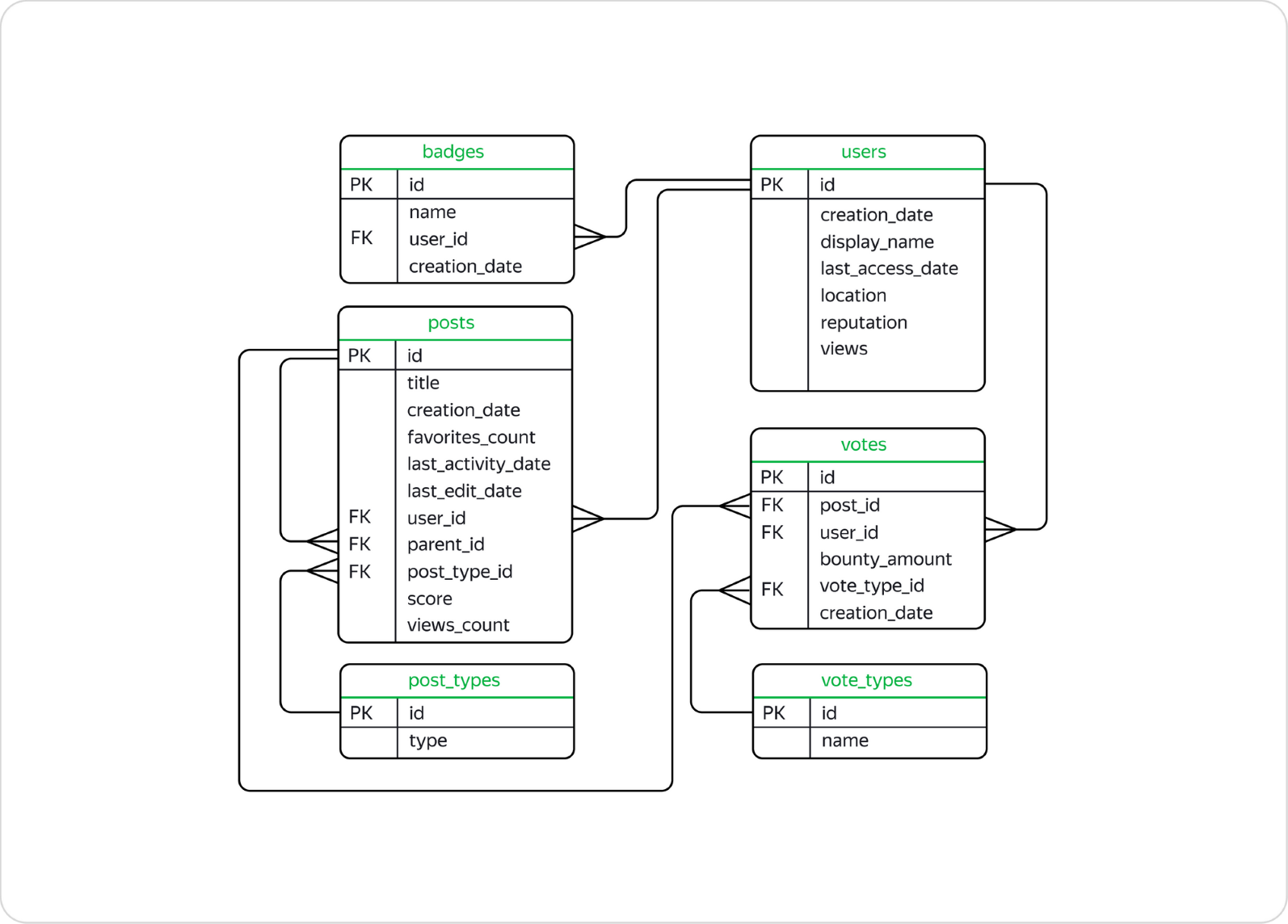
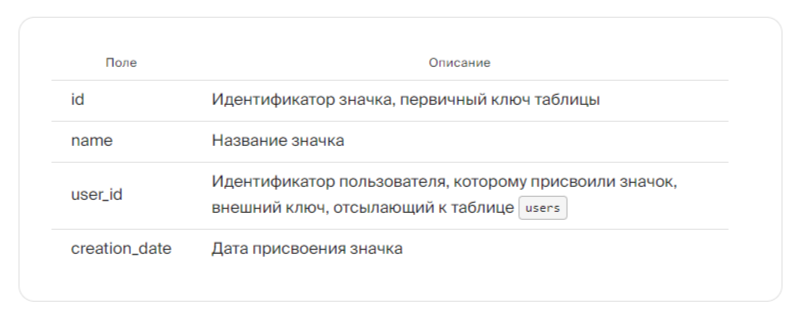
**Описание данных**



**Таблица stackoverflow.badges**

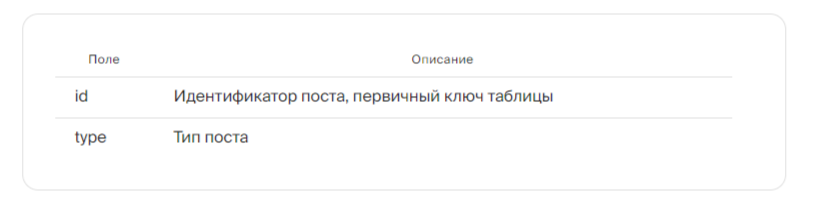
Хранит информацию о значках, которые присуждаются за разные достижения. Например, пользователь, правильно ответивший на большое количество вопросов про PostgreSQL, может получить значок postgresql.



**Таблица stackoverflow.post\_types**

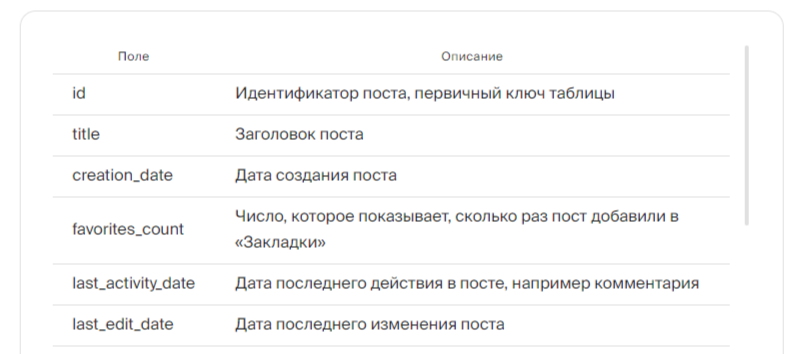
Содержит информацию о типе постов. Их может быть два:

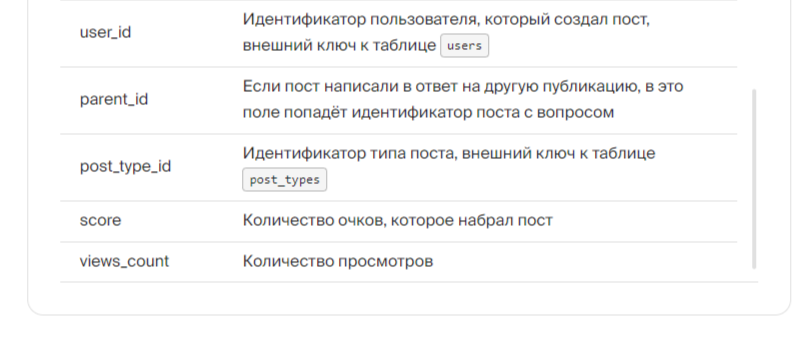
* Question — пост с вопросом;
* Answer — пост с ответом.



**Таблица stackoverflow.posts**

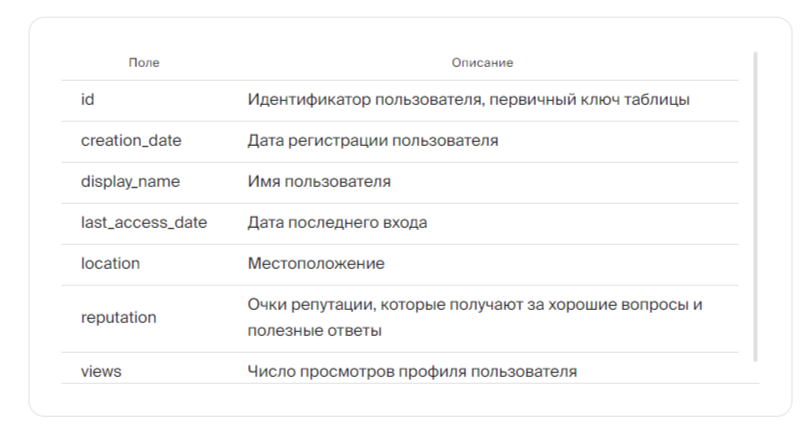
Содержит информацию о постах.





**Таблица stackoverflow.users**

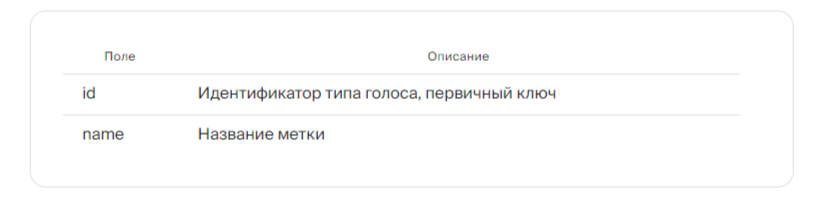
Содержит информацию о пользователях.



**Таблица stackoverflow.vote\_types**

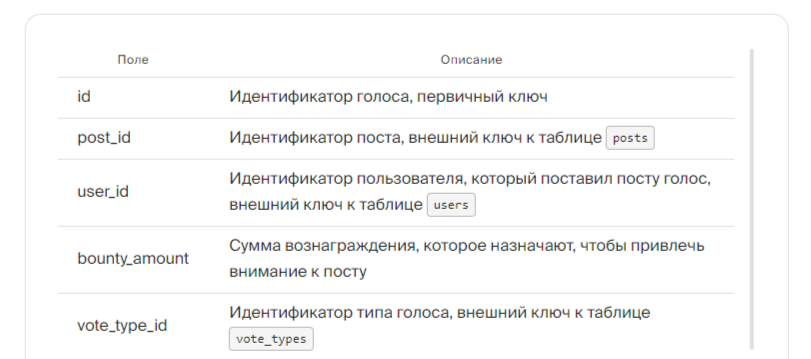
Содержит информацию о типах голосов. Голос — это метка, которую пользователи ставят посту. Типов бывает несколько:

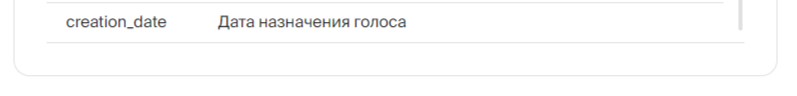
* UpMod — такую отметку получают посты с вопросами или ответами, которые пользователи посчитали уместными и полезными.
* DownMod — такую отметку получают посты, которые показались пользователям наименее полезными.
* Close — такую метку ставят опытные пользователи сервиса, если заданный вопрос нужно доработать или он вообще не подходит для платформы.
* Offensive — такую метку могут поставить, если пользователь ответил на вопрос в грубой и оскорбительной манере, например, указав на неопытность автора поста.
* Spam — такую метку ставят в случае, если пост пользователя выглядит откровенной рекламой.



**Таблица stackoverflow.votes**

Содержит информацию о голосах за посты.



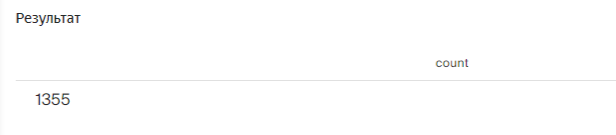


1. Найдите количество вопросов, которые набрали больше 300 очков или как минимум 100 раз были добавлены в «Закладки».

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(q.count), 0)

FROM (

SELECT COUNT(id),

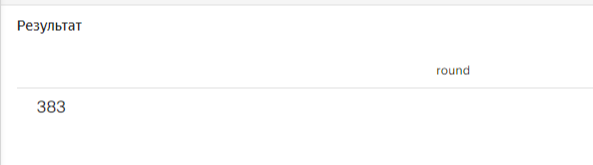
creation\_date::date

FROM stackoverflow.posts

WHERE post\_type\_id = 1

GROUP BY creation\_date::date

HAVING creation\_date::date BETWEEN '2008-11-01' AND '2008-11-18') AS q;



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

SELECT COUNT(DISTINCT b.user\_id)

FROM stackoverflow.badges AS b

JOIN stackoverflow.users AS u 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(cv.id)

FROM (SELECT ps.id

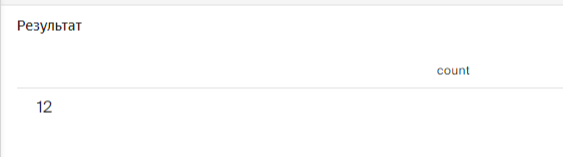
FROM stackoverflow.posts AS ps

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;



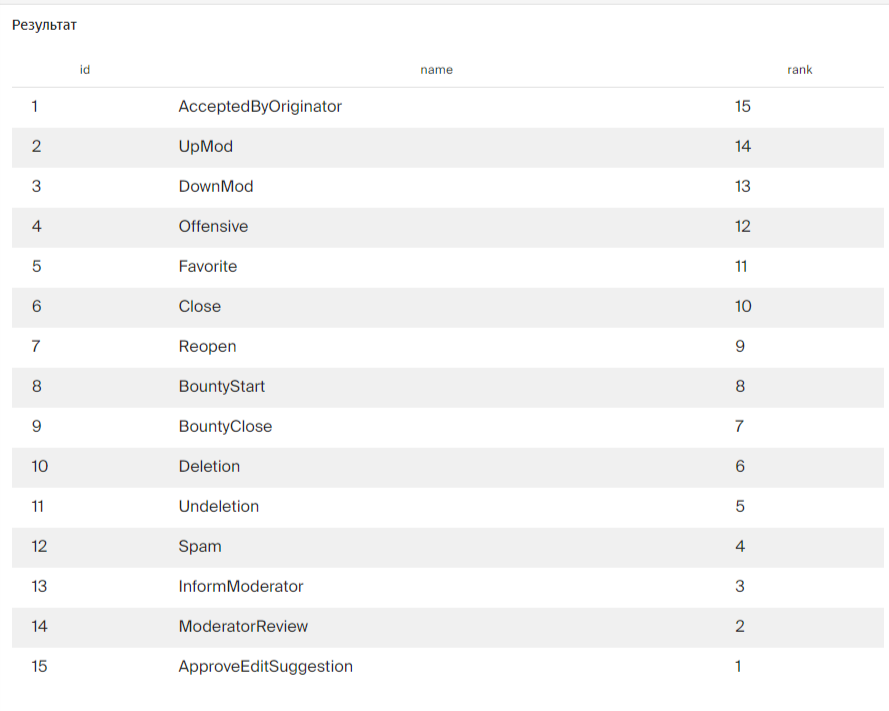
5. Выгрузите все поля таблицы vote\_types. Добавьте к таблице поле rank, в которое войдут номера записей в обратном порядке. Таблица должна быть отсортирована по полю id.

SELECT \*,

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 \*

FROM (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

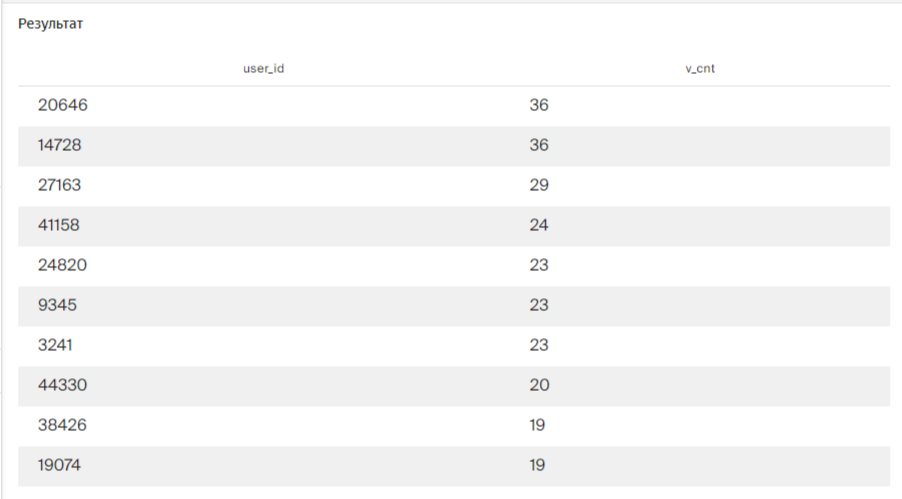
WHERE vt.name LIKE 'Close'

GROUP BY v.user\_id

ORDER BY v\_cnt DESC

LIMIT 10) AS au

ORDER BY au.v\_cnt DESC, au.user\_id DESC;



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 b.b\_cnt DESC) AS rating

FROM (SELECT user\_id,

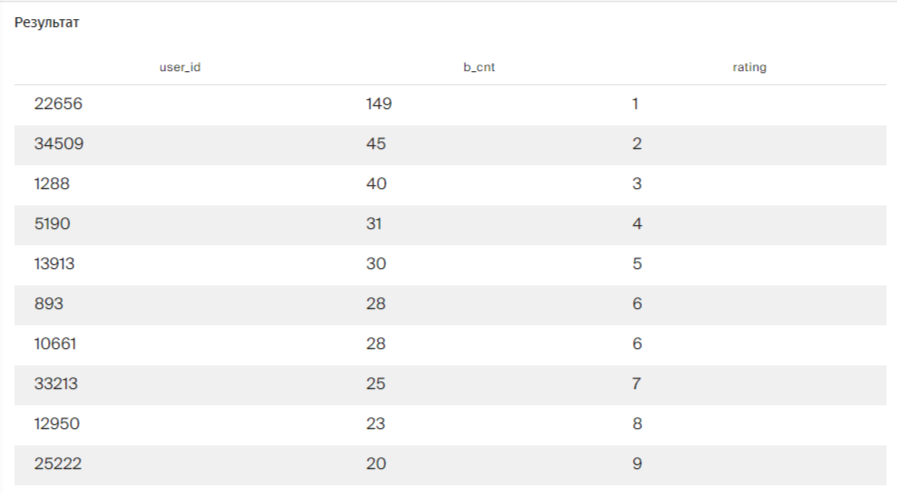
COUNT(id) AS b\_cnt

FROM stackoverflow.badges

WHERE creation\_date::date BETWEEN '2008-11-15' AND '2008-12-15'

GROUP BY user\_id

ORDER BY b\_cnt DESC, user\_id LIMIT 10) as b;



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.

WITH sc AS (

SELECT ROUND(AVG(score)) AS avg\_score,

user\_id

FROM stackoverflow.posts

WHERE title IS NOT NULL AND score <> 0

GROUP BY user\_id)

SELECT p.title,

sc.user\_id,

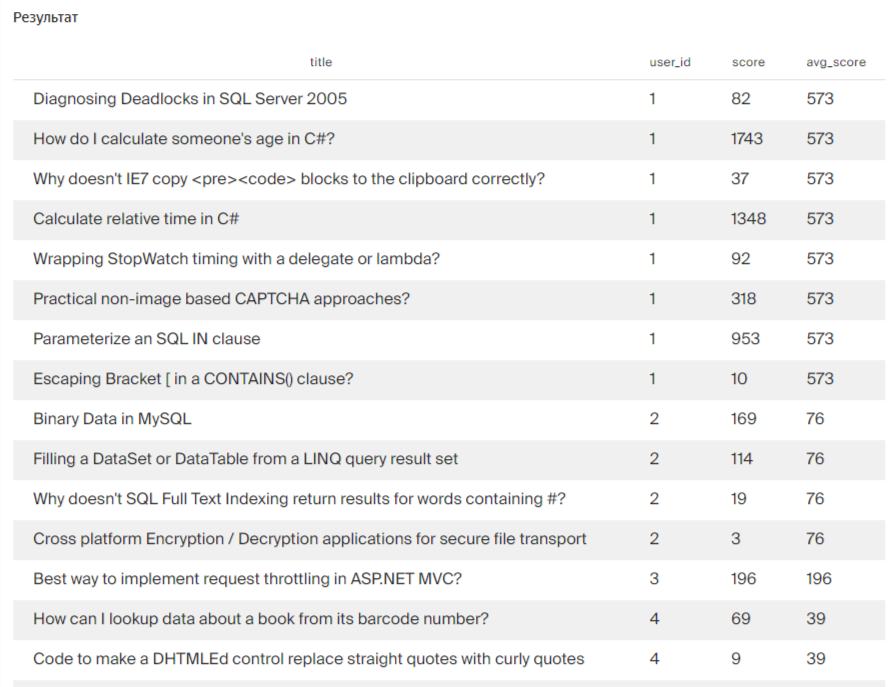
p.score,

sc.avg\_score

FROM sc

JOIN stackoverflow.posts AS p ON sc.user\_id=p.user\_id

WHERE p.title IS NOT NULL 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

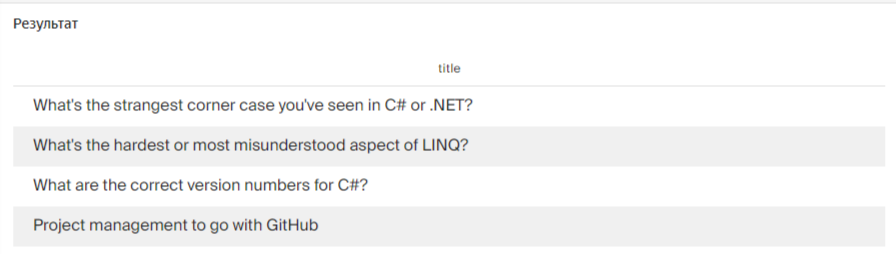
WHERE user\_id IN (SELECT user\_id

FROM stackoverflow.badges

GROUP BY user\_id

HAVING COUNT(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

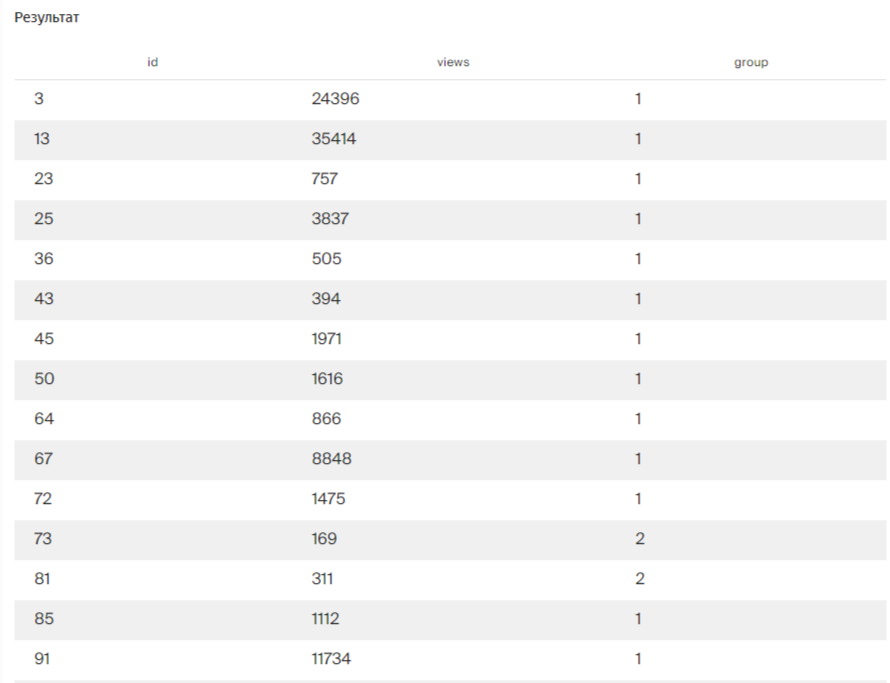
WHEN views < 100 THEN 3

ELSE 2

END AS group

FROM stackoverflow.users

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 grp AS (SELECT g.id,

g.views,

g.group,

MAX(g.views) OVER (PARTITION BY g.group) AS max

FROM (SELECT id,

views,

CASE

WHEN views >= 350 THEN 1

WHEN views < 100 THEN 3

ELSE 2

END AS group

FROM stackoverflow.users

WHERE location LIKE '%United States%' AND views > 0) as g

)

SELECT grp.id,

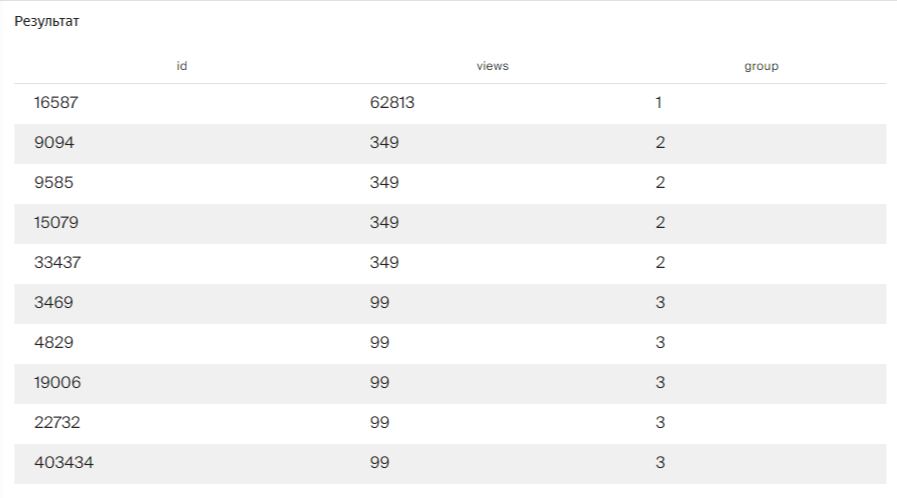
grp.views,

grp.group

FROM grp

WHERE grp.views = grp.max

ORDER BY grp.views DESC, grp.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.

SELECT \*,

SUM(uc.cnt\_reg) OVER (ORDER BY uc.days) as cnt\_users

FROM (SELECT EXTRACT(DAY FROM creation\_date::date) AS days,

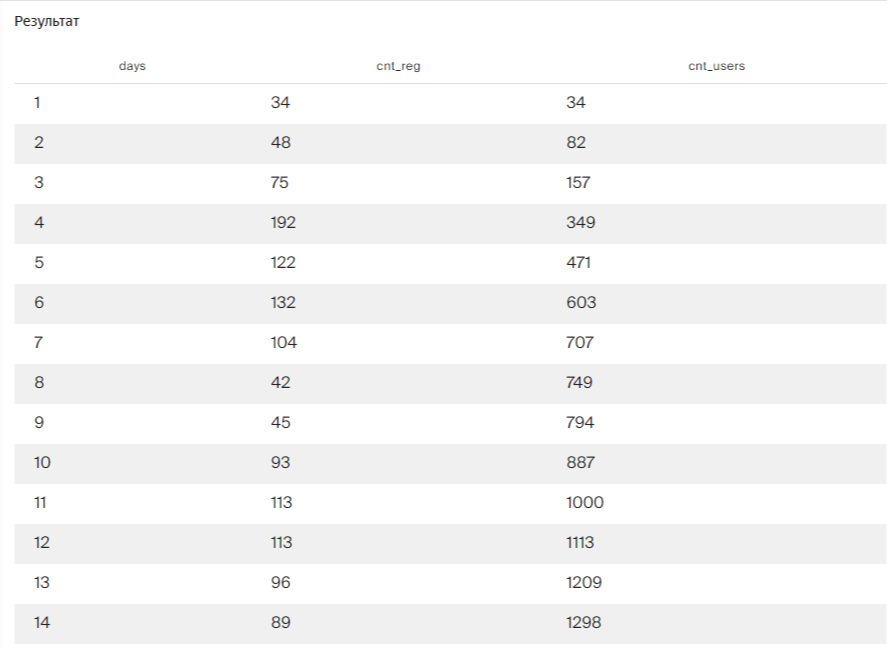
COUNT(id) AS cnt\_reg

FROM stackoverflow.users

WHERE creation\_date::date BETWEEN '2008-11-01' AND '2008-11-30'

GROUP BY EXTRACT(DAY FROM creation\_date::date)

) as uc;



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 dt AS (SELECT DISTINCT user\_id,

MIN(creation\_date) OVER (PARTITION BY user\_id) AS min\_dt

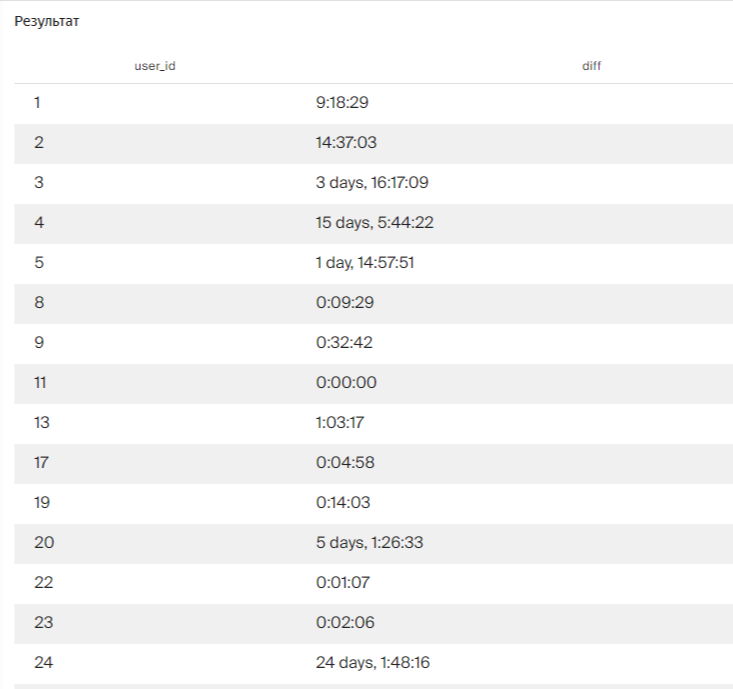
FROM stackoverflow.posts)

SELECT dt.user\_id,

(dt.min\_dt - u.creation\_date) AS diff

FROM stackoverflow.users AS u

JOIN dt ON u.id = dt.user\_id;



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.

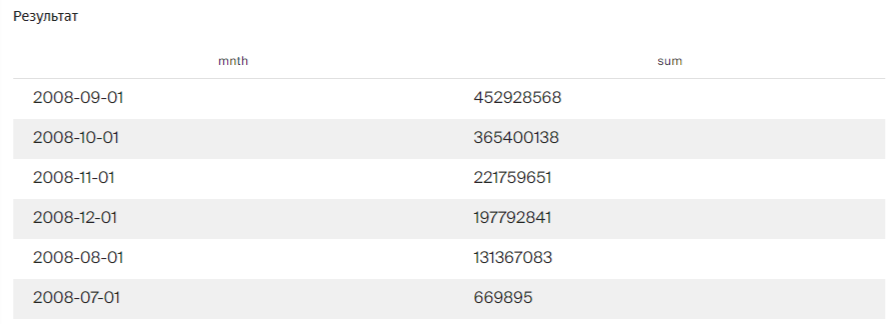
SELECT DATE\_TRUNC('month', creation\_date)::date AS mnth,

SUM(views\_count)

FROM stackoverflow.posts

GROUP BY DATE\_TRUNC('month', creation\_date)

ORDER BY SUM(views\_count) 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 p.user\_id)

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

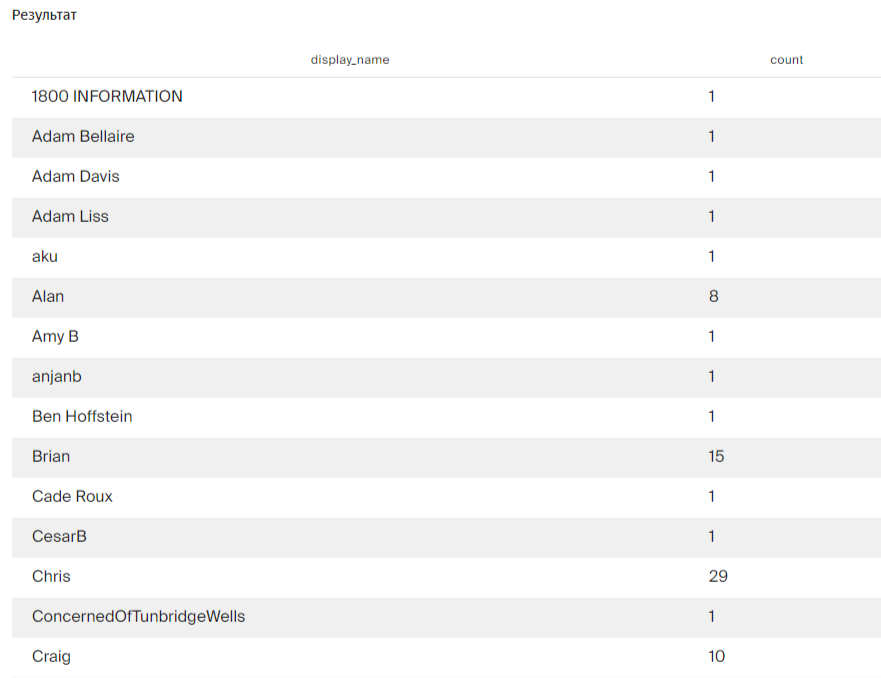
WHERE p.creation\_date::date BETWEEN u.creation\_date::date AND (u.creation\_date::date + INTERVAL '1 month')

AND pt.type LIKE '%Answer%'

GROUP BY u.display\_name

HAVING COUNT(p.id) > 100

ORDER BY u.display\_name;



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.

WITH users AS (SELECT u.id

FROM stackoverflow.posts AS p

JOIN stackoverflow.users AS u ON p.user\_id=u.id

WHERE DATE\_TRUNC('month', u.creation\_date)::date = '2008-09-01'

AND DATE\_TRUNC('month', p.creation\_date)::date = '2008-12-01'

GROUP BY u.id

HAVING COUNT(p.id) > 0)

SELECT COUNT(p.id),

DATE\_TRUNC('month', p.creation\_date)::date

FROM stackoverflow.posts AS p

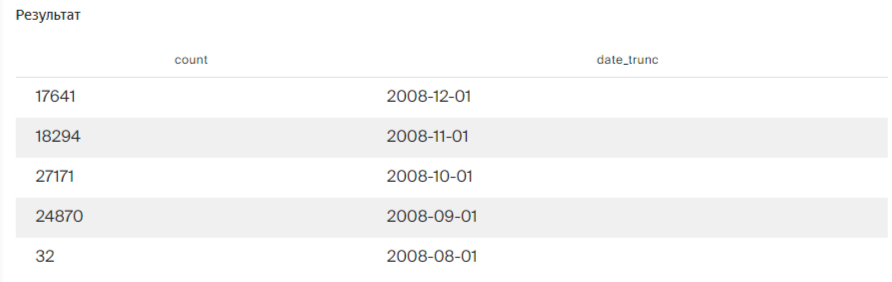
WHERE p.user\_id IN (SELECT \*

FROM users)

AND DATE\_TRUNC('year', p.creation\_date)::date = '2008-01-01'

GROUP BY DATE\_TRUNC('month', p.creation\_date)::date

ORDER BY DATE\_TRUNC('month', p.creation\_date)::date 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.

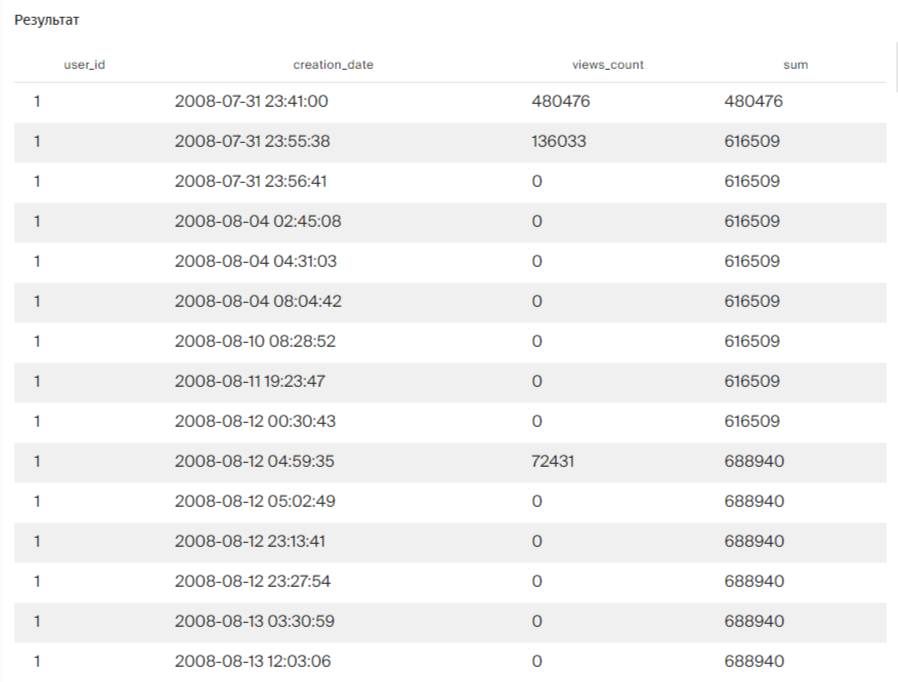
SELECT user\_id,

creation\_date,

views\_count,

SUM(views\_count) OVER (PARTITION BY user\_id ORDER BY creation\_date)

FROM stackoverflow.posts;



5. 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 ad AS (SELECT user\_id,

COUNT(DISTINCT creation\_date::date) AS cnt

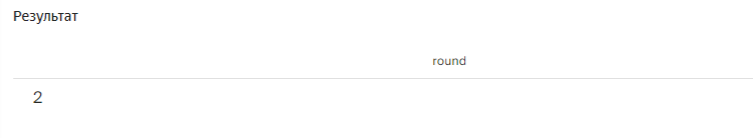
FROM stackoverflow.posts

WHERE creation\_date::date BETWEEN '2008-12-01' AND '2008-12-07'

GROUP BY user\_id)

SELECT ROUND(AVG(ad.cnt))

FROM ad;



6. 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

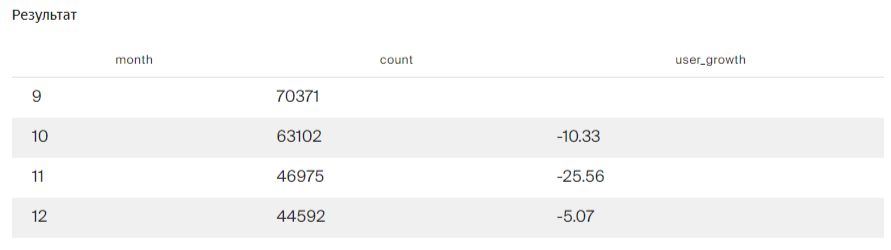
WHERE creation\_date::date BETWEEN '2008-09-01' AND '2008-12-31'

GROUP BY month)

SELECT \*,

ROUND(((count::numeric / LAG(count) OVER (ORDER BY month)) - 1) \* 100, 2) AS user\_growth

FROM month\_post;



7. 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 user\_post AS (SELECT user\_id,

COUNT(DISTINCT id) AS cnt

FROM stackoverflow.posts

GROUP BY user\_id

ORDER BY cnt DESC

LIMIT 1),

dtt AS (SELECT p.user\_id,

p.creation\_date,

extract('week' from p.creation\_date) AS week\_number

FROM stackoverflow.posts AS p

JOIN user\_post ON user\_post.user\_id = p.user\_id

WHERE DATE\_TRUNC('month', p.creation\_date)::date = '2008-10-01')

SELECT DISTINCT week\_number::numeric,

MAX(creation\_date) OVER (PARTITION BY week\_number) AS post\_dt

FROM dtt

ORDER BY week\_number;

