МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«Санкт–Петербургский государственный университет  
аэрокосмического приборостроения»

КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент, канд. техн. наук |  |  |  | Е.Л. Турнецкая |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| УПРАВЛЕНИЕ ДАННЫМИ  В ГРАФИЧЕСКОЙ БАЗЕ ДАННЫХ NEO4J |
|  |
| по курсу: Методология и технология проектирования информационных систем |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | Z4410МК |  |  |  | А. А. Шулепов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

Цель работы: получение практических навыков работы с графической базой данных Neo4j.

1. Установить Docker.

2. Создать контейнер для развертывания Neo4j.

3. Выполнить упражнения по работе с БД Sales.

4. По своему варианту, который равен номеру по списку группы спроектировать и реализовать графовую БД (варианты вынесены в отдельный файл).

5. Заполнить ее 7-10 узлами для каждой метки (7-10 записей в каждой таблице, если мы переходим на понятия реляционной БД).

6. Продумать 10 запросов разной степени сложности и зафиксировать их в письменной форме.

7. Реализовать запросы на Chyper в клиентской части БД (примеры показаны в упражнениях).

8. Выполнить отчет. В отчете должен быть вывод в формате эссе.

Лабораторная работа выполнена в соответствии с вариантом № 13

Вариант 13. ГИБДД

Государственная инспекция безопасности дорожного движения Хоббитонии занимается регистрацией и учетом автомобилей граждан страны, регистрацией ДТП. В карточке каждого автомобиля указана его мощность, максимальная скорость, год выпуска, стоимость, марка. У каждого ДТП есть дата, время, список автомобилей и адрес. У каждого гражданина Хоббитонии может быть несколько автомобилей. Автомобили он может продавать другим гражданам страны. У каждого автомобиля есть хозяин. В ДТП, к сожалению, может пострадать несколько автомобилей хоббитов.

Для начала был создан контейнер, как показано на рисунке 1.

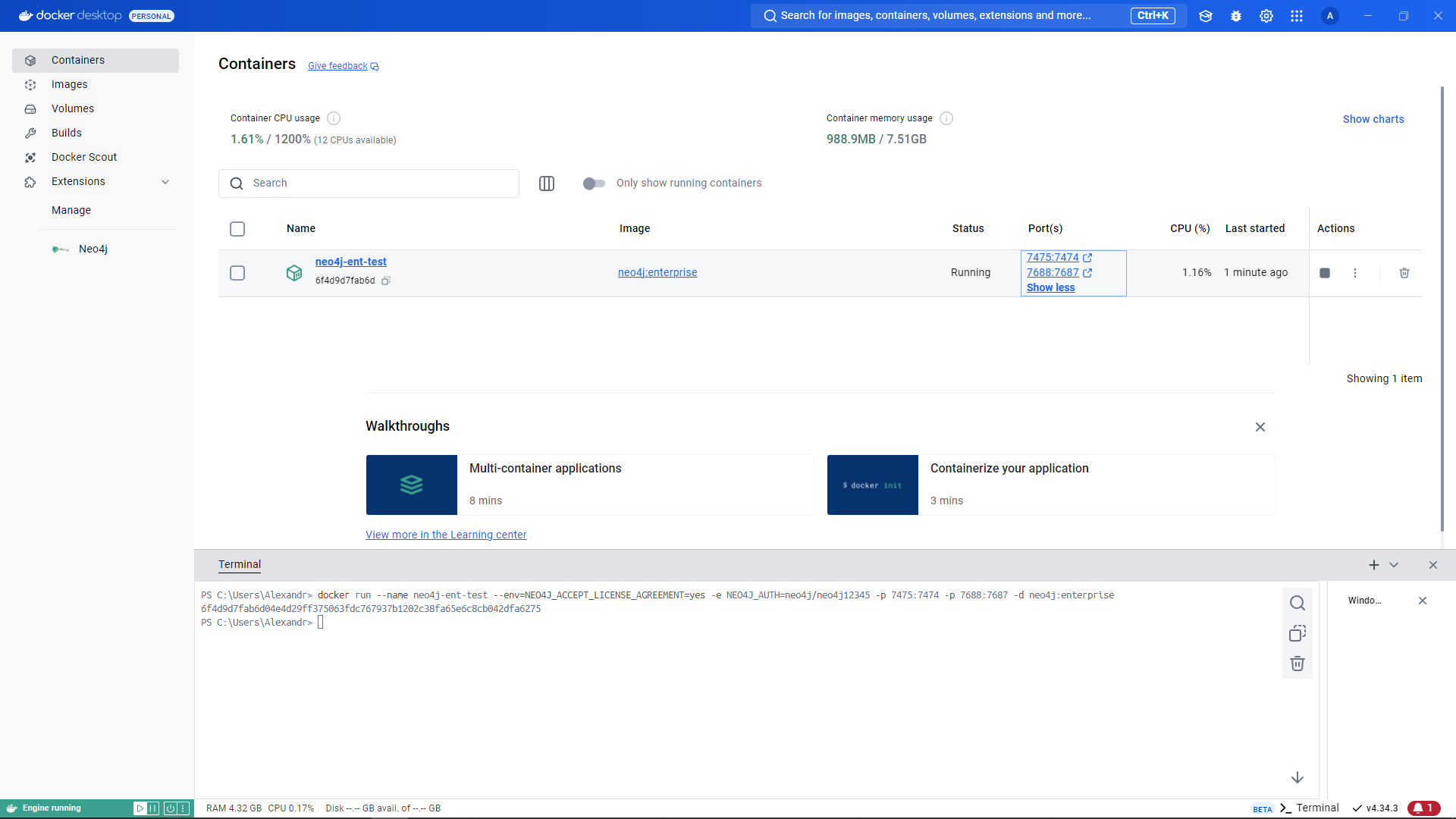


Рисунок 1 - Создание контейнера

Для ознакомления с возможности Neo4j была создана база данных sales, как показано на рисунках 2 - 16

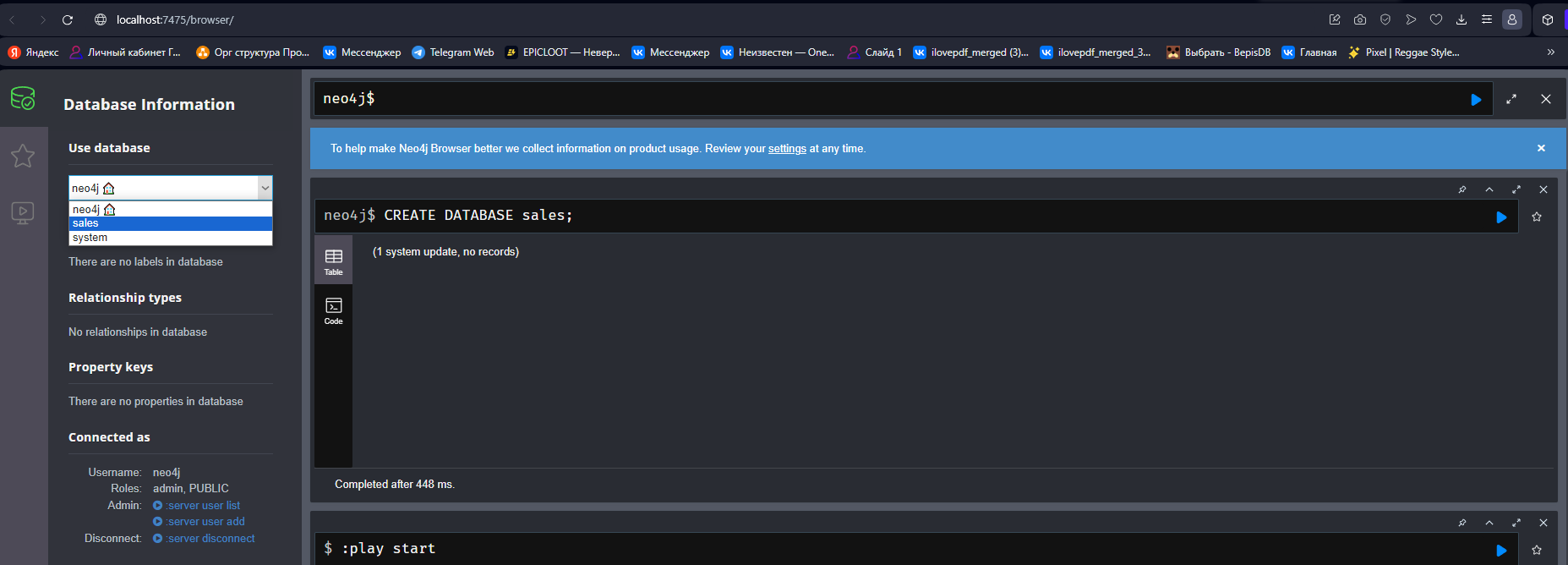


Рисунок 2 - Создание БД sales

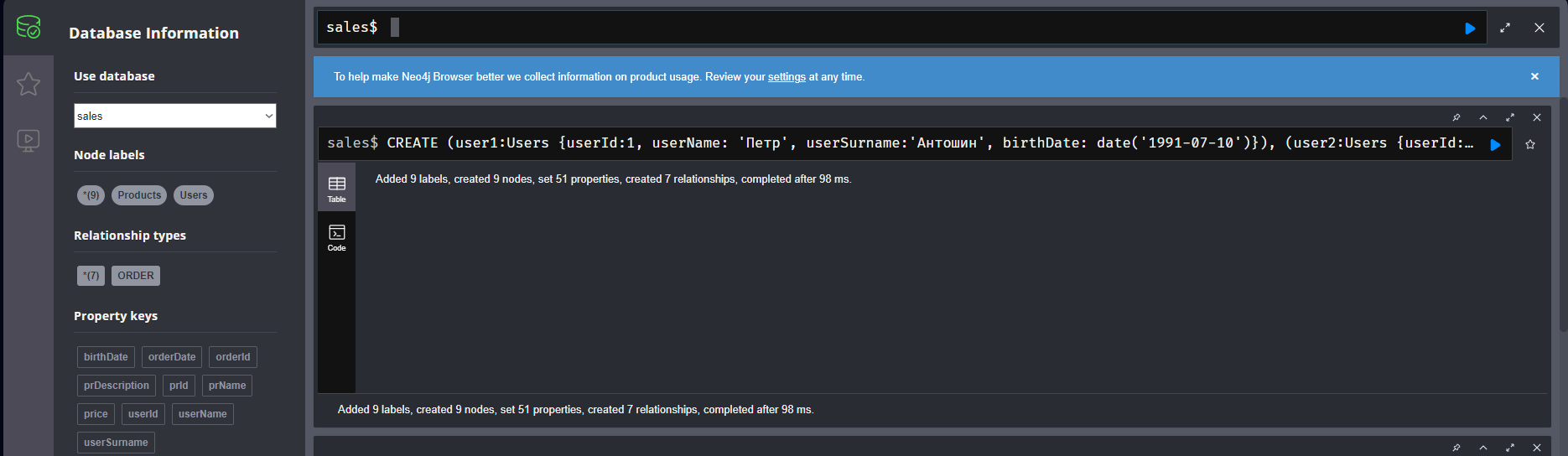


Рисунок 3 – Заполнение БД

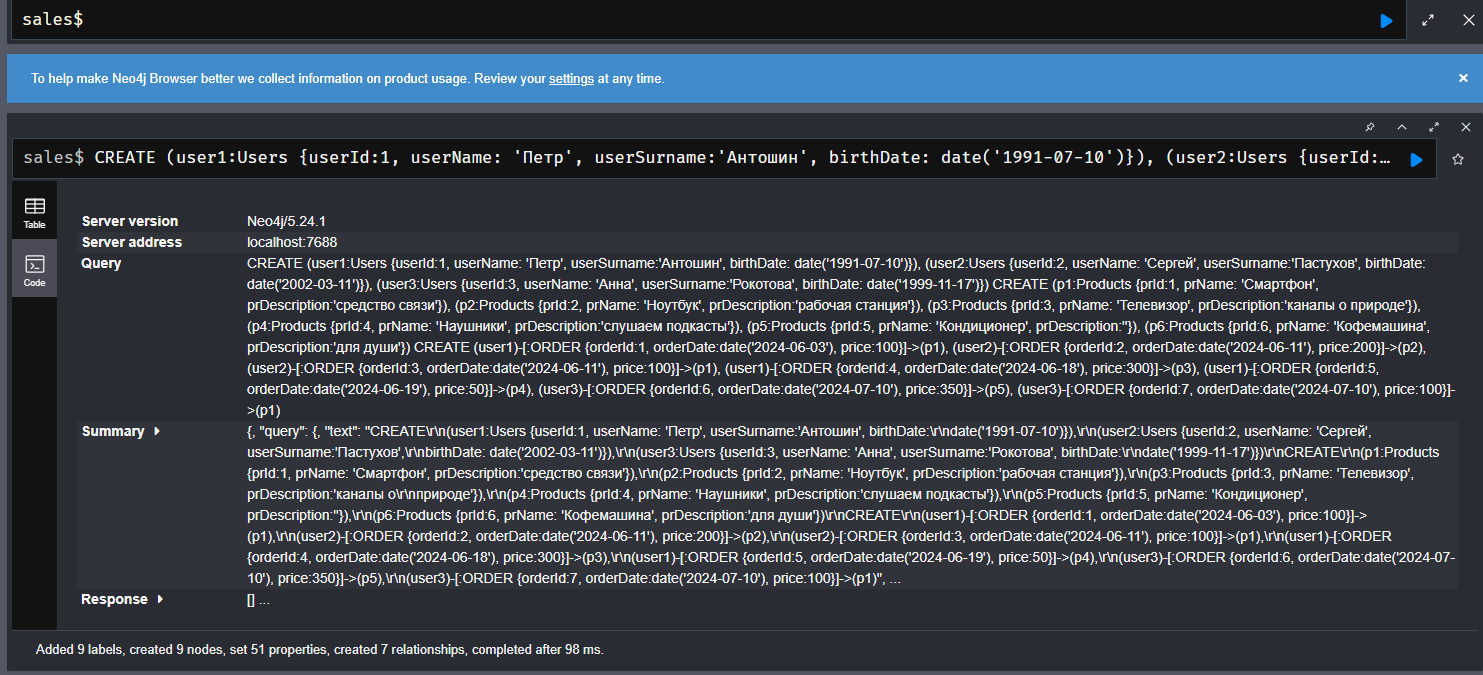


Рисунок 4 - Программный код

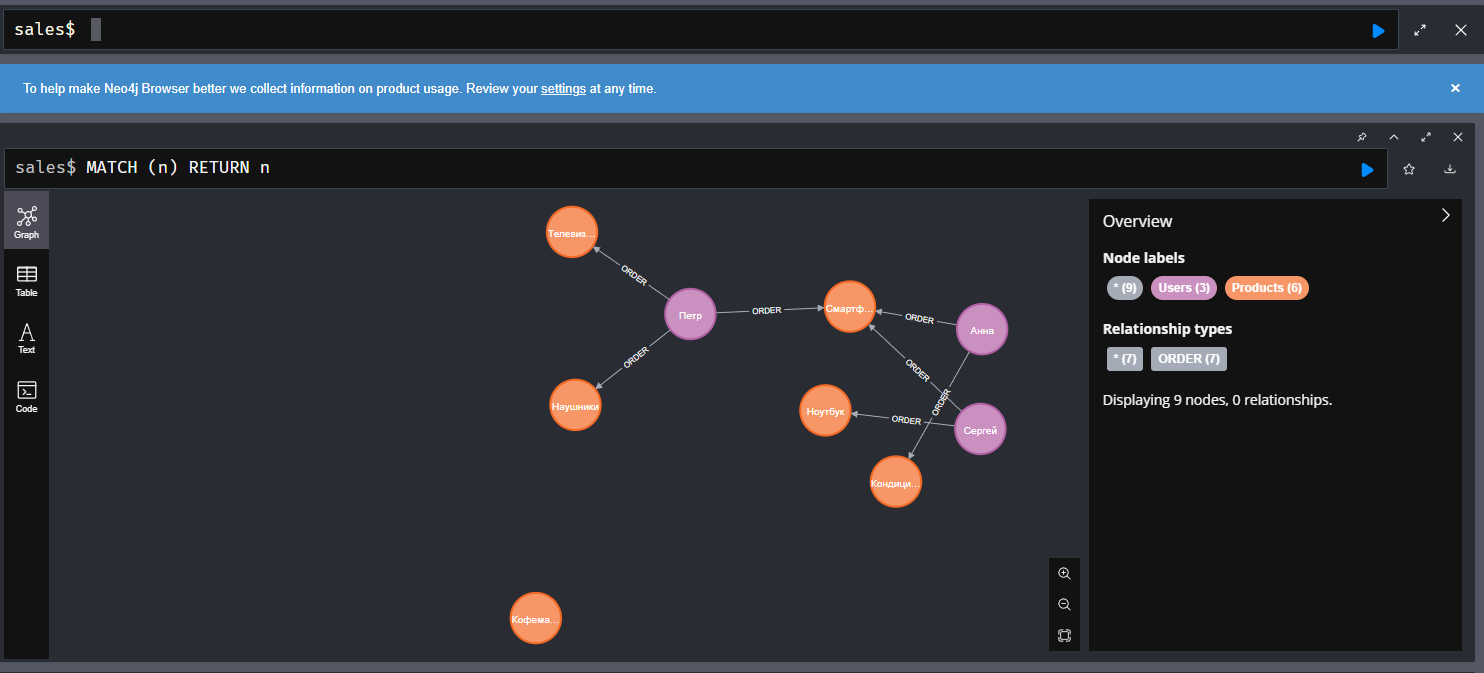


Рисунок 5 - Все данные в графическом виде

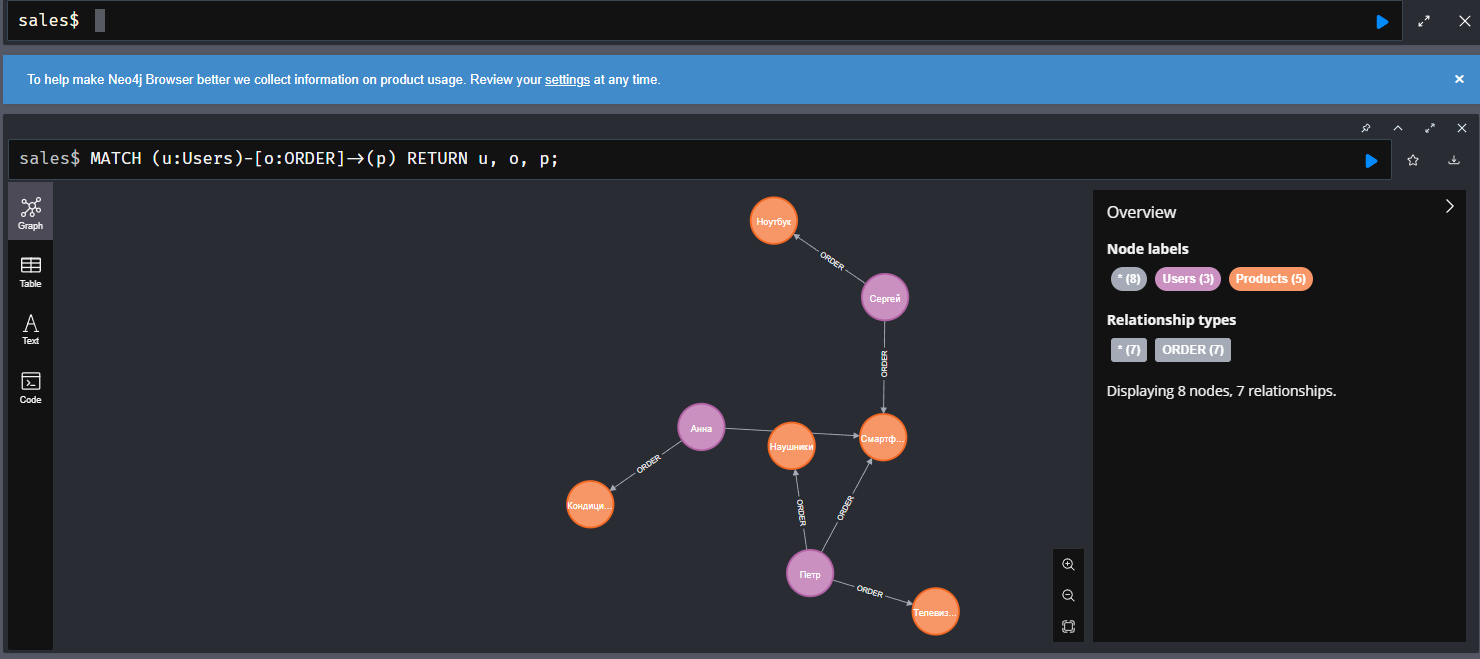


Рисунок 6 - Клиенты и товары только с заказами

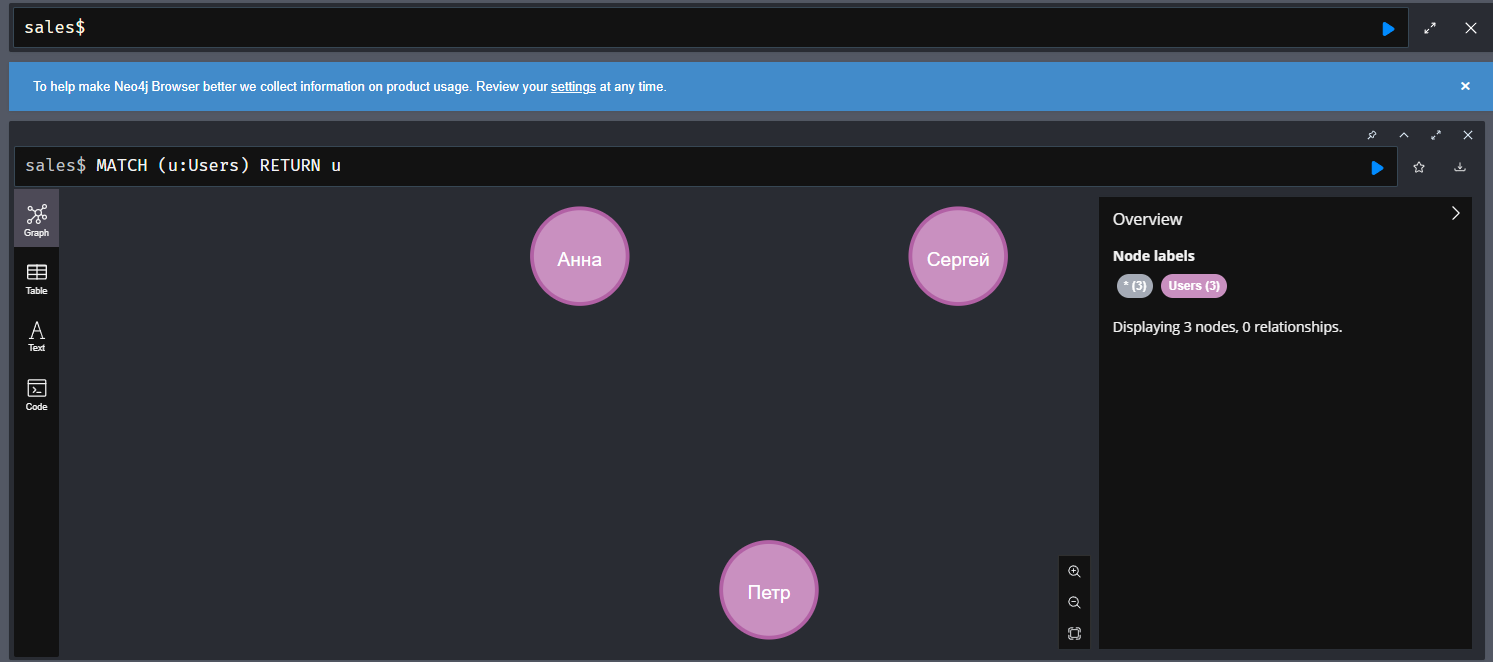


Рисунок 7 - Все пользователи системы

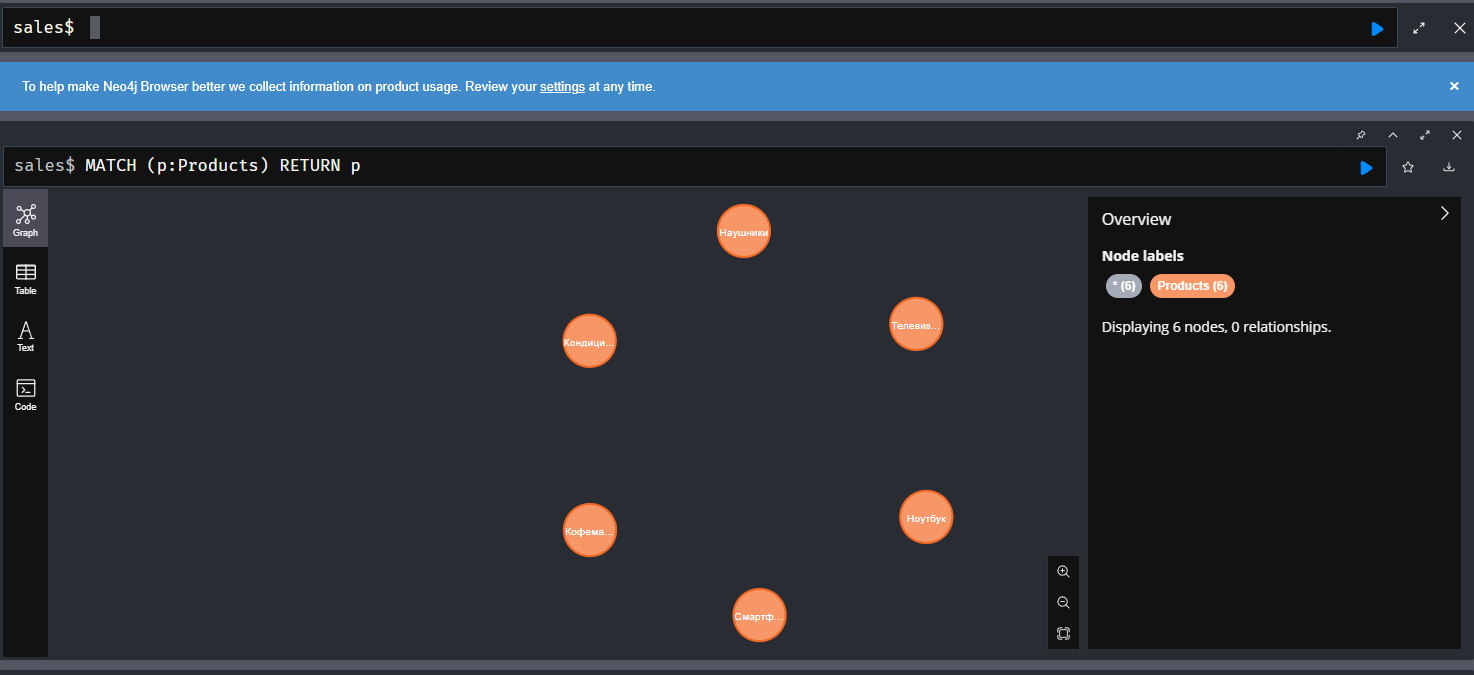


Рисунок 8- Все продукты

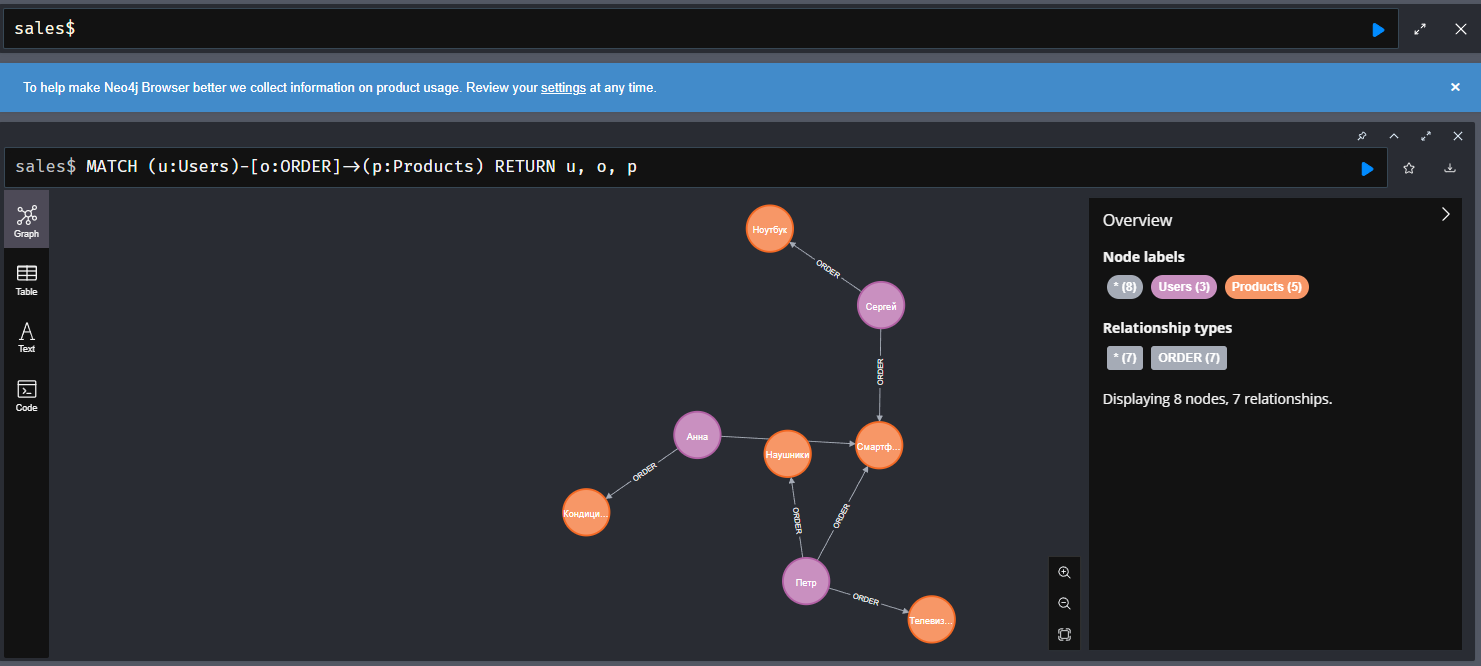


Рисунок 9 - Все заказы клиентов

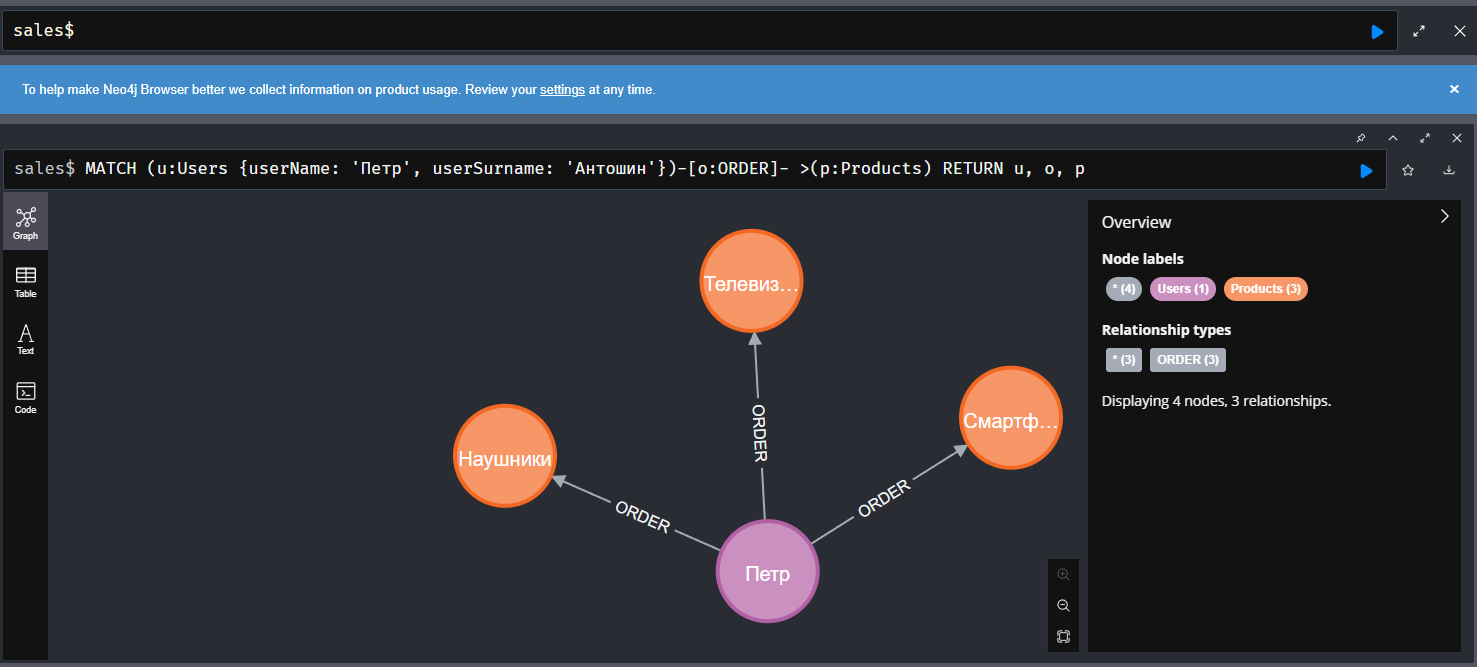


Рисунок 10 - Все заказы сделанные Петром

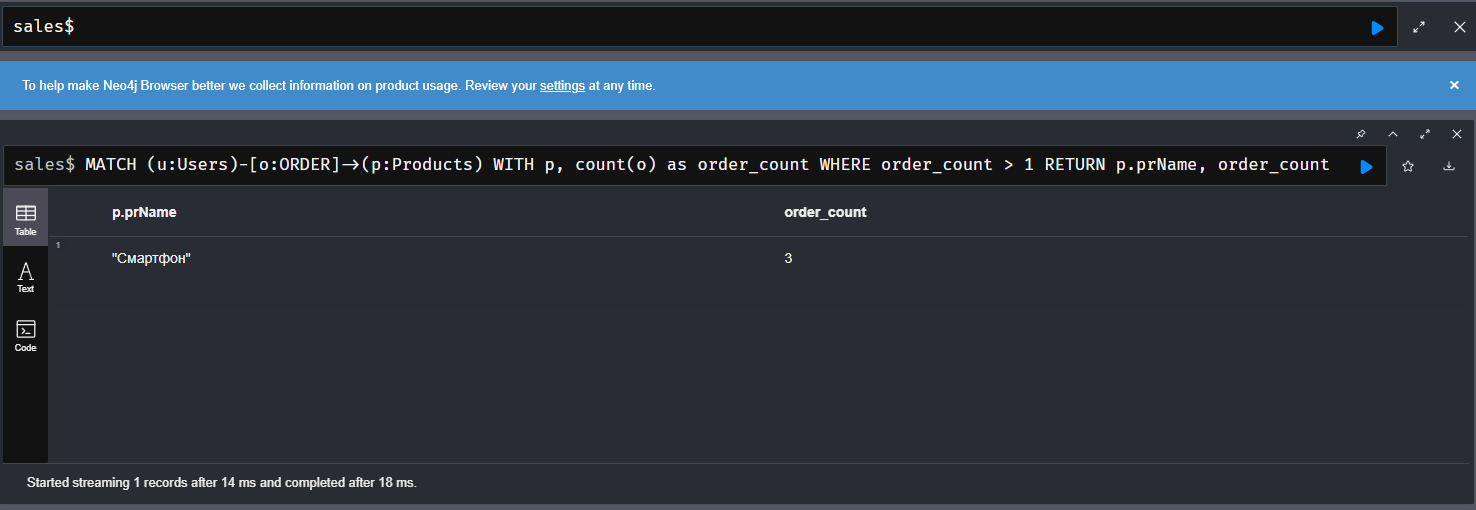


Рисунок 11 - Продукты, которые были куплены более одного раза

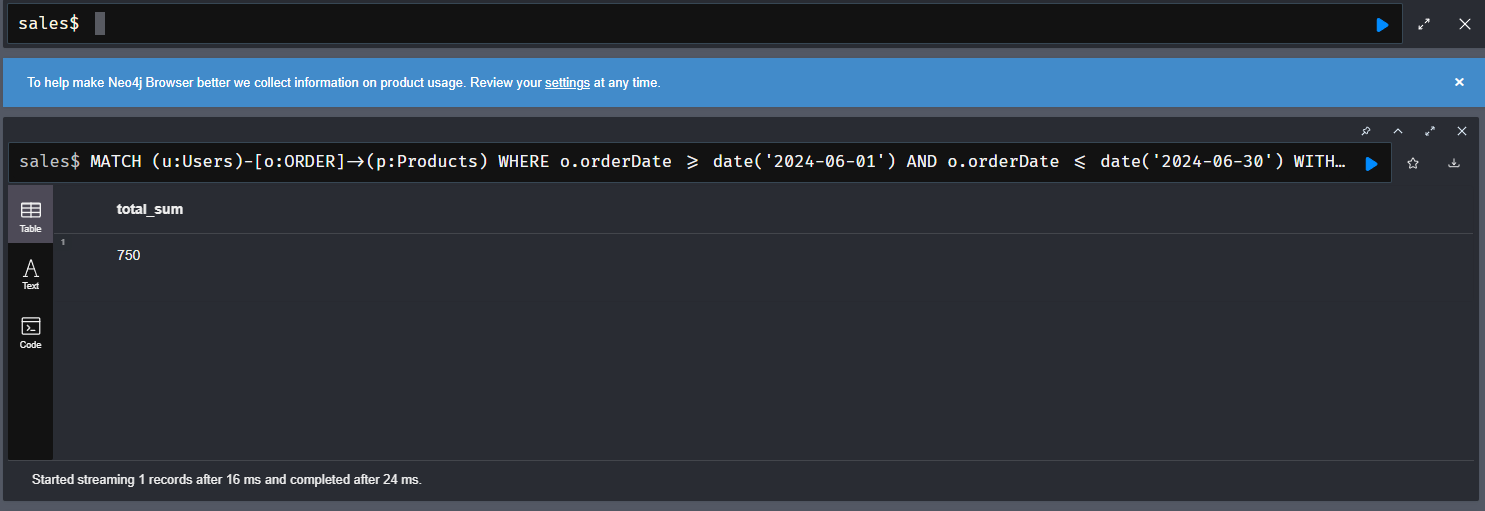


Рисунок 12 - Сумма всех заказов за июнь

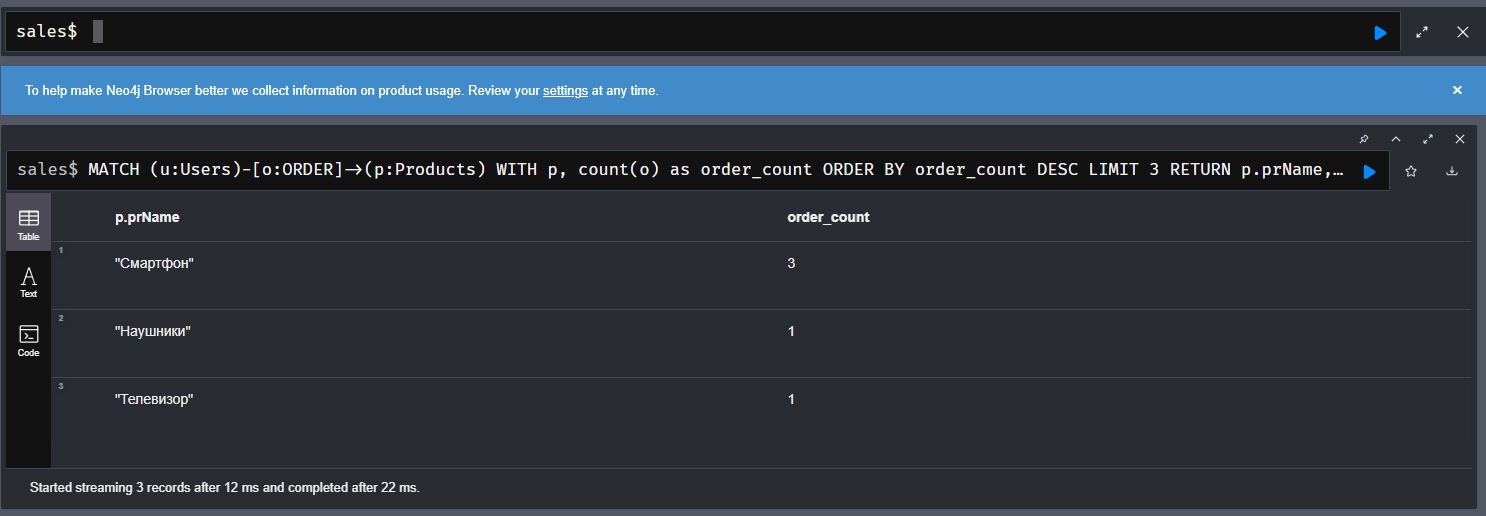


Рисунок 13 - Три самых популярных продукта

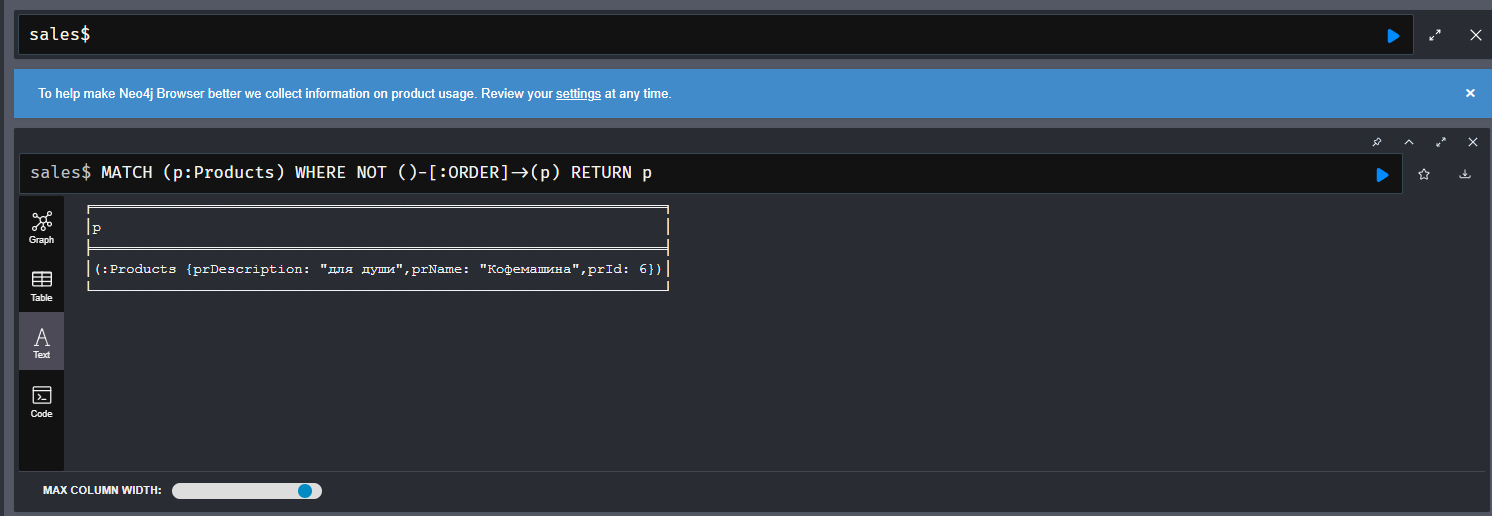


Рисунок 14 - Продукты, которые не были ни в одном заказе

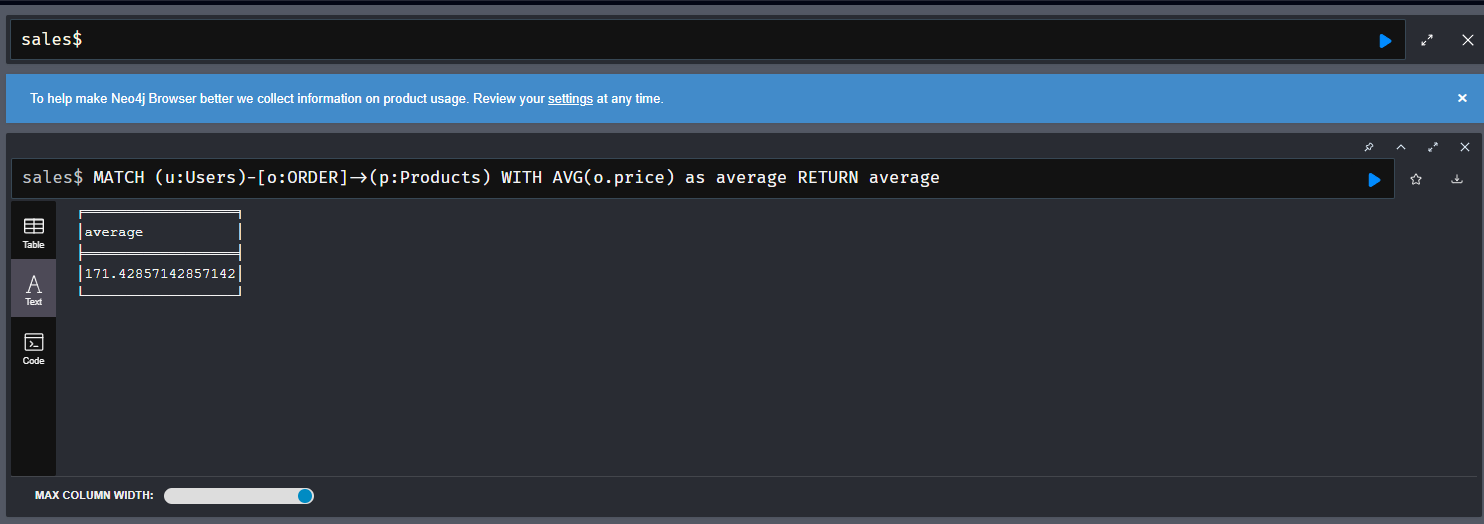


Рисунок 15 - Средняя цена всех товаров

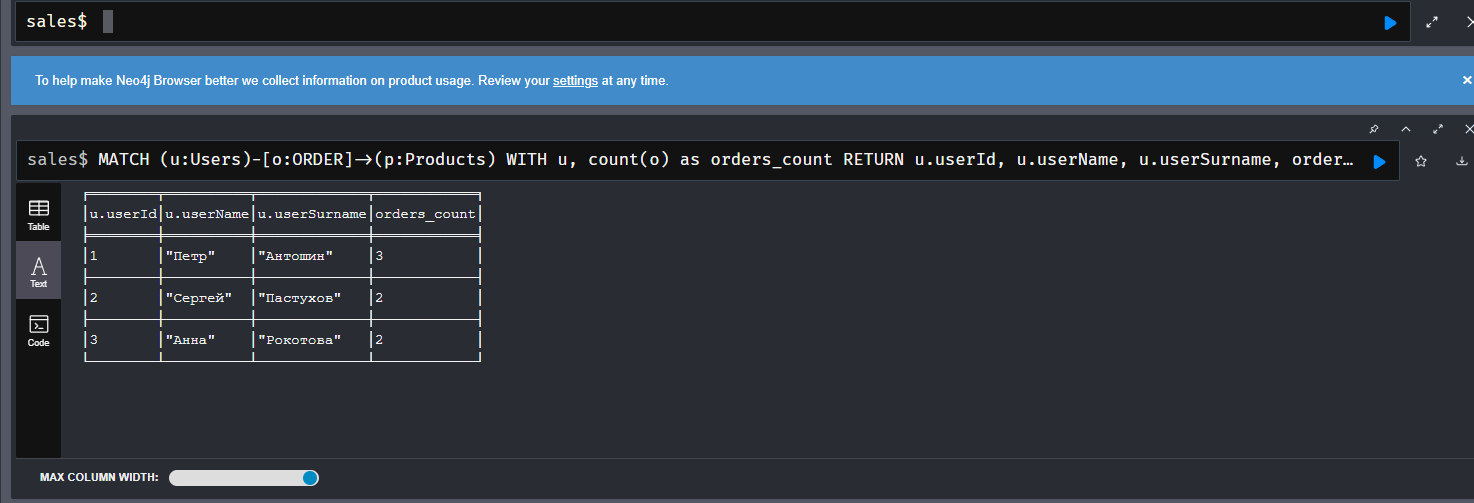


Рисунок 16 - Количество заказов, сделанных пользователями

Была создана база данных ГИБДД по варианту №13, как показано на рисунке 16 и представлено в таблице 1.

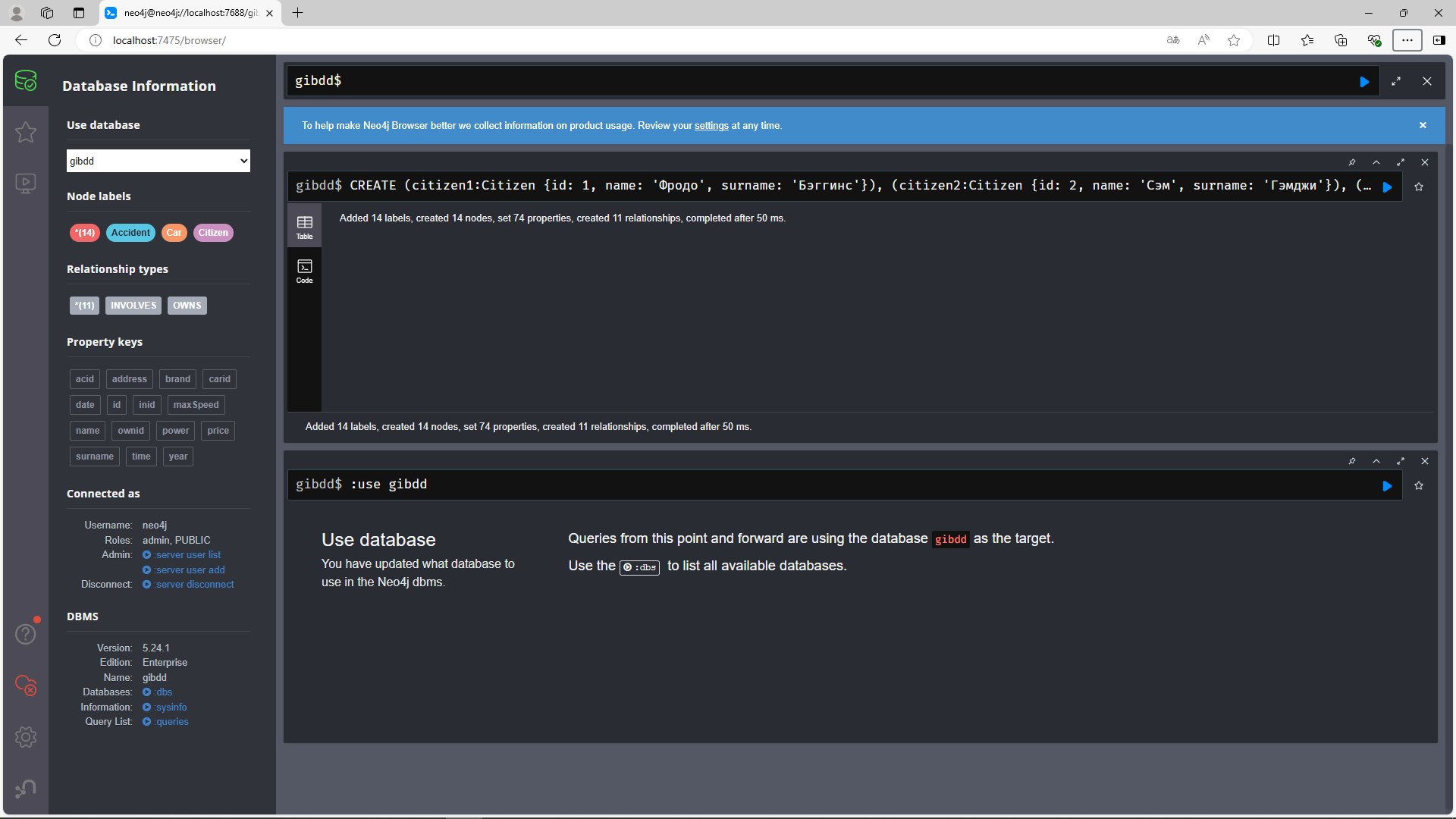


Рисунок 17 - Создание базы данных ГИБДД

Таблица 1 - Создание БД по теме ГИБДД (13 вариант)

|  |
| --- |
| CREATE  (citizen1:Citizen {id: 1, name: 'Фродо', surname: 'Бэггинс'}),  (citizen2:Citizen {id: 2, name: 'Сэм', surname: 'Гэмджи'}),  (citizen3:Citizen {id: 3, name: 'Гэндальф', surname: 'Серый'}),  (citizen4:Citizen {id: 4, name: 'Арагорн', surname: 'Статт'}),  (citizen5:Citizen {id: 5, name: 'Леголас', surname: 'Теленг'})  CREATE  (car1:Car {carid: 1, brand: 'Мерседес', power: 150, maxSpeed: 200, year: 2020, price: 30000}),  (car2:Car {carid: 2, brand: 'Тойота', power: 120, maxSpeed: 180, year: 2018, price: 20000}),  (car3:Car {carid: 3, brand: 'Форд', power: 130, maxSpeed: 190, year: 2019, price: 25000}),  (car4:Car {carid: 4, brand: 'Хонда', power: 110, maxSpeed: 170, year: 2017, price: 18000}),  (car5:Car {carid: 5, brand: 'BMW', power: 200, maxSpeed: 240, year: 2021, price: 50000}),  (car6:Car {carid: 6, brand: 'Ауди', power: 160, maxSpeed: 210, year: 2020, price: 45000})  CREATE  (accident1:Accident {acid: 1, date: date('2024-09-01'), time: time('14:30:00'), address: 'Древесная улица'}),  (accident2:Accident {acid: 2, date: date('2024-09-02'), time: time('15:00:00'), address: 'Северная улица'}),  (accident3:Accident {acid: 3, date: date('2024-09-03'), time: time('13:00:00'), address: 'Южная улица'})  CREATE  (citizen1)-[:OWNS {ownid: 1}]->(car1),  (citizen1)-[:OWNS {ownid: 2}]->(car2),  (citizen2)-[:OWNS {ownid: 3}]->(car3),  (citizen2)-[:OWNS {ownid: 4}]->(car4),  (citizen3)-[:OWNS {ownid: 5}]->(car5),  (citizen4)-[:OWNS {ownid: 6}]->(car6)  CREATE  (accident1)-[:INVOLVES {inid: 1}]->(car1),  (accident1)-[:INVOLVES {inid: 2}]->(car2),  (accident2)-[:INVOLVES {inid: 3}]->(car3),  (accident3)-[:INVOLVES {inid: 4}]->(car5),  (accident3)-[:INVOLVES {inid: 5}]->(car6) |

Как видно из таблицы было создана 5 граждан, у каждого из которых есть как минимум 1 автомобиль, также в таблице записано 3 ДТП.

Было создано 10 запросов разной сложности.

Запрос на получение всех узлов представлен на рисунке 18 и в таблице 2.

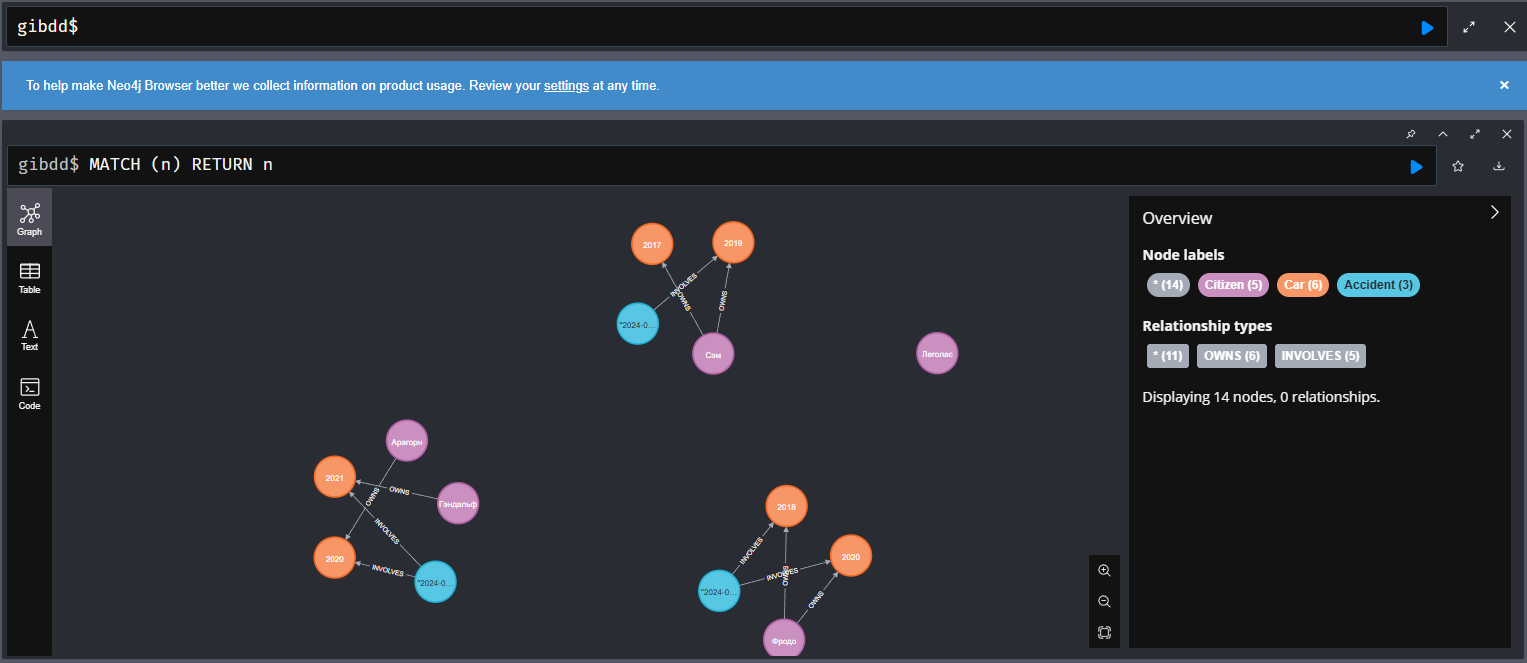


Рисунок 18 - Все узлы БД

Таблица 2 - Код запроса на получение всех узлов БД

|  |
| --- |
| MATCH (n)  RETURN n |

Запрос на получение всех автомобилей представлен на рисунке 19 и в таблице 3.

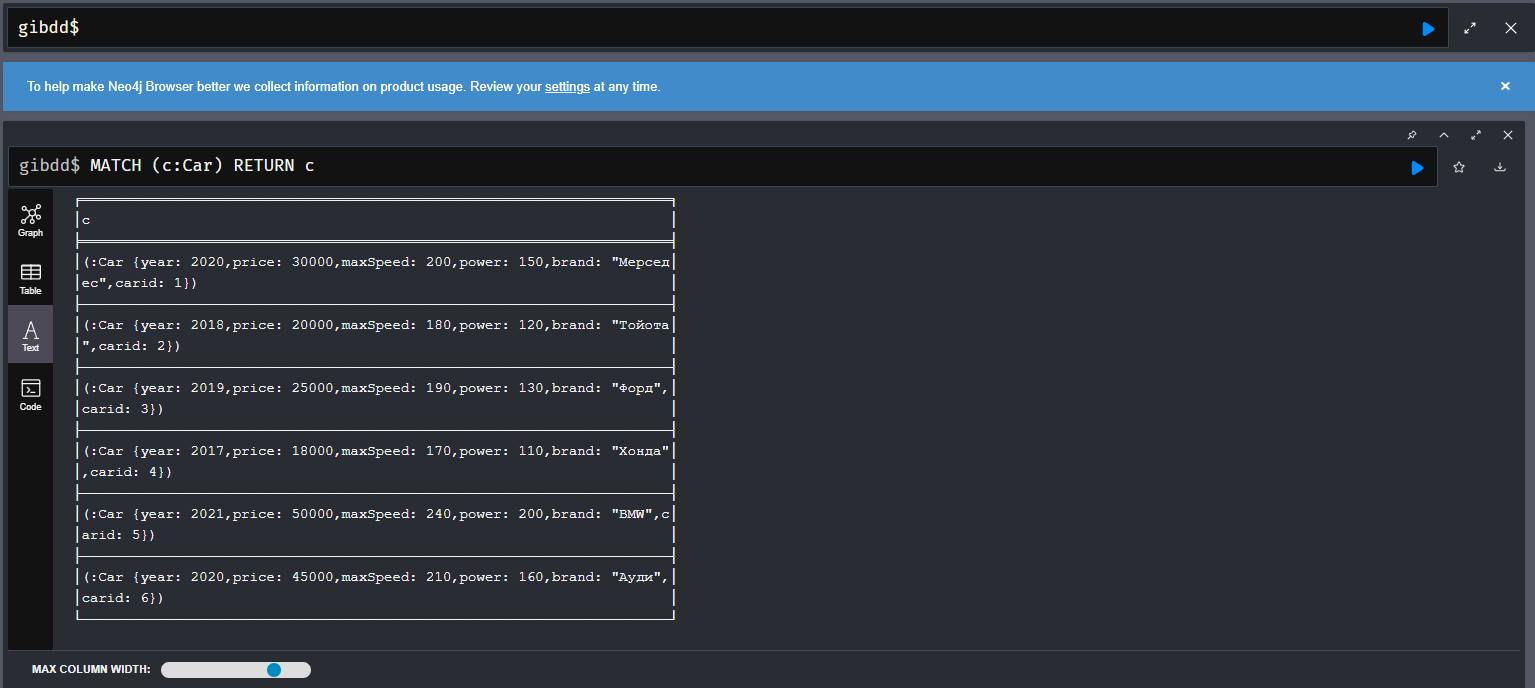


Рисунок 19 - Запрос на получение всех автомобилей

Таблица 3 - Код запрос на получение всех автомобилей

|  |
| --- |
| MATCH (c:Car)  RETURN c |

Запрос на получение всех ДТП представлен на рисунке 20 и в таблице 4

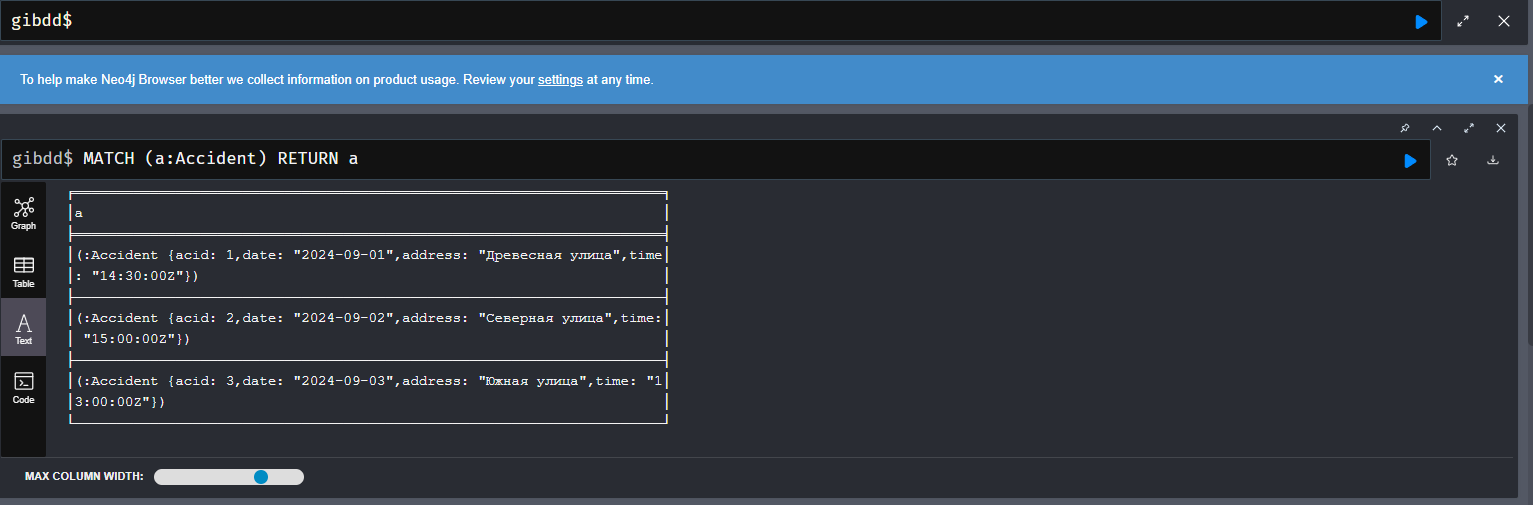


Рисунок 20 - Запрос на получение всех ДТП

Таблица 4 - Код запроса на получение всех ДТП

|  |
| --- |
| MATCH (a:Accident)  RETURN a |

Запрос на получение автомобилей, участвующих в ДТП представлен на рисунках 21 и 22, а также в таблице 5.

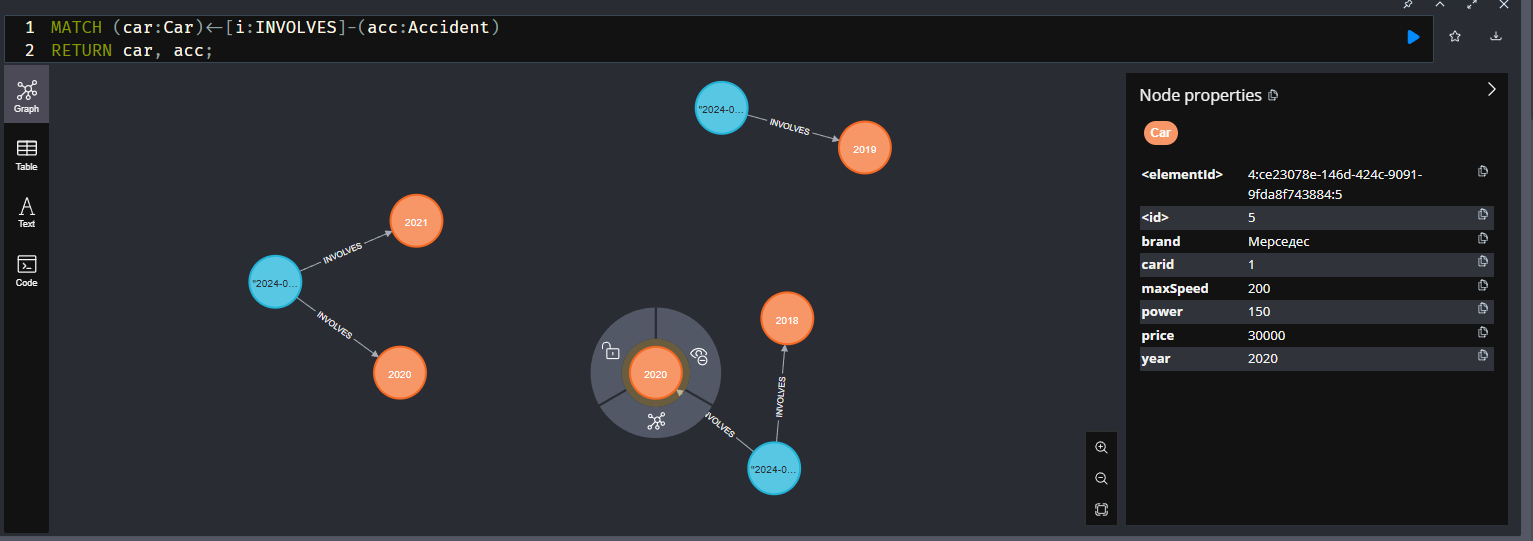


Рисунок 21 - Запрос на получение автомобилей, участвующих в ДТП

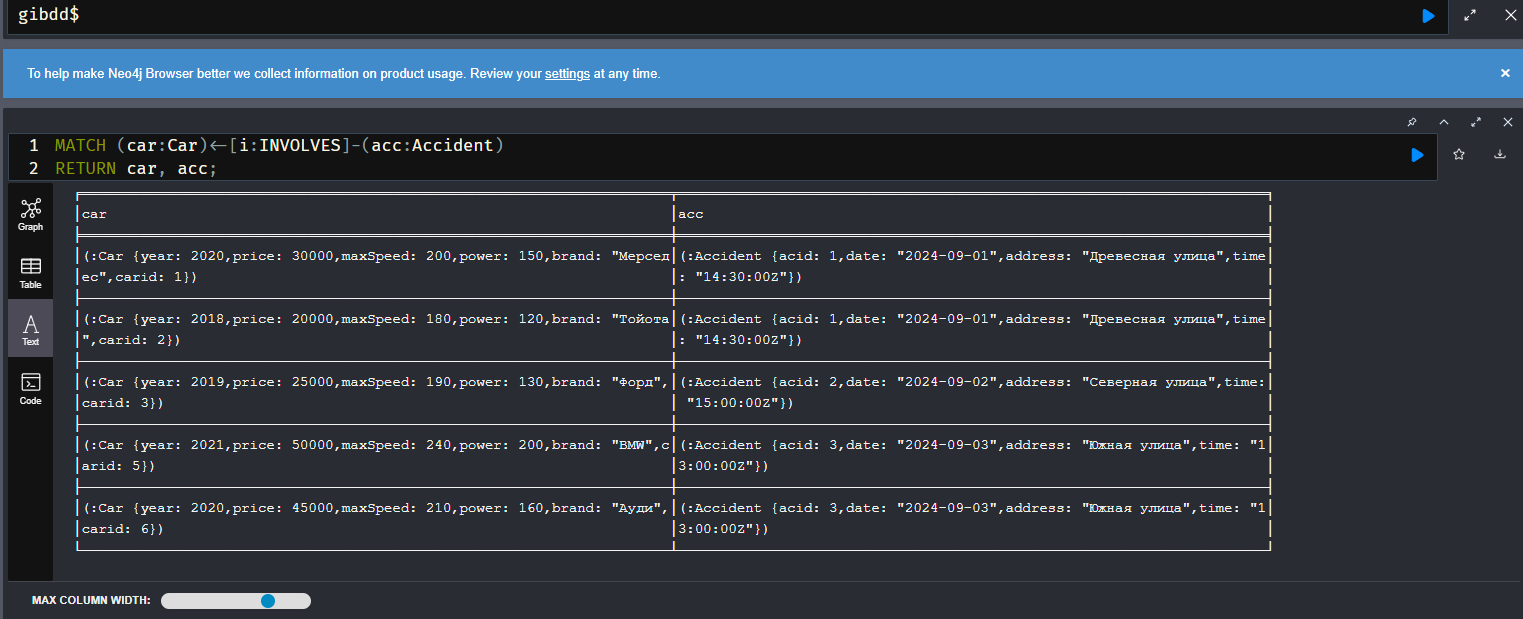


Рисунок 22 - Запрос на получение автомобилей, участвующих в ДТП

Таблица 5 – Код запроса на получение автомобилей, участвующих в ДТП

|  |
| --- |
| MATCH (car:Car)<-[i:INVOLVES]-(acc:Accident)  RETURN car, acc; |

Запрос на получение владельцев автомобилей представлен на рисунках 23 и 24, а также в таблице 6.

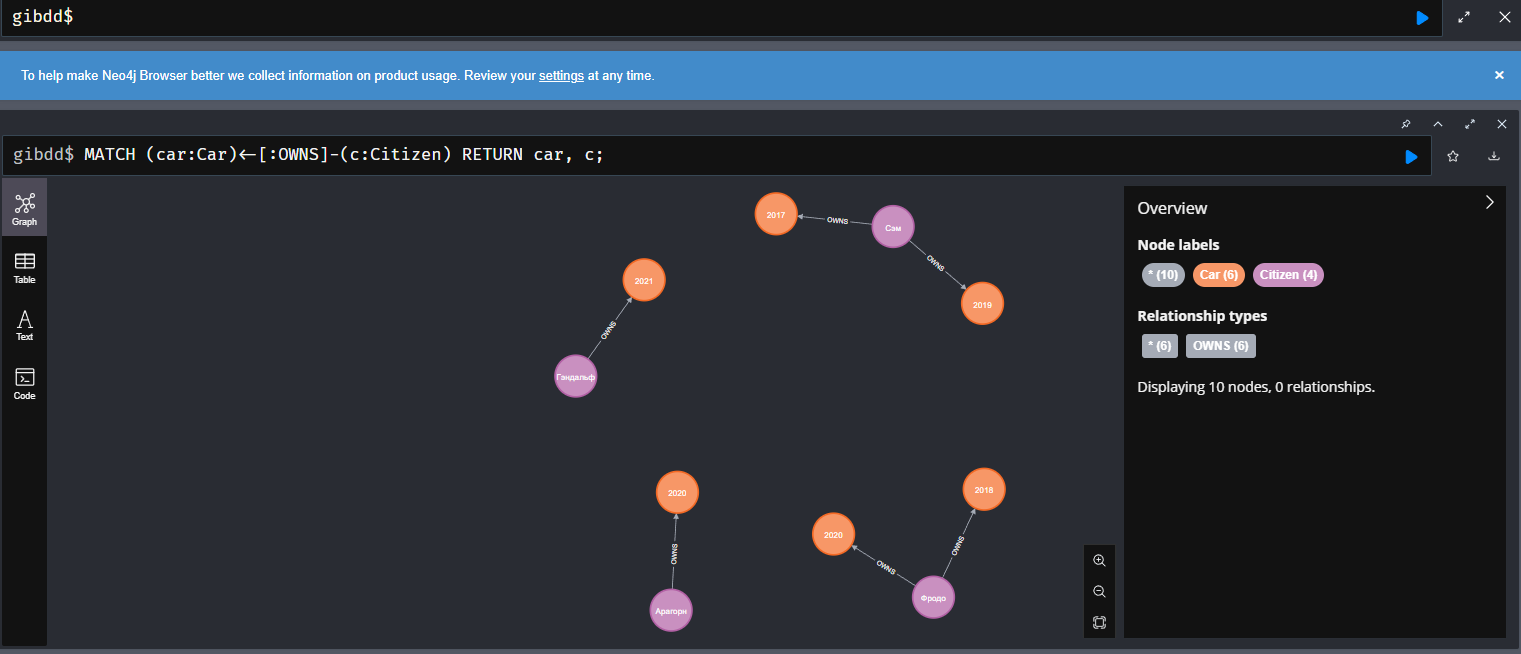


Рисунок 23 - Запрос на получение владельцев автомобилей

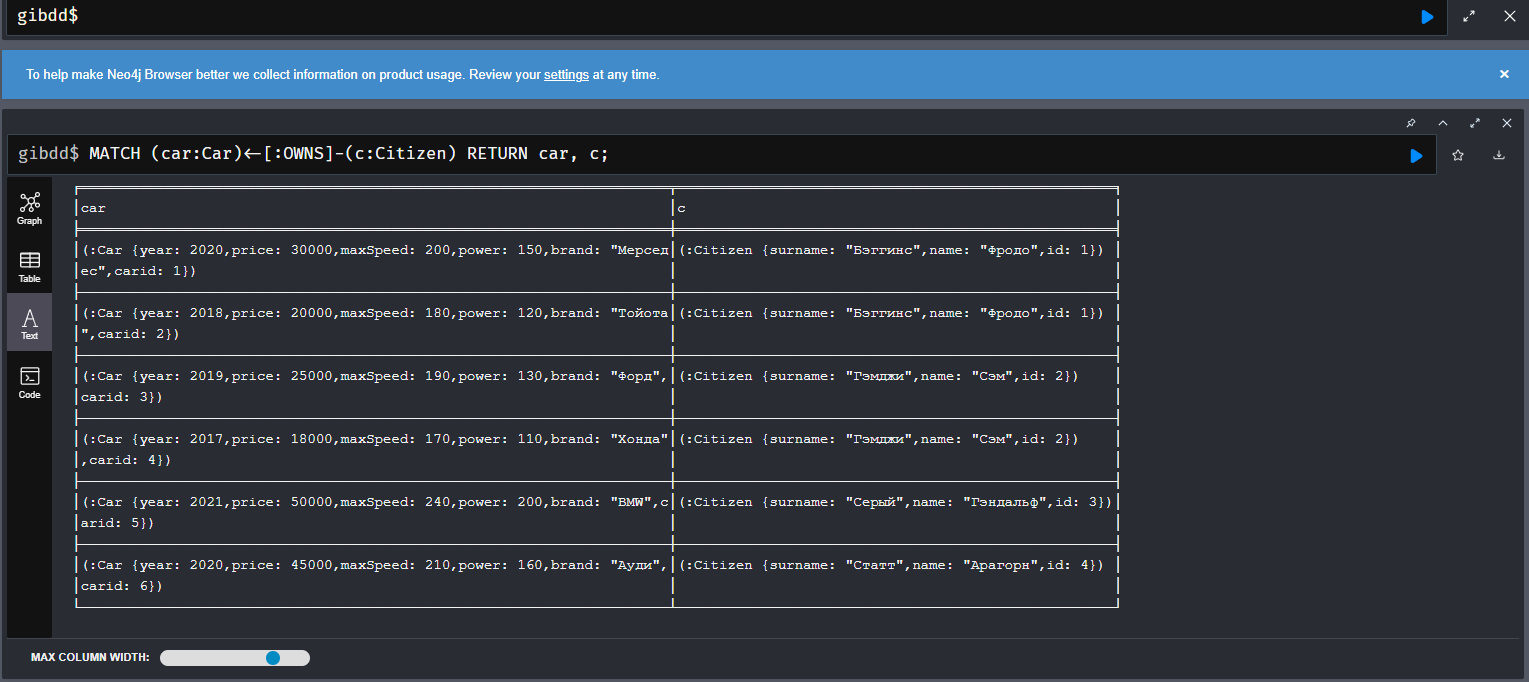


Рисунок 24 - Запрос на получение владельцев автомобилей

Таблица 6 – Код запроса на получение владельцев автомобилей

|  |
| --- |
| MATCH (car:Car)<-[:OWNS]-(c:Citizen)  RETURN car, c; |

Запрос на получение какими авто владеет Фродо Бэггинс представлен на рисунке 25 и в таблице 7.

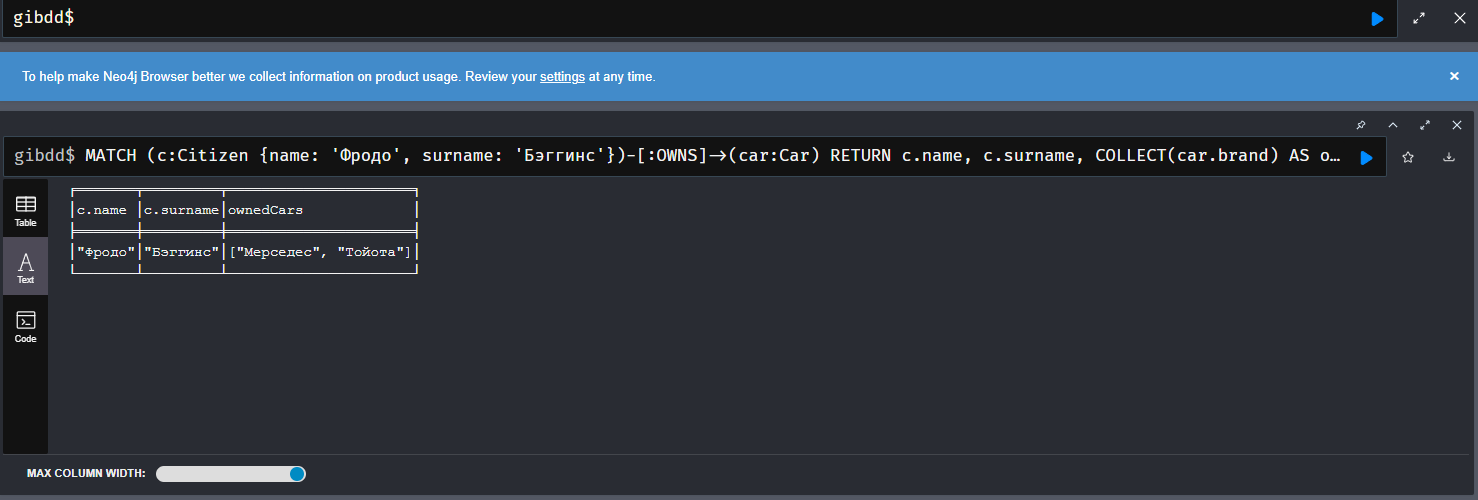


Рисунок 25 - Запрос на получение какими авто владеет Фродо Бэггинс

Таблица 7 – Код запроса на получение какими авто владеет Фродо Бэггинс

|  |
| --- |
| MATCH (c:Citizen {name: 'Фродо', surname: 'Бэггинс'})-[:OWNS]->(car:Car)  RETURN c.name, c.surname, COLLECT(car.brand) AS ownedCars |

Запрос на получение стоимости каждого автомобиля представлен на рисунке 26 и в таблице 8.

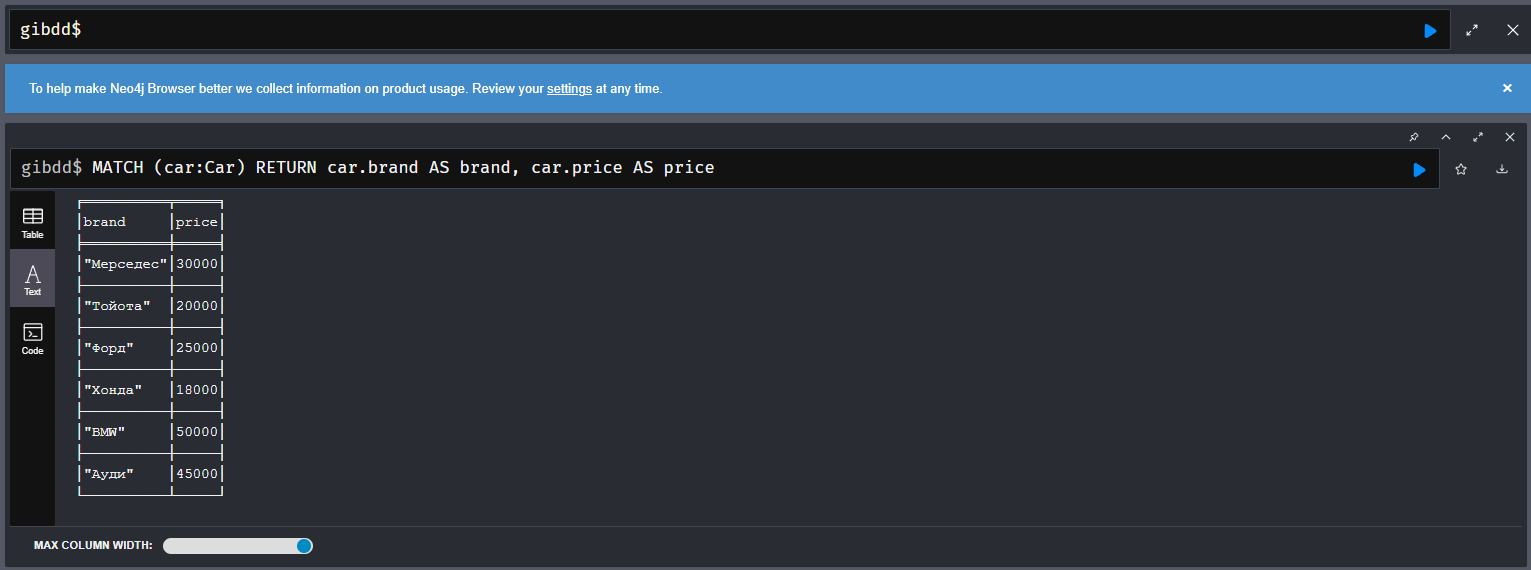


Рисунок 26 - Запрос на получение стоимости каждого автомобиля

Таблица 8 – Код запроса на получение стоимости каждого автомобиля

|  |
| --- |
| MATCH (car:Car)  RETURN car.brand AS brand, car.price AS price |

Запрос на получение 3 самых дорогих автомобиля и кто ими владеет представлен на рисунке 27 и в таблице 9.

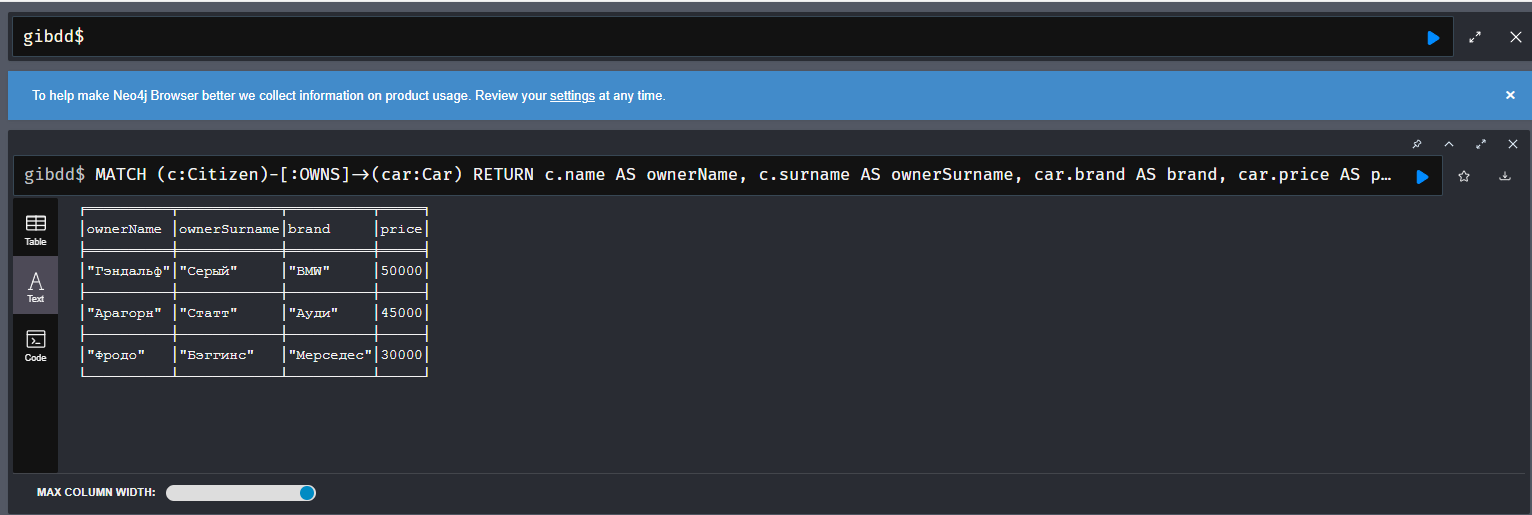


Рисунок 27 - Запрос на получение 3 самых дорогих автомобиля и кто ими владеет

Таблица 9 – Код запроса на получение 3 самых дорогих автомобиля и кто ими владеет

|  |
| --- |
| MATCH (c:Citizen)-[:OWNS]->(car:Car)  RETURN c.name AS ownerName, c.surname AS ownerSurname, car.brand AS brand, car.price AS price  ORDER BY car.price DESC  LIMIT 3 |

Запрос на получение информации кто каким автомобилем владеет и какая их общая стоимость, т.е. если у человека 2 автомобиля, то в соответствующей графе (totalPrice) будет отображаться сумма стоимости, имеющихся у человека автомобилей. Данный запрос представлен на рисунке 28 и в таблице 10.

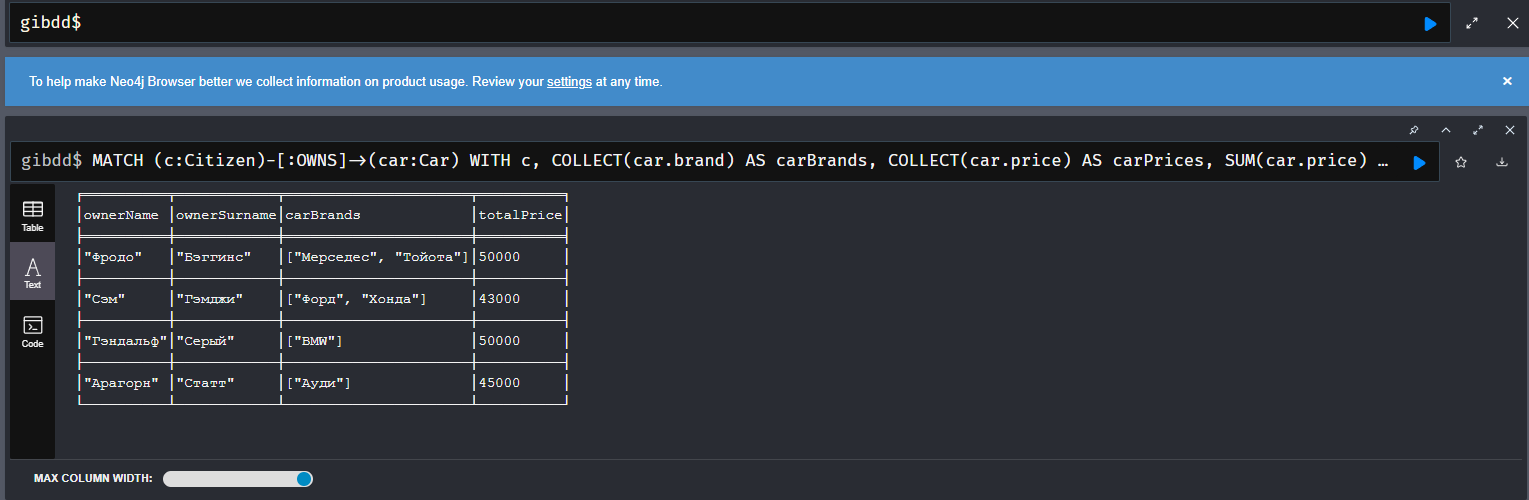


Рисунок 28 - Запрос на получение владельцев автомобилей и их общая стоимость

Таблица 10 – Код запроса на получение владельцев автомобилей и их общая стоимость

|  |
| --- |
| MATCH (c:Citizen)-[:OWNS]->(car:Car)  WITH c, COLLECT(car.brand) AS carBrands, COLLECT(car.price) AS carPrices, SUM(car.price) AS totalPrice  RETURN c.name AS ownerName, c.surname AS ownerSurname, carBrands, totalPrice |

Запрос на получение информации о том, в каком ДТП пострадало больше всего машин представлен на рисунке 29 и в таблице 11.

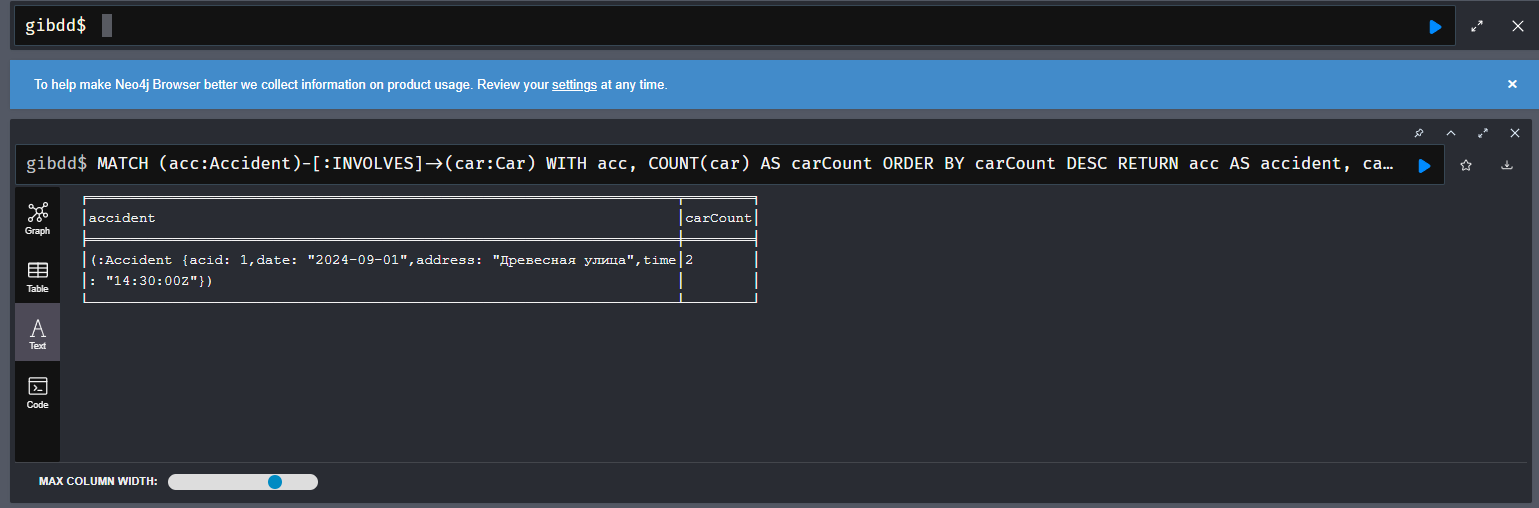


Рисунок 29 - Запрос на получение информации о том, в каком ДТП пострадало больше всего машин

Таблица 11 -Код запроса на получение информации о том, в каком ДТП пострадало больше всего машин

|  |
| --- |
| MATCH (acc:Accident)-[:INVOLVES]->(car:Car)  WITH acc, COUNT(car) AS carCount  ORDER BY carCount DESC  RETURN acc AS accident, carCount  LIMIT 1 |

Вывод: В ходе выполнения лабораторной работы были получены практические навыки по работе с графической базой данных Neo4j, а именно, были освоены некоторые команды для работы с данной средой и на их основе была создана БД по варианту 13 (ГИБДД), а также были созданы запросы различной сложности.