Задание 1:

1. Вставить двух новых сотрудников в таблицу Employees.

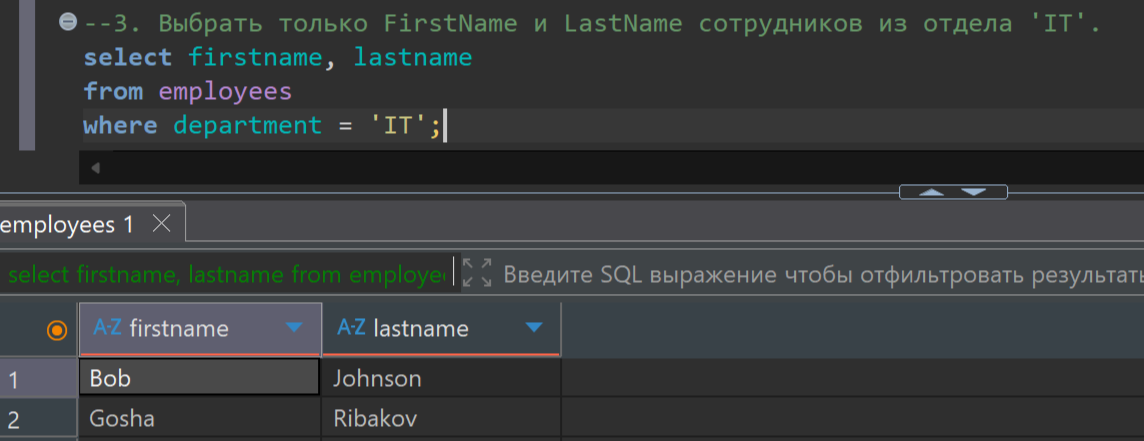
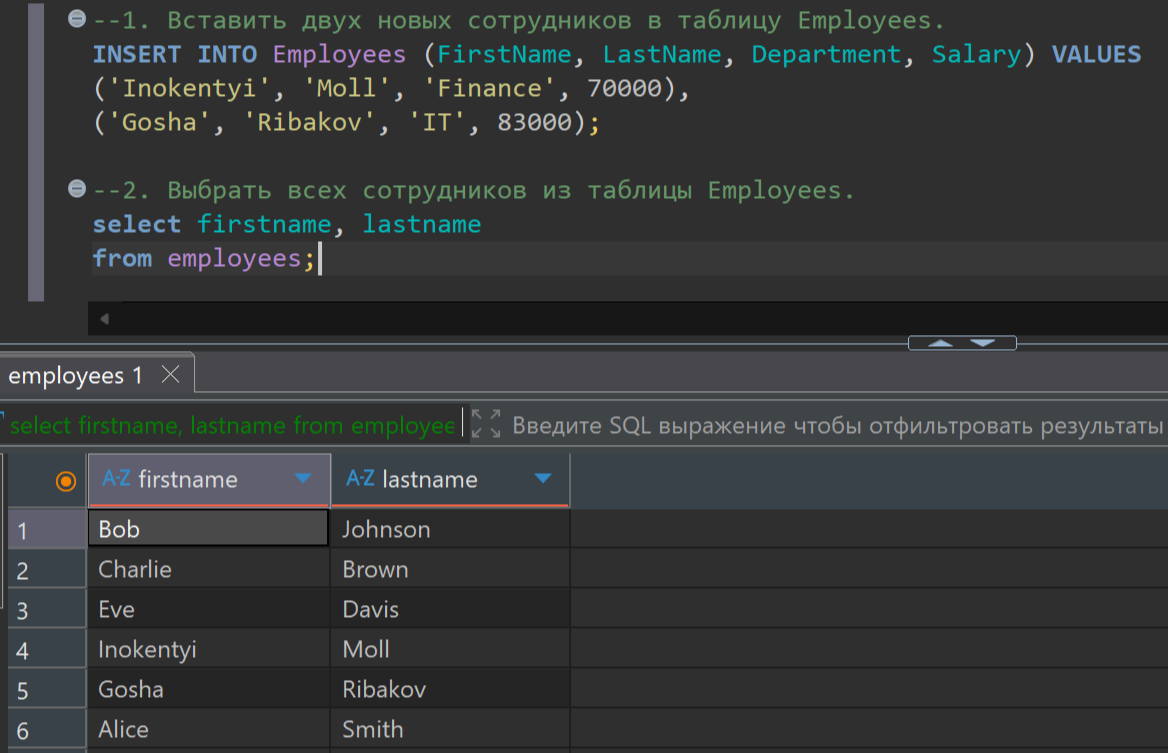
2. Выбрать всех сотрудников из таблицы Employees.

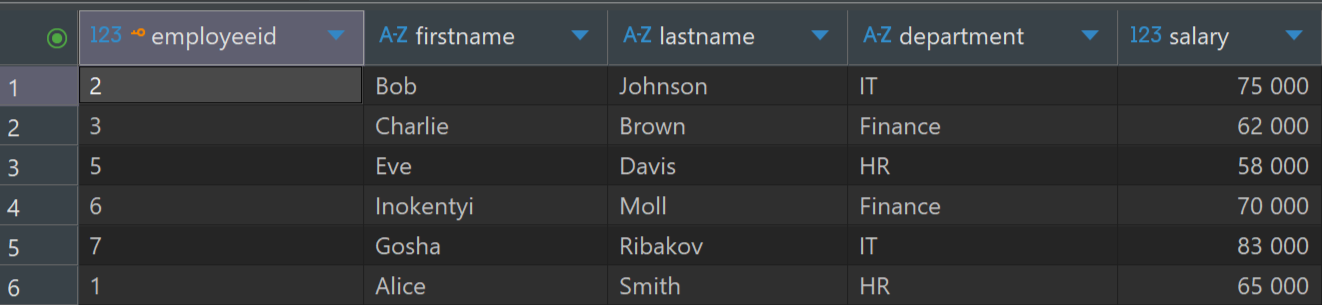
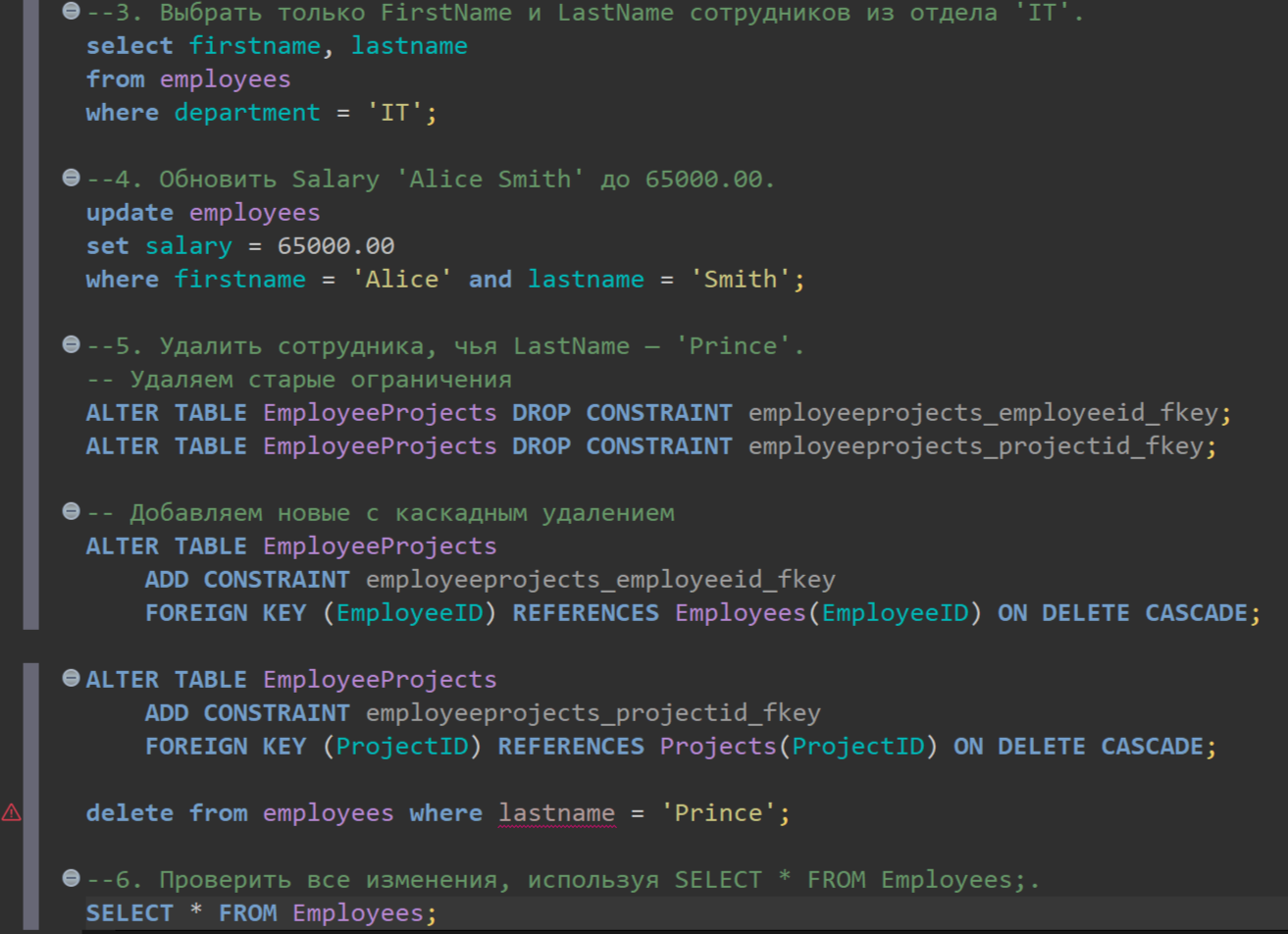
3. Выбрать только FirstName и LastName сотрудников из отдела 'IT'.

4. Обновить Salary 'Alice Smith' до 65000.00.

5. Удалить сотрудника, чья LastName — 'Prince'.

6. Проверить все изменения, используя SELECT \* FROM Employees;.





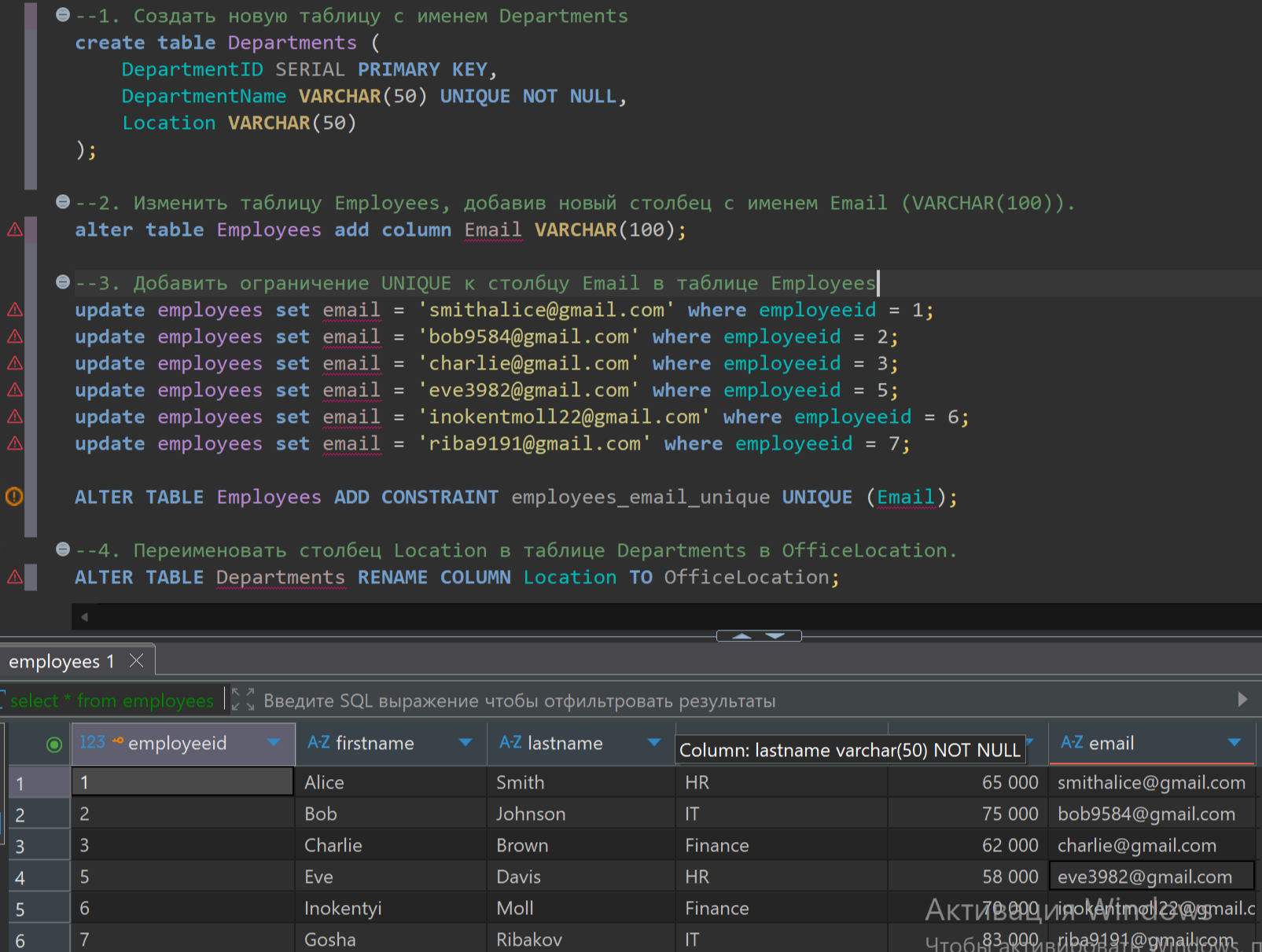
Задание 2: DDL

1. Создать новую таблицу с именем Departments со столбцами: DepartmentID (SERIAL PRIMARY KEY), DepartmentName (VARCHAR(50), UNIQUE, NOT NULL), Location (VARCHAR(50)).

2. Изменить таблицу Employees, добавив новый столбец с именем Email (VARCHAR(100)).

3. Добавить ограничение UNIQUE к столбцу Email в таблице Employees, предварительно заполнив любыми значениями.

4. Переименовать столбец Location в таблице Departments в OfficeLocation.



Задание 3: DCL

1. Создать нового пользователя PostgreSQL (роль) с именем hr\_user и простым паролем.

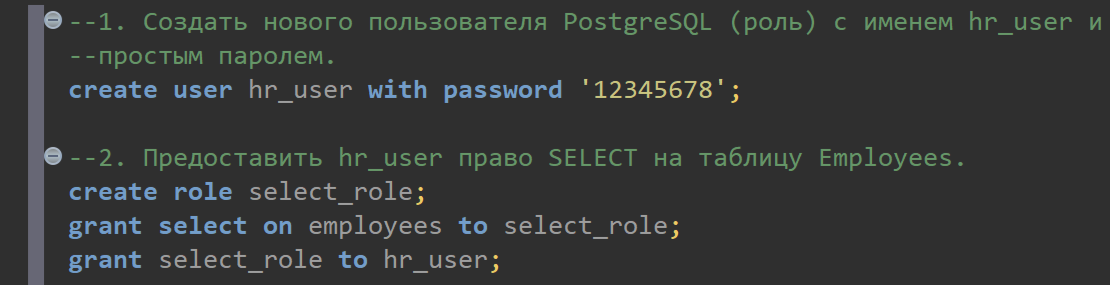
2. Предоставить hr\_user право SELECT на таблицу Employees.

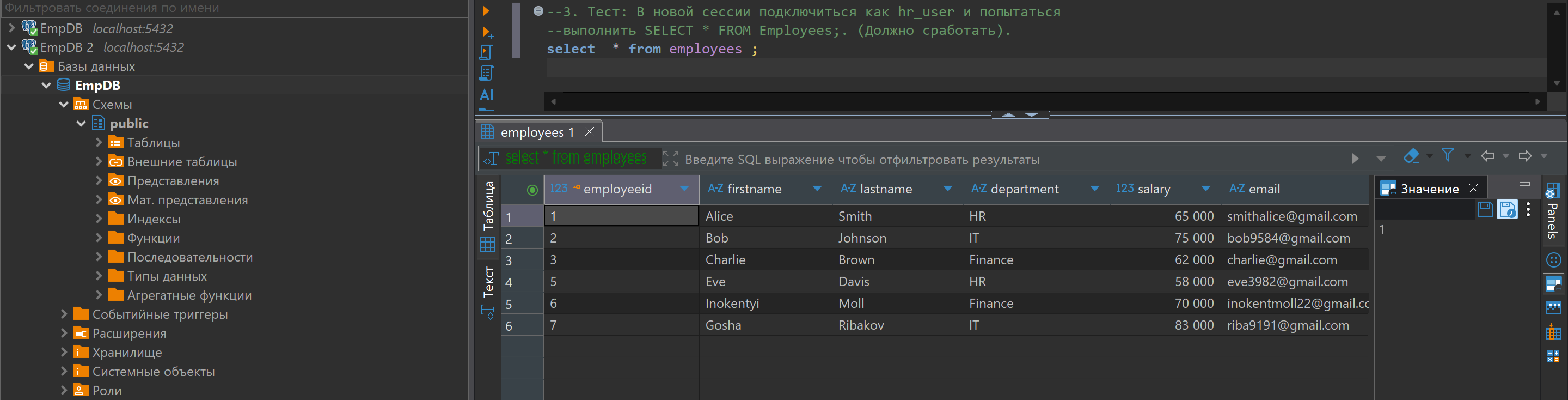
3. Тест: В новой сессии подключиться как hr\_user и попытаться выполнить SELECT \* FROM Employees;. (Должно сработать).

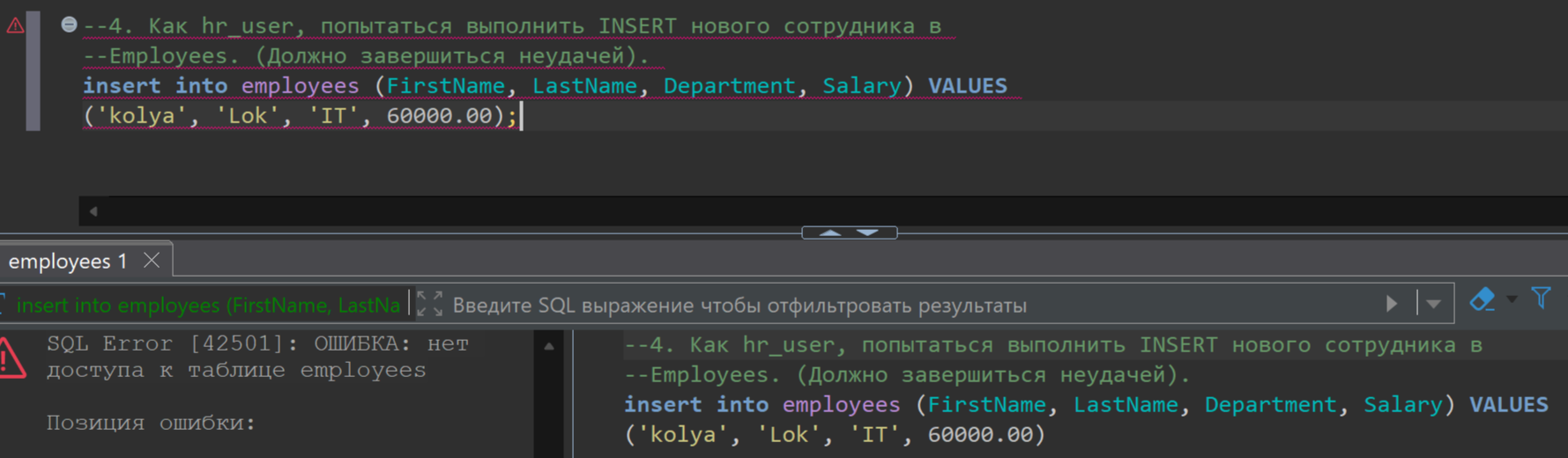
4. Как hr\_user, попытаться выполнить INSERT нового сотрудника в Employees. (Должно завершиться неудачей).

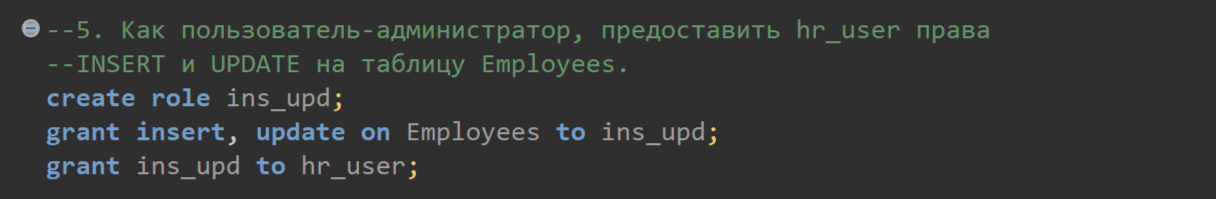
5. Как пользователь-администратор, предоставить hr\_user права INSERT и UPDATE на таблицу Employees.

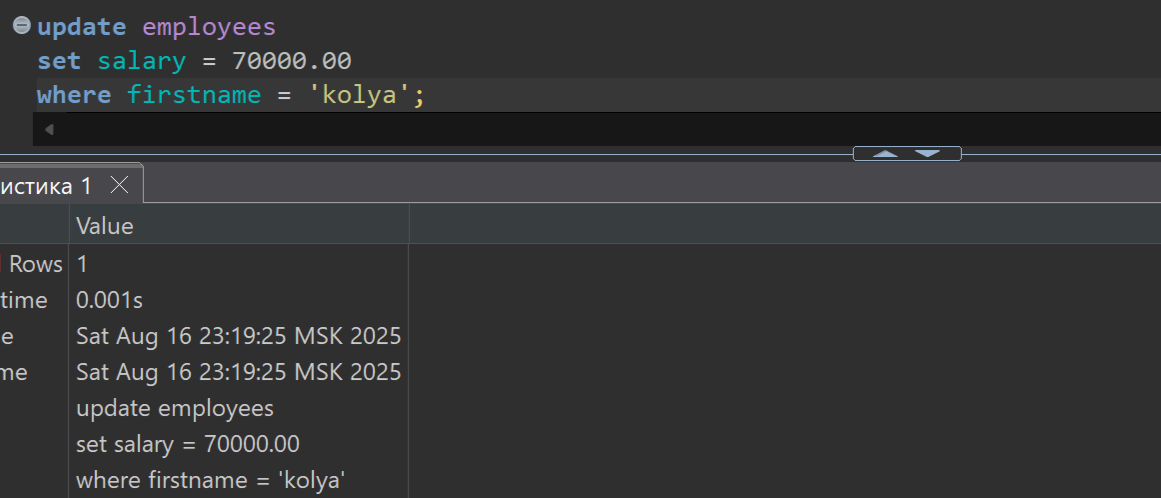
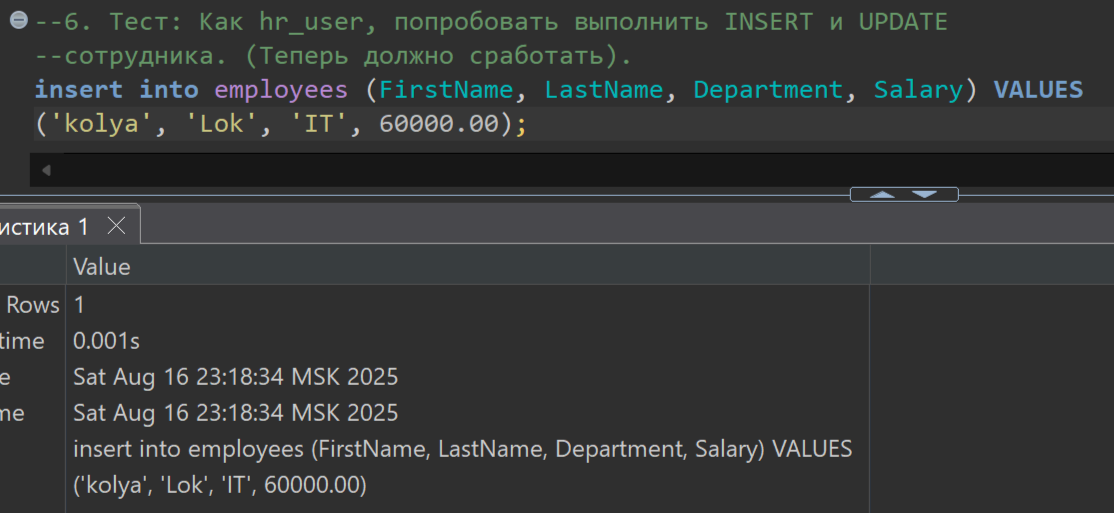
6. Тест: Как hr\_user, попробовать выполнить INSERT и UPDATE сотрудника. (Теперь должно сработать).











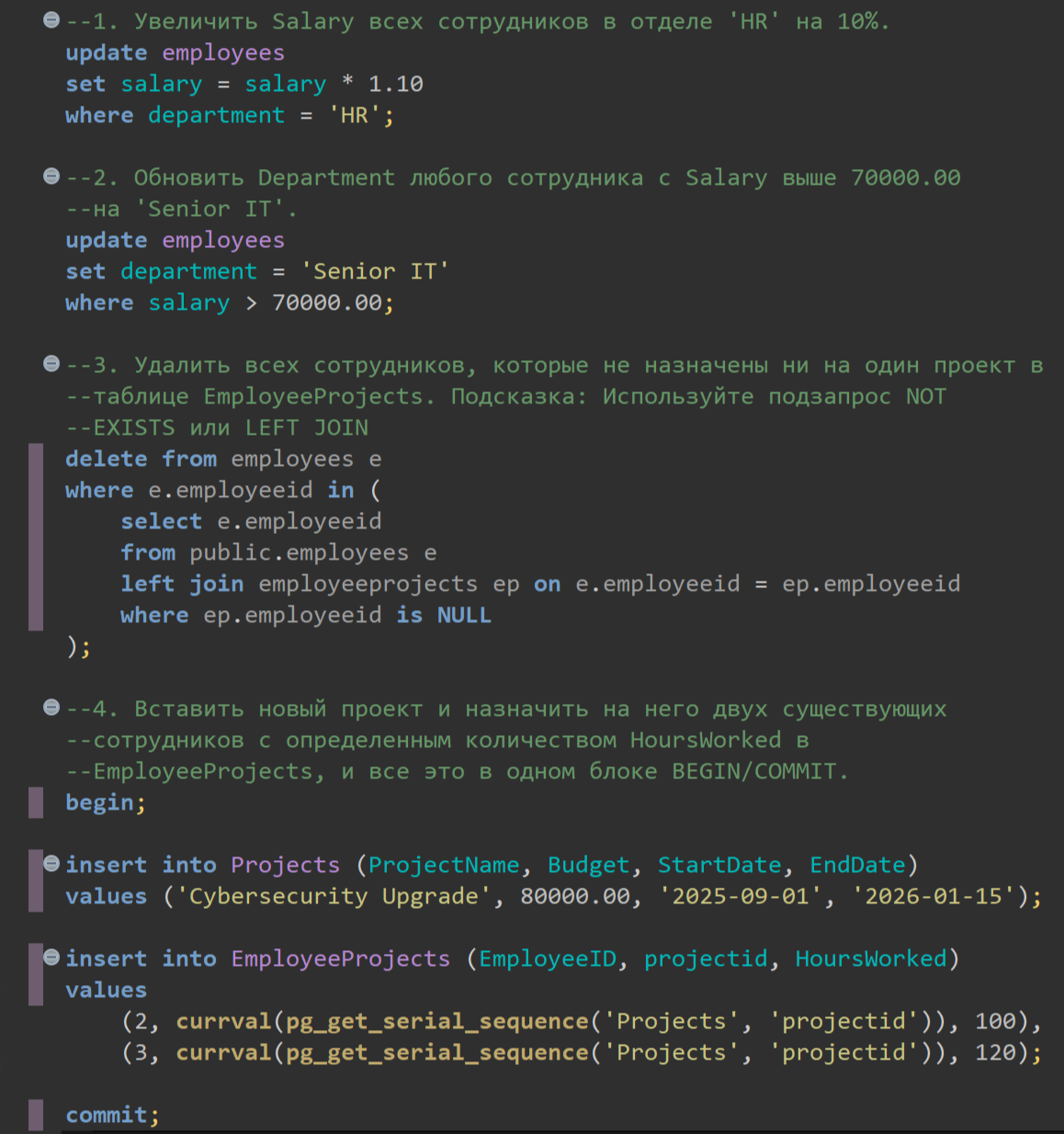
Задание 4: DML/DCL ●

1. Увеличить Salary всех сотрудников в отделе 'HR' на 10%.

2. Обновить Department любого сотрудника с Salary выше 70000.00 на 'Senior IT'.

3. Удалить всех сотрудников, которые не назначены ни на один проект в таблице EmployeeProjects. Подсказка: Используйте подзапрос NOT EXISTS или LEFT JOIN

4. Вставить новый проект и назначить на него двух существующих сотрудников с определенным количеством HoursWorked в EmployeeProjects, и все это в одном блоке BEGIN/COMMIT.



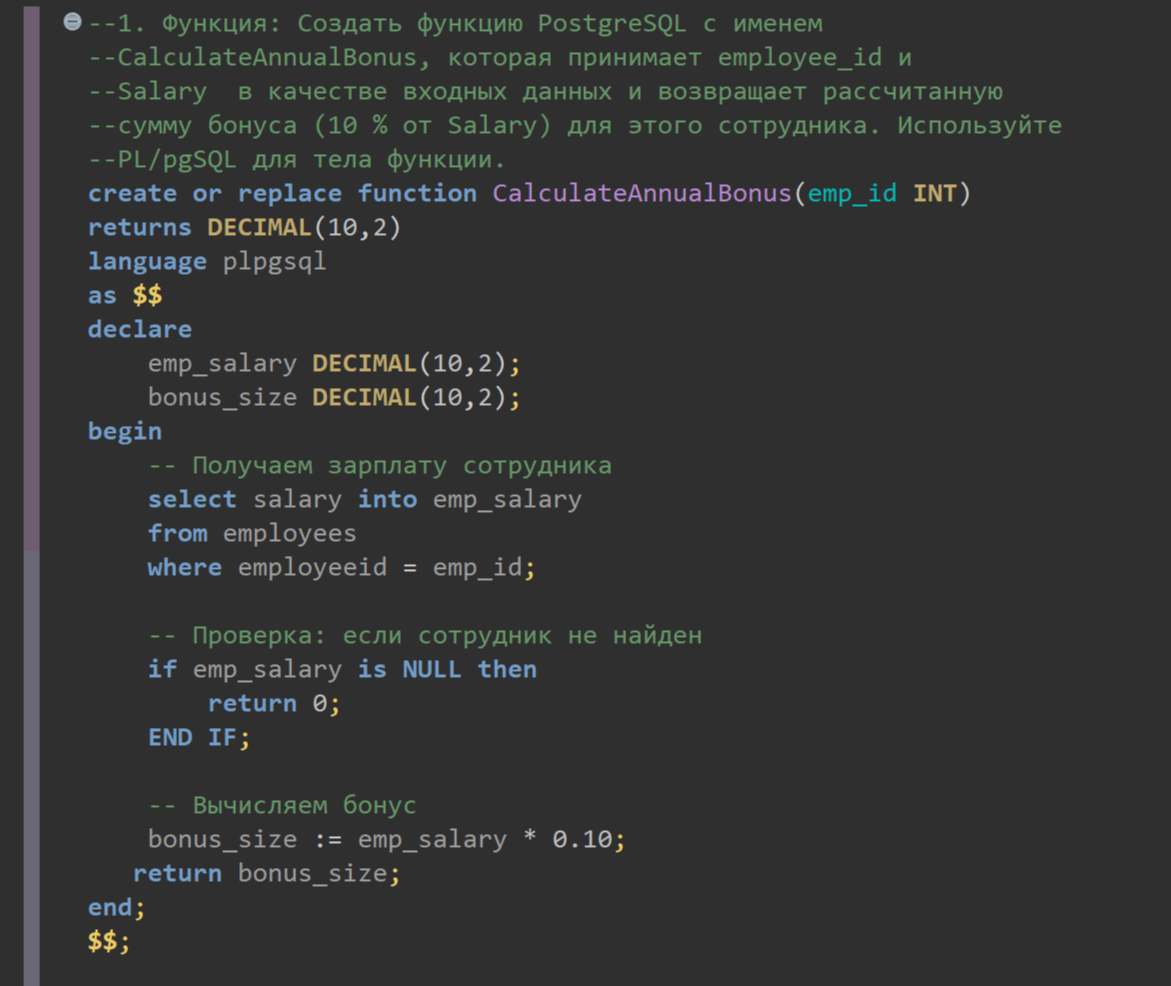
Задание 5:

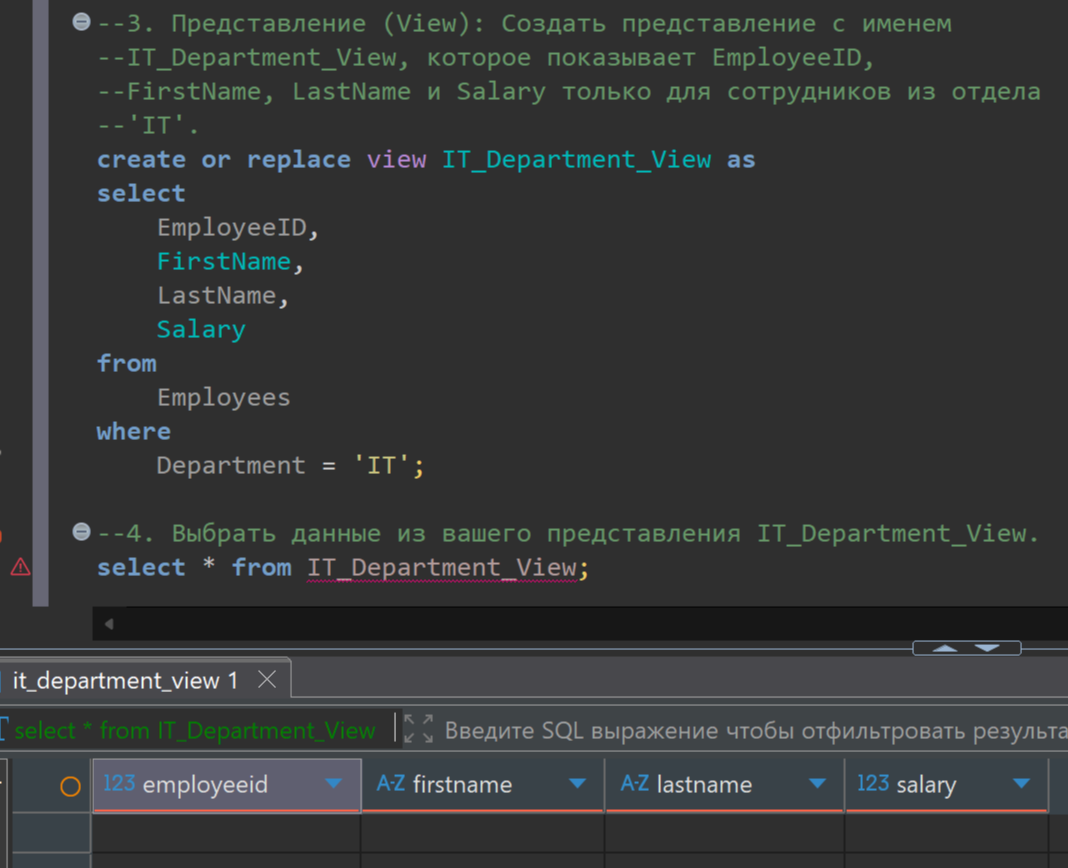
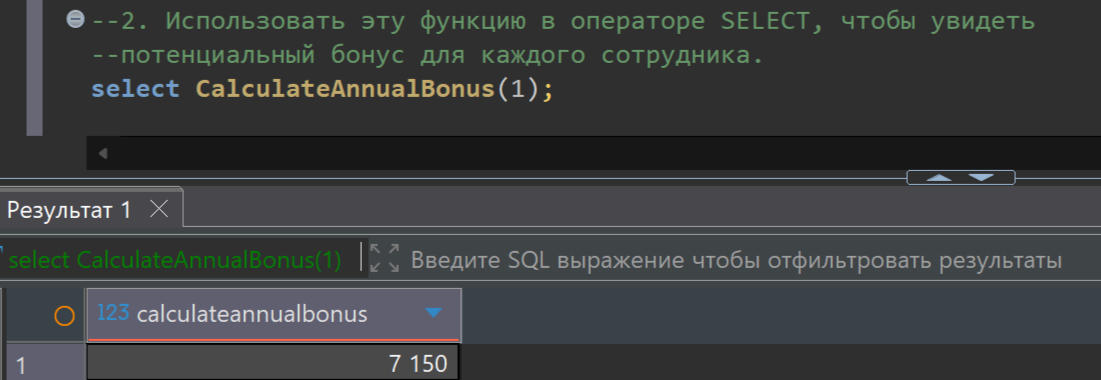
1. Функция: Создать функцию PostgreSQL с именем CalculateAnnualBonus, которая принимает employee\_id и Salary в качестве входных данных и возвращает рассчитанную сумму бонуса (10 % от Salary) для этого сотрудника. Используйте PL/pgSQL для тела функции.

2. Использовать эту функцию в операторе SELECT, чтобы увидеть потенциальный бонус для каждого сотрудника.

3. Представление (View): Создать представление с именем IT\_Department\_View, которое показывает EmployeeID, FirstName, LastName и Salary только для сотрудников из отдела 'IT'.

4. Выбрать данные из вашего представления IT\_Department\_View.





Задание 6: DML (Optional)

1. Найти ProjectName всех проектов, в которых 'Bob Johnson' работал более 150 часов.

2. Увеличить Budget всех проектов на 10%, если к ним назначен хотя бы один сотрудник из отдела 'IT'.

3. Для любого проекта, у которого еще нет EndDate (EndDate IS NULL), установить EndDate на один год позже его StartDate.

