***#Task-1***

**#Create Data Base :**

create database UniversityDatabase;

**#Use Data Base :**

use UniversityDatabase;

**#Create Departments Table :**

CREATE TABLE Departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(100) NOT NULL

);

**#Insert Data into Departments Table :**

INSERT INTO Departments (department\_id, department\_name) VALUES

(1, 'Computer Science'),

(2, 'Electrical Engineering'),

(3, 'Mechanical Engineering'),

(4, 'Civil Engineering'),

(5, 'Chemical Engineering'),

(6, 'Biology'),

(7, 'Chemistry'),

(8, 'Physics'),

(9, 'Mathematics'),

(10, 'Psychology'),

(11, 'Economics'),

(12, 'Sociology'),

(13, 'History'),

(14, 'Philosophy'),

(15, 'Political Science');

**#Create Professors Table :**

CREATE TABLE Professors (

professor\_id INT PRIMARY KEY,

first\_name VARCHAR(100) NOT NULL,

last\_name VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL,

phone VARCHAR(20) NOT NULL

);

**#Insert Data into Professors Table :**

INSERT INTO Professors (professor\_id, first\_name, last\_name, email, phone) VALUES

(1, 'Aarav', 'Sharma', 'aarav.sharma@example.com', '9876543210'),

(2, 'Priya', 'Patel', 'priya.patel@example.com', '9887654321'),

(3, 'Ravi', 'Verma', 'ravi.verma@example.com', '9876123456'),

(4, 'Sanya', 'Gupta', 'sanya.gupta@example.com', '9789345678'),

(5, 'Arjun', 'Singh', 'arjun.singh@example.com', '9776123456'),

(6, 'Meera', 'Iyer', 'meera.iyer@example.com', '9845123456'),

(7, 'Vikram', 'Reddy', 'vikram.reddy@example.com', '9998765432'),

(8, 'Anjali', 'Kumar', 'anjali.kumar@example.com', '9988776655'),

(9, 'Aditi', 'Nair', 'aditi.nair@example.com', '9898765432'),

(10, 'Manish', 'Joshi', 'manish.joshi@example.com', '9712345678'),

(11, 'Neha', 'Chopra', 'neha.chopra@example.com', '9823456789'),

(12, 'Karan', 'Mehta', 'karan.mehta@example.com', '9745123456'),

(13, 'Shreya', 'Deshmukh', 'shreya.deshmukh@example.com', '9875432109'),

(14, 'Siddharth', 'Bansal', 'siddharth.bansal@example.com', '9776543210'),

(15, 'Komal', 'Aggarwal', 'komal.aggarwal@example.com', '9967321865');

**#Create Students** **Table :**

CREATE TABLE Students (

student\_id INT PRIMARY KEY,

first\_name VARCHAR(100) NOT NULL,

last\_name VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL,

phone VARCHAR(20) NOT NULL,

date\_of\_birth DATE NOT NULL,

enrollment\_date DATE NOT NULL,

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES Departments(department\_id)

);

**#Insert Data into Students** **Table :**

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

(1, 'Amit', 'Sharma', 'amit.sharma@student.com', '9876543210', '2000-05-10', '2020-07-01', 1),

(2, 'Neha', 'Verma', 'neha.verma@student.com', '9887654321', '1999-11-22', '2019-08-15', 2),

(3, 'Raj', 'Kumar', 'raj.kumar@student.com', '9776123456', '2001-03-14', '2021-06-10', 3),

(4, 'Riya', 'Iyer', 'riya.iyer@student.com', '9845123456', '2000-07-05', '2020-09-20', 4),

(5, 'Sahil', 'Patel', 'sahil.patel@student.com', '9998765432', '1998-12-30', '2018-01-10', 5),

(6, 'Maya', 'Reddy', 'maya.reddy@student.com', '9876549876', '2001-02-20', '2020-11-01', 6),

(7, 'Vikram', 'Joshi', 'vikram.joshi@student.com', '9801122334', '1999-06-11', '2020-03-10', 7),

(8, 'Pooja', 'Gupta', 'pooja.gupta@student.com', '9888776655', '2001-07-14', '2020-08-30', 8),

(9, 'Sandeep', 'Kumar', 'sandeep.kumar@student.com', '9887765432', '1999-10-05', '2019-07-25', 9),

(10, 'Ravi', 'Mehta', 'ravi.mehta@student.com', '9712345678', '2000-04-22', '2020-12-15', 10),

(11, 'Anjali', 'Nair', 'anjali.nair@student.com', '9976543210', '2000-09-12', '2021-02-18', 1),

(12, 'Karan', 'Singh', 'karan.singh@student.com', '9765432109', '2001-01-05', '2021-03-12', 2),

(13, 'Shreya', 'Shukla', 'shreya.shukla@student.com', '9638527410', '2001-05-17', '2021-05-20', 3),

(14, 'Manish', 'Patel', 'manish.patel@student.com', '9348756234', '1999-08-19', '2020-09-05', 4),

(15, 'Tanu', 'Chopra', 'tanu.chopra@student.com', '9871234567', '2000-12-25', '2019-10-20', 5);

**#Create Courses** **Table :**

CREATE TABLE Courses (

course\_id INT PRIMARY KEY,

course\_name VARCHAR(100) NOT NULL,

department\_id INT,

professor\_id INT,

credits INT NOT NULL,

FOREIGN KEY (department\_id) REFERENCES Departments(department\_id),

FOREIGN KEY (professor\_id) REFERENCES Professors(professor\_id)

);

**#Insert Data into Courses** **Table :**

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

(1, 'Introduction to Computer Science', 1, 1, 3),

(2, 'Data Structures and Algorithms', 1, 2, 4),

(3, 'Database Management Systems', 1, 3, 3),

(4, 'Digital Circuits', 2, 4, 4),

(5, 'Operating Systems', 1, 5, 3),

(6, 'Electromagnetic Fields', 2, 6, 3),

(7, 'Advanced Programming', 1, 7, 3),

(8, 'Control Systems', 2, 8, 4),

(9, 'Software Engineering', 1, 9, 3),

(10, 'Machine Learning', 1, 10, 4),

(11, 'Signal Processing', 2, 11, 3),

(12, 'Microprocessors', 2, 12, 3),

(13, 'Computer Networks', 1, 13, 3),

(14, 'Artificial Intelligence', 1, 14, 4),

(15, 'Robotics', 2, 15, 4);

**#Create 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 into Enrollments** **Table :**

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

(1, 1, 1, '2020-07-01', 'A'),

(2, 1, 2, '2020-07-01', 'B'),

(3, 2, 3, '2019-08-15', 'A'),

(4, 3, 1, '2021-06-10', 'B'),

(5, 4, 5, '2020-09-20', 'C'),

(6, 5, 6, '2018-01-10', 'B'),

(7, 6, 7, '2020-11-01', 'A'),

(8, 7, 8, '2020-03-10', 'B'),

(9, 8, 9, '2020-08-30', 'A'),

(10, 9, 10, '2019-07-25', 'C'),

(11, 10, 11, '2020-12-15', 'A'),

(12, 11, 12, '2021-02-18', 'B'),

(13, 12, 13, '2021-03-12', 'B'),

(14, 13, 14, '2021-05-20', 'A'),

(15, 14, 15, '2019-10-20', 'C');

***Solve Query :***

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

Query 🡪 SELECT department\_name, COUNT(student\_id) AS total\_students

FROM Students

JOIN Departments ON Students.department\_id = Departments.department\_id

GROUP BY department\_name;

**2. List All Courses Taught by a Specific Professo.**

Query🡪 SELECT p.first\_name , p.last\_name , COUNT(c.course\_name) AS total\_courses

FROM Professors p

JOIN Courses c ON c.professor\_id = p.professor\_id

GROUP BY p.professor\_id;

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

Query🡪 SELECT c.course\_name,

Round(AVG(CASE

WHEN e.grade = 'A' THEN 4

WHEN e.grade = 'B' THEN 3

WHEN e.grade = 'C' THEN 2

WHEN e.grade = 'D' THEN 1

ELSE 0

END),2) AS average\_grade

FROM Enrollments e

JOIN Courses c ON e.course\_id = c.course\_id

GROUP BY c.course\_name;

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

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

FROM Students s

INNER JOIN Enrollments e ON s.student\_id = e.student\_id

WHERE e.enrollment\_id IS NULL;

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

Query🡪 SELECT d.department\_name, COUNT(c.course\_id) AS total\_courses

FROM Departments d

JOIN Courses c ON d.department\_id = c.department\_id

GROUP BY d.department\_name;

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

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

FROM Students s

JOIN Enrollments e ON s.student\_id = e.student\_id

JOIN Courses c ON e.course\_id = c.course\_id

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

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

Query🡪 SELECT c.course\_name, COUNT(e.student\_id) AS enrollment\_count

FROM Courses c

JOIN Enrollments e ON c.course\_id = e.course\_id

GROUP BY c.course\_name

ORDER BY enrollment\_count DESC

LIMIT 1;

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

Query🡪

SELECT s.department\_id,d.department\_name ,avg(total.tcredit)

as avgCreditPerStud from (SELECT e.student\_id,sum(c.credits) as tcredit FROM enrollments e

JOIN courses c on e.course\_id = c.course\_id GROUP BY e.student\_id)as total

JOIN students s ON total.student\_id = s.student\_id

JOIN departments d on s.department\_id = d.department\_id

GROUP BY s.department\_id,d.department\_name;

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

Query🡪 SELECT p.first\_name, p.last\_name

FROM Professors p

JOIN Courses c ON p.professor\_id = c.professor\_id

JOIN Departments d ON c.department\_id = d.department\_id

GROUP BY p.professor\_id

HAVING COUNT(d.department\_id) > 1;

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

Query🡪 SELECT

c.course\_name,

MAX(e.grade) AS highest\_grade,

MIN(e.grade) AS lowest\_grade

FROM Enrollments e

JOIN Courses c ON e.course\_id = c.course\_id

WHERE c.course\_name = 'Operating Systems'

GROUP BY c.course\_name;