



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-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 со сложными аналитическими запросами в СУБД PostgreSQL.

Постановка задачи:

Задание 1: использование ранжирующих функций.

Задание 2: использование агрегатных оконных функций.

Задание 3: использование функции смещения.

Задание 4: построение сводной таблицы.

- 1. С использование условной агрегации,
- 2. С использованием функции crosstab из расширения tablefunc.

ХОД РАБОТЫ

1 Начальные данные

На рисунке 1 представлены данные таблицы product.

	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 1 – Содержание таблицы product

На рисунке 2 представлены данные таблицы category.

	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 2 – Содержание таблицы category

На рисунке 3 представлены данные таблицы product_ingredient.

	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 3 – Содержание таблицы product_ingredient

На рисунке 4 представлены данные таблицы product_category.

	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 4 – Содержание таблицы product_category

На рисунке 5 представлены данные таблицы _order.

	123 ↗ id_order	123 ↗ id_client	123 ↗ id_delivery_address	123 order_amount	⌚ date_of_formation	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 5 – Содержание таблицы _order.

На рисунке 6 представлены данные таблицы order_product.

	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 6 – Содержание таблицы order_product.

2 Ранжирование записей

The screenshot shows a database interface with a code editor and a results viewer.

```

with RankedProducts as (
    select
        p.name as product_name,
        p.price,
        c.name as category_name,
        dense_rank() over (
            partition by c.id_category
            order by p.price desc
        ) as price_rank
    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
)
select
    category_name as "Категория",
    product_name as "Название блюда",
    price as "Цена",
    price_rank as "Ценовой ранг"
from
    RankedProducts
where
    price_rank <= 3
order by
    category_name, price_rank
  
```

The results table below shows the output of the query:

Таблица	A-Z Категория	A-Z Название блюда	123 Цена	123 Ценовой ранг
category(+)				
1	Appetizer	Garlic Bread	\$4.99	1
2	Beverage	Lipton tea	\$4.49	1
3	Beverage	Sprite	\$3.49	2
4	Beverage	Cola	\$2.99	3
5	Dessert	Chocolate Cake	\$6.99	1
6	Dessert	Chocolate donut	\$5.49	2
7	Dessert	Chocolate muffin	\$4.99	3
8	Pizza	Veggie Supreme	\$14.99	1
9	Pizza	Pepperoni Pizza	\$12.99	2
10	Pizza	Margherita Pizza	\$10.99	3
11	Salad	Caesar Salad	\$8.99	1

Рисунок 7 – Выполнение задания 1

3 Агрегатные вычисления в окне: нарастающие итоги

```
with MonthlySales as (
    select
        p.name as product_name,
        date_trunc('month', o.date_of_formation)::date as sale_month,
        sum(o.order_amount) as monthly_amount
    from
        _order as o
    join
        order_product as o_p on o.id_order = o_p.id_order
    join
        product as p on o_p.id_product = p.id_product
    group by
        p.name, sale_month
)
select
    product_name as "Название блюда",
    sale_month as "Месяц продажи",
    monthly_amount as "Сумма за месяц",
    sum(monthly_amount) over (
        partition by product_name
        order by sale_month
    ) as "Нарастающий итог"
from
    MonthlySales
order by
    product_name, sale_month;
```

Рисунок 8 – Запрос для задания 2

O	A-Z Название блюда	⌚ Месяц продажи	123 Сумма за месяц	123 Нарастающий итог
1	Caesar Salad	2025-09-01	\$18.98	\$18.98
2	Caesar Salad	2025-11-01	\$19.98	\$38.96
3	Chocolate Cake	2025-02-01	\$45.96	\$45.96
4	Chocolate Cake	2025-06-01	\$32.97	\$78.93
5	Chocolate donut	2025-06-01	\$32.97	\$32.97
6	Cola	2025-02-01	\$45.96	\$45.96
7	Cola	2025-06-01	\$45.97	\$91.93
8	Cola	2025-09-01	\$61.94	\$153.87
9	Cola	2025-11-01	\$64.94	\$218.81
10	Garlic Bread	2025-02-01	\$22.98	\$22.98
11	Garlic Bread	2025-09-01	\$18.98	\$41.96
12	Lipton tea	2025-06-01	\$32.97	\$32.97
13	Margherita Pizza	2025-06-01	\$45.97	\$45.97
14	Margherita Pizza	2025-09-01	\$28.97	\$74.94
15	Margherita Pizza	2025-11-01	\$25.97	\$100.91
16	Pepperoni Pizza	2025-02-01	\$15.98	\$15.98
17	Pepperoni Pizza	2025-06-01	\$12.99	\$28.97
18	Pepperoni Pizza	2025-11-01	\$38.97	\$67.94
19	Sprite	2025-11-01	\$19.98	\$19.98
20	Veggie Supreme	2025-02-01	\$22.98	\$22.98
21	Veggie Supreme	2025-09-01	\$61.94	\$84.92

Рисунок 9 – Результат выполнения запроса для задания 2

4 Анализ последовательностей: функции смещения LAG и LEAD

```

with TotalMonthlySales as (
    select
        date_trunc('month', o.date_of_formation)::date as sale_month,
        sum(o.order_amount) as total_month_amount
    from
        _order as o
    group by
        sale_month
)
select
    sale_month as "Месяц",
    total_month_amount as "Продажи за месяц",
    lag(total_month_amount::numeric, 1, 0) over (
        order by sale_month
    )::money as "Продажи за предыдущий месяц"
from
    TotalMonthlySales
order by
    sale_month;

```

Результат 1

Таблица	Месяц	Продажи за месяц	Продажи за предыдущий месяц
1	2025-02-01	\$84.92	\$0.00
2	2025-06-01	\$91.93	\$84.92
3	2025-09-01	\$80.92	\$91.93
4	2025-11-01	\$84.92	\$80.92

Рисунок 10 – Выполнение задания 3

5 Построение сводных таблиц

5.1 Условная агрегация

```

select
    p.name as "Название блюда",
    sum(case when extract(quarter from o.date_of_formation) = 1 then o.order_amount else 0::money end) as "Q1",
    sum(case when extract(quarter from o.date_of_formation) = 2 then o.order_amount else 0::money end) as "Q2",
    sum(case when extract(quarter from o.date_of_formation) = 3 then o.order_amount else 0::money end) as "Q3",
    sum(case when extract(quarter from o.date_of_formation) = 4 then o.order_amount else 0::money end) as "Q4"
from
    _order as o
join
    order_product as o_p on o.id_order = o_p.id_order
join
    product as p on o_p.id_product = p.id_product
group by
    p.name
order by
    p.name;

```

product 1

Таблица	Название блюда	Q1	Q2	Q3	Q4
1	Caesar Salad	\$0.00	\$0.00	\$18.98	\$19.98
2	Chocolate Cake	\$45.96	\$32.97	\$0.00	\$0.00
3	Chocolate donut	\$0.00	\$32.97	\$0.00	\$0.00
4	Cola	\$45.96	\$45.97	\$61.94	\$64.94
5	Garlic Bread	\$22.98	\$0.00	\$18.98	\$0.00
6	Lipton tea	\$0.00	\$32.97	\$0.00	\$0.00
7	Margherita Pizza	\$0.00	\$45.97	\$28.97	\$25.97
8	Pepperoni Pizza	\$15.98	\$12.99	\$0.00	\$38.97
9	Sprite	\$0.00	\$0.00	\$0.00	\$19.98
10	Veggie Supreme	\$22.98	\$0.00	\$61.94	\$0.00

Рисунок 11 – Выполнение задания 4, способ 1

5.2 Метод с crosstab из расширения tablefunc

The screenshot shows the pgAdmin interface with a query editor and a results grid.

SQL Editor:

```
select * from crosstab(
    'select
        p.name,
        extract(quarter from o.date_of_formation),
        sum(o.order_amount)
    from _order as o
    join order_product as o_p on o.id_order = o_p.id_order
    join product as p on o_p.id_product = p.id_product
    group by p.name, extract(quarter from o.date_of_formation)
    order by 1, 2',
    'select q from generate_series(1, 4) as q'
) as ct("Название лекарства" TEXT, "Q1" MONEY, "Q2" MONEY, "Q3" MONEY, "Q4" MONEY);
```

Results Grid:

Название лекарства	Q1	Q2	Q3	Q4
	Caesar Salad	[NULL]	[NULL]	\$18.98
Chocolate Cake	\$45.96	\$32.97	[NULL]	[NULL]
Chocolate donut	[NULL]	\$32.97	[NULL]	[NULL]
Cola	\$45.96	\$45.97	\$61.94	\$64.94
Garlic Bread	\$22.98	[NULL]	\$18.98	[NULL]
Lipton tea	[NULL]	\$32.97	[NULL]	[NULL]
Margherita Pizza	[NULL]	\$45.97	\$28.97	\$25.97
Pepperoni Pizza	\$15.98	\$12.99	[NULL]	\$38.97
Sprite	[NULL]	[NULL]	[NULL]	\$19.98
Veggie Supreme	\$22.98	[NULL]	\$61.94	[NULL]

Рисунок 12 – Выполнение задания 4, способ 2