_date date

-> );

Query OK, 0 rows affected, 3 warnings (0.56 sec)

mysql> desc Payments;

+--------------+------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------------+------+------+-----+---------+-------+

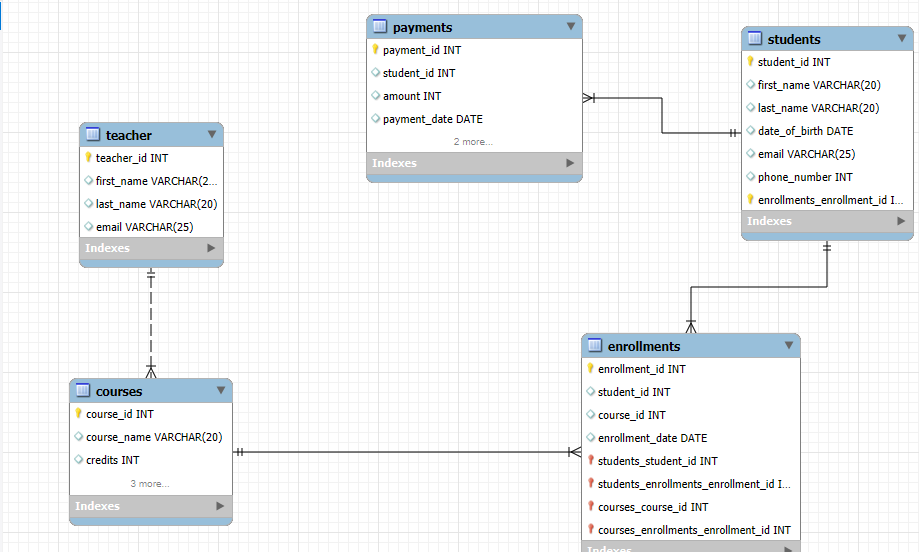
| payment\_id | int | NO | PRI | NULL | |

| student\_id | int | YES | | NULL | |

| amount | int | YES | | NULL | |

| payment\_date | date | YES | | NULL | |

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



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

CREATE TABLE Students (

-> student\_id INT PRIMARY KEY,

-> first\_name VARCHAR(50) NOT NULL,

-> last\_name VARCHAR(50) NOT NULL,

-> date\_of\_birth DATE NOT NULL,

-> email VARCHAR(100) UNIQUE NOT NULL,

-> phone\_number BIGINT UNIQUE NOT NULL

-> );

CREATE TABLE Teacher (

-> teacher\_id INT PRIMARY KEY,

-> first\_name VARCHAR(50) NOT NULL,

-> last\_name VARCHAR(50) NOT NULL,

-> email VARCHAR(100) UNIQUE NOT NULL

-> );

CREATE TABLE Courses (

-> course\_id INT PRIMARY KEY,

-> course\_name VARCHAR(100) NOT NULL,

-> credits INT NOT NULL CHECK (credits > 0),

-> teacher\_id INT,

-> FOREIGN KEY (teacher\_id) REFERENCES Teacher(teacher\_id) ON DELETE SET NULL

-> );

CREATE TABLE Enrollments (

-> enrollment\_id INT PRIMARY KEY,

-> student\_id INT,

-> course\_id INT,

-> enrollment\_date DATE NOT NULL,

-> FOREIGN KEY (student\_id) REFERENCES Students(student\_id) ON DELETE CASCADE,

-> FOREIGN KEY (course\_id) REFERENCES Courses(course\_id) ON DELETE CASCADE

-> );

CREATE TABLE Payments (

-> payment\_id INT PRIMARY KEY,

-> student\_id INT,

-> amount DECIMAL(10,2) NOT NULL CHECK (amount > 0),

-> payment\_date DATE NOT NULL,

-> FOREIGN KEY (student\_id) REFERENCES Students(student\_id) ON DELETE CASCADE

-> );

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

**i. Students**

INSERT INTO Students (student\_id, first\_name, last\_name, date\_of\_birth, email, phone\_number) VALUES

(101, 'John', 'Doe', '2000-05-12', 'john.doe@example.com', 9876543210),

(102, 'Jane', 'Smith', '2001-07-15', 'jane.smith@example.com', 8765432109),

(103, 'Michael', 'Brown', '2002-09-20', 'michael.brown@example.com', 7654321098),

(104, 'Emily', 'Davis', '2000-12-05', 'emily.davis@example.com', 6543210987),

(105, 'David', 'Wilson', '2001-03-18', 'david.wilson@example.com', 5432109876);

**ii. Teachers**

INSERT INTO Teacher (teacher\_id, first\_name, last\_name, email) VALUES

(1, 'Alice', 'Johnson', 'alice.johnson@example.com'),

(2, 'Robert', 'Miller', 'robert.miller@example.com'),

(3, 'Emma', 'Garcia', 'emma.garcia@example.com');

**iii.Course**

INSERT INTO Courses (course\_id, course\_name, credits, teacher\_id) VALUES

(201, 'Mathematics', 3, 1),

(202, 'Physics', 4, 2),

(203, 'Computer Science', 3, 3),

(204, 'Biology', 4, 1),

(205, 'Chemistry', 3, 2);

**iv.Enrollments**

INSERT INTO Enrollments (enrollment\_id, student\_id, course\_id, enrollment\_date) VALUES

(301, 101, 201, '2024-01-15'),

(302, 102, 202, '2024-01-16'),

(303, 103, 203, '2024-01-17'),

(304, 104, 204, '2024-01-18'),

(305, 105, 205, '2024-01-19');

**V. Payments**

INSERT INTO Payments (payment\_id, student\_id, amount, payment\_date) VALUES

(401, 101, 1500.00, '2024-02-10'),

(402, 102, 2000.00, '2024-02-11'),

(403, 103, 1800.00, '2024-02-12'),

(404, 104, 2200.00, '2024-02-13'),

(405, 105, 1700.00, '2024-02-14');