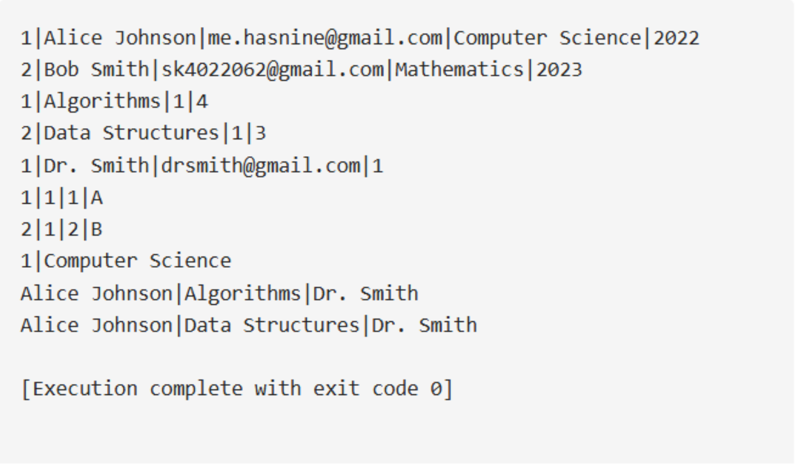
Question 1: Joining Tables

```
-- create a table
CREATE TABLE students (
  id INTEGER PRIMARY KEY,
  name TEXT NOT NULL,
  email VARCHAR(50),
  major INT,
  enrollment_year INTEGER
);
-- create a table
CREATE TABLE Courses (
  id INTEGER PRIMARY KEY,
  name TEXT NOT NULL,
  department_id INT,
  credits INT
);
-- create a table
CREATE TABLE Instructors (
  id INTEGER PRIMARY KEY,
  name TEXT NOT NULL,
  email VARCHAR(50),
  department_id INT
);
-- create a table
CREATE TABLE Enrollments (
  id INTEGER PRIMARY KEY,
  student_id INT,
  course_id INT,
  grade
);
-- create a table
CREATE TABLE Departments (
  id INTEGER PRIMARY KEY,
  name TEXT NOT NULL
);
-- insert some values
```

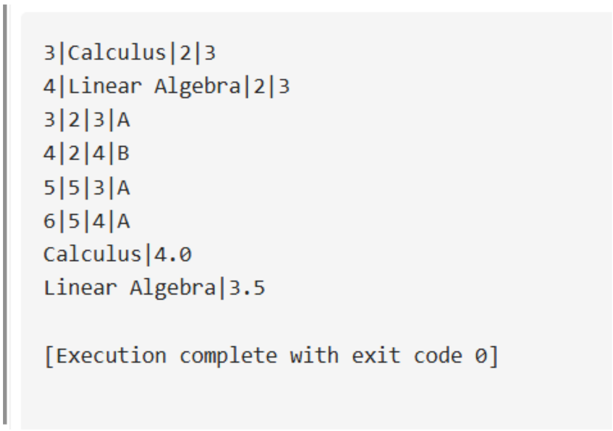
```
INSERT INTO students VALUES (1, 'Alice Johnson', 'me.hasnine@gmail.com', 'Computer Science', 2022);
INSERT INTO students VALUES (2, 'Bob Smith', 'sk4022062@gmail.com', 'Mathematics', 2023);
-- insert some values
INSERT INTO Courses VALUES (1, 'Algorithms', 1, 4);
INSERT INTO Courses VALUES (2, 'Data Structures', 1, 3);
-- insert some values
INSERT INTO Instructors VALUES (1, 'Dr. Smith', 'drsmith@gmail.com', 1);
-- insert some values
INSERT INTO Enrollments VALUES (1, 1, 1, 'A');
INSERT INTO Enrollments VALUES (2, 1, 2, 'B');
-- insert some values
INSERT INTO Departments VALUES (1, 'Computer Science');
-- fetch some values
SELECT * FROM students;
SELECT * FROM Courses;
SELECT * FROM Instructors;
SELECT * FROM Enrollments;
SELECT * FROM Departments;
--QUERY
SELECT
    s.name AS student_name,
    c.name AS course_name,
    i.name AS instructor_name
FROM
    Enrollments e
JOIN
    Students s ON e.student_id = s.id
JOIN
    Courses c ON e.course_id = c.id
JOIN
    Instructors i ON c.department_id = i.department_id
JOIN
   Departments d ON i.department_id = d.id
WHERE
    d.name = 'Computer Science';
```



```
1|Alice Johnson|me.hasnine@gmail.com|Computer Science|2022
2|Bob Smith|sk4022062@gmail.com|Mathematics|2023
1|Algorithms|1|4
2|Data Structures|1|3
1|Dr. Smith|drsmith@gmail.com|1
1|1|1|A
2|1|2|B
1|Computer Science
Alice Johnson|Algorithms|Dr. Smith
Alice Johnson|Data Structures|Dr. Smith

[Execution complete with exit code 0]
```

Question 2: Aggregations

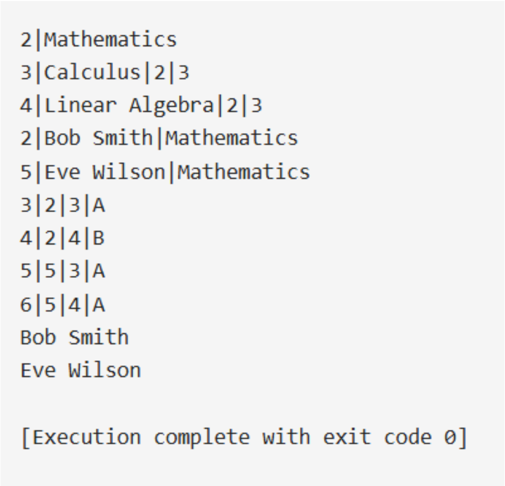
```
-- create a table
CREATE TABLE Courses (
  id INTEGER PRIMARY KEY,
  name TEXT NOT NULL,
  department_id INT,
  credits INT
);
-- create a table
CREATE TABLE Enrollments (
  id INTEGER PRIMARY KEY,
  student_id INT,
  course_id INT,
  grade INT
);

-- insert some values
INSERT INTO Courses VALUES (3, 'Calculus', 2, 3);
INSERT INTO Courses VALUES (4, 'Linear Algebra', 2, 3);
-- insert some values
INSERT INTO Enrollments VALUES (3, 2, 3, 'A');
INSERT INTO Enrollments VALUES (4, 2, 4, 'B');
INSERT INTO Enrollments VALUES (5, 5, 3, 'A');
INSERT INTO Enrollments VALUES (6, 5, 4, 'A');
-- fetch some values
SELECT * FROM Courses ;
SELECT * FROM Enrollments;
--QUERY
SELECT
    c.name AS course_name,
    AVG(
        CASE
            WHEN e.grade = 'A' THEN 4
            WHEN e.grade = 'B' THEN 3
            WHEN e.grade = 'C' THEN 2
            WHEN e.grade = 'D' THEN 1
            WHEN e.grade = 'F' THEN 0
            ELSE NULL
        END
    ) AS average_grade
FROM
    Enrollments e
JOIN
    Courses c ON e.course_id = c.id
GROUP BY
    c.name;
```



```
3|Calculus|2|3
4|Linear Algebra|2|3
3|2|3|A
4|2|4|B
5|5|3|A
6|5|4|A
Calculus|4.0
Linear Algebra|3.5

[Execution complete with exit code 0]
```

Question 3: Filtering and Grouping

```
-- create a table
CREATE TABLE Departments (
  id INTEGER PRIMARY KEY,
  name TEXT NOT NULL
);
-- create a table
CREATE TABLE Courses (
  id INTEGER PRIMARY KEY,
  name TEXT NOT NULL,
  department_id INT,
  credits INT
);
-- create a table
CREATE TABLE Students (
  id INTEGER PRIMARY KEY,
  name TEXT NOT NULL,
  major INT
);
-- create a table
CREATE TABLE Enrollments (
  id INTEGER PRIMARY KEY,
  student_id INT,
  course_id INT,
  grade INT
);
-- insert some values
INSERT INTO Departments VALUES (2, 'Mathematics');
-- insert some values
INSERT INTO Courses VALUES (3, 'Calculus', 2, 3);
INSERT INTO Courses VALUES (4, 'Linear Algebra', 2, 3);
-- insert some values
INSERT INTO Students VALUES (2, 'Bob Smith', 'Mathematics');
INSERT INTO Students VALUES (5, 'Eve Wilson', 'Mathematics');
-- insert some values
INSERT INTO Enrollments VALUES (3, 2, 3, 'A');
INSERT INTO Enrollments VALUES (4, 2, 4, 'B');
INSERT INTO Enrollments VALUES (5, 5, 3, 'A');
INSERT INTO Enrollments VALUES (6, 5, 4, 'A');
-- fetch some values
SELECT * FROM Departments;
SELECT * FROM Courses;
SELECT * FROM Students;
SELECT * FROM Enrollments;
--QUERY
SELECT
    s.name AS student_name
FROM
    Enrollments e
JOIN
    Courses c ON e.course_id = c.id
JOIN
    Students s ON e.student_id = s.id
JOIN
    Departments d ON c.department_id = d.id
WHERE
    d.name = 'Mathematics'
GROUP BY
    s.id, s.name
```



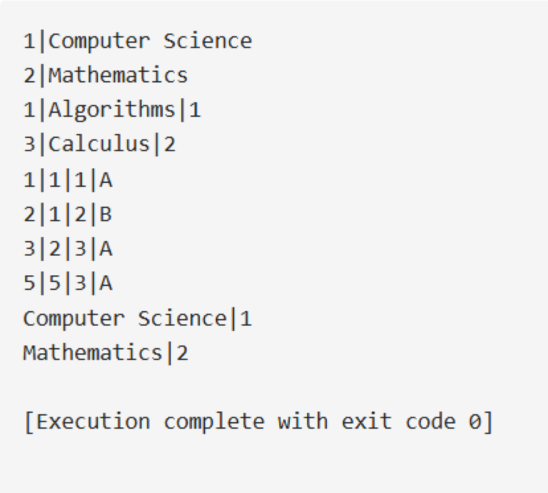
```
2|Mathematics
3|Calculus|2|3
4|Linear Algebra|2|3
2|Bob Smith|Mathematics
5|Eve Wilson|Mathematics
3|2|3|A
4|2|4|B
5|5|3|A
6|5|4|A
Bob Smith
Eve Wilson

[Execution complete with exit code 0]
```

Question 4: Grouped Totals

```
-- create a table
```

```
CREATE TABLE Departments (
  id INTEGER PRIMARY KEY,
  name TEXT NOT NULL
);
-- create a table
CREATE TABLE Courses (
  id INTEGER PRIMARY KEY,
  name TEXT NOT NULL,
  department_id INT
);
-- create a table
CREATE TABLE Enrollments (
  id INTEGER PRIMARY KEY,
  student_id INT,
  course_id INT,
  grade INT
);
-- insert some values
INSERT INTO Departments VALUES (1, 'Computer Science');
INSERT INTO Departments VALUES (2, 'Mathematics');
-- insert some values
INSERT INTO Courses  VALUES (1, 'Algorithms', 1);
INSERT INTO Courses  VALUES (3, 'Calculus', 2);
-- insert some values
INSERT INTO Enrollments  VALUES (1, 1, 1, 'A');
INSERT INTO Enrollments  VALUES (2, 1, 2, 'B');
INSERT INTO Enrollments  VALUES (3, 2, 3, 'A');
INSERT INTO Enrollments  VALUES (5, 5, 3, 'A');
-- fetch some values
SELECT * FROM Departments;
SELECT * FROM Courses;
SELECT * FROM Enrollments;
--QUERY
SELECT
```

```
    d.name AS department_name,

    COUNT(DISTINCT e.student_id) AS
total_students

FROM

    Departments d

JOIN

    Courses c ON d.id = c.department_id

JOIN

    Enrollments e ON c.id = e.course_id

GROUP BY

    d.name;
```

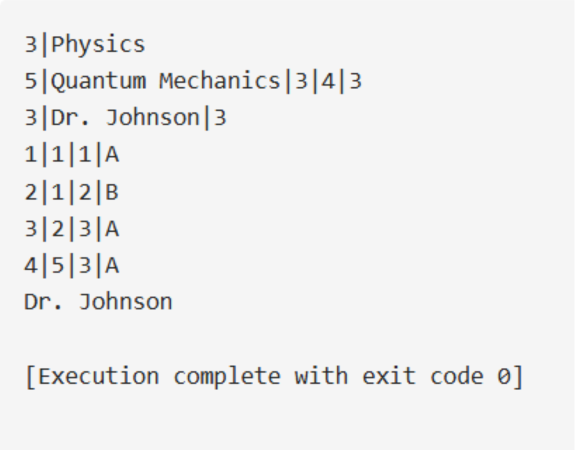


```
1|Computer Science
2|Mathematics
1|Algorithms|1
3|Calculus|2
1|1|1|A
2|1|2|B
3|2|3|A
5|5|3|A
Computer Science|1
Mathematics|2

[Execution complete with exit code 0]
```

Question 5: Identifying Instructors with No Enrollments

```
-- create a table

CREATE TABLE Departments (

  id INTEGER PRIMARY KEY,

  name TEXT NOT NULL

);

CREATE TABLE Courses (

  id INTEGER PRIMARY KEY,

  name TEXT NOT NULL,

  department_id INT,

  credits INT,

  instructor_id INT -- Add this column to link courses with instructors

);

-- create a table

CREATE TABLE Instructors (

  id INTEGER PRIMARY KEY,

  name TEXT NOT NULL,

   department_id INT

);

-- create a table

CREATE TABLE Enrollments (

  id INTEGER PRIMARY KEY,

  student_id INT,

  course_id INT,

  grade INT

);
```

```
-- insert some values

INSERT INTO Departments VALUES (3, 'Physics');

-- insert some values

INSERT INTO Courses VALUES (5, 'Quantum Mechanics', 3, 4, 3);

-- insert some values

INSERT INTO Instructors VALUES (3, 'Dr. Johnson', 3);

-- insert some values

INSERT INTO Enrollments  VALUES (1, 1, 1, 'A');

INSERT INTO Enrollments  VALUES (2, 1, 2, 'B');

INSERT INTO Enrollments  VALUES (3, 2, 3, 'A');

INSERT INTO Enrollments  VALUES (4, 5, 3, 'A');

-- fetch some values

SELECT * FROM Departments;

SELECT * FROM Courses;

SELECT * FROM Instructors;

SELECT * FROM Enrollments;

--QUERY

SELECT

    i.name AS instructor_name

FROM

    Instructors i

JOIN

    Courses c ON i.id = c.instructor_id

LEFT JOIN

    Enrollments e ON c.id = e.course_id

WHERE

    e.id IS NULL;
```



```
3|Physics
5|Quantum Mechanics|3|4|3
3|Dr. Johnson|3
1|1|1|A
2|1|2|B
3|2|3|A
4|5|3|A
Dr. Johnson

[Execution complete with exit code 0]
```