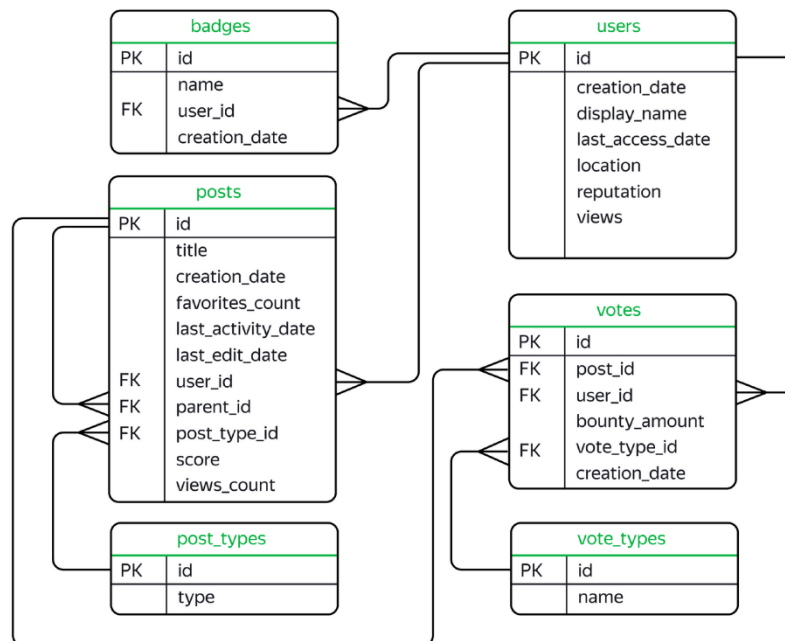


Проект состоит из SQL запросов к базе данных StackOverflow — сервиса вопросов и ответов о программировании. StackOverflow похож на социальную сеть — пользователи сервиса задают вопросы, отвечают на посты, оставляют комментарии и ставят оценки другим ответам. Запросы направлены к той версии базы, в которой хранятся данные о постах за 2008 год, но в таблицах можно найти информацию и о более поздних оценках, которые эти посты получили.

ER-диаграмма и подробное описание к ней приложено в файле `stackoverflow_sql_description.pdf`



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

```
select count(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(posts_cnt)) from
(select creation_date::date as dt, count(id) as posts_cnt
from stackoverflow.posts
where post_type_id = 1 and creation_date between '2008-11-1' and '2008-11-19'
group by dt
order by dt) as a

```

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

```

select count(distinct sb.user_id)
from stackoverflow.badges as sb
join stackoverflow.users as su on sb.user_id=su.id
where sb.creation_date::date=su.creation_date::date

```

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

```

with name as (select id
from stackoverflow.users
where display_name = 'Joel Coehoorn')
select count(distinct posts.id)
from (select sp.id
from stackoverflow.posts as sp
join name on sp.user_id=name.id) as posts
join stackoverflow.votes as sv on posts.id=sv.post_id

```

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 user_id, count(id) as votes
from stackoverflow.votes
where vote_type_id = 6
group by user_id
order by votes desc, 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 *, dense_rank() over(order by badges_cnt desc)
from(
select user_id, count(id) as badges_cnt
from stackoverflow.badges
where creation_date::date between '2008-11-15' and '2008-12-15'
group by user_id
order by badges_cnt desc) as a
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,  
       p.user_id,  
       p.score,  
       round(avg(p.score) over (partition by p.user_id))  
from stackoverflow.posts as p  
--inner join stackoverflow.users as u on p.user_id=u.id  
where p.score != 0  
and p.title != "
```

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

```
with popular as  
(select user_id from (select distinct user_id, count(id) as badges  
from stackoverflow.badges  
group by user_id) as a  
where badges>1000)  
select title  
from stackoverflow.posts as sp  
join popular on sp.user_id=popular.user_id  
where title != "
```

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
  else 3
end as view_group
from stackoverflow.users as u
where location like '%united states%' and views > 0;

```

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 s as
(select id,
  views,
  case
    when views >= 350 then 1
    when views >= 100 then 2
    else 3
  end as view_group,
  max(views) over (partition by case
                    when views >= 350 then 1
                    when views >= 100 then 2
                    else 3
                  end) as max_group
from stackoverflow.users as u
where location like '%united states%' and views > 0
order by 2 desc,
       3 asc)
select id,
  views,

```

```

view_group
from s
where views = max_group
group by 1, 2, 3
order by 2 desc,
        3 asc

```

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 s as
(select extract(day from creation_date) as dt,
        id,
        count(id)
        sum(id) over (order by extract(isodow from creation_date)) as sum_users
from stackoverflow.users
where creation_date between '2008-11-01' and '2008-12-01'
group by 1)
select dt,
        count,
        sum(count) over (order by dt) as sum_users
from s

```

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 sec as
(select u.id,
        min(u.creation_date) as dt_reg,
        min(p.creation_date) as dt_pos

```

```

--first_value(p.creation_date) over (order by u.id) - u.creation_date
from stackoverflow.users as u
inner join stackoverflow.posts as p on u.id=p.user_id
group by u.id)
select id,
       dt_pos - dt_reg
from sec

```

1. 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 cast(date_trunc('month', creation_date) as date) as month,
sum(views_count) as views -- over(partition by extract(month from creation_date)) as
views
from stackoverflow.posts
where extract(year from creation_date) = 2008
group by month
order by views desc

```

2. 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.posts p
join stackoverflow.post_types pt on p.post_type_id = pt.id
join stackoverflow.users u on u.id = p.user_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;
```

3. 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 month,
       count(*)
from stackoverflow.posts
where user_id in (select distinct u.id
                  from stackoverflow.users u
                  join stackoverflow.posts p on u.id = p.user_id
                  where date_trunc('month', u.creation_date) = '2008-09-01'
                  and date_trunc('month', p.creation_date) = '2008-12-01')
group by month
order by month desc;
```

4. 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.

```
with se as
(select user_id,
       cast(date_trunc('day', creation_date) as date) as dt,
       count(cast(date_trunc('day', creation_date) as date))
from stackoverflow.posts
where creation_date::date between '2008-12-01' and '2008-12-07'
group by 1, 2)
select round(avg(count))
```


from se

5. 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 sec as

```
(select extract(month from creation_date) as dt,
```

```
count(views_count)
```

```
from stackoverflow.posts
```

```
where cast(date_trunc('month', creation_date) as date) between '2008-09-01' and '2008-12-01'
```

```
group by 1)
```

```
select *,
```

```
round((((count::numeric / lag(count) over (order by dt)-1)) * 100, 2)
```

```
from sec
```

6. 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 a as (

```
select p.user_id,
```

```
count(p.id) as cnt
```

```
from stackoverflow.posts as p
```

```
group by p.user_id
```

order by cnt desc

limit 1)

```
select distinct extract(week from p.creation_date) as week,  
max(p.creation_date) over (order by extract(week from p.creation_date))  
from a  
join stackoverflow.posts as p on p.user_id = a.user_id  
where extract(month from p.creation_date) = 10  
and extract(year from p.creation_date) = 2008
```