



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-50-23 Враженко Д.О.*

.

(подпись)

Преподаватель *Мажей Я.В.*

(подпись)

Отчет представлен «___»_____2025 г.

Москва 2025 г.

Цель работы:

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 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: использование свободной 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 **сопроводить комментарием**, объясняющим его назначение и логику работы с учетом специфики вашей базы данных.

ХОД ВЫПОЛНЕНИЯ РАБОТЫ

Все используемые таблицы индивидуальной схемы данных:

Таблица 1. Таблица employee (Сотрудник)

	123 id_en	123 id_pc	AZ last_name	AZ first_name	AZ phone_nun	AZ registration_address	employment_date	contract_end_date
1	1	1	Петров	Иван	9123456789	ул. Ленина, 10	2020-01-15	2025-01-15
2	2	1	Сидорова	Мария	9234567890	пр. Мира, 5	2019-05-20	2024-05-20
3	3	1	Иванов	Алексей	9345678901	ул. Советская, 3	2021-03-10	2026-03-10
4	4	2	Васильев	Дмитрий	9456789012	ул. Гагарина, 7	2022-02-01	2023-02-01
5	5	3	Кузнецова	Ольга	9567890123	пр. Победы, 12	2021-07-15	2024-07-15
6	6	4	Смирнов	Андрей	9678901234	ул. Лесная, 9	2023-01-10	2023-12-31
7	7	2	Орлов	Михаил	9789012345	ул. Центральная, 15	2022-03-01	2025-03-01
8	8	3	Волкова	Екатерина	9890123456	пр. Солнечный, 8	2021-11-10	2024-11-10
9	9	4	Никитин	Артем	9901234567	ул. Весенняя, 22	2023-02-15	2024-02-15
10	10	5	Семенова	Анна	9012345678	ул. Осенняя, 11	2020-08-20	2025-08-20
11	11	6	Козлов	Денис	9123456780	пр. Гагарина, 33	2023-04-10	2024-04-10

Таблица 2. Таблица position (Должность)

	123 id_position	AZ position_name	AZ access_category	123 salary
1	1	Менеджер	Администрация	50 000
2	2	Повар	Кухня	35 000
3	3	Кассир	Обслуживание	30 000
4	4	Уборщик	Обслуживание	25 000
5	5	Старший менеджер	Администрация	70 000
6	6	Шеф-повар	Кухня	60 000
7	7	Старший кассир	Обслуживание	40 000

Таблица 3. Таблица employee_inventory (Связующая таблица Сотрудник-Инвентарь)

	123 id_employee_inventory	123 id_employee	123 id_inventory
1	1	4	1
2	2	5	2
3	3	4	3
4	4	6	4
5	5	1	5
6	6	2	6

Таблица 4. Таблица inventory (Инвентарь)

	123 id_inventory	AZ item_name	AZ category	AZ status
1	1	Нож поварской	Кухонное оборудование	В использовании
2	2	Кассовый аппарат	Кассовое оборудование	На складе
3	3	Фартук	Спецодежда	В использовании
4	4	Стол кухонный	Мебель	На складе
5	5	Компьютер	Офисная техника	В использовании
6	6	Принтер	Офисная техника	На складе

Таблица 5. Таблица *termination_request* (Запрос на увольнение)

	123 id_termination_req	123 id_deptm	123 id_employe	request_date	AZ reason	AZ status	termination_date
1	1	3	6	2023-11-01	Окончание контракта	Завершено	2023-12-31
2	2	2	5	2023-10-15	Переход на другую работу	В процессе	2024-01-01
3	3	1	4	2023-09-15	Несоответствие должности	Завершено	2023-09-30
4	4	2	8	2023-08-10	Нарушение трудовой дисциплины	Завершено	2023-08-25
5	5	3	3	2023-07-05	Смена места жительства	Завершено	2023-07-20
6	6	1	7	2023-06-20	Окончание контракта	Завершено	2023-06-30
7	7	2	9	2023-05-15	По соглашению сторон	Завершено	2023-05-31
8	8	3	10	2023-04-10	Сокращение штата	Завершено	2023-04-25
9	9	1	11	2023-03-05	Неудовлетворенность зарплатой	Завершено	2023-03-20

Таблица 6. Таблица *department* (Отдел)

	123 id_department	AZ department_name	AZ department_head
1	1	Кухня	Иван Петров
2	2	Обслуживание	Мария Сидорова
3	3	Администрация	Алексей Иванов

Таблица 7. Таблица *payment* (Выплата)

	123 id_payment	123 id_termination_request	123 amount	payment_date	AZ comment
1	1	1	25 000	2023-12-31	Окончательный расчет по увольнению
2	2	3	35 000	2023-09-30	Расчет при увольнении
3	3	4	28 000	2023-08-25	Окончательный расчет
4	4	5	45 000	2023-07-20	Компенсационные выплаты
5	5	6	32 000	2023-06-30	Расчет по увольнению
6	6	7	38 000	2023-05-31	Выплаты при увольнении
7	7	8	42 000	2023-04-25	Компенсация за сокращение
8	8	9	29 000	2023-03-20	Окончательный расчет

Таблица 8. Таблица *document* (Архив документов)

	123 id_document	AZ document_type	creation_date	123 id_termination_request	AZ content
1	1	Заявление на увольнение	2023-10-30	1	Заявление на увольнение по собственн
2	2	Приказ об увольнении	2023-12-31	1	Приказ №123 об увольнении Смирнова
3	3	Заявление на увольнение	2023-09-10	3	Заявление от Васильева Д.
4	4	Приказ об увольнении	2023-09-30	3	Приказ №124 об увольнении Васильева

Таблица 9. Таблица *request_termination_type* (Связующая таблица Запрос на увольнение-Вид увольнения)

	123 id_request_termination_type	123 id_termination_request	123 id_termination_type
1	1	1	1
2	2	2	3
3	3	3	2
4	4	4	3
5	5	5	1
6	6	6	1
7	7	7	2
8	8	8	2
9	9	9	1

Таблица 10. Таблица *termination_type* (Вид увольнения)

	123 id_termination_type	AZ type_name	AZ description
1	1	По собственному желанию	Увольнение по инициативе сотрудника
2	2	Сокращение штата	Увольнение в связи с сокращением численности персонала
3	3	Нарушение дисциплины	Увольнение за нарушение трудовой дисциплины

Задание №1: использование ранжирующих функций

SQL Query:

```
WITH ranked_employees AS (  
  SELECT  
    d.id_department,  
    d.department_name,  
    e.id_employee,  
    e.last_name || ' ' || e.first_name AS employee_name,  
    p.salary,  
    DENSE_RANK() OVER (PARTITION BY d.id_department ORDER BY p.salary DESC) as salary_rank  
  FROM department d  
  JOIN employee e ON EXISTS (  
    SELECT 1 FROM termination_request tr  
    WHERE tr.id_department = d.id_department AND tr.id_employee = e.id_employee  
  )  
  JOIN position p ON e.id_position = p.id_position  
)  
SELECT  
  id_department as "ID отдела",  
  department_name as "Название отдела",  
  id_employee as "ID сотрудника",  
  employee_name as "ФИО сотрудника",  
  salary as "Зарплата",  
  salary_rank as "Ранг"  
FROM ranked_employees  
WHERE salary_rank <= 3  
ORDER BY id_department, salary_rank;
```

ID отдела	Название отдела	ID сотрудника	ФИО сотрудника	Зарплата	Ранг
1	Кухня	11	Козлов Денис	60 000	1
2	Кухня	4	Васильев Дмитрий	35 000	2
3	Кухня	7	Орлов Михаил	35 000	2
4	Обслуживание	8	Волкова Екатерина	30 000	1
5	Обслуживание	5	Кунецова Ольга	30 000	1
6	Обслуживание	9	Никитин Артем	25 000	2
7	Администрация	10	Семенова Анна	70 000	1
8	Администрация	3	Иванов Алексей	50 000	2
9	Администрация	6	Смирнов Андрей	25 000	3

Рисунок 1 – Ранжирующие функции

Задание №2: использование агрегатных оконных функций

SQL Query:

```
WITH monthly_payments AS (  
  SELECT  
    d.id_department,  
    d.department_name as "Отдел",  
    DATE_TRUNC('month', p.payment_date) as "Период",  
    TO_CHAR(DATE_TRUNC('month', p.payment_date), 'YYYY-MM') as "Период_формат",  
    SUM(p.amount) as "Сумма_за_месяц",  
    SUM(SUM(p.amount)) OVER (  
      PARTITION BY d.id_department  
      ORDER BY DATE_TRUNC('month', p.payment_date)  
      ROWS UNBOUNDED PRECEDING  
    ) as "Нарастающий_итог"  
  FROM payment p  
  JOIN termination_request tr ON p.id_termination_request = tr.id_termination_request  
  JOIN department d ON tr.id_department = d.id_department  
  WHERE tr.status = 'Запущено'  
  GROUP BY d.id_department, d.department_name, DATE_TRUNC('month', p.payment_date)  
)  
SELECT  
  "Отдел",  
  "Период_формат" as "Период",  
  "Сумма_за_месяц" as "Выплаты за месяц",  
  "Нарастающий_итог" as "Нарастающий итог"  
FROM monthly_payments  
ORDER BY "Отдел", "Период";
```

Отдел	Период	Выплаты за месяц	Нарастающий итог
Администрация	2023-04	42 000	42 000
Администрация	2023-07	45 000	87 000
Администрация	2023-12	25 000	112 000
Кухня	2023-03	29 000	29 000
Кухня	2023-06	32 000	61 000
Кухня	2023-09	35 000	96 000
Обслуживание	2023-05	38 000	38 000
Обслуживание	2023-08	28 000	66 000

Рисунок 2 – Агрегатные оконные функции

Задание №3: использование функции смещения

WITH monthly_totals AS (
 SELECT
 DATE_TRUNC('month', payment_date) as payment_month,
 TO_CHAR(DATE_TRUNC('month', payment_date), 'YYYY-MM') as "Месяц",
 SUM(amount) as "Выплаты текущий месяц",
 COUNT(*) as "Количество увольнений"
 FROM payment
 GROUP BY DATE_TRUNC('month', payment_date)
)
SELECT
 "Месяц",
 "Выплаты текущий месяц" as "Сумма выплат",
 "Количество увольнений" as "Кол-во увольнений",
 LAG("Выплаты текущий месяц") OVER (ORDER BY payment_month) as "Выплаты предыдущий месяц",
 LAG("Количество увольнений") OVER (ORDER BY payment_month) as "Увольнения предыдущий месяц",
 "Выплаты текущий месяц" - LAG("Выплаты текущий месяц") OVER (ORDER BY payment_month) as "Разница суммы"
FROM monthly_totals
ORDER BY payment_month;

	Месяц	123 Сумма выплат	123 Кол-во увольнений	123 Выплаты предыдущий месяц	123 Увольнения предыдущий месяц	123 Разница суммы
1	2023-03	29 000	1	(NULL)	(NULL)	
2	2023-04	42 000	1	29 000	1	13 000
3	2023-05	38 000	1	42 000	1	-4 000
4	2023-06	32 000	1	38 000	1	-6 000
5	2023-07	45 000	1	32 000	1	13 000
6	2023-08	28 000	1	45 000	1	-17 000
7	2023-09	35 000	1	28 000	1	7 000
8	2023-12	25 000	1	35 000	1	-10 000

Рисунок 3 – Функция смещения

Задание №4: построение сводной таблицы

SELECT
 d.department_name as "Отдел",
 SUM(CASE WHEN EXTRACT(QUARTER FROM p.payment_date) = 1 THEN p.amount ELSE 0 END) as "Квартал 1",
 SUM(CASE WHEN EXTRACT(QUARTER FROM p.payment_date) = 2 THEN p.amount ELSE 0 END) as "Квартал 2",
 SUM(CASE WHEN EXTRACT(QUARTER FROM p.payment_date) = 3 THEN p.amount ELSE 0 END) as "Квартал 3",
 SUM(CASE WHEN EXTRACT(QUARTER FROM p.payment_date) = 4 THEN p.amount ELSE 0 END) as "Квартал 4",
 SUM(p.amount) as "Итого"
FROM payment p
JOIN termination_request tr ON p.id_termination_request = tr.id_termination_request
JOIN department d ON tr.id_department = d.id_department
GROUP BY d.department_name
ORDER BY d.department_name;

	Отдел	123 Квартал 1	123 Квартал 2	123 Квартал 3	123 Квартал 4	123 Итого
1	Администрация	0	42 000	45 000	25 000	112 000
2	Кухня	29 000	32 000	35 000	0	96 000
3	Обслуживание	0	38 000	28 000	0	66 000

Рисунок 4 – SUM и CASE

DBEaver 25.2.1 - <dbstud> Задачи

Файл Редактирование Навигация Поиск Редактор SQL База данных Око Справка

SQL Commit Rollback T Auto dbstud vrazhenko.do@dbstud

CREATE EXTENSION IF NOT EXISTS tablefunc;

```
SELECT *
FROM crosstab(
    'SELECT
        d.department_name,
        EXTRACT(QUARTER FROM p.payment_date) as quarter,
        SUM(p.amount) as total_amount
    FROM payment p
    JOIN termination_request tr ON p.id_termination_request = tr.id_termination_request
    JOIN department d ON tr.id_department = d.id_department
    GROUP BY d.department_name, EXTRACT(QUARTER FROM p.payment_date)
    ORDER BY 1, 2',
    'VALUES (1), (2), (3), (4)')
AS ct(
    department_name text,
    "Квартал 1" numeric,
    "Квартал 2" numeric,
    "Квартал 3" numeric,
    "Квартал 4" numeric
);
```

Результат 1

Введите SQL выражение чтобы отфильтровать результаты

	A2 department_name	123 Квартал 1	123 Квартал 2	123 Квартал 3	123 Квартал 4
1	Администрация	[NULL]	42 000	45 000	25 000
2	Кухня	29 000	32 000	35 000	[NULL]
3	Обслуживание	[NULL]	38 000	28 000	[NULL]

Обновить Save Cancel Экспорт данных ... 200 3 3 строк получено - 0.016с, 2025-10-13 в 00:32:49

MSK ru Запись Инт. вставка 21:3:654 Set: 0 | 0 Сохранить

vrazhenko...
Название: vrazhenko...
Комментарий:
Таблицы
Внешние таблицы
Представления
Мат. представления
Индексы
Функции
Последовательности
Типы данных
Агрегатные функции
Права доступа
Исходный код

Рисунок 5 – crosstab