| МИНОБРНАУКИ РОССИИ |
| --- |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра цифровой трансформации (ЦТ)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Разработка баз данных»

**Практическое занятие № 1**

| Студенты группы ИНБО-01-17 | *ИКБО-42-23 Голев С.С.*  *.* | (подпись) | |
| --- | --- | --- | --- |
| Ассистент | *Морозов Д.В.* | (подпись) | |
| Отчет представлен | «\_\_\_»\_\_\_\_\_\_\_\_2025 г. | |  | |

# **СОДЕРЖАНИЕ**

[СОДЕРЖАНИЕ](#_qtztl25p8b4o) 2

[ПОСТАНОВКА ЗАДАЧИ](#_gve1tcvxsn07) 3

[ХОД ВЫПОЛНЕНИЯ РАБОТЫ](#_xh9isxirxoe5) 4

[Создание структуры данных](#_xq6l4tlju8gq) 4

[Элементы списка выборки – SELECT](#_x07xmoqaknfk) 6

[Составление запросов на выборку](#_ljss3wxy3ck9) 9

# 

# **ПОСТАНОВКА ЗАДАЧИ**

Для выполнения практической работы необходимо последовательно выполнить следующие шаги, основываясь на логической модели данных, которая была спроектирована в рамках курса «Проектирование баз данных» в предыдущем семестре:

1. На основе логической модели данных, созданной в прошлом семестре, письменно описать не менее 5 различных бизнес-правил и не менее 3 ограничений целостности для таблиц. Выбор бизнес-правил и ограничений целостности производится на усмотрение студента. Результаты представить в виде таблицы.

2. С использованием DDL-оператора CREATE TABLE создать все необходимые таблицы (согласно созданной в прошлом семестре логической модели данных) в СУБД Postgres Pro, корректно реализовав все описанные ограничения целостности.

3. Заполнить созданные таблицы согласованными тестовыми данными (не менее 5-7 записей на таблицу, где это применимо) с помощью оператора INSERT INTO.

4. Составить и выполнить не менее 6 SQL-запросов к таблицам, иллюстрирующих использование различных операторов SELECT и WHERE, согласно перечню, указанному в задании (см. Ход выполнения работы). В составленных запросах должны быть использованы все приведённые примеры – .

5. Составить и выполнить по два SQL-запроса к таблицам для демонстрации работы предложений ORDER BY, GROUP BY и HAVING.

6. Каждый SQL-запрос сопроводить комментарием, объясняющим его назначение и логику работы.

# **ХОД ВЫПОЛНЕНИЯ РАБОТЫ**

Для таблицы ordering из логической модели, составим описание атрибутов и ограничений.

*Таблица 1 – Описание ограничений для таблицы ordering*

| Название | Тип данных | Ограничение | Бизнес-правило |
| --- | --- | --- | --- |
| id | INT | PRIMARY | Уникальный идентификатор заказа |
| total | TEXT | NOT NULL, CHECK (>0) | Обязательная цена не равная нулю |
| date | INT | NOT NULL, CHECK  (date >= CURRENT\_DATE) | Дата доставки продукта, не может быть в прошлом |
| status | TEXT | NOT NULL | Обязательное наличие статуса заказа |
| delivery\_method | INT | NOT NULL | Обязательный метод заказа |

## Создание структуры данных

С помощью оператора CREATE TABLE создадим таблицу основываясь на логической модели.

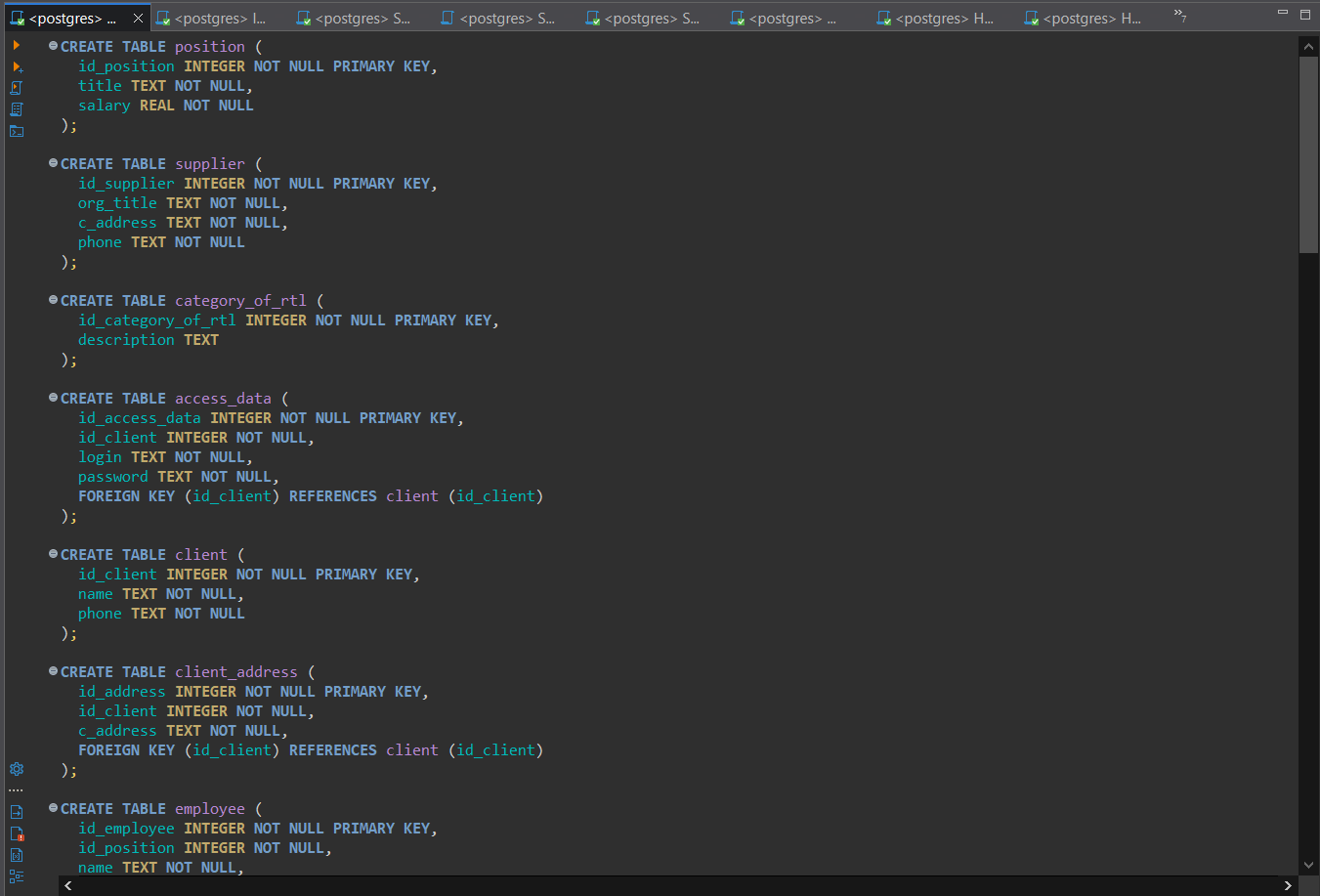


Рисунок 1 – Скрипт для создания таблиц

Заполним эту таблицу данными с помощью оператора INSERT INTO.

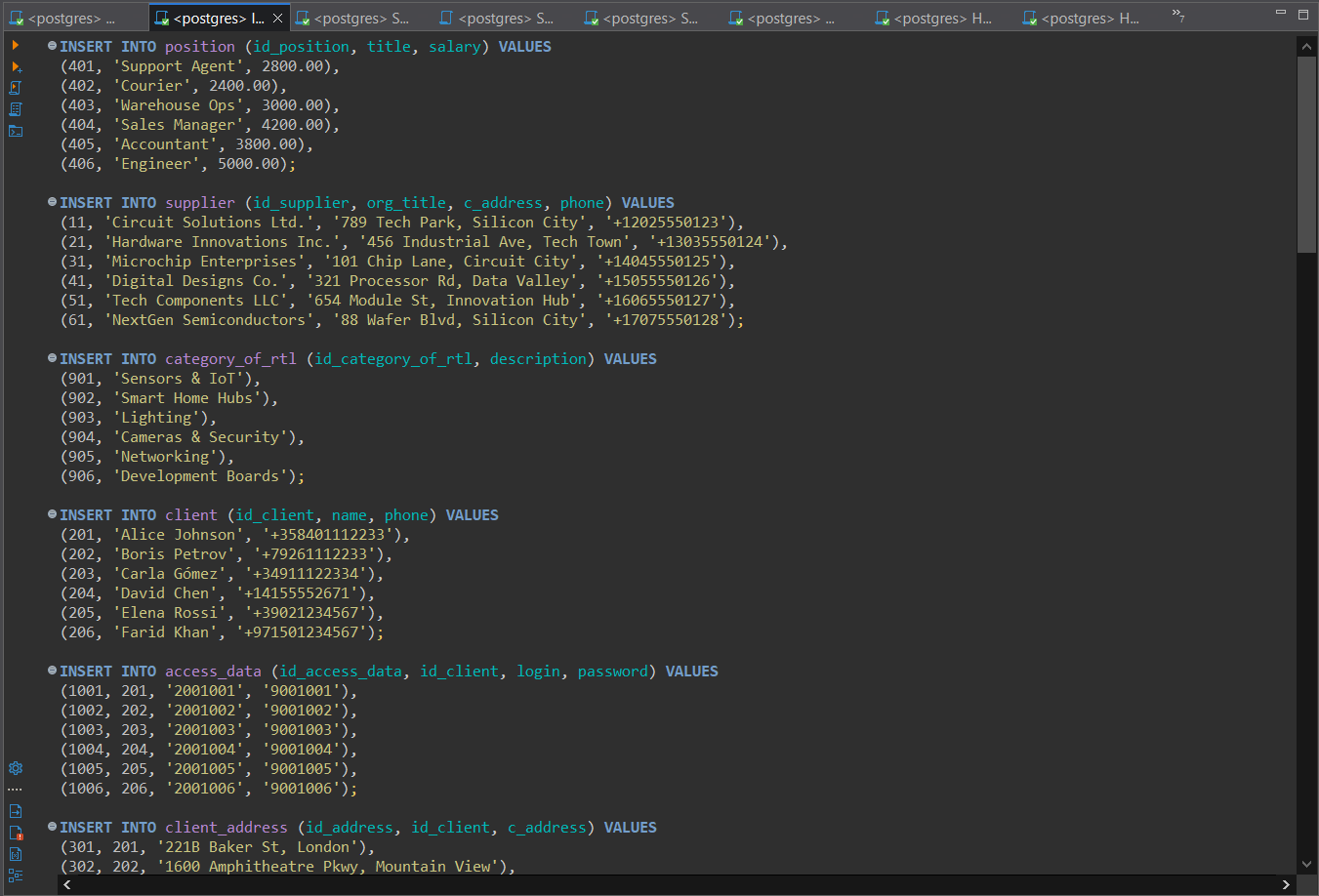


Рисунок 2 – Скрипт для заполнения таблиц

## Элементы списка выборки – SELECT

Напишем скрипты с помощью оператора SELECT для разных сценариев.

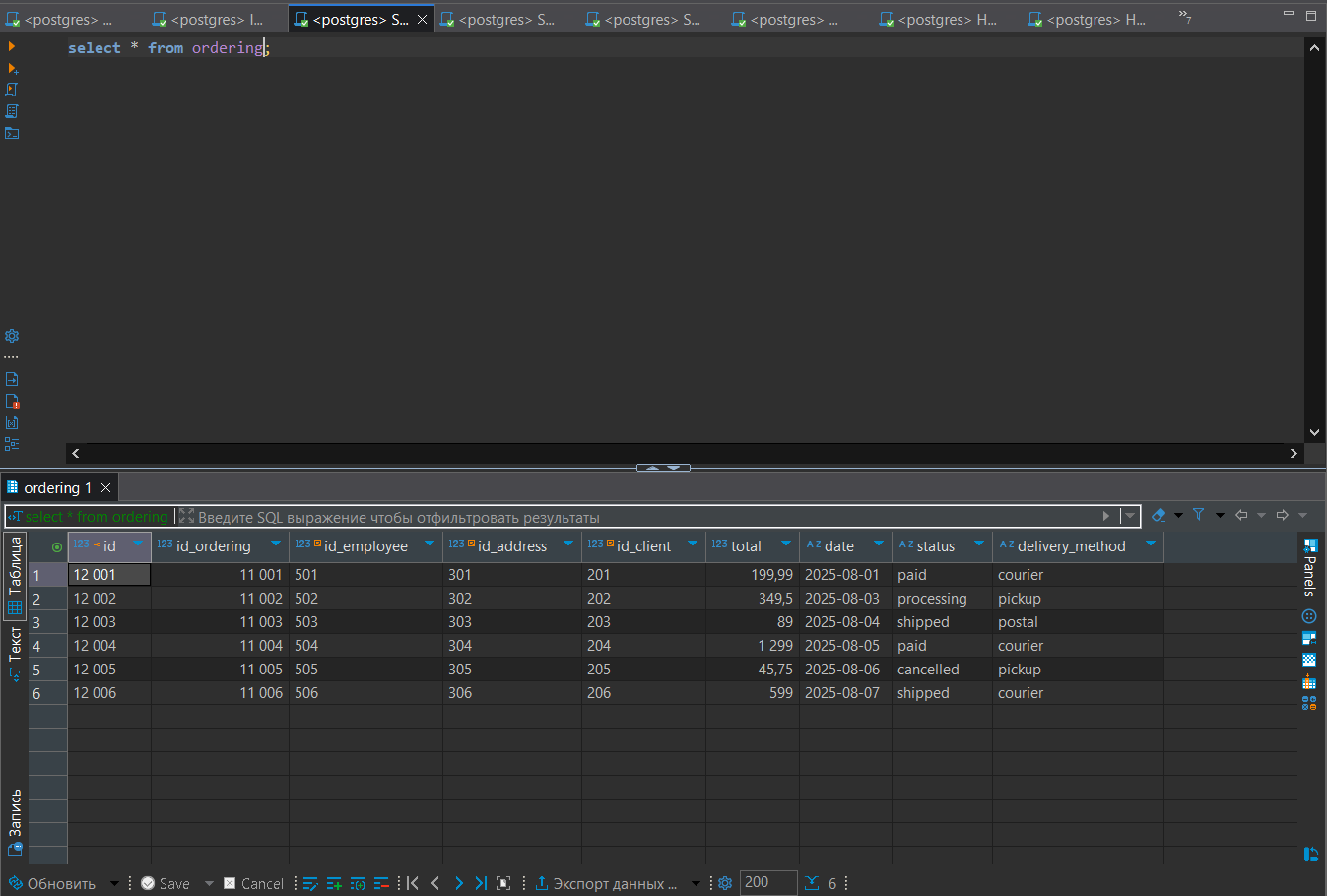


Рисунок 3 – SELECT скрипт для вывода всего содержимого таблицы.

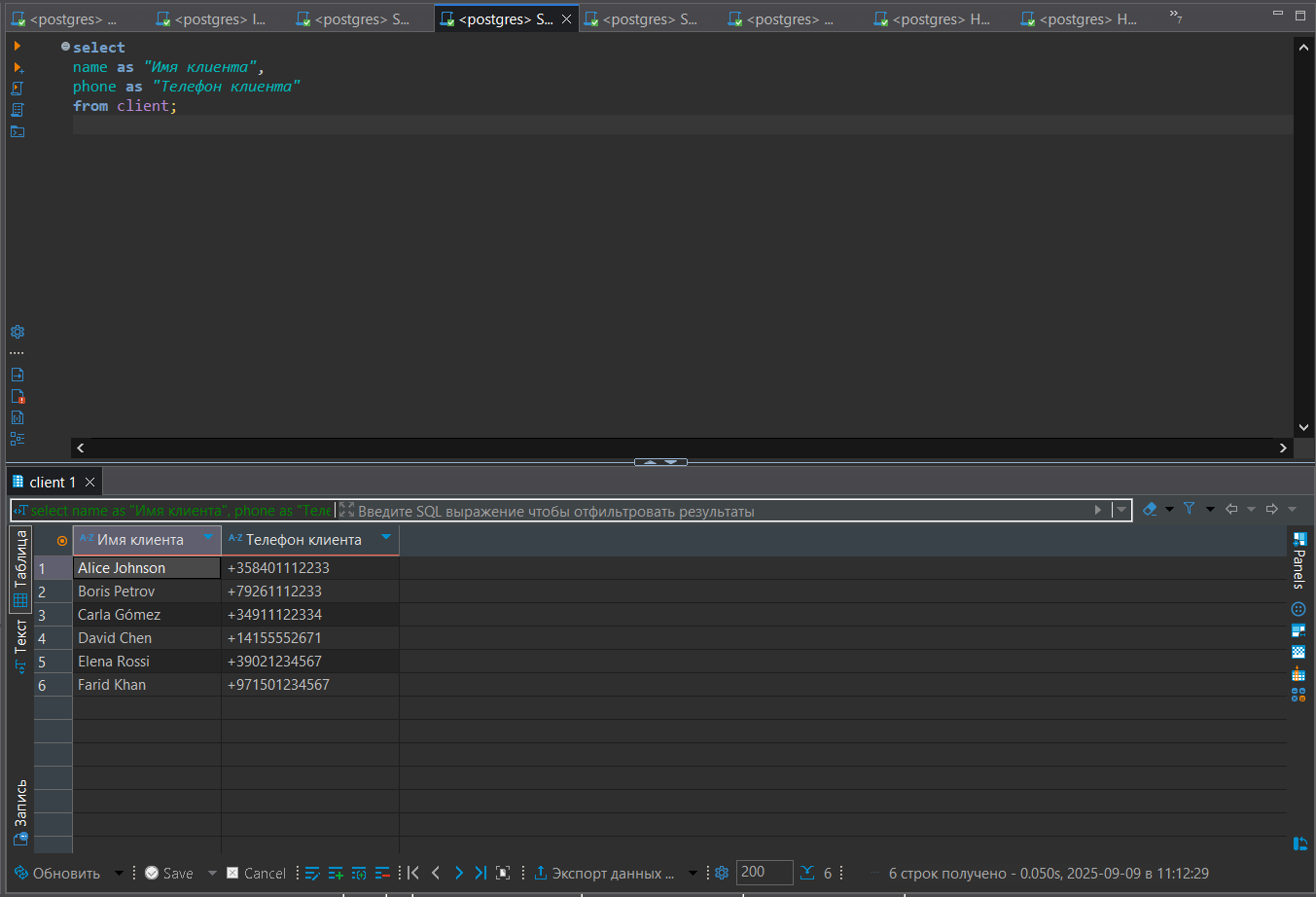


Рисунок 4 – SELECT скрипт для вывода конкретных полей таблицы.

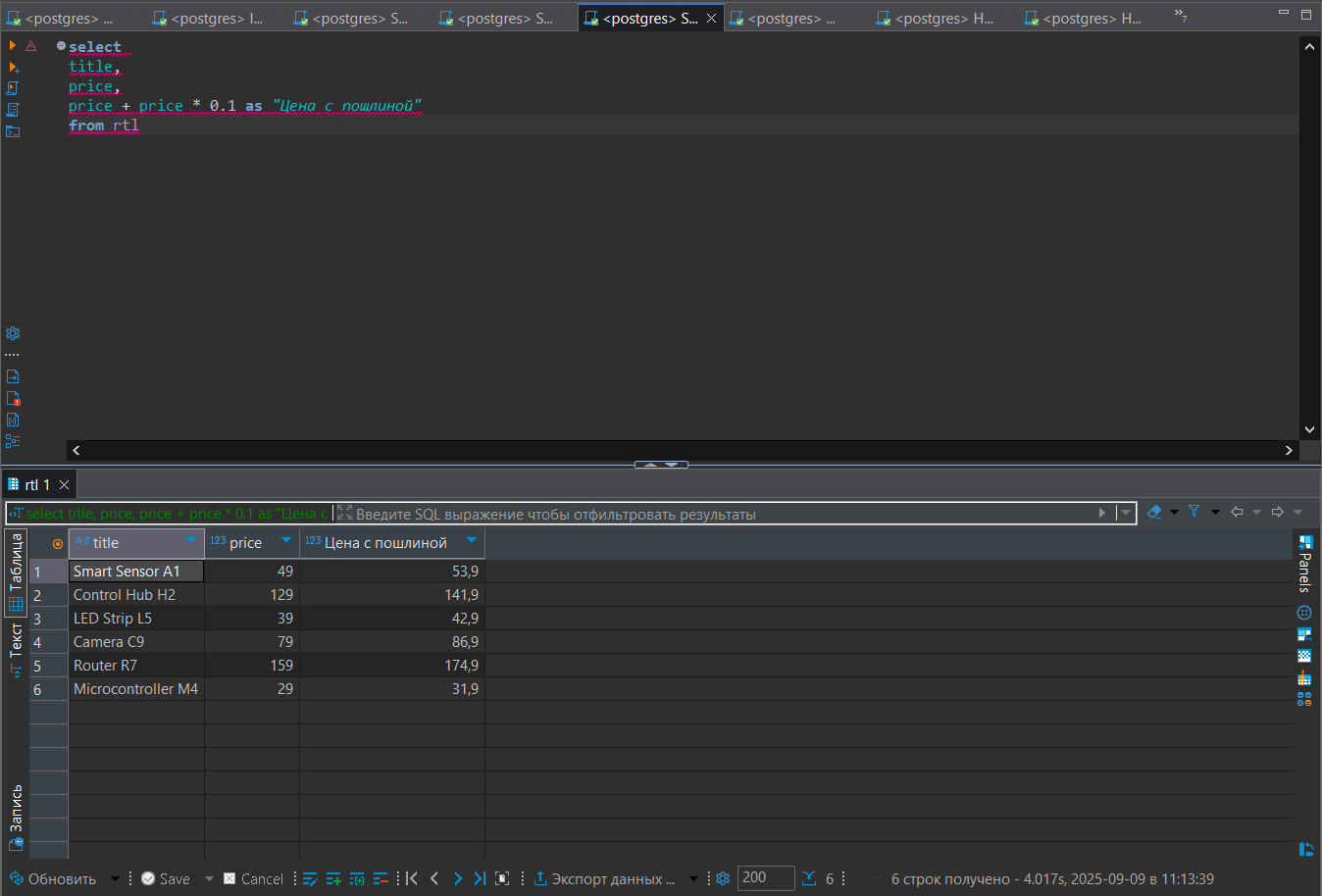


Рисунок 5 – SELECT скрипт для вывода конкретных полей таблицы с использованием математических операций.

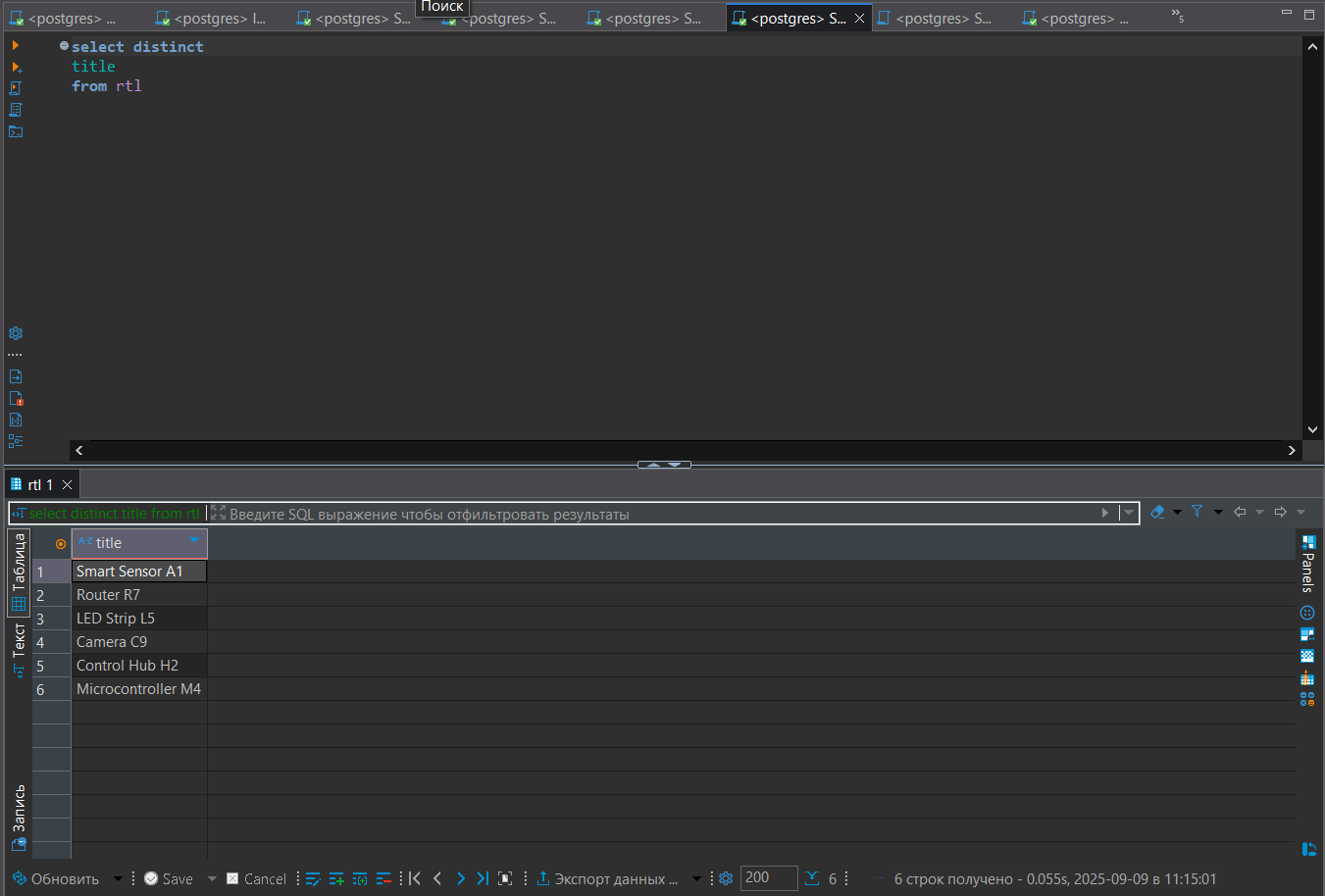


Рисунок 6 – SELECT скрипт для вывода уникальных полей таблицы.

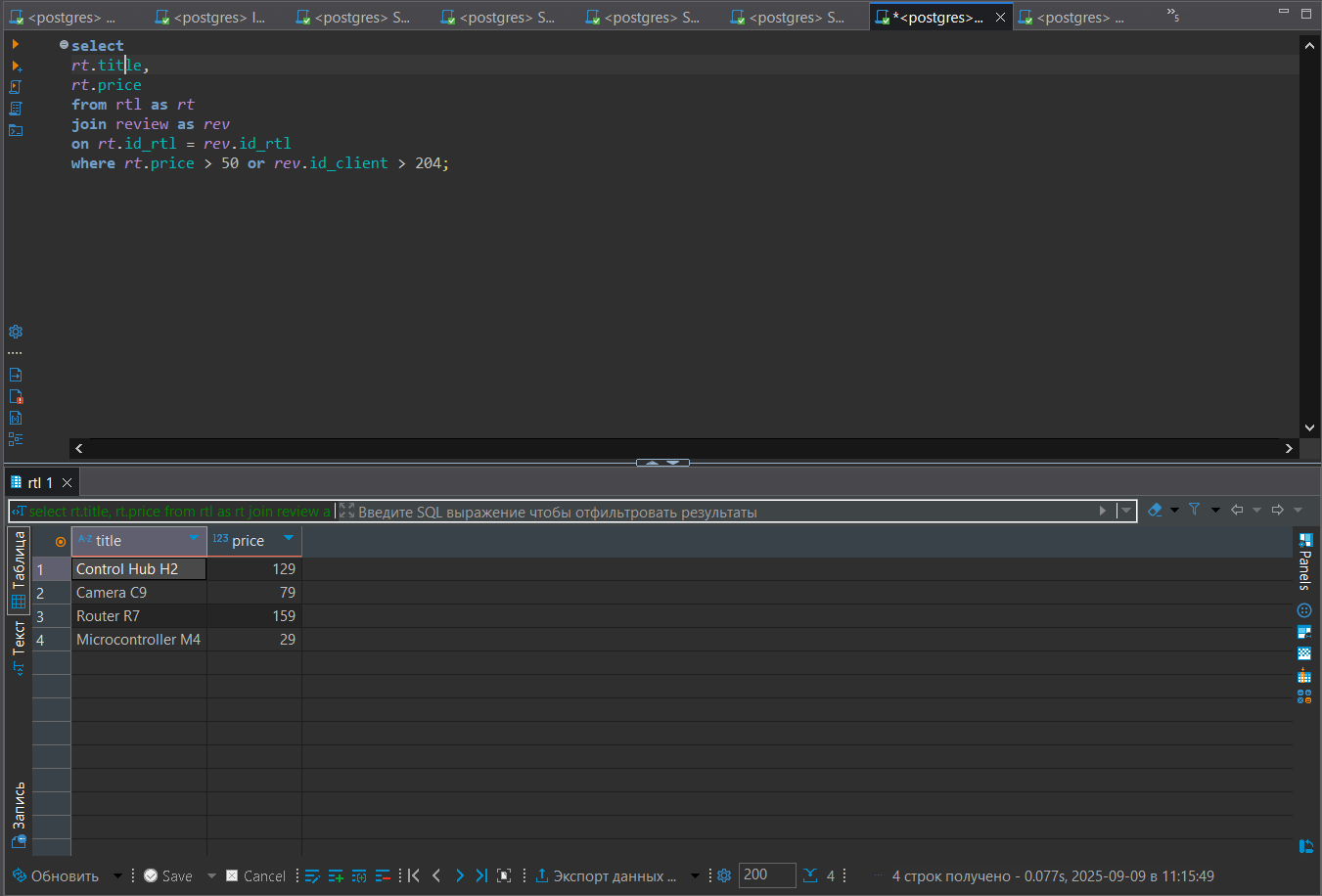


Рисунок 7 – SELECT скрипт для вывода полей таблицы в которых присутствует ‘er.

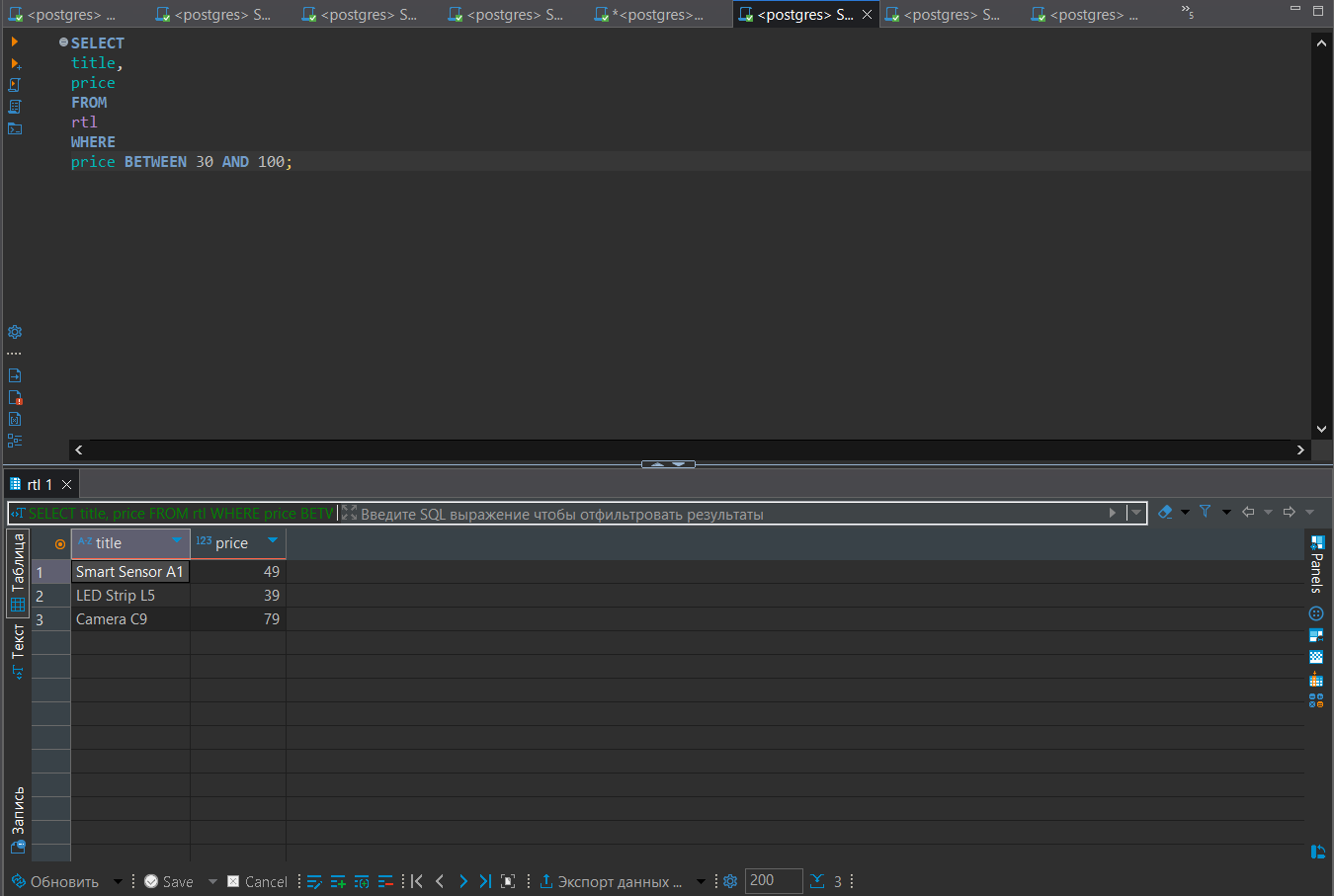


Рисунок 8 – SELECT скрипт для вывода полей таблицы соответствующих выбранному диапазону.

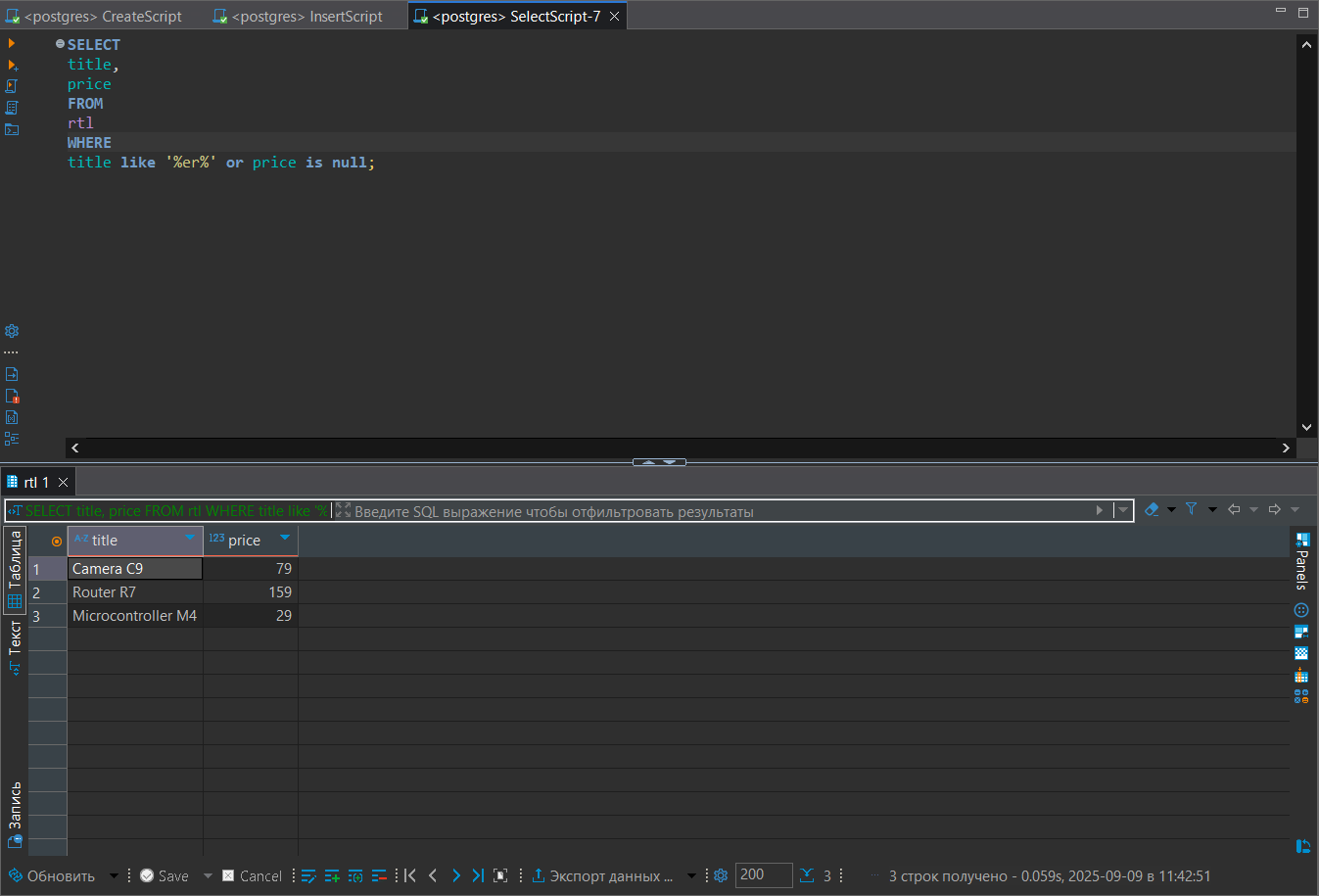


Рисунок 8.1 – SELECT скрипт для вывода полей таблицы с помощью операторов LIKE и IS NULL.

## Составление запросов на выборку

Напишем скрипты используя операторы ий ORDER BY, GROUP BY и HAVING.

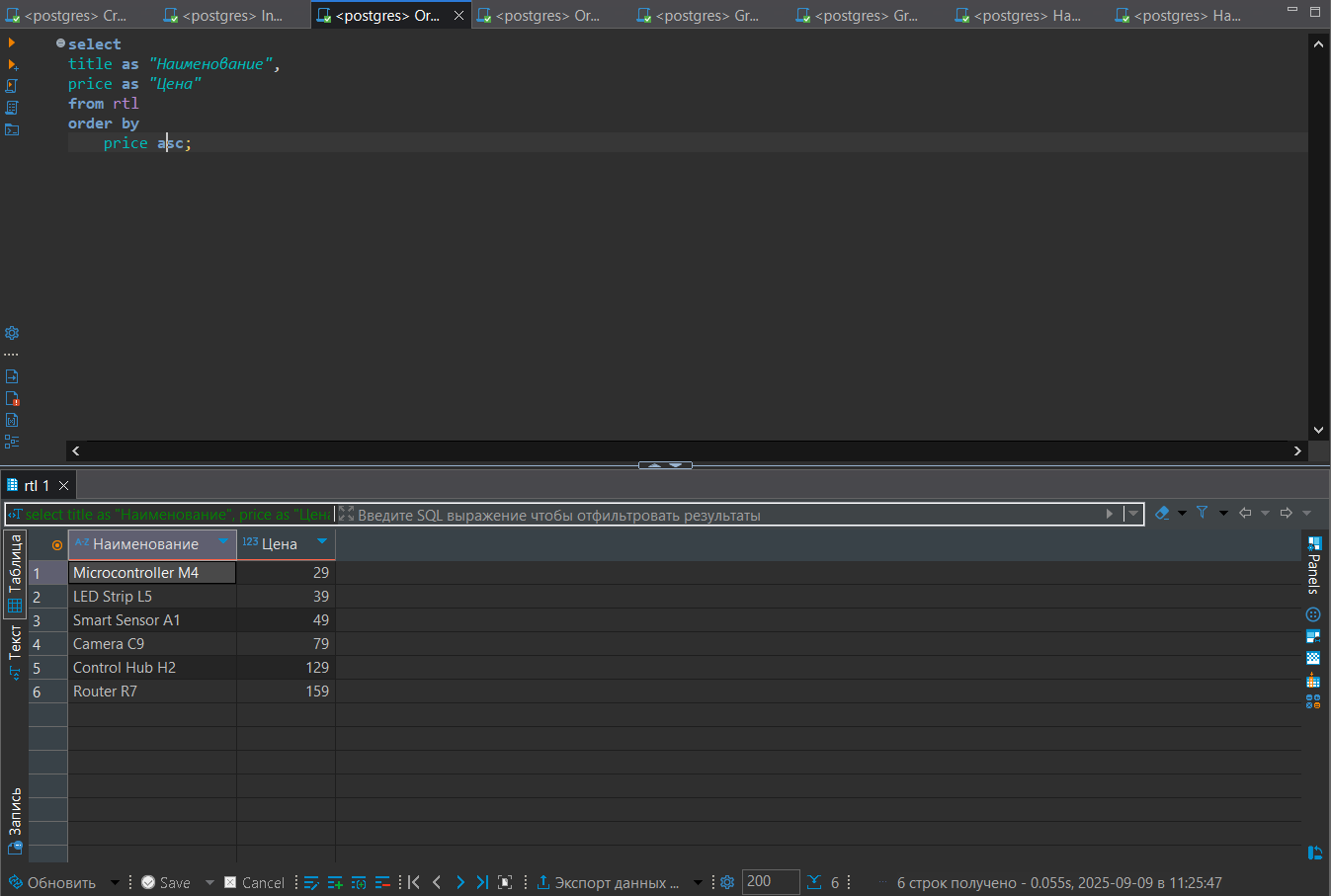


Рисунок 9 – ORDER BY скрипт с использованием оператора ASC.

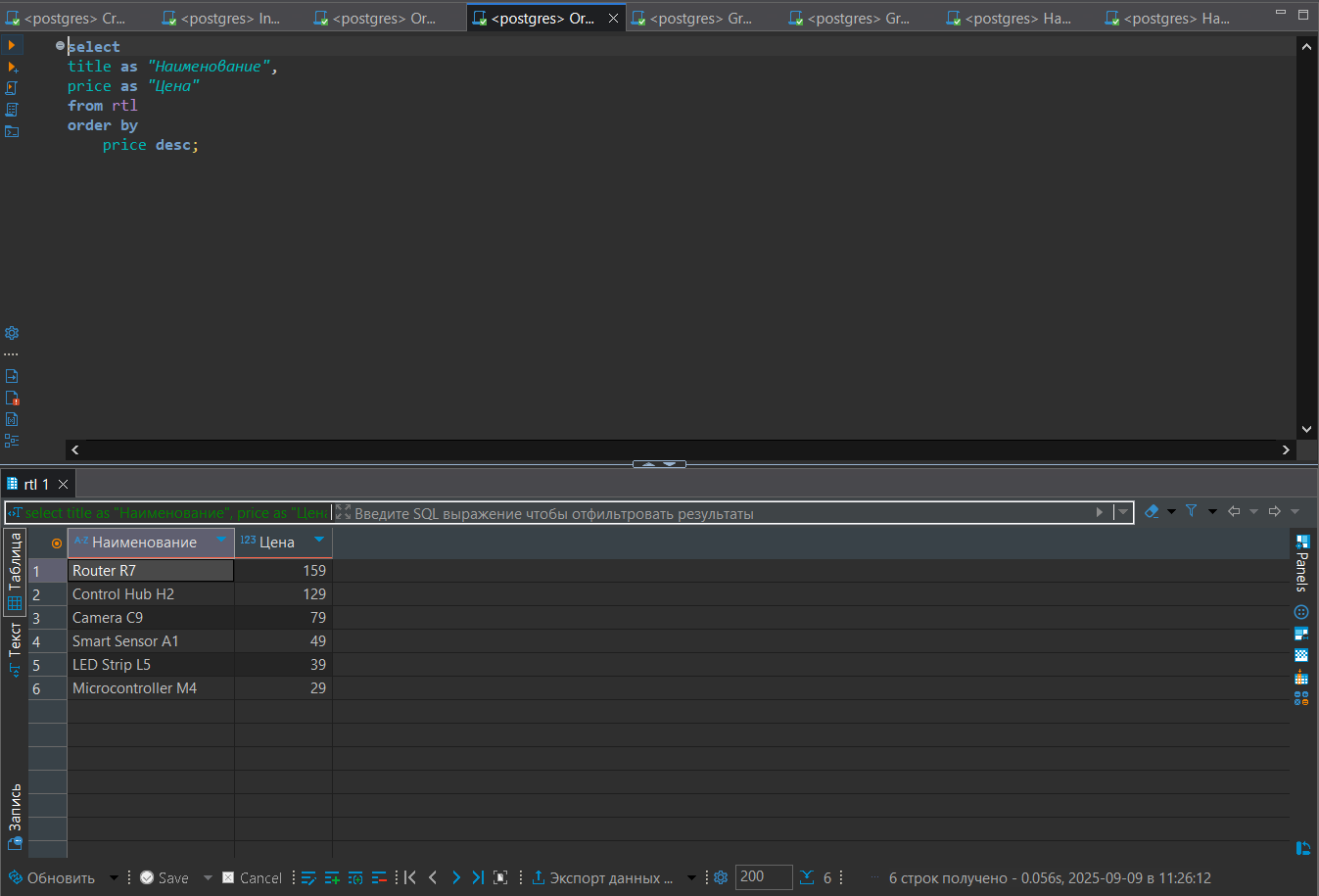


Рисунок 10 – ORDER BY скрипт с использованием оператора DESC.

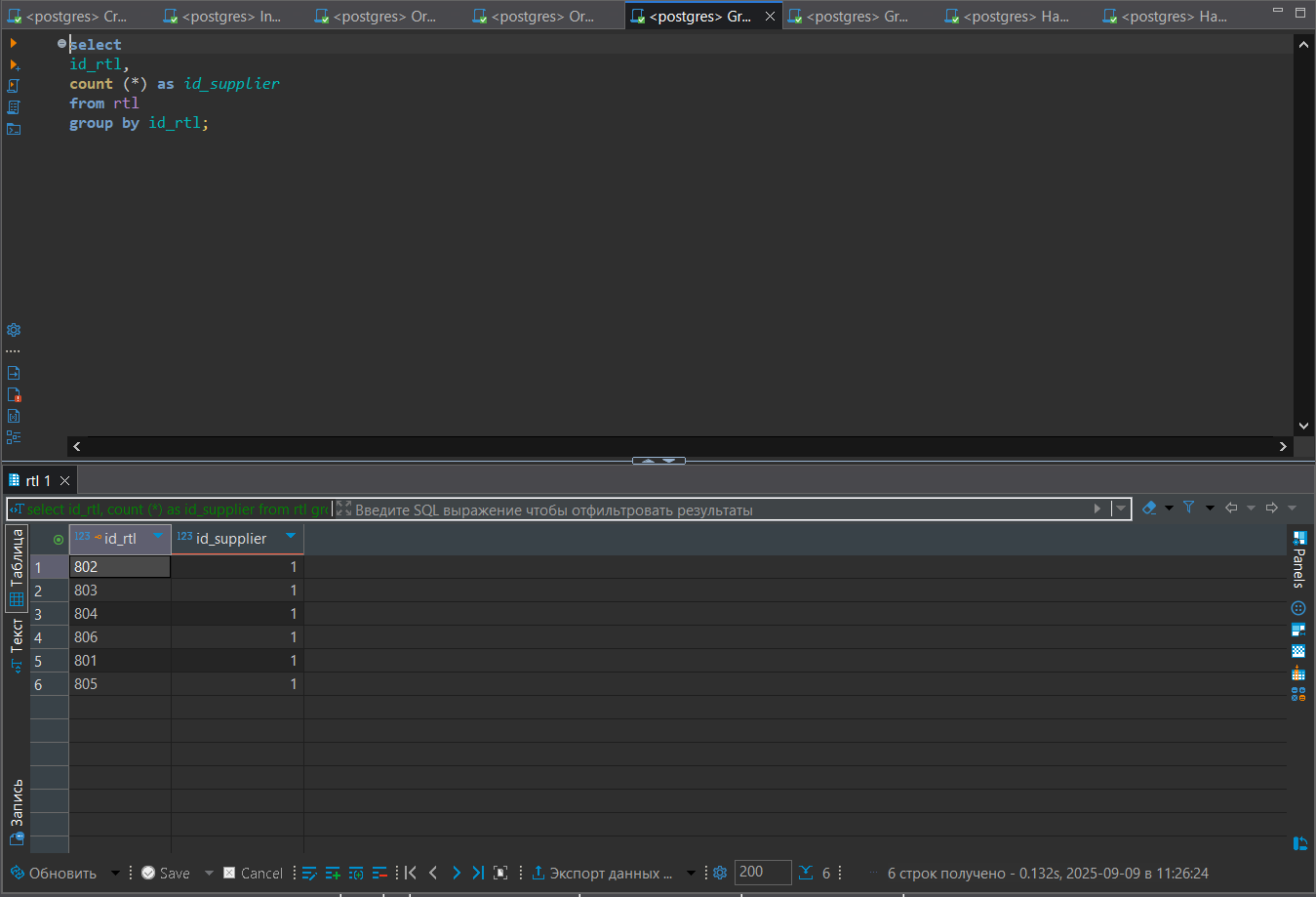


Рисунок 11 – GROUP BY скрипт с использованием оператора COUNT.

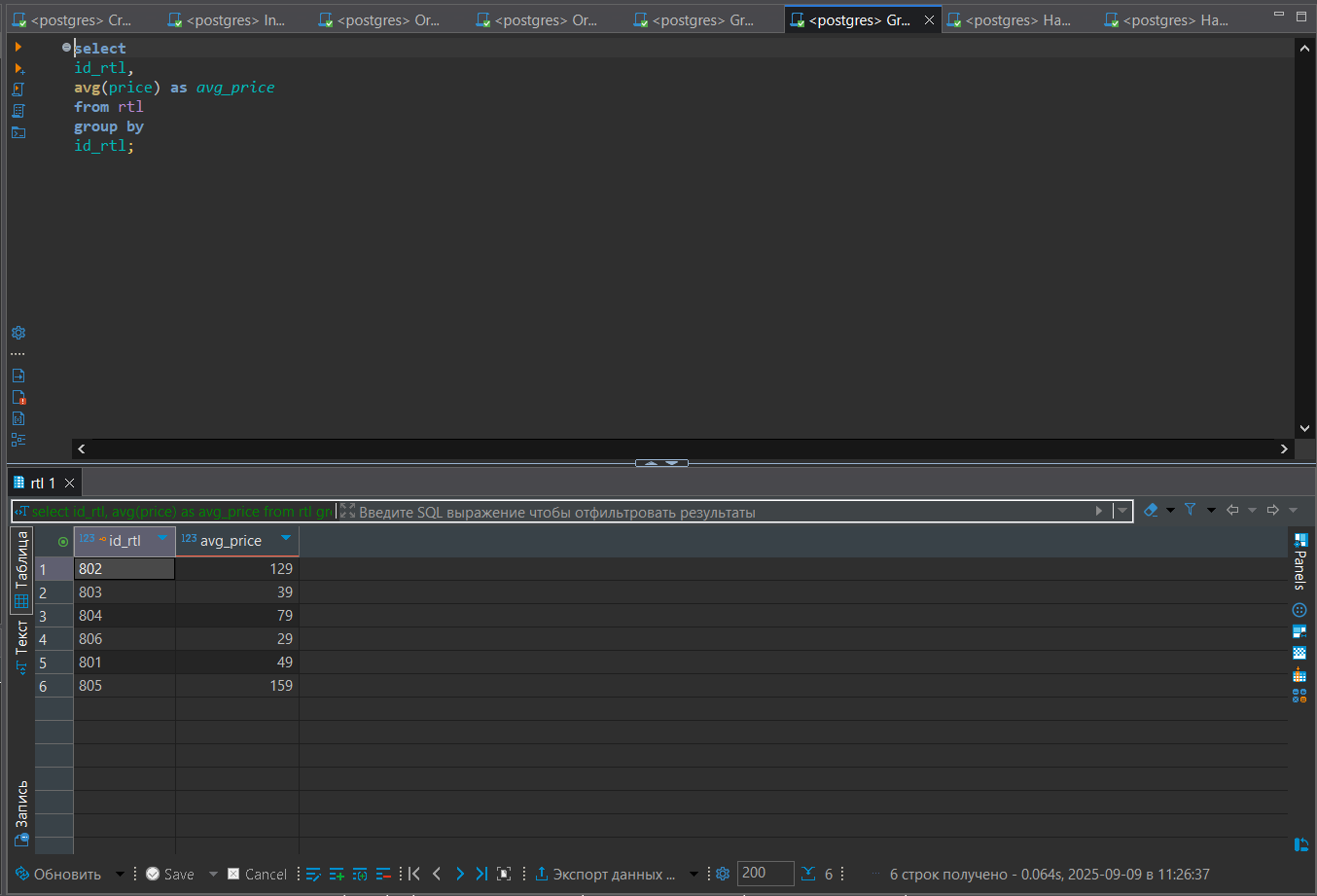


Рисунок 11 – GROUP BY скрипт с использованием оператора AVG.

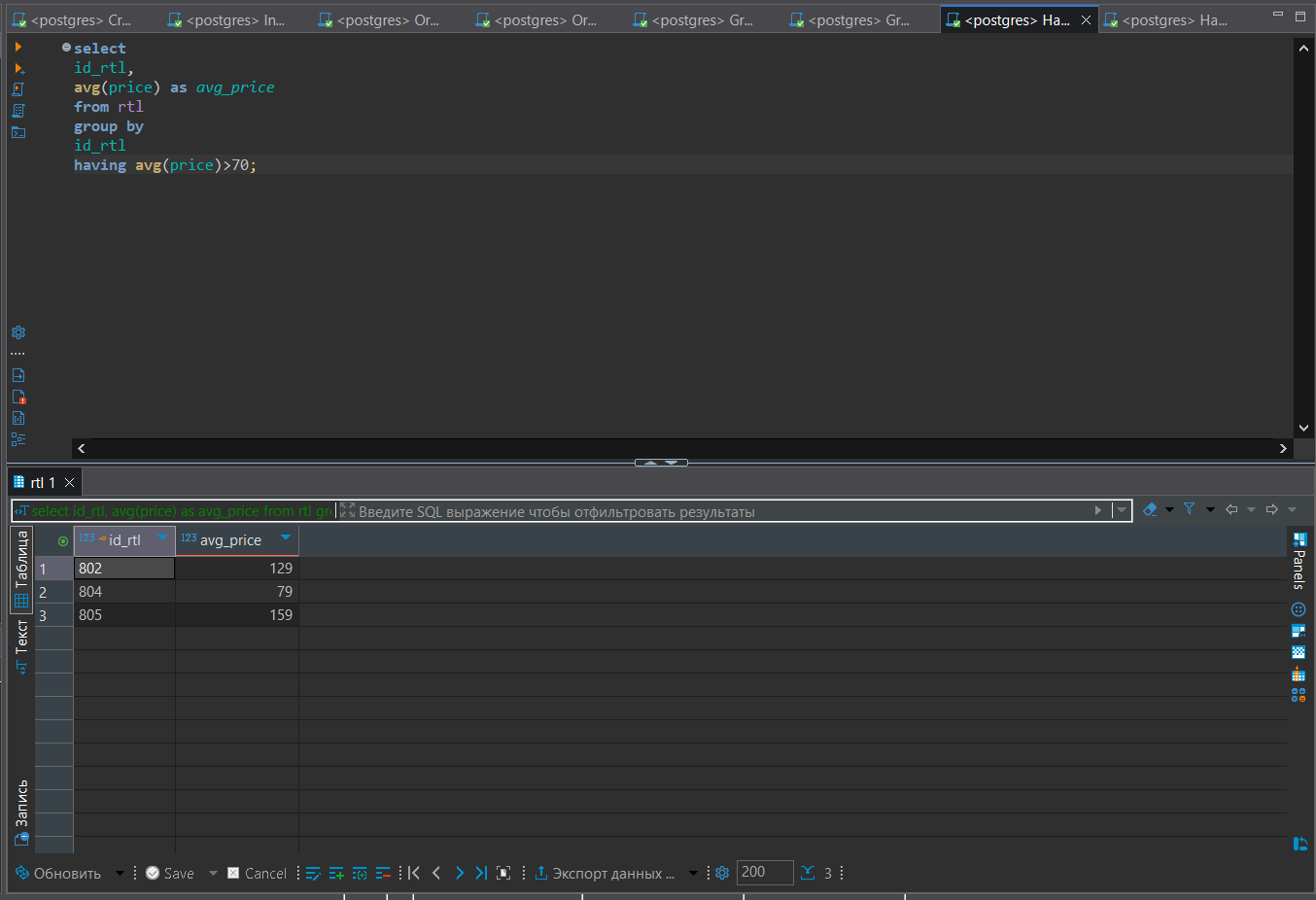


Рисунок 12 – HAVING скрипт с использованием оператора AVG.

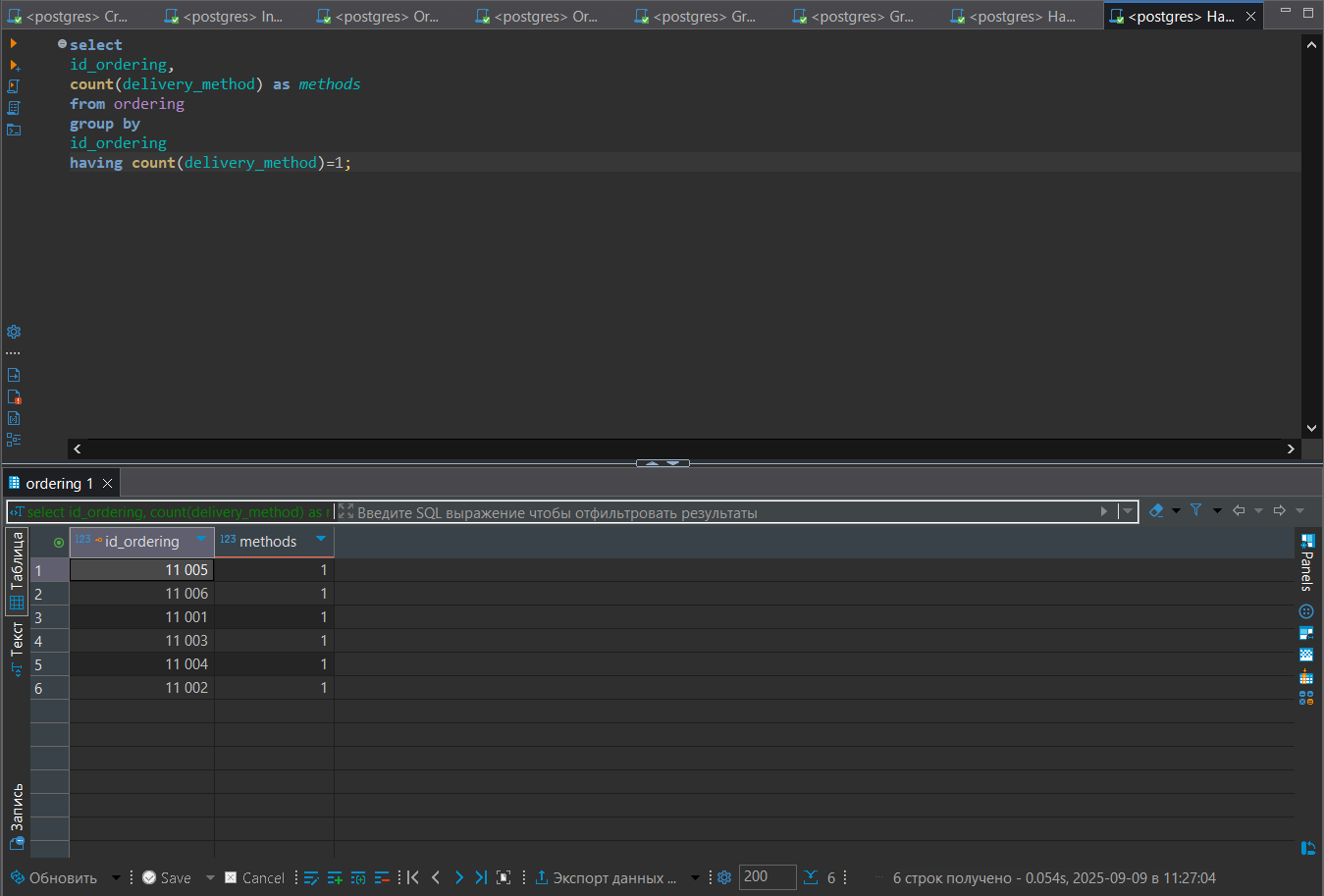


Рисунок 13 – HAVING скрипт с использованием оператора COUNT.