1. Table Structure

Students Table

```
sql
CREATE TABLE Students (
    StudentID INT PRIMARY KEY,
    StudentName VARCHAR(100) NOT NULL,
    DateOfBirth DATE,
    TeacherID INT
);
```

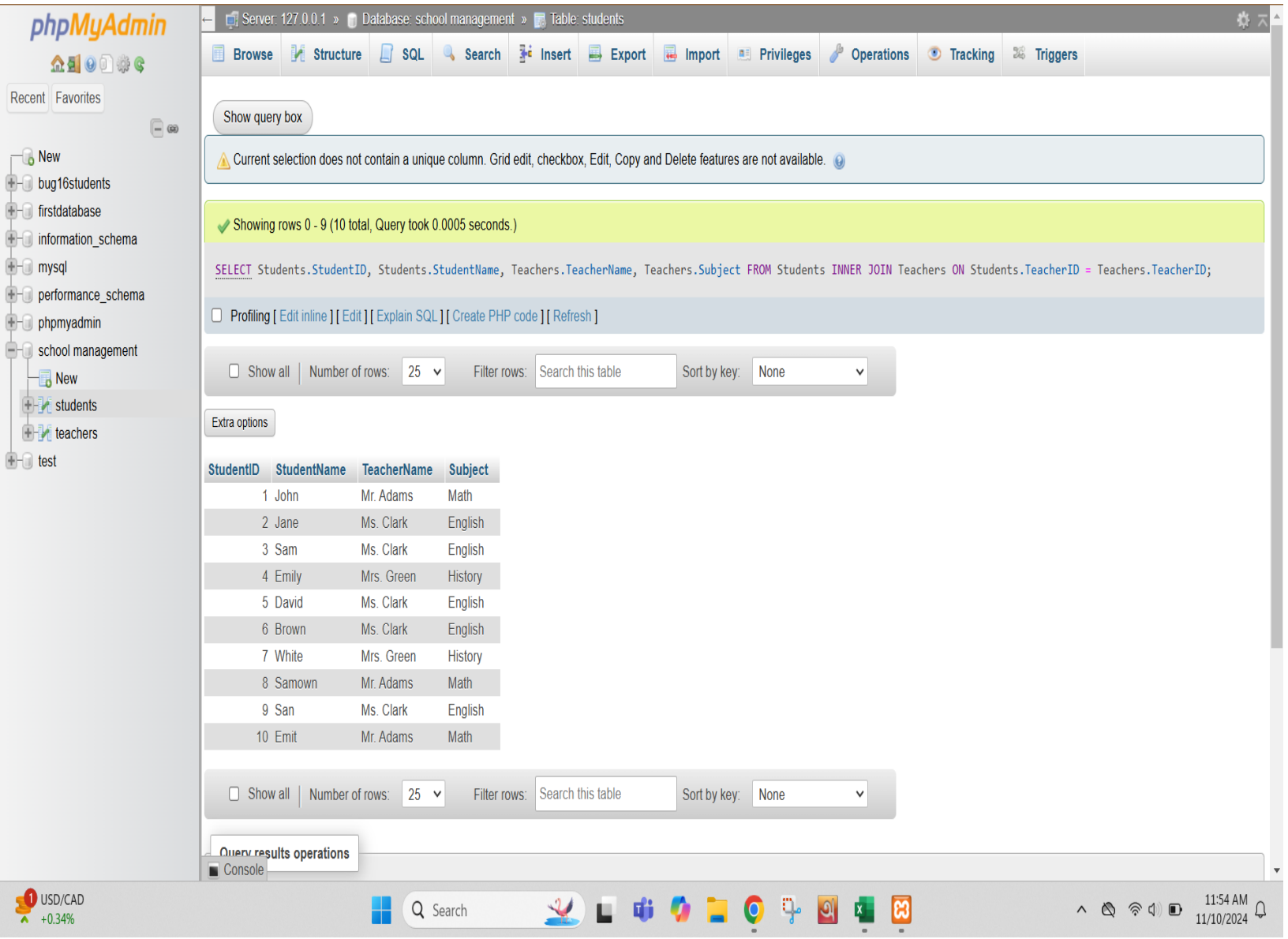
Teachers Table

```
CREATE TABLE Teachers (
    TeacherID INT PRIMARY KEY,
    TeacherName VARCHAR(100) NOT NULL,
    Subject VARCHAR(100) NOT NULL
);
```

2. Types of SQL Joins

a. INNER JOIN

The INNER JOIN returns only the rows where there is a match between the Students and Teachers tables based on the TeacherID.

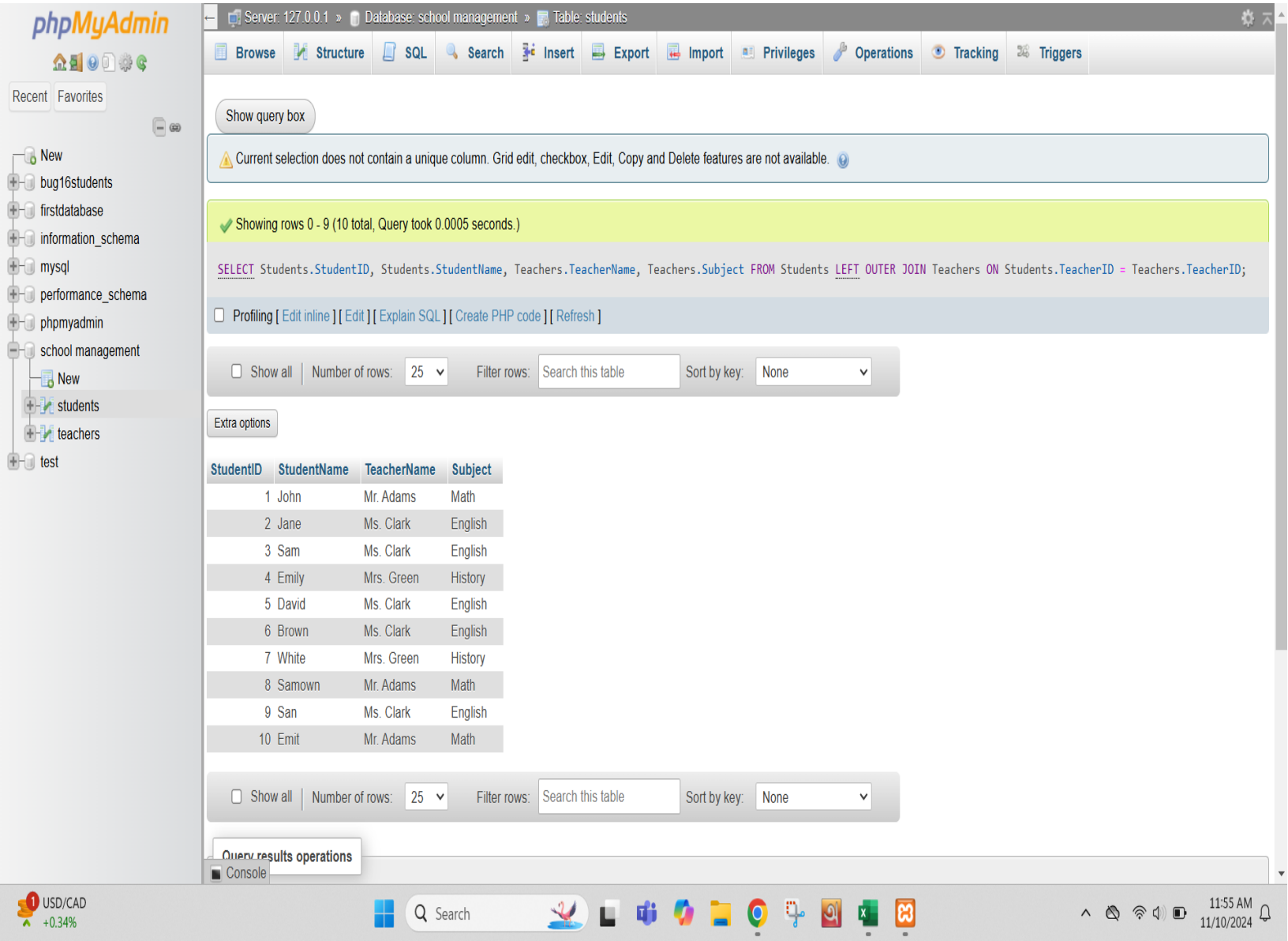


Explanation:

- This query retrieves students who are associated with a teacher (i.e., TeacherID matches in both tables).
- If a student doesn't have a TeacherID, they will not be included in the result.

b. LEFT (OUTER) JOIN

The LEFT JOIN (also known as LEFT OUTER JOIN) returns all rows from the Students table and the matched rows from the Teachers table. If no match is found, NULL values are returned for the Teachers columns.

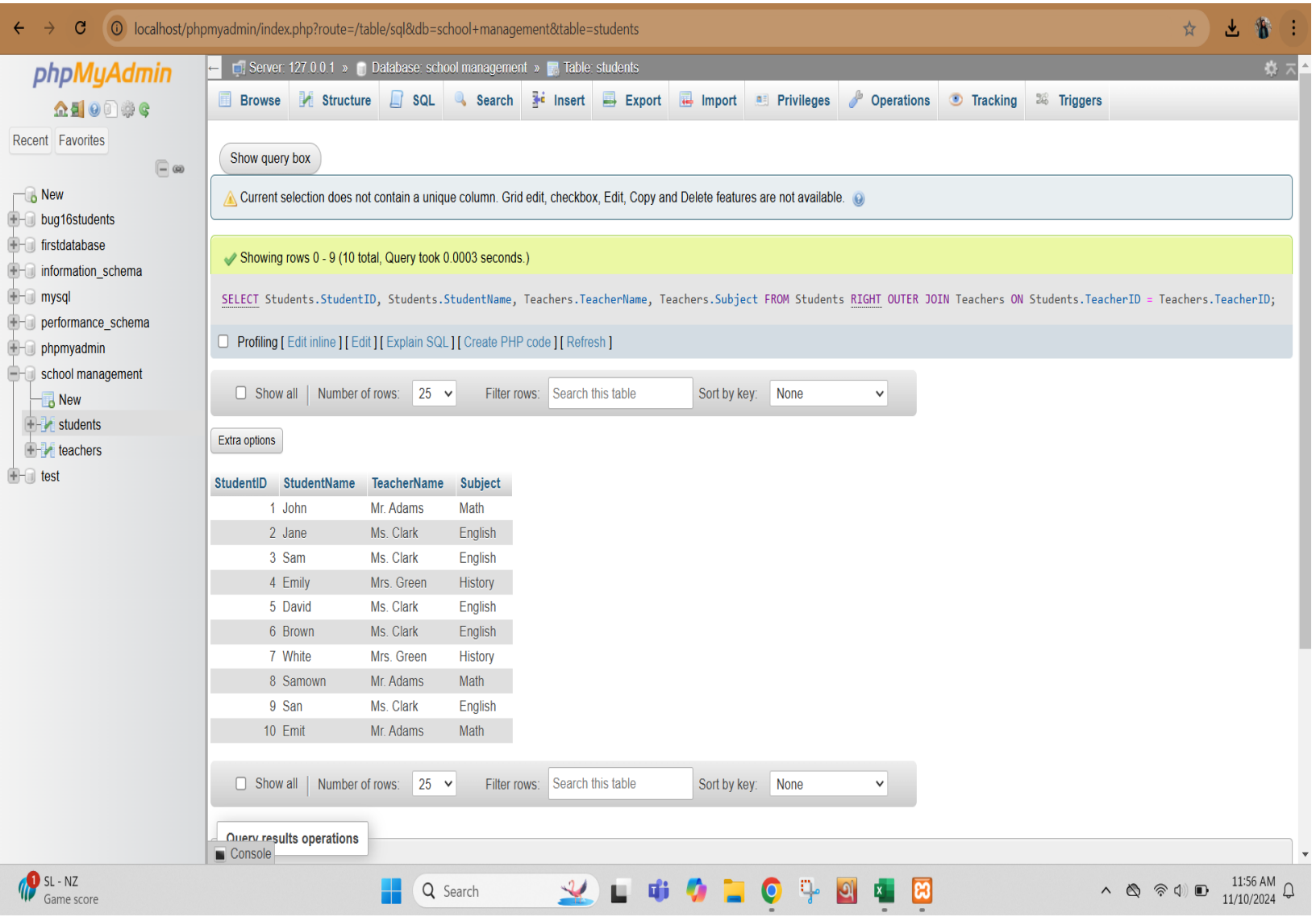


Explanation:

- This query retrieves all students, regardless of whether they have a matching teacher.
- If a student does not have a TeacherID, NULL is returned for the teacher-related columns.

c. RIGHT (OUTER) JOIN

The RIGHT JOIN (also known as RIGHT OUTER JOIN) returns all rows from the Teachers table and the matched rows from the Students table. If no match is found, NULL values are returned for the Students columns.

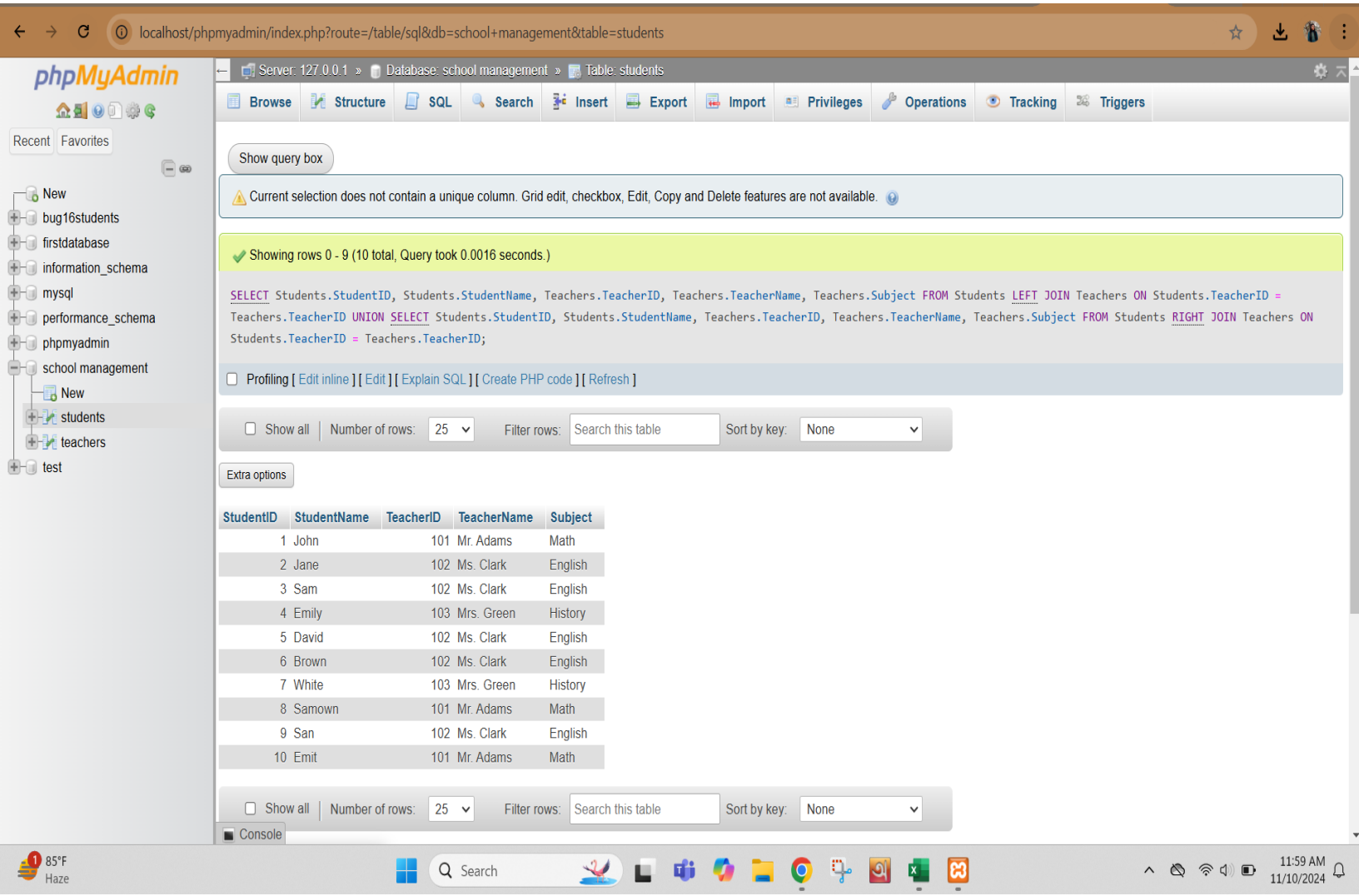


Explanation:

- This query retrieves all teachers, regardless of whether they are associated with any students.
- If a teacher has no students (i.e., no matching TeacherID in the Students table), NULL is returned for the student-related columns.

d. FULL OUTER JOIN (Using UNION)

MySQL does not support FULL OUTER JOIN directly. However, you can simulate a FULL OUTER JOIN by combining a LEFT JOIN and a RIGHT JOIN using UNION.



Explanation:

- **First Query (LEFT JOIN):** Returns all students and their associated teachers, or NULL for teachers if no match is found.
- **Second Query (RIGHT JOIN):** Returns all teachers and their associated students, or NULL for students if no match is found.
- The UNION combines the results of both queries and eliminates any duplicates, ensuring that all records from both tables are returned.
- This effectively simulates the behavior of a FULL OUTER JOIN in MySQL.

4. Testing the Queries

To test the queries:

1. Ensure you have populated the Students and Teachers tables with sample data.
2. Run each query individually to see the results for different types of joins.

```
INSERT INTO Students (StudentID, StudentName, DateOfBirth, TeacherID)
VALUES
(1, 'John', '2005-03-01', 101),
(2, 'Jane', '2004-07-10', 102),
(3, 'Sam', '2005-11-20', 102),
(4, 'Emily', '2005-05-15', 103),
(5, 'David', '2004-09-30', 102),
(6, 'Brown', '2005-11-25', 102),
(7, 'White', '2005-05-18', 103),
(8, 'Samown', '2005-11-22', 101),
(9, 'San', '2005-11-21', 102),
(10, 'Emit', '2005-05-13', 101);
```

```
INSERT INTO Teachers (TeacherID, TeacherName, Subject)
VALUES
(101, 'Mr. Adams', 'Math'),
(102, 'Ms. Clark', 'English'),
(103, 'Mrs. Green', 'History');
```