-- Task 1: Academic Management System (using SQL)

-- Step 1: Database Creation

-- Create the Studentinfo table with columns STU\_ID, STU\_NAME, DOB, PHONE\_NO, EMAIL\_ID, ADDRESS

CREATE TABLE Studentinfo (

STU\_ID SERIAL PRIMARY KEY,

STU\_NAME VARCHAR(100),

DOB DATE,

PHONE\_NO VARCHAR(15),

EMAIL\_ID VARCHAR(100),

ADDRESS TEXT

);

-- Create the Coursesinfo table with columns COURSE\_ID, COURSE NAME, COURSE INSTRUCTOR NAME

CREATE TABLE Coursesinfo (

COURSE\_ID SERIAL PRIMARY KEY,

COURSE\_NAME VARCHAR(100),

COURSE\_INSTRUCTOR\_NAME VARCHAR(100)

);

-- Create the Enrollmentinfo table with columns ENROLLMENT\_ID, STU\_ID, COURSE\_ID, ENROLL\_STATUS (Enrolled/Not Enrolled)

CREATE TABLE Enrollmentinfo (

ENROLLMENT\_ID SERIAL PRIMARY KEY,

STU\_ID INT REFERENCES Studentinfo(STU\_ID),

COURSE\_ID INT REFERENCES Coursesinfo(COURSE\_ID),

ENROLL\_STATUS VARCHAR(15)

);

-- Step 2: Data Creation

-- Insert sample data for Studentinfo table

INSERT INTO Studentinfo (STU\_NAME, DOB, PHONE\_NO, EMAIL\_ID, ADDRESS)

VALUES

('Rajesh Kumar', '1998-05-12', '9876543210', 'rajesh@example.com', '123 Main St, Delhi'),

('Sneha Gupta', '2000-02-18', '7890123456', 'sneha@example.com', '456 Elm St, Mumbai');

-- Insert sample data for Coursesinfo table

INSERT INTO Coursesinfo (COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME)

VALUES

('Mathematics', 'Dr. Verma'),

('Physics', 'Prof. Singh');

-- Insert sample data for Enrollmentinfo table

INSERT INTO Enrollmentinfo (STU\_ID, COURSE\_ID, ENROLL\_STATUS)

VALUES

(1, 1, 'Enrolled'),

(1, 2, 'Enrolled'),

(2, 1, 'Enrolled'),

(2, 2, 'Not Enrolled');

-- Step 3: Retrieve Student Information

-- a) Retrieve student details, including name, contact information, and enrollment status

SELECT STU\_NAME, PHONE\_NO, ENROLL\_STATUS

FROM Studentinfo

JOIN Enrollmentinfo ON Studentinfo.STU\_ID = Enrollmentinfo.STU\_ID;

-- b) Retrieve a list of courses in which a specific student is enrolled

SELECT S.STU\_NAME, C.COURSE\_NAME, E.ENROLL\_STATUS

FROM Studentinfo AS S

JOIN Enrollmentinfo AS E ON S.STU\_ID = E.STU\_ID

JOIN Coursesinfo AS C ON E.COURSE\_ID = C.COURSE\_ID

WHERE S.STU\_NAME = 'Rajesh Kumar';

-- c) Retrieve course information, including course name and instructor information

SELECT COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

FROM Coursesinfo;

-- d) Retrieve course information for a specific course

SELECT COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

FROM Coursesinfo

WHERE COURSE\_NAME = 'Mathematics';

-- e) Retrieve course information for multiple courses

SELECT COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

FROM Coursesinfo

WHERE COURSE\_NAME IN ('Mathematics', 'Physics');

-- f) Test the queries to ensure accurate retrieval of student information.

-- Execute the queries and verify the results against the expected output.

-- Step 4: Reporting and Analytics (Using joining queries)

-- a) Retrieve the number of students enrolled in each course

SELECT C.COURSE\_NAME, COUNT(E.STU\_ID) AS NUM\_ENROLLED

FROM Coursesinfo AS C

LEFT JOIN Enrollmentinfo AS E ON C.COURSE\_ID = E.COURSE\_ID

GROUP BY C.COURSE\_NAME;

-- b) Retrieve the list of students enrolled in a specific course

SELECT S.STU\_NAME, E.ENROLL\_STATUS

FROM Studentinfo AS S

JOIN Enrollmentinfo AS E ON S.STU\_ID = E.STU\_ID

JOIN Coursesinfo AS C ON E.COURSE\_ID = C.COURSE\_ID

WHERE C.COURSE\_NAME = 'Mathematics';

-- c) Retrieve the count of enrolled students for each instructor

SELECT C.COURSE\_INSTRUCTOR\_NAME, COUNT(E.STU\_ID) AS NUM\_ENROLLED

FROM Coursesinfo AS C

LEFT JOIN Enrollmentinfo AS E ON C.COURSE\_ID = E.COURSE\_ID

GROUP BY C.COURSE\_INSTRUCTOR\_NAME;

-- d) Retrieve the list of students who are enrolled in multiple courses

SELECT S.STU\_NAME, COUNT(E.COURSE\_ID) AS NUM\_ENROLLED

FROM Studentinfo AS S

JOIN Enrollmentinfo AS E ON S.STU\_ID = E.STU\_ID

GROUP BY S.STU\_NAME;

-- e) Retrieve the courses with the highest number of enrolled students

SELECT C.COURSE\_NAME, COUNT(E.STU\_ID) AS NUM\_ENROLLED

FROM Coursesinfo AS C

LEFT JOIN Enrollmentinfo AS E ON C.COURSE\_ID = E.COURSE\_ID

GROUP BY C.COURSE\_NAME

ORDER BY NUM\_ENROLLED DESC;