CASE STUDY:

STUDENT INFORMATION MANAGEMENT SYSTEM

Students Name:P.Manikanta

VTU:29010

Reggistration Number:24UIE0040

To implement a **Student Information Management System (SIMS)** using SQL, you need to focus on defining the structure of the database (tables and relationships) and writing SQL queries for common operations like adding, viewing, updating, and deleting records.

Step 1: Creating the Database and Tables

You can create a new database and define the necessary tables for students, courses, enrollments, grades, and attendance.

```
Copy        -- Create the database

CREATE DATABASE StudentManagement;


-- Use the new database

USE StudentManagement;


-- Create Students table

CREATE TABLE Students (

   StudentID INT AUTO_INCREMENT PRIMARY KEY,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   DateOfBirth DATE,
   Email VARCHAR(100),
   Phone VARCHAR(15),
   Address TEXT
);

-- Create Courses table
CREATE TABLE Courses (
   CourseID INT AUTO_INCREMENT PRIMARY KEY,
   CourseName VARCHAR(100),
   Credits INT,
   Instructor VARCHAR(100)
);

-- Create Enrollments table
CREATE TABLE Enrollments (
   EnrollmentID INT AUTO_INCREMENT PRIMARY KEY,
   StudentID INT,
   CourseID INT,
   Semester VARCHAR(10),
   Year INT,
   FOREIGN KEY (StudentID) REFERENCES Students(StudentID) ON DELETE CASCADE,
   FOREIGN KEY (CourseID) REFERENCES Courses(CourseID) ON DELETE CASCADE
);

-- Create Grades table
CREATE TABLE Grades (
   GradeID INT AUTO_INCREMENT PRIMARY KEY,
   EnrollmentID INT,
   Grade VARCHAR(2),
   FOREIGN KEY (EnrollmentID) REFERENCES Enrollments(EnrollmentID) ON DELETE CASCADE
);

-- Create Attendance table
CREATE TABLE Attendance (
   AttendanceID INT AUTO_INCREMENT PRIMARY KEY,
   EnrollmentID INT,
   Date DATE,
   Status VARCHAR(10),
   FOREIGN KEY (EnrollmentID) REFERENCES Enrollments(EnrollmentID) ON DELETE CASCADE
);
```

Step 2: Inserting Sample Data

Once your tables are created, you can insert some sample data into the tables.

```
Copy        -- Insert sample students
INSERT INTO Students (FirstName, LastName, DateOfBirth, Email, Phone, Address)
VALUES
('John', 'Doe', '2000-05-15', 'john@example.com', '1234567890', '123 Main St'),
('Jane', 'Smith', '1999-08-20', 'jane@example.com', '9876543210', '456 Elm St');

-- Insert sample courses
INSERT INTO Courses (CourseName, Credits, Instructor)
VALUES
('Database Systems', 3, 'Dr. Smith'),
('Web Development', 4, 'Prof. Johnson');

-- Insert sample enrollments
INSERT INTO Enrollments (StudentID, CourseID, Semester, Year)
```

```
VALUES
(1, 1, 'Fall', 2023),
(2, 2, 'Spring', 2023);

-- Insert sample grades
INSERT INTO Grades (EnrollmentID, Grade)
VALUES
(1, 'A'),
(2, 'B');

-- Insert sample attendance records
INSERT INTO Attendance (EnrollmentID, Date, Status)
VALUES
(1, '2023-09-01', 'Present'),
(1, '2023-09-02', 'Absent'),
(2, '2023-09-01', 'Present'),
(2, '2023-09-02', 'Present');
```

Step 3: Basic CRUD Operations

Now, I will provide SQL queries for common operations.

1. View All Students:

```
Copy      SELECT * FROM Students;
```

2. View a Specific Student by ID:

```
Copy      SELECT * FROM Students WHERE StudentID = 1;
```

3. Update Student Information:

```
Copy      UPDATE Students
SET Email = 'john.doe@example.com', Phone = '1112223333'
WHERE StudentID = 1;
```

4. Delete a Student Record:

Copy

```
DELETE FROM Students WHERE StudentID = 1;
```

5. View All Courses:

Copy

```
SELECT * FROM Courses;
```

6. Enroll a Student in a Course:

Copy

```
INSERT INTO Enrollments (StudentID, CourseID, Semester, Year)
VALUES (1, 2, 'Fall', 2023);
```

7. Record a Grade:

Copy

```
INSERT INTO Grades (EnrollmentID, Grade)
VALUES (1, 'B+');
```

8. Track Attendance:

Copy

```
INSERT INTO Attendance (EnrollmentID, Date, Status)
VALUES (1, '2023-09-03', 'Present');
```

9. Get Student Grades:

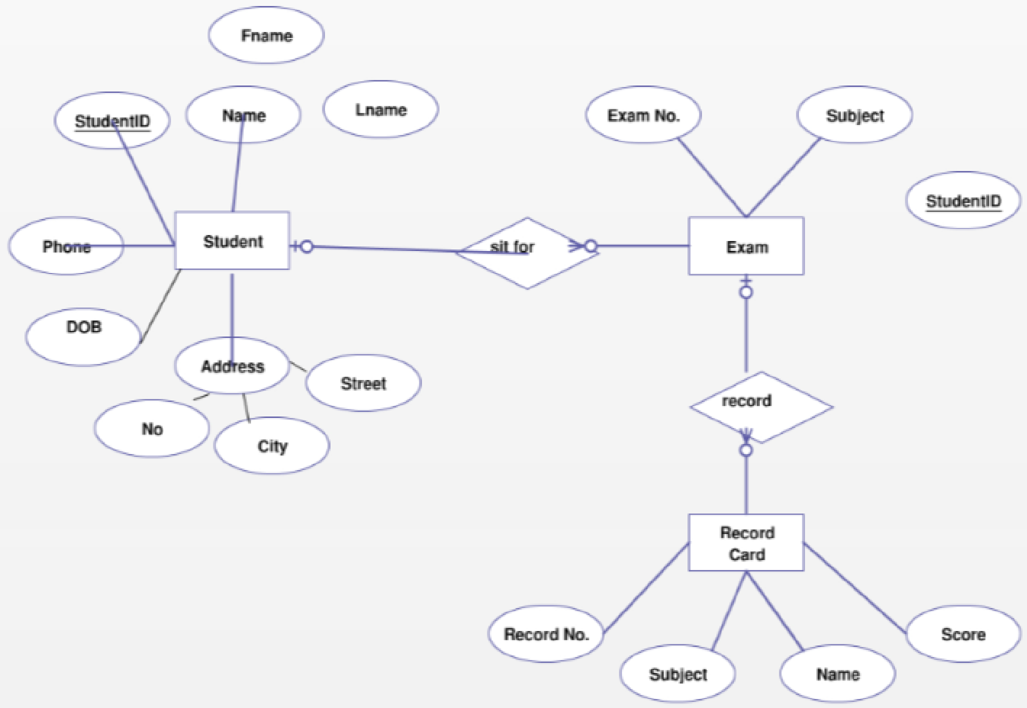
Copy

```
SELECT Students.FirstName, Students.LastName, Courses.CourseName,
Grades.Grade FROM Grades JOIN Enrollments ON Grades.EnrollmentID =
Enrollments.EnrollmentID JOIN Students ON Enrollments.StudentID = Students.StudentID
JOIN Courses ON Enrollments.CourseID = Courses.CourseID;
```

Conclusion

The SQL commands provided above outline a complete implementation of a **Student Information Management System**.

Student information system


Fname
StudentID
Name
Lname
Exam No.
Subject
StudentID
Phone
Student
sit for
Exam
DOB
Address
Street
No
City
record
Record Card
Record No.
Subject
Name
Score