Guided Practice

**-- Create the database**

CREATE DATABASE IF NOT EXISTS exam\_db;

USE exam\_db;

**-- Create the students table**

CREATE TABLE IF NOT EXISTS students (

student\_id INT PRIMARY KEY,

name VARCHAR(50),

age INT,

major VARCHAR(50)

);

**-- Create the courses table**

CREATE TABLE IF NOT EXISTS courses (

course\_id INT PRIMARY KEY,

course\_name VARCHAR(50),

credits INT

);

**-- Create the enrollments table**

CREATE TABLE IF NOT EXISTS enrollments (

enrollment\_id INT PRIMARY KEY,

student\_id INT,

course\_id INT,

FOREIGN KEY (student\_id) REFERENCES students(student\_id),

FOREIGN KEY (course\_id) REFERENCES courses(course\_id)

);

**-- Create the grades table**

CREATE TABLE IF NOT EXISTS grades (

grade\_id INT PRIMARY KEY,

enrollment\_id INT,

grade INT,

FOREIGN KEY (enrollment\_id) REFERENCES enrollments(enrollment\_id)

);

**-- Insert sample data into the tables**

INSERT INTO students VALUES (1, 'John', 20, 'Computer Science');

INSERT INTO students VALUES (2, 'Mary', 22, 'Mathematics');

INSERT INTO courses VALUES (1, 'Programming', 4);

INSERT INTO courses VALUES (2, 'Algebra', 3);

INSERT INTO enrollments VALUES (1, 1, 1);

INSERT INTO enrollments VALUES (2, 2, 2);

INSERT INTO grades VALUES (1, 1, 90);

INSERT INTO grades VALUES (2, 2, 85);

Diagrama

Descripción generada automáticamente

**-- Example queries for the exam**

**-- 1. Show all students majoring in Computer Science.**

SELECT \* FROM students WHERE major = 'Computer Science';

**-- 2. Get the course name and grade of John.**

SELECT c.course\_name, g.grade

FROM students s

INNER JOIN enrollments e ON s.student\_id = e.student\_id

INNER JOIN courses c ON e.course\_id = c.course\_id

INNER JOIN grades g ON e.enrollment\_id = g.enrollment\_id

WHERE s.name = 'John';

**-- 3. Show courses with more than 3 credits.**

SELECT \* FROM courses WHERE credits > 3;

**--4. Show the name and age of students who are older than 21.**

SELECT name, age FROM students WHERE age > 21;

**--5. Get the names of courses with no enrollments.**

SELECT course\_name FROM courses

WHERE course\_id NOT IN (SELECT course\_id FROM enrollments);

**--6. Show students who have scored above 80.**

SELECT s.name FROM students s

INNER JOIN enrollments e ON s.student\_id = e.student\_id

INNER JOIN grades g ON e.enrollment\_id = g.enrollment\_id

WHERE g.grade > 80;

**--7 Get the average grades of all students.**

SELECT AVG(grade) AS average\_grades FROM grades;

**--8 Get the names and credits of courses John is enrolled in.**

SELECT c.course\_name, c.credits

FROM students s

INNER JOIN enrollments e ON s.student\_id = e.student\_id

INNER JOIN courses c ON e.course\_id = c.course\_id

WHERE s.name = 'Juan';

**--9 Show students enrolled in more than one course.**

SELECT s.name, COUNT(e.enrollment\_id) AS course\_count

FROM students s

INNER JOIN enrollments e ON s.student\_id = e.student\_id

GROUP BY s.name

HAVING COUNT(e.enrollment\_id) > 1;

**--10. Show courses in which Mary is enrolled.**

SELECT c.course\_name

FROM students s

INNER JOIN enrollments e ON s.student\_id = e.student\_id

INNER JOIN courses c ON e.course\_id = c.course\_id

WHERE s.name = 'Maria';

**--11. Get the total number of credits for all courses.**

SELECT SUM(credits) AS total\_credits FROM courses;

**--12. Show courses with no grades assigned.**

SELECT c.course\_name

FROM courses c

LEFT JOIN enrollments e ON c.course\_id = e.course\_id

LEFT JOIN grades g ON e.enrollment\_id = g.enrollment\_id

WHERE g.enrollment\_id IS NULL;

**--13. Get the names and ages of students enrolled in the Programming course.**

SELECT s.name, s.age

FROM students s

INNER JOIN enrollments e ON s.student\_id = e.student\_id

INNER JOIN courses c ON e.course\_id = c.course\_id

WHERE c.course\_name = 'Programming';

**--14 Show the average grade for each course.**

SELECT c.course\_name, AVG(g.grade) AS average\_grade

FROM courses c

LEFT JOIN enrollments e ON c.course\_id = e.course\_id

LEFT JOIN grades g ON e.enrollment\_id = g.enrollment\_id

GROUP BY c.course\_name;

**--15 Get the names and majors of students enrolled in any course.**

SELECT s.name, s.major

FROM students s

INNER JOIN enrollments e ON s.student\_id = e.student\_id;