

Задание 1: Найдите количество вопросов, которые набрали больше 300 очков или как минимум 100 раз были добавлены в «Закладки».

```
SELECT COUNT(p.id)
FROM stackoverflow.posts AS p
JOIN stackoverflow.post_types AS t ON p.post_type_id=t.id
WHERE type = 'Question'
AND (score > 300 OR favorites_count >= 100);
```

Задание 2: Сколько в среднем в день задавали вопросов с 1 по 18 ноября 2008 включительно? Результат округлите до целого числа.

```
SELECT ROUND(AVG(q.questions_cnt))
FROM
  (SELECT DATE_TRUNC('day', creation_date)::date AS day,
    COUNT(p.id) AS questions_cnt
  FROM stackoverflow.posts AS p
  JOIN stackoverflow.post_types AS t ON p.post_type_id=t.id
  WHERE creation_date::date BETWEEN '2008-11-1' AND '2008-11-18'
  AND type='Question'
  GROUP BY 1) AS q;
```

Задание 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 u.creation_date::date = b.creation_date::date;
```

Задание 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';
```

Задание 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 *,
  RANK() OVER(ORDER BY id DESC)
FROM stackoverflow.vote_types
ORDER BY id;
```

Задание 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 u.id AS users,
       COUNT(v.id) AS votes_cnt
FROM stackoverflow.users AS u
JOIN stackoverflow.votes AS v ON u.id=v.user_id
JOIN stackoverflow.vote_types AS t ON v.vote_type_id=t.id
WHERE name = 'Close'
GROUP BY u.id
ORDER BY 2 DESC,
       1 DESC
LIMIT 10;

```

Задание 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 AS users,
       COUNT(b.id) AS badges_cnt,
       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 b.creation_date::date BETWEEN '2008-11-15' AND '2008-12-15'
GROUP BY 1
ORDER BY 2 DESC,
       1
LIMIT 10;

```

Задание 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,
       user_id,
       score,
       ROUND(AVG(score) OVER(PARTITION BY user_id))
FROM stackoverflow.posts
WHERE title IS NOT NULL
AND score <> 0;

```

Задание 9: Отобразите заголовки постов, которые были написаны пользователями, получившими более 1000 значков. Посты без заголовков не должны попасть в список.

```
SELECT title
FROM stackoverflow.posts
WHERE user_id IN (SELECT user_id
                  FROM stackoverflow.badges
                  GROUP BY user_id
                  HAVING COUNT(id) > 1000)
AND title IS NOT NULL;
```

Задание 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 < 350 AND views >= 100 THEN 2
         WHEN views < 100 THEN 3
       END
FROM stackoverflow.users
WHERE views <> 0
AND location LIKE '%United States%';
```

Задание 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 tab AS
(SELECT id,
       views,
       CASE
         WHEN views >= 350 THEN 1
         WHEN views < 350 AND views >= 100 THEN 2
         WHEN views < 100 THEN 3
       END AS groups
FROM stackoverflow.users
WHERE views <> 0
AND location LIKE '%United States%')

SELECT *
FROM tab
```

```
WHERE views IN (SELECT MAX(views)
                FROM tab
                GROUP BY groups)
ORDER BY views DESC,
        id;
```

Задание 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.

```
WITH us AS
(SELECT EXTRACT(DAY FROM creation_date) AS day,
     COUNT(id) AS users_cnt
 FROM stackoverflow.users
 WHERE EXTRACT(MONTH FROM creation_date) = 11
       AND EXTRACT(YEAR FROM creation_date) = 2008
 GROUP BY day)
```

```
SELECT day,
       users_cnt,
       SUM(users_cnt) OVER (ORDER BY day)
FROM us;
```

Задание 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.

```
WITH d AS
(SELECT user_id,
     MIN(creation_date) AS first_date
 FROM stackoverflow.posts
 GROUP BY 1)

SELECT u.id,
     first_date - creation_date AS diff_time
FROM stackoverflow.users AS u
JOIN d ON u.id=d.user_id;
```

Задание 14: 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 DATE_TRUNC('month', creation_date)::date AS month,
     SUM(views_count) AS total_views
FROM stackoverflow.posts
GROUP BY 1
ORDER BY 2 DESC;
```

Задание 15: 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 users  
FROM stackoverflow.users AS u  
JOIN stackoverflow.posts AS p ON u.id=p.user_id  
WHERE p.post_type_id = 2  
      AND DATE_TRUNC('day', p.creation_date) <= DATE_TRUNC('day', u.creation_date) + INTERVAL '1 month'  
GROUP BY 1  
HAVING COUNT(p.id) > 100  
ORDER BY 1;
```

Задание 16: 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.

```
WITH s AS  
(SELECT u.id  
 FROM stackoverflow.users AS u  
 JOIN stackoverflow.posts AS p ON u.id=p.user_id  
 WHERE EXTRACT(MONTH FROM u.creation_date) = 9),  
d AS  
(SELECT user_id,  
       DATE_TRUNC('month', creation_date)::date  
 FROM stackoverflow.posts  
 WHERE EXTRACT(MONTH FROM creation_date) = 12)  
  
SELECT DATE_TRUNC('month', creation_date)::date AS month,  
       COUNT(id) AS posts  
FROM stackoverflow.posts  
WHERE user_id IN (SELECT s.id  
                  FROM s JOIN d ON s.id=d.user_id)  
GROUP BY 1  
ORDER BY 1 DESC;
```

Задание 17: 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 ROWS BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW) AS sum_views
FROM stackoverflow.posts
ORDER BY 1,
2;

```

Задание 18: 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.

```

WITH a AS
(SELECT user_id,
COUNT(DISTINCT creation_date::date) As days
FROM stackoverflow.posts
WHERE creation_date::date BETWEEN '2008-12-1' AND '2008-12-7'
GROUP BY 1)

SELECT ROUND(AVG(days)) AS average
FROM a;

```

Задание 19: 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.

```

WITH t AS
(SELECT EXTRACT(MONTH FROM creation_date) AS month,
COUNT(id) AS posts_cnt
FROM stackoverflow.posts
WHERE creation_date::date BETWEEN '2008-09-01' AND '2008-12-31'
GROUP BY 1)

SELECT *,
ROUND((posts_cnt*100.0/LAG(posts_cnt) OVER(ORDER BY month)) -100, 2) AS percent
FROM t;

```

Задание 20: 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.

```

SELECT DISTINCT
EXTRACT(WEEK FROM creation_date) AS week,
LAST_VALUE(creation_date) OVER(PARTITION BY EXTRACT(WEEK FROM creation_date) ORDER BY
creation_date ROWS BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING)
FROM stackoverflow.posts

```

```
WHERE creation_date::date BETWEEN '2008-10-01' AND '2008-10-31'  
AND user_id IN (SELECT user_id  
                FROM stackoverflow.posts  
                GROUP BY 1  
                ORDER BY COUNT(id) DESC  
                LIMIT 1);
```