



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 №5

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 у студентов углубленных навыков работы с объектами баз данных в СУБД PostgreSQL, смещаая акцент от прямого манипулирования данными к созданию переиспользуемых логических конструкций.

Постановка задачи:

Задание 1: создание модифицируемого представления.

Задание 2: модификация данных через представление.

1. Добавить новую запись с помощью оператора INSERT.
2. Удалить существующую запись с помощью оператора DELETE.

Задание 3: создание немодифицируемого аналитического представления.

Задание 4: использование аналитического представления в запросах.

Задание 5: создание и обновление материализованного.

1. Создать материализованное представление для ускорения выполнения ресурсоемкого аналитического запроса.
2. Продемонстрировать процесс обновления данных в представлении с помощью команды REFRESH MATERIALIZED VIEW *viewName*.

Задание 6: разработка пользовательской функции для аналитических вычислений.

1. Разработать пользовательскую функцию, которая инкапсулирует комплексный аналитический расчет.
2. Продемонстрировать вызов функции в составе SELECT-запроса.

Задание 7: разработка хранимой процедуры для выполнения сложной операции.

Задание 8: демонстрация вызова хранимой процедуры.

ХОД РАБОТЫ

Начальные данные

	123 ↪ id_employee	123 ↗ id_job_position	A-Z first_name	A-Z second_name	A-Z third_name	A-Z app_account_password_hash	A-Z phone_number
1	1	1	Alice	Johnson	[NULL]	emp_hash_1	555-3333
2	2	2	Bob	Williams	Lee	emp_hash_2	555-4444
3	3	3	Carol	Martinez	[NULL]	emp_hash_3	555-8888
4	4	4	Dave	Anderson	Paul	emp_hash_4	555-9999
5	5	5	Eva	Garcia	Maria	emp_hash_5	555-0000
6	6	2	Frank	Taylor	[NULL]	emp_hash_6	555-1212
7	7	3	Grace	Thomas	Elizabeth	emp_hash_7	555-1313

Рисунок 1 – Содержание таблицы employee

	123 ↪ id_product	A-Z name	A-Z description	123 price
1	1	Margherita Pizza	Classic pizza with tomato and cheese	\$10.99
2	2	Pepperoni Pizza	Pizza with pepperoni slices	\$12.99
3	3	Cola	Refreshing soft drink	\$2.99
4	4	Veggie Supreme	Pizza with assorted vegetables	\$14.99
5	5	Chocolate Cake	Rich chocolate dessert	\$6.99
6	6	Caesar Salad	Fresh salad with Caesar dressing	\$8.99
7	7	Garlic Bread	Toasted bread with garlic butter	\$4.99
8	8	Sprite	Sprite	\$3.49
9	9	Lipton tea	A 1 liter bottle of Lipton tea	\$4.49
10	10	Chocolate muffin	Tasty little chocolate dessert	\$4.99
11	11	Chocolate donut	A chocolate glazed donut	\$5.49

Рисунок 2 – Содержание таблицы product

	123 ↪ id_category	A-Z name	A-Z description
1	1	Pizza	Various types of pizzas
2	2	Beverage	Drinks and beverages
3	3	Appetizer	Starters and snacks
4	4	Dessert	Sweet treats
5	5	Salad	Fresh salads

Рисунок 3 – Содержание таблицы category

	123 ↪ id_product_ingredient	123 ↗ id_product	123 ↗ id_ingredient	123 ingredient_weight
1	1	1	1	500
2	2	1	2	200
3	3	1	3	150
4	4	2	1	500
5	5	2	2	200
6	6	2	3	150
7	7	2	4	100
8	8	4	1	500
9	9	4	2	200
10	10	4	3	150
11	11	4	5	80
12	12	4	6	70
13	13	4	7	60

Рисунок 4 – Содержание таблицы product_ingredient

	123 ↪ id_product_category	123 ↩ id_product	123 ↩ id_category
1	1	1	1
2	2	2	1
3	3	3	2
4	4	4	1
5	5	5	4
6	6	6	5
7	7	7	3
8	8	8	2
9	9	9	2
10	10	10	4
11	11	11	4

Рисунок 5 – Содержание таблицы product_category

	123 ↪ id_ingredient	A-Z name
1	1	Flour
2	2	Cheese
3	3	Tomato Sauce
4	4	Pepperoni
5	5	Mushrooms
6	6	Onions
7	7	Olives

Рисунок 6 – Содержание таблицы ingredient

	123 ↪ id_pizzeria	A-Z phone_number	123 ↩ id_address
1	1	555-0001	1
2	2	555-0002	2
3	3	555-0003	3
4	4	555-0004	4
5	5	555-0005	5

Рисунок 7 – Содержание таблицы pizzeria

	123 ↪ id_pizzeria_storage	123 ↩ id_pizzeria	123 ↩ id_storage
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5

Рисунок 8 – Содержание таблицы pizzeria_storage

	123 ↪ id_storage_ingredient	123 ↗ id_storage	123 ↗ id_ingredient	123 ingredient_weight
1	4	2	4	2 000
2	5	3	5	1 500
3	6	4	6	1 800
4	7	5	7	1 200
5	2	1	2	5 000
6	1	1	1	10 000
7	3	1	3	3 000

Рисунок 9 – Содержание таблицы storage_ingredient

	123 ↪ id_client	A-Z first_name	A-Z second_name	A-Z third_name	A-Z phone_number	123 ↗ id_delivery_address	A-Z account_password_id_hash
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 10 – Содержание таблицы client

	123 ↪ id_address	A-Z city	A-Z street	123 house_number	123 floor	123 unit_number
1	1	New York	Main St	123	[NULL]	[NULL]
2	2	Los Angeles	Oak Ave	456	3	5
3	3	Chicago	Pine Rd	789	2	[NULL]
4	4	Boston	Maple St	321	1	2
5	5	Seattle	Cedar Ave	654	4	8
6	6	Miami	Palm Blvd	987	[NULL]	[NULL]
7	7	Denver	Elm St	147	2	3

Рисунок 11 – Содержание таблицы address

	123 ↪ id_order	123 ↗ id_client	123 ↗ id_delivery_address	123 order_amount	⌚ date_ofFormation	123 ↗ delivery_method_id	123 ↗ status_id
1	4	4	4	\$18.98	2025-09-08	1	1
2	8	1	1	\$38.97	2025-11-21	2	1
3	11	4	4	\$45.97	2025-06-16	1	2
4	12	5	5	\$32.97	2025-06-10	2	1
5	1	1	1	\$25.97	2025-11-05	2	3
6	2	2	2	\$15.98	2025-02-06	1	2
7	3	3	3	\$32.97	2025-09-07	2	3
8	5	5	5	\$45.96	2025-02-09	2	2
9	6	6	6	\$12.99	2025-06-10	1	3
10	7	7	7	\$28.97	2025-09-11	1	4
11	9	2	2	\$22.98	2025-02-07	1	2
12	10	3	3	\$19.98	2025-11-21	2	1

Рисунок 12 – Содержание таблицы _order.

	123 id_order_product	123 id_order	123 id_product	123 product_count
1	1	1	1	2
2	2	1	3	1
3	3	2	2	1
4	4	3	4	2
5	5	3	3	2
6	6	4	6	1
7	7	4	7	2
8	8	5	5	3
9	9	5	3	3
10	10	6	2	1
11	11	7	1	1
12	12	7	4	1
13	13	7	3	2
14	14	8	2	2
15	15	8	3	2
16	16	9	4	1
17	17	9	7	3
18	18	10	6	2
19	19	10	8	1
20	20	11	1	3
21	21	11	3	1
22	22	12	5	2
23	23	12	9	2
24	24	12	11	1

Рисунок 13 – Содержание таблицы order_product.

1 Создание модифицируемого представления.

```

create or replace view chefs as
select *
from employee as e
where
    e.id_job_position = 2;
select * from chefs;

```

The screenshot shows the SQL editor with the above code. Below it, the results of a query on the 'chefs' view are displayed in a table:

id_employee	id_job_position	first_name	second_name	third_name	app_account_password_hash	phone_number
1	2	Bob	Williams	Lee	emp_hash_2	555-4444
2	2	Frank	Taylor	[NULL]	emp_hash_6	555-1212

Рисунок 14

2 Модификация данных через представление.

2.1 Добавить новую запись с помощью оператора INSERT.

```

insert into chefs (id_job_position, first_name, second_name, third_name, app_account_password_hash, phone_number,
                   id_registration_address, employment_date, employment_contract_end_date)
values (2, 'John', 'Silksong', null, 'hash', '123-4567', 1, '2025-01-01', '2025-01-02');

select * from employee where first_name = 'John' and second_name = 'Silksong';

```

The screenshot shows the SQL editor with the above code. Below it, the results of a query on the 'employee' table are displayed in a table:

id_employee	id_job_position	first_name	second_name	third_name	app_account_password_hash	phone_number
1	2	John	Silksong	[NULL]	hash	123-4567

Рисунок 15

2.2 Удалить существующую запись с помощью оператора DELETE.

```
delete from chefs
where first_name = 'John' and second_name = 'Silksong';

select * from employee where first_name = 'John' and second_name = 'Silksong';
```

The screenshot shows a database interface with two tabs: 'employee 1' and 'Статистика 1'. The main area contains the SQL code for deleting a record from the 'chefs' table where 'first_name' is 'John' and 'second_name' is 'Silksong', followed by a select query to verify the deletion. Below the code is a search bar with the placeholder 'Введите SQL выражение чтобы отфильтровать результаты'. At the bottom, there is a table view with columns: id_employee, id_job_position, first_name, second_name, and others. The table is currently empty.

Рисунок 16

3 Создание немодифицируемого аналитического представления.

```
create or replace view category_counts as
select
    c.name,
    count(p_c.id_product)
from
    category as c
join
    product_category as p_c
on
    c.id_category = p_c.id_category
join
    product as p
on
    p_c.id_product = p.id_product
group by
    c.name;

select * from category_counts;
```

The screenshot shows a database interface with two tabs: 'category_counts 1' and 'Статистика 1'. The main area contains the SQL code for creating a view named 'category_counts'. The view selects the name of categories and counts the number of products associated with each category. Below the code is a search bar with the placeholder 'Введите SQL выражение чтобы отфильтровать результаты'. At the bottom, there is a table view with columns: name and count. The table contains the following data:

	name	count
1	Pizza	4
2	Salad	1
3	Appetizer	1
4	Beverage	3
5	Dessert	3

Рисунок 17

4 Использование аналитического представления в запросах.

The screenshot shows a database interface with a query editor and a results viewer. The query is:

```
select
    name, count
from
    category_counts as p_co
where
    p_co.name = 'Pizza'
```

The results table has columns 'name' and 'count'. There is one row with value 'Pizza' and count '4'.

	A-Z name	123 count
1	Pizza	4

Рисунок 18

5 Создание и обновление материализованного представления.

5.1 Создать материализованное представление.

The screenshot shows a database interface with a query editor and a results viewer. The query is:

```
create materialized view pizzas as
select
    p.*, c.name as category_name
from
    product as p
join
    product_category as p_c
on
    p.id_product = p_c.id_product
join
    category as c
on
    p_c.id_category = c.id_category
where
    c.name = 'Pizza';
select * from pizzas;
```

The results table has columns 'id_product', 'name', 'description', 'price', and 'category_name'. There are three rows with values: Margherita Pizza, Pepperoni Pizza, and Veggie Supreme.

Таблица	i23 id_product	A-Z name	A-Z description	i23 price	A-Z category_name
1	1	Margherita Pizza	Classic pizza with tomato and cheese	\$10.99	Pizza
2	2	Pepperoni Pizza	Pizza with pepperoni slices	\$12.99	Pizza
3	4	Veggie Supreme	Pizza with assorted vegetables	\$14.99	Pizza

Рисунок 19

5.2 Продемонстрировать процесс обновления данных в представлении.

	123 id_product	A-Z name	A-Z description	123 price
1	1	Margherita Pizza	Classic pizza with tomato and cheese	\$10.99
2	2	Pepperoni Pizza	Pizza with pepperoni slices	\$12.99
3	3	Cola	Refreshing soft drink	\$2.99
4	4	Veggie Supreme	Pizza with assorted vegetables	\$14.99
5	5	Chocolate Cake	Rich chocolate dessert	\$6.99
6	6	Caesar Salad	Fresh salad with Caesar dressing	\$8.99
7	7	Garlic Bread	Toasted bread with garlic butter	\$4.99
8	8	Sprite	Sprite	\$3.49
9	9	Lipton tea	A 1 liter bottle of Lipton tea	\$4.49
10	10	Chocolate muffin	Tasty little chocolate dessert	\$4.99
11	11	Chocolate donut	A chocolate glazed donut	\$5.49
12	12	Pizza base	If you don't want any topping	\$1.00

Рисунок 20 – Содержание таблицы product после добавления новой пиццы

	123 id_product_category	123 id_product	123 id_category
1	1	1	1
2	2	2	1
3	3	3	2
4	4	4	1
5	5	5	4
6	6	6	5
7	7	7	3
8	8	8	2
9	9	9	2
10	10	10	4
11	11	11	4
12	12	12	1

Рисунок 21 – Содержание таблицы product_category после добавления новой связи пицца-категория

select * from pizzas;						
pizzas 1 × Статистика 1						
select * from pizzas Введите SQL выражение чтобы отфильтровать результаты						
Таблица						
1	123 id_product	A-Z name	A-Z description	123 price	A-Z category_name	
1	1	Margherita Pizza	Classic pizza with tomato and cheese	\$10.99	Pizza	
2	2	Pepperoni Pizza	Pizza with pepperoni slices	\$12.99	Pizza	
3	4	Veggie Supreme	Pizza with assorted vegetables	\$14.99	Pizza	

Рисунок 22 – Материализованное представление до выполнения refresh

refresh materialized view pizzas;						
select * from pizzas;						
pizzas 1 × Статистика 1						
Таблица						
1	123 id_product	A-Z name	A-Z description	123 price	A-Z category_name	
1	1	Margherita Pizza	Classic pizza with tomato and cheese	\$10.99	Pizza	
2	2	Pepperoni Pizza	Pizza with pepperoni slices	\$12.99	Pizza	
3	4	Veggie Supreme	Pizza with assorted vegetables	\$14.99	Pizza	
4	12	Pizza base	If you don't want any topping	\$1.00	Pizza	

Рисунок 23

6 Разработка пользовательской функции для аналитических вычислений.

6.1 Разработать пользовательскую функцию, которая инкапсулирует комплексный аналитический расчет.

```
CREATE OR REPLACE FUNCTION
get_category_count(p_category_name TEXT)
RETURNS TEXT AS $$
DECLARE
    category_exists BOOLEAN;
    r_category_count INTEGER;
BEGIN
    SELECT EXISTS(SELECT 1 FROM category
                 WHERE name = p_category_name) INTO category_exists;

    IF NOT category_exists THEN
        RAISE EXCEPTION 'Category with name % does not exist', p_category_name;
    END IF;

    SELECT COUNT(p_c.id_category)
    INTO r_category_count
    FROM category AS c
    JOIN product_category AS p_c
    ON c.id_category = p_c.id_category
    WHERE c.name = p_category_name
    GROUP BY c.name;

    RETURN r_category_count;
END;
$$ LANGUAGE plpgsql;
```

Рисунок 24

Статистика 1	
Name	Value
Updated Rows	0
Execute time	0.0s
Start time	Mon Nov 03 03:05:00 MSK 2025
Finish time	Mon Nov 03 03:05:00 MSK 2025
Query	CREATE OR REPLACE FUNCTION get_category_count(p_category_name TEXT) RETURNS TEXT AS \$\$ DECLARE category_exists BOOLEAN; r_category_count INTEGER; BEGIN SELECT EXISTS(SELECT 1 FROM category WHERE name = p_category_name) INTO category_exists; IF NOT category_exists THEN RAISE EXCEPTION 'Category with name % does not exist', p_category_name; END IF; SELECT COUNT(p_c.id_category) INTO r_category_count FROM category AS c JOIN product_category AS p_c ON c.id_category = p_c.id_category WHERE c.name = p_category_name GROUP BY c.name; RETURN r_category_count; END; \$\$ LANGUAGE plpgsql;

Рисунок 25

6.2 Продемонстрировать вызов функции в составе SELECT-запроса.

Рисунок 26

7 Разработка хранимой процедуры для выполнения сложной операции.

```
create or replace procedure order_products(
    p_id_pizzeria int,
    p_id_client int,
    p_id_delivery_address int,
    p_id_product int,
    p_quantity_sold int,
    out p_success boolean,
    out p_message text
)
language plpgsql as $$

declare
    current_id_storage int;
    pizzeria_exists boolean;
    client_exists boolean;
    delivery_address_exists boolean;
    product_exists boolean;
    current_stock int;
    current_price numeric;
    total_sale_amount numeric;
    new_id_order int;

begin
```

Рисунок 27 – Процедура, часть 1

```

begin
    select exists(select 1 from pizzeria where id_pizzeria = p_id_pizzeria)
        into pizzeria_exists;
    if not pizzeria_exists then
        p_success := false;
        p_message := 'Ошибка: пиццерия с ID ' || p_id_pizzeria || ' не найдена.';
        return;
    end if;

    select exists(select 1 from client where id_client = p_id_client)
        into client_exists;
    if not client_exists then
        p_success := false;
        p_message := 'Ошибка: покупатель с ID ' || p_id_client || ' не найден.';
        return;
    end if;

    select exists(select 1 from address where id_address = p_id_delivery_address)
        into delivery_address_exists;
    if not delivery_address_exists then
        p_success := false;
        p_message := 'Ошибка: адрес с ID ' || p_id_delivery_address || ' не найден.';
        return;
    end if;

    select price::numeric
    into current_price
    from product where id_product = p_id_product for update;

    select id_storage
    into current_id_storage
    from pizzeria_storage where id_pizzeria = p_id_pizzeria;

    with max_ingredient_quantities as (
        select
            p_i.id_ingredient as id_ingredient,
            (s_i.ingredient_weight / p_i.ingredient_weight)::int as max_quantity
        from product_ingredient as p_i
        join storage_ingredient as s_i on p_i.id_ingredient = s_i.id_ingredient
        where p_i.id_product = p_id_product and s_i.id_storage = current_id_storage
    )
    select min(max_quantity)
    into current_stock
    from max_ingredient_quantities;

```

Рисунок 28 – Процедура, часть 2

```

if current_stock >= p_quantity_sold then
    with required_ingredients as (
        select id_ingredient, ingredient_weight as weight
        from product_ingredient where id_product = p_id_product
    )
    update storage_ingredient
    set ingredient_weight = ingredient_weight - r_i.weight * p_quantity_sold
    from required_ingredients as r_i
    where storage_ingredient.id_storage = current_id_storage and storage_ingredient.id_ingredient = r_i.id_ingredient;

    total_sale_amount := p_quantity_sold * current_price;

    insert into _order (id_client, id_delivery_address, order_amount, date_of_formation, delivery_method_id, status_id)
    values (p_id_client, p_id_delivery_address, total_sale_amount, CURRENT_DATE, 1, 3)
    returning id_order into new_id_order;

    insert into order_product (id_order, id_product, product_count)
    values (new_id_order, p_id_product, p_quantity_sold);

    p_success := true;
    p_message := 'Продано.';
ELSE
    p_success := FALSE;
    p_message := 'Ошибка: недостаточно товара на складе. В наличии: ' ||
    current_stock || ', запрашивается: ' || p_quantity_sold;
END IF;
end;
$$;

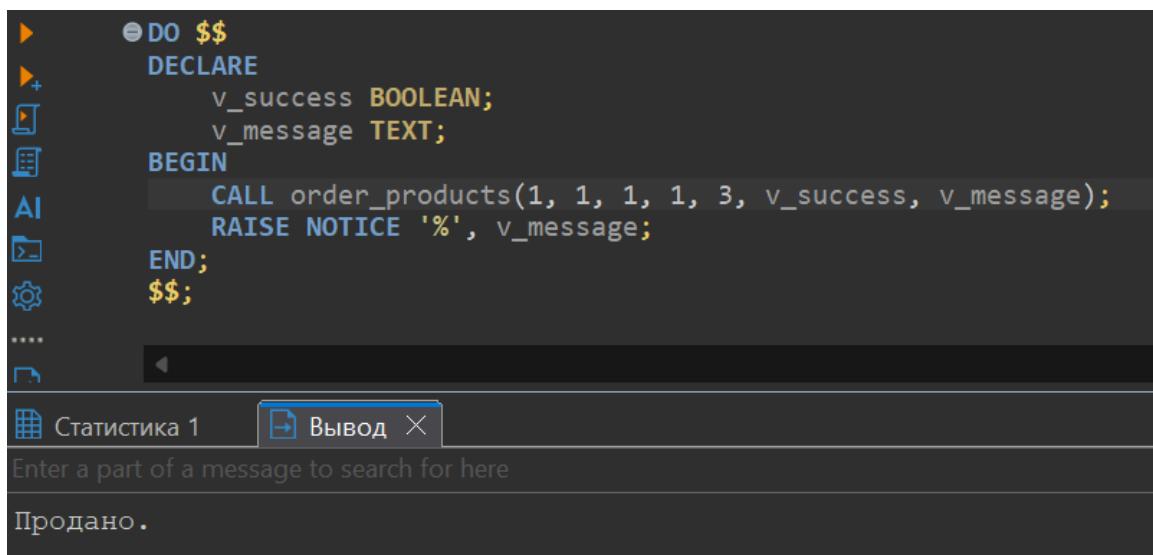
```

Рисунок 29 – Процедура, часть 3

Статистика 1	
Name	Value
Updated Rows	0
Execute time	0.0s
Start time	Mon Nov 03 03:25:06 MSK 2025
Finish time	Mon Nov 03 03:25:06 MSK 2025
Query	<pre>create or replace procedure order_products(p_id_pizzeria int, p_id_client int, p_id_delivery_address int, p_id_product int, p_quantity_sold int, out p_success boolean, out p_message text) language plpgsql as \$\$ declare current_id_storage int; pizzeria_exists boolean; client_exists boolean; delivery_address_exists boolean; product_exists boolean; current_stock int; current_price numeric; total_sale_amount numeric; new_id_order int; begin select exists(select 1 from pizzeria where id_pizzeria = p_id_pizzeria) into pizzeria_exists; if not pizzeria_exists then p_success := false; p_message := 'Ошибка: пиццерия с ID ' p_id_pizzeria ' не найдена.';</pre>

Рисунок 30

8 Демонстрация вызова хранимой процедуры.



The screenshot shows a PostgreSQL query editor interface. The code area contains a DO block that calls the order_products procedure with specific parameters and handles its output through RAISE NOTICE. The message 'Продано.' is visible in the output pane, indicating the procedure executed successfully.

```

DO $$ 
DECLARE
    v_success BOOLEAN;
    v_message TEXT;
BEGIN
    CALL order_products(1, 1, 1, 1, 3, v_success, v_message);
    RAISE NOTICE '%', v_message;
END;
$$;

```

Статистика 1 Вывод

Enter a part of a message to search for here

Продано.

Рисунок 31 – Удачный запрос

The screenshot shows a PostgreSQL command-line interface. In the top pane, a stored procedure is defined:

```
DO $$  
DECLARE  
    v_success BOOLEAN;  
    v_message TEXT;  
BEGIN  
    CALL order_products(1, 1, 1, 1, 1000, v_success, v_message);  
    RAISE NOTICE '%', v_message;  
END;  
$$;
```

In the bottom pane, the output shows an error message:

Ошибка: недостаточно товара на складе. В наличии: 17, запрашивается: 1000

Рисунок 32 – Неудачный запрос

13	15	1	1	\$32.97	2025-11-03	1	3
----	----	---	---	---------	------------	---	---

Рисунок 33 – Новая строка в таблице _order

25	27	15	1	3
----	----	----	---	---

Рисунок 34 – Новая строка в таблице order_product