**Internship Task - RDBMS and SQL Task #1**

**SQL case-based assignment using a University Database schema. This assignment will involve queries related to students, courses, departments, professors, and enrollments. I'll walk through the case, describe the database schema, and then provide 10 SQL queries related to university data analysis.**

create database university;

CREATE TABLE Departments(

department\_id INT PRIMARY KEY,

department\_name VARCHAR(100)

);

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

(1, 'Computer Application'),

(2, 'Electronics'),

(3, ' Graphic design'),

(4, 'Mining Engineering'),

(5, 'Biotechnology');

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)

)

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

(1, 'Shantanu', 'Verma', 'shantanu.verma@student.in', '626512340', '2001-05-20', '2023-08-01', 1),

(2, 'Ankita', 'Singh', 'ankita.singh@student.in', '9876322341', '2002-03-15', '2023-08-01', 1),

(3, 'Aman', 'Gupta', 'aman.gupta@student.in', '8765123542', '2001-12-10', '2023-08-01', 2),

(4, 'Kapil', 'Sharma', 'kapil.sharma@student.in', '8666512343', '2002-07-25', '2023-08-01', 3),

(5, 'Kritika', 'singh', 'kritika.singh@student.in', '894512344', '2001-09-18', '2023-08-01', 2),

(6, 'Shraya', 'Pandey', 'shraya.pandey@student.in', '89656512345', '2002-01-11', '2023-08-01', 4),

(7, 'Rohan', 'Mishra', 'rohan.mishra@student.in', '9876512346', '2000-11-30', '2023-08-01', 1),

(8, 'Rishikesh', 'Tomar', 'rishikesh.tomar@student.in', '9876512347', '2002-02-14', '2023-08-01', 3),

(9, 'Sneha', 'Gupta', 'sneha.gupta@student.in', '9876512348', '2001-06-21', '2023-08-01', 5),

(10, 'Sneha', 'Tiwari', 'sneha.tiwari@student.in', '9876512349', '2001-04-28', '2023-08-01', 4);

CREATE TABLE Professors(

professor\_id INT Primary key,

first\_name varchar(100),

last\_name varchar(100),

email VARCHAR(100),

phone VARCHAR(20)

);

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

(1, 'Kunal', 'Kumar', 'kunal.kumar@university.in', '987653210'),

(2, 'Sandeep', 'Sharma', 'sandeep.sharma@university.in', '987654311'),

(3, 'Ruchi', 'Singh', 'ruchi.singh@university.in', '987643212'),

(4, 'Mira', 'Rajput', 'mira.rajput@university.in', '987643213'),

(5, 'Avinash', 'Patel', 'avinash.patel@university.in', '8676543214');

CREATE TABLE Courses(

Course\_id int PRIMARY key,

Course\_name varchar(100),

department\_id INT ,

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

professor\_id INT,

FOREIGN key (professor\_id ) REFERENCES Professors(professor\_id ),

credits INT

)

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

(1, 'Data Structures', 1, 1, 4),

(2, 'Digital Electronics', 2, 2, 3),

(3, 'Thermodynamics', 3, 3, 4),

(4, 'Structural Analysis', 4, 4, 4),

(5, 'Linear Algebra', 5, 5, 3),

(6, 'Algorithms', 1, 1, 4),

(7, 'VLSI Design', 2, 2, 3),

(8, 'Fluid Mechanics', 3, 3, 4),

(9, 'Transportation Engineering', 4, 4, 3),

(10, 'Abstract Algebra', 5, 5, 3);

CREATE TABLE Enrollments(

enrollment\_id INT Primary key,

student\_id INT ,

FOREIGN key (student\_id ) REFERENCES Students(student\_id ),

course\_id INT,

FOREIGN key (course\_id ) REFERENCES Courses(course\_id ),

enrollment\_date DATE ,

grade VARCHAR(5)

)

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

(1, 1, 1, '2023-08-05', 'A'),

(2, 2, 1, '2023-08-05', 'B'),

(3, 3, 2, '2023-08-06', 'A'),

(4, 4, 3, '2023-08-07', 'C'),

(5, 5, 2, '2023-08-07', 'B'),

(6, 6, 4, '2023-08-08', 'A'),

(7, 7, 6, '2023-08-09', 'B'),

(8, 8, 8, '2023-08-10', 'C'),

(9, 9, 9, '2023-08-11', 'A'),

(10, 10, 10, '2023-08-12', 'B'),

(11, 1, 6, '2023-08-13', 'A'),

(12, 2, 7, '2023-08-14', 'B'),

(13, 3, 8, '2023-08-15', 'A'),

(14, 4, 9, '2023-08-16', 'C'),

(15, 5, 10, '2023-08-17', 'B');

--------------------------------------------------------------------------------------------

**#SQL Queries for the Case Study#**

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

Ans=

SELECT d.department\_name,count(student\_id) as students from Departments as d LEFT JOIN Students as s on d.department\_id=s.department\_id GROUP by d.department\_id;

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

Ans=

SELECT course\_name from courses WHERE professor\_id in(SELECT professor\_id from professors where first\_name='Rajesh' && last\_name='Gupta')

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

Ans=

SELECT course\_name ,

avg (case

when grade="A" then 90

when grade="B" then 60

when grade="c" then 40

else null

end)as avg\_grade

FROM courses as c left join enrollments as e on c.Course\_id=e.course\_id GROUP by course\_name

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

Ans=

select first\_name ,last\_name from students as s left JOIN enrollments as e on s.student\_id=e.Student\_id WHERE e.student\_id is null

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

Ans=

SELECT d.department\_name, count(c.course\_id) as courses FROM departments as d left JOIN courses as 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')**

**Ans=**

SELECT student\_id FROM students WHERE department\_id in(SELECT department\_id from courses WHERE course\_name='Algorithms')

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

Ans=

SELECT course\_id

FROM (

SELECT course\_id, COUNT(\*) AS enrollment\_count

FROM Enrollments

GROUP BY course\_id

) AS SubQuery

ORDER BY enrollment\_count DESC

LIMIT 1;

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

Ans=

SELECT

d.department\_name,

AVG(c.credits) AS avg\_credits\_per\_student

FROM

students AS s

LEFT JOIN

departments AS d

ON

s.department\_id = d.department\_id

LEFT JOIN

courses AS c

ON

d.department\_id = c.department\_id

GROUP BY

d.department\_id, d.department\_name;

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

Ans=

SELECT first\_name ,last\_name from professors WHERE professor\_id in (select professor\_id FROM courses

GROUP by professor\_id HAVING count(department\_id)>1);

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

**Ans=**

SELECT max(case

when grade="A" then 90

when grade="B" then 60

when grade="c" then 40

else null

end)as highest\_Grade,

min(case

when grade="A" then 90

when grade="B" then 60

when grade="c" then 40

else null

end) as lowest\_Grade ,course\_name

FROM courses as c LEFT join enrollments as e on c.Course\_id=e.course\_id GROUP by course\_name HAVING course\_name = 'Data Structures'