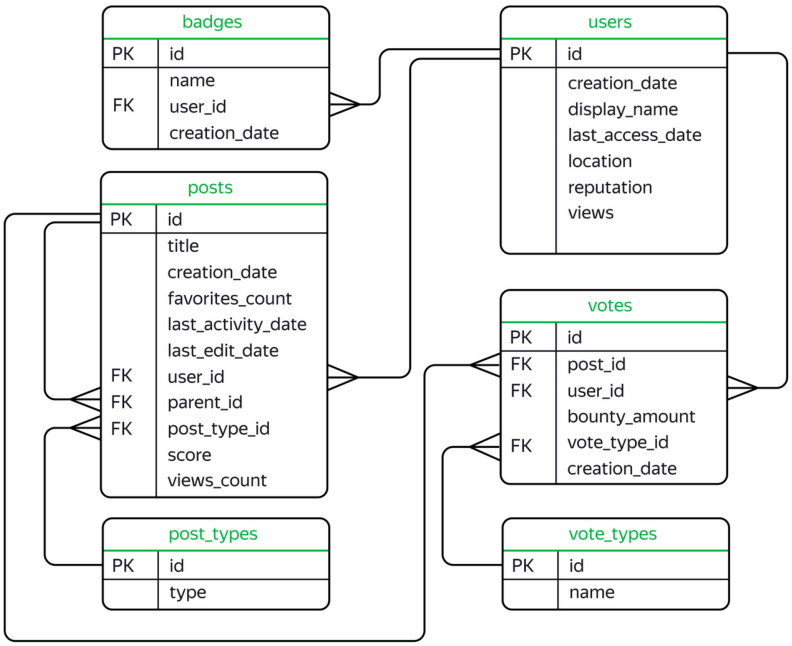
Продвинутый SQL



В самостоятельном проекте вы будете работать с базой данных StackOverflow — сервиса вопросов и ответов о программировании. StackOverflow похож на социальную сеть — пользователи сервиса задают вопросы, отвечают на посты, оставляют комментарии и ставят оценки другим ответам.

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

```
SELECT COUNT(post_type_id)
FROM stackoverflow.posts
WHERE post_type_id = 1 AND
(score > 300
OR 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 (a.cnt))
FROM
(SELECT DATE_TRUNC('DAY', creation_date)::DATE,
COUNT (id) as cnt
FROM stackoverflow.posts
WHERE (DATE_TRUNC('DAY', creation_date)::DATE BETWEEN '2008-11-01' AND '2008-11-18') and post_type_id = 1
GROUP BY DATE_TRUNC('DAY', creation_date)::DATE) as a;
```

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

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

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

```
select count( distinct id_p)
from (select distinct p.id as id_p,
v.id as vot_cnt
from stackoverflow.posts AS p
full join stackoverflow.users AS u ON p.user_id=u.id
full join stackoverflow.votes AS v ON p.id=v.post_id
where u.display_name='Joel Coehoorn'
group by 1,2) as a
where vot_cnt >= 1;
```

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 *,
ROW_NUMBER() OVER (ORDER BY id DESC) as rank
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(v.id) as count_votes
FROM stackoverflow.votes v
JOIN stackoverflow.vote_types vt ON v.vote_type_id = vt.id
WHERE vt.name = 'Close'
GROUP BY v.user_id
ORDER BY count_votes DESC,
v.user_id 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 user_id,
COUNT(id) as count_badges,
DENSE_RANK() OVER (ORDER BY COUNT(id)DESC) AS rating
FROM stackoverflow.badges
WHERE creation_date::date BETWEEN '2008-11-15' AND '2008-12-15'
GROUP BY user_id
ORDER BY count_badges DESC, user_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 title,
user_id,
score,
ROUND(AVG(score) OVER (PARTITION BY user_id)) as avg_score
```

```
FROM stackoverflow.posts
WHERE title IS NOT NULL
AND score!=0
GROUP BY title, user_id,score;
```

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
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 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
ELSE 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.

```
SELECT id,
group_number,
views
FROM
(SELECT *,
MAX(views) OVER (PARTITION BY group_number) AS max_value
FROM
(SELECT id,
views,
CASE
WHEN views >=350 THEN 1
WHEN views < 350 AND views>=100 THEN 2
ELSE 3
END AS group_number
FROM stackoverflow.users
WHERE views!=0
AND location LIKE '%United States%') AS a) AS b
WHERE max_value = views
ORDER BY 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 users_1 AS (SELECT EXTRACT(DAY FROM creation_date) AS day_reg,
COUNT(id) AS count_id
FROM stackoverflow.users u
WHERE EXTRACT(MONTH FROM creation_date) = 11
AND EXTRACT(YEAR FROM creation_date) = 2008
GROUP BY day_reg)
SELECT *,
SUM(count_id) OVER (ORDER BY day_reg)
FROM users_1;
```

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 intervall AS (SELECT DISTINCT u.id,
p.creation_date - u.creation_date as difference
FROM stackoverflow.users u
JOIN stackoverflow.posts p ON u.id = p.user_id)
SELECT id,
MIN(difference)
FROM intervall
```

```
GROUP BY 1
ORDER BY 1;
```

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)
FROM stackoverflow.posts
WHERE EXTRACT(YEAR FROM creation_date)=2008
GROUP BY DATE_TRUNC('month',creation_date)
ORDER BY sum 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 user_id)
FROM stackoverflow.users u
JOIN stackoverflow.posts p ON u.id = p.user_id
JOIN stackoverflow.post_types pt ON p.post_type_id = pt.id
WHERE DATE_TRUNC('day', p.creation_date) >= DATE_TRUNC('day', u.creation_date)
AND DATE_TRUNC('day', p.creation_date) <= DATE_TRUNC('day', u.creation_date) + INTERVAL '1 month'
AND pt.type = 'Answer'
GROUP BY u.display_name
HAVING COUNT(*) > 100
ORDER BY 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 b AS (SELECT u.id as users_id
FROM stackoverflow.users u
JOIN stackoverflow.posts p ON u.id=p.user_id
WHERE DATE_TRUNC('day', u.creation_date)::date BETWEEN '2008-09-01' AND '2008-09-30'
AND DATE_TRUNC('day', p.creation_date)::date BETWEEN '2008-12-01' AND '2008-12-31')
SELECT DISTINCT COUNT(p.id) OVER (PARTITION BY DATE_TRUNC('month', p.creation_date)),
DATE_TRUNC('month', p.creation_date)::date as months
FROM b
JOIN stackoverflow.posts p ON b.users_id=p.user_id
GROUP BY p.creation_date,
p.id
ORDER BY months 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,
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 b AS
(SELECT distinct user_id,
DATE_TRUNC('day', creation_date) AS day
FROM stackoverflow.posts
WHERE DATE_TRUNC('day', creation_date) BETWEEN '2008-12-01' AND '2008-12-07')
SELECT ROUND(AVG(days_count)) AS count_days
FROM
(SELECT user_id,
COUNT(day) AS days_count
FROM b
GROUP BY user_id
ORDER BY user_id) AS b2;
```

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 b as
(SELECT EXTRACT(MONTH FROM creation_date) months,
COUNT(id) posts_count
FROM stackoverflow.posts
WHERE creation_date BETWEEN '2008-09-01' AND '2008-12-31'
GROUP BY months)
SELECT *,
ROUND(CAST((posts_count - LAG(posts_count)
OVER (ORDER BY months)) AS numeric) / (LAG(posts_count)
OVER (ORDER BY months))*100, 2) perc
FROM b;
```

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 dt AS
(SELECT EXTRACT(WEEK FROM creation_date) week,
creation_date
FROM stackoverflow.posts
WHERE creation_date::timestamp BETWEEN '2008-10-01 00:00:00' AND '2008-10-31 23:59:59'
AND user_id IN
(SELECT user_id
FROM stackoverflow.posts
GROUP BY  user_id
ORDER BY  count(id) DESC
LIMIT 1) )
SELECT DISTINCT week,
LAST_VALUE(creation_date) OVER (PARTITION BY week ORDER BY week)
FROM dt
ORDER BY week;
```