

## Задача №1. Напишите SQL-запрос

Имеется таблица city\_population с населением городов: city (наименование города), population (численность населения).

Необходимо написать запрос, который выводит город с минимальным населением.

*Результат — скрипт в формате .sql*

## Задача №2. Напишите SQL-запрос

Имеется таблица A такого вида:

ID	Name	Val
1	A	10
1	B	9
2	B	8
2	C	7
3	A	6
3	C	5

Напишите SQL-запрос, который преобразует таблицу в такой вид:

ID	A	B	C
1	10	9	
2		8	7
3	6		5

*Результат — скрипт в формате .sql*

## Задача №3. Напишите SQL-запрос

Есть таблица пользователей user (user\_id — id пользователя, installed\_at — дата установки) и таблица активности client\_session (user\_id, created\_at — таймстемп активности).

Необходимо написать SQL-запрос который считает Retention 1, 3, 7 дня по пользователям с группировкой установок по месяцам (с января 2020-го года).

*Результат — скрипт в формате .sql*

# тестовое\_задание-valiotti-sql

## Задание 1

```
-- Используется диалект PostgreSQL.
```

```
SELECT city
FROM city_population
ORDER BY population
LIMIT 1;
```

## Задание 2

```
-- Используется диалект PostgreSQL.
```

```
CREATE EXTENSION IF NOT EXISTS tablefunc;
```

```
SELECT * FROM
crosstab
(
    'SELECT * FROM foo ORDER BY ID',
    'SELECT DISTINCT Name FROM foo ORDER BY Name'
) AS (ID INT, A INT, B INT, C INT)
ORDER BY ID;
```

## Задание 3

```
-- Используется диалект PostgreSQL.
```

```
SELECT to_char(date_inst, 'YYYY-MM') AS month_inst,
       date_diff AS date_diff,
       SUM(ret_base) AS month_ret_base,
       SUM(cht_size) AS month_cht_size,
       SUM(ret_base)::float / SUM(cht_size)::float AS month_ret_rate
FROM
(
    WITH inst
    AS
    (
        SELECT installed_at::date as date_inst, COUNT(DISTINCT user_id) AS cht_size
        FROM user
        WHERE installed_at >= '2020-01-01'
          AND installed_at <= now()::date - interval '7 days'
        GROUP BY 1
    )
    ,
    ret
    AS
    (
        SELECT installed_at::date AS date_inst,
               created_at::date - installed_at::date AS date_diff,
               COUNT(DISTINCT user.user_id) AS ret_base
        FROM user
        LEFT JOIN client_session USING(user_id)
        WHERE installed_at >= '2020-01-01'
          AND installed_at <= now()::date - interval '7 days'
          AND created_at::date - installed_at::date IN (1, 3, 7)
        GROUP BY 1, 2
    )
    SELECT inst.date_inst AS date_inst,
           ret.date_diff AS date_diff,
           ret.ret_base AS ret_base,
           inst.cht_size as cht_size,
           ret.ret_base::float / inst.cht_size::float AS ret_rate
    FROM ret LEFT JOIN inst USING(date_inst)
) AS foo
GROUP BY 1, 2
ORDER BY 1, 2
```