

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

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

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

```
SELECT ROUND(AVG (day_cnt))
FROM
(SELECT COUNT(id) AS day_cnt,
creation_date::date AS dt
FROM stackoverflow.posts
WHERE post_type_id = 1
AND creation_date::date BETWEEN '2008-11-01' AND '2008-11-18'
GROUP BY dt) AS d
```

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

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

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

```
SELECT COUNT(cv.id)
FROM (SELECT ps.id
FROM stackoverflow.posts AS ps
JOIN stackoverflow.votes AS v ON ps.id=v.post_id
JOIN stackoverflow.users AS u ON ps.user_id=u.id
WHERE u.display_name LIKE 'Joel Coehoorn' AND v.id > 0
GROUP BY ps.id) AS cv;
```

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 v.user_id,  
       COUNT(vt.id) AS v_cnt  
FROM stackoverflow.votes AS v  
JOIN stackoverflow.vote_types AS vt ON vt.id=v.vote_type_id  
WHERE vt.name LIKE 'Close'  
GROUP BY v.user_id  
ORDER BY v_cnt 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 *,  
       DENSE_RANK() OVER(ORDER BY count DESC)  
FROM(  
  SELECT user_id,  
         COUNT(*)  
  FROM stackoverflow.badges  
  WHERE creation_date::date BETWEEN '2008-11-15' AND '2008-12-15'  
  GROUP BY user_id  
  ORDER BY count DESC, user_id  
  LIMIT 10  
) AS ucnt
```

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
```

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

```
WITH top AS (SELECT DISTINCT user_id,  
                COUNT(id) OVER(PARTITION BY user_id) AS point_count  
FROM stackoverflow.badges  
ORDER BY 2 DESC)  
  
SELECT title  
FROM top  
JOIN stackoverflow.posts AS p ON top.user_id=p.user_id  
WHERE point_count >1000  
AND title IS NOT NULL
```

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

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

```

max AS (
SELECT id,
       category,
       MAX(views) OVER(PARTITION BY category) AS max_cat
FROM category)

```

```

SELECT c.id AS id,
       c.category,
       c.views AS id_views
FROM category AS c JOIN max AS m ON c.id=m.id
WHERE c.views=m.max_cat
ORDER BY id_views DESC, id

```

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 november AS(
SELECT EXTRACT(DAY FROM creation_date::date) AS day,
       COUNT(id) AS day_cnt
FROM stackoverflow.users
WHERE creation_date::date BETWEEN '2008-11-01' AND '2008-11-30'
GROUP BY 1
ORDER BY 1)

```

```

SELECT *,
       SUM(day_cnt) OVER(ORDER BY day) AS days_sum
FROM november

```

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 users_post AS
(SELECT user_id,
       MIN(creation_date) OVER (PARTITION BY user_id) AS first_post
FROM stackoverflow.posts)

```

```

SELECT DISTINCT user_id,
       first_post-creation_date AS diff
FROM users_post AS up
JOIN stackoverflow.users AS u ON u.id = up.user_id

```

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 DISTINCT DATE_TRUNC('month', creation_date)::date AS month,
       SUM(views_count) OVER(PARTITION BY DATE_TRUNC('month', creation_date)) AS
views_count
FROM stackoverflow.posts
WHERE EXTRACT(YEAR FROM creation_date) = 2008
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,  
       COUNT(DISTINCT p.user_id)  
FROM stackoverflow.users AS u  
JOIN stackoverflow.posts AS p ON u.id=p.user_id  
WHERE p.post_type_id=2  
AND p.creation_date::date BETWEEN u.creation_date::date AND (u.creation_date::date  
+ INTERVAL '1 month')  
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.

```
WITH september_users AS  
(SELECT p.user_id  
FROM stackoverflow.users AS u  
JOIN stackoverflow.posts AS p ON u.id=p.user_id  
WHERE u.creation_date::date BETWEEN '2008-09-01' AND '2008-09-30'  
AND p.creation_date::date BETWEEN '2008-12-01' AND '2008-12-31')  
  
SELECT DISTINCT DATE_TRUNC('month', creation_date)::date AS month,  
       COUNT(id) OVER(PARTITION BY DATE_TRUNC('month', creation_date))  
FROM stackoverflow.posts  
WHERE user_id IN (SELECT p.user_id  
                  FROM stackoverflow.users AS u  
                  JOIN stackoverflow.posts AS p ON u.id=p.user_id  
                  WHERE u.creation_date::date BETWEEN '2008-09-01' AND  
'2008-09-30'  
                  AND p.creation_date::date BETWEEN '2008-12-01' AND  
'2008-12-31')  
AND EXTRACT(YEAR FROM creation_date) = 2008  
ORDER BY month DESC
```

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

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 day_cnt AS
(SELECT user_id,
      COUNT (DISTINCT creation_date::date) AS day_cnt
FROM stackoverflow.posts
WHERE creation_date::date BETWEEN '2008-12-01' AND '2008-12-07'
GROUP BY user_id)

SELECT ROUND(AVG(day_cnt))
FROM day_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.

```
WITH month_cnt AS
(SELECT EXTRACT (MONTH FROM creation_date::date) AS month,
      COUNT(id) AS posts_cnt
FROM stackoverflow.posts
WHERE creation_date BETWEEN '2008-09-01' AND '2008-12-31'
GROUP BY month)

SELECT *,
      ROUND(((posts_cnt::numeric / LAG(posts_cnt) OVER ( ORDER BY month)) -1) *100.0,
2)
FROM month_cnt
```

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.

```
WITH top_user AS
(SELECT user_id,
      COUNT(id) AS post_cnt
FROM stackoverflow.posts
GROUP BY user_id
ORDER BY post_cnt DESC
LIMIT 1)

SELECT EXTRACT(WEEK FROM creation_date) AS week,
      MAX(creation_date)
FROM top_user AS tu
JOIN stackoverflow.posts AS p ON tu.user_id=p.user_id
WHERE creation_date::date BETWEEN '2008-10-01' AND '2008-10-31'
GROUP BY week
```