

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 MySQL command-line interface. The title bar says "MySQL - 5.7.26". The main window has a dark background with white text. On the left, there's a yellow-bordered "SQL" dropdown menu. In the center, the SQL query "ALTER TABLE Enrollments ADD COLUMN Grade DECIMAL(4,2);" is displayed in a text input field. Below the input field, the status bar shows "1 / 1" and "1 - 0 of 0". To the right of the status bar are several icons: a magnifying glass, a double arrow, a question mark, and a refresh symbol.

]

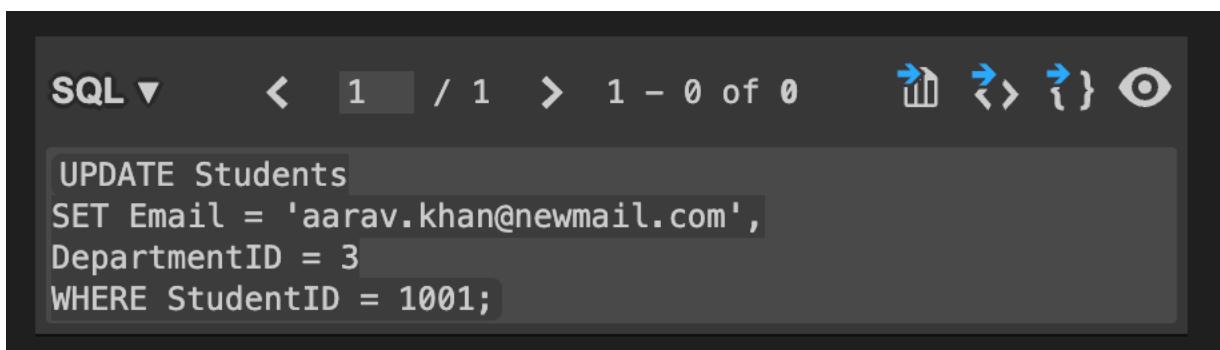
3-1

```
UPDATE Students
```

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

```
DepartmentID = 3
```

```
WHERE StudentID = 1001;
```



A screenshot of a MySQL command-line interface. The title bar says "MySQL - 5.7.26". The main window has a dark background with white text. On the left, there's a yellow-bordered "SQL" dropdown menu. In the center, the SQL query "UPDATE Students SET Email = 'aarav.khan@newmail.com', DepartmentID = 3 WHERE StudentID = 1001;" is displayed in a text input field. Below the input field, the status bar shows "1 / 1" and "1 - 0 of 0". To the right of the status bar are several icons: a magnifying glass, a double arrow, a question mark, and a refresh symbol.

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 query execution interface. At the top, there are navigation buttons for SQL, back, forward, and search, along with a page number indicator '1 / 1' and a total count '1 - 1 of 1'. Below the interface, the SQL query is displayed:

```
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 results are presented in a table:

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;
```

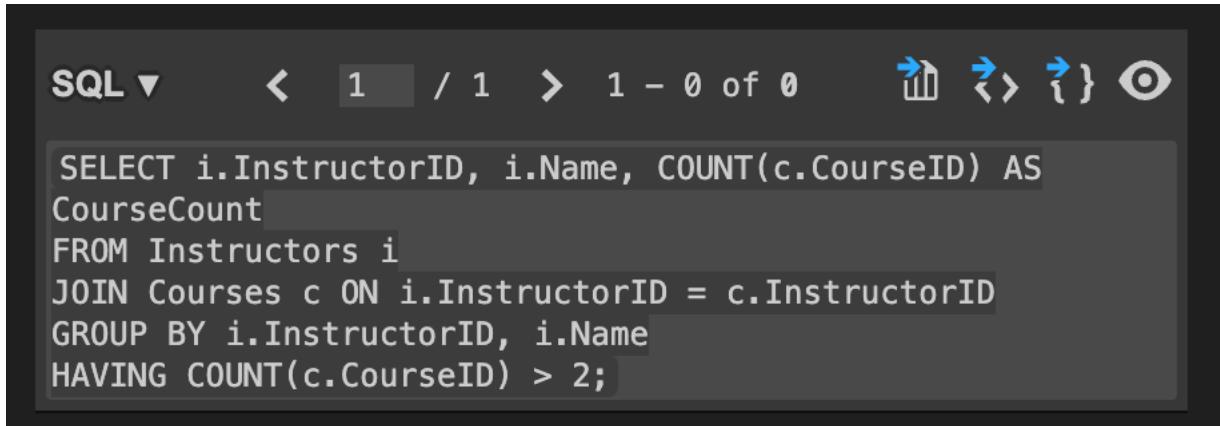
The screenshot shows a SQL query execution interface with the following details:

- SQL ▼**: A dropdown menu for SQL-related options.
- Navigation**: Buttons for navigating between pages (1 - 10 of 10), zooming in and out, and switching between tabs.
- Query Text**: The SQL query itself, identical to the one shown above.
- Results Table**: A table displaying the results of the query:

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 editor interface. The top bar shows "SQL ▼" and navigation icons for back, forward, and search. Below the bar, the current page is indicated as "1 / 1" and the total number of rows is "1 - 0 of 0". The main area contains the SQL query code:

```
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;
```

SQL ▼ < 1 / 1 > 1 - 10 of 10 ⚡ ⚡ ⚡ ⚡ ⚡

```
SELECT c.CourseName, AVG(e.Grade) AS AverageGrade  
FROM Courses c  
JOIN Enrollments e ON c.CourseID = e.CourseID  
GROUP BY c.CourseName;
```

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;
```

The screenshot shows a SQL query results interface with a dark theme. At the top, there is a toolbar with various icons and a status bar indicating "SQL ▼" and "1 / 1". Below the toolbar is a table with two columns: "CourseName" and "EnrollmentCount". The table contains ten rows of data, each representing a course and its enrollment count. The data is as follows:

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