Que2.Perform SQL queries to create tables and extract meaningful information from a Student and Course database

Ans. Here are some examples of SQL queries to create tables and extract meaningful information from a Student and Course database:

Creating Tables

-- Create Student table

CREATE TABLE Student (

StudentID INT PRIMARY KEY,

Name VARCHAR(255),

Email VARCHAR(255),

Major VARCHAR(255)

);

-- Create Course table

CREATE TABLE Course (

CourseID INT PRIMARY KEY,

CourseName VARCHAR(255),

Credits INT,

Department VARCHAR(255)

);

-- Create Enrollment table (many-to-many relationship)

CREATE TABLE Enrollment (

StudentID INT,

CourseID INT,

Grade VARCHAR(2),

PRIMARY KEY (StudentID, CourseID),

FOREIGN KEY (StudentID) REFERENCES Student(StudentID),

FOREIGN KEY (CourseID) REFERENCES Course(CourseID)

);

Inserting Sample Data

-- Insert students

INSERT INTO Student (StudentID, Name, Email, Major)

VALUES

(1, 'John Doe', 'johndoe@example.com', 'Computer Science'),

(2, 'Jane Smith', 'janesmith@example.com', 'Mathematics'),

(3, 'Bob Johnson', 'bobjohnson@example.com', 'Engineering');

-- Insert courses

INSERT INTO Course (CourseID, CourseName, Credits, Department)

VALUES

(1, 'Introduction to Programming', 3, 'Computer Science'),

(2, 'Calculus I', 4, 'Mathematics'),

(3, 'Physics I', 4, 'Engineering');

-- Insert enrollments

INSERT INTO Enrollment (StudentID, CourseID, Grade)

VALUES

(1, 1, 'A'),

(1, 2, 'B'),

(2, 2, 'A'),

(3, 3, 'A');

Extracting Meaningful Information

1. List all students enrolled in a specific course

SELECT s.Name, s.Email

FROM Student s

JOIN Enrollment e ON s.StudentID = e.StudentID

JOIN Course c ON e.CourseID = c.CourseID

WHERE c.CourseName = 'Introduction to Programming';

2. List all courses taken by a specific student

SELECT c.CourseName, c.Credits

FROM Course c

JOIN Enrollment e ON c.CourseID = e.CourseID

JOIN Student s ON e.StudentID = s.StudentID

WHERE s.Name = 'John Doe';

3. Calculate the GPA of a student

SELECT s.Name, AVG(

CASE e.Grade

WHEN 'A' THEN 4.0

WHEN 'B' THEN 3.0

WHEN 'C' THEN 2.0

WHEN 'D' THEN 1.0

WHEN 'F' THEN 0.0

END

) AS GPA

FROM Student s

JOIN Enrollment e ON s.StudentID = e.StudentID

WHERE s.Name = 'John Doe';

4. List the top 3 courses with the highest enrollment

SELECT c.CourseName, COUNT(e.StudentID) AS EnrollmentCount

FROM Course c

JOIN Enrollment e ON c.CourseID = e.CourseID

GROUP BY c.CourseName

ORDER BY EnrollmentCount DESC

LIMIT 3;

5. Find students who have taken all courses offered by a specific department

SELECT s.Name, s.Email

FROM Student s

WHERE s.StudentID IN (

SELECT e.StudentID

FROM Enrollment e

JOIN Course c ON e.CourseID = c.CourseID

WHERE c.Department = 'Computer Science'

GROUP BY e.StudentID

HAVING COUNT(DISTINCT c.CourseID) = (

SELECT COUNT(\*)

FROM Course

WHERE Department = 'Computer Science'

  )

);