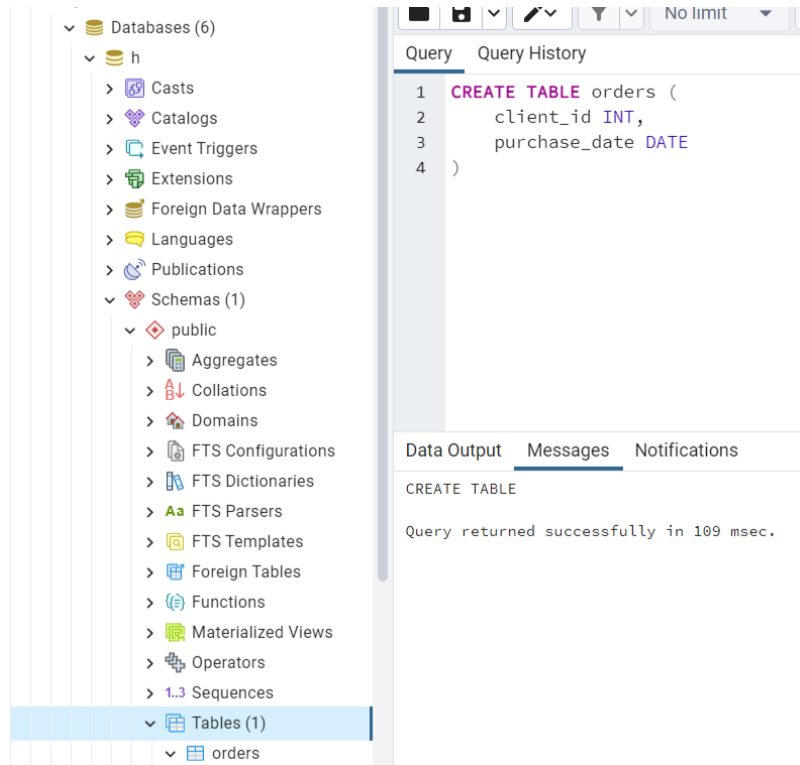


Смирнов Николай

Задание 1 (SQL) (для кандидатов)

Для начала перенесём данные из csv файла в таблицу в pgAdmin4.

Создадим новую БД, создадим таблицу orders с нужными полями



Перенесём данные в таблицу orders

```
SQL Shell (psql)  
Server [localhost]:  
Database [postgres]: h  
Port [5432]:  
Username [postgres]:  
Пароль пользователя postgres:  
psql (10.23)  
ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается от основной  
                    страницы Windows (1251).  
                    8-битовые (русские) символы могут отображаться некорректно.  
                    Подробнее об этом смотрите документацию psql, раздел  
                    "Notes for Windows users".  
Введите "help", чтобы получить справку.  
  
h=# \COPY orders FROM 'C:\data\Orders.csv' DELIMITER ',' CSV HEADER;  
ОШИБКА: значение поля типа date/time вне диапазона: "6/18/2015"  
ПОДСКАЗКА: Возможно, вам нужно изменить настройку "datestyle".  
КОНТЕКСТ: COPY orders, строка 2, столбец purchase_date: "6/18/2015"  
h=# SET datestyle = 'ISO, MDY';  
SET  
h=# \COPY orders FROM 'C:\data\Orders.csv' DELIMITER ',' CSV HEADER;  
COPY 271  
h=#
```

Теперь можно выполнять запросы.

1. Новые торговые точки

То есть, нужно для каждого месяца вывести торговые точки, которых раньше никогда не было, и в этом месяце они появились впервые за всё время.

Запрос:

```
WITH FirstPurchases AS (  
    SELECT  
        client_id,  
        MIN(purchase_date) AS first_purchase_date  
    FROM  
        Orders  
    GROUP BY  
        client_id  
)  
MonthlyFirstPurchases AS (  
    SELECT  
        client_id,  
        DATE_TRUNC('month', first_purchase_date) AS purchase_month  
    FROM  
        FirstPurchases  
)  
SELECT  
    TO_CHAR(m.purchase_month, 'YYYY/MM') AS purchase_month,  
    m.client_id  
FROM  
    MonthlyFirstPurchases m  
WHERE  
    NOT EXISTS (  
        SELECT 1
```

FROM MonthlyFirstPurchases m2

WHERE m2.client_id = m.client_id AND m2.purchase_month < m.purchase_month

)

ORDER BY

purchase_month;

Query

Query History

```

1  WITH FirstPurchases AS (
2      SELECT
3          client_id,
4          MIN(purchase_date) AS first_purchase_date
5      FROM
6          Orders
7      GROUP BY
8          client_id
9  ),
10 MonthlyFirstPurchases AS (
11     SELECT
12         client_id,
13         DATE_TRUNC('month', first_purchase_date) AS purchase_month
14     FROM
15         FirstPurchases
16 )
17 SELECT

```

Data Output

Messages

Notifications

≡+

📄

▼

📋

🗑️

🔍

⬇️

📈

	purchase_month text	client_id integer
1	2015/05	3984
2	2015/05	3965
3	2015/05	8791
4	2015/05	7128
5	2015/05	3903
6	2015/05	2712
7	2015/05	7300

Total rows: 115 of 115

Query complete 00:00:00.129

2. Торговые точки, сделавшие заказ в прошлом месяце и в этом

Запрос

WITH MonthlyOrders AS (

SELECT

 $client_id,$

TO_CHAR(purchase_date, 'YYYY-MM') AS month_year

FROM

Orders

GROUP BY

client_id, month_year

),

PreviousMonthOrders AS (

SELECT

client_id,

month_year,

*LEAD(month_year) OVER (PARTITION BY client_id ORDER BY
month_year) AS next_month_year*

FROM

MonthlyOrders

)

SELECT

current.month_year,

current.client_id

FROM

PreviousMonthOrders current

JOIN

PreviousMonthOrders previous

ON

current.client_id = previous.client_id

AND

previous.next_month_year = current.month_year

ORDER BY

current.month_year, current.client_id;

h/postgres@PostgreSQL 10

Query Query History

```

1 WITH MonthlyOrders AS (
2     SELECT
3         client_id,
4         TO_CHAR(purchase_date, 'YYYY-MM') AS month_year
5     FROM
6         Orders
7     GROUP BY
8         client_id, month_year
9 ),
10 PreviousMonthOrders AS (
11     SELECT
12         client_id,
13         month_year,
14         LEAD(month_year) OVER (PARTITION BY client_id ORDER BY month_year) AS next_month_year
15     FROM
16         MonthlyOrders
17 )

```

Data Output Messages Notifications

	month_year text	client_id integer
1	2015-06	250
2	2015-06	251
3	2015-06	253
4	2015-06	527
5	2015-06	747
6	2015-06	886
7	2015-06	...

Total rows: 153 of 153 Query complete 00:00:00.061

3. Торговые точки, которые когда-то что-то заказали (только не в прошлом месяце) и вернувшиеся.

Для каждого месяца выписать точки, которые когда-то были, но не в прошлом месяце.

Запрос

```

WITH OrderMonths AS (
    SELECT
        client_id,
        TO_CHAR(purchase_date, 'YYYY-MM') AS month
    FROM Orders
    GROUP BY client_id, TO_CHAR(purchase_date, 'YYYY-MM')
),
PreviousOrders AS (
    SELECT
        o1.client_id,
        o1.month AS current_month,

```

```

        o2.month AS previous_month
FROM OrderMonths o1
LEFT JOIN OrderMonths o2
    ON o1.client_id = o2.client_id
    AND TO_CHAR((TO_DATE(o1.month || '-01', 'YYYY-MM-DD') - INTERVAL
'1 month'), 'YYYY-MM') = o2.month
WHERE o2.month IS NULL -- нет заказов в прошлом месяце
),
ReturnedOrders AS (
    SELECT
        po.client_id,
        po.current_month
    FROM PreviousOrders po
    JOIN OrderMonths o
        ON po.client_id = o.client_id
        AND o.month < po.current_month -- есть заказы в более ранних месяцах
    GROUP BY po.client_id, po.current_month
)
SELECT
    current_month,
    client_id
FROM ReturnedOrders
ORDER BY current_month, client_id;

```

The screenshot shows a PostgreSQL query editor with the following SQL query:

```

1 WITH OrderMonths AS (
2     SELECT
3         client_id,
4         TO_CHAR(purchase_date, 'YYYY-MM') AS month
5     FROM Orders
6     GROUP BY client_id, TO_CHAR(purchase_date, 'YYYY-MM')
7 ),
8 PreviousOrders AS (
9     SELECT
10        o1.client_id,
11        o1.month AS current_month,
12        o2.month AS previous_month
13     FROM OrderMonths o1
14     LEFT JOIN OrderMonths o2
15         ON o1.client_id = o2.client_id
16         AND TO_CHAR((TO_DATE(o1.month || '-01', 'YYYY-MM-DD') - INTERVAL '1 month'), 'YYYY-MM') < TO_CHAR(o2.month, 'YYYY-MM')
17 )

```

The results are displayed in a table with the following data:

	current_month	client_id
1	2015-07	641
2	2015-07	2066
3	2015-07	2154
4	2015-07	5604
5	2015-07	6513
6	2015-08	5103

Total rows: 6 of 6 Query complete 00:00:00.121

4. Торговые точки, отвалившие в этом месяце.

Это значит, нужны точки, которые в данном месяце сделали последний заказ и больше никогда не делали заказов.

WITH LastOrders AS (

SELECT

client_id,

TO_CHAR(MAX(purchase_date), 'YYYY-MM') AS last_order_month

FROM Orders

GROUP BY client_id

),

MonthlyOrders AS (

SELECT

client_id,

TO_CHAR(purchase_date, 'YYYY-MM') AS order_month

FROM Orders

GROUP BY client_id, TO_CHAR(purchase_date, 'YYYY-MM')

)

SELECT

mo.order_month AS month,

mo.client_id

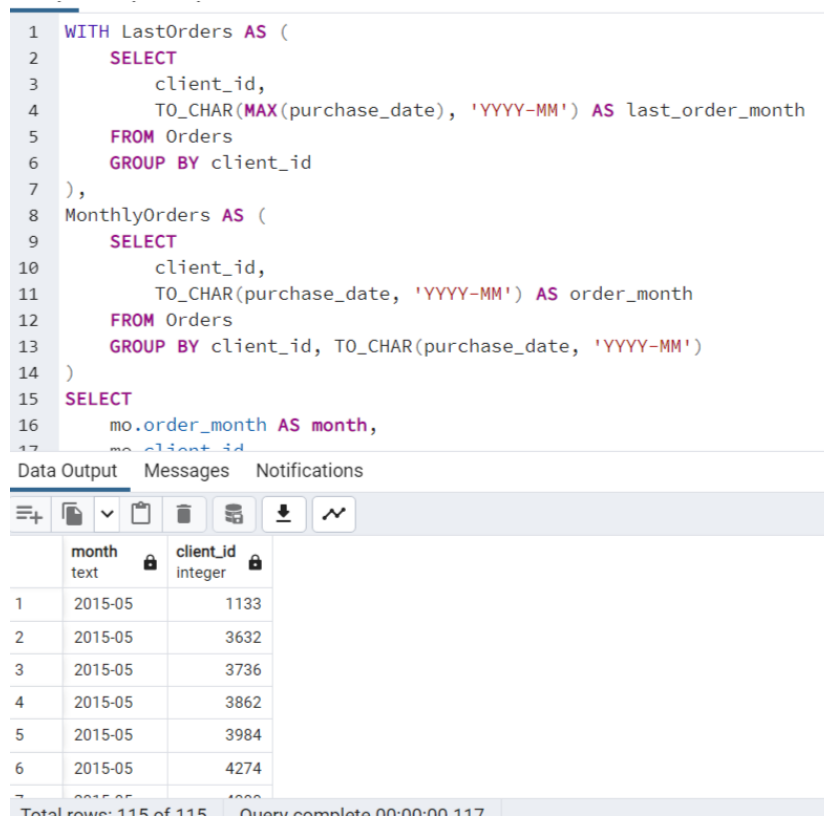
FROM MonthlyOrders mo

JOIN LastOrders lo

ON mo.client_id = lo.client_id

AND mo.order_month = lo.last_order_month

ORDER BY month, client_id;



The screenshot shows a SQL query editor with a query window and a results window. The query window contains a SQL query that joins 'MonthlyOrders' and 'LastOrders' tables. The results window shows a table with two columns: 'month' (text) and 'client_id' (integer). The table contains 115 rows of data, with the first 6 rows visible in the screenshot. The status bar at the bottom indicates 'Total rows: 115 of 115' and 'Query complete 00:00:00.117'.

```
1 WITH LastOrders AS (  
2     SELECT  
3         client_id,  
4         TO_CHAR(MAX(purchase_date), 'YYYY-MM') AS last_order_month  
5     FROM Orders  
6     GROUP BY client_id  
7 ),  
8 MonthlyOrders AS (  
9     SELECT  
10        client_id,  
11        TO_CHAR(purchase_date, 'YYYY-MM') AS order_month  
12    FROM Orders  
13    GROUP BY client_id, TO_CHAR(purchase_date, 'YYYY-MM')  
14 )  
15 SELECT  
16     mo.order_month AS month,  
17     mo.client_id
```

	month text	client_id integer
1	2015-05	1133
2	2015-05	3632
3	2015-05	3736
4	2015-05	3862
5	2015-05	3984
6	2015-05	4274
7	2015-05	1000

Total rows: 115 of 115 Query complete 00:00:00.117

Результаты запросов представлены в виде файлов
query1.csv, query2.csv, query3.csv и query4.csv.

