

-- 1.Database Creation: a)Create the StudentInfo table with columns STU\_ID, STU\_NAME, DOB, PHONE\_NO, EMAIL\_ID,ADDRESS.

```
CREATE TABLE StudentInfo (
    STU_ID integer PRIMARY KEY,
    STU_NAME varchar(100),
    DOB DATE,
    PHONE_NO varchar(100),
    EMAIL_ID varchar(100),
    ADDRESS varchar(100)
);
```

-- b)Create the CoursesInfo table with columns COURSE\_ID, COURSE\_NAME, COURSE\_INSTRUCTOR NAME

```
CREATE TABLE CoursesInfo (
    COURSE_ID integer PRIMARY KEY,
    COURSE_NAME varchar(100),
    COURSE_INSTRUCTOR_NAME varchar(100)
);
```

-- c)Create the EnrollmentInfo with columns ENROLLMENT\_ID, STU\_ID, COURSE\_ID,

```
CREATE TABLE EnrollmentInfo (
    ENROLLMENT_ID VARCHAR(100) PRIMARY KEY,
    STU_ID INTEGER,
    COURSE_ID INTEGER,
    ENROLL_STATUS VARCHAR(50) CHECK (ENROLL_STATUS IN ('Enrolled', 'Not
Enrolled'))
);
```

-- ENROLL\_STATUS(Enrolled/Not Enrolled). The FOREIGN KEY constraint in the EnrollmentInfo table references the STU\_ID column in the StudentInfo table and the COURSE\_ID column in the CoursesInfo table.

```
ALTER TABLE EnrollmentInfo  
ADD CONSTRAINT fk_student  
FOREIGN KEY (STU_ID) REFERENCES StudentInfo(STU_ID);
```

```
ALTER TABLE EnrollmentInfo  
ADD CONSTRAINT fk_course  
FOREIGN KEY (COURSE_ID) REFERENCES CoursesInfo(COURSE_ID);
```

-- 2.Data Creation: Insert some sample data for StudentInfo table , CoursesInfo table, EnrollmentInfo with respective fields.

```
INSERT INTO StudentInfo VALUES  
(1, 'Clark', '1994-03-03', '9876543210', 'clark@infotech.com', 'california'),  
(2, 'Alen', '1995-05-13', '9807655410', 'alen@infotech.com', 'newyork'),  
(3, 'Dave', '1996-09-23', '9887543210', 'dave@infotech.com', 'texas');
```

```
INSERT INTO CoursesInfo VALUES  
(11, 'English', 'Mr. Smith'),  
(12, 'Accounts', 'Mr. Jones'),  
(13, 'Maths', 'Mr. William');
```

```
INSERT INTO EnrollmentInfo VALUES  
('EI01', 1, 11, 'Not Enrolled'),  
('EI02', 1, 11, 'Enrolled'),  
('EI03', 2, 12, 'Enrolled'),  
('EI04', 3, 13, 'Enrolled'),  
('EI05', 3, 12, 'Not Enrolled'),  
('EI06', 2, 13, 'Not Enrolled'),
```

('EI07', 1, 12, 'Enrolled');

-- 3) Retrieve the Student Information : (a) Write a query to retrieve student details, such as student name, contact informations, and Enrollment status.

SELECT

```
SI.STU_NAME AS StudentName,  
SI.PHONE_NO AS PhoneNumber,  
SI.EMAIL_ID AS EmailAddress,  
SI.ADDRESS AS Address,  
EI.ENROLL_STATUS AS EnrollmentStatus
```

FROM

StudentInfo SI

INNER JOIN

```
EnrollmentInfo EI ON SI.STU_ID = EI.STU_ID;
```

Output:

StudentName	PhoneNumber	EmailAddress	Address	EnrollmentStatus
Clark	9876543210	clark@infotech.com	california	Not Enrolled
Clark	9876543210	clark@infotech.com	california	Enrolled
Clark	9876543210	clark@infotech.com	california	Enrolled
Alen	9807655410	alen@infotech.com	newyork	Enrolled
Alen	9807655410	alen@infotech.com	newyork	Not Enrolled
Dave	9887543210	dave@infotech.com	texas	Enrolled
Dave	9887543210	dave@infotech.com	texas	Not Enrolled

-- (b) Write a query to retrieve a list of courses in which a specific student is enrolled.

SELECT

```
CI.COURSE_NAME AS CourseName,  
EI.ENROLL_STATUS AS EnrollmentStatus
```

FROM

EnrollmentInfo EI

INNER JOIN

```
CoursesInfo CI ON EI.COURSE_ID = CI.COURSE_ID
```

WHERE

EI.STU\_ID = 1

AND EI.ENROLL\_STATUS = 'Enrolled';

CourseName	EnrollmentStatus
English	Enrolled
Accounts	Enrolled

-- (c) Write a query to retrieve course information, including course name, instructor information.

SELECT

COURSE\_NAME AS CourseName,

COURSE\_INSTRUCTOR\_NAME AS CourseInstructorName

FROM

CoursesInfo;

CourseName	CourseInstructorName
English	Mr. Smith
Accounts	Mr. Jones
Maths	Mr. William

-- (d) Write a query to retrieve course information for a specific course .

SELECT

COURSE\_NAME AS CourseName,

COURSE\_INSTRUCTOR\_NAME AS CourseInstructorName

FROM

CoursesInfo

WHERE

COURSE\_ID = 11;

CourseName	CourseInstructorName
English	Mr. Smith

-- (e) Write a query to retrieve course information for multiple courses.

```

SELECT
    COURSE_ID AS CourseID,
    COURSE_NAME AS CourseName,
    COURSE_INSTRUCTOR_NAME AS CourseInstructorName
FROM
    CoursesInfo;

```

CourseID	CourseName	CourseInstructorName
11	English	Mr. Smith
12	Accounts	Mr. Jones
13	Maths	Mr. William

-- 4. Reporting and Analytics (Using joining queries) (a) Write a query to retrieve the number of students enrolled in each course

```

SELECT
    CI.COURSE_NAME AS CourseName,
    COUNT(EI.STU_ID) AS NumberOfStudentsEnrolled
FROM
    CoursesInfo CI
INNER JOIN
    EnrollmentInfo EI ON CI.COURSE_ID = EI.COURSE_ID
WHERE
    EI.ENROLL_STATUS = 'Enrolled'
GROUP BY
    CI.COURSE_NAME, CI.COURSE_ID;

```

CourseName	NumberOfStudentsEnrolled
English	1
Accounts	2
Maths	1

-- (b) Write a query to retrieve the list of students enrolled in a specific course

```

SELECT
    SI.STU_NAME AS StudentName,
    CI.COURSE_NAME AS Course
FROM
    StudentInfo SI
INNER JOIN
    EnrollmentInfo EI ON SI.STU_ID = EI.STU_ID
INNER JOIN
    CoursesInfo CI ON EI.COURSE_ID = CI.COURSE_ID
WHERE
    EI.COURSE_ID = 11
    AND EI.ENROLL_STATUS = 'Enrolled';

```

StudentName	Course
Clark	English

-- (c) Write a query to retrieve the count of enrolled students for each instructor.

```

SELECT
    CI.COURSE_INSTRUCTOR_NAME AS CourseInstructorName,
    COUNT(EI.STU_ID) AS NumberOfStudentsEnrolled

```

```
FROM
    CoursesInfo CI
INNER JOIN
    EnrollmentInfo EI ON CI.COURSE_ID = EI.COURSE_ID
WHERE
    EI.ENROLL_STATUS = 'Enrolled'
GROUP BY
    CI.COURSE_INSTRUCTOR_NAME, CI.COURSE_ID;
```

CourseInstructorName	NumberOfStudentsEnrolled
Mr. Smith	1
Mr. Jones	2
Mr. William	1

-- d) Write a query to retrieve the list of students who are enrolled in multiple courses

```
SELECT
    SI.STU_ID AS StudentID,
    SI.STU_NAME AS StudentName,
    COUNT(EI.ENROLLMENT_ID) AS NumberOfCoursesEnrolled
FROM
    StudentInfo SI
INNER JOIN
    EnrollmentInfo EI ON SI.STU_ID = EI.STU_ID
WHERE
    EI.ENROLL_STATUS = 'Enrolled'
GROUP BY
    SI.STU_ID, SI.STU_NAME
HAVING
    COUNT(EI.ENROLLMENT_ID) > 1;
```

```

+-----+-----+-----+
| StudentID | StudentName | NumberOfCoursesEnrolled |
+-----+-----+-----+
|       1 | Clark      |                 2 |
+-----+-----+-----+

```

-- (e) Write a query to retrieve the courses that have the highest number of enrolled students(arranging from highest to lowest)

```

SELECT
    CI.COURSE_NAME AS CourseName,
    COUNT(EI.STU_ID) AS NumberOfStudentsEnrolled
FROM
    CoursesInfo CI
INNER JOIN
    EnrollmentInfo EI ON CI.COURSE_ID = EI.COURSE_ID
WHERE
    EI.ENROLL_STATUS = 'Enrolled'
GROUP BY
    CI.COURSE_NAME, CI.COURSE_ID
ORDER BY
    NumberOfStudentsEnrolled DESC;

```

```

+-----+-----+
| CourseName | NumberOfStudentsEnrolled |
+-----+-----+
| Accounts   |                 2 |
| English    |                 1 |
| Maths      |                 1 |
+-----+-----+

```

-- end