



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 №4
по дисциплине «Разработка баз данных»

Студент группы

ИКБО-50-23. Павлов Н.С..

(подпись)

Преподаватель

Мажей Я. В.

(подпись)

Москва 2025 г.

СОДЕРЖАНИЕ

1. ПОСТАНОВКА ЗАДАЧИ	3
2 ВЫПОЛНЕНИЕ РАБОТЫ	5
2.1 ИСХОДНЫЕ ТАБЛИЦЫ	5
2.2 ЗАДАНИЕ 1	7
2.3 ЗАДАНИЕ 2	8
2.4 ЗАДАНИЕ 3	9

1. ПОСТАНОВКА ЗАДАЧИ

Цель: Целью данной практической работы является формирование у студентов углубленных навыков работы со сложными аналитическими запросами в СУБД PostgreSQL.

Задачи:

Задание №1: использование ранжирующих функций

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: использование агрегатных оконных функций

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: использование функции смещения

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: построение сводной таблицы

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.

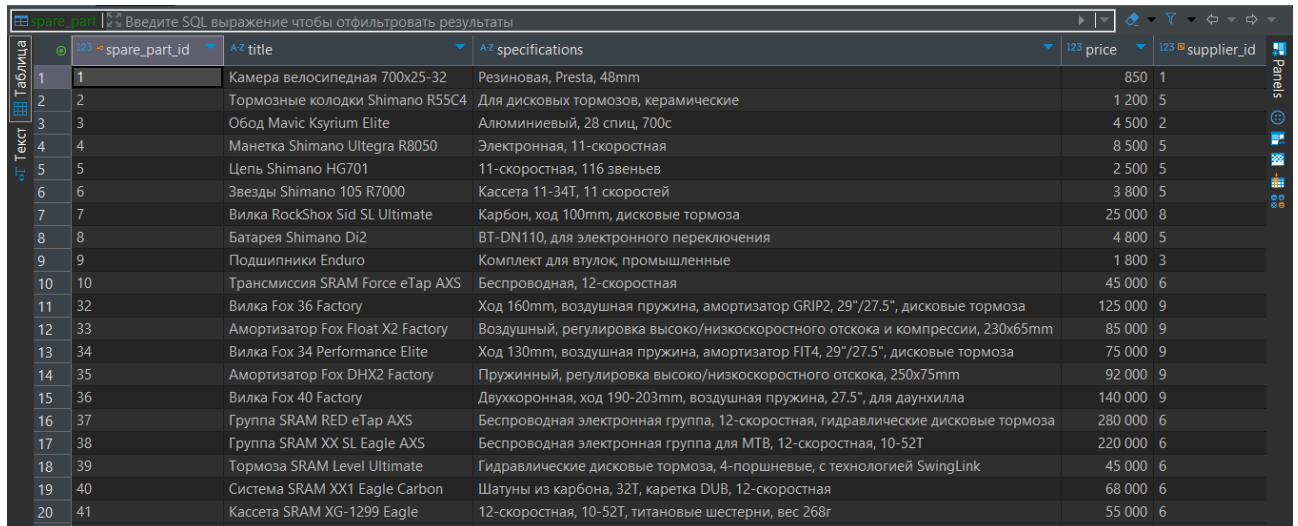
Строки в отчете должны представлять основные сущности, а столбцы — категории. Задачу необходимо решить двумя способами:

1. С использованием условной агрегации (комбинация SUM и CASE).
2. С использованием функции crosstab из расширения tablefunc.

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

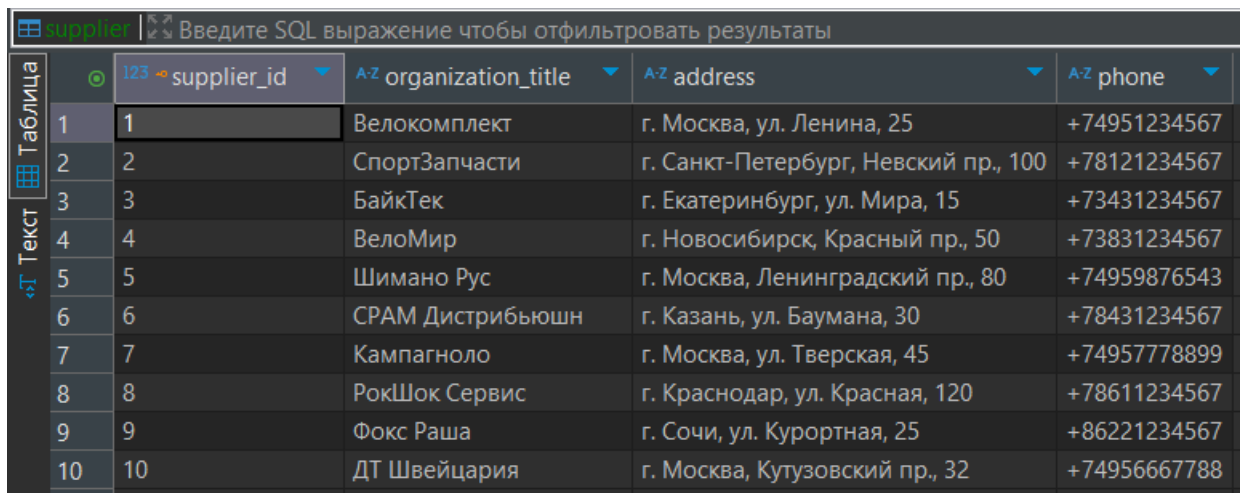
2 ВЫПОЛНЕНИЕ РАБОТЫ

2.1 ИСХОДНЫЕ ТАБЛИЦЫ



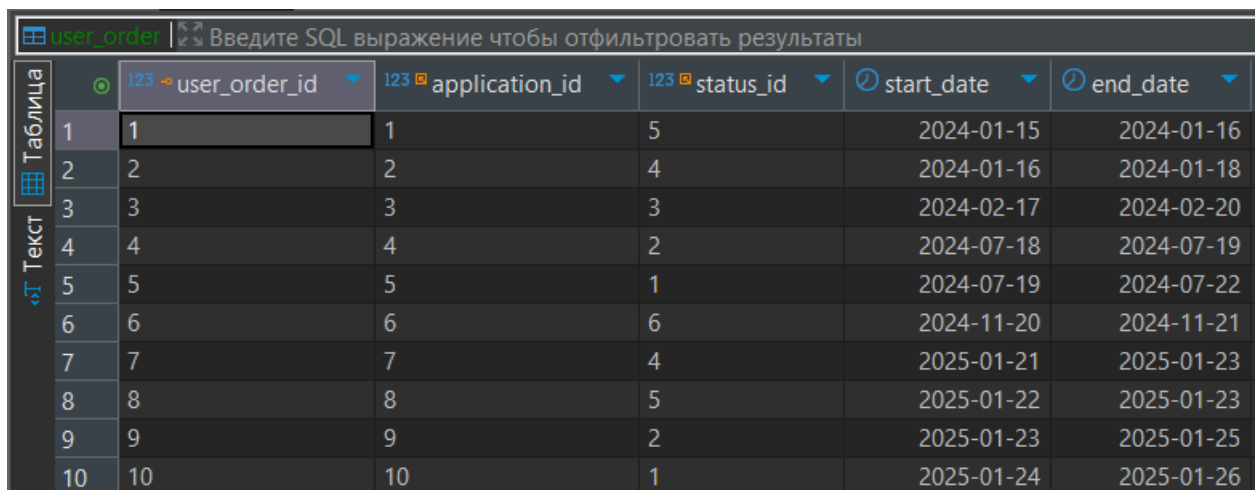
	123 spare_part_id	A-Z title	A-Z specifications	123 price	123 supplier_id
1	1	Камера велосипедная 700x25-32	Резиновая, Presta, 48mm	850	1
2	2	Тормозные колодки Shimano R55C4	Для дисковых тормозов, керамические	1 200	5
3	3	Обод Mavic Ksyrium Elite	Алюминиевый, 28 спиц, 700c	4 500	2
4	4	Манетка Shimano Ultegra R8050	Электронная, 11-скоростная	8 500	5
5	5	Цепь Shimano HG701	11-скоростная, 116 звеньев	2 500	5
6	6	Звезды Shimano 105 R7000	Кассета 11-34T, 11 скоростей	3 800	5
7	7	Вилка RockShox Sid SL Ultimate	Карбон, ход 100mm, дисковые тормоза	25 000	8
8	8	Батарея Shimano Di2	BT-DN110, для электронного переключения	4 800	5
9	9	Подшипники Enduro	Комплект для втулок, промышленные	1 800	3
10	10	Трансмиссия SRAM Force eTap AXS	Беспроводная, 12-скоростная	45 000	6
11	32	Вилка Fox 36 Factory	Ход 160mm, воздушная пружина, амортизатор GRIP2, 29"/27.5", дисковые тормоза	125 000	9
12	33	Амортизатор Fox Float X2 Factory	Воздушный, регулировка высоко/низкоскоростного отскока и компрессии, 230x65mm	85 000	9
13	34	Вилка Fox 34 Performance Elite	Ход 130mm, воздушная пружина, амортизатор FIT4, 29"/27.5", дисковые тормоза	75 000	9
14	35	Амортизатор Fox DHX2 Factory	Пружинный, регулировка высоко/низкоскоростного отскока, 250x75mm	92 000	9
15	36	Вилка Fox 40 Factory	Двухкоронная, ход 190-203mm, воздушная пружина, 27.5", для даунхилла	140 000	9
16	37	Группа SRAM RED eTap AXS	Беспроводная электронная группа, 12-скоростная, гидравлические дисковые тормоза	280 000	6
17	38	Группа SRAM XX SL Eagle AXS	Беспроводная электронная группа для MTB, 12-скоростная, 10-52T	220 000	6
18	39	Тормоза SRAM Level Ultimate	Гидравлические дисковые тормоза, 4-поршневые, с технологией SwingLink	45 000	6
19	40	Система SRAM XX1 Eagle Carbon	Шатуны из карбона, 32T, каретка DUB, 12-скоростная	68 000	6
20	41	Кассета SRAM XG-1299 Eagle	12-скоростная, 10-52T, титановые шестерни, вес 268g	55 000	6

Рисунок 1 – Таблица spare_part



	123 supplier_id	A-Z organization_title	A-Z address	A-Z phone
1	1	Велокомплект	г. Москва, ул. Ленина, 25	+74951234567
2	2	СпортЗапчасти	г. Санкт-Петербург, Невский пр., 100	+78121234567
3	3	БайкТек	г. Екатеринбург, ул. Мира, 15	+73431234567
4	4	ВелоМир	г. Новосибирск, Красный пр., 50	+73831234567
5	5	Шимано Рус	г. Москва, Ленинградский пр., 80	+74959876543
6	6	СРАМ Дистрибьюшн	г. Казань, ул. Баумана, 30	+78431234567
7	7	Кампагноло	г. Москва, ул. Тверская, 45	+74957778899
8	8	РокШок Сервис	г. Краснодар, ул. Красная, 120	+78611234567
9	9	Фокс Раша	г. Сочи, ул. Курортная, 25	+86221234567
10	10	ДТ Швейцария	г. Москва, Кутузовский пр., 32	+74956667788

Рисунок 2 – Таблица supplier



	123 user_order_id	123 application_id	123 status_id	start_date	end_date
1	1	1	5	2024-01-15	2024-01-16
2	2	2	4	2024-01-16	2024-01-18
3	3	3	3	2024-02-17	2024-02-20
4	4	4	2	2024-07-18	2024-07-19
5	5	5	1	2024-07-19	2024-07-22
6	6	6	6	2024-11-20	2024-11-21
7	7	7	4	2025-01-21	2025-01-23
8	8	8	5	2025-01-22	2025-01-23
9	9	9	2	2025-01-23	2025-01-25
10	10	10	1	2025-01-24	2025-01-26

Рисунок 3 – Таблица user_order

estimate

Введите SQL выражение чтобы отфильтровать результаты

	<div>123 estimate_id</div>	<div>123 user_order_id</div>	<div>123 price_of_work</div>	<div>123 price_of_spare_parts</div>	<div>123 total_price</div>
1	1	1	500	850	1 350
2	2	2	800	1 200	2 000
3	3	3	1 500	4 500	6 000
4	4	4	1 000	8 500	9 500
5	5	5	2 000	2 500	4 500
6	6	6	1 200	3 800	5 000
7	7	7	3 000	25 000	28 000
8	8	8	1 000	4 800	5 800
9	9	9	1 500	1 800	3 300
10	10	10	4 000	45 000	49 000

Рисунок 4 – Таблица estimate

2.2 ЗАДАНИЕ 1

The screenshot shows a database client interface with a SQL script editor and a results table.

SQL Script:

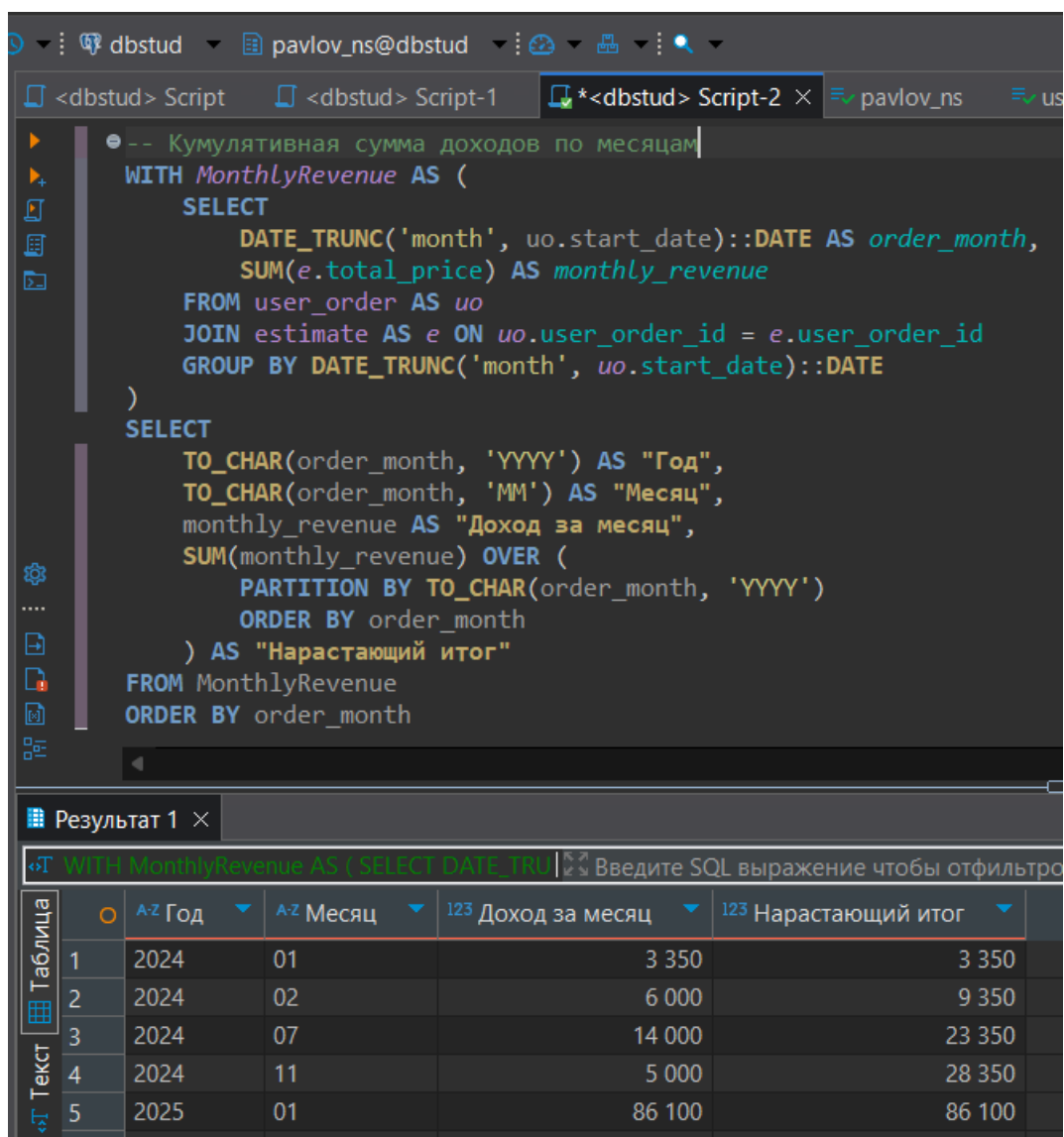
```
-- Ранжирование запчастей по цене внутри каждого поставщика
WITH RankedSpareParts AS (
    SELECT
        s.organization_title AS supplier_name,
        sp.title AS spare_part_name,
        sp.price,
        DENSE_RANK() OVER (
            PARTITION BY sp.supplier_id
            ORDER BY sp.price DESC
        ) AS price_rank
    FROM spare_part AS sp
    JOIN supplier AS s ON sp.supplier_id = s.supplier_id
)
SELECT
    supplier_name AS "Поставщик",
    spare_part_name AS "Название запчасти",
    price AS "Цена",
    price_rank AS "Ранг по цене"
FROM RankedSpareParts
WHERE price_rank <= 3
ORDER BY supplier_name, price_rank
```

Results Table:

	A-Z Поставщик	A-Z Название запчасти	Цена	Ранг по цене
1	БайкТек	Подшипники Enduro	1 800	1
2	Велокомплект	Камера велосипедная 700x25-32	850	1
3	РокШок Сервис	Вилка RockShox Sid SL Ultimate	25 000	1
4	СпортЗапчасти	Обод Mavic Ksyrium Elite	4 500	1
5	СРАМ Дистрибьюшн	Группа SRAM RED eTap AXS	280 000	1
6	СРАМ Дистрибьюшн	Группа SRAM XX SL Eagle AXS	220 000	2
7	СРАМ Дистрибьюшн	Система SRAM XX1 Eagle Carbon	68 000	3
8	Фокс Раша	Вилка Fox 40 Factory	140 000	1
9	Фокс Раша	Вилка Fox 36 Factory	125 000	2
10	Фокс Раша	Амортизатор Fox DHX2 Factory	92 000	3
11	Шимано Рус	Манетка Shimano Ultegra R8050	8 500	1
12	Шимано Рус	Батарея Shimano Di2	4 800	2
13	Шимано Рус	Звезды Shimano 105 R7000	3 800	3

Рисунок 5 – Запрос с ранжирующей функцией DENSE_RANK()

2.3 ЗАДАНИЕ 2



Скриншот интерфейса базы данных, демонстрирующий SQL-запрос с агрегатной функцией нарастания.

SQL-запрос:

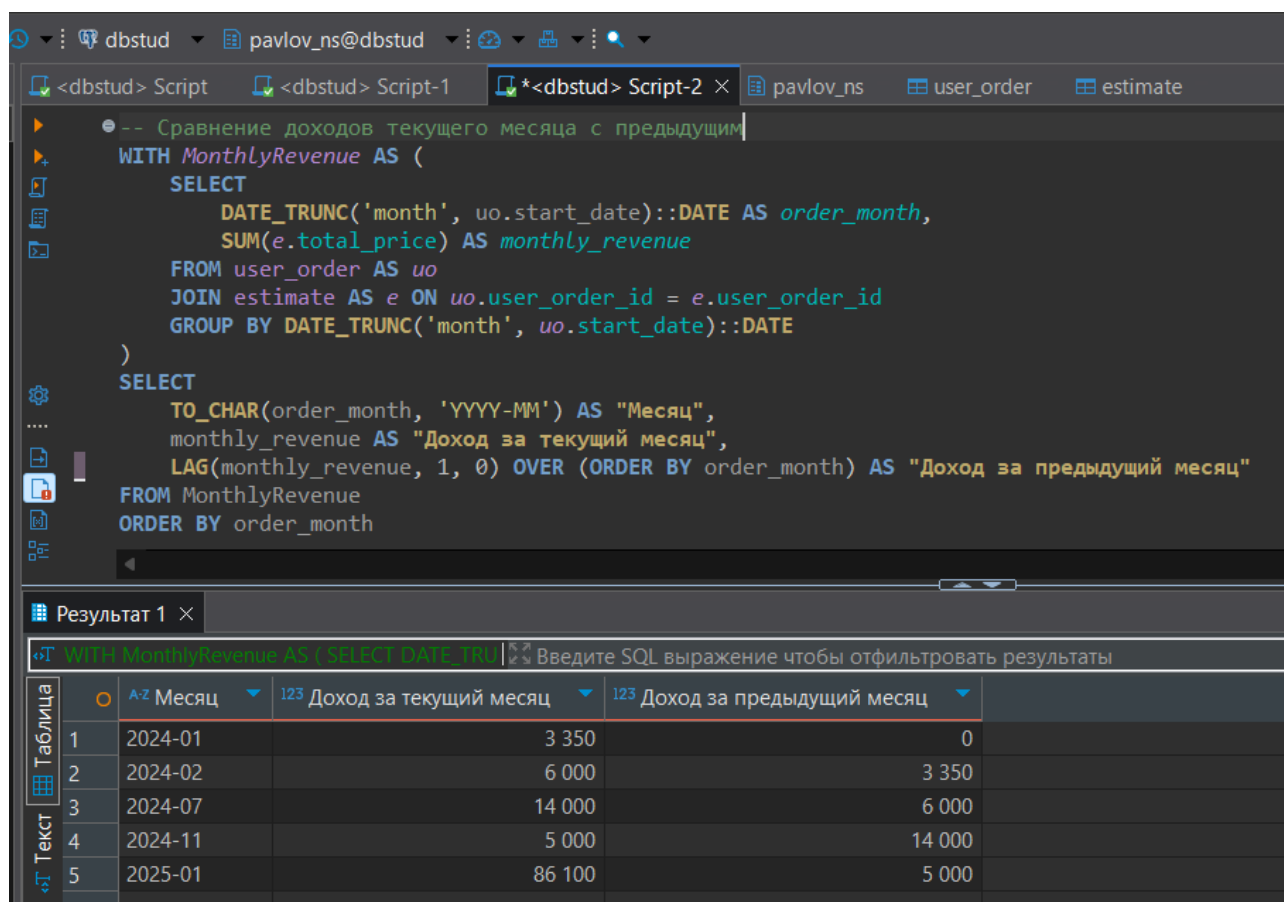
```
-- Кумулятивная сумма доходов по месяцам
WITH MonthlyRevenue AS (
    SELECT
        DATE_TRUNC('month', uo.start_date)::DATE AS order_month,
        SUM(e.total_price) AS monthly_revenue
    FROM user_order AS uo
    JOIN estimate AS e ON uo.user_order_id = e.user_order_id
    GROUP BY DATE_TRUNC('month', uo.start_date)::DATE
)
SELECT
    TO_CHAR(order_month, 'YYYY') AS "Год",
    TO_CHAR(order_month, 'MM') AS "Месяц",
    monthly_revenue AS "Доход за месяц",
    SUM(monthly_revenue) OVER (
        PARTITION BY TO_CHAR(order_month, 'YYYY')
        ORDER BY order_month
    ) AS "Нарастающий итог"
FROM MonthlyRevenue
ORDER BY order_month
```

Результат 1:

	Год	Месяц	Доход за месяц	Нарастающий итог
1	2024	01	3 350	3 350
2	2024	02	6 000	9 350
3	2024	07	14 000	23 350
4	2024	11	5 000	28 350
5	2025	01	86 100	86 100

Рисунок 6 – Запрос с агрегатной функцией нарастания

2.4 ЗАДАНИЕ 3



```
-- Сравнение доходов текущего месяца с предыдущим
WITH MonthlyRevenue AS (
  SELECT
    DATE_TRUNC('month', uo.start_date)::DATE AS order_month,
    SUM(e.total_price) AS monthly_revenue
  FROM user_order AS uo
  JOIN estimate AS e ON uo.user_order_id = e.user_order_id
  GROUP BY DATE_TRUNC('month', uo.start_date)::DATE
)
SELECT
  TO_CHAR(order_month, 'YYYY-MM') AS "Месяц",
  monthly_revenue AS "Доход за текущий месяц",
  LAG(monthly_revenue, 1, 0) OVER (ORDER BY order_month) AS "Доход за предыдущий месяц"
FROM MonthlyRevenue
ORDER BY order_month
```

Результат 1 ×

WITH MonthlyRevenue AS (SELECT DATE_TRUNC('month', uo.start_date)::DATE AS order_month, SUM(e.total_price) AS monthly_revenue FROM user_order AS uo JOIN estimate AS e ON uo.user_order_id = e.user_order_id GROUP BY DATE_TRUNC('month', uo.start_date)::DATE) SELECT TO_CHAR(order_month, 'YYYY-MM') AS "Месяц", monthly_revenue AS "Доход за текущий месяц", LAG(monthly_revenue, 1, 0) OVER (ORDER BY order_month) AS "Доход за предыдущий месяц" FROM MonthlyRevenue ORDER BY order_month

	Месяц	Доход за текущий месяц	Доход за предыдущий месяц
1	2024-01	3 350	0
2	2024-02	6 000	3 350
3	2024-07	14 000	6 000
4	2024-11	5 000	14 000
5	2025-01	86 100	5 000

Рисунок 7 – Запрос с использованием функции смещения

2.5 ЗАДАНИЕ 4

dbstud pavlov_ns@dbstud <dbstud> Script-2 x pavlov_ns user_order estimate

```
-- Анализ запчастей по поставщикам и ценовым категориям
SELECT
    s.organization_title AS "Поставщик",
    COUNT(CASE WHEN sp.price < 1000 THEN 1 END) AS "До 1000 руб",
    COUNT(CASE WHEN sp.price BETWEEN 1000 AND 5000 THEN 1 END) AS "1000-5000 руб",
    COUNT(CASE WHEN sp.price BETWEEN 5001 AND 20000 THEN 1 END) AS "5001-20000 руб",
    COUNT(CASE WHEN sp.price BETWEEN 20001 AND 50000 THEN 1 END) AS "20001-50000 руб",
    COUNT(CASE WHEN sp.price > 50000 THEN 1 END) AS "Свыше 50000 руб"
FROM supplier AS s
JOIN spare_part AS sp ON s.supplier_id = sp.supplier_id
GROUP BY s.supplier_id
ORDER BY 1
```

supplier 1 x

SELECT organization_title AS "Поставщик" Введите SQL выражение чтобы отфильтровать результаты

	Поставщик	До 1000 руб	1000-5000 руб	5001-20000 руб	20001-50000 руб	Свыше 50000 руб
1	БайкТек	0	1	0	0	0
2	Велокомплект	1	0	0	0	0
3	РокШок Сервис	0	0	0	1	0
4	СпортЗапчасти	0	1	0	0	0
5	СРАМ Дистрибьюшн	0	0	0	2	4
6	Фокс Раша	0	0	0	0	5
7	Шимано Рус	0	4	1	0	0

Рисунок 8 – Сводная таблица с использованием условной агрегации

dbstud pavlov_ns@dbstud <dbstud> Script-2 x pavlov_ns user_order estimate

```
-- Анализ запчастей по поставщикам и ценовым категориям
SELECT * FROM crosstab(
    'SELECT
        s.organization_title AS supplier_name,
        CASE
            WHEN sp.price < 1000 THEN ''До 1000 руб''
            WHEN sp.price BETWEEN 1000 AND 5000 THEN ''1000-5000 руб''
            WHEN sp.price BETWEEN 5001 AND 20000 THEN ''5001-20000 руб''
            WHEN sp.price BETWEEN 20001 AND 50000 THEN ''20001-50000 руб''
            ELSE ''Свыше 50000 руб''
        END AS price_category,
        COUNT(*) AS part_count
    FROM supplier AS s
    JOIN spare_part AS sp ON s.supplier_id = sp.supplier_id
    GROUP BY s.organization_title,
        CASE
            WHEN sp.price < 1000 THEN ''До 1000 руб''
            WHEN sp.price BETWEEN 1000 AND 5000 THEN ''1000-5000 руб''
            WHEN sp.price BETWEEN 5001 AND 20000 THEN ''5001-20000 руб''
            WHEN sp.price BETWEEN 20001 AND 50000 THEN ''20001-50000 руб''
            ELSE ''Свыше 50000 руб''
        END
    ORDER BY 1, 2',
    $$VALUES ('До 1000 руб'), ('1000-5000 руб'), ('5001-20000 руб'), ('20001-50000 руб'), ('Свыше 50000 руб')$$
) AS ct("Поставщик" TEXT, "До 1000 руб" INT, "1000-5000 руб" INT, "5001-20000 руб" INT, "20001-50000 руб" INT, "Свыше 50000 руб" INT)
```

Результат 1 x

SELECT * FROM crosstab(Введите SQL выражение чтобы отфильтровать результаты

	Поставщик	До 1000 руб	1000-5000 руб	5001-20000 руб	20001-50000 руб	Свыше 50000 руб
1	БайкТек	[NULL]	1	[NULL]	[NULL]	[NULL]
2	Велокомплект	1	[NULL]	[NULL]	[NULL]	[NULL]
3	РокШок Сервис	[NULL]	[NULL]	[NULL]	1	[NULL]
4	СпортЗапчасти	[NULL]	1	[NULL]	[NULL]	[NULL]
5	СРАМ Дистрибьюшн	[NULL]	[NULL]	[NULL]	2	4
6	Фокс Раша	[NULL]	[NULL]	[NULL]	[NULL]	5
7	Шимано Рус	[NULL]	4	1	[NULL]	[NULL]

Рисунок 9 – Сводная таблица с использованием функции crosstab