Отчет по практической работе №2

Предмет: Теория баз данных  
Практическая работа №2  
Название работы: Работа с операторами SELECT, JOIN и другими в PostgreSQL

Группа: БИСО-02-23  
ФИО: Макаревич Сергей Витальевич  
Дата: []

# Задание

1. Создать 2 сущности: таблицы «клиент» (client) и «товар» (item).  
2. Заполнить таблицы данными.  
3. Выполнить запросы с использованием операторов UNION, INTERSECT, EXCEPT, CROSS JOIN, DISTINCT.  
4. Рассмотреть варианты связей между таблицами.

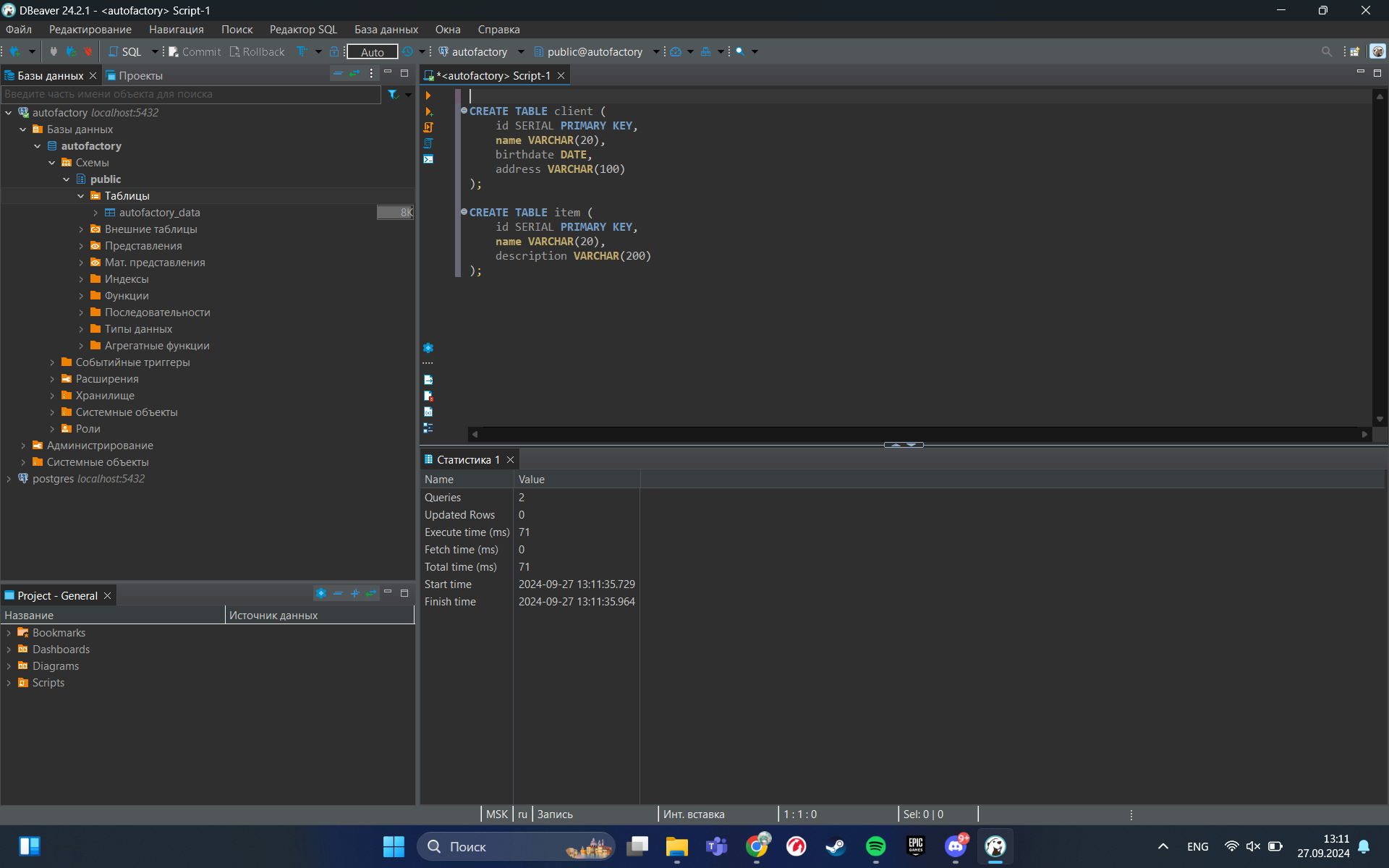
# Теоретическая часть

SQL операторы:  
- UNION и UNION ALL — объединяют результаты двух выборок.  
- INTERSECT — возвращает пересечение двух выборок.  
- EXCEPT — возвращает разность двух выборок.  
- CROSS JOIN — выполняет декартово произведение двух таблиц.  
- DISTINCT — возвращает уникальные значения.  
- JOIN — используется для соединения данных из нескольких таблиц.

# Практическая часть

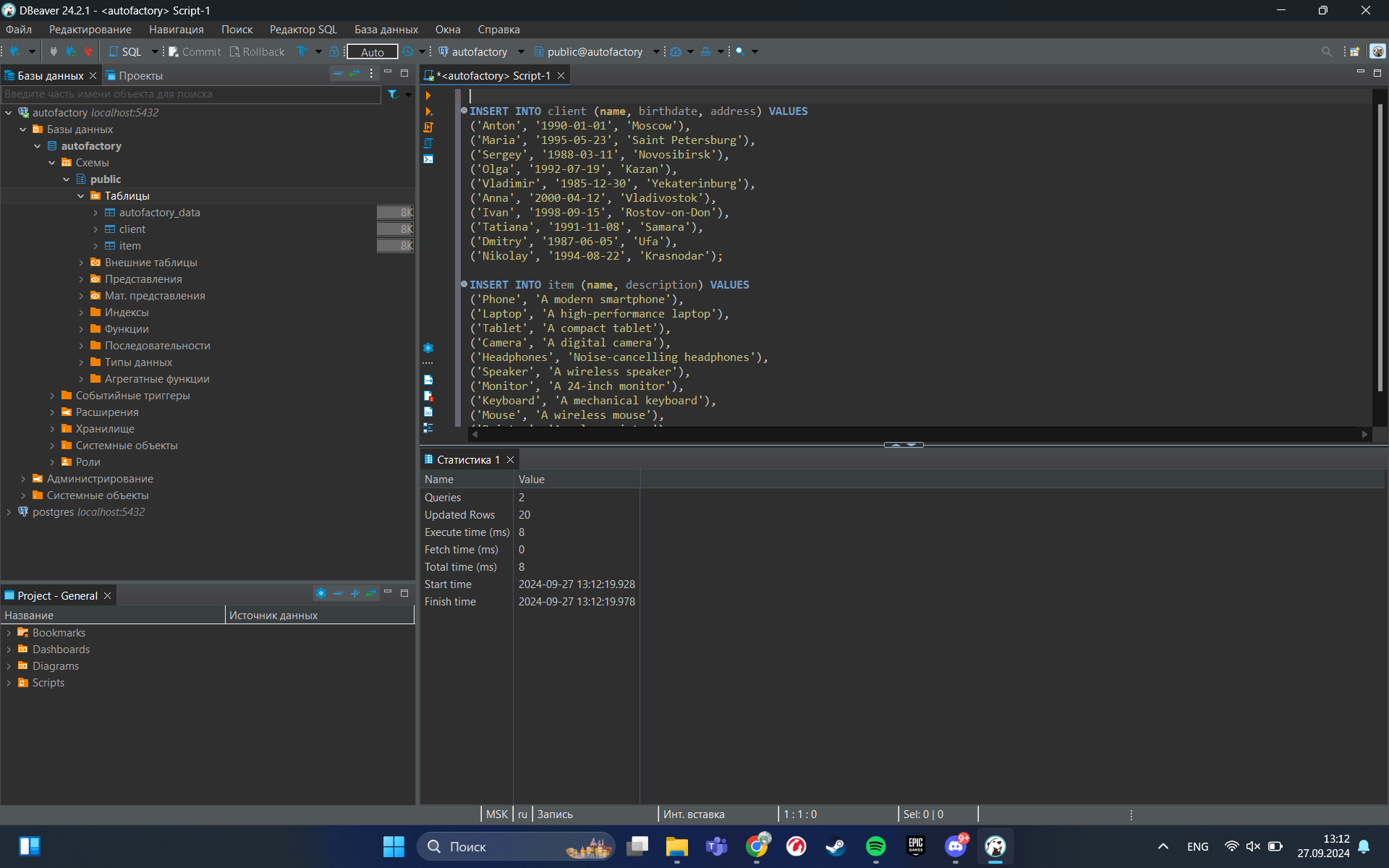
**1. Создание таблиц:**

CREATE TABLE client (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(20),  
 birthdate DATE,  
 address VARCHAR(100)  
);  
  
CREATE TABLE item (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(20),  
 description VARCHAR(200)  
);



**2. Заполнение таблиц:**

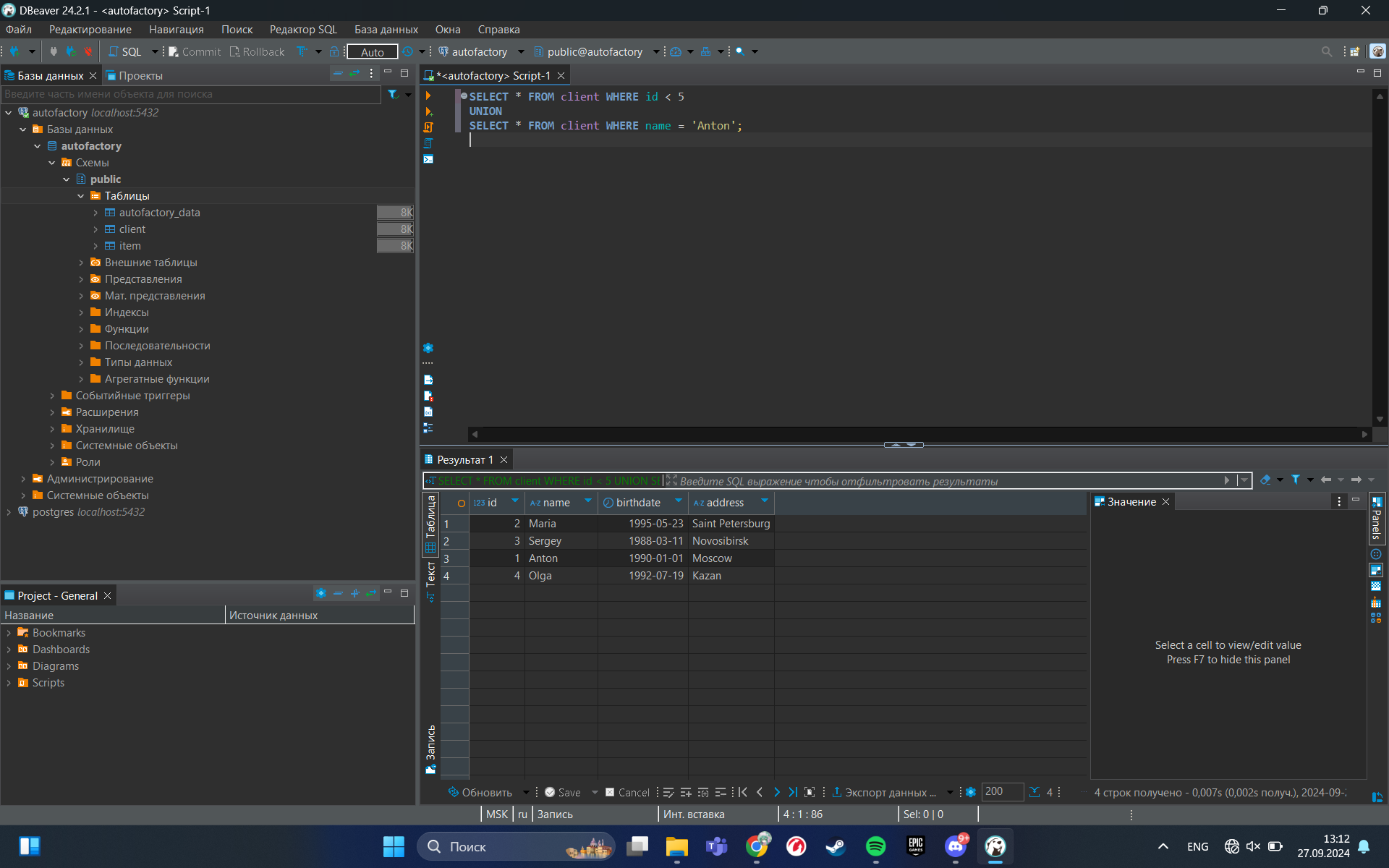
INSERT INTO client (name, birthdate, address) VALUES  
('Anton', '1990-01-01', 'Moscow'),  
('Maria', '1995-05-23', 'Saint Petersburg'),  
('Sergey', '1988-03-11', 'Novosibirsk'),  
('Olga', '1992-07-19', 'Kazan'),  
('Vladimir', '1985-12-30', 'Yekaterinburg'),  
('Anna', '2000-04-12', 'Vladivostok'),  
('Ivan', '1998-09-15', 'Rostov-on-Don'),  
('Tatiana', '1991-11-08', 'Samara'),  
('Dmitry', '1987-06-05', 'Ufa'),  
('Nikolay', '1994-08-22', 'Krasnodar');  
  
INSERT INTO item (name, description) VALUES  
('Phone', 'A modern smartphone'),  
('Laptop', 'A high-performance laptop'),  
('Tablet', 'A compact tablet'),  
('Camera', 'A digital camera'),  
('Headphones', 'Noise-cancelling headphones'),  
('Speaker', 'A wireless speaker'),  
('Monitor', 'A 24-inch monitor'),  
('Keyboard', 'A mechanical keyboard'),  
('Mouse', 'A wireless mouse'),  
('Printer', 'A color printer');



**3. Использование операторов UNION и UNION ALL:**

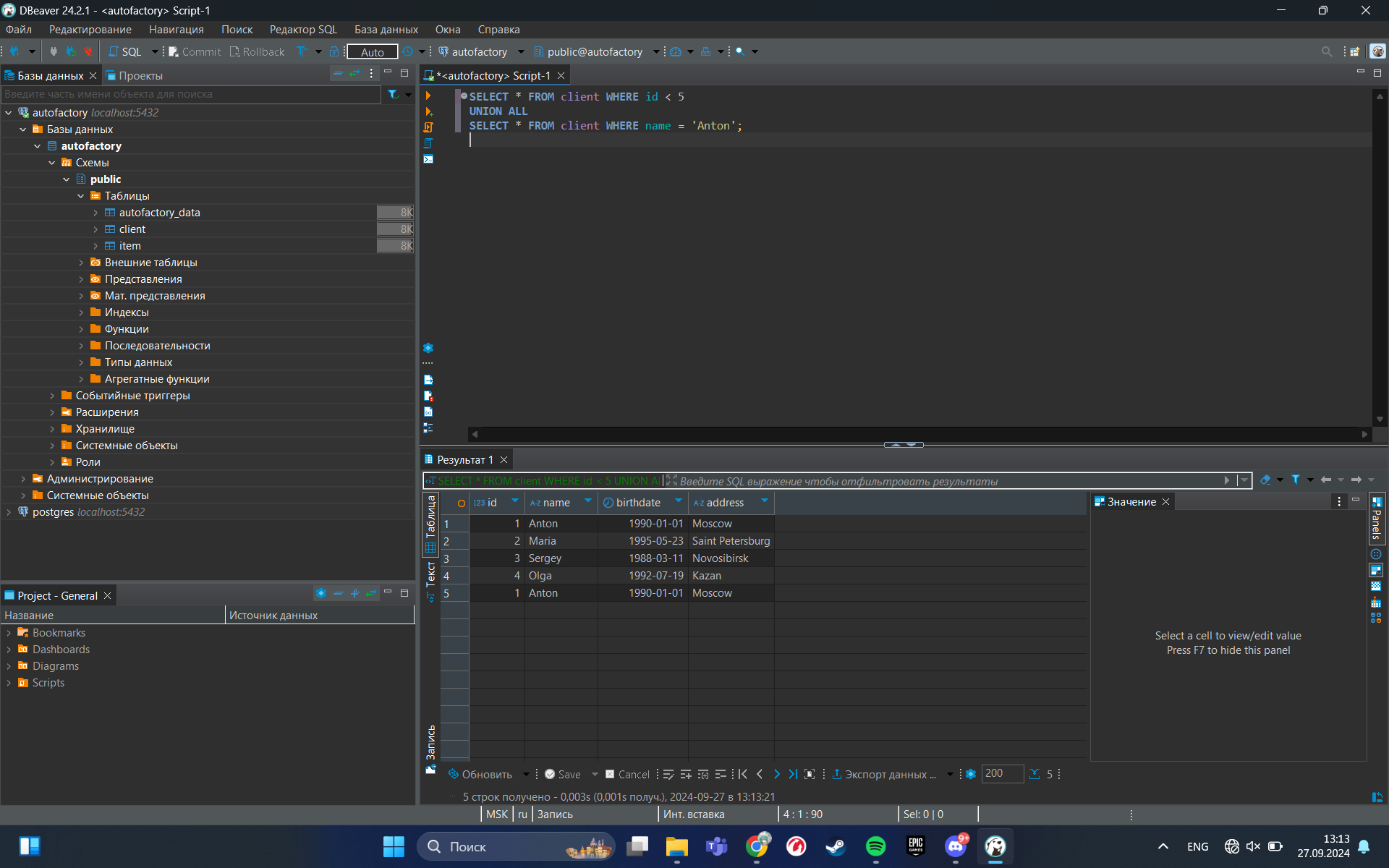
UNION(Не допускать дубликатов):

SELECT \* FROM client WHERE id < 5  
UNION  
SELECT \* FROM client WHERE name = 'Anton';



UNION ALL(Допустить дубликаты и упорядочить их):

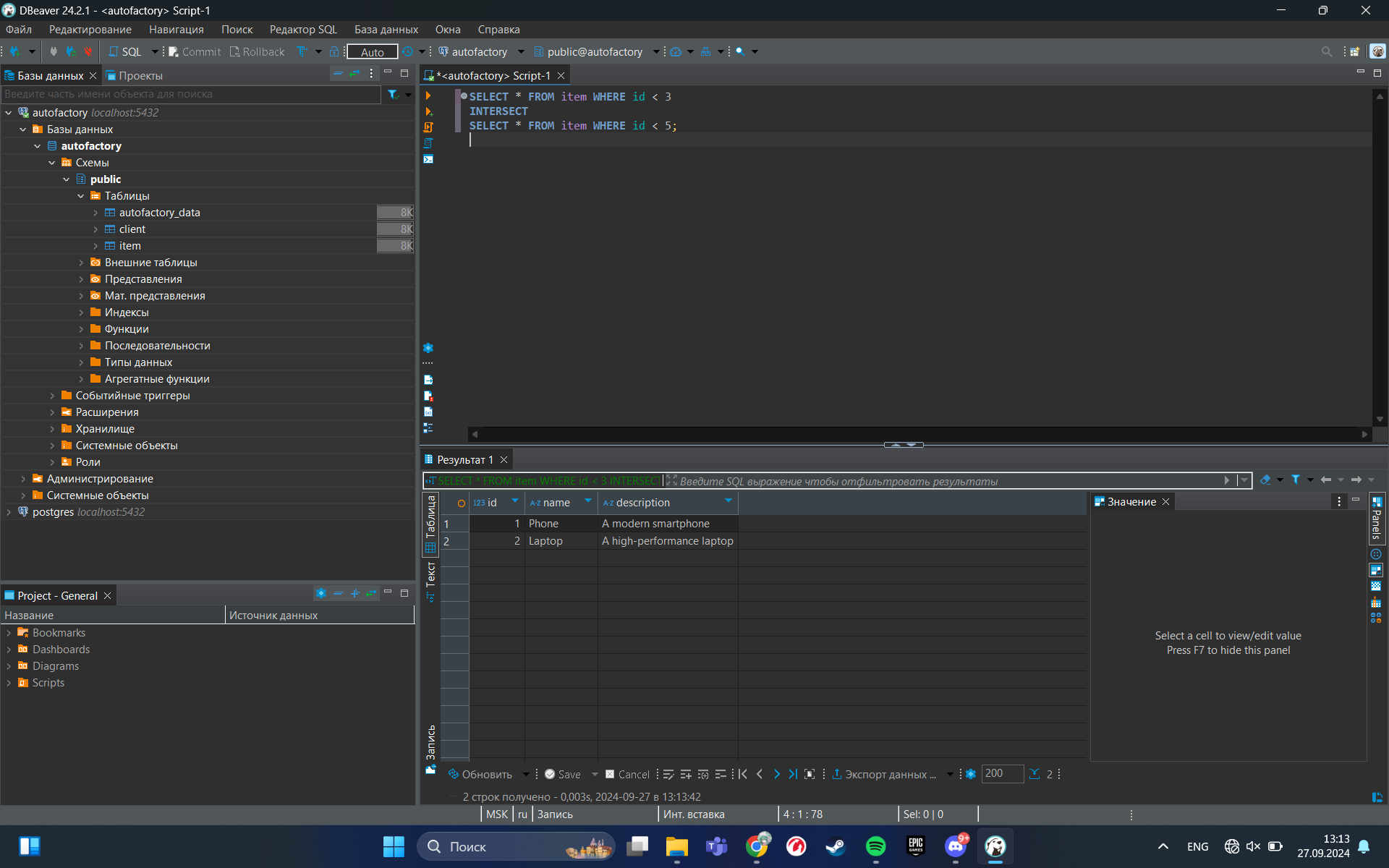
SELECT \* FROM client WHERE id < 5  
UNION ALL  
SELECT \* FROM client WHERE name = 'Anton';



Различия: UNION исключает дубликаты, UNION ALL сохраняет их.

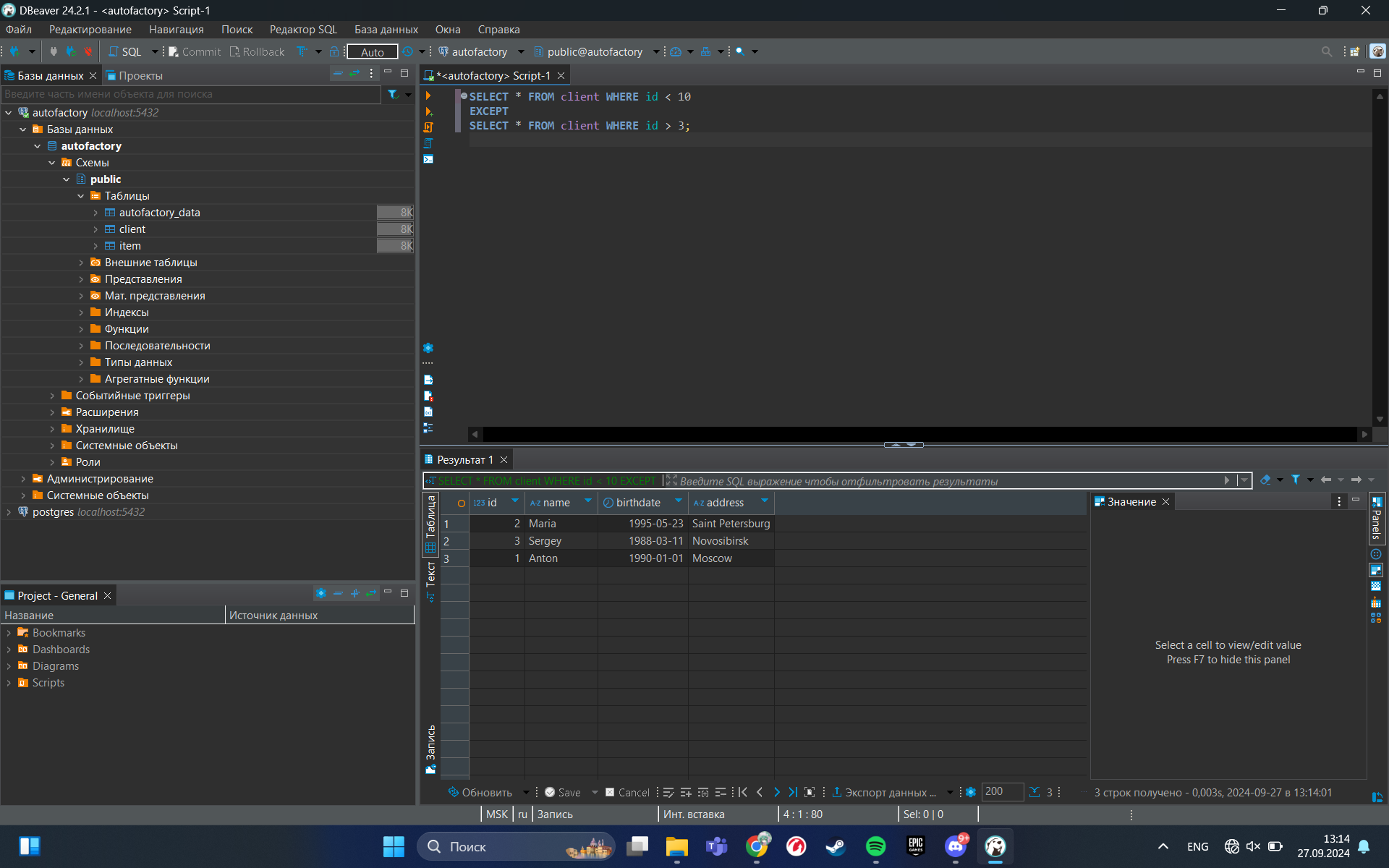
**4. Оператор INTERSECT(Получаем пересечение двух выборок):**

SELECT \* FROM item WHERE id < 3  
INTERSECT  
SELECT \* FROM item WHERE id < 5;



**5. Оператор EXCEPT(Вычитаем выборку из др. выборки):**

SELECT \* FROM client WHERE id < 10  
EXCEPT  
SELECT \* FROM client WHERE id > 3;



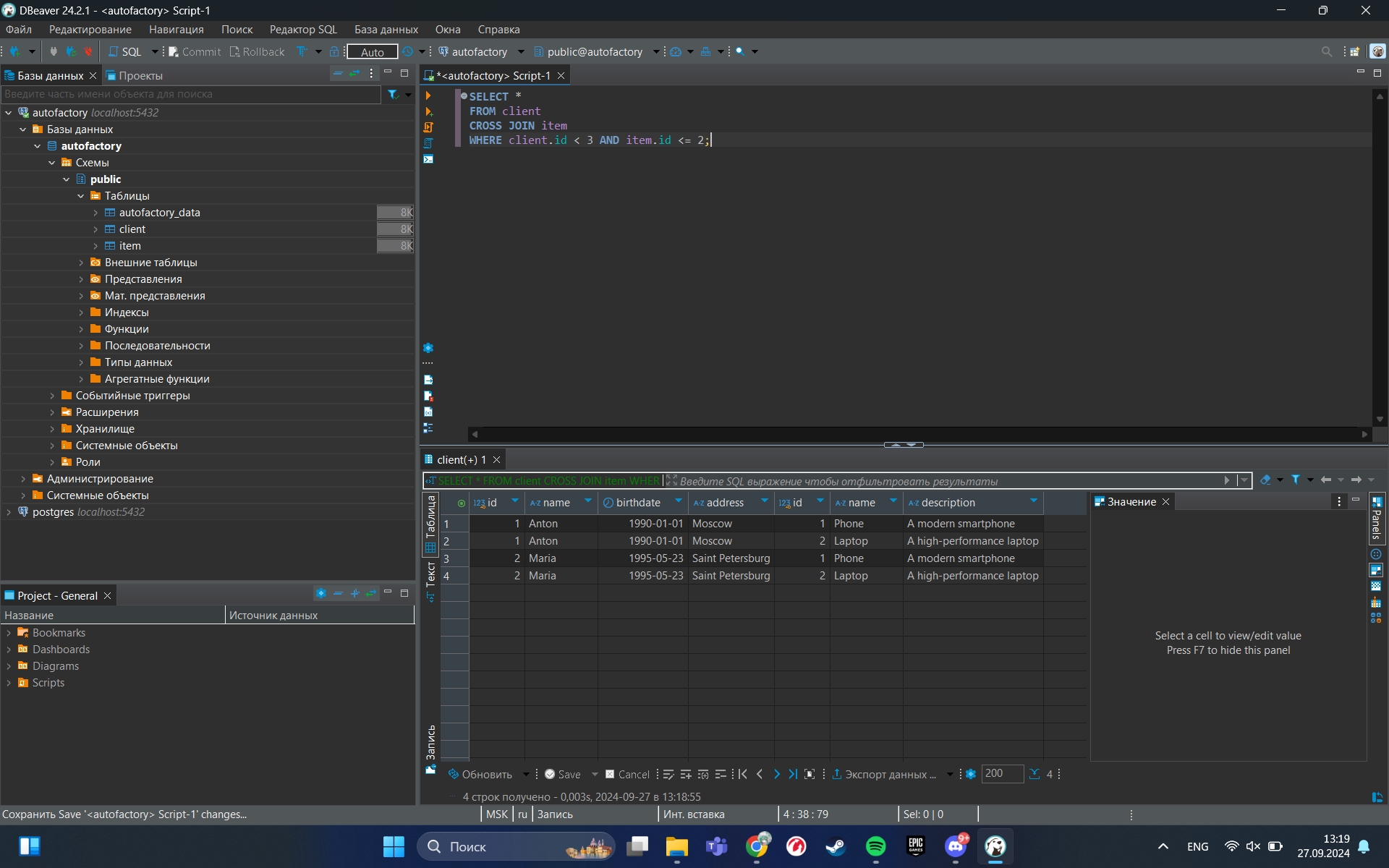
**6. Оператор CROSS JOIN(Находим декартово произведение):**

SELECT \*

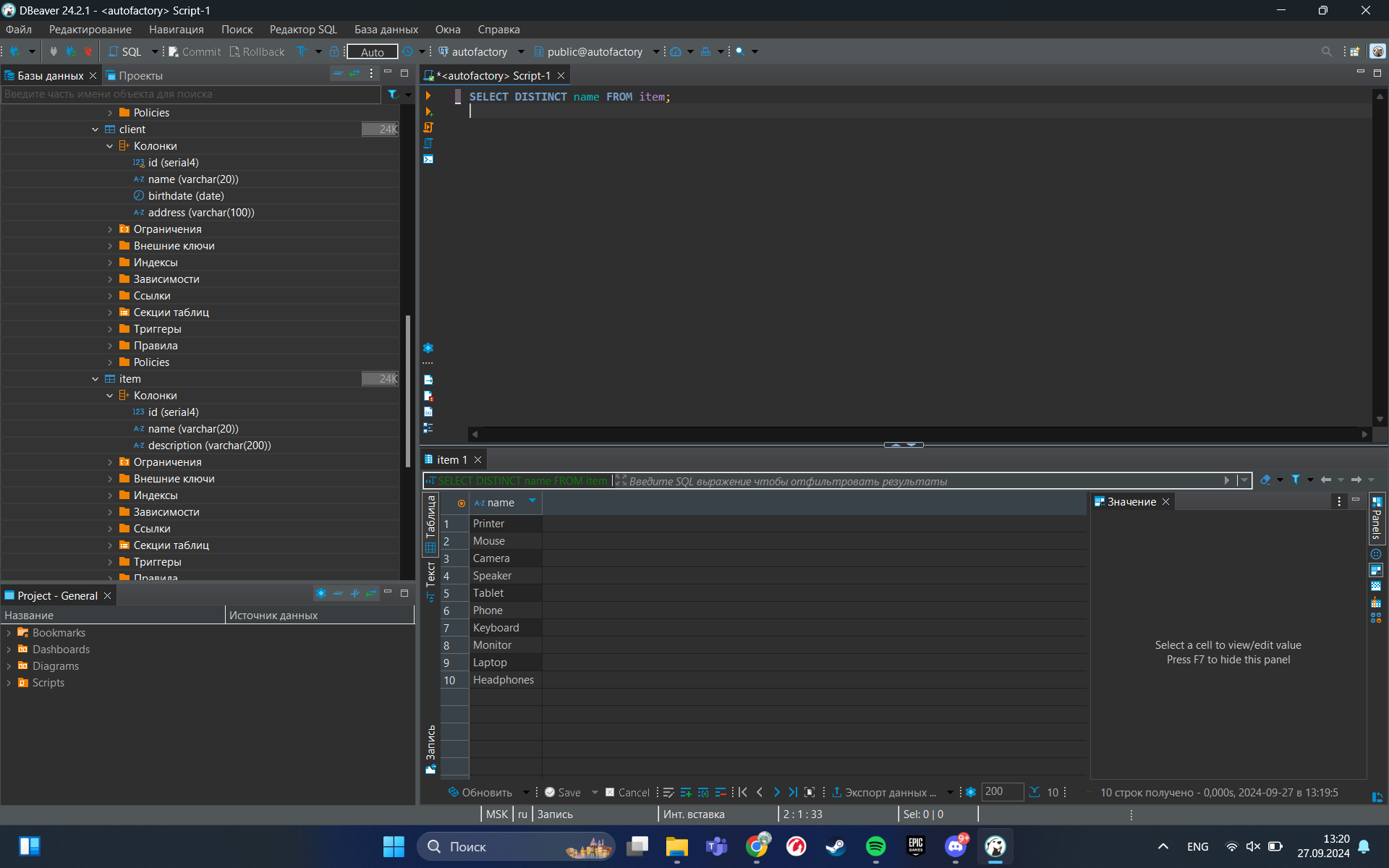
FROM client

CROSS JOIN item

WHERE client.id < 3 AND item.id <= 2;



**7. Оператор DISTINCT(Проекция, Уникальные значения атрибута из таблици):**

SELECT DISTINCT name FROM item;

**9. Соединение одной записи клиента с несколькими товарами:**

ALTER TABLE item ADD COLUMN id\_client integer;

UPDATE item

SET id\_client = 1

WHERE id = 1;

UPDATE item

SET id\_client = 2

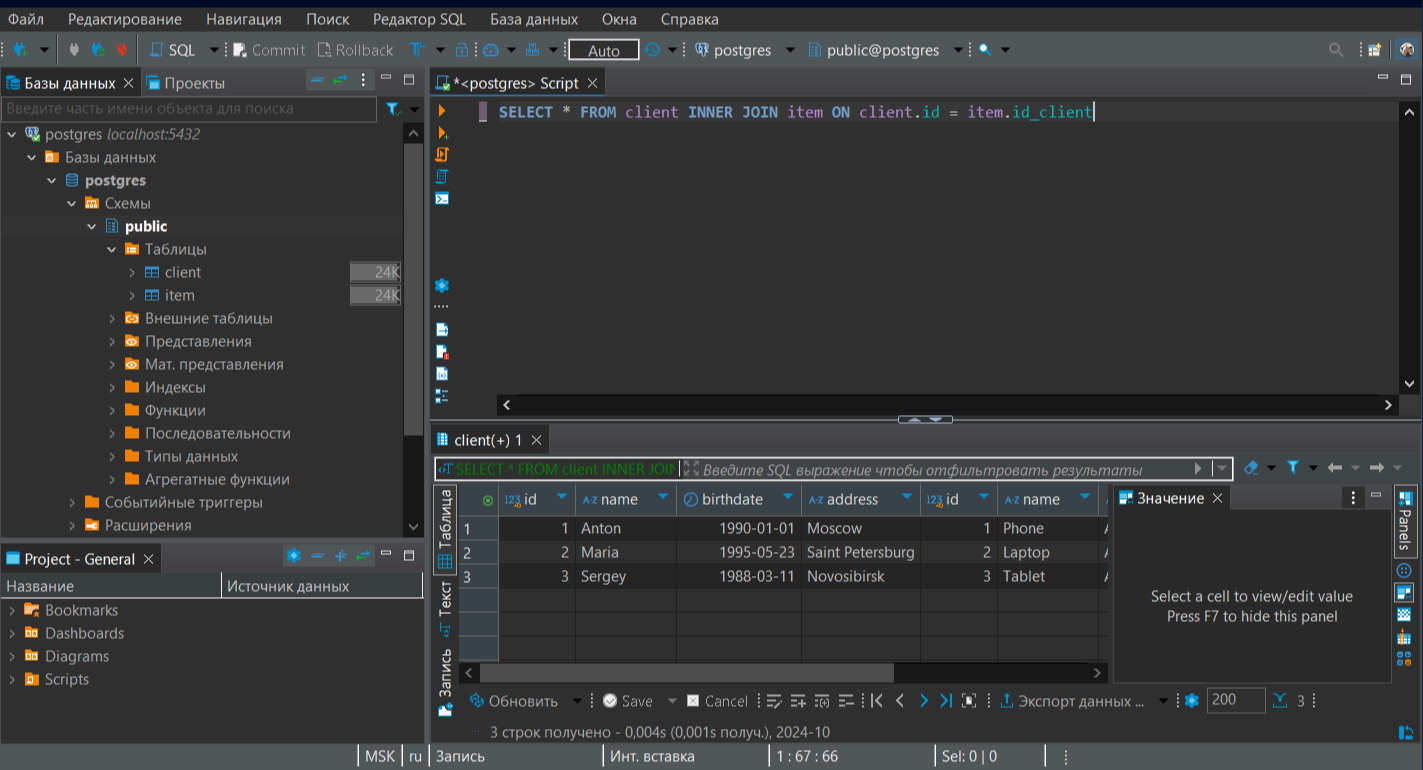
WHERE id = 2;

UPDATE item

SET id\_client = 3

WHERE id = 3;

SELECT \* FROM client INNER JOIN item ON client.id = item.id\_client



**10. Соединение нескольких клиентов с несколькими товарами:**

SELECT c.\*, i.\*

FROM client AS c

CROSS JOIN (

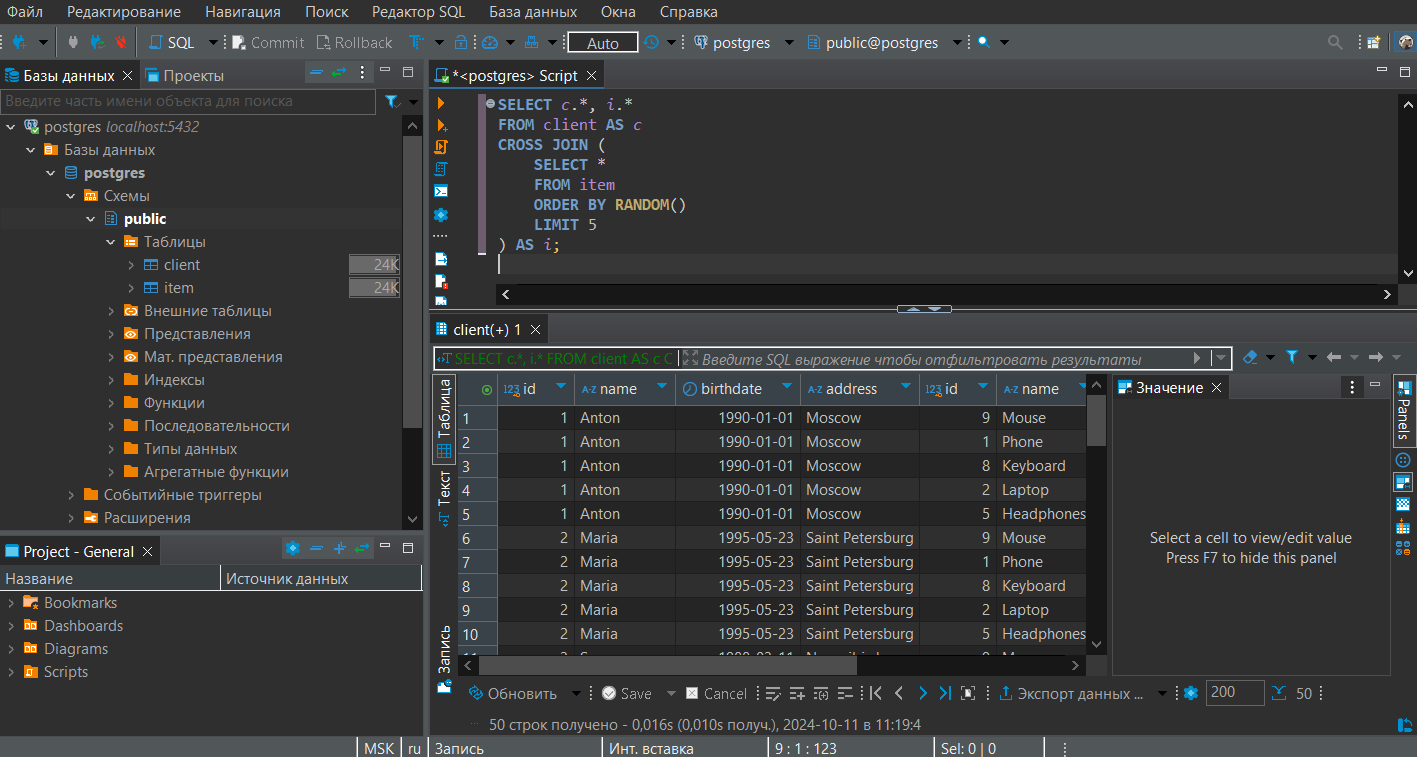
SELECT \*

FROM item

ORDER BY RANDOM()

LIMIT 5

) AS i;



# Вывод

В ходе выполнения работы рассмотрены операторы для объединения и манипуляции данными в PostgreSQL. Выполнены запросы с использованием UNION, INTERSECT, EXCEPT, CROSS JOIN и другие.