

# Задание 1

Загружаем Orders.csv в mysql.

```
CREATE TABLE Orders (  
    client_id INT,  
    purchase_date DATE  
);
```

```
LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Orders.csv'  
INTO TABLE Orders  
FIELDS TERMINATED BY ','  
LINES TERMINATED BY '\\n'  
IGNORE 1 ROWS  
(@client_id, @purchase_date)  
SET client_id = CAST(@client_id AS UNSIGNED),  
    purchase_date = STR_TO_DATE(@purchase_date, '%m/%d/%Y');
```

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the 'horse' database structure, including the 'orders' table with columns 'client\_id' and 'purchase\_date'. The 'Table: orders' information pane shows the columns and their data types: 'client\_id' as 'int' and 'purchase\_date' as 'date'. The main query editor shows a query: `SELECT * FROM horse.orders;`. The 'Result Grid' pane displays the query results, showing a list of rows with 'client\_id' and 'purchase\_date' values. The 'Output' pane at the bottom shows the execution of the query, indicating it was successful and limited to 1000 rows.

client_id	purchase_date
165	2015-06-18
205	2015-06-23
235	2015-06-15
250	2015-05-25
250	2015-06-24
250	2015-07-24
251	2015-05-02
251	2015-06-01
251	2015-07-01
253	2015-05-20
253	2015-06-19
253	2015-07-19
400	2015-06-03
440	2015-06-10
472	2015-06-10
527	2015-05-13
527	2015-06-12
527	2015-07-12
611	2015-06-23
641	2015-05-04
641	2015-07-28
669	2015-06-05
747	2015-05-25
747	2015-06-24
747	2015-07-24
766	2015-06-21
781	2015-06-29
827	2015-06-09
886	2015-05-02
886	2015-06-01
886	2015-07-01
913	2015-06-30
939	2015-05-19
939	2015-06-18
939	2015-07-18
944	2015-06-15
963	2015-05-13
963	2015-06-12
976	2015-06-11
999	2015-06-29
1019	2015-05-16

## 1. Новые торговые точки:

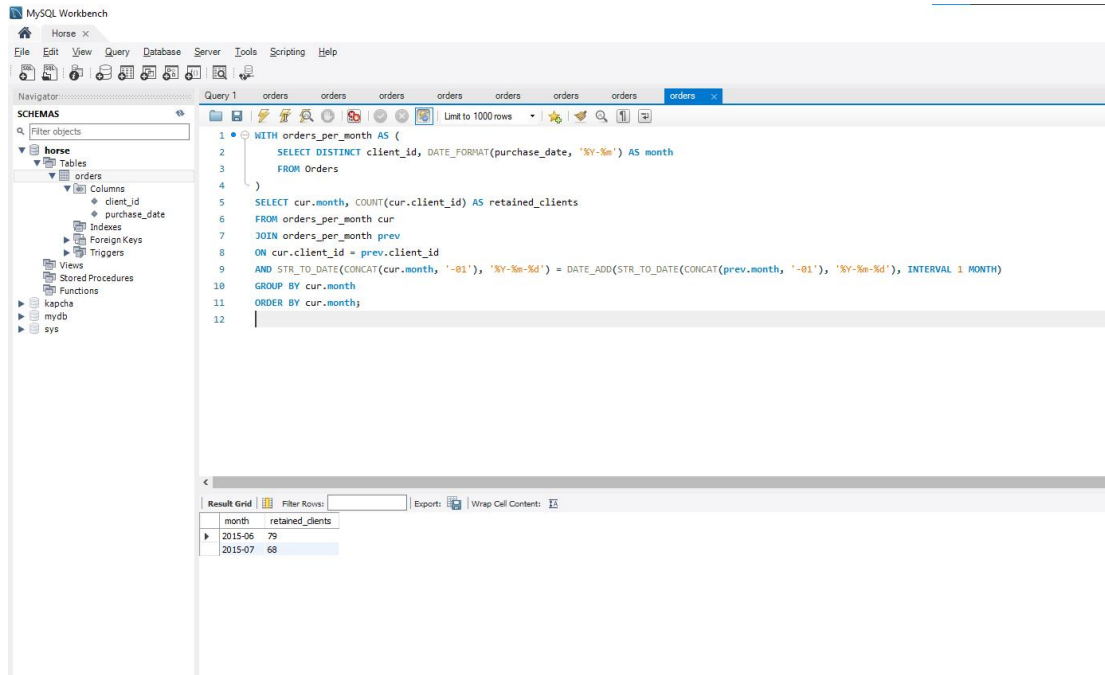
```
WITH first_orders AS (  
    SELECT client_id, MIN(DATE_FORMAT(purchase_date, '%Y-%m')) AS first_month  
    FROM Orders  
    GROUP BY client_id  
)  
SELECT first_month AS month, COUNT(client_id) AS new_clients  
FROM first_orders  
GROUP BY first_month  
ORDER BY first_month;
```

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view with 'horse' as the selected database, containing tables 'orders', 'kapcha', 'mydb', and 'sys'. The 'orders' table is expanded, showing columns 'client\_id' and 'purchase\_date'. The main query editor displays the SQL code from the first block. Below the editor, the 'Result Grid' shows the output of the query:

month	new_clients
2015-05	100
2015-06	15

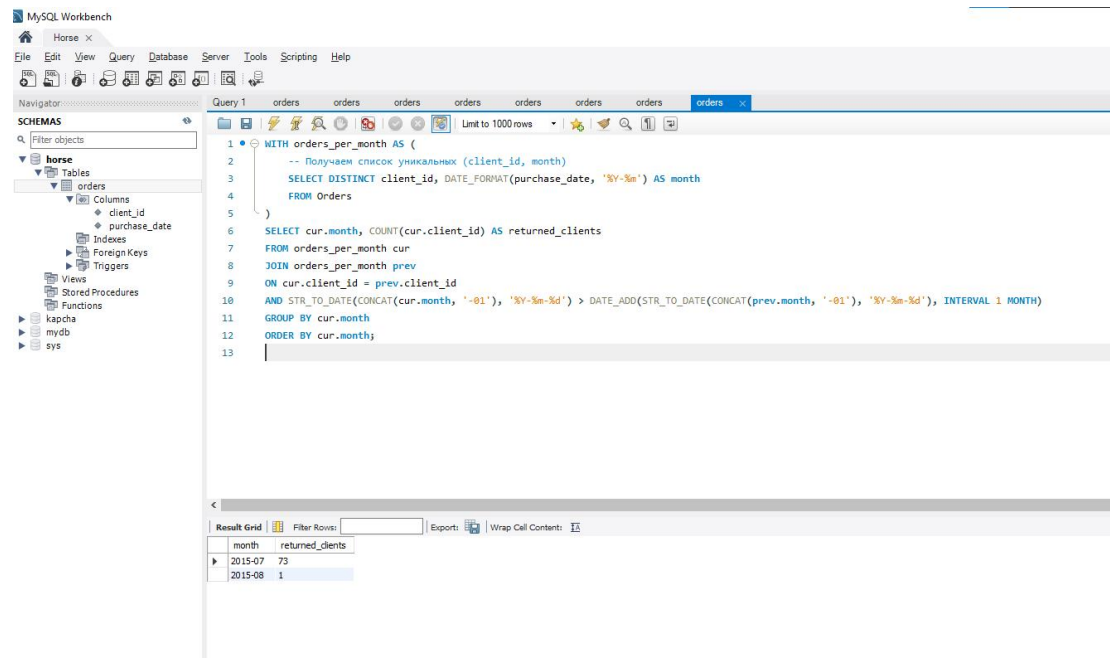
## 2. Торговые точки, сделавшие заказ в прошлом месяце и в этом

```
WITH orders_per_month AS (  
    SELECT DISTINCT client_id, DATE_FORMAT(purchase_date, '%Y-%m') AS month  
    FROM Orders  
)  
SELECT cur.month, COUNT(cur.client_id) AS retained_clients  
FROM orders_per_month cur  
JOIN orders_per_month prev  
ON cur.client_id = prev.client_id  
AND STR_TO_DATE(CONCAT(cur.month, '-01'), '%Y-%m-%d') =  
DATE_ADD(STR_TO_DATE(CONCAT(prev.month, '-01'), '%Y-%m-%d'), INTERVAL 1 MONTH)  
GROUP BY cur.month  
ORDER BY cur.month;
```



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

```
WITH orders_per_month AS (
  -- Получаем список уникальных (client_id, month)
  SELECT DISTINCT client_id, DATE_FORMAT(purchase_date, '%Y-%m') AS month
  FROM Orders
)
SELECT cur.month, COUNT(cur.client_id) AS returned_clients
FROM orders_per_month cur
JOIN orders_per_month prev
ON cur.client_id = prev.client_id
AND STR_TO_DATE(CONCAT(cur.month, '-01'), '%Y-%m-%d') >
DATE_ADD(STR_TO_DATE(CONCAT(prev.month, '-01'), '%Y-%m-%d'), INTERVAL 1 MONTH)
GROUP BY cur.month
ORDER BY cur.month;
```



#### 4. Торговые точки, отвалившие в этом месяце.

```

WITH orders_per_month AS (
  SELECT DISTINCT client_id, DATE_FORMAT(purchase_date, '%Y-%m') AS month
  FROM Orders
)
SELECT DATE_FORMAT(
  STR_TO_DATE(CONCAT(prev.month, '-01'), '%Y-%m-%d') + INTERVAL 1 MONTH, '%Y-%m'
) AS month,
COUNT(prev.client_id) AS churned_clients
FROM orders_per_month prev
LEFT JOIN orders_per_month cur
ON prev.client_id = cur.client_id
AND cur.month = DATE_FORMAT(
  STR_TO_DATE(CONCAT(prev.month, '-01'), '%Y-%m-%d') + INTERVAL 1 MONTH, '%Y-%m'
)
WHERE cur.client_id IS NULL
AND prev.month < (SELECT MAX(month) FROM orders_per_month)
GROUP BY month
ORDER BY month;

```

MySQL Workbench

Horse x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

horse

Tables

orders

Columns

client\_id

purchase\_date

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

kapcha

mydb

sys

Query 1

orders orders orders orders orders orders orders orders orders

Limit to 1000 rows

```
1 WITH orders_per_month AS (  
2     SELECT DISTINCT client_id, DATE_FORMAT(purchase_date, '%Y-%m') AS month  
3     FROM Orders  
4 )  
5 SELECT DATE_FORMAT(  
6     STR_TO_DATE(CONCAT(prev.month, '-01'), '%Y-%m-%d') + INTERVAL 1 MONTH, '%Y-%m'  
7 ) AS month,  
8 COUNT(prev.client_id) AS churned_clients  
9 FROM orders_per_month prev  
10 LEFT JOIN orders_per_month cur  
11 ON prev.client_id = cur.client_id  
12 AND cur.month = DATE_FORMAT(  
13     STR_TO_DATE(CONCAT(prev.month, '-01'), '%Y-%m-%d') + INTERVAL 1 MONTH, '%Y-%m'  
14 )  
15 WHERE cur.client_id IS NULL  
16 AND prev.month < (SELECT MAX(month) FROM orders_per_month)  
17 GROUP BY month  
18 ORDER BY month;  
19
```

Result Grid

Filter Rows:

Export: Wrap Cell Content: W

month	churned_clients
2015-06	21
2015-07	26
2015-08	73