

SQL Test Questions

Aaria Raj IMAECO (2437002)

-- Create the Students table

```
CREATE TABLE Students (  
    student_id INT PRIMARY KEY,  
    name VARCHAR(50),  
    age INT,  
    grade_level INT  
);
```

-- Create the Courses table

```
CREATE TABLE Courses (  
    course_id INT PRIMARY KEY,  
    course_name VARCHAR(50),  
    credits INT  
);
```

-- Create the Enrollments table

```
CREATE TABLE Enrollments (  
    enrollment_id INT PRIMARY KEY,  
    student_id INT,  
    course_id INT,  
    grade DECIMAL(3, 2),  
    FOREIGN KEY (student_id) REFERENCES Students(student_id),  
    FOREIGN KEY (course_id) REFERENCES Courses(course_id)
```

```
);
```

```
-- Insert sample data into Students
```

```
INSERT INTO Students (student_id, name, age, grade_level) VALUES
```

```
(1, 'Alice', 20, 3),
```

```
(2, 'Bob', 21, 4),
```

```
(3, 'Charlie', 19, 3),
```

```
(4, 'Diana', 22, 4),
```

```
(5, 'Ethan', 20, 3);
```

```
-- Insert sample data into Courses
```

```
INSERT INTO Courses (course_id, course_name, credits) VALUES
```

```
(1, 'Math', 3),
```

```
(2, 'Science', 4),
```

```
(3, 'History', 3),
```

```
(4, 'Art', 2);
```

```
-- Insert sample data into Enrollments
```

```
INSERT INTO Enrollments (enrollment_id, student_id, course_id, grade) VALUES
```

```
(1, 1, 1, 3.5),
```

```
(2, 1, 2, 4.0),
```

```
(3, 2, 1, 2.5),
```

```
(4, 2, 3, 3.0),
```

```
(5, 3, 2, 3.0),
```

```
(6, 3, 4, 2.0),
```

(7, 4, 1, 3.5),
(8, 4, 3, 4.0),
(9, 5, 2, 3.0),
(10, 5, 4, 3.5);

Q1. SELECT AVG(grade) AS average_grade

FROM Enrollments;

average_grade
3.200000

Q2. SELECT Students.name, Courses.course_name

FROM Enrollments

JOIN Students ON Enrollments.student_id = Students.student_id

JOIN Courses ON Enrollments.course_id = Courses.course_id;

name	course_name
Alice	Math
Alice	Science
Bob	Math
Bob	History

Charlie	Science
Charlie	Art
Diana	Math
Diana	History
Ethan	Science
Ethan	Art

Q3. SELECT grade_level, COUNT(*) AS number_of_students

FROM Students

GROUP BY grade_level;

grade_level	number_of_students
3	3
4	2

Q4. SELECT Courses.course_name, MAX(Enrollments.grade) AS max_grade

FROM Enrollments

JOIN Courses ON Enrollments.course_id = Courses.course_id

GROUP BY Courses.course_name;

course_name	max_grade
Art	3.50
History	4.00
Math	3.50
Science	4.00

Q5. SELECT AVG(Enrollments.grade) AS average_grade

FROM Enrollments

JOIN Students ON Enrollments.student_id = Students.student_id

WHERE Students.grade_level = 3;

average_grade
3.166666

Q6. SELECT Students.name, Courses.course_name, Courses.credits

FROM Enrollments

JOIN Students ON Enrollments.student_id = Students.student_id

JOIN Courses ON Enrollments.course_id = Courses.course_id;

name	course_name	credits
Alice	Math	3
Alice	Science	4
Bob	Math	3
Bob	History	3

Charlie	Science	4
Charlie	Art	2
Diana	Math	3
Diana	History	3
Ethan	Science	4
Ethan	Art	2

Q7. SELECT Courses.course_name, AVG(Enrollments.grade) AS average_grade

FROM Enrollments

JOIN Courses ON Enrollments.course_id = Courses.course_id

GROUP BY Courses.course_name

HAVING AVG(Enrollments.grade) > 3.0;

course_name	average_grade
History	3.500000
Math	3.166666
Science	3.333333

Q8. SELECT DISTINCT Students.name

```
FROM Students
WHERE Students.student_id NOT IN (
SELECT student_id
FROM Enrollments
WHERE grade = 4.0
);
```

name
Bob
Charlie
Ethan

```
Q9. WITH AverageGrades AS (
SELECT student_id, AVG(grade) AS student_avg_grade
FROM Enrollments
GROUP BY student_id
),
OverallAverage AS (
SELECT AVG(grade) AS overall_avg_grade
FROM Enrollments
)
SELECT Students.name
FROM AverageGrades
JOIN Students ON AverageGrades.student_id = Students.student_id
WHERE AverageGrades.student_avg_grade > (SELECT overall_avg_grade FROM
OverallAverage);
```

name
Alice
Diana
Ethan

Q10. SELECT Students.name, COUNT(Enrollments.course_id) AS total_courses,
 AVG(Enrollments.grade) AS average_grade
 FROM Enrollments
 JOIN Students ON Enrollments.student_id = Students.student_id
 GROUP BY Students.name;

name	total_courses	average_grade
Alice	2	3.750000
Bob	2	2.750000
Charlie	2	2.500000
Diana	2	3.750000
Ethan	2	3.250000