



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 №4**

Студенты группы

*ИКБО-65-23 Олефиров.Г.Г.*

---

(подпись)

Ассистент

Морозов Д.В.

---

(подпись)

Отчет представлен

«\_\_\_»\_\_\_\_\_ 2025 г.

Москва 2025 г.

## **ПРАКТИЧЕСКАЯ РАБОТА №4.**

### **Условная логика, подзапросы и обобщенные табличные выражения (CTE) в POSTGRES PRO**

**Постановка задачи:** адаптировать каждую из поставленных задач к логической структуре и предметной области вашей базы данных.:

**Задание 1:** Для каждой основной «родительской» сущности в вашей БД (например, производитель, категория товара, автор) определить три наиболее значимых по некоторому числовому признаку дочерних сущности (например, три самых дорогих товара, три самые популярные книги по количеству продаж). В результирующей таблице должны быть указаны идентификатор группы, идентификатор дочерней сущности, её числовой признак и ранг. Для расчёта ранга использовать функцию RANK() или DENSE\_RANK().

**Задание 2:** Для ключевой сущности, имеющей транзакции по времени (например, товар, услуга), рассчитать нарастающий итог (кумулятивную сумму) по некоторому показателю (например, объем продаж, количество заказов) с разбивкой по временным периодам (месяцам или годам). Отчёт должен содержать идентификатор сущности (id/название/...), временной период, сумму за период и кумулятивную сумму.

**Задание 3:** Провести сравнительный анализ общих показателей по периодам. Для каждого периода (например, месяца), начиная со второго, необходимо вывести общий показатель за текущий период и аналогичный показатель за предыдущий период в одной строке. Это позволит наглядно оценить динамику. Необходимо использовать функцию LAG().

**Задание 4:** Создать сводный отчет, который агрегирует некоторый числовой показатель для основной сущности по категориям, представленным в виде столбцов. Например, показать общую сумму продаж для каждого товара по кварталам года. Строки в отчете должны представлять основные сущности, а столбцы — категории.

## Таблицы, используемые для построения запросов:

Order								
	123 ↗ id_order	123 ↗ id_address	123 ↗ id_client	123 ↗ id_discount_card	123 ↗ 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			
	123 ↗ id_address	123 ↗ 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			
	123 ↗ 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						
	123 ↗ id_bakery_product	123 ↗ id_warehouse	123 ↗ id_bakery	▲▼ title	▲▼ 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	id_category_bakery	id_product_category	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	id_client	surname	name	middle_name	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	customer_id	full_name	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	id_discount_card	id_client	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

	123 ↗ id_document	123 ↘ id_order	⌚ date_of_formation	🔤 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

	123 ↗ id_employee	123 ↘ id_position	🔤 surname	🔤 name	🔤 middle_name	🔤 telephone	🔤 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

	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

	123 ↗ order_item_id	123 ↘ order_id	123 ↘ product_id	123 ↗ 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 результат

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 результат

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 результат

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

## Задание №1: Использование ранжирующих функций

The screenshot shows the DBVisualizer interface. On the left, the project tree is visible with several databases and scripts. The main area contains a SQL script titled 'Script-5' which uses the DENSE\_RANK() function to find the top 3 most expensive products for each bakery. Below the script, a table named 'bakery(+1)' displays the results. The table has columns: Пекарня (Bakery), Название продукта (Product Name), Цена (Price), and Ранг (Rank). The results show various products from different bakeries, with their prices and ranks.

Пекарня	Название продукта	Цена	Ранг
пр. Победы, д. 10	Торт Наполеон	1 200	1
пр. Победы, д. 10	Эклеры	45	2
ул. Ленина, д. 25	Бородинский хлеб	85.5	1
ул. Ленина, д. 25	Батон нарезной	65	2
ул. Мира, д. 5	Пирог с яблоками	350	1
ул. Мира, д. 5	Печенье овсяное	25	2
ул. Садовая, д. 15	Чизкейк	280	1

Рисунок 1 - Для каждой пекарни определить 3 самых дорогих продукта

**Объяснение:** Запрос использует DENSE\_RANK() для присвоения ранга продуктам по убыванию цены в пределах каждой пекарни. Это позволяет найти самые дорогие продукты, при этом продукты с одинаковой ценой получат одинаковый ранг.

## Задание №2: Использование агрегатных оконных функций

The screenshot shows the DBBeaver interface with a SQL script in the main window and its execution results below.

```
--> ... Рассчет нарастающего итога заказов по месяцам
WITH MonthlyOrders AS (
    SELECT
        DATE_TRUNC('month', t.Transaction_time)::DATE AS order_month,
        SUM(o.Order_amount) AS monthly_amount
    FROM
        "Order" o
    JOIN
        Transaction t ON o.ID_Order = t.ID_Order
    GROUP BY
        DATE_TRUNC('month', t.Transaction_time)
)
SELECT
    order_month AS "Месяц",
    monthly_amount AS "Сумма за месяц",
    SUM(monthly_amount) OVER (
        ORDER BY order_month
    ) AS "Нарастающий итог"
FROM
    MonthlyOrders
ORDER BY
    order_month;
```

Результат 1

Месяц	Сумма за месяц	Нарастающий итог
2024-01-01	6 000	6 000

Рисунок 2 - Рассчитать нарастающий итог суммы заказов по месяцам

**Объяснение:** Сначала агрегируем заказы по месяцам, затем используем `SUM() OVER()` с `ORDER BY` для расчета кумулятивной суммы. Это показывает, как общая выручка накапливается с течением времени.

### Задание №3: Использование функции смещения

The screenshot shows the DBeaver interface with a SQL script in the center. The script is titled 'MonthlySales' and uses the LAG function to compare monthly sales. The results grid shows one row for January 2024.

```
WITH MonthlySales AS (
    SELECT
        DATE_TRUNC('month', t.Transaction_time)::DATE AS sale_month,
        SUM(o.Order_amount) AS total_amount
    FROM
        "Order" o
    JOIN
        Transaction t ON o.ID_Order = t.ID_Order
    GROUP BY
        DATE_TRUNC('month', t.Transaction_time)
)
SELECT
    sale_month AS "Месяц",
    total_amount AS "Продажи за текущий месяц",
    LAG(total_amount, 1, 0) OVER (ORDER BY sale_month) AS "Продажи за предыдущий месяц"
FROM
    MonthlySales
ORDER BY
    sale_month;
```

Месяц	Продажи за текущий месяц	Продажи за предыдущий месяц
2024-01-01	6 000	0

Рисунок 3 - Сравнить общую сумму заказов за каждый месяц с предыдущим месяцем

**Объяснение:** Функция LAG() позволяет получить значение из предыдущей строки (предыдущего месяца), что дает возможность сравнивать показатели текущего и предыдущего периодов в одной строке.

## Задание №4: Построение сводной таблицы

### Способ 1: Условная агрегация (SUM + CASE)

The screenshot shows the DBeaver interface with a SQL script editor and a results grid.

**SQL Script:**

```
SELECT
    b.Address AS "Пекарня",
    SUM(CASE WHEN EXTRACT(QUARTER FROM t.Transaction_time) = 1 THEN o.Order_amount ELSE 0 END) AS "Q1",
    SUM(CASE WHEN EXTRACT(QUARTER FROM t.Transaction_time) = 2 THEN o.Order_amount ELSE 0 END) AS "Q2",
    SUM(CASE WHEN EXTRACT(QUARTER FROM t.Transaction_time) = 3 THEN o.Order_amount ELSE 0 END) AS "Q3",
    SUM(CASE WHEN EXTRACT(QUARTER FROM t.Transaction_time) = 4 THEN o.Order_amount ELSE 0 END) AS "Q4"
FROM
    "Order" o
JOIN
    Transaction t ON o.ID_Order = t.ID_Order
JOIN
    Bakery_product bp ON bp.ID_Bakery_product = o.ID_Order -- Исправлено соединение
JOIN
    Bakery b ON bp.ID_Bakery = b.ID_Bakery
GROUP BY
    b.Address
ORDER BY
    b.Address;
```

**Results Grid:**

	Пекарня	Q1	Q2	Q3	Q4
1	пр. Победы, д. 10	1 850	0	0	0
2	ул. Ленина, д. 25	2 350	0	0	0
3	ул. Мира, д. 5	1 800	0	0	0

Рисунок 4 - сумма заказов по клиентам по кварталам

### Способ 2: метод с crosstab

The screenshot shows the DBeaver interface with a SQL script editor and a results grid.

**SQL Script:**

```
-- Активация расширения (если еще не активировано)
CREATE EXTENSION IF NOT EXISTS tablefunc;

-- Сводная таблица с использованием crosstab
SELECT * FROM crosstab(
    'SELECT
        c.Surname || '' '' || c.Name AS client_name,
        EXTRACT(QUARTER FROM t.Transaction_time) AS quarter,
        SUM(o.Order_amount) AS total_amount
    FROM
        "Order" o
    JOIN
        Transaction t ON o.ID_Order = t.ID_Order
    JOIN
        Client c ON o.ID_Client = c.ID_Client
    GROUP BY
        c.ID_Client, c.Surname, c.Name, EXTRACT(QUARTER FROM t.Transaction_time)
    ORDER BY 1, 2',
    'SELECT q FROM generate_series(1,4) AS q'
) AS ct("Клиент" TEXT, "Q1" NUMERIC, "Q2" NUMERIC, "Q3" NUMERIC, "Q4" NUMERIC);
```

**Results Grid:**

	Клиент	Q1	Q2	Q3	Q4
1	Иванов Иван	1 500	[NULL]	[NULL]	[NULL]
2	Кузнецова Елена	650	[NULL]	[NULL]	[NULL]
3	Петрова Мария	850	[NULL]	[NULL]	[NULL]
4	Сидоров Алексей	1 200	[NULL]	[NULL]	[NULL]
5	Смирнов Дмитрий	1 800	[NULL]	[NULL]	[NULL]

Рисунок 5 - Результат запроса