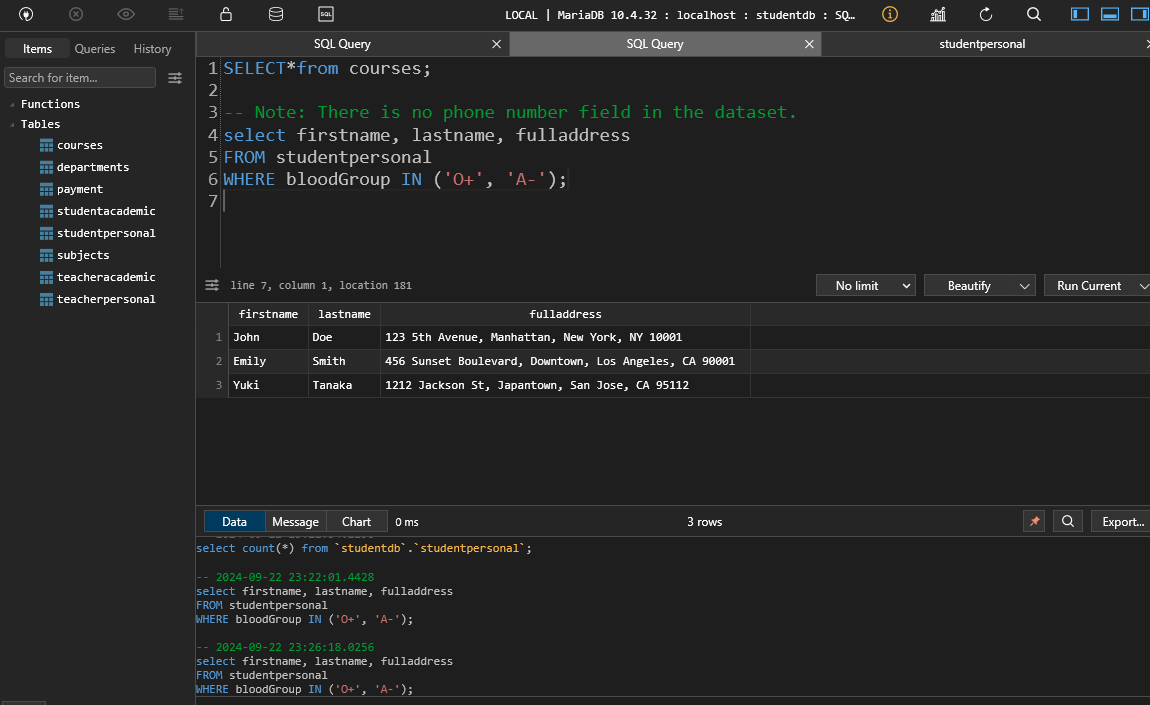
**1. Write a query to find the students' first name, last name, phone number, and address for those who have the blood groups O+ and A-**

**-- Note: There is no phone number field in the dataset.**

**select firstname, lastname, fulladdress**

**FROM studentpersonal**

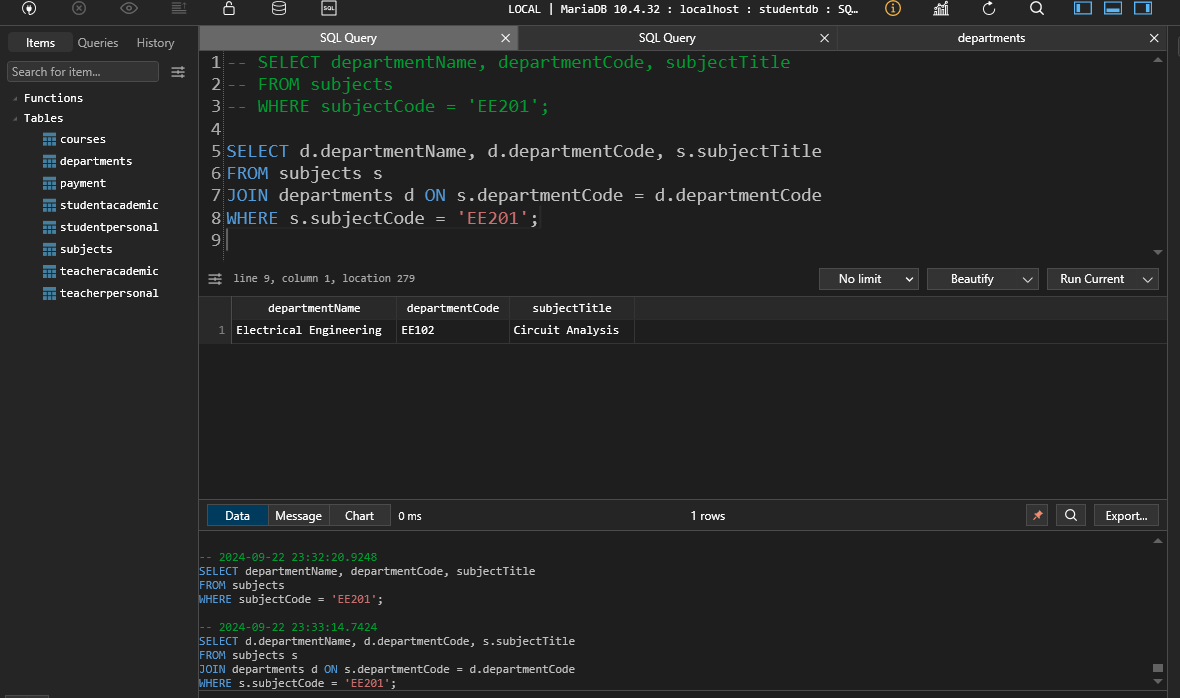
**WHERE bloodGroup IN ('O+', 'A-');**

**2. Write a query to find the department name, department code, and subject title for the subject with the subject code "EE201**

**SELECT d.departmentName, d.departmentCode, s.subjectTitle**

**FROM subjects s**

**JOIN departments d ON s.departmentCode = d.departmentCode**

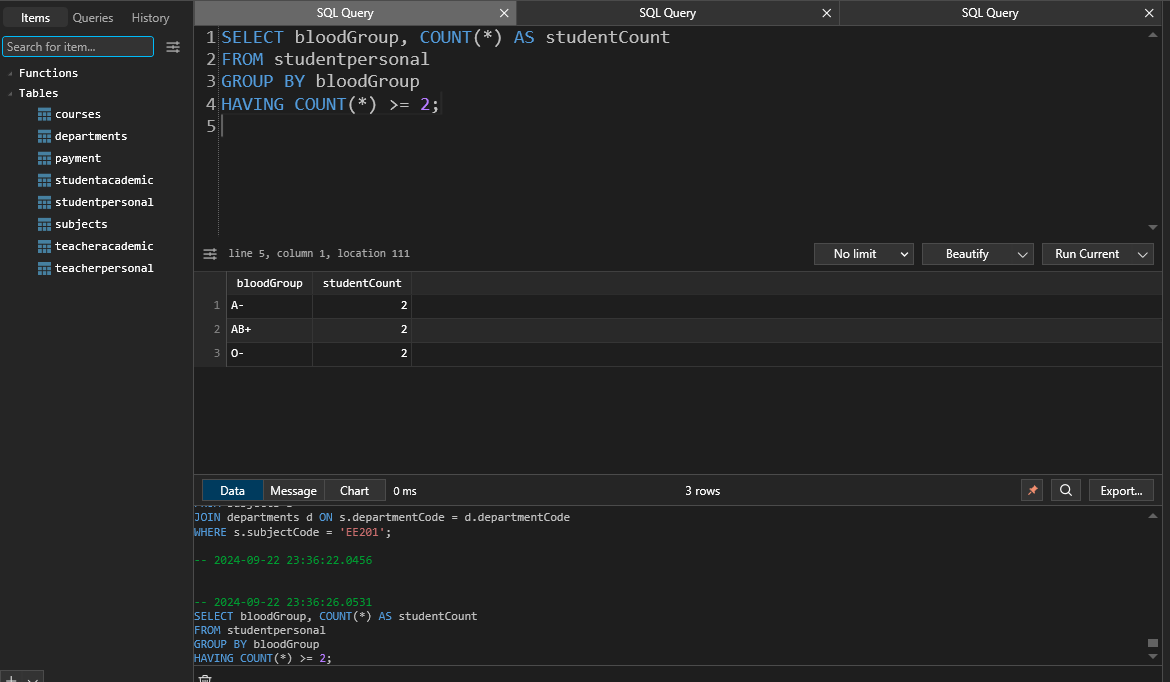
**WHERE s.subjectCode = 'EE201';**

3. Show the count of students for each blood group that has at least 2 students

SELECT bloodGroup, COUNT(\*) AS studentCount

FROM studentpersonal

GROUP BY bloodGroup

HAVING COUNT(\*) >= 2;

4. Write a query to find the students' first name, last name, and subject title for those enrolled in the department CS101.

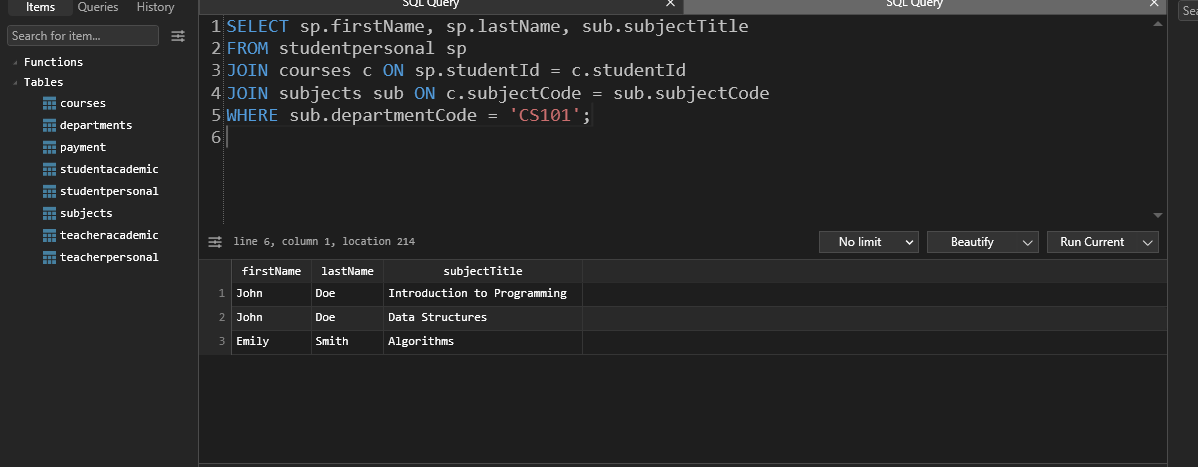
SELECT sp.firstName, sp.lastName, sub.subjectTitle

FROM studentpersonal sp

JOIN courses c ON sp.studentId = c.studentId

JOIN subjects sub ON c.subjectCode = sub.subjectCode

WHERE sub.departmentCode = 'CS101';



5. Find the total semester fee collected for the Computer Science department.

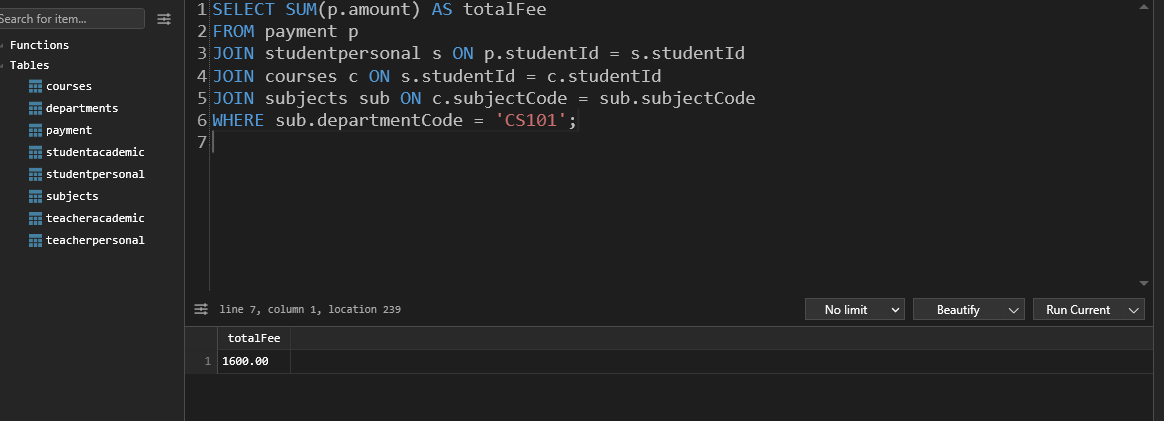
SELECT SUM(p.amount) AS totalFee

FROM payment p

JOIN studentpersonal s ON p.studentId = s.studentId

JOIN courses c ON s.studentId = c.studentId

JOIN subjects sub ON c.subjectCode = sub.subjectCode

WHERE sub.departmentCode = 'CS101';

6. Write a query to update the semester fee by 10% for students in the CS101 department.

UPDATE payment p

SET p.amount = p.amount \* 1.10

WHERE p.studentId IN (

SELECT s.studentId

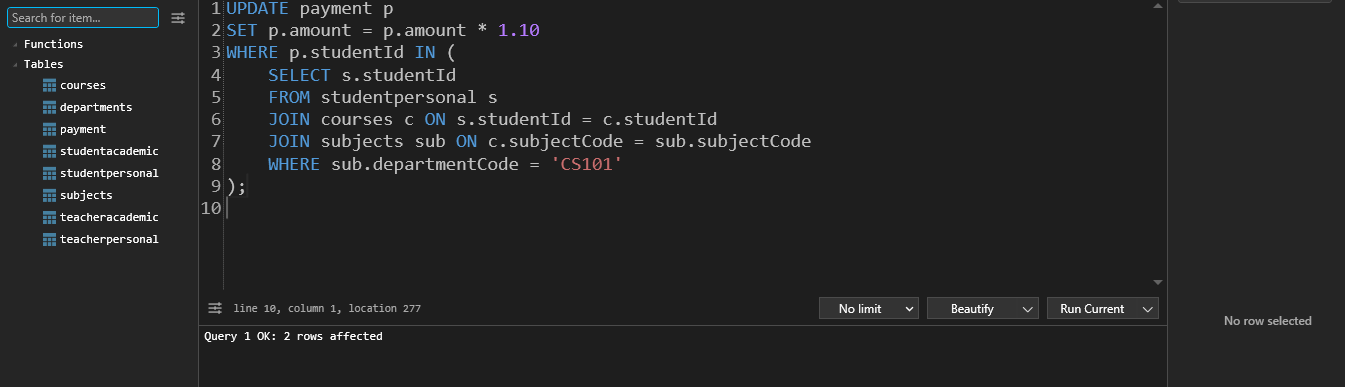
FROM studentpersonal s

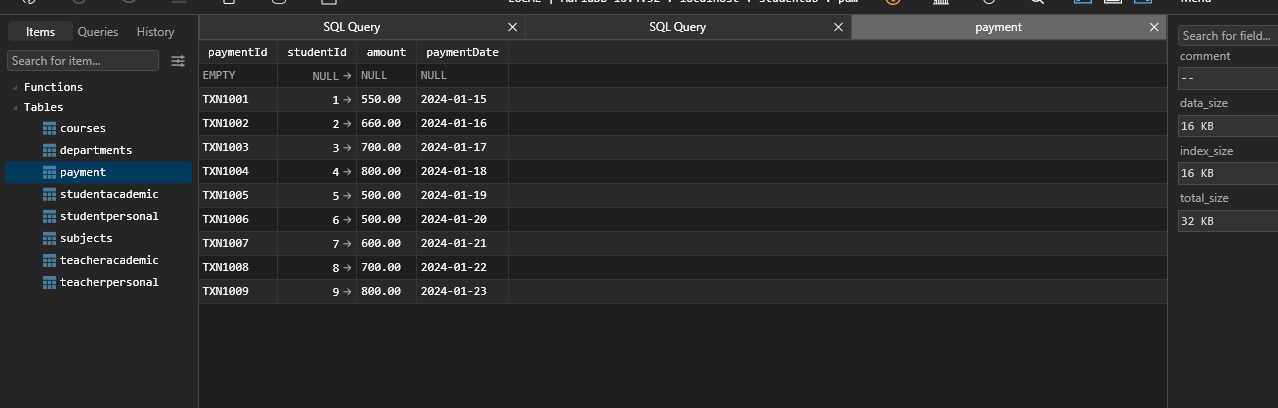
JOIN courses c ON s.studentId = c.studentId

JOIN subjects sub ON c.subjectCode = sub.subjectCode

WHERE sub.departmentCode = 'CS101'

);



****

**7. Write a query to find the students' first name, last name, teacher's first name, and teacher's designation who are enrolled in the course CS101.**

**SELECT s.firstname AS studentFirstName, s.lastname AS studentLastName,**

**t.firstname AS teacherFirstName, ta.Designation AS teacherDesignation**

**FROM studentpersonal s**

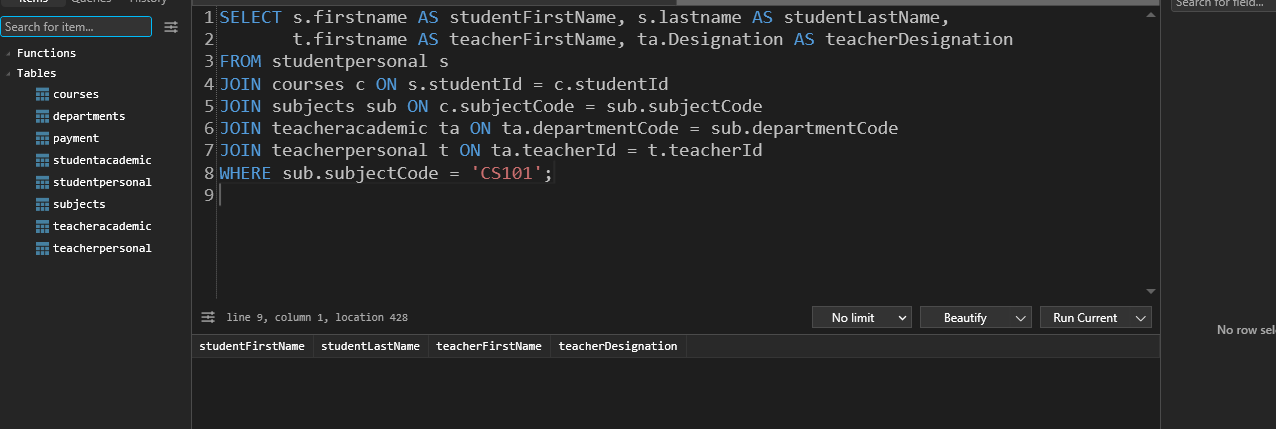
**JOIN courses c ON s.studentId = c.studentId**

**JOIN subjects sub ON c.subjectCode = sub.subjectCode**

**JOIN teacheracademic ta ON ta.departmentCode = sub.departmentCode**

**JOIN teacherpersonal t ON ta.teacherId = t.teacherId**

**WHERE sub.subjectCode = 'CS101';**

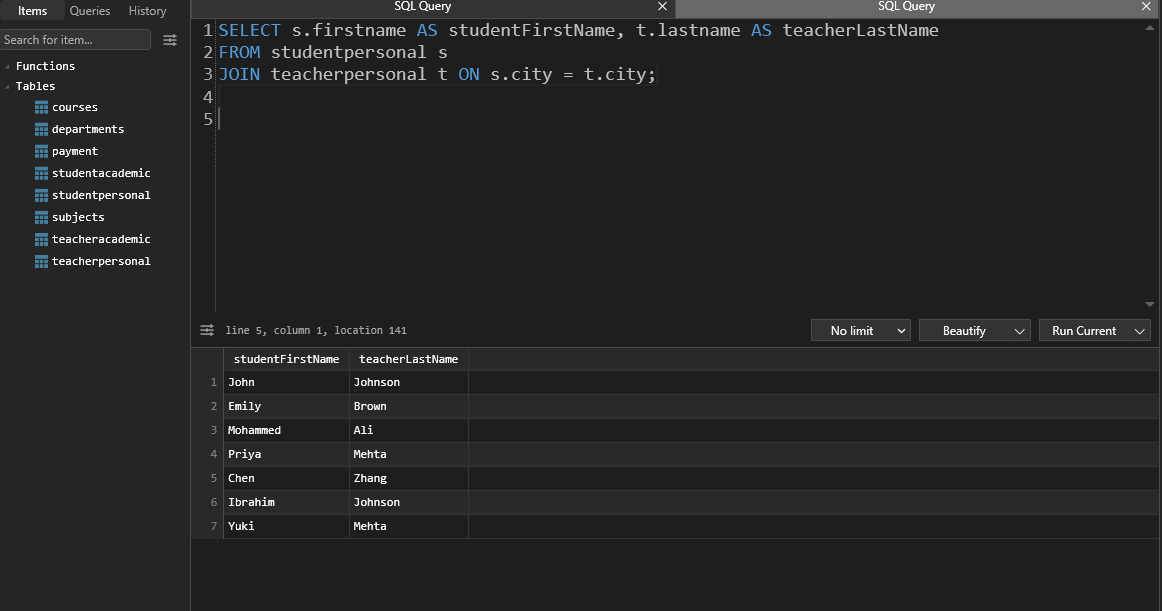
****

**8. Write a query to show the student's first name and the teacher's last name where both the teacher and student are from the same city.**

**SELECT s.firstname AS studentFirstName, t.lastname AS teacherLastName**

**FROM studentpersonal s**

**JOIN teacherpersonal t ON s.city = t.city;**

****

**9. Write a query to show the student's first name, last name, email, subject code, and subject title for the students enrolled in the subject "Algorithms**

**SELECT s.firstname AS studentFirstName, s.lastname AS studentLastName,**

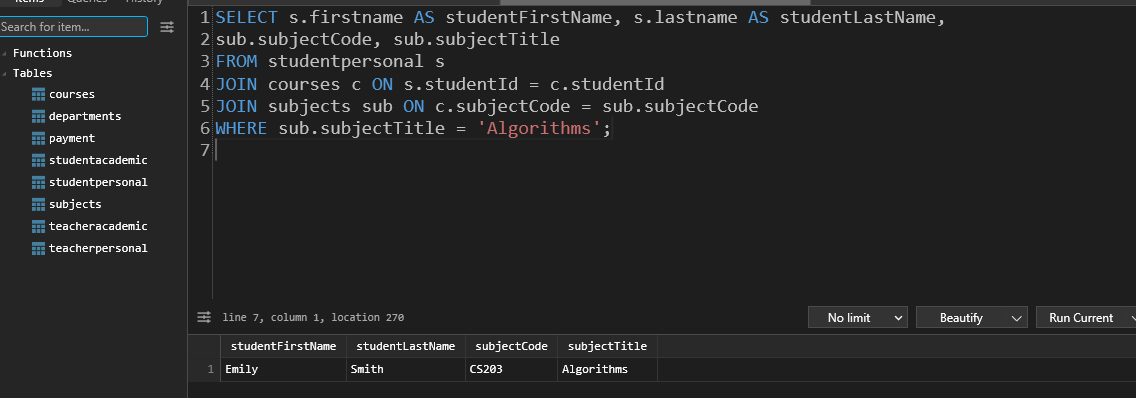
**sub.subjectCode, sub.subjectTitle**

**FROM studentpersonal s**

**JOIN courses c ON s.studentId = c.studentId**

**JOIN subjects sub ON c.subjectCode = sub.subjectCode**

**WHERE sub.subjectTitle = 'Algorithms';**

****

**10. Write a query to promote teachers who are currently Lecturers to Sr. Lecturers, and promote Sr. Lecturers to Associate Professors.**

**UPDATE teacheracademic**

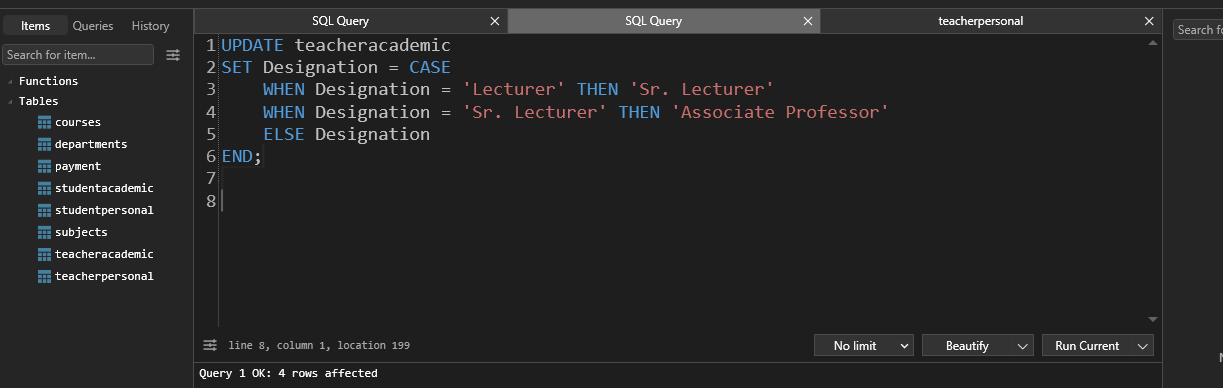
**SET Designation = CASE**

**WHEN Designation = 'Lecturer' THEN 'Sr. Lecturer'**

**WHEN Designation = 'Sr. Lecturer' THEN 'Associate Professor'**

**ELSE Designation**

**END;**

****

**11. Write a query to find the teacher's first name and last name who has the second highest salary.**

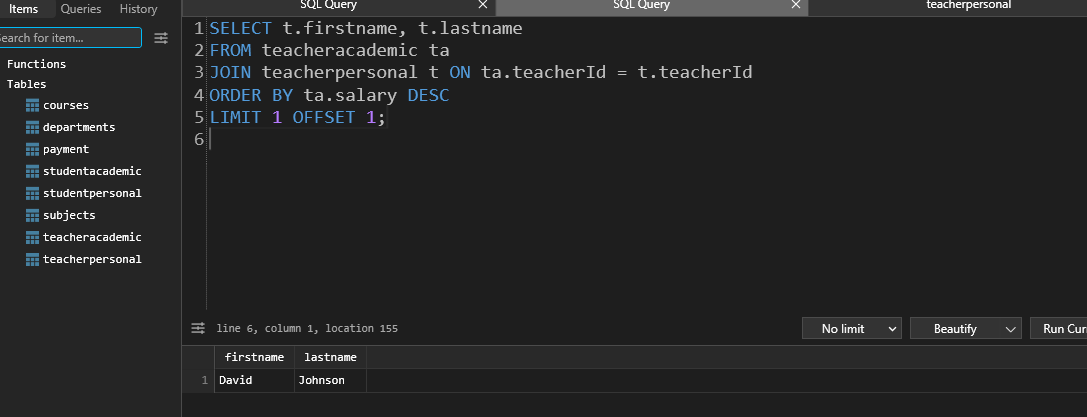
**SELECT t.firstname, t.lastname**

**FROM teacheracademic ta**

**JOIN teacherpersonal t ON ta.teacherId = t.teacherId**

**ORDER BY ta.salary DESC**

**LIMIT 1 OFFSET 1;**

****

**12. Write a query to find the teacher who has the second highest salary and list the students enrolled in their course. Show the teacher's first name, last name, and the students' first name, last name, city, and department code.**

**SELECT t.firstname AS teacherFirstName, t.lastname AS teacherLastName,**

**s.firstname AS studentFirstName, s.lastname AS studentLastName,**

**s.city, sub.departmentCode**

**FROM teacheracademic ta**

**JOIN teacherpersonal t ON ta.teacherId = t.teacherId**

**JOIN subjects sub ON ta.departmentCode = sub.departmentCode**

**JOIN courses c ON sub.subjectCode = c.subjectCode**

**JOIN studentpersonal s ON c.studentId = s.studentId**

**WHERE ta.teacherId = (**

**SELECT teacherId**

**FROM teacheracademic**

**ORDER BY salary DESC**

**LIMIT 1 OFFSET 1**

**);**

