Student Management System

Create Database:

```
-- Creating DATABASE
create database StudentManagementSystem;
use StudentManagementSystem;
```

Student Table:

```
-- Students Table
CREATE TABLE Students (
  StudentID INT PRIMARY KEY,
  Name VARCHAR(50) NOT NULL,
  Email VARCHAR(100) NOT NULL,
  Phone VARCHAR(20) NOT NULL,
  Address VARCHAR(200) NOT NULL
);
```

```
-- Insert data into students
INSERT INTO Students (StudentID, Name, Email, Phone, Address)
VALUES
(1, 'Venkat', 'venkat01@gmail.com', '123-456-7890', '123 Main St,adyar,chennai'),
(2, 'varadha', 'varadha02@gmail.com', '987-654-3210', '456 ramakrishna St,egmore,chennai'),
(3, 'sri', 'sri03@gmail.com', '555-123-4567', '789 bharatiyar st ,ambattur,chennai'),
(4, 'ram', 'ram04@gmail.com', '901-234-5678', '321 main St ,guindy,chennai'),
(5, 'vivek', 'vivek05@gmail.com', '111-222-3333', '444 kamaraj St,mylapore,chennai');

select * from students;
```

StudentID	Name	Email	Phone	Address
1	Venkat	venkat01@gmail.com	123-456-7890	123 Main St,adyar,chennai
2	varadha	varadha02@gmail.com	987-654-3210	456 ramakrishna St,egmore,chennai
3	sri	sri03@gmail.com	555-123-4567	789 bharatiyar st ,ambattur,chennai
4	ram	ram04@gmail.com	901-234-5678	321 main St ,guindy,chennai
5	vivek	vivek05@gmail.com	111-222-3333	444 kamaraj St,mylapore,chennai

Course table:

```
-- Courses Table
CREATE TABLE Courses (
  CourseID INT PRIMARY KEY,
  CourseName VARCHAR(100) NOT NULL,
  CourseDescription VARCHAR(200) NOT NULL,
  Credits INT NOT NULL
);

-- Insert data into courses
INSERT INTO Courses (CourseID, CourseName, CourseDescription, Credits)
VALUES
(1, 'data science 101', ' introduction to machine learning', 3),
(2, 'full stack developement 102', 'Introduction to full stack ', 4),
(3, 'data analytics 103', 'Introduction to data analytics', 3),
(4, 'cloud computing 104', 'Introduction to cloud computing', 4),
(5, 'devops 105', 'Introduction to devops', 3);

select * from courses;
```

CourseID	CourseName	CourseDescription	Credits
1	data science 101	introduction to machine learning	3
2	full stack developement 102	Introduction to full stack	4
3	data analytics 103	Introduction to data analytics	3
4	cloud computing 104	Introduction to cloud computing	4
5	devops 105	Introduction to devops	3

Enrollment table:

```
-- Enrollments Table
CREATE TABLE Enrollments (
  EnrollmentID INT PRIMARY KEY,
  StudentID INT NOT NULL,
  CourseID INT NOT NULL,
  EnrollmentDate DATE NOT NULL,
  FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
  FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)
);
```

```
-- Insert data into Enrollments
INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, EnrollmentDate)
VALUES
(1, 1, 1, '2024-01-01'),
(2, 1, 2, '2024-02-01'),
(3, 2, 1, '2024-03-01'),
(4, 2, 3, '2024-04-01'),
(5, 3, 2, '2024-05-01'),
(6, 3, 4, '2024-06-01'),
(7, 4, 1, '2024-07-01'),
(8, 4, 5, '2024-08-01'),
(9, 5, 3, '2024-09-01'),
(10, 5, 4, '2024-10-01');

select * from enrollments;
```

EnrollmentID	StudentID	CourseID	EnrollmentDate
1	1	1	2024-01-01
2	1	2	2024-02-01
3	2	1	2024-03-01
4	2	3	2024-04-01
5	3	2	2024-05-01
6	3	4	2024-06-01
7	4	1	2024-07-01
8	4	5	2024-08-01
9	5	3	2024-09-01
10	5	4	2024-10-01

Grades Table:

```
-- Grades Table
CREATE TABLE Grades (
  GradeID INT PRIMARY KEY,
  EnrollmentID INT NOT NULL,
  Grade VARCHAR(2) NOT NULL,
  GPA DECIMAL(3,2) NOT NULL,
  FOREIGN KEY (EnrollmentID) REFERENCES Enrollments(EnrollmentID)
);
```

```sql
-- Insert data into Grades
INSERT INTO Grades (GradeID, EnrollmentID, Grade, GPA)
VALUES
(1, 1, 'A', 9.0),
(2, 2, 'B', 8.0),
(3, 3, 'A', 9.0),
(4, 4, 'B', 8.0),
(5, 5, 'C', 7.0),
(6, 6, 'A', 9.0),
(7, 7, 'B', 8.0),
(8, 8, 'C', 7.0),
(9, 9, 'D', 6.0),
(10, 10, 'F', 4.0);
```

GradeID	EnrollmentID	Grade	GPA
1	1	A	9.00
2	2	B	8.00
3	3	A	9.00
4	4	B	8.00
5	5	C	7.00
6	6	A	9.00
7	7	B	8.00
8	8	C	7.00
9	9	D	6.00
10	10	F	4.00

Instructor Table:

```sql
-- Instructors Table
CREATE TABLE Instructors (
  InstructorID INT PRIMARY KEY,
  Name VARCHAR(50) NOT NULL,
  Email VARCHAR(100) NOT NULL,
  Phone VARCHAR(20) NOT NULL,
  Department VARCHAR(50) NOT NULL
);

-- Insert data into Instructors
INSERT INTO Instructors (InstructorID, Name, Email, Phone, Department)
VALUES
(1, 'mr.shiva', 'shiva@gmail.com', '555-123-4567', 'information technology'),
(2, 'mr.kumar', 'kumar@gmail.com', '901-234-5678', 'computer Science'),
(3, 'mr.rajesh', 'rajesh@gmail.com', '111-222-3333', 'artificial intelligence'),
(4, 'mrs.sheela', 'sheela@gmail.com', '444-555-6666', 'computer science'),
(5, 'mrs.rajee', 'rajee@gmail.com', '777-888-9999', 'information technology');
```

InstructorID	Name	Email	Phone	Department
1	mr.shiva	shiva@gmail.com	555-123-4567	information technology
2	mr.kumar	kumar@gmail.com	901-234-5678	computer Science
3	mr.rajesh	rajesh@gmail.com	111-222-3333	artificial intelligence
4	mrs.sheela	sheela@gmail.com	444-555-6666	computer science
5	mrs.rajee	rajee@gmail.com	777-888-9999	information technology

Course Instructor Table:

```sql
-- courseInstructors Table
CREATE TABLE CourseInstructors (
  CourseID INT NOT NULL,
  InstructorID INT NOT NULL,
  FOREIGN KEY (CourseID) REFERENCES Courses(CourseID),
  FOREIGN KEY (InstructorID) REFERENCES Instructors(InstructorID)
);

-- Insert data into courseinstructors
INSERT INTO CourseInstructors (CourseID, InstructorID)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5);

select * from courseinstructors;
```

CourseID	InstructorID
1	1
2	2
3	3
4	4
5	5

Joins:

```
-- Using JOINS
SELECT s.StudentID, s.Name, c.CourseName, g.Grade, g.GPA, i.Name AS InstructorName
FROM Students s
INNER JOIN Enrollments e ON s.StudentID = e.StudentID
INNER JOIN Courses c ON e.CourseID = c.CourseID
INNER JOIN Grades g ON e.EnrollmentID = g.EnrollmentID
INNER JOIN CourseInstructors ci ON c.CourseID = ci.CourseID
INNER JOIN Instructors i ON ci.InstructorID = i.InstructorID
WHERE c.CourseID = 1;
```

StudentID	Name	CourseName	Grade	GPA	InstructorName
1	Venkat	data science 101	A	9.00	mr.shiva
2	varadha	data science 101	A	9.00	mr.shiva
4	ram	data science 101	B	8.00	mr.shiva

View:

```
-- View
CREATE VIEW StudentTranscript AS
SELECT s.StudentID, s.Name, c.CourseName, g.Grade, g.GPA
FROM Students s
JOIN Enrollments e ON s.StudentID = e.StudentID
JOIN Courses c ON e.CourseID = c.CourseID
JOIN Grades g ON e.EnrollmentID = g.EnrollmentID;
select * from studenttranscript;
```

Sub query:

```
-- SUBQUERY
SELECT *
FROM Students
WHERE StudentID IN (
SELECT StudentID
FROM Enrollments
WHERE CourseID = 1
);
```

StudentID	Name
1	Venkat
2	varadha
4	ram
NULL	NULL

Stored Procedure:

```
-- STORED PROCEDURE:

DELIMITER $$

CREATE PROCEDURE GetStudentTranscript()
BEGIN
    SELECT s.StudentID, s.Name, c.CourseName, g.Grade, g.GPA
    FROM Students s
    JOIN Enrollments e ON s.StudentID = e.StudentID
    JOIN Courses c ON e.CourseID = c.CourseID
    JOIN Grades g ON e.EnrollmentID = g.EnrollmentID;
END $$

DELIMITER ;
```

```
call getstudenttranscript();
```

StudentID	Name	CourseName	Grade	GPA
1	Venkat	data science 101	A	9.00
1	Venkat	full stack developement 102	B	8.00
2	varadha	data science 101	A	9.00
2	varadha	data analytics 103	B	8.00
3	sri	full stack developement 102	C	7.00
3	sri	cloud computing 104	A	9.00
4	ram	data science 101	B	8.00
4	ram	devops 105	C	7.00
5	vivek	data analytics 103	D	6.00
5	vivek	cloud computing 104	F	4.00