Assignment 3:

-- Create Students table

CREATE TABLE Students (

student\_id INT PRIMARY KEY,

student\_name VARCHAR(100),

student\_major VARCHAR(100)

);

-- Create Courses table

CREATE TABLE Courses (

course\_id INT PRIMARY KEY,

course\_name VARCHAR(100),

course\_description VARCHAR(255)

);

-- Create Enrollments table

CREATE TABLE Enrollments (

enrollment\_id INT PRIMARY KEY,

student\_id INT,

course\_id INT,

enrollment\_date DATE,

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

FOREIGN KEY (course\_id) REFERENCES Courses(course\_id)

);

-- Insert data into Students table

INSERT INTO Students (student\_id, student\_name, student\_major) VALUES

(1, 'Alice', 'Computer Science'),

(2, 'Bob', 'Biology'),

(3, 'Charlie', 'History'),

(4, 'Diana', 'Mathematics');

-- Insert data into Courses table

INSERT INTO Courses (course\_id, course\_name, course\_description) VALUES

(101, 'Introduction to CS', 'Basics of Computer Science'),

(102, 'Biology Basics', 'Fundamentals of Biology'),

(103, 'World History', 'Historical events and cultures'),

(104, 'Calculus I', 'Introduction to Calculus'),

(105, 'Data Structures', 'Advanced topics in CS');

-- Insert data into Enrollments table

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

(1, 1, 101, '2023-01-15'),

(2, 2, 102, '2023-01-20'),

(3, 3, 103, '2023-02-01'),

(4, 1, 105, '2023-02-05'),

(5, 4, 104, '2023-02-10'),

(6, 2, 101, '2023-02-12'),

(7, 3, 105, '2023-02-15'),

(8, 4, 101, '2023-02-20'),

(9, 1, 104, '2023-03-01'),

(10, 2, 104, '2023-03-05');

**1. Inner Join:**

**Question:** Retrieve the list of students and their enrolled courses.

**2. Left Join:**

**Question:** List all students and their enrolled courses, including those who haven't enrolled in any course.

**3. Right Join:**

**Question:** Display all courses and the students enrolled in each course, including courses with no enrolled students.

**4. Self Join:**

**Question:** Find pairs of students who are enrolled in at least one common course.

**5. Complex Join:**

**Question:** Retrieve students who are enrolled in 'Introduction to CS' but not in 'Data Structures'.

Windows function:

**1. Using ROW\_NUMBER():**

**Question:** List all students along with a row number based on their enrollment date in ascending order.

**2. Using RANK():**

**Question:** Rank students based on the number of courses they are enrolled in, handling ties by assigning the same rank.

**3. Using DENSE\_RANK():**

**Question:** Determine the dense rank of courses based on their enrollment count across all students