CREATE TABLE Students (

StudentID INT PRIMARY KEY,

Name VARCHAR(50),

Age INT,

Major VARCHAR(50)

);

CREATE TABLE Courses (

CourseID INT PRIMARY KEY,

CourseName VARCHAR(50),

Credits INT

);

CREATE TABLE Enrollments (

EnrollmentID INT PRIMARY KEY,

StudentID INT,

CourseID INT,

Grade CHAR(2),

FOREIGN KEY (StudentID) REFERENCES Students(StudentID),

FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)

);

CREATE TABLE Departments (

DeptID INT PRIMARY KEY,

DeptName VARCHAR(50)

);

ALTER TABLE Students ADD Email VARCHAR(100);

DROP TABLE Departments;

INSERT INTO Students (StudentID, Name, Age, Major, Email) VALUES

(1, 'Alice', 20, 'Computer Science', 'alice@example.com'),

(2, 'Bob', 22, 'Data Science', 'bob@example.com'),

(3, 'Charlie', 19, 'AI', 'charlie@example.com'),

(4, 'David', 21, 'Computer Science', 'david@example.com'),

(5, 'Eva', 23, 'Data Science', 'eva@example.com');

INSERT INTO Courses VALUES

(101, 'DBMS', 4),

(102, 'AI', 3),

(103, 'Data Science', 4);

INSERT INTO Enrollments VALUES

(1001, 1, 101, 'A'),

(1002, 2, 102, 'B'),

(1003, 3, 103, 'A'),

(1004, 1, 102, 'C'),

(1005, 4, 103, 'B');

UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1;

DELETE FROM Students WHERE Age < 18;

SELECT Name, Major FROM Students WHERE Age > 19;

SELECT AVG(Age) AS AvgAge FROM Students;

SELECT Major, COUNT(\*) AS StudentCount

FROM Students

GROUP BY Major

HAVING COUNT(\*) > 1;

SELECT \* FROM Students WHERE Age > 20 AND Major = 'Computer Science';

SELECT StudentID, Grade,

RANK() OVER (ORDER BY Grade DESC) AS RankInClass

FROM Enrollments;

SELECT s.Name, c.CourseName

FROM Students s

INNER JOIN Enrollments e ON s.StudentID = e.StudentID

INNER JOIN Courses c ON e.CourseID = c.CourseID;

SELECT s.Name, c.CourseName

FROM Students s

LEFT JOIN Enrollments e ON s.StudentID = e.StudentID

LEFT JOIN Courses c ON e.CourseID = c.CourseID;

SELECT s.Name, c.CourseName FROM Students s CROSS JOIN Courses c;

SELECT s1.Name AS Student1, s2.Name AS Student2

FROM Students s1

JOIN Students s2

ON s1.Major = s2.Major

AND s1.StudentID <> s2.StudentID;

SELECT Name FROM Students

WHERE Age > (SELECT AVG(Age) FROM Students);

SELECT Name FROM Students s

WHERE EXISTS (

SELECT \* FROM Enrollments e

WHERE e.StudentID = s.StudentID AND e.Grade = 'A'

);

SELECT Name FROM Students

UNION

SELECT CourseName FROM Courses;

SELECT StudentID FROM Enrollments

INTERSECT

SELECT StudentID FROM Students;

SELECT StudentID FROM Students

EXCEPT

SELECT StudentID FROM Enrollments;

SELECT \* FROM Students LIMIT 3;

SELECT Name, Age FROM Students ORDER BY Age DESC, Name ASC;