



SQL ПРОЕКТ

ЧАСТЬ 1

Задача 1

Найдите количество вопросов, которые набрали больше 300 очков или как минимум 100 раз были добавлены в «Закладки».

```
SELECT COUNT(distinct p.id)
FROM stackoverflow.posts AS p
LEFT OUTER JOIN stackoverflow.post_types AS pt ON p.post_type_id = pt.id
WHERE (score > 300 OR favorites_count >= 100) AND
type = 'Question'
;
```

Задача 2

Сколько в среднем в день задавали вопросов с 1 по 18 ноября 2008 включительно? Результат округлите до целого числа.

```
SELECT ROUND(AVG(count),0)
FROM(
SELECT DATE_TRUNC('day',creation_date)::date,
COUNT(p.id) AS count
FROM stackoverflow.posts AS p
LEFT OUTER JOIN stackoverflow.post_types AS pt ON p.post_type_id = pt.id
WHERE DATE_TRUNC('day',creation_date) BETWEEN '2008-11-01' AND '2008-11-18'
AND type = 'Question'
GROUP BY DATE_TRUNC('day',creation_date)::date
ORDER BY DATE_TRUNC('day',creation_date)::date) AS i
;
```

Задача 3

Сколько пользователей получили значки сразу в день регистрации? Выведите количество уникальных пользователей.

```
SELECT COUNT(DISTINCT(u.id))
FROM stackoverflow.users AS u
JOIN stackoverflow.badges AS b ON u.id = b.user_id
WHERE DATE_TRUNC('day', u.creation_date)::date = DATE_TRUNC('day', b.creation_date)::date
;
```

Задача 4

Сколько уникальных постов пользователя с именем Joel Coehoorn получили хотя бы один голос?

```
SELECT COUNT(DISTINCT(p.id))
FROM stackoverflow.posts AS p
JOIN stackoverflow.users AS u ON p.user_id = u.id
JOIN stackoverflow.votes AS v ON p.id = v.post_id
WHERE display_name = 'Joel Coehoorn'
AND vote_type_id >=1
;
```

Задача 5

Выгрузите все поля таблицы `vote_types`. Добавьте к таблице поле `rank`, в которое войдут номера записей в обратном порядке. Таблица должна быть отсортирована по полю `id`.

```
SELECT *
FROM(
SELECT * ,
      ROW_NUMBER() OVER (ORDER BY id DESC) AS rank
FROM stackoverflow.vote_types) AS i
ORDER BY id
;
```

Задача 6

Отберите 10 пользователей, которые поставили больше всего голосов типа `Close`. Отобразите таблицу из двух полей: идентификатором пользователя и количеством голосов. Отсортируйте данные сначала по убыванию количества голосов, потом по убыванию значения идентификатора пользователя.

```

SELECT *
FROM(
SELECT u.id,
      COUNT(u.id)
FROM stackoverflow.users AS u
JOIN stackoverflow.votes AS v ON u.id = v.user_id
JOIN stackoverflow.vote_types AS vt ON v.vote_type_id = vt.id
WHERE name = 'Close'
GROUP BY u.id
ORDER BY COUNT(u.id) DESC
LIMIT 10) AS i
ORDER BY count DESC,
      id DESC
;

```

Задача 7

Отберите 10 пользователей по количеству значков, полученных в период с 15 ноября по 15 декабря 2008 года включительно. Отобразите несколько полей:

- идентификатор пользователя;
- число значков;
- место в рейтинге — чем больше значков, тем выше рейтинг.

Пользователям, которые набрали одинаковое количество значков, присвойте одно и то же место в рейтинге.

Отсортируйте записи по количеству значков по убыванию, а затем по возрастанию значения идентификатора пользователя.

```

SELECT u.id,
      COUNT(b.id),
      DENSE_RANK() OVER (ORDER BY COUNT(b.id) DESC)
FROM stackoverflow.users AS u
JOIN stackoverflow.badges AS b ON u.id = b.user_id
WHERE DATE_TRUNC('day', b.creation_date)::date BETWEEN '2008-11-15' AND '2008-12-15'
GROUP BY u.id
ORDER BY COUNT(b.id) DESC,
      u.id
LIMIT 10
;

```

Задача 8

Сколько в среднем очков получает пост каждого пользователя?

Сформируйте таблицу из следующих полей:

- заголовок поста;
- идентификатор пользователя;
- число очков поста;
- среднее число очков пользователя за пост, округлённое до целого числа.

Не учитывайте посты без заголовка, а также те, что набрали ноль очков.

```
SELECT title,
       u.id,
       score,
       ROUND(AVG(score) OVER (PARTITION BY u.id), 0) AS user_avg
FROM stackoverflow.users AS u
JOIN stackoverflow.posts AS p ON u.id = p.user_id
WHERE score != 0 AND title IS NOT NULL
;
```

Задача 9

Отобразите заголовки постов, которые были написаны пользователями, получившими более 1000 значков. Посты без заголовков не должны попасть в список.

```
WITH users AS (
  SELECT u.id,
         COUNT(b.id) AS count
  FROM stackoverflow.users AS u
  LEFT OUTER JOIN stackoverflow.badges AS b ON u.id = b.user_id
  GROUP BY u.id
  HAVING COUNT(b.id) > 1000)

SELECT title
FROM users AS u
JOIN stackoverflow.posts AS p ON u.id = p.user_id
WHERE title IS NOT NULL
;
```

Задача 10

Напишите запрос, который выгрузит данные о пользователях из США (англ. United States). Разделите пользователей на три группы в зависимости от количества просмотров их профилей:

- пользователям с числом просмотров больше либо равным 350 присвойте группу **1**;
- пользователям с числом просмотров меньше 350, но больше либо равно 100 — группу **2**;
- пользователям с числом просмотров меньше 100 — группу **3**.

Отобразите в итоговой таблице идентификатор пользователя, количество просмотров профиля и группу. Пользователи с нулевым количеством просмотров не должны войти в итоговую таблицу.

```
SELECT id,
       views,
       CASE
         WHEN views >= 350 THEN 1
         WHEN views >= 100 AND views < 350 THEN 2
         WHEN views < 100 THEN 3
       END
FROM stackoverflow.users
WHERE location LIKE '%United%States%' AND views != 0
;
```

Задача 11

Дополните предыдущий запрос. Отобразите лидеров каждой группы — пользователей, которые набрали максимальное число просмотров в своей группе. Выведите поля с идентификатором пользователя, группой и количеством просмотров. Отсортируйте таблицу по убыванию просмотров, а затем по возрастанию значения идентификатора.

```
WITH users AS(
SELECT id,
       views,
       CASE
         WHEN views >= 350 THEN 1
         WHEN views >= 100 AND views < 350 THEN 2
         WHEN views < 100 THEN 3
       END AS groups
FROM stackoverflow.users
```

```

WHERE location LIKE '%United%States%' AND views != 0)

SELECT id,
       views,
       groups
FROM(
SELECT id,
       views,
       groups,
       MAX(views) OVER (PARTITION BY groups) AS max_views
FROM users) AS i
WHERE groups = 1 AND views = max_views
OR groups = 2 AND views = max_views
OR groups = 3 AND views = max_views
ORDER BY 2 DESC,
       1
;

```

Задача 12

Посчитайте ежедневный прирост новых пользователей в ноябре 2008 года. Сформируйте таблицу с полями:

- номер дня;
- число пользователей, зарегистрированных в этот день;
- сумму пользователей с накоплением.

```

SELECT EXTRACT(DAY FROM creation_date) AS day,
       COUNT(id),
       SUM(COUNT(id)) OVER (ORDER BY EXTRACT(DAY FROM creation_date)) AS id_cum
FROM stackoverflow.users
WHERE EXTRACT(MONTH FROM creation_date) = 11 AND
      EXTRACT(YEAR FROM creation_date) = 2008
GROUP BY EXTRACT(DAY FROM creation_date)
;

```

Задача 13

Для каждого пользователя, который написал хотя бы один пост, найдите интервал между регистрацией и временем создания первого поста. Отобразите:

- идентификатор пользователя;
- разницу во времени между регистрацией и первым постом.

```

SELECT u.id,
       MIN(p.creation_date) - u.creation_date
FROM stackoverflow.users AS u
JOIN stackoverflow.posts AS p ON u.id = p.user_id
WHERE u.id IN (SELECT u.id
               FROM stackoverflow.users AS u
               JOIN stackoverflow.posts AS p ON u.id = p.user_id
               GROUP BY u.id
               HAVING COUNT(p.id) != 0)
GROUP BY 1,
u.creation_date
;

```

ЧАСТЬ 2

Задача 1

Выведите общую сумму просмотров постов за каждый месяц 2008 года. Если данных за какой-либо месяц в базе нет, такой месяц можно пропустить. Результат отсортируйте по убыванию общего количества просмотров.

```

SELECT DISTINCT DATE_TRUNC('month', creation_date)::date,
sum(views_count) over (partition by date_trunc('month', creation_date)::date)
from stackoverflow.posts
where extract(year from creation_date) = 2008
order by 2 desc

```

Задача 2

Выведите имена самых активных пользователей, которые в первый месяц после регистрации (включая день регистрации) дали больше 100 ответов. Вопросы, которые задавали пользователи, не учитывайте. Для каждого имени пользователя выведите количество уникальных значений `user_id`. Отсортируйте результат по полю с именами в лексикографическом порядке.

```

SELECT u.display_name,
       COUNT(DISTINCT p.user_id) as count_id
FROM stackoverflow.users as u
JOIN stackoverflow.posts as p on u.id = p.user_id
JOIN stackoverflow.post_types as pt on p.post_type_id = pt.id
WHERE type = 'Answer'
      AND p.creation_date::date <= (u.creation_date::date + INTERVAL '1 month')
GROUP BY u.display_name

```

```
HAVING COUNT(p.id) > 100
ORDER BY u.display_name
```

Задача 3

Выведите количество постов за 2008 год по месяцам. Отберите посты от пользователей, которые зарегистрировались в сентябре 2008 года и сделали хотя бы один пост в декабре того же года. Отсортируйте таблицу по значению месяца по убыванию.

```
SELECT DATE_TRUNC('month', p.creation_date)::date AS date,
       COUNT(DISTINCT p.id) AS count
FROM stackoverflow.users AS u
JOIN stackoverflow.posts AS p ON u.id = p.user_id
WHERE u.id IN(
    SELECT DISTINCT u.id
    FROM stackoverflow.users AS u
    JOIN stackoverflow.posts AS p ON u.id = p.user_id
    WHERE DATE_TRUNC('month', u.creation_date)::date = '2008-09-01'
    AND DATE_TRUNC('month', p.creation_date)::date = '2008-12-01')
GROUP BY 1
ORDER BY 1 DESC
;
```

Задача 4

Используя данные о постах, выведите несколько полей:

- идентификатор пользователя, который написал пост;
- дата создания поста;
- количество просмотров у текущего поста;
- сумму просмотров постов автора с накоплением.

Данные в таблице должны быть отсортированы по возрастанию идентификаторов пользователей, а данные об одном и том же пользователе — по возрастанию даты создания поста.

```
SELECT user_id,
       creation_date,
       views_count,
       SUM(views_count) OVER (PARTITION BY user_id ORDER BY creation_date)
FROM stackoverflow.posts
```



```
ORDER BY 1,  
        2  
;
```

Задача 5

Сколько в среднем дней в период с 1 по 7 декабря 2008 года включительно пользователи взаимодействовали с платформой? Для каждого пользователя отберите дни, в которые он или она опубликовали хотя бы один пост. Нужно получить одно целое число — не забудьте округлить результат.

```
SELECT ROUND(AVG(date), 0)  
FROM(  
    SELECT u.id,  
           COUNT(DISTINCT DATE_TRUNC('day', p.creation_date)::date) AS date  
    FROM stackoverflow.users AS u  
    JOIN stackoverflow.posts AS p ON u.id = p.user_id  
    WHERE DATE_TRUNC('day', p.creation_date)::date BETWEEN '2008-12-01' AND '2008-12-07'  
    GROUP BY u.id) AS i  
;
```

Задача 6

На сколько процентов менялось количество постов ежемесячно с 1 сентября по 31 декабря 2008 года? Отобразите таблицу со следующими полями:

- номер месяца;
- количество постов за месяц;
- процент, который показывает, насколько изменилось количество постов в текущем месяце по сравнению с предыдущим.

Если постов стало меньше, значение процента должно быть отрицательным, если больше — положительным. Округлите значение процента до двух знаков после запятой.

Напомним, что при делении одного целого числа на другое в PostgreSQL в результате получится целое число, округлённое до ближайшего целого вниз. Чтобы этого избежать, переведите делимое в тип `numeric`.

```
SELECT date,  
       count,
```

```

ROUND((count::numeric/(lag::numeric/100))-100, 2)

FROM(
    SELECT EXTRACT(MONTH FROM creation_date) AS date,
           COUNT(id) AS count,
           LAG(COUNT(id)) OVER () AS lag
    FROM stackoverflow.posts
    WHERE EXTRACT(MONTH FROM creation_date) BETWEEN 9 AND 12
    GROUP BY 1) AS i
;

```

Задача 7

Выгрузите данные активности пользователя, который опубликовал больше всего постов за всё время. Выведите данные за октябрь 2008 года в таком виде:

- номер недели;
- дата и время последнего поста, опубликованного на этой неделе.

```

WITH j AS(
    SELECT EXTRACT(WEEK FROM p.creation_date) AS week,
           p.creation_date AS date
    FROM stackoverflow.users AS u
    JOIN stackoverflow.posts AS p ON u.id = p.user_id
    WHERE u.id IN(
        SELECT id
        FROM(
            SELECT u.id AS id,
                   COUNT(p.id)
            FROM stackoverflow.users AS u
            JOIN stackoverflow.posts AS p ON u.id = p.user_id
            GROUP BY 1
            ORDER BY 2 DESC
            LIMIT 1) AS i)
        AND EXTRACT(MONTH FROM p.creation_date) = 10)

    SELECT DISTINCT week,
           MAX(date) OVER (PARTITION BY week)
    FROM j
    ORDER BY 1
;

```