# Assignment on Database (MySQL)

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-.**

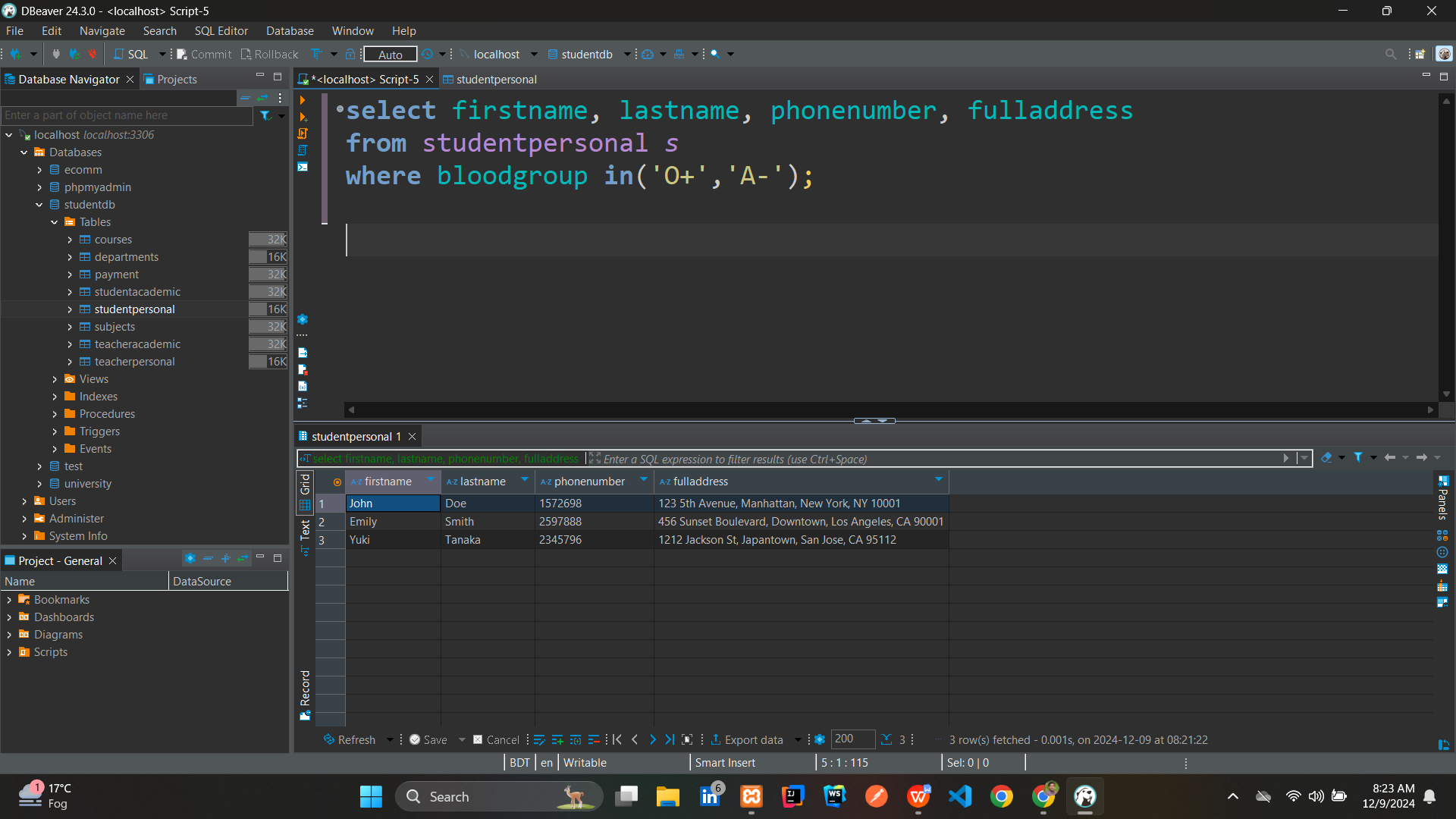
* **Query:**

**select** firstname, lastname, phonenumber, fulladdress

**from** studentpersonal *s*

**where** bloodgroup **in**('O+','A-');

* **Output:**

****

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

* **Query:**

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

**from** subjects *s* **join** departments *d*

**on** *s*.departmentCode = *d*.departmentCode

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

* **Output:**

****

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

* **Query:**

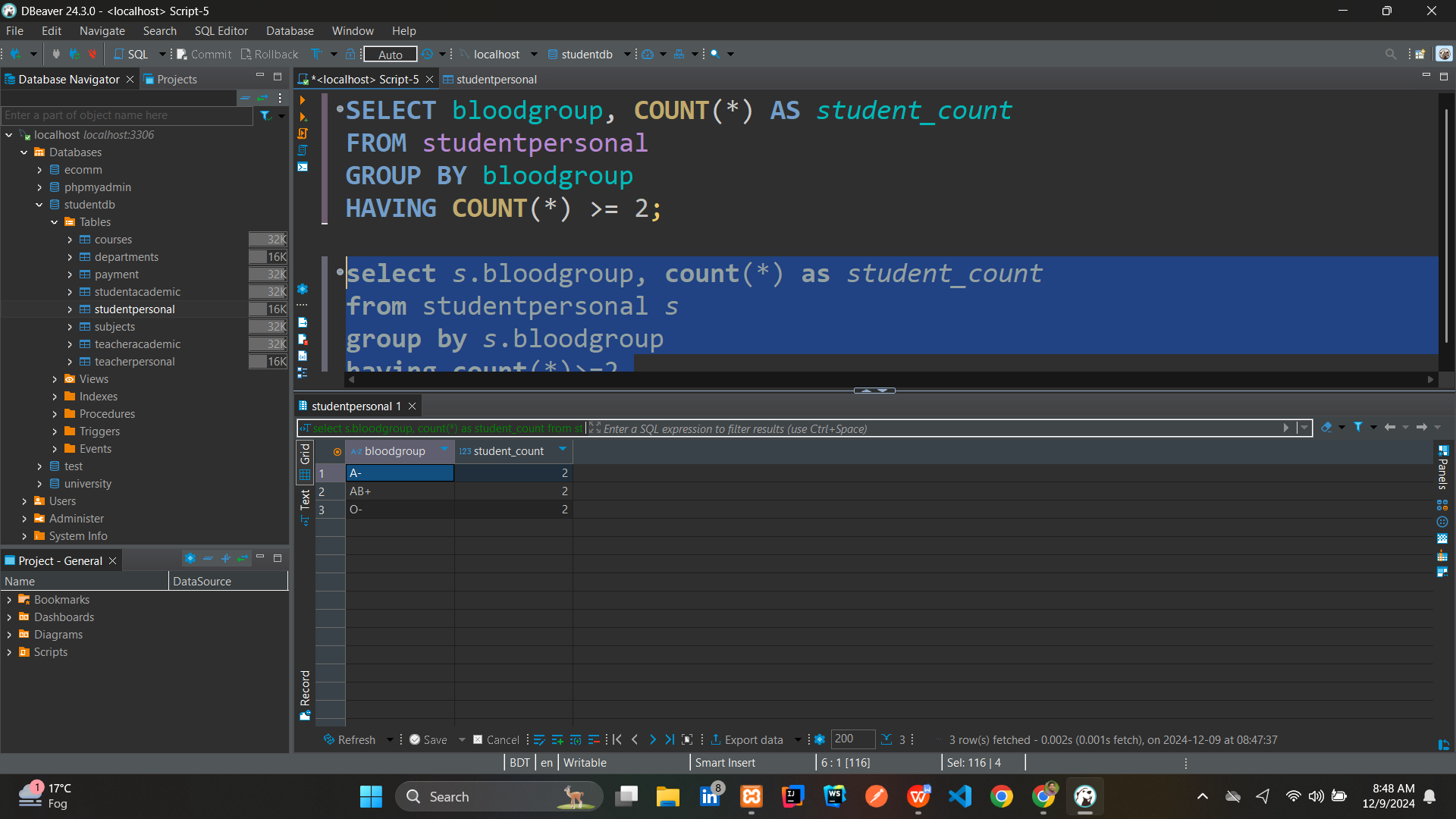
**select** *s*.bloodgroup, **count**(\*) **as** *student\_count*

**from** studentpersonal *s*

**group** **by** *s*.bloodgroup

**having** **count**(\*)>=2

* **Output:**

****

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

* **Query:**

**select** *sp*.firstname, *sp*.lastname, *s*.subjectTitle

**from** studentpersonal *sp* **join** studentacademic *sa*

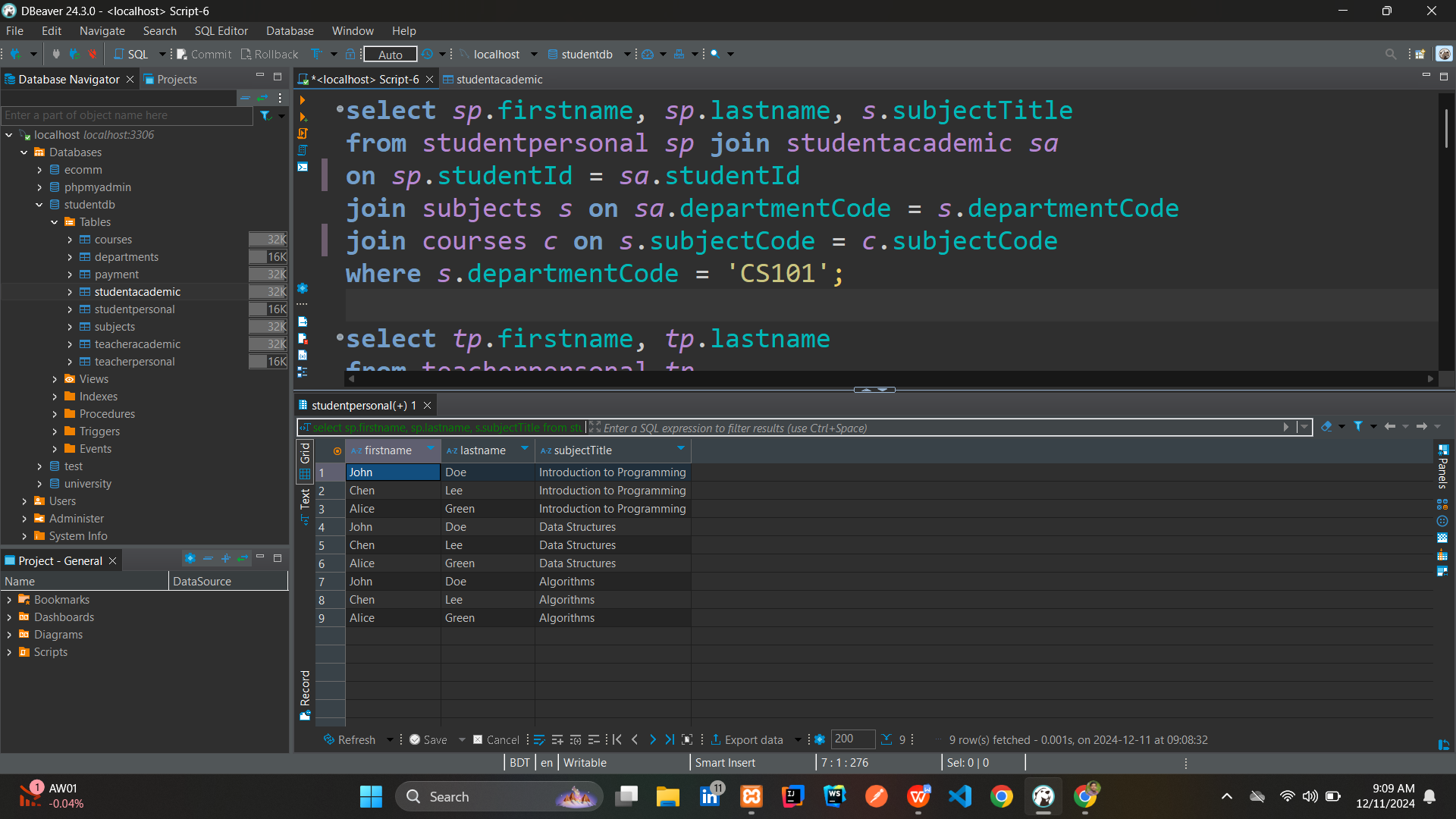
**on** *sp*.studentId = *sa*.studentId

**join** subjects *s* **on** *sa*.departmentCode = *s*.departmentCode

**join** courses *c* **on** *s*.subjectCode = *c*.subjectCode

**where** *sa*.departmentCode = 'CS101';

* **Output:**

****

1. **Find the total semester fee collected for the Computer Science department.**

* **Query:**

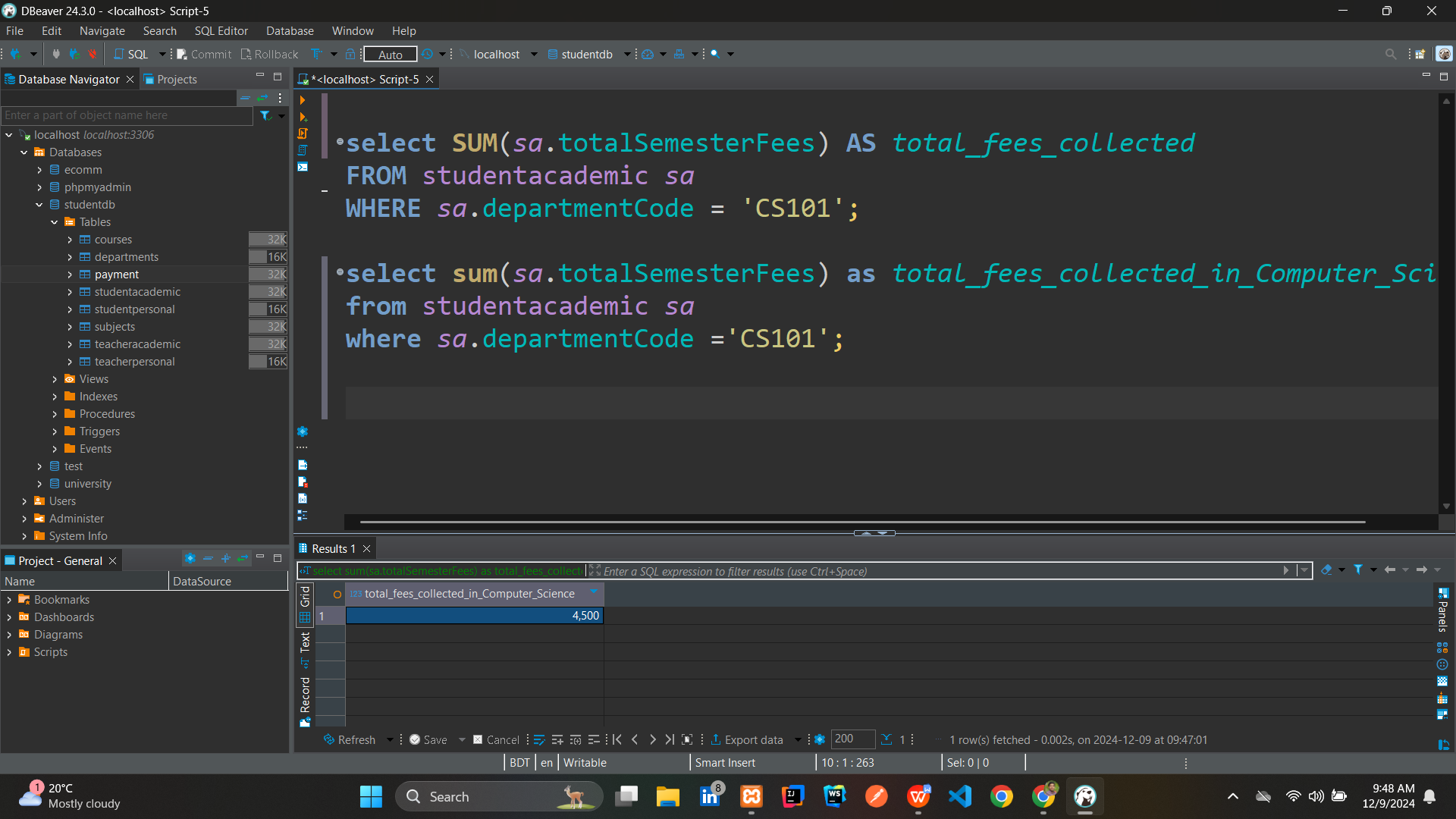
**select** **sum**(*sa*.totalSemesterFees) **as** *total\_fees\_collected\_in\_Computer\_*

*Science*

**from** studentacademic *sa*

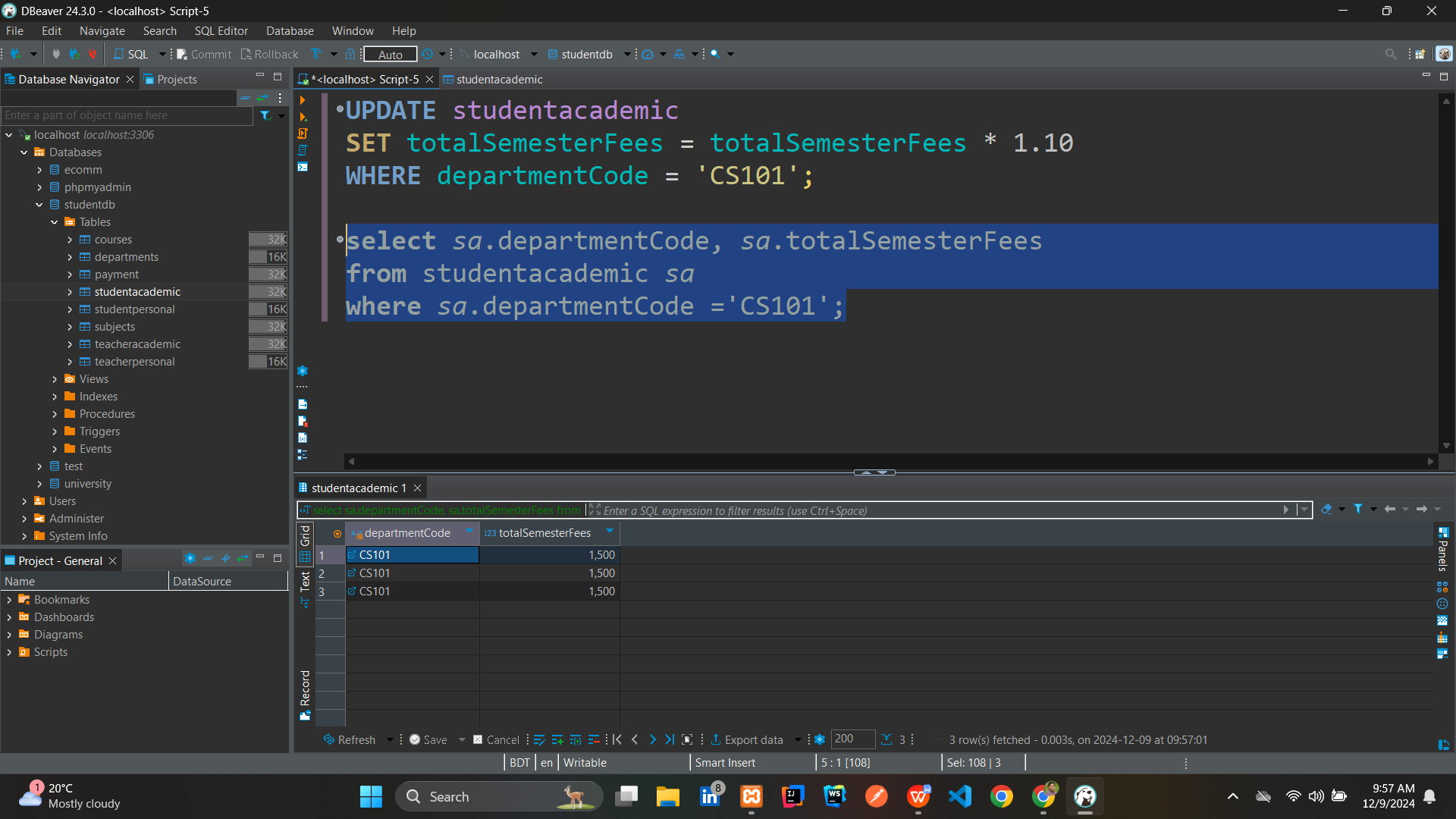
**where** *sa*.departmentCode ='CS101';

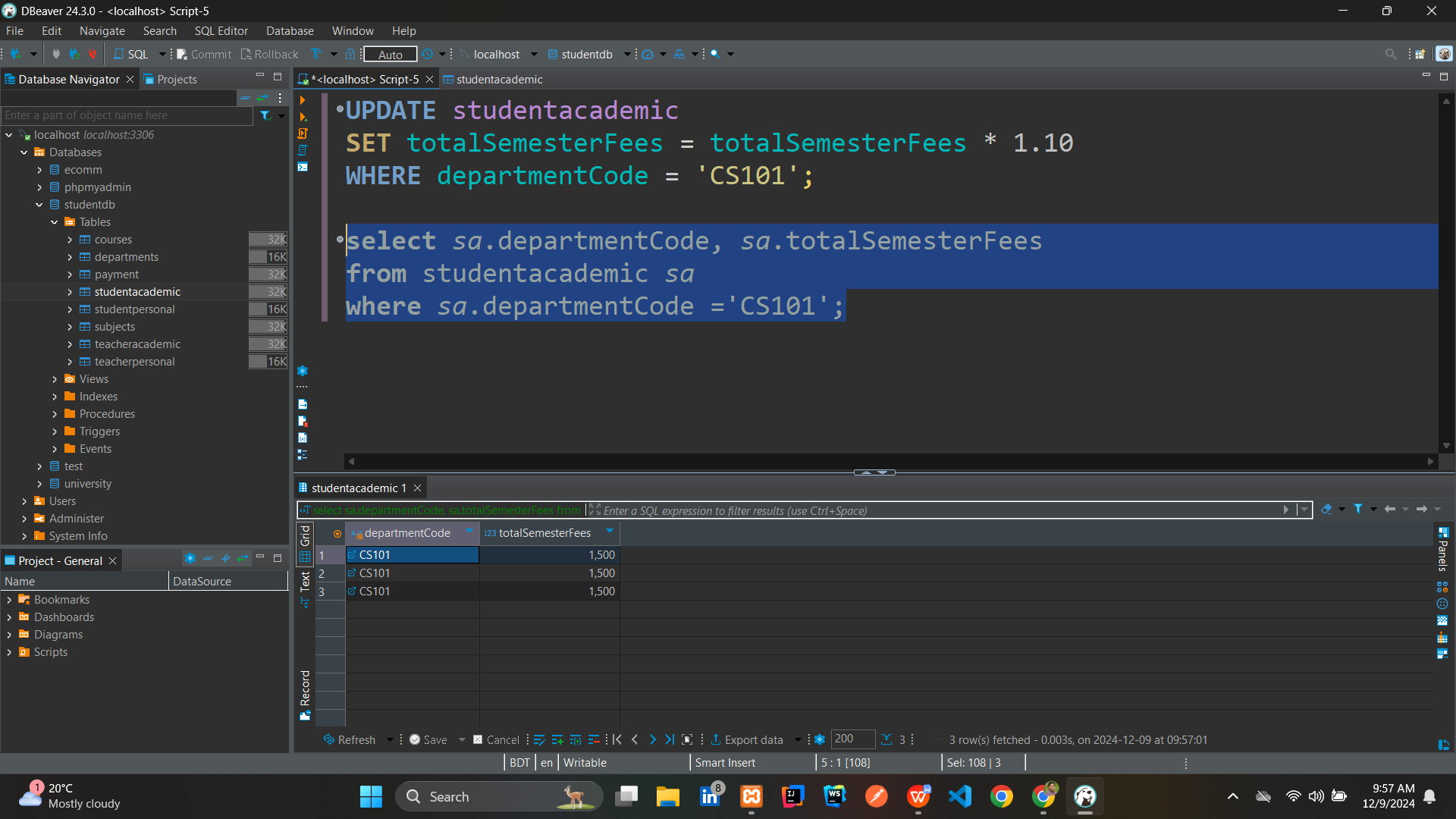
* **Output:**

****

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

* **Before Updation:**

****

****

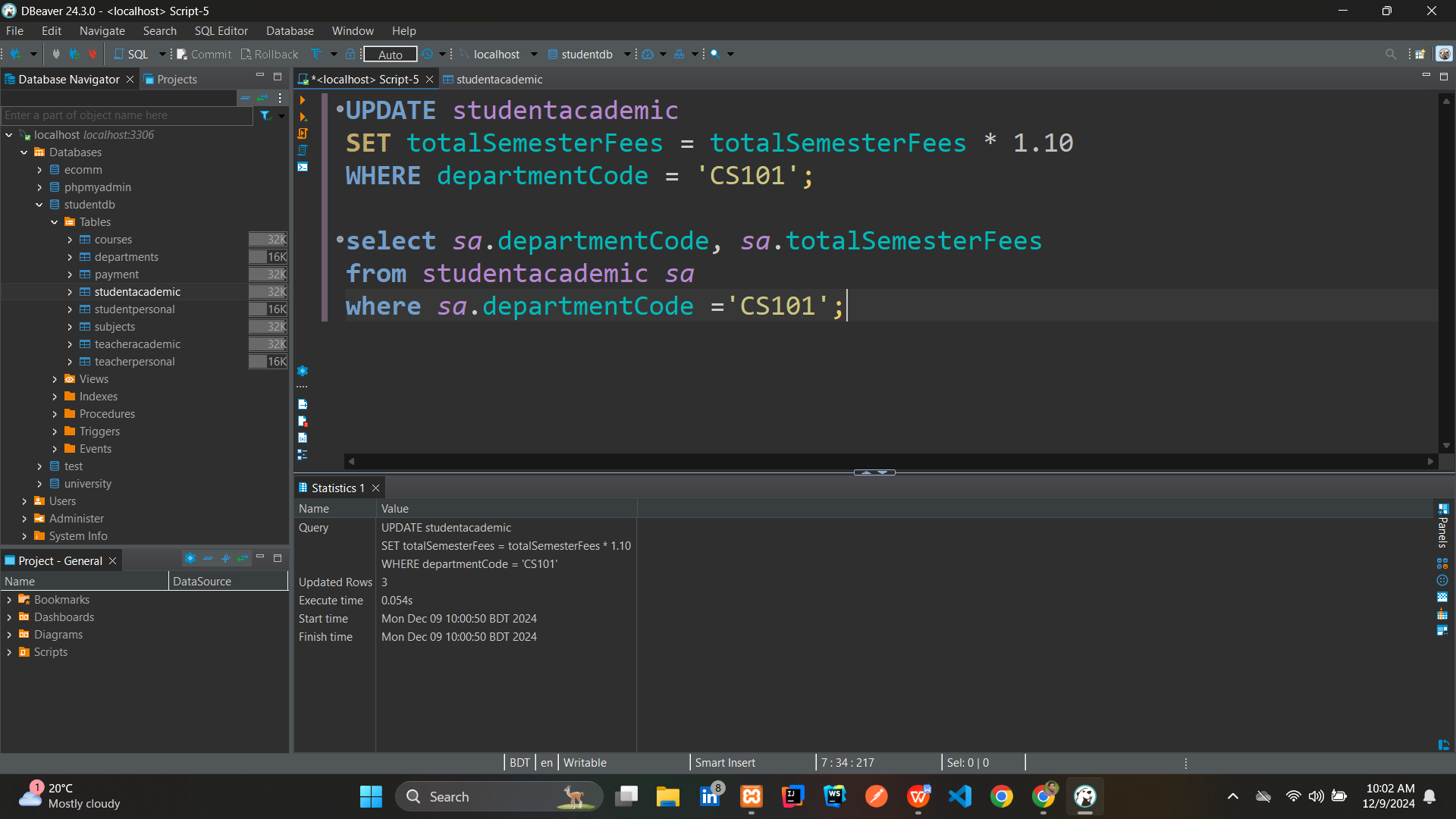
* **After executing the below Query:**

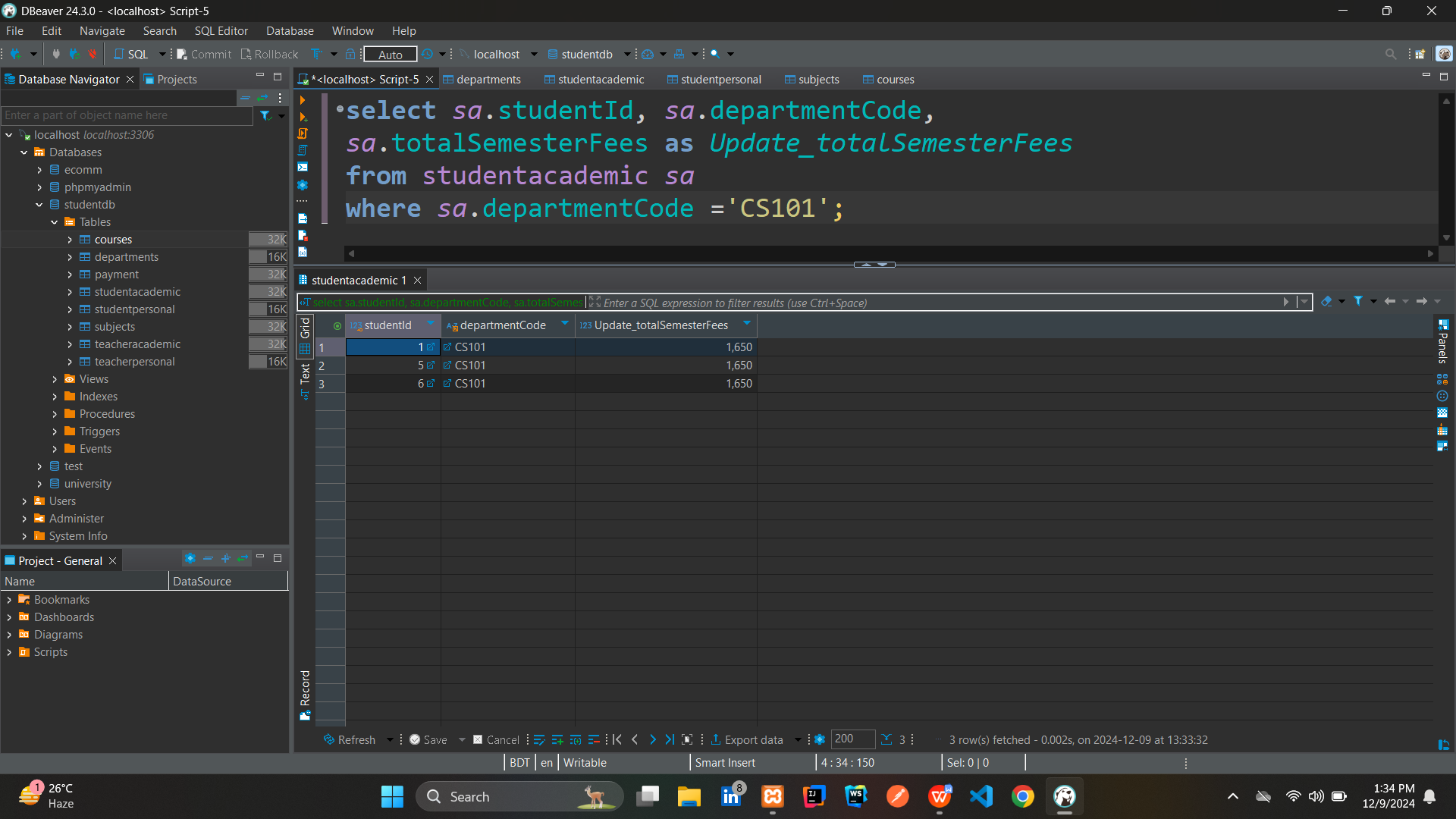
**UPDATE** studentacademic

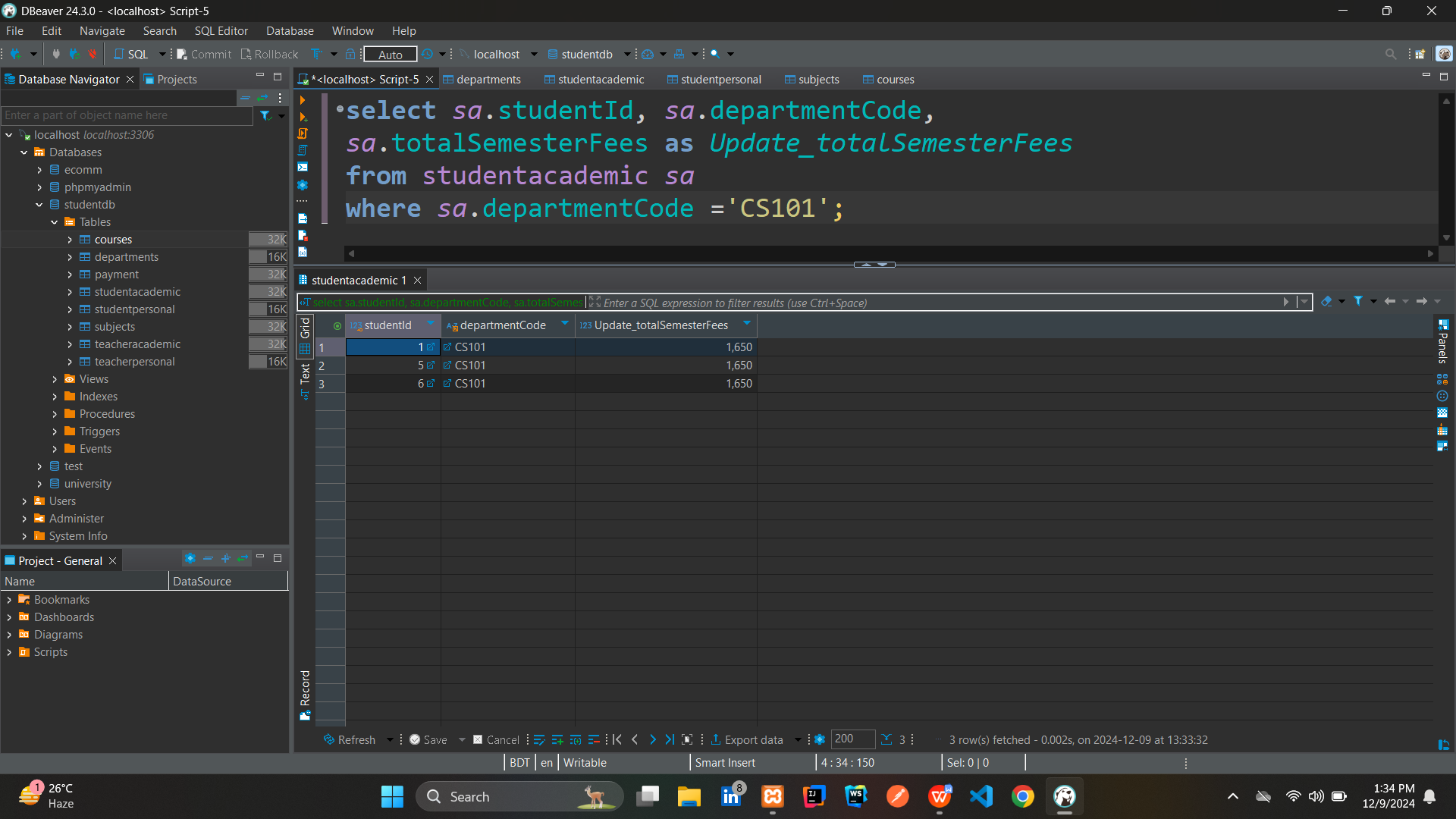
**SET** totalSemesterFees = totalSemesterFees \* 1.10

**WHERE** departmentCode = 'CS101';

* **Updation Output:**

****

****

****

1. **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.**

* **Query:**

**select** *sp*.firstname **as** *student\_firstname*, *sp*.lastname **as** *student\_lastname*,

*tp*.firstname **as** *teacher\_firstname*, *ta*.Designation **as** *teacher\_designation*

**from** studentacademic *sa* **join** teacheracademic *ta*

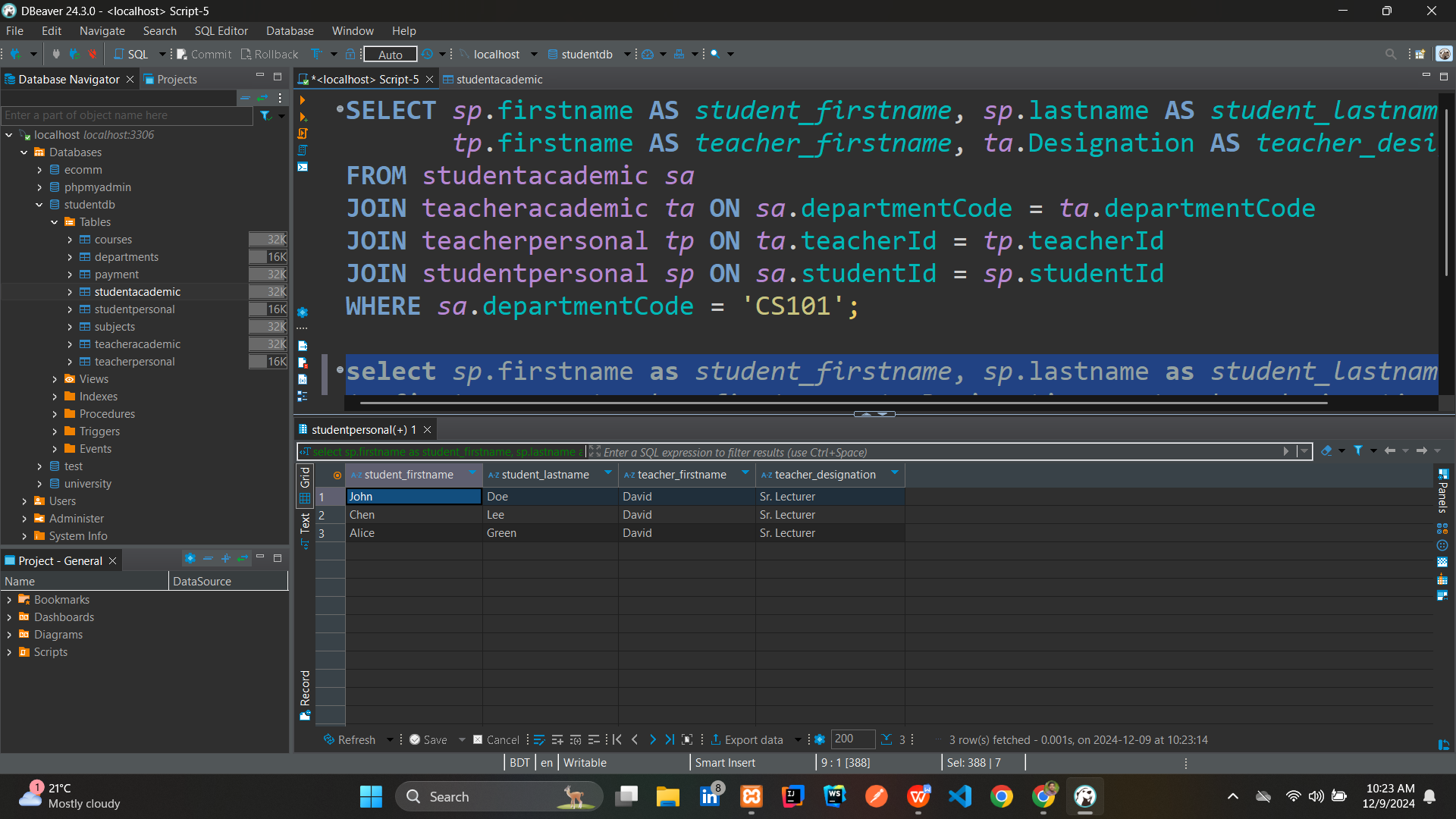
**on** *sa*.departmentCode = *ta*.departmentCode

**join** studentpersonal *sp* **on** *sa*.studentId = *sp*.studentId

**join** teacherpersonal *tp* **on** *ta*.teacherId = *tp*.teacherId

**where** *sa*.departmentCode = 'CS101';

* **Output:**

****

1. **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.**

* **Query:**

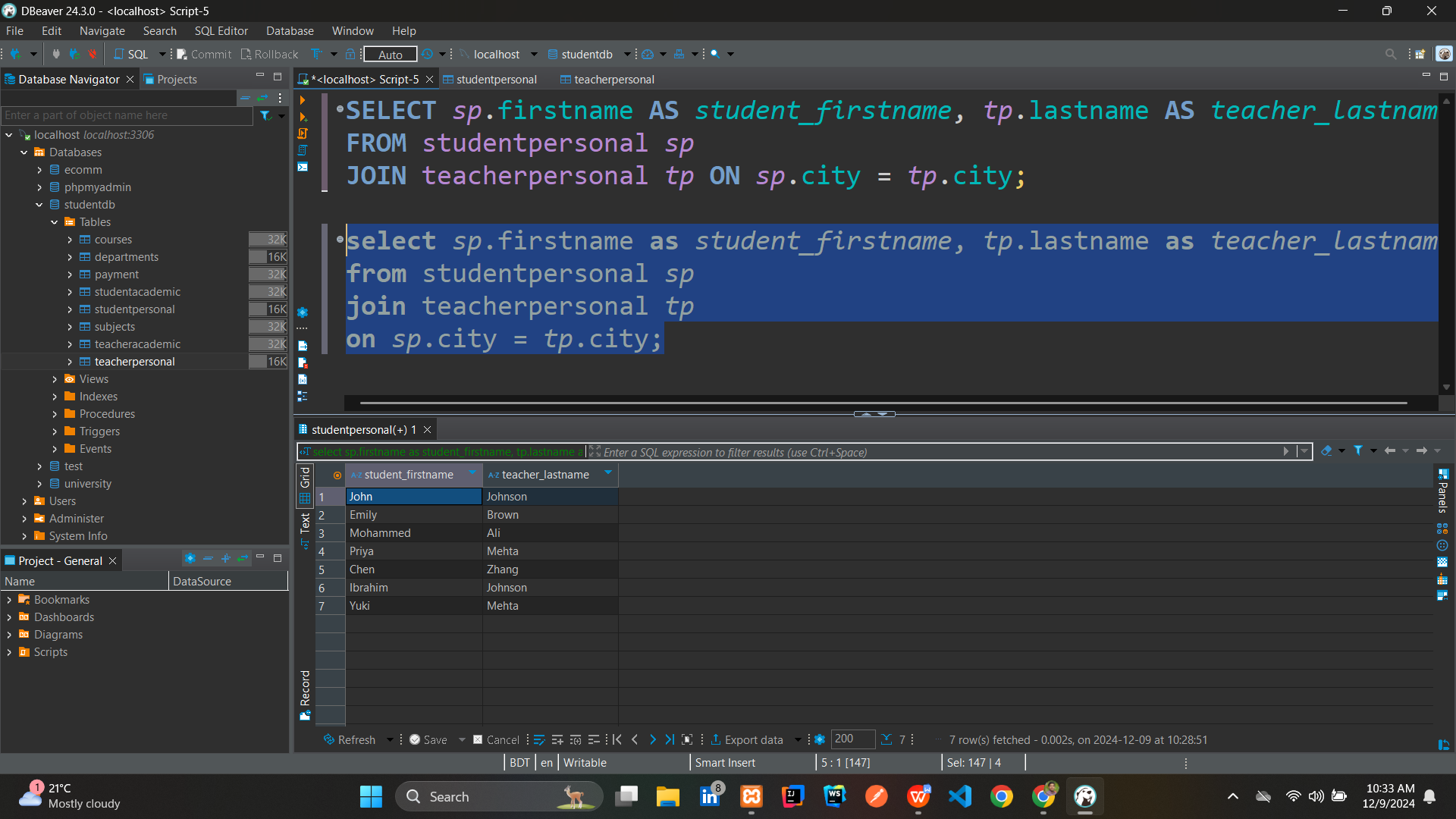
**select** *sp*.firstname **as** *student\_firstname*, *tp*.lastname **as** *teacher\_lastname*

**from** studentpersonal *sp*

**join** teacherpersonal *tp*

**on** *sp*.city = *tp*.city;

* **Output:**

****

1. **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."**

* **Query:**

**select** *sp*.firstname, *sp*.lastname, *sp*.email, *c*.subjectCode,

*s*.subjectTitle

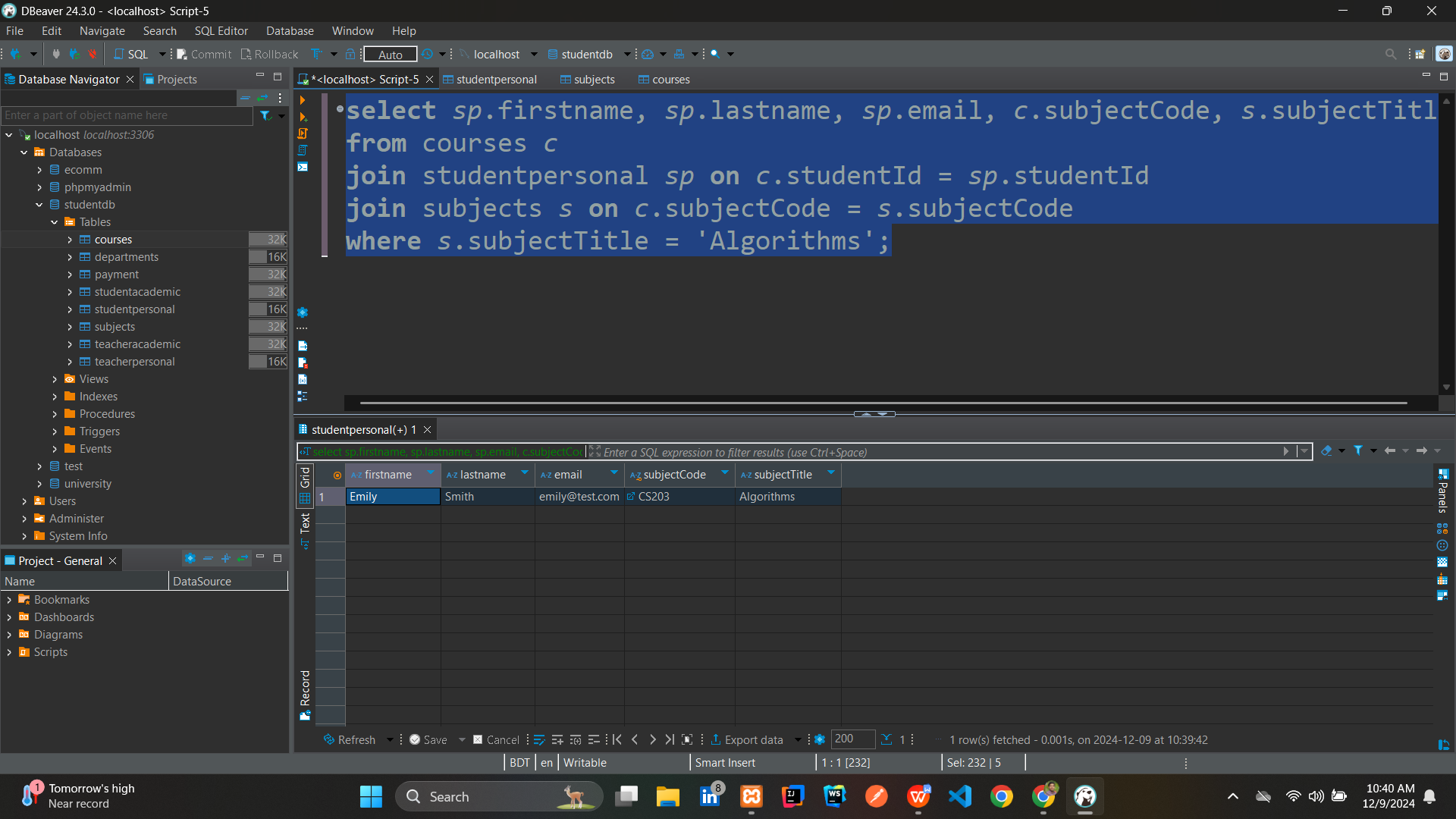
**from** courses *c*

**join** studentpersonal *sp* **on** *c*.studentId = *sp*.studentId

**join** subjects *s* **on** *c*.subjectCode = *s*.subjectCode

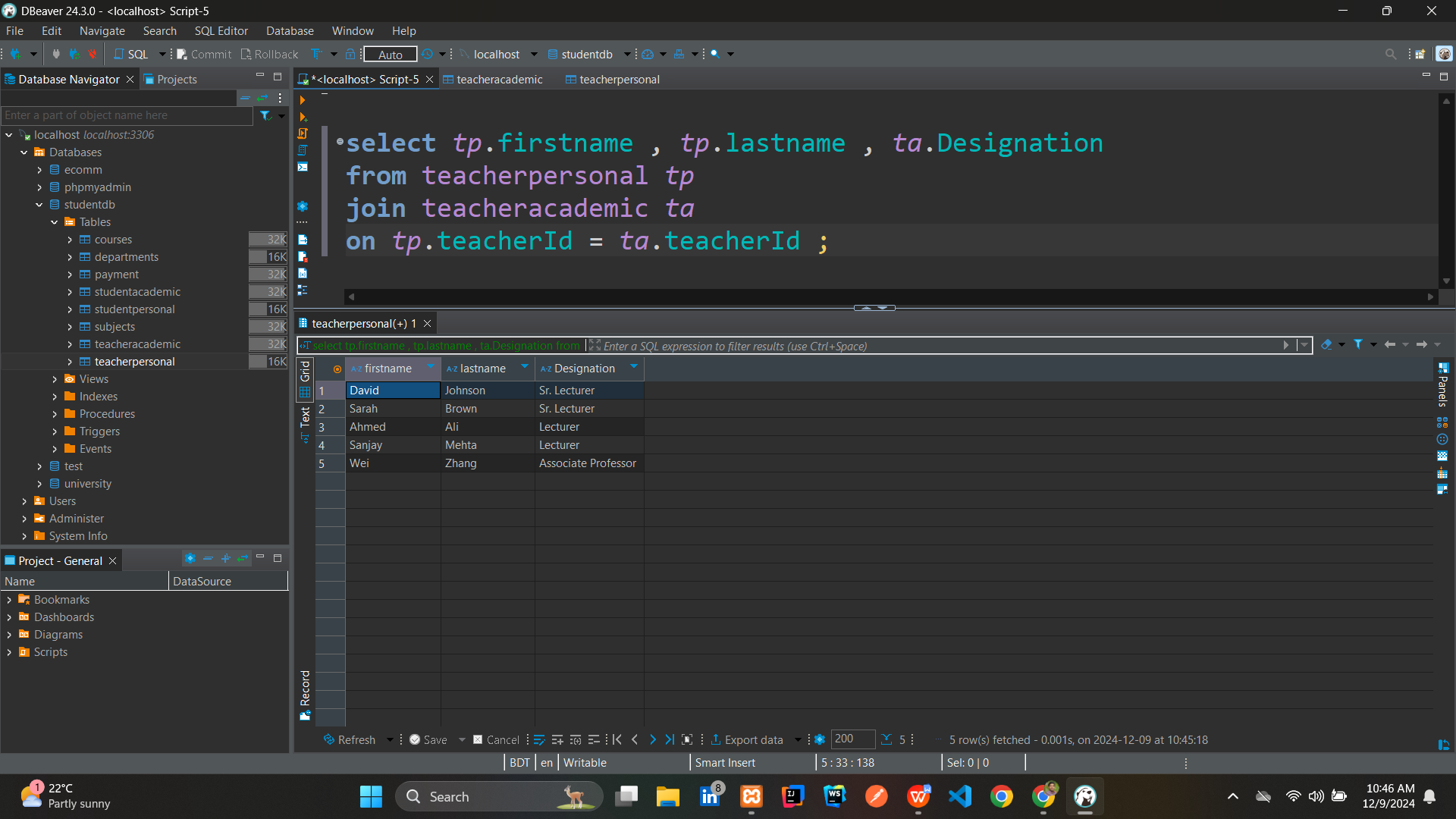
**where** *s*.subjectTitle ='Algorithms';

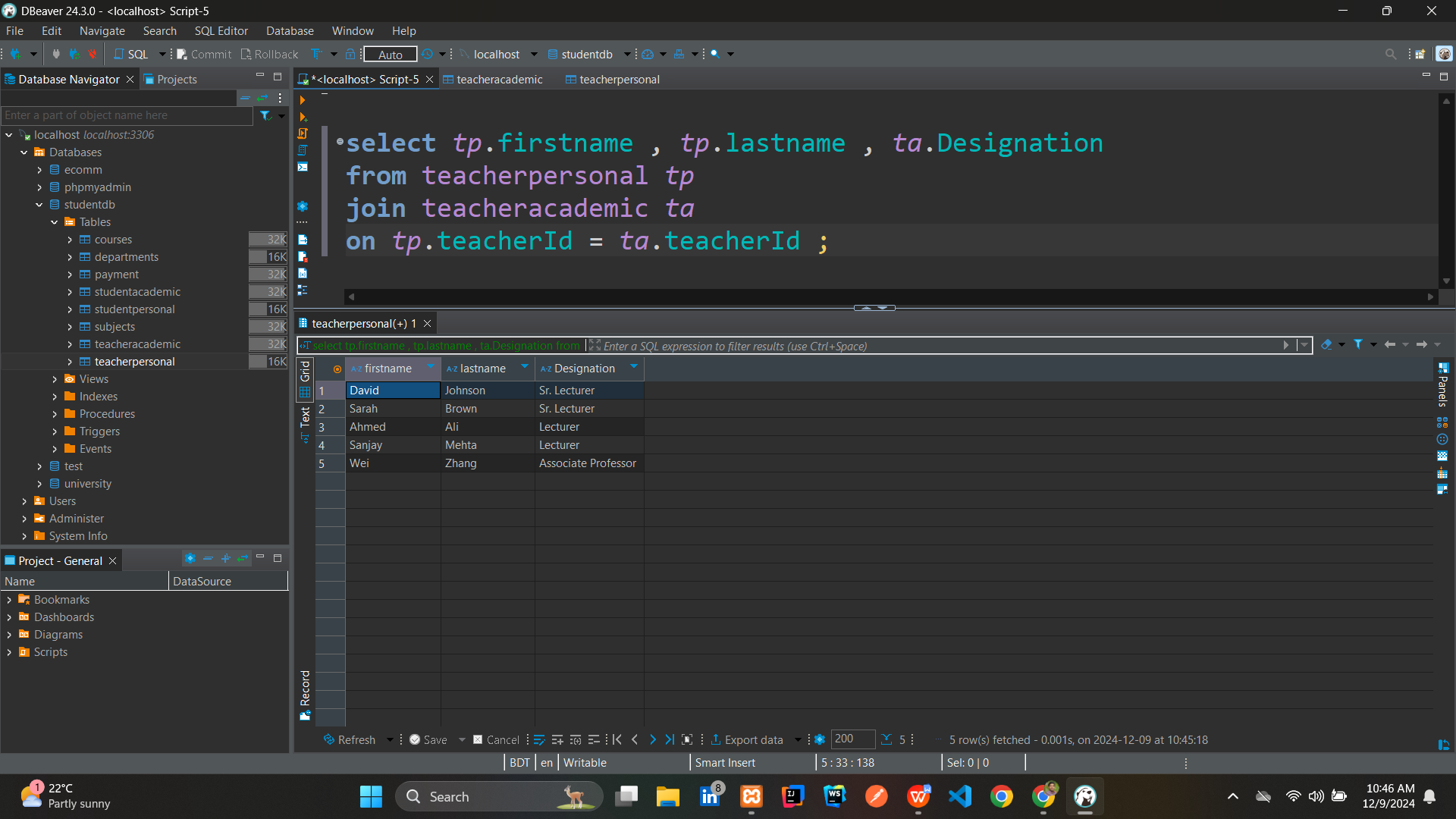
* **Output:**

****

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

* **Before Promotion:**

****

****

* **After executing the below Query:**

**UPDATE** teacheracademic

**SET** Designation =

**CASE**

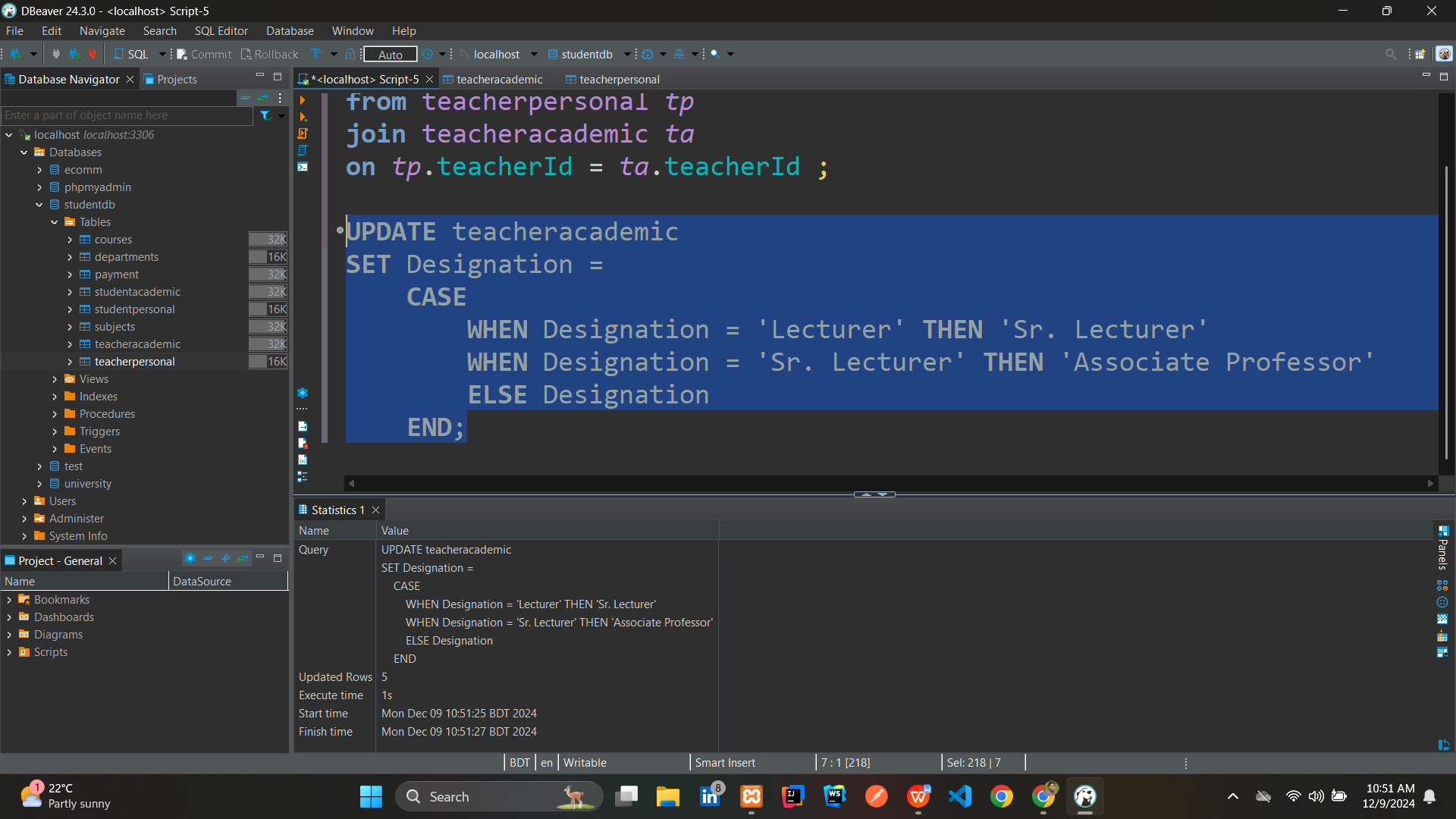
**WHEN** Designation = 'Lecturer' **THEN** 'Sr. Lecturer'

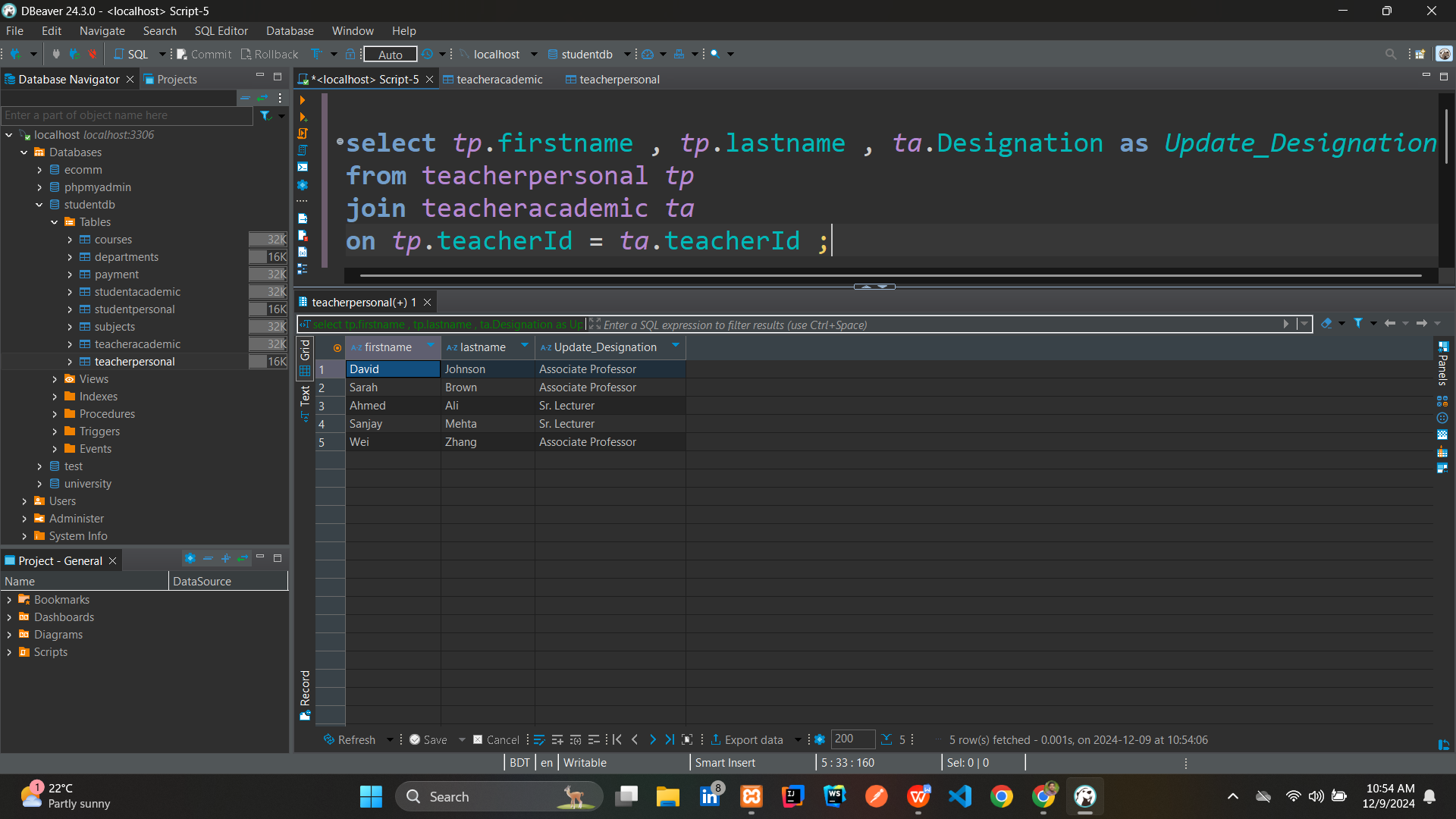
**WHEN** Designation = 'Sr. Lecturer' **THEN** 'Associate Professor'

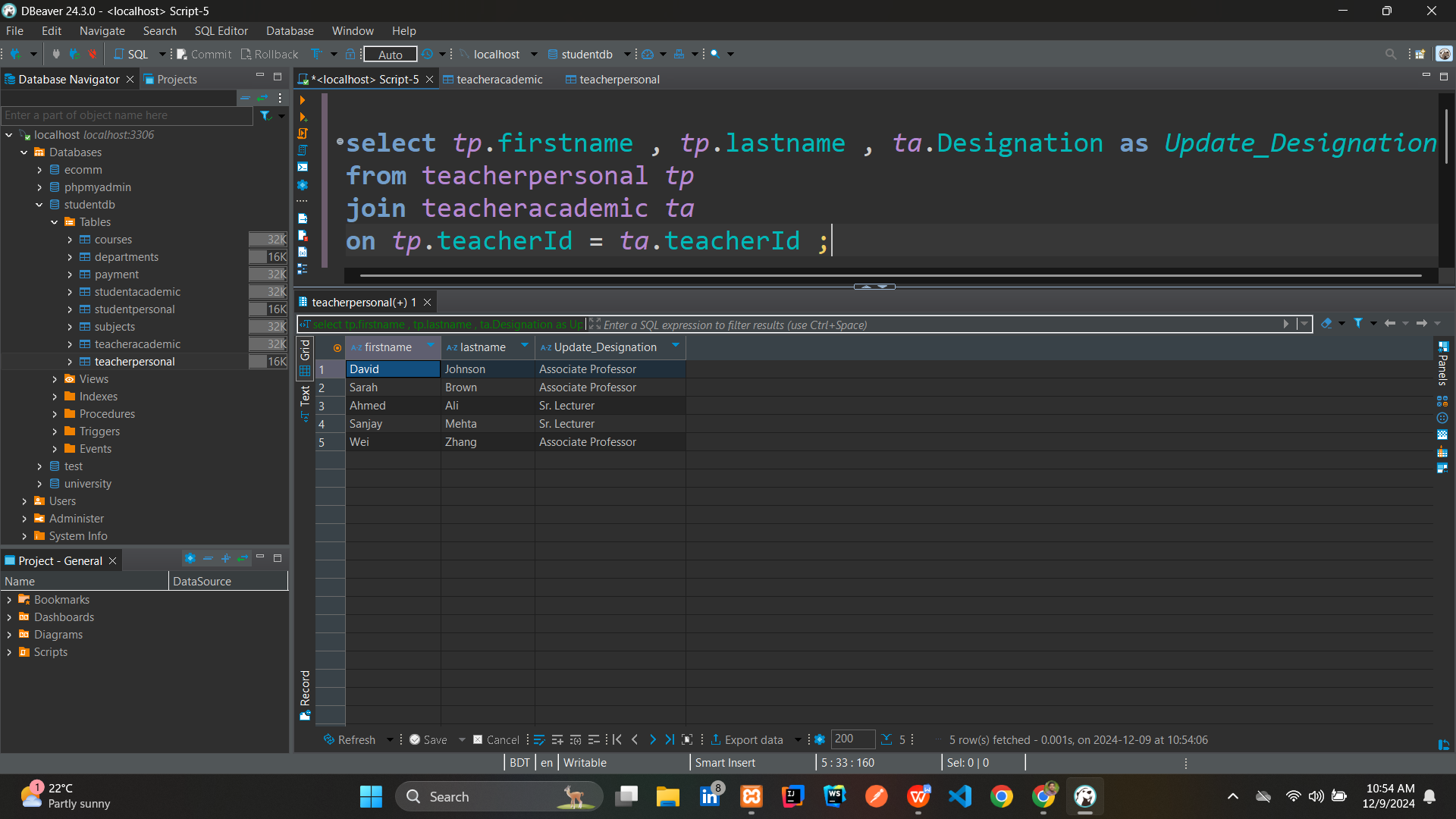
**ELSE** Designation

**END**;

* **Output:**

****

****

****

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

* **Query:**

**select** *tp*.firstname, *tp*.lastname

**from** teacherpersonal *tp*

**join** teacheracademic *ta* **on** *tp*.teacherId = *ta*.teacherId

**where** *ta*.salary = (

**select** **max**(salary)

**from** teacheracademic

**where** salary < (

**select** **max**(salary)

**from** teacheracademic)

);

**OR**

**select** *tp*.firstname, *tp*.lastname

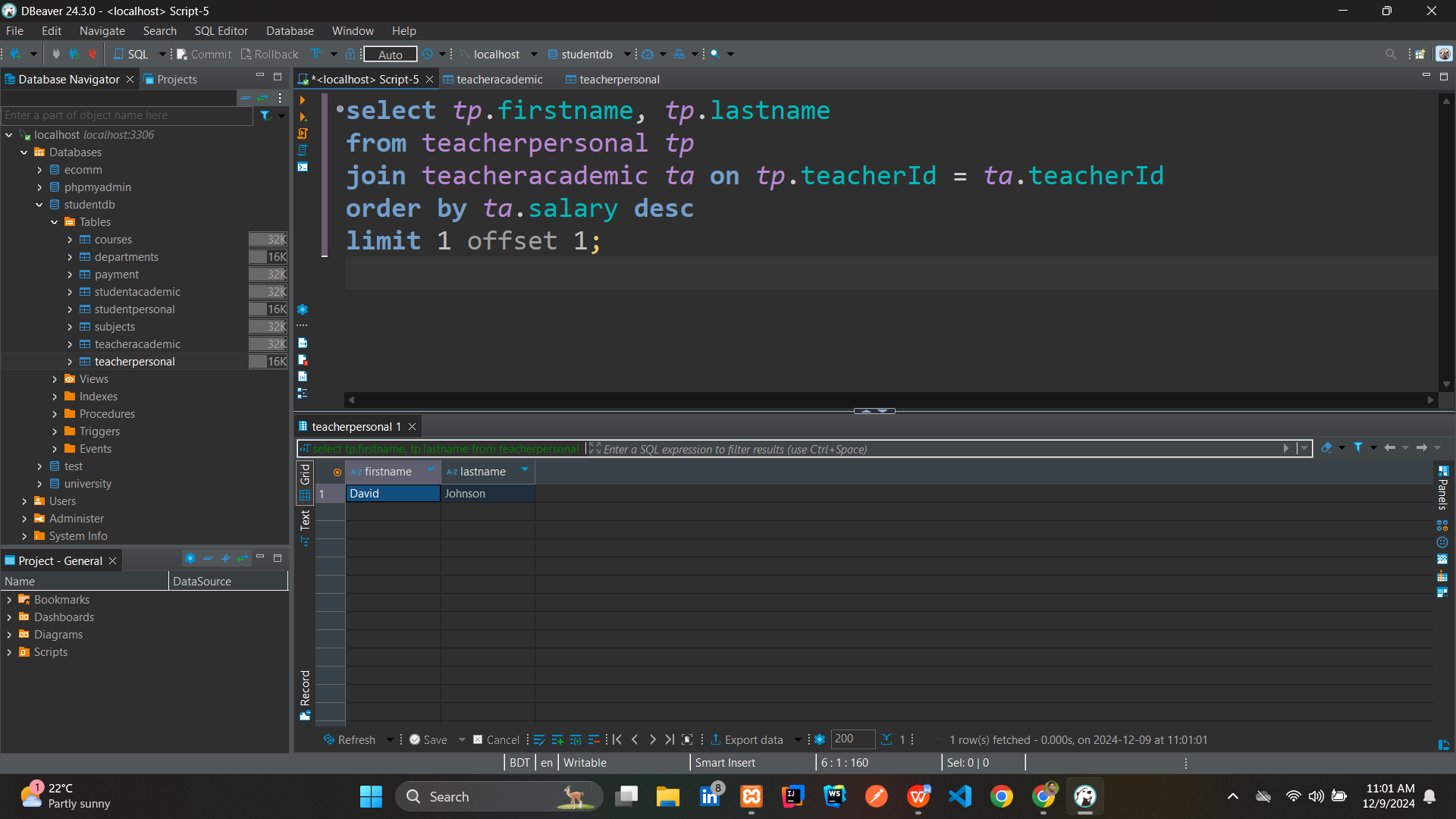
**from** teacherpersonal *tp*

**join** teacheracademic *ta* **on** *tp*.teacherId = *ta*.teacherId

**order** **by** *ta*.salary **desc**

**limit** 1 offset 1;

* **Output:**

****

1. **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.**

* **Query:**

**with** *RankedSalaries* **as** (

**select**

*ta*.teacherId,

*ta*.departmentCode,

*ta*.salary,

**rank**() **over** (**order** **by** *ta*.salary **desc**) **as** *rank*

**from** teacheracademic *ta*),

*SecondHighestSalary* **as** (

**select**

*rs*.teacherId,

*rs*.departmentCode

**from** *RankedSalaries* *rs*

**where** *rs*.*rank* = 2)

**select**

*tp*.firstname **as** *teacher\_firstname*,

*tp*.lastname **as** *teacher\_lastname*,

*sp*.firstname **as** *student\_firstname*,

*sp*.lastname **as** *student\_lastname*,

*sp*.city **as** *student\_city*,

*sa*.departmentCode **as** *department\_code*

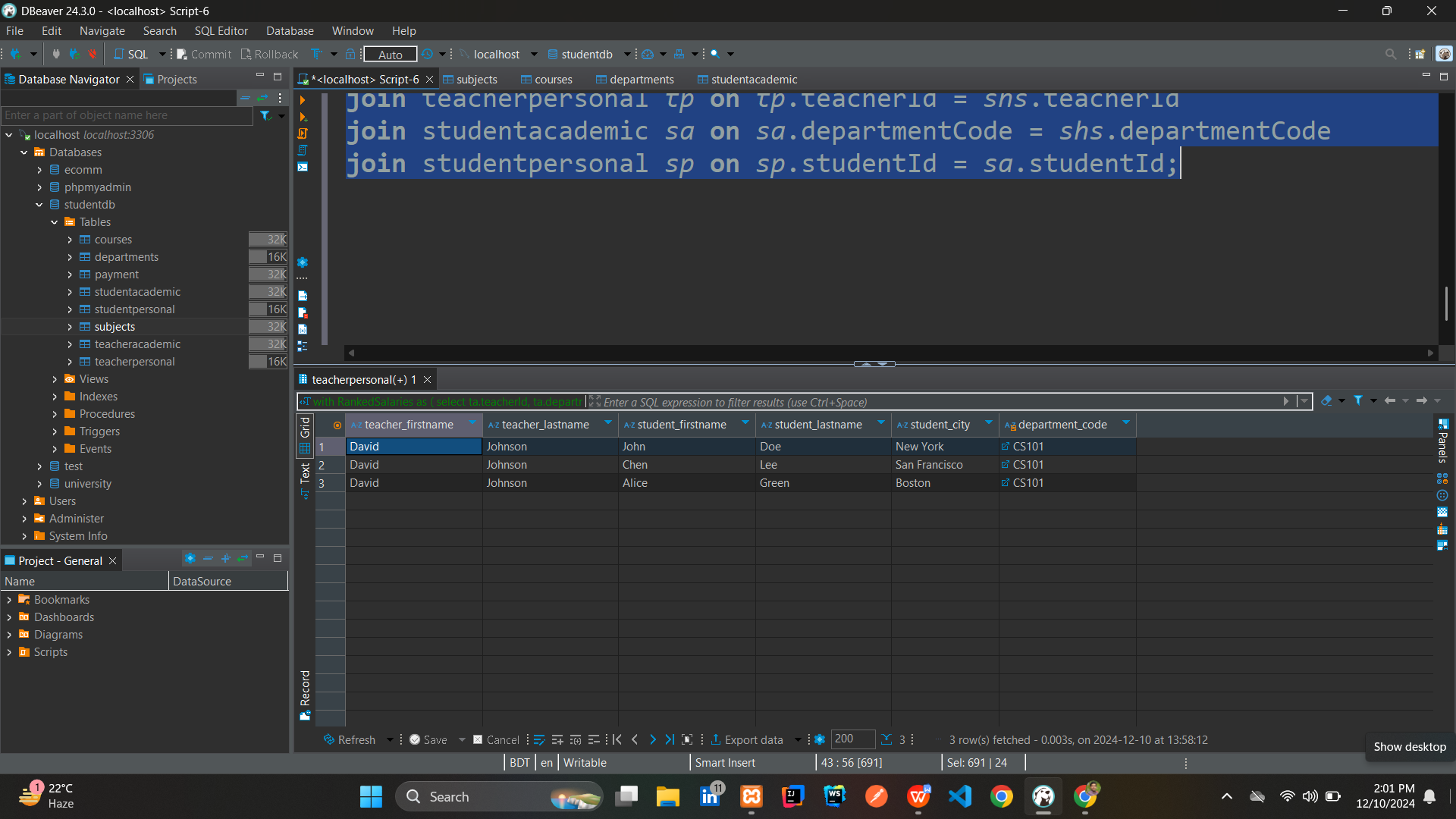
**from** *SecondHighestSalary* *shs*

**join** teacherpersonal *tp* **on** *tp*.teacherId = *shs*.teacherId

**join** studentacademic *sa* **on** *sa*.departmentCode = *shs*.departmentCode

**join** studentpersonal *sp* **on** *sp*.studentId = *sa*.studentId;

* **Output:**

****

**THE END**