

Описание данных

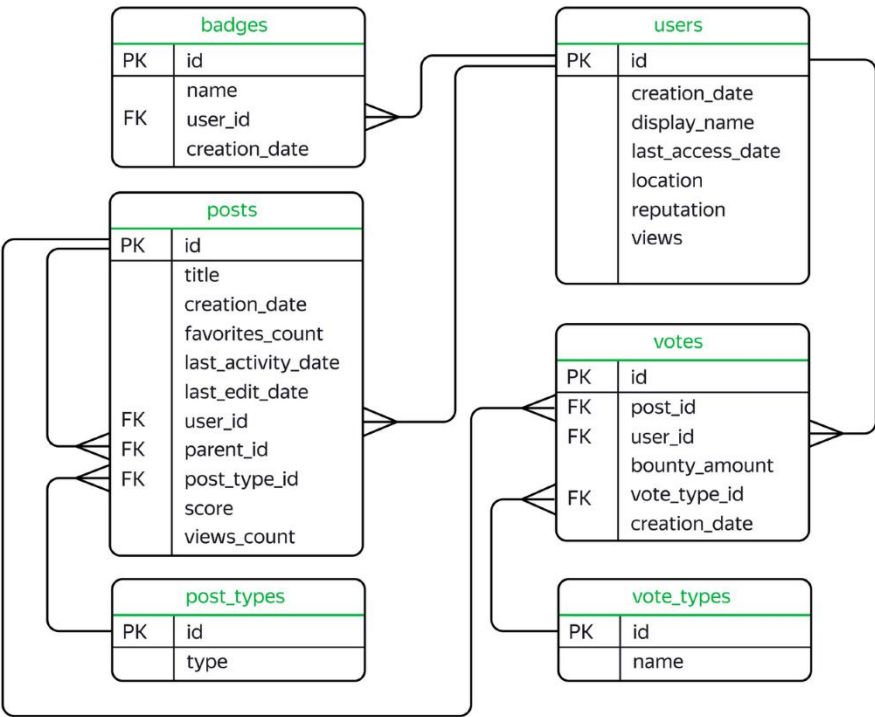


Таблица `stackoverflow.badges`

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

Поле	Описание
id	Идентификатор значка, первичный ключ таблицы
name	Название значка
user_id	Идентификатор пользователя, которому присвоили значок, внешний ключ, отсылающий к таблице <code>users</code>
creation_date	Дата присвоения значка

Таблица `stackoverflow.post_types`

Содержит информацию о типе постов. Их может быть два:

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

Поле	Описание
id	Идентификатор поста, первичный ключ таблицы
type	Тип поста

Таблица `stackoverflow.posts`

Содержит информацию о постах.

Поле	Описание
id	Идентификатор поста, первичный ключ таблицы
title	Заголовок поста
creation_date	Дата создания поста
favorites_count	Число, которое показывает, сколько раз пост добавили в «Закладки»
last_activity_date	Дата последнего действия в посте, например комментария
last_edit_date	Дата последнего изменения поста
user_id	Идентификатор пользователя, который создал пост, внешний ключ к таблице <code>users</code>
parent_id	Если пост написали в ответ на другую публикацию, в это поле попадёт идентификатор поста с вопросом
post_type_id	Идентификатор типа поста, внешний ключ к таблице <code>post_types</code>
score	Количество очков, которое набрал пост
views_count	Количество просмотров

Таблица `stackoverflow.users`

Содержит информацию о пользователях.

Поле	Описание
id	Идентификатор пользователя, первичный ключ таблицы
creation_date	Дата регистрации пользователя
display_name	Имя пользователя
last_access_date	Дата последнего входа
location	Местоположение
reputation	Очки репутации, которые получают за хорошие вопросы и полезные ответы
views	Число просмотров профиля пользователя

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.

Поле	Описание
id	Идентификатор типа голоса, первичный ключ
name	Название метки

Таблица `stackoverflow.votes`

Содержит информацию о голосах за посты.

Поле	Описание
id	Идентификатор голоса, первичный ключ
post_id	Идентификатор поста, внешний ключ к таблице <code>posts</code>
user_id	Идентификатор пользователя, который поставил посту голос, внешний ключ к таблице <code>users</code>
bounty_amount	Сумма вознаграждения, которое назначают, чтобы привлечь внимание к посту
vote_type_id	Идентификатор типа голоса, внешний ключ к таблице <code>vote_types</code>
creation_date	Дата назначения голоса

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);
```

Результат

count
1355

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;
```

Результат

round

383

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
```

Результат

count

7047

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;
```

Результат

count

12

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;
```

Результат

id	name	rank
1	AcceptedByOriginator	15
2	UpMod	14
3	DownMod	13
4	Offensive	12
5	Favorite	11
6	Close	10
7	Reopen	9
8	BountyStart	8
9	BountyClose	7
10	Deletion	6
11	Undeletion	5
12	Spam	4
13	InformModerator	3
14	ModeratorReview	2
15	ApproveEditSuggestion	1

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;
```

Результат	
user_id	v_cnt
20646	36
14728	36
27163	29
41158	24
24820	23
9345	23
3241	23
44330	20
38426	19
19074	19

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;
```

Результат		
user_id	b_cnt	rating
22656	149	1
34509	45	2
1288	40	3
5190	31	4
13913	30	5
893	28	6
10661	28	6
33213	25	7
12950	23	8
25222	20	9

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;
```


Результат

title	user_id	score	avg_score
Diagnosing Deadlocks in SQL Server 2005	1	82	573
How do I calculate someone's age in C#?	1	1743	573
Why doesn't IE7 copy <pre><code> blocks to the clipboard correctly?	1	37	573
Calculate relative time in C#	1	1348	573
Wrapping Stopwatch timing with a delegate or lambda?	1	92	573
Practical non-image based CAPTCHA approaches?	1	318	573
Parameterize an SQL IN clause	1	953	573
Escaping Bracket [in a CONTAINS() clause?	1	10	573
Binary Data in MySQL	2	169	76
Filling a DataSet or DataTable from a LINQ query result set	2	114	76
Why doesn't SQL Full Text Indexing return results for words containing #?	2	19	76
Cross platform Encryption / Decryption applications for secure file transport	2	3	76
Best way to implement request throttling in ASP.NET MVC?	3	196	196
How can I lookup data about a book from its barcode number?	4	69	39
Code to make a DHTMLed control replace straight quotes with curly quotes	4	9	39

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;
```

Результат

title
What's the strangest corner case you've seen in C# or .NET?
What's the hardest or most misunderstood aspect of LINQ?
What are the correct version numbers for C#?
Project management to go with GitHub

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;
```

Результат

id	views	group
3	24396	1
13	35414	1
23	757	1
25	3837	1
36	505	1
43	394	1
45	1971	1
50	1616	1
64	866	1
67	8848	1
72	1475	1
73	169	2
81	311	2
85	1112	1
91	11734	1

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;

```

Результат

id	views	group
16587	62813	1
9094	349	2
9585	349	2
15079	349	2
33437	349	2
3469	99	3
4829	99	3
19006	99	3
22732	99	3
403434	99	3

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;

```

Результат			
	days	cnt_reg	cnt_users
1	34	34	
2	48	82	
3	75	157	
4	192	349	
5	122	471	
6	132	603	
7	104	707	
8	42	749	
9	45	794	
10	93	887	
11	113	1000	
12	113	1113	
13	96	1209	
14	89	1298	

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;
```

Результат	
user_id	diff
1	9:18:29
2	14:37:03
3	3 days, 16:17:09
4	15 days, 5:44:22
5	1 day, 14:57:51
8	0:09:29
9	0:32:42
11	0:00:00
13	1:03:17
17	0:04:58
19	0:14:03
20	5 days, 1:26:33
22	0:01:07
23	0:02:06
24	24 days, 1:48:16

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;
```

Результат

mnth	sum
2008-09-01	452928568
2008-10-01	365400138
2008-11-01	221759651
2008-12-01	197792841
2008-08-01	131367083
2008-07-01	669895

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;
```

Результат

display_name	count
1800 INFORMATION	1
Adam Bellaire	1
Adam Davis	1
Adam Liss	1
aku	1
Alan	8
Amy B	1
anjanb	1
Ben Hoffstein	1
Brian	15
Cade Roux	1
CesarB	1
Chris	29
ConcernedOfTunbridgeWells	1
Craig	10

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;
```

Результат

count	date_trunc
17641	2008-12-01
18294	2008-11-01
27171	2008-10-01
24870	2008-09-01
32	2008-08-01

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;
```

Результат

user_id	creation_date	views_count	sum
1	2008-07-31 23:41:00	480476	480476
1	2008-07-31 23:55:38	136033	616509
1	2008-07-31 23:56:41	0	616509
1	2008-08-04 02:45:08	0	616509
1	2008-08-04 04:31:03	0	616509
1	2008-08-04 08:04:42	0	616509
1	2008-08-10 08:28:52	0	616509
1	2008-08-11 19:23:47	0	616509
1	2008-08-12 00:30:43	0	616509
1	2008-08-12 04:59:35	72431	688940
1	2008-08-12 05:02:49	0	688940
1	2008-08-12 23:13:41	0	688940
1	2008-08-12 23:27:54	0	688940
1	2008-08-13 03:30:59	0	688940
1	2008-08-13 12:03:06	0	688940

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;
```

Результат

round

2

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;
```

Результат

month	count	user_growth
9	70371	
10	63102	-10.33
11	46975	-25.56
12	44592	-5.07

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),
```

```
dt 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 dt
ORDER BY week_number;
```

Результат

week_number	post_dt
40	2008-10-05 09:00:58
41	2008-10-12 21:22:23
42	2008-10-19 06:49:30
43	2008-10-26 21:44:36
44	2008-10-31 22:16:01