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 *ИКБО-20-23 Комисарик М.А.*
.

_____
(подпись)

Ассистент *Брайловский А.В.*

_____
(подпись)

Отчет представлен «___»_____2025 г.

Москва 2025 г.

ПОСТАНОВКА ЗАДАЧИ

Цель работы: формирование и закрепление у студентов фундаментальных навыков работы с реляционными базами данных на примере СУБД Postgres Pro.

Постановка задачи: 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 РАБОТЫ

1 Анализ и описание ограничений целостности

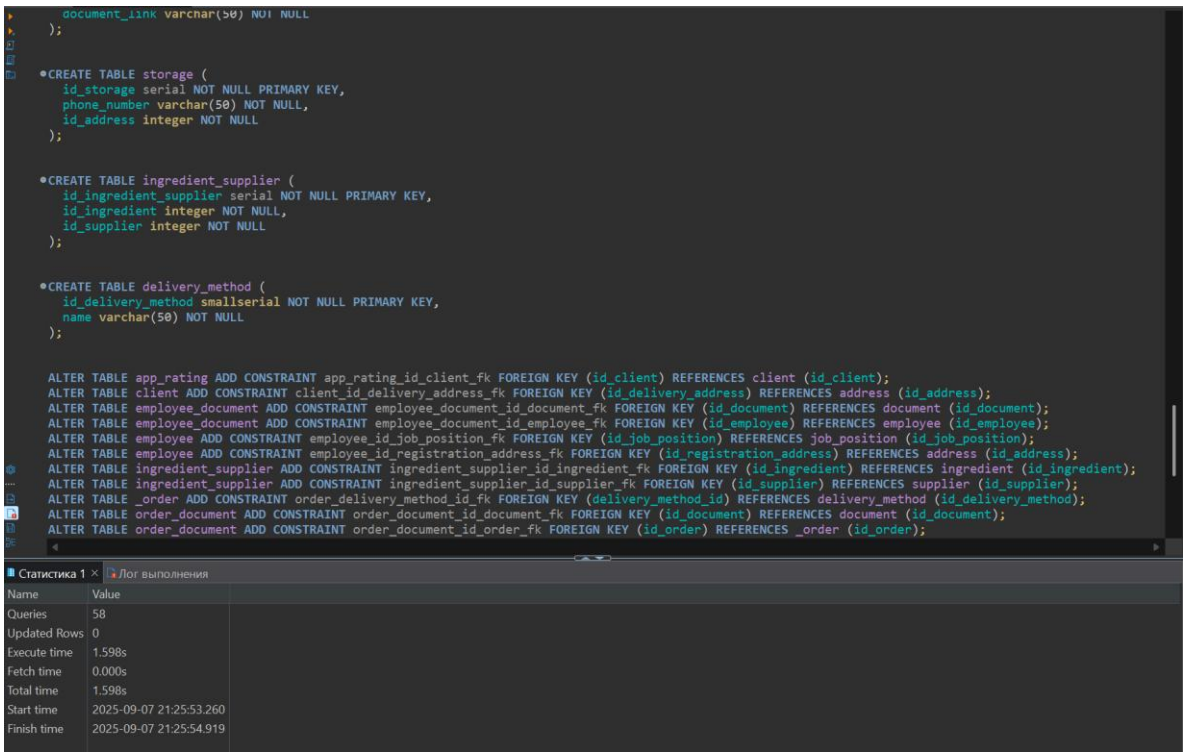
В таблице 1 расположены 5 столбцов из таблицы "_order", для каждого из которых указано ограничение целостности и бизнес-правило.

Таблица 1 – Ограничения целостности таблицы _order

Название столбца	**Тип данных**	**Ограничение**	**Обоснование (Бизнес-правило)**
id_order	SERIAL	PRIMARY KEY	Уникальный идентификатор заказа, генерируется автоматически.
id_client	INT	FOREIGN KEY (client)	Ссылка на пользователя. Заказ не может существовать без пользователя.
id_delivery_address	INT	FOREIGN KEY (address)	Ссылка на адрес доставки. Заказ не может существовать без адреса доставки
order_amount	MONEY	NOT NULL	Стоимость заказа. Любой заказ имеет стоимость, составляемую из цен входящих в него блюд.
date_of_formation	DATE	NOT NULL	Дата формирования заказа. Любой заказ имеет дату своего формирования.

2 Создание структуры данных

На рисунке 1 представлен фрагмент кода запроса для создания всех таблиц и добавления всех ограничений целостности.



```
    document_link varchar(50) NOT NULL
);


CREATE TABLE storage (
    id_storage serial NOT NULL PRIMARY KEY,
    phone_number varchar(50) NOT NULL,
    id_address integer NOT NULL
);


CREATE TABLE ingredient_supplier (
    id_ingredient_supplier serial NOT NULL PRIMARY KEY,
    id_ingredient integer NOT NULL,
    id_supplier integer NOT NULL
);


CREATE TABLE delivery_method (
    id_delivery_method smallserial NOT NULL PRIMARY KEY,
    name varchar(50) NOT NULL
);


ALTER TABLE app_rating ADD CONSTRAINT app_rating_id_client_fk FOREIGN KEY (id_client) REFERENCES client (id_client);
ALTER TABLE client ADD CONSTRAINT client_id_delivery_address_fk FOREIGN KEY (id_delivery_address) REFERENCES address (id_address);
ALTER TABLE employee_document ADD CONSTRAINT employee_document_id_document_fk FOREIGN KEY (id_document) REFERENCES document (id_document);
ALTER TABLE employee_document ADD CONSTRAINT employee_document_id_employee_fk FOREIGN KEY (id_employee) REFERENCES employee (id_employee);
ALTER TABLE employee ADD CONSTRAINT employee_id_job_position_fk FOREIGN KEY (id_job_position) REFERENCES job_position (id_job_position);
ALTER TABLE employee ADD CONSTRAINT employee_id_registration_address_fk FOREIGN KEY (id_registration_address) REFERENCES address (id_address);
ALTER TABLE ingredient_supplier ADD CONSTRAINT ingredient_supplier_id_ingredient_fk FOREIGN KEY (id_ingredient) REFERENCES ingredient (id_ingredient);
ALTER TABLE ingredient_supplier ADD CONSTRAINT ingredient_supplier_id_supplier_fk FOREIGN KEY (id_supplier) REFERENCES supplier (id_supplier);
ALTER TABLE _order ADD CONSTRAINT order_delivery_method_id_fk FOREIGN KEY (delivery_method_id) REFERENCES delivery_method (id_delivery_method);
ALTER TABLE order_document ADD CONSTRAINT order_document_id_document_fk FOREIGN KEY (id_document) REFERENCES document (id_document);
ALTER TABLE order_document ADD CONSTRAINT order_document_id_order_fk FOREIGN KEY (id_order) REFERENCES _order (id_order);
```

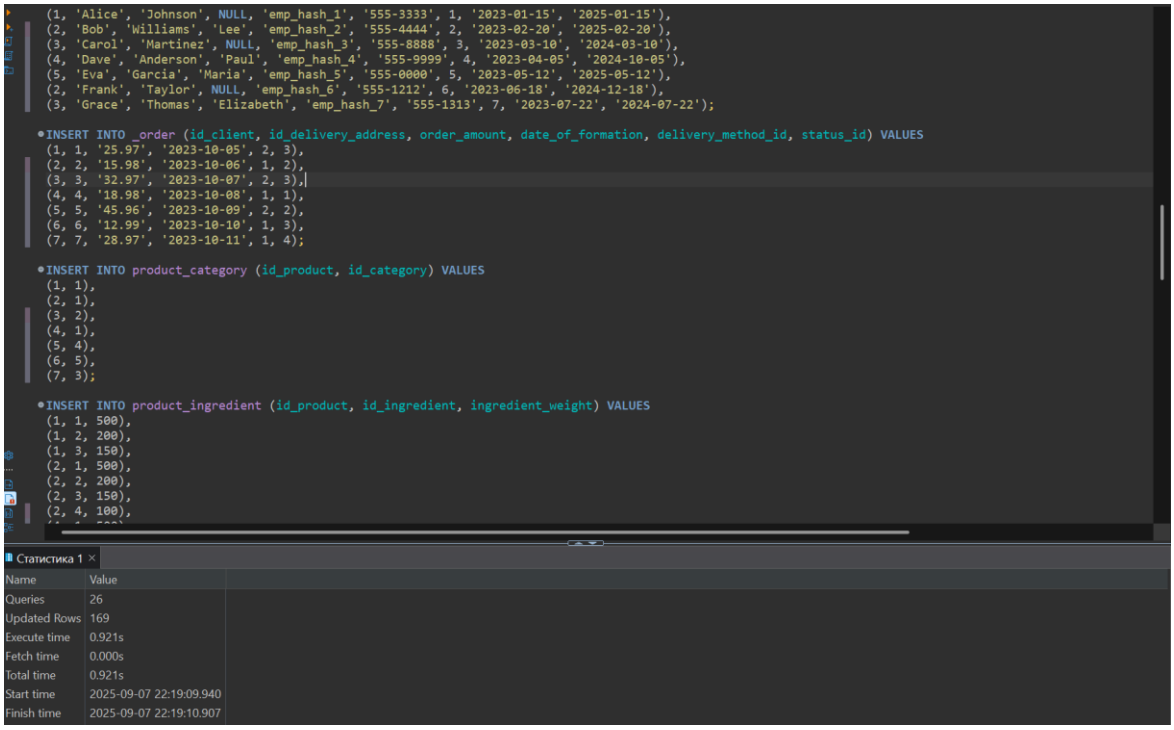
Статистика 1 | Лог выполнения

Name	Value
Queries	58
Updated Rows	0
Execute time	1.598s
Fetch time	0.000s
Total time	1.598s
Start time	2025-09-07 21:25:53.260
Finish time	2025-09-07 21:25:54.919

Рисунок 1 – Создание таблиц и добавление зависимостей

3 Заполнение таблиц данными

На рисунке 2 представлен фрагмент кода запроса для заполнения таблиц данными.



```
(1, 'Alice', 'Johnson', NULL, 'emp_hash_1', '555-3333', 1, '2023-01-15', '2025-01-15'),
(2, 'Bob', 'Williams', 'Lee', 'emp_hash_2', '555-4444', 2, '2023-02-20', '2025-02-20'),
(3, 'Carol', 'Martinez', NULL, 'emp_hash_3', '555-8888', 3, '2023-03-10', '2024-03-10'),
(4, 'Dave', 'Anderson', 'Paul', 'emp_hash_4', '555-9999', 4, '2023-04-05', '2024-10-05'),
(5, 'Eva', 'Garcia', 'Maria', 'emp_hash_5', '555-0000', 5, '2023-05-12', '2025-05-12'),
(2, 'Frank', 'Taylor', NULL, 'emp_hash_6', '555-1212', 6, '2023-06-18', '2024-12-18'),
(3, 'Grace', 'Thomas', 'Elizabeth', 'emp_hash_7', '555-1313', 7, '2023-07-22', '2024-07-22');

INSERT INTO _order (id_client, id_delivery_address, order_amount, date_of_formation, delivery_method_id, status_id) VALUES
(1, 1, '25.97', '2023-10-05', 2, 3),
(2, 2, '15.98', '2023-10-06', 1, 2),
(3, 3, '32.97', '2023-10-07', 2, 3),
(4, 4, '18.98', '2023-10-08', 1, 1),
(5, 5, '45.96', '2023-10-09', 2, 2),
(6, 6, '12.99', '2023-10-10', 1, 3),
(7, 7, '28.97', '2023-10-11', 1, 4);

INSERT INTO product_category (id_product, id_category) VALUES
(1, 1),
(2, 1),
(3, 2),
(4, 1),
(5, 4),
(6, 5),
(7, 3);

INSERT INTO product_ingredient (id_product, id_ingredient, ingredient_weight) VALUES
(1, 1, 500),
(1, 2, 200),
(1, 3, 150),
(2, 1, 500),
(2, 2, 200),
(2, 3, 150),
(2, 4, 100),
```

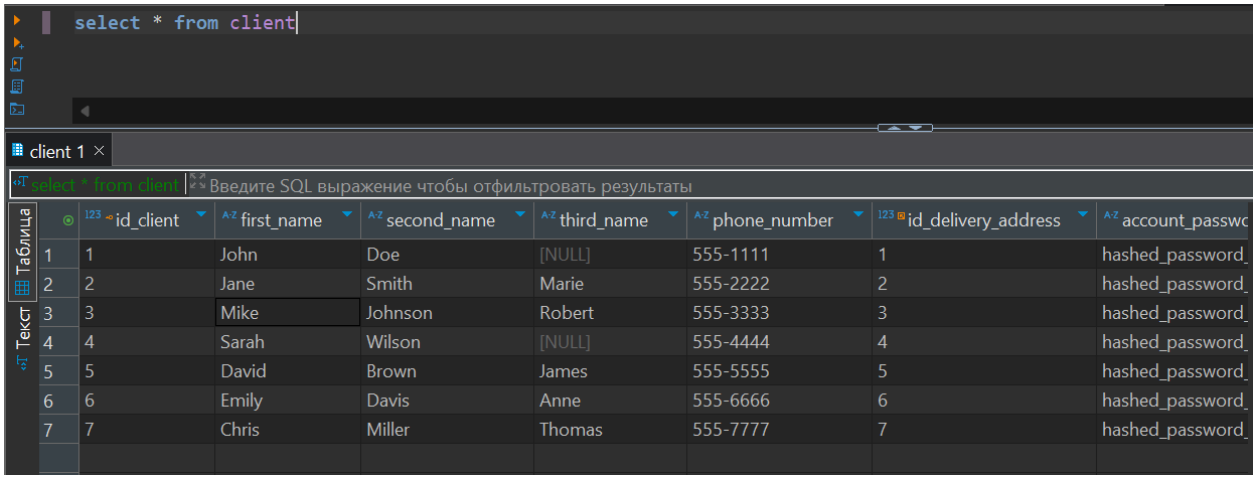
Статистика 1

Name	Value
Queries	26
Updated Rows	169
Execute time	0.921s
Fetch time	0.000s
Total time	0.921s
Start time	2025-09-07 22:19:09.940
Finish time	2025-09-07 22:19:10.907

Рисунок 2 – Заполнение таблиц данными

4 Составление запросов на выборку (часть 1)

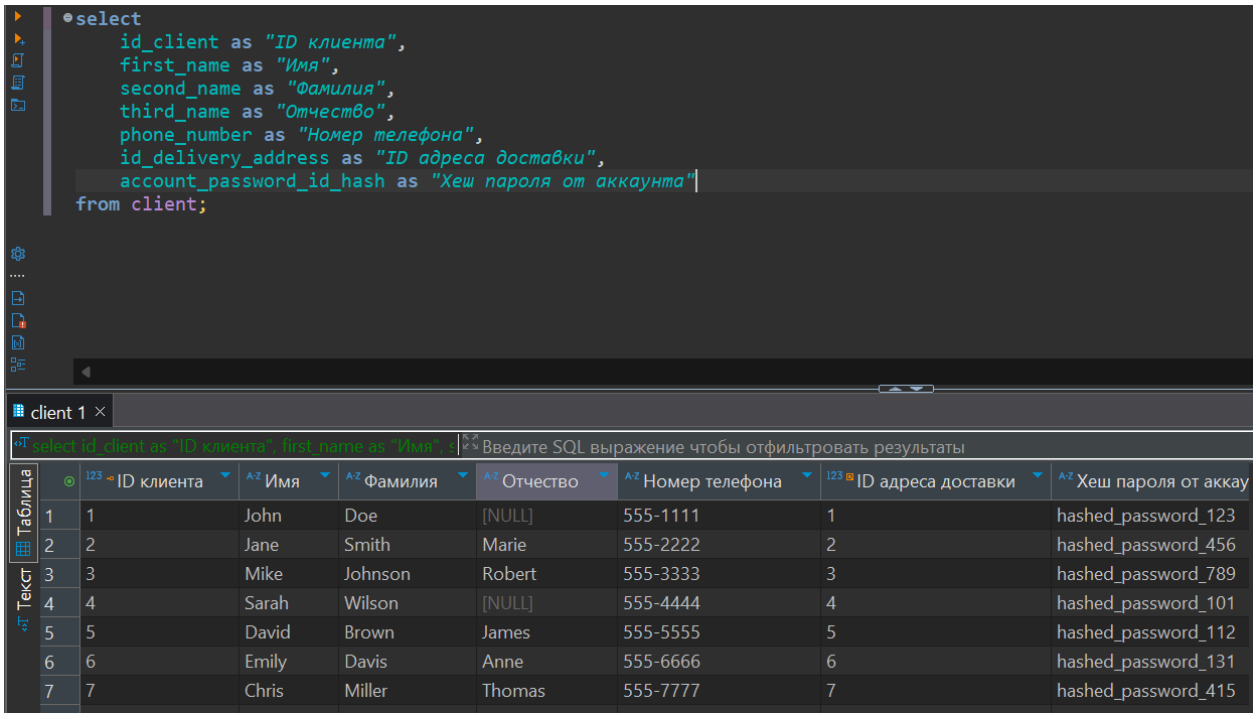
На рисунке 3 представлен запрос на получение всех строк таблицы client.



	id_client	first_name	second_name	third_name	phone_number	id_delivery_address	account_passwc
1	1	John	Doe	[NULL]	555-1111	1	hashed_password_
2	2	Jane	Smith	Marie	555-2222	2	hashed_password_
3	3	Mike	Johnson	Robert	555-3333	3	hashed_password_
4	4	Sarah	Wilson	[NULL]	555-4444	4	hashed_password_
5	5	David	Brown	James	555-5555	5	hashed_password_
6	6	Emily	Davis	Anne	555-6666	6	hashed_password_
7	7	Chris	Miller	Thomas	555-7777	7	hashed_password_

Рисунок 3 – Первый запрос из пункта 4

На рисунке 4 представлен запрос на получение всех строк таблицы client с переименованием всех имен столбцов на выходе.



	ID клиента	Имя	Фамилия	Отчество	Номер телефона	ID адреса доставки	Хеш пароля от аккау
1	1	John	Doe	[NULL]	555-1111	1	hashed_password_123
2	2	Jane	Smith	Marie	555-2222	2	hashed_password_456
3	3	Mike	Johnson	Robert	555-3333	3	hashed_password_789
4	4	Sarah	Wilson	[NULL]	555-4444	4	hashed_password_101
5	5	David	Brown	James	555-5555	5	hashed_password_112
6	6	Emily	Davis	Anne	555-6666	6	hashed_password_131
7	7	Chris	Miller	Thomas	555-7777	7	hashed_password_415

Рисунок 4 – Второй запрос из пункта 4

На рисунке 5 представлен запрос на получение всех строк таблицы client, в колонке "first_name" которых находится строка ‘John’.

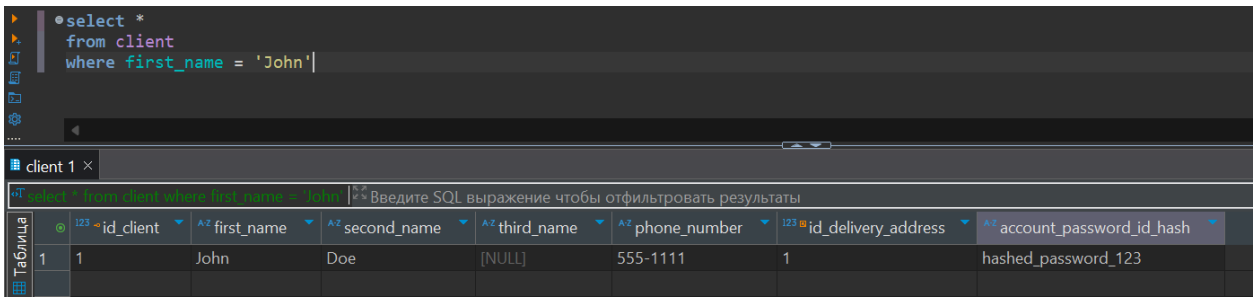


Рисунок 5 – Третий запрос из пункта 4

На рисунке 6 представлен запрос на получение всех строк таблицы product, в колонке "price" которых находится денежная сумма в пределах от 5 до 12 долларов.

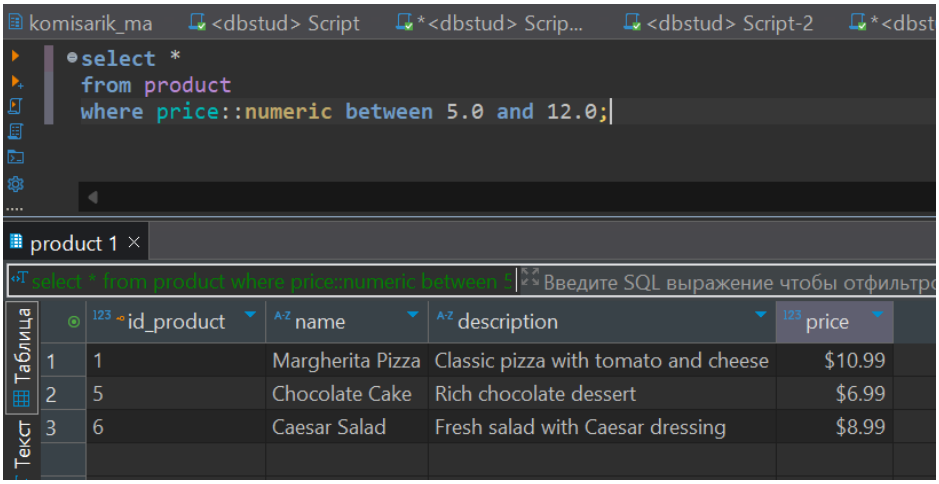


Рисунок 6 – Четвертый запрос из пункта 4

На рисунке 7 представлен запрос на получение всех строк таблицы client, в колонке "first_name" которых находится одна из строк ‘John’, ‘Myla’, ‘Mike’, ‘Eve’, ‘Emily’.

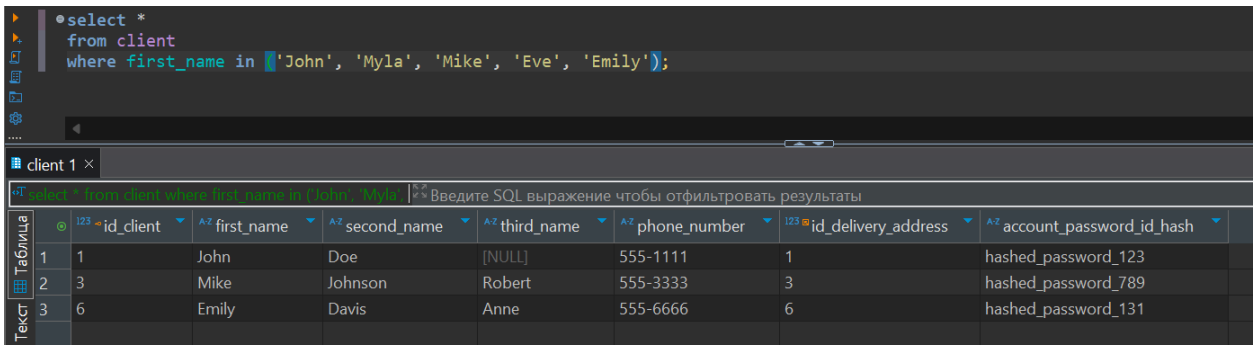


Рисунок 7 – Пятый запрос из пункта 4

На рисунке 8 представлен запрос на получение всех строк таблицы client, в колонке "third_name" которых находится нулевой элемент.

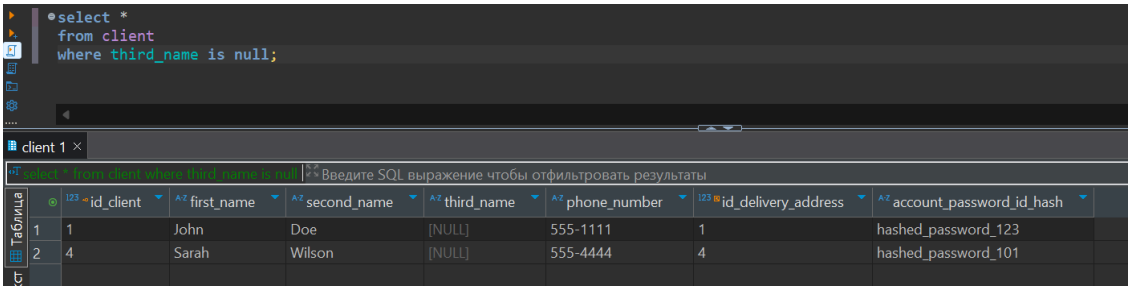


Рисунок 8 – Шестой запрос из пункта 4

5 Составление запросов на выборку (часть 2)

На рисунке 9 представлен запрос на получение всех строк таблицы product, отсортированных по значению столбца "price" в возрастающем порядке.

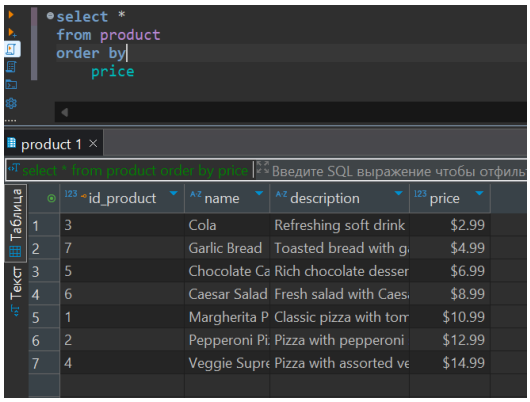


Рисунок 9 – Первый запрос из пункта 5

На рисунке 10 представлен запрос на получение всех строк таблицы product, отсортированных по значению столбца "price" в убывающем порядке.

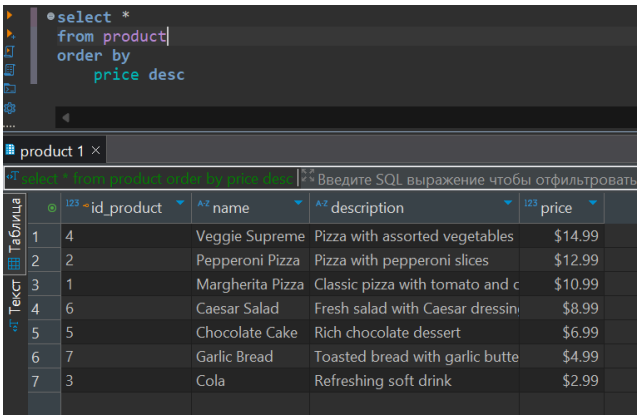


Рисунок 10 – Второй запрос из пункта 5

На рисунке 11 представлен запрос на получение веса каждого продукта, составленного из весов каждого входящего ингредиента.

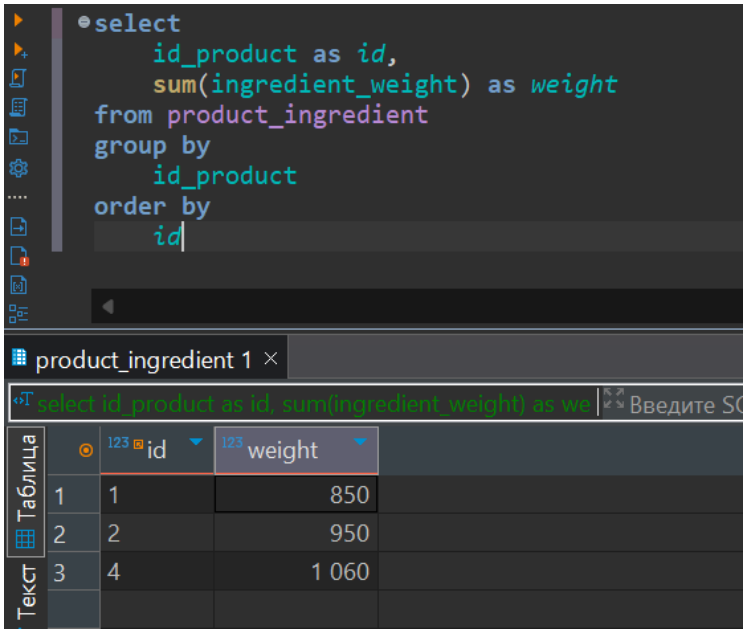


Рисунок 11 – Третий запрос из пункта 5

На рисунке 12 представлен запрос на получение количества входящих блюд (не уникальных) в каждый заказ.

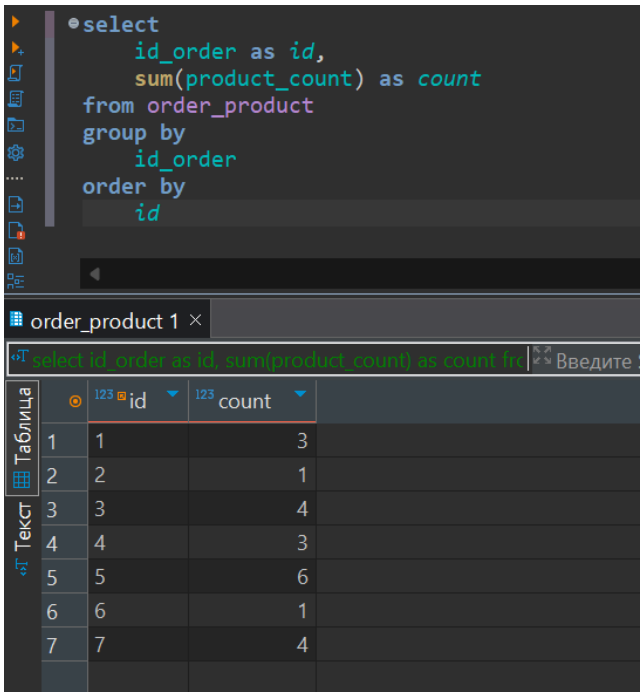


Рисунок 12 – Четвертый запрос из пункта 5

На рисунке 13 представлен запрос на получение всех заказов, которые включают в себя более 1 уникального блюда.

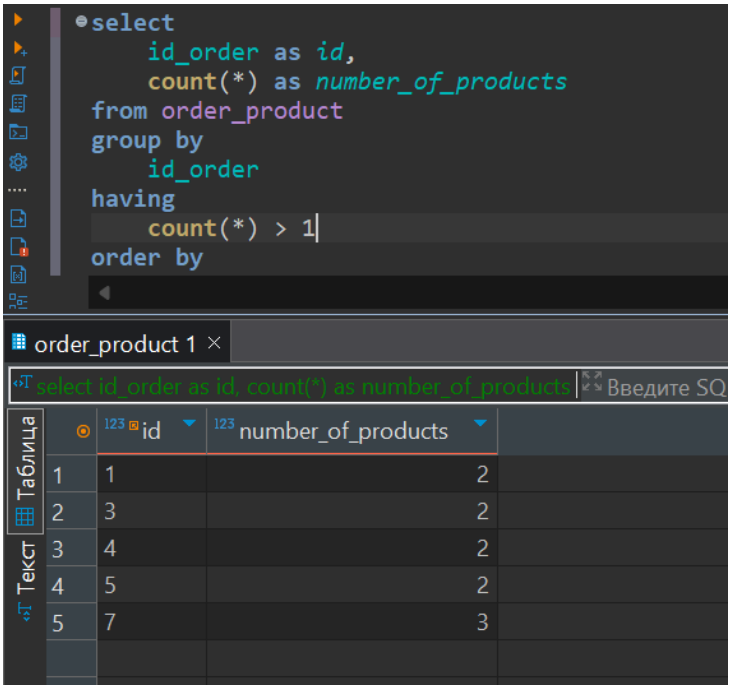


Рисунок 13 – Пятый запрос из пункта 5

На рисунке 14 представлен запрос на получение всех блюд, которые весят меньше килограмма.

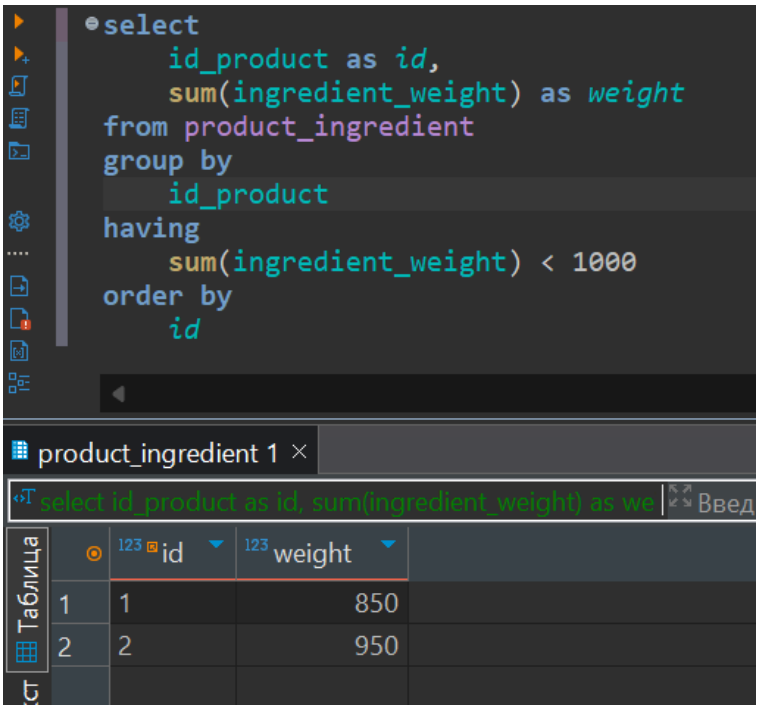


Рисунок 14 – Шестой запрос из пункта 5