



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 №6

Студенты группы

ИКБО-65-23 Олефиров.Г.Г.

(подпись)

Ассистент

Морозов Д.В.

(подпись)

Отчет представлен

«___»_____ 2025 г.

Москва 2025 г.

ПРАКТИЧЕСКАЯ РАБОТА №6.

ТРИГГЕРЫ И КУРСОРЫ В POSTGRESQL

Постановка задачи:

Для выполнения практической работы необходимо последовательно выполнить два задания, демонстрирующих применение триггеров и курсоров.

Ваша задача — адаптировать каждую из поставленных задач к логической структуре и предметной области вашей базы данных.

Задание №1: триггеры

Проанализировать предметную область своей базы данных и выявить не менее трёх бизнес-правил, реализация которых в виде ограничений целостности возможна только с помощью триггеров.

ВАЖНО: триггеры должны быть разными.

Для каждого правила:

- описать алгоритм его работы, указав таблицу, событие и последовательность действий;
- написать код триггерной функции на PL/pgSQL и оператор CREATE TRIGGER;
- продемонстрировать работу триггера на примерах DML-операций, которые как успешно выполняются, так и корректно прерываются триггером (два запроса).

1. Аудит и логирование.

Запись факта изменения данных.

Пример: при любом изменении столбца salary в таблице employees создавать запись в salary_log, сохраняя OLD.salary, NEW.salary и CURRENT_USER.

2. Сложная валидация (проверка).

Запрет операции на основе данных из другой таблицы.

Пример: триггер BEFORE INSERT на таблицу order_items должен проверить, что NEW.quantity не превышает quantity_on_hand в таблице products. Если превышает – вызвать RAISE EXCEPTION.

3. Поддержание согласованности (денормализация).

Автоматическое обновление связанных данных.

Пример: триггер AFTER INSERT на таблицу sales автоматически обновляет столбец total_spent в таблице customers.

4. Защита данных.

Запрет определенных операций.

Пример: триггер BEFORE DELETE на таблицу departments может запретить удаление, если в таблице employees еще есть сотрудники,

ссылающиеся на этот отдел.

Задание №2: курсоры

Разработать два скрипта на PL/pgSQL, демонстрирующих оба способа обработки данных:

- С использованием явного курсора (DECLARE/OPEN/FETCH/CLOSE).
- С использованием неявного курсора (цикл FOR...IN).

Каждый SQL-запрос сопроводить комментарием, объясняющим его назначение и логику работы с учетом специфики вашей базы данных.

1.1 Используемые таблицы

Order								
Таблица	id_order	id_address	id_client	id_discount_card	order_amount	order_time	order_status	
1	1	1	1	1	1 500	10:30:00	Доставлен	
2	2	3	2	2	850	11:45:00	В обработке	
3	3	4	3	3	1 200	12:15:00	Готов к выдаче	
4	4	5	4	4	650	13:20:00	Доставляется	
5	5	6	5	5	1 800	14:30:00	Принят	

Таблица 1 – Order

address			
Таблица	id_address	id_client	delivery_address
1	1	1	ул. Ленина, д. 10, кв. 5
2	2	1	ул. Мира, д. 15, кв. 12
3	3	2	пр. Победы, д. 20, кв. 3
4	4	3	ул. Садовая, д. 8, кв. 7
5	5	4	ул. Центральная, д. 3, кв. 9
6	6	5	ул. Молодежная, д. 12, кв. 4

Таблица 2 – Address

bakery			
Таблица	id_bakery	address	telephone
1	1	ул. Ленина, д. 25	+7 (495) 123-45-67
2	2	пр. Победы, д. 10	+7 (495) 234-56-78
3	3	ул. Мира, д. 5	+7 (495) 345-67-89
4	4	ул. Садовая, д. 15	+7 (495) 456-78-90

Таблица 3 – Bakery

bakery_product | Введите SQL выражение чтобы отфильтровать результаты

Таблица	123 ↗ id_bakery_product	123 ↗ id_warehouse	123 ↗ id_bakery	A-Z title	A-Z description	123 price
1	1	1	1	Бородинский хлеб	Ржаной хлеб с тмином	85,5
2	2	1	1	Батон нарезной	Пшеничный батон	65
3	3	2	2	Торт Наполеон	Слоеный торт с заварным кремом	1 200
4	4	2	2	Эклеры	Пирожное с заварным кремом	45
5	5	3	3	Пирог с яблоками	Дрожжевой пирог с яблочной начинкой	350
6	6	3	3	Печенье овсяное	Печенье с изюмом и овсяными хлопьями	25
7	7	1	4	Круассан	Слоеное изделие с шоколадом	75
8	8	2	4	Чизкейк	Десерт из творожного сыра	280

Таблица 4 – Bakery_product

category_bakery | Введите SQL выражение чтобы отфильтровать результаты

Таблица	123 ↗ id_category_bakery	123 ↗ id_product_category	123 ↗ id_bakery_product
1	1	1	1
2	2	1	2
3	3	4	3
4	4	2	4
5	5	3	5
6	6	5	6
7	7	2	7
8	8	4	8

Таблица 5 – Category_bakery

client | Введите SQL выражение чтобы отфильтровать результаты

Таблица	123 ↗ id_client	A-Z surname	A-Z name	A-Z middle_name	A-Z phone_number
1	1	Иванов	Иван	Иванович	+7 (916) 123-45-67
2	2	Петрова	Мария	Сергеевна	+7 (917) 234-56-78
3	3	Сидоров	Алексей	Петрович	+7 (918) 345-67-89
4	4	Кузнецова	Елена	Владимировна	+7 (919) 456-78-90
5	5	Смирнов	Дмитрий	Александрович	+7 (920) 567-89-01

Таблица 6 – Client

customers | Введите SQL выражение чтобы отфильтровать результаты

Таблица	123 ↗ customer_id	A-Z full_name	A-Z phone
1	1	Иван Иванов	89001112233
2	2	Петр Петров	89002223344
3	3	Анна Смирнова	89003334455
4	4	Мария Кузнецова	89004445566
5	5	Сергей Сидоров	89005556677

Таблица 7 – Customers

discount_card | Введите SQL выражение чтобы отфильтровать результаты

Таблица	123 ↗ id_discount_card	123 ↗ id_client	123 discount_amount
1	1	1	5
2	2	2	10
3	3	3	7
4	4	4	15
5	5	5	3

Таблица 8 – Discount_card

document | Введите SQL выражение чтобы отфильтровать результаты

Таблица	123 ↗ id_document	123 ↗ id_order	⌚ date_of_formation	Α-Ζ link_to_the_document
1	1	1	2024-01-15	/documents/order_1.pdf
2	2	2	2024-01-15	/documents/order_2.pdf
3	3	3	2024-01-15	/documents/order_3.pdf
4	4	4	2024-01-15	/documents/order_4.pdf
5	5	5	2024-01-15	/documents/order_5.pdf

Таблица 9 – Document

employee | Введите SQL выражение чтобы отфильтровать результаты

Таблица	123 ↗ id_employee	123 ↗ id_position	Α-Ζ surname	Α-Ζ name	Α-Ζ middle_name	Α-Ζ telephone	Α-Ζ registration_address	⌚ date_of_employment
1	1	1	Васильев	Андрей	Николаевич	+7 (911) 111-11-11	ул. Рабочая, д. 1, кв. 1	2023-01-15
2	2	2	Николаева	Ольга	Ивановна	+7 (922) 222-22-22	ул. Трудовая, д. 2, кв. 2	2023-02-20
3	3	3	Козлова	Ирина	Петровна	+7 (933) 333-33-33	ул. Заводская, д. 3, кв. 3	2023-03-10
4	4	4	Федоров	Сергей	Александрович	+7 (944) 444-44-44	ул. Транспортная, д. 4, кв. 4	2023-04-05
5	5	5	Дмитриева	Анна	Владимировна	+7 (955) 555-55-55	ул. Управленческая, д. 5, кв.	2023-05-12

Таблица 10 –Employee

load | Введите SQL выражение чтобы отфильтровать результаты

Таблица	123 ↗ id_load	123 ↗ id_employee	123 ↗ id_order
1	1	4	1
2	2	4	2
3	3	4	3
4	4	4	4
5	5	4	5

Таблица 11 – Load

order_items | Введите SQL выражение чтобы отфильтровать результаты

Таблица	order_item_id	order_id	product_id	quantity
1	1	1	1	2
2	2	1	4	1
3	3	2	2	3
4	4	3	3	2
5	5	4	5	5
6	6	5	1	1
7	7	5	2	2

Таблица 12 – Order_items

orders | Введите SQL выражение чтобы отфильтровать результаты

Таблица	order_id	customer_id	order_date
1	1	1	2025-09-01
2	2	2	2025-09-02
3	3	3	2025-09-03
4	4	4	2025-09-04
5	5	5	2025-09-05

Таблица 13 –Orders

post | Введите SQL выражение чтобы отфильтровать результаты

Таблица	id_position	title	salary
1	1	Пекарь	45 000
2	2	Кондитер	50 000
3	3	Кассир	35 000
4	4	Курьер	40 000
5	5	Менеджер	60 000

Таблица 14 – Post

product_category | Введите SQL выражение чтобы отфильтровать результаты

Таблица	id_product_category	description
1	1	Хлеб
2	2	Кондитерские изделия
3	3	Пироги
4	4	Торты
5	5	Печенье

Таблица 15 – Product_category

products | Введите SQL выражение чтобы отфильтровать результаты

Таблица	product_id	name	price	quantity_in_stock
1	1	Булочка с корицей	50	100
2	2	Булочка с маком	45	80
3	3	Слойка с яблоком	60	50
4	4	Круассан	75	30
5	5	Булочка с шоколадом	70	40

Таблица 16 – Products

review | Введите SQL выражение чтобы отфильтровать результаты

Таблица	id_review	id_bakery	id_client	evaluation	content
1	1	1	1	5	Отличный хлеб, всегда свежий!
2	2	2	2	4	Вкусные торты, но дорого
3	3	3	3	5	Пироги просто восхитительные!
4	4	4	4	3	Нормально, но можно лучше
5	5	1	5	5	Лучшая пекарня в городе!

Таблица 17 – Review

transaction | Введите SQL выражение чтобы отфильтровать результаты

Таблица	id_transaction	id_order	transaction_time
1	1	1	2024-01-15 10:35:00.000
2	2	2	2024-01-15 11:50:00.000
3	3	3	2024-01-15 12:20:00.000
4	4	4	2024-01-15 13:25:00.000
5	5	5	2024-01-15 14:35:00.000

Таблица 18 – Transaction

warehouse | Введите SQL выражение чтобы отфильтровать результаты

Таблица	id_warehouse	address
1	1	ул. Центральная, д. 1, склад №1
2	2	ул. Промышленная, д. 15, склад №2
3	3	ул. Складская, д. 7, склад №3

19 - Warehouse

1.2.1 Триггер 1: Проверка минимальной цены товара

The screenshot shows the DBeaver interface with the following details:

- Project - General** pane on the left.
- Базы данных** (Databases) pane on the left, showing databases: oknarkin_mi, oknuev_ue, okunkov_pd, and olefirov_gg. Under olefirov_gg, there are tables: Order, address, bakery, bakery_product, category_bakery, client, customers, discount_card, document, employee, load, order_items, orders, post, product_category, products, review, transaction, and warehouse.
- Script Editor** tab bar at the top: <dbstud> Script, <dbstud> Script-2, <dbstud> Script-6.
- Script Content**:

```
CREATE OR REPLACE FUNCTION bakery_check_min_price()
RETURNS TRIGGER AS $$
BEGIN
    -- Проверяем, что цена не ниже минимальной (10 рублей)
    IF NEW.Price < 10 THEN
        RAISE EXCEPTION 'Цена товара не может быть ниже 10 рублей. Указана: %', NEW.Price;
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```
- Statistics** pane on the right:

 - Статистика 1: Updated Rows = 0, Execute time = 0.012s, Start time = Wed Nov 12 22:34:20 MSK 2025, Finish time = Wed Nov 12 22:34:20 MSK 2025.
 - Query: CREATE OR REPLACE FUNCTION bakery_check_min_price() RETURNS TRIGGER AS \$\$ BEGIN -- Проверяем, что цена не ниже минимальной (10 рублей) IF NEW.Price < 10 THEN RAISE EXCEPTION 'Цена товара не может быть ниже 10 рублей. Указана: %', NEW.Price; END IF; RETURN NEW; END; \$\$ LANGUAGE plpgsql;

Рисунок 1 – Результат создания 1 триггерной функции

The screenshot shows the DBeaver interface with the following details:

- Project - General** pane on the left.
- Базы данных** (Databases) pane on the left, showing databases: oknarkin_mi, oknuev_ue, okunkov_pd, and olefirov_gg. Under olefirov_gg, there are tables: Order, address, bakery, bakery_product, category_bakery, client, customers, discount_card, document, employee, load, order_items, orders, post, product_category, products, review, transaction, and warehouse.
- Script Editor** tab bar at the top: <dbstud> Script, <dbstud> Script-2, <dbstud> Script-6.
- Script Content**:

```
CREATE OR REPLACE FUNCTION bakery_check_min_price()
RETURNS TRIGGER AS $$
BEGIN
    -- Проверяем, что цена не ниже минимальной (10 рублей)
    IF NEW.Price < 10 THEN
        RAISE EXCEPTION 'Цена товара не может быть ниже 10 рублей. Указана: %', NEW.Price;
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

Below this, a new section is added:

```
CREATE TRIGGER bakery_trg_min_price
BEFORE INSERT OR UPDATE ON Bakery_product
FOR EACH ROW
EXECUTE FUNCTION bakery_check_min_price();
```
- Statistics** pane on the right:

 - Статистика 1: Queries = 2, Updated Rows = 0, Execute time = 0.029s, Fetch time = 0.000s, Total time = 0.029s, Start time = 2025-11-12 22:33:55.079, Finish time = 2025-11-12 22:33:55.122.

Рисунок 2 – Результат привязки 1 триггерной функции

1.2.2 Демонстрация (тестирование) работы триггера

The screenshot shows the DBBeaver interface with the following details:

- Left Panel:** Shows the database structure under "Базы данных" (Databases). It includes a tree view of tables and a "Project - General" section with "Название" (Name) and "Источник данных" (Data Source).
- Central Panel:** Displays the SQL script for the trigger function "bakery_check_min_price()".

```
CREATE OR REPLACE FUNCTION bakery_check_min_price()
RETURNS TRIGGER AS $$
BEGIN
    -- Проверяем, что цена не ниже минимальной (10 рублей)
    IF NEW.Price < 10 THEN
        RAISE EXCEPTION 'Цена товара не может быть ниже 10 рублей. Указана: %', NEW.Price;
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

- Bottom Panel:** Shows the "Execution Error" section with the following message:

Error occurred during SQL query execution
Причина:
SQL Error [P0001]: ERROR: Цена товара не может быть ниже 10 рублей. Указана: 5
Где: PL/pgSQL function bakery_check_min_price() line 5 at RAISE

Рисунок 4 – Результаты проверки

1.3.1 Триггер 2: Автоматическое назначение курьера на заказ

The screenshot shows the DBBeaver interface with the following details:

- Left Panel:** Shows the database structure under "Базы данных" (Databases). It includes a tree view of tables and a "Project - General" section with "Название" (Name) and "Источник данных" (Data Source).
- Central Panel:** Displays the SQL script for the trigger function "bakery_auto_assign_courier()".

```
CREATE OR REPLACE FUNCTION bakery_auto_assign_courier()
RETURNS TRIGGER AS $$
DECLARE
    courier_id BIGINT;
BEGIN
    IF NEW.Order_Status = 'Доставляется' AND OLD.Order_Status != 'Доставляется' THEN
        SELECT ID_Employee INTO courier_id
        FROM Employee
        WHERE ID_Position = 4 -- ID должности "Курьер"
        LIMIT 1;
        IF courier_id IS NOT NULL THEN
            INSERT INTO Load (ID_Employee, ID_Order)
            VALUES (courier_id, NEW.ID_Order);
            RAISE NOTICE 'Курьер % назначен на заказ %', courier_id, NEW.ID_Order;
        ELSE
            RAISE NOTICE 'Нет доступных курьеров для заказа %', NEW.ID_Order;
        END IF;
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```
- Bottom Panel:** Shows the "Статистика 1" (Statistics 1) section with the following data:

Name	Value
Updated Rows	0
Execute time	0.015s
Start time	Wed Nov 12 22:46:57 MSK 2025
Finish time	Wed Nov 12 22:46:57 MSK 2025
Query	-- Триггерная функция для назначения курьера CREATE OR REPLACE FUNCTION bakery_auto_assign_courier() RETURNS TRIGGER AS \$\$ DECLARE courier_id BIGINT; BEGIN IF NEW.Order_Status = 'Доставляется' AND OLD.Order_Status != 'Доставляется' THEN SELECT ID_Employee INTO courier_id FROM Employee WHERE ID_Position = 4 -- ID должности "Курьер" LIMIT 1; IF courier_id IS NOT NULL THEN INSERT INTO Load (ID_Employee, ID_Order) VALUES (courier_id, NEW.ID_Order); RAISE NOTICE 'Курьер % назначен на заказ %', courier_id, NEW.ID_Order; ELSE RAISE NOTICE 'Нет доступных курьеров для заказа %', NEW.ID_Order; END IF; END IF; RETURN NEW; END; \$\$ LANGUAGE plpgsql;

Рисунок 5 – Результат создания 2 триггерной функции

```

CREATE OR REPLACE FUNCTION bakery_auto_assign_courier()
RETURNS TRIGGER AS $$
DECLARE
    courier_id BIGINT;
BEGIN
    IF NEW.Order_status = 'Доставляется' AND OLD.Order_status != 'Доставляется' THEN
        SELECT ID_Employee INTO courier_id
        FROM Employee
        WHERE ID_Position = 4 -- ID должности "Курьер"
        LIMIT 1;
    IF courier_id IS NOT NULL THEN
        INSERT INTO Load (ID_Employee, ID_Order)
        VALUES (courier_id, NEW.ID_Order);
        RAISE NOTICE 'Курьер % назначен на заказ %', courier_id, NEW.ID_Order;
    ELSE
        RAISE NOTICE 'Нет доступных курьеров для заказа %', NEW.ID_Order;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Создаем триггер
CREATE TRIGGER bakery_trg_auto_courier
AFTER UPDATE OF Order_status ON "Order"
FOR EACH ROW
EXECUTE FUNCTION bakery_auto_assign_courier();

```

Статистика 1 | Вывод | Лог выполнения

Name	Value
Queries	2
Updated Rows	0
Execute time	0.029s
Fetch time	0.000s
Total time	0.029s
Start time	2025-11-12 22:42:38.063
Finish time	2025-11-12 22:42:38.103

Рисунок 6 – Результат привязки 2 триггерной функции

Задание 1.3.2 Демонстрация (тестирование) работы триггера

```

BEGIN
    IF NEW.Order_status = 'Доставляется' AND OLD.Order_status != 'Доставляется' THEN
        SELECT ID_Employee INTO courier_id
        FROM Employee
        WHERE ID_Position = 4 -- ID должности "Курьер"
        LIMIT 1;
    IF courier_id IS NOT NULL THEN
        INSERT INTO Load (ID_Employee, ID_Order)
        VALUES (courier_id, NEW.ID_Order);
        RAISE NOTICE 'Курьер % назначен на заказ %', courier_id, NEW.ID_Order;
    ELSE
        RAISE NOTICE 'Нет доступных курьеров для заказа %', NEW.ID_Order;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Тест: Изменение статуса заказа на "Доставляется"
UPDATE "Order" SET Order_Status = 'Доставляется' WHERE ID_Order = 6;
-- Проверка назначение курьера
SELECT * FROM Load WHERE ID_Order = 6;

```

load 1 | Статистика 1 | Вывод | Лог выполнения

id_load	id_employee	id_order
1	5	4

Рисунок 8 – Результаты проверки

Задание 1.4.1 Триггер 3: Проверка даты при формировании документа

The screenshot shows the DBeaver interface with the following details:

- Project - General** pane on the left.
- Script Editor** pane in the center containing the SQL code for creating a trigger function:

```
CREATE OR REPLACE FUNCTION bakery_check_document_date()
RETURNS TRIGGER AS $$
BEGIN
    -- Проверяем, что дата не в будущем
    IF NEW.Date_of_formation > CURRENT_DATE THEN
        RAISE EXCEPTION 'Дата документа не может быть в будущем: %', NEW.Date_of_formation;
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

- Statistics** pane at the bottom showing execution results:

Name	Value
Updated Rows	0
Execute time	0.009s
Start time	Wed Nov 12 23:15:41 MSK 2025
Finish time	Wed Nov 12 23:15:41 MSK 2025
Query	CREATE OR REPLACE FUNCTION bakery_check_document_date() RETURNS TRIGGER AS \$\$ BEGIN -- Проверяем, что дата не в будущем IF NEW.Date_of_formation > CURRENT_DATE THEN RAISE EXCEPTION 'Дата документа не может быть в будущем: %', NEW.Date_of_formation; END IF; RETURN NEW; END; \$\$ LANGUAGE plpgsql;

Рисунок 9 – Результат создания 3 триггерной функции

The screenshot shows the DBeaver interface with the following details:

- Project - General** pane on the left.
- Script Editor** pane in the center containing the SQL code for creating a trigger function and binding it to a trigger:

```
CREATE OR REPLACE FUNCTION bakery_check_document_date()
RETURNS TRIGGER AS $$
BEGIN
    -- Проверяем, что дата не в будущем
    IF NEW.Date_of_formation > CURRENT_DATE THEN
        RAISE EXCEPTION 'Дата документа не может быть в будущем: %', NEW.Date_of_formation;
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Создаем триггер
CREATE TRIGGER bakery_trg_document_date
BEFORE INSERT ON Document
FOR EACH ROW
EXECUTE FUNCTION bakery_check_document_date();
```

- Statistics** pane at the bottom showing execution results:

Name	Value
Queries	2
Updated Rows	0
Execute time	0.032s
Fetch time	0.000s
Total time	0.032s
Start time	2025-11-12 23:16:16.051
Finish time	2025-11-12 23:16:16.088

Рисунок 10 – Результат привязки 3 триггерной функции

Задание 1.4.2 Демонстрация (тестирование) работы триггера

```

-- Проверим какие клиенты есть и их заказы
SELECT c.ID_Client, c.Surname, c.Name, COUNT(o.ID_Order) as order_count
FROM Client c
LEFT JOIN "Order" o ON c.ID_Client = o.ID_Client
GROUP BY c.ID_Client, c.Surname, c.Name;

-- ТЕСТ 1: Клиент с заказами (например, ID=1) - ДОЛЖЕН РАБОТАТЬ
INSERT INTO Review (ID_Review, ID_Bakery, ID_Client, Evaluation, Content)
VALUES ((SELECT MAX(ID_Review) + 1 FROM Review), 1, 1, 5, 'Отличный сервис!');

-- ТЕСТ 2: Клиент без заказов (например, если есть ID=5 без заказов) - ДОЛЖНА БЫТЬ ОШИБКА
INSERT INTO Review (ID_Review, ID_Bakery, ID_Client, Evaluation, Content)
VALUES ((SELECT MAX(ID_Review) + 1 FROM Review), 1, 5, 3, 'Не понравилось');

```

	id_client	surname	name	order_count
1	3	Сидоров	Алексей	1
2	5	Смирнов	Дмитрий	1
3	4	Кузнецова	Елена	1
4	6	Тестовый	Клиент	0
5	2	Петрова	Мария	1
6	1	Иванов	Иван	1

Рисунок 11 – Результаты проверки

Задание 2. Использование курсоров

2.2 Явный курсор

```

-- Явный курсор: анализ клиентов
DO $$
DECLARE
    client_cursor CURSOR FOR
        SELECT ID_Client, Surname, Name, Phone_number
        FROM Client;
    client_record RECORD;
BEGIN
    OPEN client_cursor;
    LOOP
        FETCH client_cursor INTO client_record;
        EXIT WHEN NOT FOUND;
        RAISE NOTICE 'Клиент: %, Телефон: %',
                      client_record.Surname || ' ' || client_record.Name,
                      client_record.Phone_number;
    END LOOP;
    CLOSE client_cursor;
END $$;

```

```

Клиент: Петрова Мария, Телефон: +7 (918) 234-56-78
Клиент: Сидоров Алексей, Телефон: +7 (918) 345-67-89
Клиент: Кузнецова Елена, Телефон: +7 (918) 456-78-90
Клиент: Смирнов Дмитрий, Телефон: +7 (920) 567-89-01
Клиент: Тестовый Клиент, Телефон: +7 (999) 999-99-99
Клиент: Иванов Иван, Телефон: +7 (916) 123-45-67
Клиент: Петрова Мария, Телефон: +7 (918) 234-56-78
Клиент: Сидоров Алексей, Телефон: +7 (918) 345-67-89
Клиент: Кузнецова Елена, Телефон: +7 (918) 456-78-90
Клиент: Смирнов Дмитрий, Телефон: +7 (920) 567-89-01
Клиент: Тестовый Клиент, Телефон: +7 (999) 999-99-99

```

Рисунок 12 - Результат запроса

2.3 Неявный курсор

The screenshot shows the DBeaver 25.0.5 interface. In the top navigation bar, the 'File', 'Редактирование' (Edit), 'Навигация' (Navigation), 'Помощь' (Help), 'Редактор SQL' (SQL Editor), 'База данных' (Database), 'Окна' (Windows), and 'Справка' (Help) menus are visible. The 'Auto' connection mode is selected. The 'Bases de datos' (Databases) and 'Проекты' (Projects) tabs are shown. The 'olefirov_gg' database is selected, displaying tables like 'Order', 'address', 'bakery', etc. The 'Script' tab contains a PL/pgSQL script:

```
-- Неявный курсор: анализ товаров
DO $$$
DECLARE
    product_record RECORD;
BEGIN
    FOR product_record IN
        SELECT Title, Price FROM Bakery_product
    LOOP
        RAISE NOTICE 'Товар: %, Цена: % руб.', product_record.Title,
                      product_record.Price;
    END LOOP;
END $$;
```

The 'Output' tab shows the results of the query:

```
Товар: Батон нарезной, Цена: 65 руб.
Товар: Торт Наполеон, Цена: 1200 руб.
Товар: Эклер, Цена: 45 руб.
Товар: Пирог с яйлаками, Цена: 350 руб.
Товар: Печенье овсяное, Цена: 25 руб.
Товар: Круассан, Цена: 75 руб.
Товар: Чизкейк, Цена: 260 руб.
Товар: Багет французский, Цена: 90 руб.
Товар: Бородинский хлеб, Цена: 50 руб.
```

The bottom status bar shows the time as 13:8:315 and the date as 12.11.2025.

Рисунок 13 – Результат запроса