



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 *ИНБО-12-23. Албахтин И.В.*

(подпись)

Ассистент *Брайловский А.В.*

(подпись)

Москва 2025 г.

ПРАКТИЧЕСКАЯ РАБОТА №4.

АНАЛИТИЧЕСКИЕ ЗАПРОСЫ: ОКОННЫЕ ФУНКЦИИ И ПОСТРОЕНИЕ СВОДНЫХ ТАБЛИЦ

Цель:

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

Постановка задачи:

Для выполнения практической работы необходимо последовательно выполнить четыре задачи, используя собственную базу данных. Все примеры в данном документе основаны на демонстрационной базе данных «Аптека», содержащей таблицы `manufacturers` (производители), `medicines` (лекарства) и `sales` (продажи).

Ваша задача — адаптировать каждую из поставленных задач к логической структуре и предметной области вашей базы данных. Приведенные ниже формулировки и последующие примеры кода служат шаблоном для понимания, какой тип аналитического запроса требуется составить.

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 **результатирующей 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.

ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОЙ РАБОТЫ

Таблица 1. Таблица worker (Сотрудник)

The screenshot shows a database management interface. At the top, there are tabs for 'albakhtin_iv', '*<dbstud> Script-9', 'review_id', 'maintenance', and 'mair'. The main editor displays a complex SQL query using a CROSSTAB function to generate a pivot table of invoice data. Below this, a second query 'SELECT * FROM worker LIMIT 10;' is highlighted. At the bottom, a table view for 'worker 1' is shown, displaying the first two rows of data.

```
SELECT * FROM crosstab(  
    'SELECT  
        EXTRACT(YEAR FROM m.start_date) AS year,  
        EXTRACT(QUARTER FROM m.start_date) AS quarter  
        SUM(i.total_amount)  
    FROM invoice i  
    JOIN maintenance m ON i.maintenance_id = m.mainte  
    GROUP BY year, quarter  
    ORDER BY 1, 2',  
    'SELECT q FROM generate_series(1,4) AS q'  
) AS ct(year NUMERIC, Q1 NUMERIC, Q2 NUMERIC, Q3 NUMERIC)
```

```
SELECT * FROM worker LIMIT 10;
```

worker 1

SELECT * FROM worker LIMIT 10 | Введите SQL выражение чтобы отфильтровать результаты

Таблица	123 worker_id	A-Z name	A-Z position	A-Z phone
1	1	Сергей	Механик	+79112223344
2	2	Ольга	Диагност	+79210009988

Таблица 2. Таблица maintenance (ТО)

albaktin_iv

*<dbstud> Script-9 X

review_id

maintenance

maintenance_work

maintenance

GROUP BY year, quarter

ORDER BY 1, 2',

'SELECT q FROM generate_series(1,4) AS q'

) AS ct(year NUMERIC, Q1 NUMERIC, Q2 NUMERIC, Q3 NUMERIC, Q4 NUMERIC);

SELECT * FROM worker LIMIT 10;

SELECT maintenance_id, car_id, worker_id, start_date, end_date, status

FROM maintenance

LIMIT 10;

-- 3. Таблица счетов

SELECT invoice_id maintenance_id total amount payment status

maintenance 1 X

SELECT maintenance_id, car_id, worker_id, start_date Введите SQL выражение чтобы отфильтровать результаты

	123 maintenance_id	123 car_id	123 worker_id	start_date	end_date	A-Z status
1	34	4	4	2025-02-10	[NULL]	waiting
2	35	5	5	2025-03-20	2025-09-10	completed
3	36	6	2	2025-03-20	[NULL]	in progress
4	37	7	3	2025-04-25	2025-09-11	completed
5	38	8	1	2025-04-25	[NULL]	planned
6	39	9	4	2025-05-15	2025-09-15	completed
7	40	10	5	2025-05-15	[NULL]	in progress
8	33	3	2	2025-11-05	2025-11-15	completed
9	31	1	1	2025-07-15	2025-07-20	completed
10	32	2	3	2025-09-10	2025-09-20	in progress

Таблица 3. Таблица invoice (счета)

<div> <div> <div>albakhtin_iv</div> <div>*<dbstud> Script-9 X</div> <div>review_id</div> <div>maintenance</div> <div>maintenance_work</div> <div>ma</div> </div> <div> <div>SELECT * FROM worker LIMIT 10;</div> <div>SELECT maintenance_id, car_id, worker_id, start_date, end_date, stat FROM maintenance LIMIT 10;</div> <div>SELECT invoice_id, maintenance_id, total_amount, payment_status FROM invoice LIMIT 10;</div> </div> </div>					
<div> <div>invoice 1 X</div> <div> <div>SELECT invoice_id, maintenance_id, total_amount, p</div> <div>Введите SQL выражение чтобы отфильтровать результаты</div> </div> </div>					
<div> <div>Таблица</div> <div>Текст</div> <div>Таблица</div> <div>Текст</div> <div>Таблица</div> <div>Текст</div> <div>Таблица</div> <div>Текст</div> <div>Таблица</div> <div>Текст</div> <div>Таблица</div> <div>Текст</div> </div>	<div> <div>123 invoice_id</div> <div>123 maintenance_id</div> <div>123 total_amount</div> <div>A-Z payment_status</div> </div>				
	1	3	31	35 000	Оплачено
	2	4	32	18 000	Ожидает оплаты
	3	5	33	26 000	Оплачено
	4	6	34	40 000	Оплачено
	5	7	35	15 000	Оплачено
	6	8	36	17 000	Оплачено
	7	9	37	22 000	Ожидает оплаты
	8	10	38	19 500	Оплачено
	9	11	39	31 000	Оплачено
	10	12	40	28 000	Оплачено

Задание 1. Использование ранжирующих функций

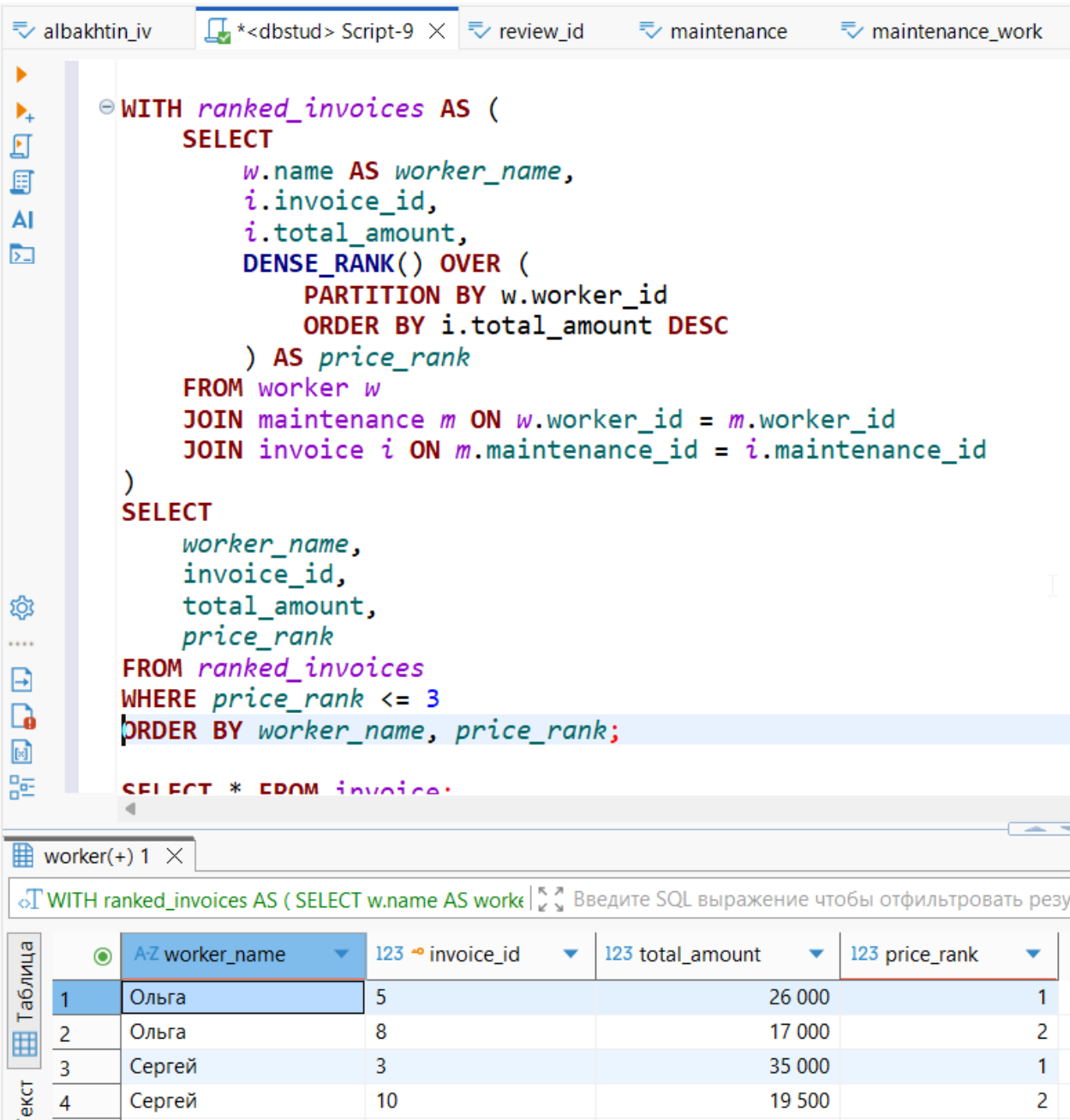


Рисунок 1 – Три самых дорогих ТО для каждого работника по сумме счёта

Задание 2. Использование агрегатных оконных функций

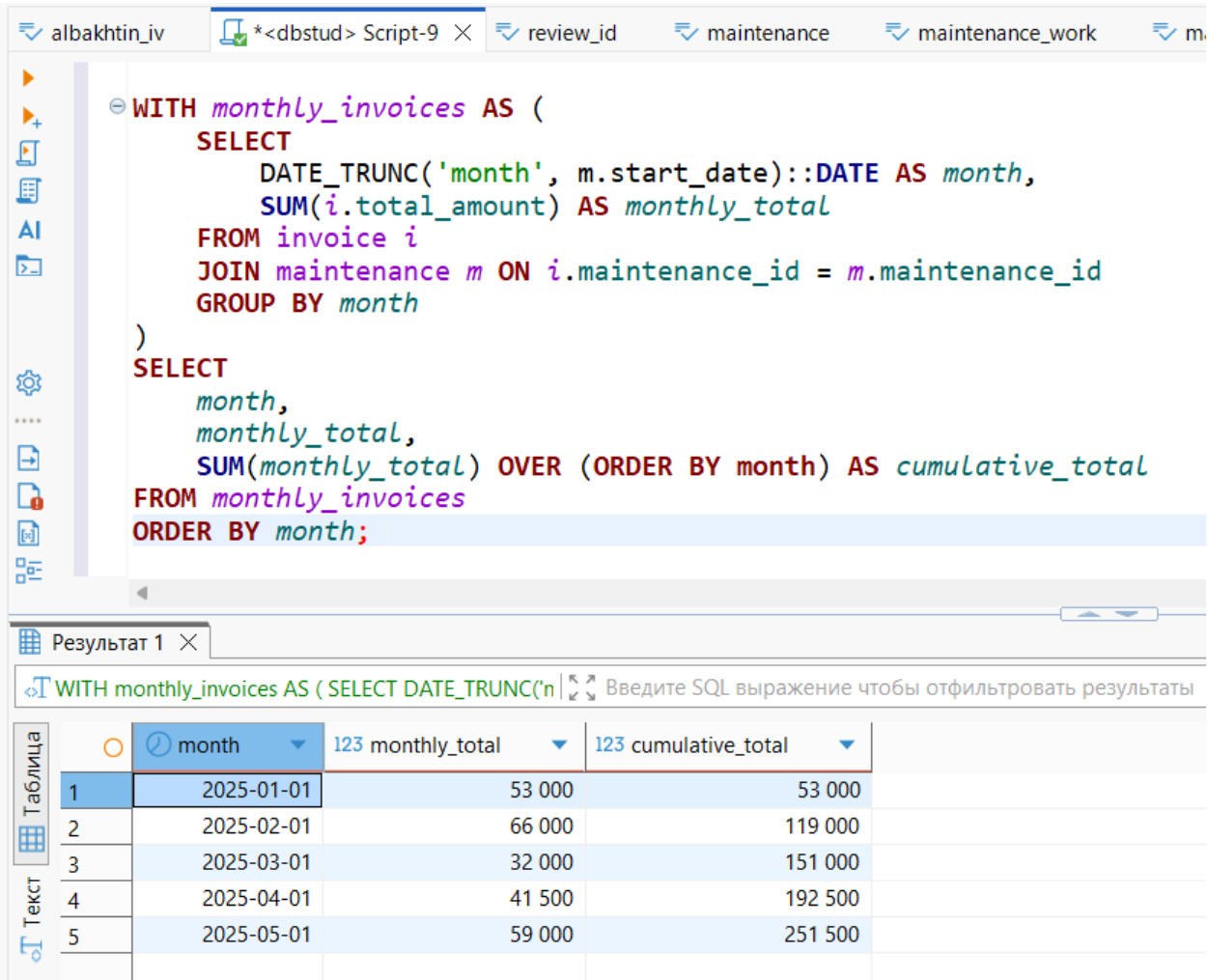
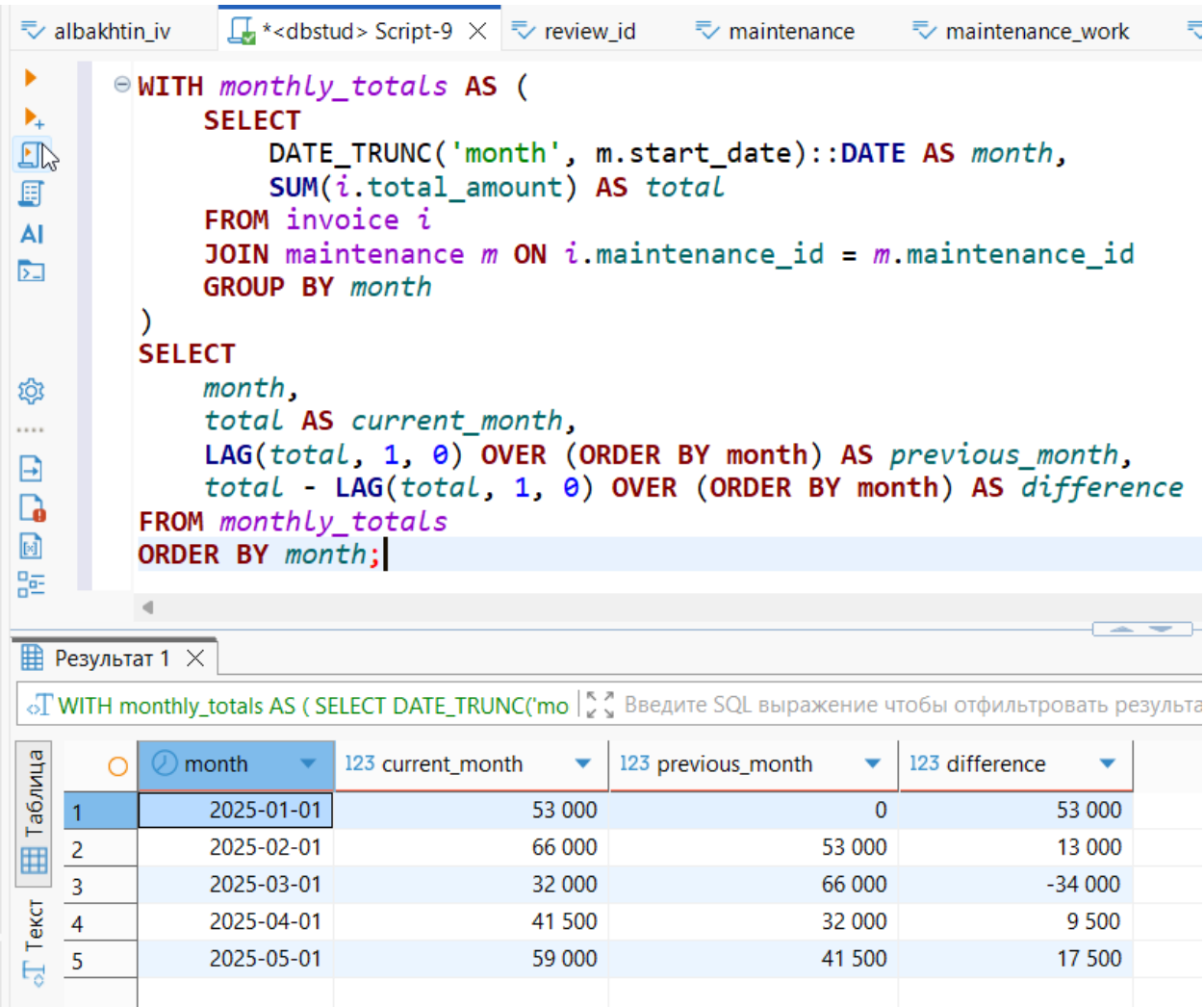


Рисунок 2 – Изменения суммы ТО по месяцам

Задание 3. Функции смещения (LAG)



**Рисунок 3 – Динамика роста или упадка по сравнению с предыдущим
месяцем**

Задание 4. Построение сводной таблицы

The screenshot shows a SQL query in a script editor. The query filters for maintenance_id = 32 and calculates the sum of total_amount for each quarter of the year 2025. The results are displayed in a table with columns for the year and quarters 1 through 4.

```
end_date = '2025-09-20'
WHERE maintenance_id = 32;

SELECT
    EXTRACT(YEAR FROM m.start_date) AS year,
    SUM(CASE WHEN EXTRACT(QUARTER FROM m.start_date) = 1 THEN i.total_amount ELSE 0 END) AS Q1,
    SUM(CASE WHEN EXTRACT(QUARTER FROM m.start_date) = 2 THEN i.total_amount ELSE 0 END) AS Q2,
    SUM(CASE WHEN EXTRACT(QUARTER FROM m.start_date) = 3 THEN i.total_amount ELSE 0 END) AS Q3,
    SUM(CASE WHEN EXTRACT(QUARTER FROM m.start_date) = 4 THEN i.total_amount ELSE 0 END) AS Q4
FROM invoice i
JOIN maintenance m ON i.maintenance_id = m.maintenance_id
GROUP BY year
ORDER BY year;
```

Результат 1

SELECT EXTRACT(YEAR FROM m.start_date) AS year, | Введите SQL выражение чтобы отфильтровать результаты

	123 year	123 q1	123 q2	123 q3	123 q4
1	2 025	72 000	100 500	53 000	26 000

Рисунок 4 - Показать сумму счетов по кварталам (вариант 1)

The screenshot shows a SQL query that creates a table function crosstab and uses it to generate a pivot table of maintenance amounts by year and quarter. The results are displayed in a table with columns for the year and quarters 1 through 4.

```
CREATE EXTENSION IF NOT EXISTS tablefunc;

SELECT * FROM crosstab(
    'SELECT
        EXTRACT(YEAR FROM m.start_date) AS year,
        EXTRACT(QUARTER FROM m.start_date) AS quarter,
        SUM(i.total_amount)
    FROM invoice i
    JOIN maintenance m ON i.maintenance_id = m.maintenance_id
    GROUP BY year, quarter
    ORDER BY 1, 2',
    'SELECT q FROM generate_series(1,4) AS q'
) AS ct(year NUMERIC, Q1 NUMERIC, Q2 NUMERIC, Q3 NUMERIC, Q4 NUMERIC);
```

Результат 1

SELECT * FROM crosstab('SELECT EXTRACT(YEAR FR | Введите SQL выражение чтобы отфильтровать результаты

	123 year	123 q1	123 q2	123 q3	123 q4
1	2 025	72 000	100 500	53 000	26 000

Рисунок 5 – Показать сумму счетов по кварталам (вариант 2)

ВЫВОД

В ходе выполнения задания была построена сводная таблица, отражающая распределение сумм счетов по кварталам за 2025 год. Для расчёта использовались агрегирующие выражения с условием CASE и функция crosstab из расширения tablefunc. Результаты показали, что операции в 1 и 2 квартале были активными, а в 3 и 4 кварталах данных пока нет. Таким образом, были освоены методы группировки данных по временным периодам и построение сводных таблиц в PostgreSQL.