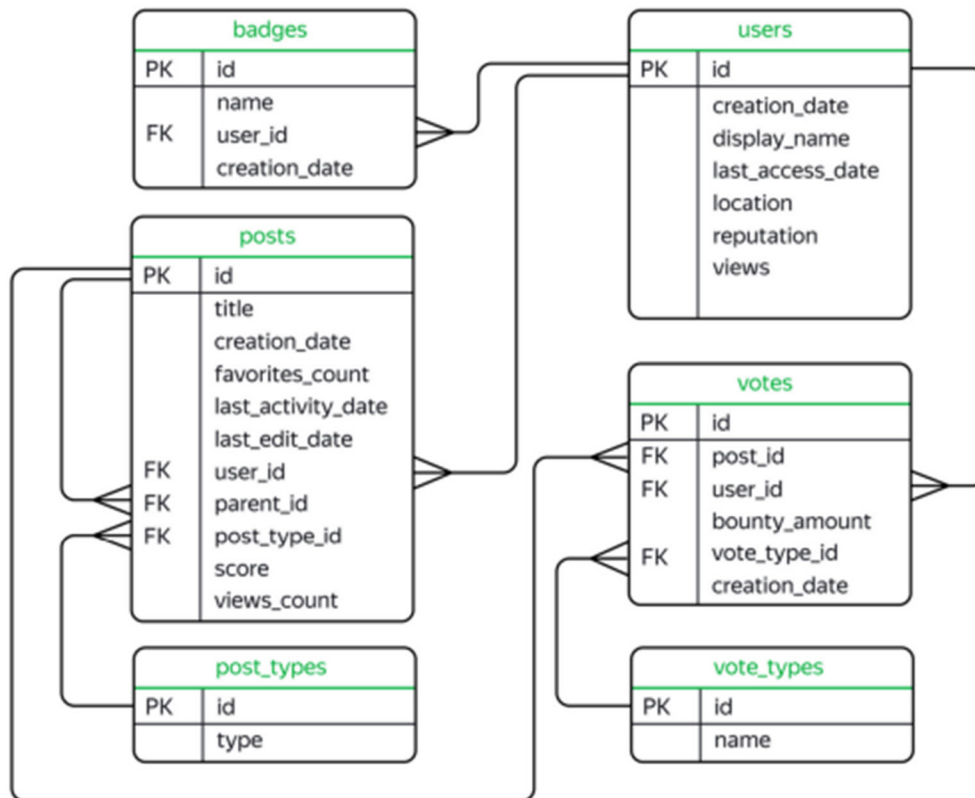


Реляционная база данных StackOverflow с информацией о постах, вопросах, ответах, комментариях и оценках.



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

```

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

2. Сколько в среднем в день задавали вопросов с 1 по 18 ноября 2008 включительно?

```

WITH questions AS(
    
```

```

SELECT COUNT(*) as q_a
FROM stackoverflow.posts
WHERE post_type_id IN(
    SELECT id
    FROM stackoverflow.post_types
    WHERE type = 'Question')
AND CAST(creation_date AS date) BETWEEN '2008-11-1' AND '2008-11-18'
GROUP BY CAST(creation_date AS date)
)
SELECT ROUND(AVG(q_a))
FROM questions;

```

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

```

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

```

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

```

SELECT COUNT(DISTINCT title)
FROM stackoverflow.posts p
JOIN stackoverflow.users u ON p.user_id = u.id
JOIN stackoverflow.votes v ON u.id = v.user_id
WHERE u.display_name = 'Joel Coehoorn'
AND p.id IN (SELECT post_id
    FROM stackoverflow.votes);

```

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

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

6. Отобрать 10 пользователей, которые поставили больше всего голосов типа Close. Отобразить таблицу из двух полей: идентификатором пользователя и количеством голосов. Отсортировать данные сначала по убыванию количества голосов, потом по убыванию значения идентификатора пользователя.

```
SELECT u.id,  
    COUNT(vt.id)  
FROM stackoverflow.vote_types vt  
JOIN stackoverflow.votes v 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 COUNT(vt.id) DESC, u.id DESC  
LIMIT 10;
```

7. Отобрать 10 пользователей по количеству значков, полученных в период с 15 ноября по 15 декабря 2008 года включительно.

Отобразить несколько полей:

идентификатор пользователя;

число значков;

место в рейтинге — чем больше значков, тем выше рейтинг.

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

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

```
WITH cnt_badges AS(  
    SELECT user_id,  
        COUNT(id)
```

```

FROM stackoverflow.badges
WHERE CAST(creation_date AS date) BETWEEN '2008-11-15' AND '2008-12-15'
GROUP BY user_id)
SELECT *,
    DENSE_RANK() OVER(ORDER BY count DESC)
FROM cnt_badges
LIMIT 10;

```

8. Сколько в среднем очков получает пост каждого пользователя?

Сформировать таблицу из следующих полей:

заголовок поста;

идентификатор пользователя;

число очков поста;

среднее число очков пользователя за пост, округлённое до целого числа.

Не учитывать 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. Отобразить 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(name) > 1000)

AND title IS NOT null;

10. Написать запрос, который выгрузит данные о пользователях из США (англ. United States).

Разделить пользователей на три группы в зависимости от количества просмотров их профилей:

пользователям с числом просмотров больше либо равным 350 присвоить группу 1;

пользователям с числом просмотров меньше 350, но больше либо равно 100 — группу 2;

пользователям с числом просмотров меньше 100 — группу 3.

Отобразить 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 location LIKE '%United States%'
AND views !=0;
```

11. Дополнить предыдущий запрос. Отобразить лидеров каждой группы — пользователей, которые набрали максимальное число просмотров в своей группе. Вывести поля с идентификатором пользователя, группой и количеством просмотров. Отсортировать таблицу по убыванию просмотров, а затем по возрастанию значения идентификатора.

```
WITH t1 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 grp
    FROM stackoverflow.users
    WHERE location LIKE ('%United States%') AND views !=0),
t2 AS (
    SELECT grp,
        MAX(views) AS max
    FROM t1
    GROUP BY grp)
SELECT t1.id,
    t2.grp,
    t2.max
FROM t1
JOIN t2 ON t2.grp = t1.grp AND t1.views = t2.max
ORDER BY t2.max DESC, t1.id;

```

12. Посчитать ежедневный прирост новых пользователей в ноябре 2008 года. Сформировать таблицу с полями:

номер дня;

число пользователей, зарегистрированных в этот день;

сумму пользователей с накоплением.

```

WITH cnt_day AS(
    SELECT EXTRACT(DAY FROM creation_date :: date) AS day,
        COUNT (id)
    FROM stackoverflow.users
    WHERE creation_date :: date BETWEEN '2008-11-01' AND '2008-11-30'
    GROUP BY day)
SELECT *,
    SUM(count) OVER(ORDER BY day)
FROM cnt_day;

```

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

идентификатор пользователя;

разницу во времени между регистрацией и первым постом.

```
WITH p AS (  
    SELECT user_id,  
           MIN(creation_date) as first_post_date  
    FROM stackoverflow.posts  
    GROUP BY user_id)  
SELECT p.user_id,  
       p.first_post_date - creation_date  
FROM p  
JOIN stackoverflow.users u on p.user_id = u.id;
```

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

```
SELECT DATE_TRUNC('month', creation_date)::date AS dt,  
       SUM(views_count) AS vc  
FROM stackoverflow.posts  
WHERE CAST (creation_date AS DATE) BETWEEN '2008-01-01' AND '2008-12-31'  
GROUP BY 1  
ORDER BY vc DESC;
```

15. Вывести имена самых активных пользователей, которые в первый месяц после регистрации (включая день регистрации) дали больше 100 ответов. Вопросы, которые задавали пользователи, не учитывать. Для каждого имени пользователя вывести количество уникальных значений user_id. Отсортировать результат по полю с именами в лексикографическом порядке.

```
SELECT display_name,  
       COUNT(DISTINCT p.user_id) AS total_answerss  
FROM stackoverflow.users AS u  
JOIN stackoverflow.posts AS p ON u.id=p.user_id  
WHERE post_type_id=2
```

```
AND p.creation_date :: date <= u.creation_date :: date + INTERVAL '1 month'

GROUP BY display_name

HAVING COUNT(DISTINCT p.id) > 100;
```

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

```
SELECT DATE_TRUNC('month', creation_date) :: date AS month,
       COUNT (id) AS num_of_posts
FROM stackoverflow.posts
WHERE user_id IN (
    SELECT DISTINCT u.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')
GROUP BY 1
ORDER BY 1 DESC;
```

17. Используя данные о постах, вывести 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;
```


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

```
WITH days AS (  
    SELECT user_id,  
           COUNT(DISTINCT creation_date :: date)  
    FROM stackoverflow.posts  
    WHERE creation_date :: date BETWEEN '2008-12-01' AND '2008-12-07'  
    GROUP BY user_id)  
SELECT ROUND(AVG(count))  
FROM days;
```

19. На сколько процентов менялось количество постов ежемесячно с 1 сентября по 31 декабря 2008 года? Отобразить 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. Округлить значение процента до двух знаков после запятой.

```
WITH table1 AS (  
    SELECT EXTRACT(MONTH FROM creation_date :: date) AS month,  
           COUNT(id) AS number_cm  
    FROM stackoverflow.posts  
    WHERE creation_date :: date BETWEEN '2008-09-01' AND '2008-12-31'  
    GROUP BY month  
)  
SELECT *,  
       ROUND(((number_cm :: numeric) / LAG(number_cm) OVER(ORDER BY month)-1)*100, 2) AS dif  
FROM table1;
```

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

номер недели;

дата и время последнего поста, опубликованного на этой неделе.

```
WITH posting_user AS (  
    SELECT user_id,  
           COUNT(id)  
    FROM stackoverflow.posts  
    GROUP BY user_id  
    ORDER BY 2 DESC  
    LIMIT 1)  
SELECT EXTRACT(WEEK FROM creation_date ::date) AS week_number,  
       MAX(creation_date)  
FROM stackoverflow.posts  
WHERE CAST(DATE_TRUNC('month', creation_date) AS date) = '2008-10-01'  
AND user_id IN (SELECT user_id  
                FROM posting_user  
                )  
GROUP BY week_number;
```