

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 p
Join stackoverflow.post_types AS pt ON pt.id = p.post_type_id
Where (favorites_count >=100 OR score > 300) AND pt.type = 'Question'
```

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

```
With cnt_ids AS (
  Select Count(p.id) cnt_id,
    DATE_TRUNC('day', creation_date) :: date AS dates
  From stackoverflow.posts p
  Join stackoverflow.post_types AS pt ON pt.id = p.post_type_id
  Where pt.type = 'Question' AND DATE_TRUNC('day', creation_date) :: date Between '2008-11-01' AND '2008-11-18'
  Group BY dates
)
Select ROUND(AVG(cnt_id))
From cnt_ids
```

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

```
Select Count(Distinct u.id)
From stackoverflow.users u
Join stackoverflow.badges b ON u.id = b.user_id
Where DATE_TRUNC('day', b.creation_date) :: date = DATE_TRUNC('day', u.creation_date) :: date
```

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

```
Select COUNT(Distinct p.id)
From stackoverflow.users AS u
Join stackoverflow.posts AS p ON u.id = p.user_id
Join stackoverflow.votes AS v ON v.post_id = p.id
--- голос v.id - идентификатор голоса. v.post_id - ключ к посту
Where u.display_name = 'Joel Coehoorn'
```

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
```

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 Distinct u.id users,
       Count(v.id) cnt_vote
From stackoverflow.votes v
Join stackoverflow.vote_types vt ON vt.id = v.vote_type_id
Join stackoverflow.users u ON v.user_id = u.id
Where vt.name = 'Close'
Group BY u.id
Order BY cnt_vote DESC, users Desc
Limit 10
```

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

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 p.title AS titles,
       u.id AS users,
       p.score,
       ROUND(AVG(p.score) OVER (PARTITION BY u.id)) AS avg_score
FROM stackoverflow.users AS u
JOIN stackoverflow.posts AS p ON p.user_id = u.id
WHERE p.title notnull AND p.score !=0
```

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 AS p
LEFT JOIN stackoverflow.users AS uu ON p.user_id = uu.id
WHERE p.user_id IN
  (SELECT u.id
   FROM stackoverflow.users AS u
   JOIN stackoverflow.badges AS b ON u.id = b.user_id
   GROUP BY u.id
   HAVING COUNT(DISTINCT b.id)>1000)
AND title IS NOT NULL;
```

10. Напишите запрос, который выгрузит данные о пользователях из Канады (англ. Canada). 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 количеством просмотров меньше либо равным нулю 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 AS group
From stackoverflow.users
WHERE views != 0
      AND location LIKE '%Canada%'
```

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 groups_user AS (
  Select id AS users_id,
        views,
        CASE
          WHEN views >= 350 THEN 1
          WHEN views >= 100
            AND views < 350 THEN 2
          WHEN views < 100 THEN 3
        END AS groups_rang
  FROM stackoverflow.users
```

```

WHERE views != 0
AND location LIKE '%Canada%'),
top_users AS(
SELECT *,
MAX(views) OVER (PARTITION BY groups_rang) AS top_rangs
FROM groups_user
)
Select users_id,
groups_rang,
top_rangs
From top_users
Where views = top_rangs
Order BY 3 Desc, 1

```

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 users_cnt AS (
SELECT
EXTRACT(DAY FROM creation_date ::date) AS reg_day,
COUNT(DISTINCT id) as cnt_users
FROM stackoverflow.users
WHERE DATE_TRUNC('month', creation_date)::DATE = '2008-11-01'
GROUP BY 1
)
SELECT *,
SUM(cnt_users) OVER(ORDER BY reg_day) AS sum_users
FROM users_cnt;

```

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 rangs_data AS(
SELECT *,
ROW_NUMBER() OVER (PARTITION BY user_id ORDER BY creation_date) AS rng_data
FROM stackoverflow.posts
)
Select user_id,
p.creation_date -u.creation_date AS dif_data
From stackoverflow.users u
JOIN rangs_data p ON p.user_id = u.id
WHERE rng_data = '1'

```

14. Выведите общую сумму просмотров у постов, опубликованных в 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)
FROM stackoverflow.posts
Group BY month
Order BY 2 DESC
```

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 as names, count(distinct p.user_id) as cnt
FROM stackoverflow.posts as p
join stackoverflow.users as u on p.user_id=u.id
join stackoverflow.post_types as pt on pt.id=p.post_type_id
WHERE p.creation_date::date BETWEEN u.creation_date::date and (u.creation_date::date +
INTERVAL '1 month') and pt.type like '%Answer%'
GROUP BY u.display_name
having count(p.id) > 100
ORDER BY u.display_name;
```

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.

```
Select DATE_TRUNC('month', creation_date)::date as mon,
       count( user_id)
From stackoverflow.posts
Where user_id in (
    Select u.id
    From stackoverflow.users u
    where DATE_TRUNC('month', u.creation_date)::date ='2008-09-01'
    and
    u.id in (
        Select user_id
        From stackoverflow.posts p
        Where DATE_TRUNC('month', p.creation_date)::date = '2008-12-01'
    )
)
GROUP BY mon
Order by mon desc
```

17. Используя данные о постах, выведите несколько полей:
- идентификатор пользователя, который написал пост;
 - дата создания поста;
 - количество просмотров у текущего поста;
 - сумма 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
```

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 users_day AS (Select Distinct user_id,  
                      (DATE_TRUNC('day', p.creation_date)::date) AS cnt_day  
From stackoverflow.posts p  
Where DATE_TRUNC('day', p.creation_date)::date Between '2008-12-01' AND '2008--12-7'  
---Group BY 1  
),  
cnt AS (  
  Select user_id,  
         COUNT(cnt_day) all_cnt  
  From users_day  
  Group BY 1  
)  
Select ROUND(AVG(all_cnt))  
From cnt
```

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.

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.

```
WITH month_post AS (SELECT EXTRACT(MONTH from creation_date::date) AS month,  
                          COUNT(DISTINCT id)  
FROM stackoverflow.posts  
WHERE creation_date::date BETWEEN '2008-09-01' AND '2008-12-31'  
GROUP BY month)  
SELECT *,  
       ROUND(((count::numeric / LAG(count) OVER (ORDER BY month)) - 1) * 100, 2) AS  
user_growth  
FROM month_post;
```

20. Найдите пользователя, который опубликовал больше всего постов за всё время с момента регистрации. Выведите данные его активности 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,  
MAX(creation_date) OVER (PARTITION BY extract('week' from creation_date)) AS post_dt  
From stackoverflow.posts  
where user_id = 22656 and DATE_TRUNC('month',creation_date::date) = '2008-10-01'  
ORDER BY week
```