**Task -1 : University Task**

CREATE DATABASE University;

USE University;

**Create Tables :**

**Departments Table-**

CREATE TABLE Departments(department\_id int PRIMARY KEY, department\_name varchar(100));

**Professors Table-**

CREATE TABLE Professors(professor\_id int PRIMARY KEY,first\_name varchar(100),last\_name varchar(100),email varchar(100),phone varchar(20));

**Students Table-**

CREATE TABLE Students(student\_id int PRIMARY KEY,first\_name varchar(100),last\_name varchar(100),email varchar(100),phone varchar(20),date\_of\_birth date,enrollment\_date date,department\_id int,FOREIGN KEY (department\_id) REFERENCES departments(department\_id));

**Courses Table-**

CREATE TABLE Courses(course\_id int PRIMARY KEY,course\_name varchar(100),department\_id int,professor\_id int,credits int,FOREIGN KEY (department\_id) REFERENCES departments(department\_id),FOREIGN KEY (professor\_id) REFERENCES professors(professor\_id));

**Enrollments Table-**

CREATE TABLE Enrollments(enrollment\_id int PRIMARY KEY,student\_id int,course\_id int,enrollment\_date date,grade varchar(5),FOREIGN KEY (student\_id) REFERENCES students(student\_id),FOREIGN KEY (course\_id) REFERENCES courses(course\_id));

**Insert Data in the Tables :**

**Departments Table –**

INSERT INTO Departments (department\_id, department\_name) VALUES (1, 'Computer Science'), (2, 'Mechanical Engineering'), (3, 'Electronics and Communication'), (4, 'Civil Engineering'), (5, 'Business Administration');

**Professors Table –**

INSERT INTO Professors (professor\_id, first\_name, last\_name, email, phone) VALUES (1, 'Ramesh', 'Kumar', 'ramesh.kumar@example.com', '9876543210'), (2, 'Anita', 'Sharma', 'anita.sharma@example.com', '8765432109'), (3, 'Vikram', 'Patel', 'vikram.patel@example.com', '7654321098'), (4, 'Suresh', 'Gupta', 'suresh.gupta@example.com', '6543210987'), (5, 'Neha', 'Verma', 'neha.verma@example.com', '5432109876');

**Courses Table –**

INSERT INTO Courses (course\_id, course\_name, department\_id, professor\_id, credits) VALUES (1, 'Database Systems', 1, 1, 4), (2, 'Data Structures', 1, 1, 4), (3, 'Thermodynamics', 2, 2, 3), (4, 'Microprocessors', 3, 3, 3), (5, 'Digital Signal Processing', 3, 3, 3), (6, 'Structural Analysis', 4, 4, 3), (7, 'Business Strategy', 5, 5, 3), (8, 'Marketing Management', 5, 5, 3), (9, 'Artificial Intelligence', 1, 1, 5), (10, 'Fluid Mechanics', 2, 2, 4);

**Students Table –**

INSERT INTO Students (student\_id, first\_name, last\_name, email, phone, date\_of\_birth, enrollment\_date, department\_id) VALUES (11, 'Amit', 'Sharma', 'amit.sharma@example.com', '9998887776', '2002-05-14', '2020-08-15', 1), (12, 'Priya', 'Mehta', 'priya.mehta@example.com', '8887776665', '2001-07-22', '2019-07-15', 1), (13, 'Rahul', 'Singh', 'rahul.singh@example.com', '7776665554', '2000-12-01', '2018-08-15', 2), (14, 'Sneha', 'Reddy', 'sneha.reddy@example.com', '6665554443', '2003-03-18', '2021-08-15', 3), (15, 'Vikas', 'Gupta', 'vikas.gupta@example.com', '5554443332', '2002-10-10', '2020-08-15', 3), (16, 'Pooja', 'Nair', 'pooja.nair@example.com', '4443332221', '2001-01-25', '2019-07-15', 4), (17, 'Karan', 'Kapoor', 'karan.kapoor@example.com', '3332221110', '2000-11-05', '2018-08-15', 4), (18, 'Neha', 'Joshi', 'neha.joshi@example.com', '2221110009', '2003-06-30', '2021-08-15', 5), (19, 'Arjun', 'Pandey', 'arjun.pandey@example.com', '1110009998', '2002-04-04', '2020-08-15', 5), (20, 'Meera', 'Shah', 'meera.shah@example.com', '9998887777', '2001-09-15', '2019-07-15', 1);

**Enrollments Table –**

INSERT INTO Enrollments (enrollment\_id, student\_id, course\_id, enrollment\_date, grade) VALUES (1, 11, 1, '2020-08-15', 'A'), (2, 11, 2, '2020-08-15', 'B'), (3, 12, 3, '2019-07-15', 'A'), (4, 12, 4, '2019-07-15', 'B'), (5, 13, 5, '2018-08-15', 'C'), (6, 13, 6, '2018-08-15', 'A'), (7, 14, 7, '2021-08-15', 'B'), (8, 15, 8, '2020-08-15', 'A'), (9, 15, 9, '2020-08-15', 'B'), (10, 16, 10, '2019-07-15', 'C'),

**SQL Queries for the Case Study**

1. **Find the Total Number of Students in Each Department**

SELECT d.department\_name,count(s.student\_id) as total\_students

FROM students AS s

JOIN departments AS d

ON

s.department\_id = d.department\_id

GROUP by d.department\_name;

1. **List All Courses Taught by a Specific Professor**

SELECT c.course\_name

FROM courses AS c

JOIN professors AS p

on c.professor\_id = p.professor\_id

where p.first\_name ="Ramesh" AND p.last\_name ="Kumar";

1. **Find the Average Grade of Students in Each Course**

Select c.course\_name , avg(CASE grade

WHEN 'A' THEN 4

WHEN 'B' THEN 3

WHEN 'C' THEN 2

ELSE 0 END) as average\_grade

from courses as c

JOIN enrollments as e

on c.course\_id = e.course\_id

GROUP by c.course\_name;

1. **List All Students Who Have Not Enrolled in Any Courses**

SELECT first\_name, last\_name

FROM Students

WHERE student\_id NOT IN (SELECT student\_id FROM Enrollments);

1. **Find the Number of Courses Offered by Each Department**

SELECT d.department\_name,COUNT(c.course\_id) as Total\_courses

from departments as d

JOIN courses as c

on d.department\_id = c.department\_id

GROUP by d.department\_name;

1. **List All Students Who Have Taken a Specific Course (e.g., 'Database Systems')**

SELECT s.first\_name,s.last\_name

FROM students as s

JOIN enrollments as e

on s.student\_id = e.student\_id

JOIN courses as c

on e.course\_id = c.course\_id

WHERE c.course\_name ='Data Structures';

1. **Find the Most Popular Course Based on Enrollment Numbers**

SELECT c.course\_name,count(e.enrollment\_id) as total\_enrollments

FROM courses as c

JOIN enrollments as e

on c.course\_id = e.enrollment\_id

GROUP by c.course\_name

ORDER BY total\_enrollments desc

limit 1;

1. **Find the Average Number of Credits Per Student in a Department**

SELECT d.department\_name,AVG(c.credits) AS avg\_credits

from students as s

JOIN enrollments as e

on s.student\_id = e.student\_id

JOIN courses as c

on e.course\_id = c.course\_id

JOIN departments as d

on c.department\_id = d.department\_id

GROUP by d.department\_name;

1. **List All Professors Who Teach in More Than One Department**

SELECT p.first\_name,p.last\_name from professors as p

JOIN courses as c

on p.professor\_id = c.professor\_id

GROUP by p.professor\_id

HAVING COUNT(DISTINCT c.department\_id)>1;

1. **Get the Highest and Lowest Grade in a Specific Course (e.g., 'Operating Systems')**

SELECT MAX(e.grade) as highest\_grade ,

MIN(e.grade) as lowest\_grade

from enrollments as e

join courses as c

on e.course\_id = c.course\_id

WHERE c.course\_name='Database System';