CREATE TABLE StudentInfo (

STU\_ID INT PRIMARY KEY,

STU\_NAME VARCHAR(50),

DOB DATE,

PHONE\_NO VARCHAR(15),

EMAIL\_ID VARCHAR(100),

ADDRESS VARCHAR(200)

);

CREATE TABLE CoursesInfo (

COURSE\_ID INT PRIMARY KEY,

COURSE\_NAME VARCHAR(50),

COURSE\_INSTRUCTOR\_NAME VARCHAR(50)

);

CREATE TABLE Enrollmentinfo (

ENROLLMENT\_ID INT PRIMARY KEY,

STU\_ID INT,

COURSE\_ID INT,

ENROLL\_STATUS VARCHAR(20),

FOREIGN KEY (STU\_ID) REFERENCES StudentInfo(STU\_ID),

FOREIGN KEY (COURSE\_ID) REFERENCES CoursesInfo(COURSE\_ID)

);

-- Data Creation

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

VALUES

(1, 'Raj Laxman', '1995-05-15', '123-456-7890', 'raj@email.com', 'Pushpanjali'),

(2, 'Mohit Jain', '1996-08-20', '987-654-3210', 'mohit@email.com', 'Prernatirth');

INSERT INTO CoursesInfo (COURSE\_ID, COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME)

VALUES

(1, 'Maths', 'Prof. Balaji'),

(2, 'History', 'Prof. Cyril');

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

VALUES

(1, 1, 101, 'Enrolled'),

(2, 1, 102, 'Enrolled'),

(3, 2, 101, 'Enrolled');

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

SELECT SI.STU\_NAME, SI.PHONE\_NO, SI.EMAIL\_ID, EI.ENROLL\_STATUS

FROM StudentInfo SI

INNER JOIN Enrollmentinfo EI ON SI.STU\_ID = EI.STU\_ID;

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

SELECT CI.COURSE\_NAME

FROM CoursesInfo CI

INNER JOIN Enrollmentinfo EI ON CI.COURSE\_ID = EI.COURSE\_ID

WHERE EI.STU\_ID = 1;

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

SELECT CI.COURSE\_NAME, CI.COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo CI;

-- d) Retrieve course information for a specific course

SELECT CI.COURSE\_NAME, CI.COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo CI

WHERE CI.COURSE\_ID = 1;

-- e) Retrieve course information for multiple courses

SELECT CI.COURSE\_NAME, CI.COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo CI

WHERE CI.COURSE\_ID IN (1,2);

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

-- Reporting and Analytics (Using Joining Queries)

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

SELECT CI.COURSE\_NAME, COUNT(EI.STU\_ID) AS ENROLLED\_STUDENTS

FROM CoursesInfo CI

LEFT JOIN Enrollmentinfo EI ON CI.COURSE\_ID = EI.COURSE\_ID

GROUP BY CI.COURSE\_NAME;

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

SELECT SI.STU\_NAME

FROM StudentInfo SI

INNER JOIN Enrollmentinfo EI ON SI.STU\_ID = EI.STU\_ID

WHERE EI.COURSE\_ID = 1;

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

SELECT CI.COURSE\_INSTRUCTOR\_NAME, COUNT(EI.STU\_ID) AS ENROLLED\_STUDENTS

FROM CoursesInfo CI

LEFT JOIN Enrollmentinfo EI ON CI.COURSE\_ID = EI.COURSE\_ID

GROUP BY CI.COURSE\_INSTRUCTOR\_NAME;

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

SELECT SI.STU\_NAME

FROM StudentInfo SI

INNER JOIN Enrollmentinfo EI ON SI.STU\_ID = EI.STU\_ID

GROUP BY SI.STU\_ID

HAVING COUNT(EI.ENROLLMENT\_ID) > 1;

-- e) Retrieve the courses that have the highest number of enrolled students

SELECT CI.COURSE\_NAME, COUNT(EI.STU\_ID) AS ENROLLED\_STUDENTS

FROM CoursesInfo CI

LEFT JOIN Enrollmentinfo EI ON CI.COURSE\_ID = EI.COURSE\_ID

GROUP BY CI.COURSE\_NAME

ORDER BY ENROLLED\_STUDENTS DESC;

OUTPUT—





