**Project report on College Management System**

**Abstract**

This project involves creating a database to manage student records, courses, and faculty details in a college. It provides hands-on experience in designing a relational database for a real-world scenario, illustrating how students, courses, and faculty are interconnected.

**Introduction**

The college has multiple academic departments, such as English, Mathematics, History, etc. Each department offers a variety of courses, and instructors can teach more than one course. For example, a professor may teach both Statistics and Calculus. Students can enroll in multiple courses, and each course can have multiple students but only one instructor to avoid overlaps.

The database schema includes the following tables:

* **Departments**: Contains information about academic departments.
* **Faculty**: Contains information about faculty members.
* **Courses**: Contains information about courses offered by departments.
* **Students**: Contains information about students.
* **Enrolments**: Contains information about student enrolments in courses.

**SQL Scripts**

**1. Create the Database**

CREATE DATABASE CollegeDB

USE CollegeDB;

**2. Create Tables**

**Departments Table**

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY AUTO\_INCREMENT,

DepartmentName VARCHAR(100) NOT NULL

);

**Faculty Table**

CREATE TABLE Faculty (

FacultyID INT PRIMARY KEY AUTO\_INCREMENT,

FacultyName VARCHAR(100) NOT NULL,

DepartmentID INT,

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

**Courses Table**

CREATE TABLE Courses (

CourseID INT PRIMARY KEY AUTO\_INCREMENT,

CourseName VARCHAR(100) NOT NULL,

DepartmentID INT,

FacultyID INT,

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID),

FOREIGN KEY (FacultyID) REFERENCES Faculty(FacultyID)

);

**Students Table**

CREATE TABLE Students (

StudentID INT PRIMARY KEY AUTO\_INCREMENT,

StudentName VARCHAR(100) NOT NULL,

DepartmentID INT,

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

**Enrolment Table**

CREATE TABLE Enrollment (

EnrollmentID INT PRIMARY KEY AUTO\_INCREMENT,

StudentID INT,

CourseID INT,

FOREIGN KEY (StudentID) REFERENCES Students(StudentID),

FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)

);

**3. Insert Sample Data**

**Insert Departments**

INSERT INTO Departments (DepartmentName) VALUES

('English'),

('Mathematics'),

('History');

**Insert Faculty**

INSERT INTO Faculty (FacultyName, DepartmentID) VALUES

('Dr. Smith', 2),

('Prof. Johnson', 2),

('Dr. Brown', 1),

('Prof. White', 3);

**Insert Courses**

INSERT INTO Courses (CourseName, DepartmentID, FacultyID) VALUES

('Calculus', 2, 1),

('Statistics', 2, 2),

('Literature', 1, 3),

('World History', 3, 4);

**Insert Students**

INSERT INTO Students (StudentName, DepartmentID) VALUES

('Alice', 2),

('Bob', 2),

('Charlie', 1),

('David', 3);

**Insert Enrollments**

INSERT INTO Enrollment (StudentID, CourseID) VALUES

(1, 1),

(1, 2),

(2, 1),

(3, 3),

(4, 4);

**Query Examples**

**List All Students**

SELECT \* FROM Students;

**List All Courses**

SELECT \* FROM Courses;

**List All Faculty**

SELECT \* FROM Faculty;

**List All Enrollments**

SELECT \* FROM Enrollment;

**List All Courses a Specific Student is Enrolled In**

SELECT c.CourseName

FROM Courses c

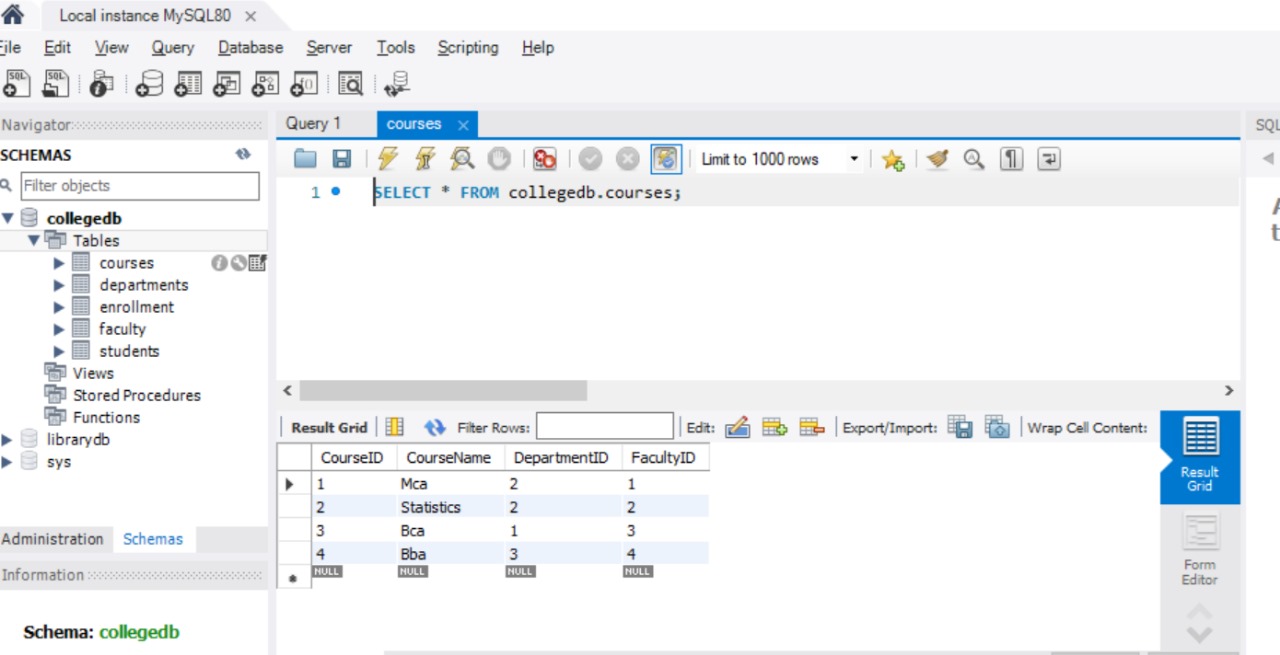
JOIN Enrollment e ON c.CourseID = e.CourseID

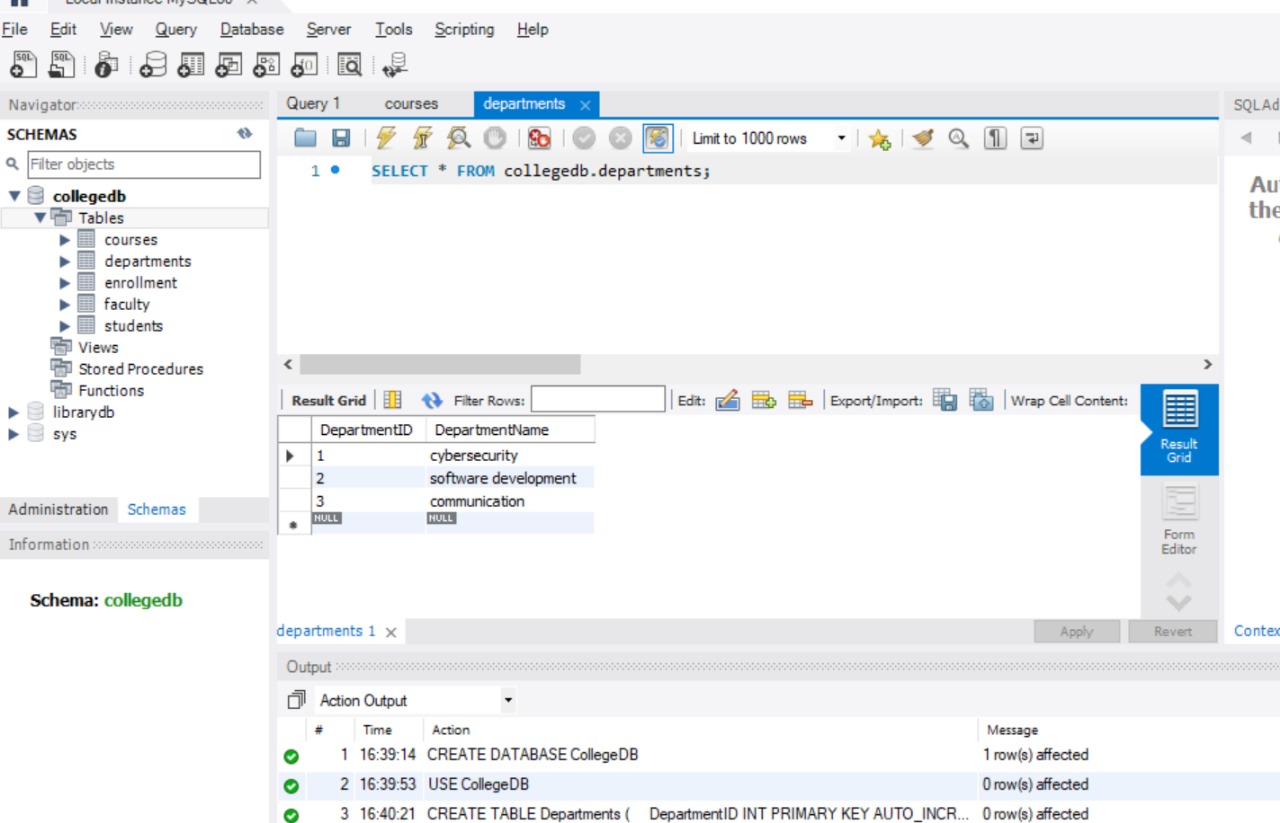
WHERE e.StudentID = 1

**Conclusion**

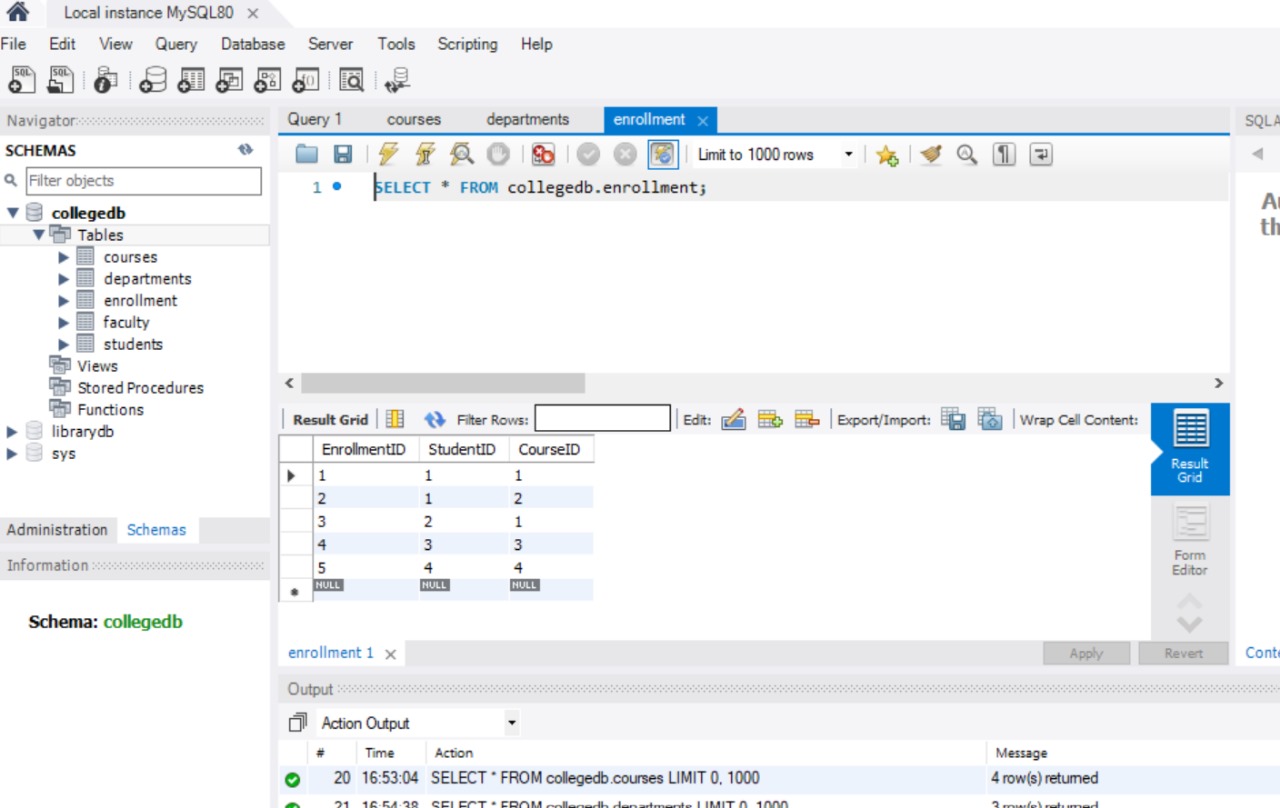
This project demonstrates the creation and management of a relational database for a college management system. It includes designing the database schema, inserting sample data, and performing various queries to manage and retrieve information about students, courses, and faculty.

**Output**

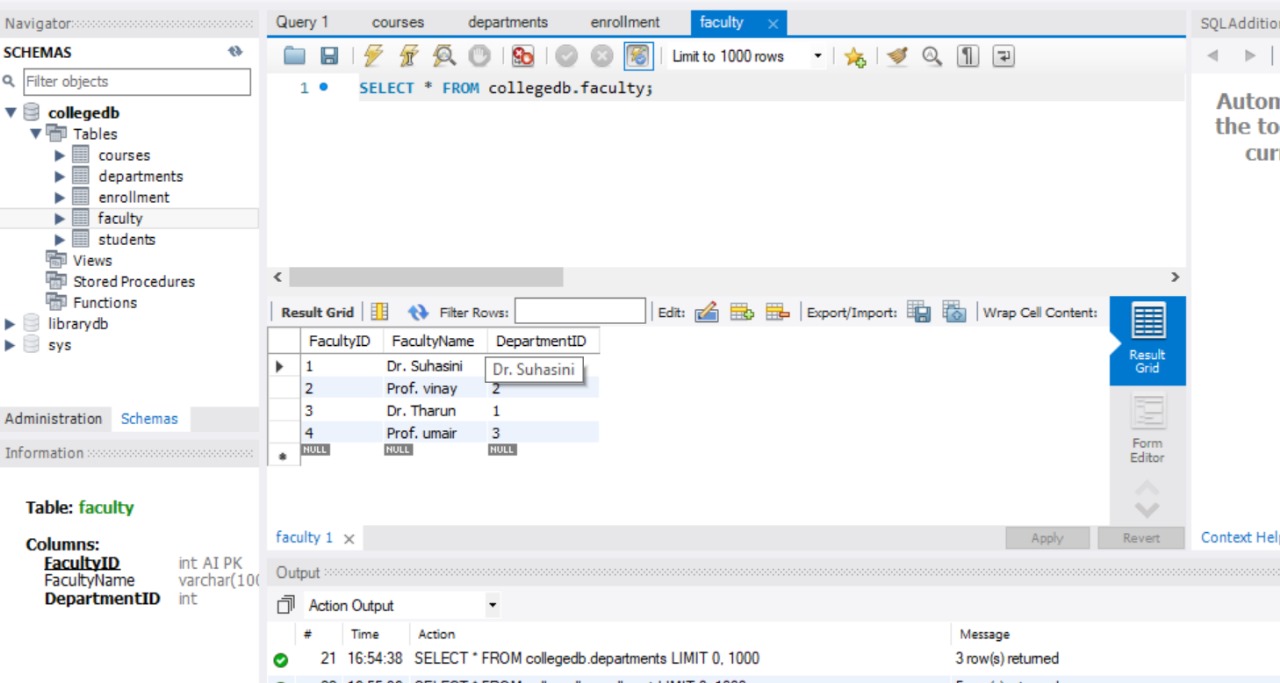




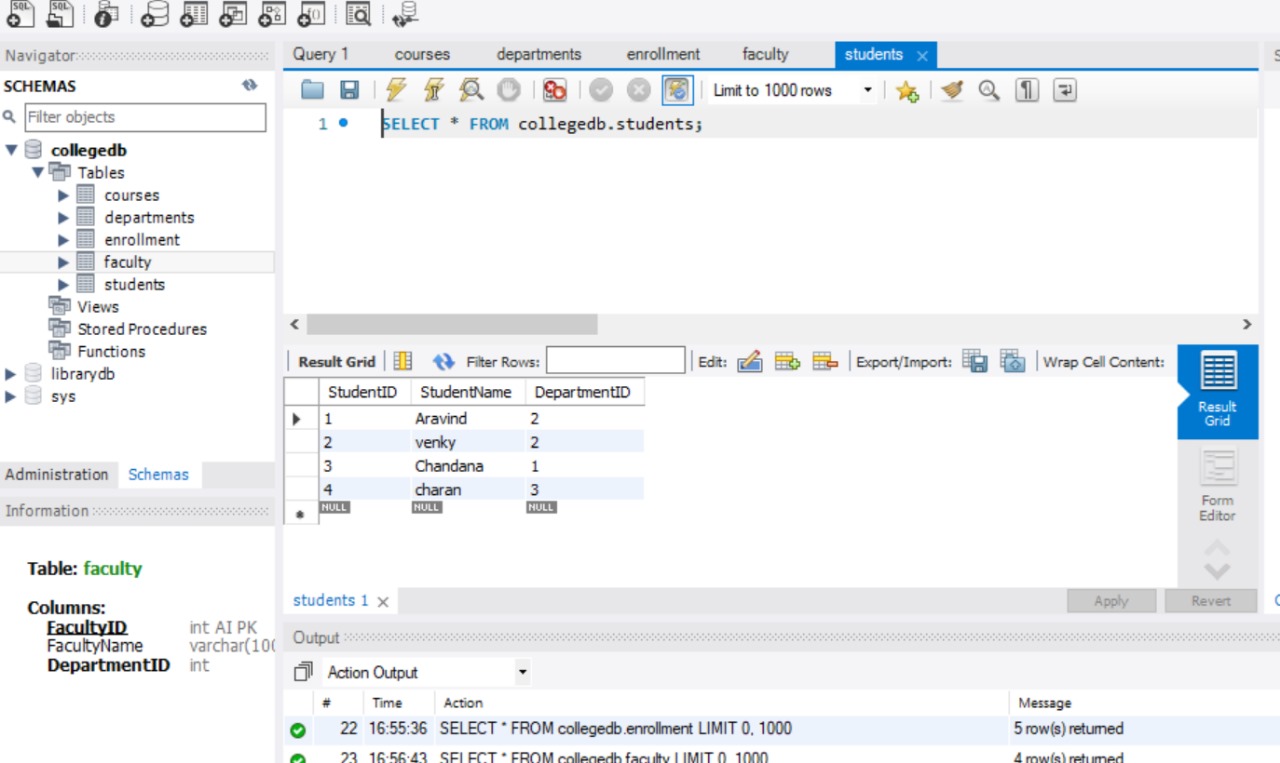
**Departments**



**Enrollment**



**Faculity**



**students**