

Name: Ashfaaq Feroz Muhammad

Id: 2023ebcs005

Topic: SGA 1- Database Creation and Data Manipulation for an Education System

3:

```
ALTER TABLE Enrollments ADD COLUMN Grade DECIMAL(4,2);
```

A screenshot of a SQL editor interface with a dark theme. At the top, there is a toolbar with a dropdown menu set to 'SQL', navigation arrows, a line counter '1 / 1', a status indicator '1 - 0 of 0', and icons for table view, undo, redo, and a search icon. Below the toolbar, the SQL command 'ALTER TABLE Enrollments ADD COLUMN Grade DECIMAL(4,2);' is entered in a text area.

]

3-1

```
UPDATE Students
```

```
SET Email = 'aarav.khan@newmail.com',
```

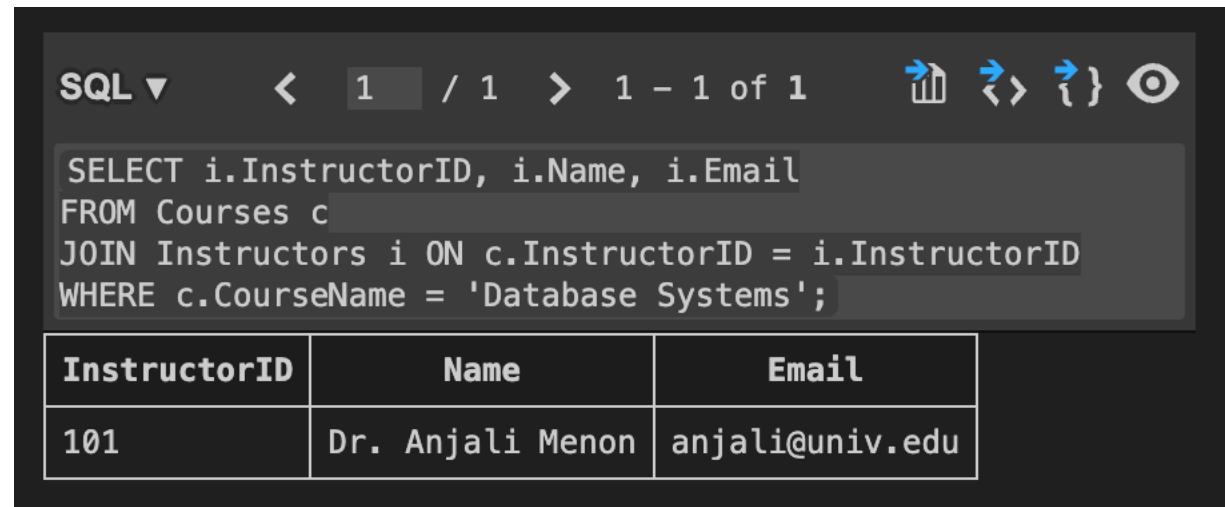
```
    DepartmentID = 3
```

```
WHERE StudentID = 1001;
```

A screenshot of a SQL editor interface with a dark theme. The toolbar is identical to the one in the previous block. The text area contains the following SQL command: 'UPDATE Students', 'SET Email = 'aarav.khan@newmail.com'', 'DepartmentID = 3', and 'WHERE StudentID = 1001;'. Each line of the command is on a new line.

3-2

```
SELECT i.InstructorID, i.Name, i.Email  
FROM Courses c  
JOIN Instructors i ON c.InstructorID = i.InstructorID  
WHERE c.CourseName = 'Database Systems';
```

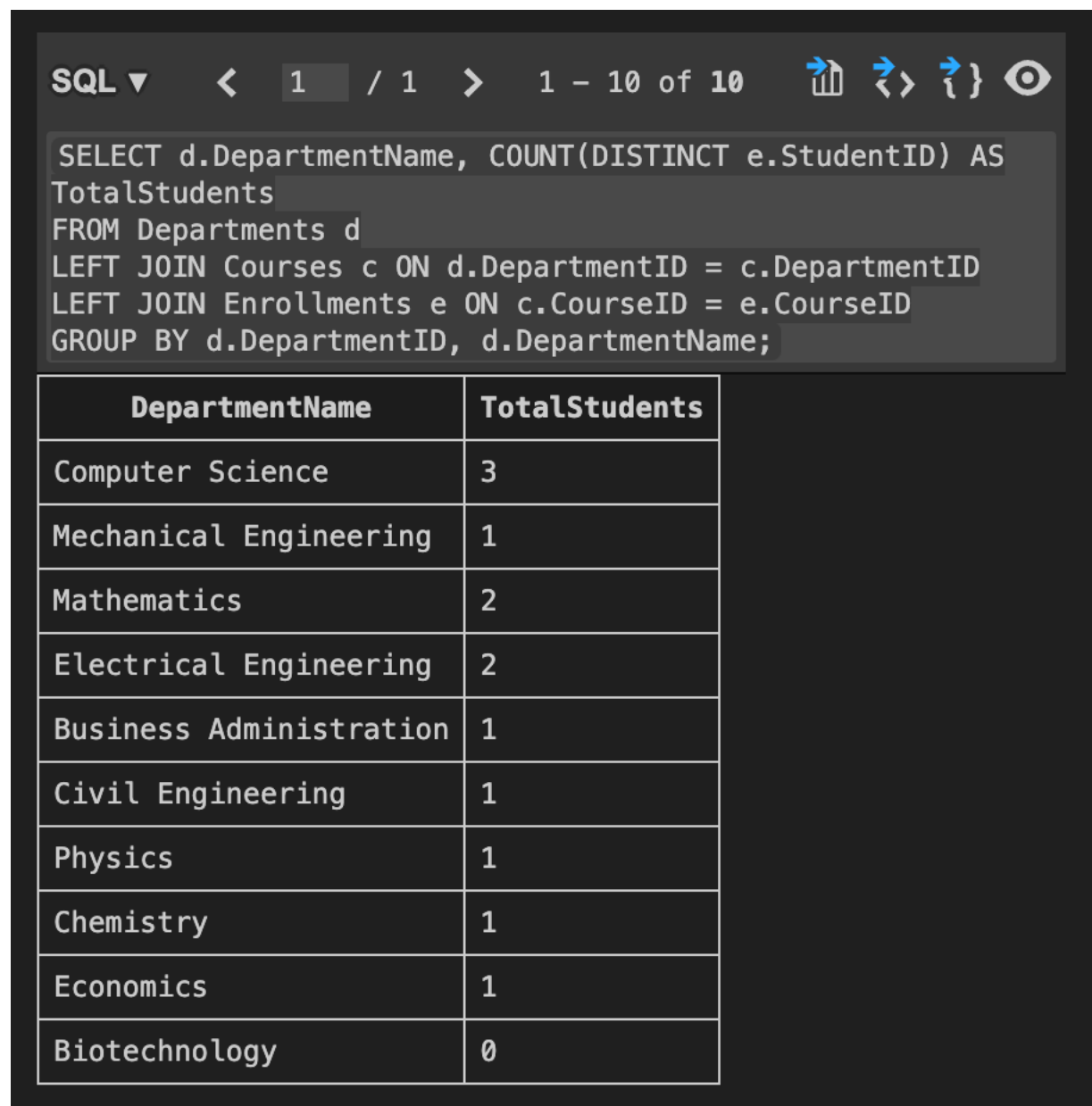


The screenshot shows a SQL IDE interface. At the top, there is a toolbar with icons for SQL, navigation, and execution. Below the toolbar, the SQL query is displayed in a text area. The query is: `SELECT i.InstructorID, i.Name, i.Email FROM Courses c JOIN Instructors i ON c.InstructorID = i.InstructorID WHERE c.CourseName = 'Database Systems';`. Below the query, the result is shown as a table with three columns: **InstructorID**, **Name**, and **Email**. The table contains one row of data: **101**, **Dr. Anjali Menon**, and **anjali@univ.edu**.

| InstructorID | Name             | Email           |
|--------------|------------------|-----------------|
| 101          | Dr. Anjali Menon | anjali@univ.edu |

3-3

```
SELECT d.DepartmentName, COUNT(DISTINCT e.StudentID) AS TotalStudents
FROM Departments d
LEFT JOIN Courses c ON d.DepartmentID = c.DepartmentID
LEFT JOIN Enrollments e ON c.CourseID = e.CourseID
GROUP BY d.DepartmentID, d.DepartmentName;
```

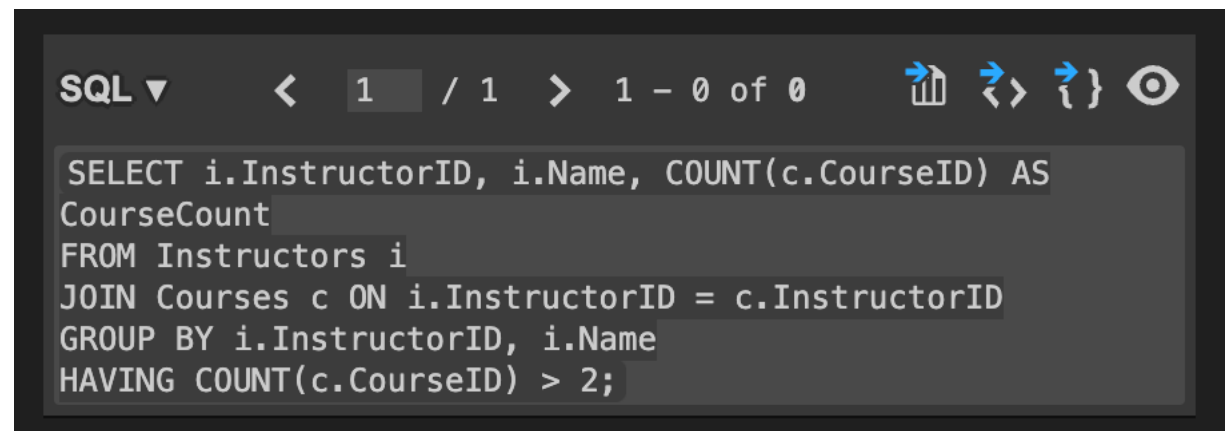


The screenshot shows a SQL IDE interface. At the top, there is a toolbar with icons for SQL, navigation, and execution. Below the toolbar, the SQL query is displayed in a text area. Below the query, the results are shown in a table with two columns: DepartmentName and TotalStudents. The table contains 10 rows of data, including departments like Computer Science, Mechanical Engineering, Mathematics, Electrical Engineering, Business Administration, Civil Engineering, Physics, Chemistry, Economics, and Biotechnology.

| DepartmentName          | TotalStudents |
|-------------------------|---------------|
| Computer Science        | 3             |
| Mechanical Engineering  | 1             |
| Mathematics             | 2             |
| Electrical Engineering  | 2             |
| Business Administration | 1             |
| Civil Engineering       | 1             |
| Physics                 | 1             |
| Chemistry               | 1             |
| Economics               | 1             |
| Biotechnology           | 0             |

3-4

```
SELECT i.InstructorID, i.Name, COUNT(c.CourseID) AS CourseCount
FROM Instructors i
JOIN Courses c ON i.InstructorID = c.InstructorID
GROUP BY i.InstructorID, i.Name
HAVING COUNT(c.CourseID) > 2;
```

A screenshot of a SQL IDE interface. The top bar shows 'SQL' with a dropdown arrow, navigation icons (back, forward, search), and a status bar indicating '1 / 1' and '1 - 0 of 0'. The main editor area contains the same SQL query as the previous block, with syntax highlighting: keywords in blue, identifiers in white, and string literals in red. The query is: 

```
SELECT i.InstructorID, i.Name, COUNT(c.CourseID) AS
CourseCount
FROM Instructors i
JOIN Courses c ON i.InstructorID = c.InstructorID
GROUP BY i.InstructorID, i.Name
HAVING COUNT(c.CourseID) > 2;
```

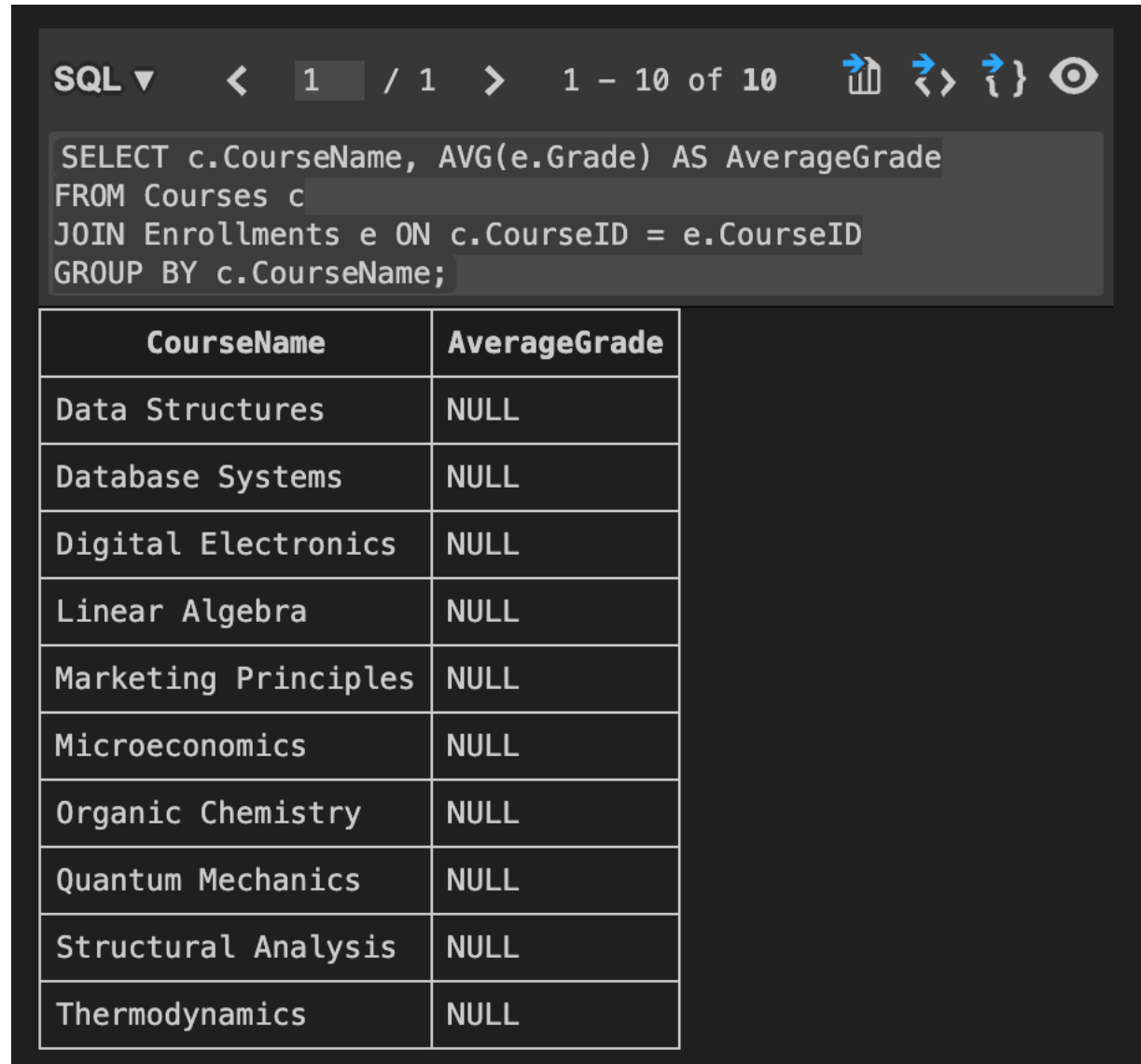
3-5

```
SELECT c.CourseName, AVG(e.Grade) AS AverageGrade
```

```
FROM Courses c
```

```
JOIN Enrollments e ON c.CourseID = e.CourseID
```

```
GROUP BY c.CourseName;
```

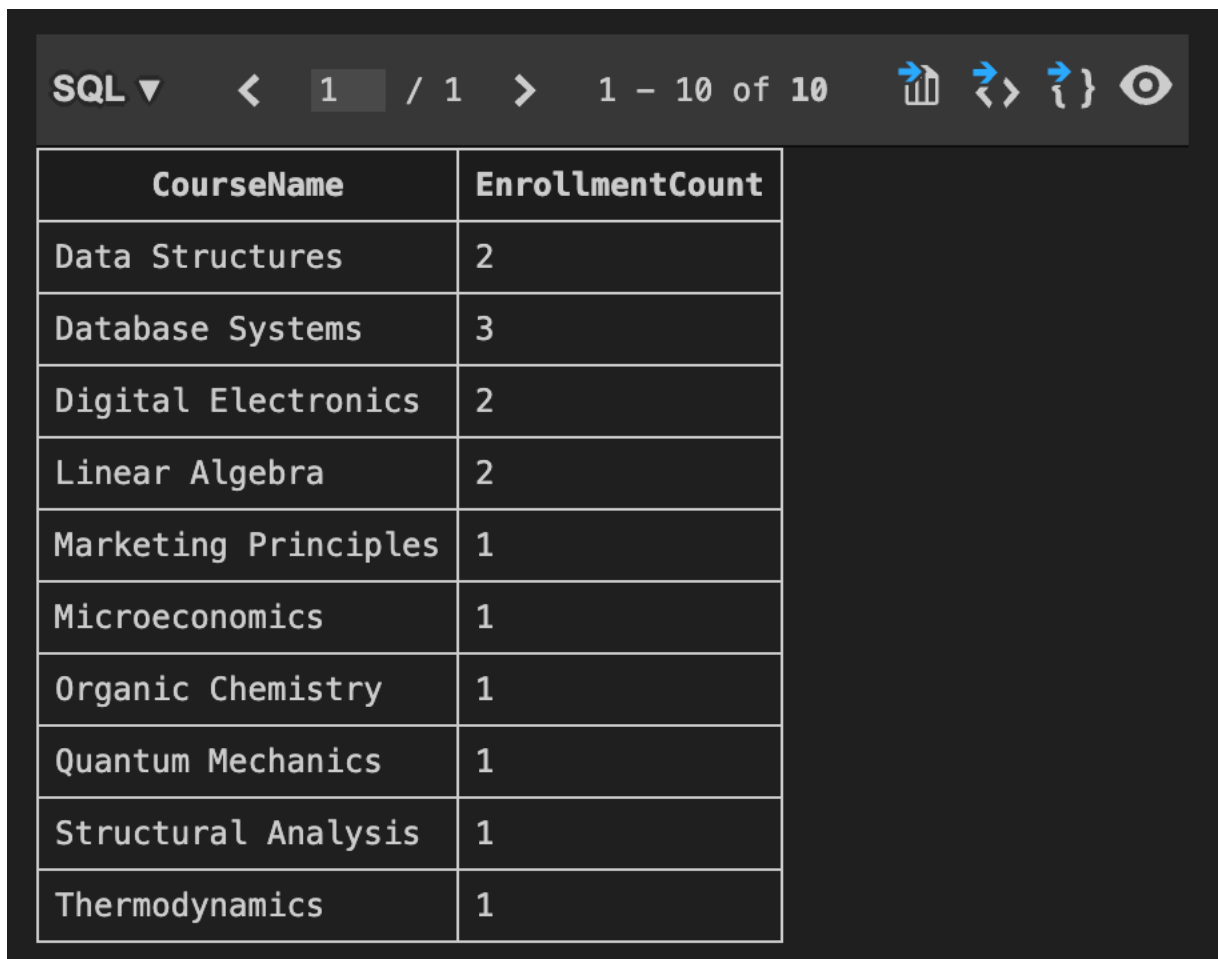


The screenshot shows a SQL IDE interface with a dark theme. At the top, there is a toolbar with icons for SQL, navigation, and execution. Below the toolbar, the SQL query is displayed in a text area. Below the query, the results are shown in a table with two columns: CourseName and AverageGrade. The table contains 11 rows of data, all with NULL values in the AverageGrade column.

| CourseName           | AverageGrade |
|----------------------|--------------|
| Data Structures      | NULL         |
| Database Systems     | NULL         |
| Digital Electronics  | NULL         |
| Linear Algebra       | NULL         |
| Marketing Principles | NULL         |
| Microeconomics       | NULL         |
| Organic Chemistry    | NULL         |
| Quantum Mechanics    | NULL         |
| Structural Analysis  | NULL         |
| Thermodynamics       | NULL         |

3-6

```
SELECT c.CourseName, COUNT(e.EnrollmentID) AS EnrollmentCount
FROM Courses c
LEFT JOIN Enrollments e ON c.CourseID = e.CourseID
GROUP BY c.CourseName;
```



| CourseName           | EnrollmentCount |
|----------------------|-----------------|
| Data Structures      | 2               |
| Database Systems     | 3               |
| Digital Electronics  | 2               |
| Linear Algebra       | 2               |
| Marketing Principles | 1               |
| Microeconomics       | 1               |
| Organic Chemistry    | 1               |
| Quantum Mechanics    | 1               |
| Structural Analysis  | 1               |
| Thermodynamics       | 1               |