

Примеры использования СУБД PostgreSQL на курсах по анализу данных

- Readme SQL.txt
- Readme SQL_Копия.txt
- SQL_1_Топ-3 аптеки_PostgreSQL.sql
- SQL_2_Топ-3 лекарства_PostgreSQL.sql
- SQL_3_Аптеки от 1.8 млн оборота_PostgreSQL.sql
- SQL_4_Накопленная сумма продаж по каждой аптеке_PostgreSQL.sql
- SQL_4a_Накопленная сумма продаж по каждой аптеке_PostgreSQL.sql
- SQL_5_Количество клиентов в аптеках_PostgreSQL.sql
- SQL_6_Лучшие клиенты_PostgreSQL.sql
- SQL_7_Накопленная сумма по клиентам_PostgreSQL.sql
- SQL_8_Самые частые клиенты аптек Горздрав и Здравсити_PostgreSQL.sql
- SQL_9_Сравнение динамики продаж между Москвой и Санкт-Петербургом по аптекам_1ый вариант_PostgreSQL.sql
- SQL_9_Сравнение динамики продаж между Москвой и Санкт-Петербургом по аптекам_2-ой вариант_PostgreSQL.sql
- SQL_10_Лекарства от насморка_PostgreSQL.sql
- SQL_11_Кто наши клиенты_PostgreSQL.sql

```
/*  
    1. Запрос "Топ-3 аптеки"  
    а. Вывести топ 3 аптеки по объему продаж (GROUP BY, LIMIT)  
*/  
SELECT  
    pharmacy_name,  
    SUM(price * count) AS total_sales  
FROM  
    pharma_orders  
GROUP BY  
    pharmacy_name  
ORDER BY  
    total_sales DESC  
LIMIT 3;
```

```
/*  
    2. Запрос "Топ-3 лекарства"  
    а. Вывести топ 3 лекарства по объему продаж  
*/  
SELECT  
    drug,  
    SUM(price * count) AS total_sales  
FROM  
    pharma_orders  
GROUP BY  
    drug  
ORDER BY  
    total_sales DESC  
LIMIT 3;
```

```
/*
    3. Запрос "Аптеки от 1.8 млн оборота"
    а. Вывести аптеки, имеющие более 1.8 млн оборота (HAVING)
*/
SELECT
    pharmacy_name,
    SUM(price * count) AS total_revenue
FROM
    pharma_orders
GROUP BY
    pharmacy_name
HAVING
    SUM(price * count) > 1800000;

|
```

```
/*  
    4. Запрос "Накопленная сумма продаж по каждой аптеке" (OVER)  
*/  
SELECT  
    pharmacy_name,  
    order_id,  
    drug,  
    price,  
    count,  
    city,  
    report_date,  
    SUM(price * count) OVER (PARTITION BY pharmacy_name ORDER BY report_date) AS cumulative_sales  
FROM  
    pharma_orders;
```

```
/*  
    4. Запрос "Накопленная сумма продаж по каждой аптеке" (OVER)  
*/  
SELECT  
    pharmacy_name,  
    MIN(report_date) AS start_date,  
    MAX(report_date) AS end_date,  
    SUM(price * count) AS cumulative_sales  
FROM  
    pharma_orders  
GROUP BY  
    pharmacy_name;
```

/*

5. Запрос "Количество клиентов в аптеках"

а. Соединить таблицы заказов и клиентов (JOIN)

б. Посчитать кол-во уникальных клиентов на каждую аптеку (DISTINCT)

с. Отсортировать аптеки по убыванию кол-ва клиентов (ORDER BY)

*/

SELECT

o.pharmacy_name,

COUNT(DISTINCT o.customer_id) AS unique_client_count

FROM

pharma_orders o

JOIN

customers c ON o.customer_id = c.customer_id

GROUP BY

o.pharmacy_name

ORDER BY

unique_client_count DESC;

```
/*
```

```
    6. Запрос "Лучшие клиенты"
```

```
    a. Соединить таблицы заказов и клиентов (JOIN)
```

```
    b. Посчитать total сумму заказов для каждого клиента
```

```
    c. Проранжировать клиентов по убыванию суммы заказа (row_number)
```

```
    d. Оставить топ-10 клиентов
```

```
*/
```

```
SELECT
```

```
    c.customer_id,
```

```
    c.last_name,
```

```
    c.first_name,
```

```
    c.second_name,
```

```
    SUM(o.price * o.count) AS total_order_value,
```

```
    ROW_NUMBER() OVER (ORDER BY SUM(o.price * o.count) DESC) AS rank
```

```
FROM
```

```
    pharma_orders o
```

```
JOIN
```

```
    customers c ON o.customer_id = c.customer_id
```

```
GROUP BY
```

```
    c.customer_id, c.last_name, c.first_name, c.second_name
```

```
ORDER BY
```

```
    total_order_value DESC
```

```
LIMIT 10;
```



```
/*  
    7. Запрос "Накопленная сумма по клиентам"  
    а. Соединить таблицы заказов и клиентов  
    б. Соединить ФИО в одно поле  
    в. Рассчитать накопленную сумму по каждому клиенту  
*/  
SELECT  
    CONCAT(c.last_name, ' ', c.first_name, ' ', c.second_name) AS full_name,  
    SUM(po.price * po.count) AS total_amount  
FROM  
    customers c  
JOIN  
    pharma_orders po ON c.customer_id = po.customer_id  
GROUP BY  
    c.customer_id, c.last_name, c.first_name, c.second_name  
ORDER BY  
    total_amount DESC;
```

```

/*
    8. Запрос "Самые частые клиенты аптек Горздрав и Здравсити"
        а Сделать две временные таблицы: для аптеки горздрав и здравсити (WITH)
        б. Внутри каждой соединить таблицы заказов и клиентов (JOIN)
        с. Внутри каждой привести данные в формат "клиент - кол-во заказов в аптеке"
        д. Внутри каждой оставить топ 10 клиентов каждой аптеки
        е. Объединить клиентов с помощью UNION
*/
WITH gorzdrav_clients AS (
    SELECT
        CONCAT(c.last_name, ' ', c.first_name, ' ', c.second_name) AS client_name,
        COUNT(po.order_id) AS order_count
    FROM
        pharma_orders po
    JOIN
        customers c ON po.customer_id = c.customer_id
    WHERE
        po.pharmacy_name = 'Горздрав'
    GROUP BY
        c.customer_id, c.last_name, c.first_name, c.second_name
    ORDER BY
        order_count DESC
    LIMIT 10
),

```

```
zdravsiiti_clients AS (  
    SELECT  
        CONCAT(c.last_name, ' ', c.first_name, ' ', c.second_name) AS client_name,  
        COUNT(po.order_id) AS order_count  
    FROM  
        pharma_orders po  
    JOIN  
        customers c ON po.customer_id = c.customer_id  
    WHERE  
        po.pharmacy_name = 'Здравсити'  
    GROUP BY  
        c.customer_id, c.last_name, c.first_name, c.second_name  
    ORDER BY  
        order_count DESC  
    LIMIT 10  
)
```

```
SELECT
    client_name,
    order_count,
    'Горздрав' AS pharmacy
FROM
    gorzdrav_clients
```

UNION ALL

```
SELECT
    client_name,
    order_count,
    'Здравсити' AS pharmacy
FROM
    zdravsiti_clients;
```

/* 1-ый вариант

Запрос 9. "Сравнение динамики продаж между Москвой и Санкт-Петербургом по аптекам":

- а. Посчитать продажи лекарств в Москве по аптекам и по месяцам;
- б. Посчитать продажи лекарств в Санкт-Петербурге по аптекам и по месяцам;
- с. Соединить таблицы по аптекам и по месяцам и найти разницу в процентах;

*/

```
WITH msk_sales AS (  
    SELECT  
        pharmacy_name,  
        drug,  
        DATE_TRUNC('month', TO_DATE(report_date, 'YYYY-MM-DD')) AS month,  
        SUM(price * count) AS total_sales,  
        ROUND(  
            SUM(price * count) * 100.0 /  
            SUM(SUM(price * count)) OVER (PARTITION BY DATE_TRUNC('month', TO_DATE(report_date, 'YYYY-MM-DD')), pharmacy_name),  
            2  
        ) AS msk_perc_month  
    FROM  
        pharma_orders  
    WHERE  
        city = 'Москва'  
    GROUP BY  
        pharmacy_name, drug, month  
) ,
```

```
spb_sales AS (  
    SELECT  
        pharmacy_name,  
        drug,  
        DATE_TRUNC('month', TO_DATE(report_date, 'YYYY-MM-DD')) AS month,  
        SUM(price * count) AS total_sales,  
        ROUND(  
            SUM(price * count) * 100.0 /  
            SUM(SUM(price * count)) OVER (PARTITION BY DATE_TRUNC('month', TO_DATE(report_date, 'YYYY-MM-DD')), pharmacy_name),  
            2  
        ) AS spb_perc_month  
    FROM  
        pharma_orders  
    WHERE  
        city = 'Санкт-Петербург'  
    GROUP BY  
        pharmacy_name, drug, month  
)
```

```
SELECT
    COALESCE(msk.pharmacy_name, spb.pharmacy_name) AS pharmacy_name,
    COALESCE(msk.drug, spb.drug) AS drug,
    COALESCE(msk.month, spb.month) AS month,
    'Москва' AS msk_city,
    COALESCE(ROUND(msk.msk_perc_month, 2), 0.00) AS msk_perc_month,
    'Санкт-Петербург' AS spb_city,
    COALESCE(ROUND(spb.spb_perc_month, 2), 0.00) AS spb_perc_month,
    COALESCE(ROUND(msk.msk_perc_month, 2), 0.00) - COALESCE(ROUND(spb.spb_perc_month, 2), 0.00) AS
diff_perc_month
FROM
    msk_sales msk
FULL OUTER JOIN
    spb_sales spb ON
        msk.pharmacy_name = spb.pharmacy_name
        AND msk.drug = spb.drug
        AND msk.month = spb.month
ORDER BY
```

```

/*      2-ой вариант
Запрос 9. "Сравнение динамики продаж между Москвой и Санкт-Петербургом по аптекам":
    а. Посчитать продажи лекарств в Москве по аптекам и по месяцам;
    б. Посчитать продажи лекарств в Санкт-Петербурге по аптекам и по месяцам;
    в. Соединить таблицы по аптекам и по месяцам и найти разницу в процентах;

*/
WITH msk_sales AS (
    SELECT
        pharmacy_name,
        drug,
        TO_CHAR(TO_DATE(report_date, 'YYYY-MM-DD'), 'TMMonth') AS month,
        SUM(price * count) AS total_sales,
        ROUND(SUM(price * count) * 100.0 / SUM(SUM(price * count)) OVER (PARTITION BY pharmacy_name,
            TO_CHAR(TO_DATE(report_date, 'YYYY-MM-DD'), 'TMMonth'))), 2) AS msk__perc_month
    FROM
        pharma_orders
    WHERE
        city = 'Москва'
    GROUP BY
        pharmacy_name, drug, month
),

```



```
spb_sales AS (  
  SELECT  
    pharmacy_name,  
    drug,  
    TO_CHAR(TO_DATE(report_date, 'YYYY-MM-DD'), 'TMMonth') AS month,  
    SUM(price * count) AS total_sales,  
    ROUND(SUM(price * count) * 100.0 / SUM(SUM(price * count)) OVER (PARTITION BY pharmacy_name,  
      TO_CHAR(TO_DATE(report_date, 'YYYY-MM-DD'), 'TMMonth'))), 2) AS spb__perc_month  
  FROM  
    pharma_orders  
  WHERE  
    city = 'Санкт-Петербург'  
  GROUP BY  
    pharmacy_name, drug, month  
)  
,
```

```
SELECT COALESCE(msk.pharmacy_name,
spb.pharmacy_name) AS pharmacy_name,
COALESCE(msk.drug, spb.drug) AS drug,
COALESCE(msk.month, spb.month) AS month,
COALESCE(ROUND(msk.msk__perc_month, 2), 0.00) AS
msk__perc_month,
COALESCE(ROUND(spb.spb__perc_month, 2), 0.00) AS
spb__perc_month,
COALESCE(ROUND(msk.msk__perc_month, 2), 0.00) -
COALESCE(ROUND(spb.spb__perc_month, 2), 0.00) AS
diff_perc_monthFROM msk_sales mskFULL OUTER JOIN
spb_sales spb ON msk.pharmacy_name = spb.pharmacy_name
AND msk.drug = spb.drug AND msk.month =
spb.monthORDER BY pharmacy_name, drug, month;
```

/*

Запрос 10. Лекарства от насморка.

Выделяем препараты, начинающиеся со слова “аква” (с использованием оператора LIKE).

Приводим данные к нижнему регистру, группируем и подсчитываем общий объем продаж для каждого препарата.

Ранжируем по убыванию объема продаж и подсчитываем долю продаж каждого лекарства в общем объеме.

*/

```
WITH lower_case_drugs AS (  
    SELECT  
        LOWER(drug) AS drug_name,  
        SUM(price * count) AS total_sales  
    FROM  
        pharma_orders  
    WHERE  
        LOWER(drug) LIKE '%аква%'  
    GROUP BY  
        LOWER(drug)  
)  
total_sales AS (  
    SELECT  
        SUM(total_sales) AS overall_sales  
    FROM  
        lower_case_drugs  
)
```

```
SELECT
    drug_name,|
    total_sales,
    ROUND(total_sales * 100.0 / (SELECT overall_sales FROM total_sales), 2) AS sales_percentage
FROM
    lower_case_drugs
ORDER BY
    total_sales DESC;
```

```

/*
    Запрос 11. Кто наши клиенты.
    а. Вычисляем возраст клиентов на основе даты рождения с использованием функции для работы с
датоми;
    б. Затем используем оператор CASE WHEN для расчета, кто наши клиенты;
    в. Описываем каждую группу мужчин: мужчины младше 30 лет, мужчины от 30 до 45лет, и мужчины
45+ лет;
    г. Описываем каждую группу женщин: женщины младше 30 лет, женщины от 30 до 45лет, и женщины
45+ лет;
    е. Подсчитываем долю продаж на каждую из этих групп.
*/
WITH age_gender_sales AS (
    SELECT
        c.customer_id,
        c.gender,
        SUM(po.price * po.count) AS total_sales,
        EXTRACT(YEAR FROM AGE(TO_DATE(c.date_of_birth, 'YYYY-MM-DD')))) AS age
    FROM
        customers c
    JOIN
        pharma_orders po ON c.customer_id = po.customer_id
    GROUP BY
        c.customer_id, c.gender, c.date_of_birth
),

```

```
grouped_sales AS (  
    SELECT  
        CASE  
            WHEN gender = 'муж' AND age < 30 THEN '1Мужчины младше 30 лет'  
            WHEN gender = 'муж' AND age >= 30 AND age < 45 THEN '2Мужчины от 30 до 45 лет'  
            WHEN gender = 'муж' AND age >= 45 THEN '3Мужчины 45+ лет'  
            WHEN gender = 'жен' AND age < 30 THEN '4Женщины младше 30 лет'  
            WHEN gender = 'жен' AND age >= 30 AND age < 45 THEN '5Женщины от 30 до 45 лет'  
            WHEN gender = 'жен' AND age >= 45 THEN '6Женщины 45+ лет'  
            ELSE 'Не определено'  
        END AS age_group,  
        SUM(total_sales) AS group_sales  
    FROM  
        age_gender_sales  
    GROUP BY  
        age_group  
)  
total_sales AS (  
    SELECT  
        SUM(total_sales) AS overall_sales  
    FROM  
        age_gender_sales  
)
```

```
|  
SELECT  
    gs.age_group,  
    gs.group_sales,  
    ROUND((gs.group_sales / ts.overall_sales) * 100, 2) AS sales_share_percentage  
FROM  
    grouped_sales gs,  
    total_sales ts  
ORDER BY  
    gs.age_group;
```