

Проект - Анализ сервиса вопросов и ответов по программированию

Цель - выполнить задачи в базе данных [StackOverflow](#) — сервиса вопросов и ответов о программировании.

Задания:

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

```
1 SELECT COUNT(p.id)
2     FROM stackoverflow.posts AS p
3 LEFT JOIN stackoverflow.post_types AS pt
4     ON p.post_type_id = pt.id
5 WHERE pt.type = 'Question' AND (score > 300 OR favorites_count >= 100);
6
```

Результат

count

1355

Посмотрим сколько в среднем в день задавали вопросов с 1 по 18 ноября 2008 включительно. Результат округлим до целого числа.

```
1 SELECT ROUND(AVG(question_cnt))
2 FROM (
3     SELECT creation_date :: date AS question_date,
4           COUNT(p.id) AS question_cnt
5     FROM stackoverflow.posts AS p
6 LEFT JOIN stackoverflow.post_types AS pt
7     ON p.post_type_id = pt.id
8     WHERE pt.type = 'Question' AND creation_date :: date BETWEEN '2008-11-01' AND
9           '2008-11-18'
10    GROUP BY creation_date :: date
11 ) AS Question
```

Результат

round

383

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

```

1 SELECT COUNT(DISTINCT u.id) AS user
2   FROM stackoverflow.users AS u
3   JOIN stackoverflow.badges AS b
4     ON u.id = b.user_id
5  WHERE u.creation_date :: date = b.creation_date :: date

```

Результат

user

7047

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

```

1 WITH votes_cnts AS (
2   SELECT p.id AS id_post,
3          COUNT(v.id) AS votes_cnt
4   FROM stackoverflow.users AS u
5   JOIN stackoverflow.posts AS p
6     ON p.user_id = u.id
7   JOIN stackoverflow.votes AS v
8     ON v.post_id = p.id
9   WHERE display_name = 'Joel Coehoorn'
10  GROUP BY p.id
11 )
12 SELECT COUNT(id_post)
13   FROM votes_cnts
14  WHERE votes_cnt > 0;

```

Результат

count

12

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`.

```

1 SELECT *,
2     RANK() OVER (ORDER BY id DESC)
3 FROM stackoverflow.vote_types
4 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

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.

```

1 SELECT v.user_id AS user_votes,
2     COUNT (v.id) AS votes_cnt
3 FROM stackoverflow.votes AS v
4 JOIN stackoverflow.vote_types AS vt
5     ON vt.id = v.vote_type_id
6 WHERE name = 'Close'
7 GROUP BY v.user_id
8 ORDER BY votes_cnt DESC,
9     v.user_id DESC
10 LIMIT 10;

```

Результат

user_votes	votes_cnt
20646	36
14728	36
27163	29
41158	24
24820	23

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.

```
1 SELECT user_id,  
2         COUNT (id) AS badges_cnt,  
3         DENSE_RANK() OVER (ORDER BY COUNT (id) DESC )  
4     FROM stackoverflow.badges  
5     WHERE creation_date :: date BETWEEN '2008-11-15' AND '2008-12-15'  
6 GROUP BY user_id  
7 ORDER BY badges_cnt DESC,  
8         user_id  
9 LIMIT 10;
```

Результат

user_id	badges_cnt	dense_rank
22656	149	1
34509	45	2
1288	40	3
5100	31	4

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.

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

```

1 SELECT title,
2     user_id,
3     score,
4     ROUND(AVG(score) OVER (PARTITION BY user_id))
5 FROM stackoverflow.posts
6 WHERE title IS NOT NULL
7     AND score != 0;

```

Результат

title	user_id	score	round
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

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

```

1 SELECT title
2 FROM stackoverflow.posts
3 WHERE user_id IN (
4     SELECT user_id
5     FROM stackoverflow.badges
6     GROUP BY user_id
7     HAVING COUNT(id) > 1000)
8 AND title IS NOT NULL;
9

```

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

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.

```

1 SELECT id,
2     views,
3     CASE
4         WHEN views < 100 THEN 3
5         WHEN views >= 100 AND views < 350 THEN 2
6         ELSE 1
7     END AS group
8 FROM stackoverflow.users
9 WHERE location LIKE '%United States%' AND views != 0
10 ORDER BY views DESC;

```

Результат

id	views	group
16587	62813	1
1228	58635	1
36305	43436	1
143305	37431	1
13	35414	1

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.

```

1 WITH us_users AS (
2     SELECT id AS user_id,
3           views AS views_cnt,
4           CASE
5               WHEN views < 100 THEN 3
6               WHEN views >= 100 AND views < 350 THEN 2
7               ELSE 1
8           END AS groups
9     FROM stackoverflow.users
10    WHERE location LIKE '%United States%' AND views != 0
11  )
12
13 SELECT user_id,
14        groups,
15        views_cnt
16 FROM (
17     SELECT user_id,
18           views_cnt,
19           groups,
20           MAX(views_cnt) OVER (PARTITION BY groups ORDER BY views_cnt DESC) AS
max_views

```

Результат

user_id	groups	views_cnt
16587	1	62813

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.

```

1 SELECT days,
2         users_cnt,
3         SUM(users_cnt) OVER (ORDER BY days)
4 FROM (
5     SELECT EXTRACT (DAY FROM creation_date) AS days,
6            COUNT(id) AS users_cnt
7     FROM stackoverflow.users
8     WHERE creation_date :: date BETWEEN '2008-11-01' AND '2008-11-30'
9     GROUP BY EXTRACT (DAY FROM creation_date)
10 ) AS user_november;

```

Результат

	days	users_cnt	sum
1		34	34
2		48	82
3		75	157
4		192	349
5		122	471
6		132	603

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.*


```

1 SELECT DISTINCT p.user_id,
2     MIN(p.creation_date ) OVER (PARTITION BY p.user_id) - u.creation_date AS
   interval
3 FROM stackoverflow.posts AS p
4 LEFT JOIN stackoverflow.users AS u
5     ON p.user_id = u.id;

```

Результат

user_id	interval
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

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.

```

1 SELECT CAST(DATE_TRUNC('month', creation_date) AS date) AS month_posts,
2     SUM(views_count) AS views_count
3 FROM stackoverflow.posts
4 WHERE EXTRACT (YEAR FROM creation_date) = '2008'
5 GROUP BY month_posts
6 ORDER BY views_count DESC;

```

Результат

month_posts	views_count
2008-09-01	452928568
2008-10-01	365400138
2008-11-01	221759651
2008-12-01	197792841
2008-08-01	131367083

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.

```
1 SELECT u.display_name,  
2        COUNT(DISTINCT p.user_id)  
3 FROM stackoverflow.users u  
4 JOIN stackoverflow.posts p  
5     ON u.id = p.user_id  
6 JOIN stackoverflow.post_types pt  
7     ON p.post_type_id = pt.id  
8 WHERE p.creation_date::date BETWEEN u.creation_date::date AND u.creation_date::date  
9      + INTERVAL '1 month' AND pt.type = 'Answer'  
9 GROUP BY u.display_name  
10 HAVING COUNT(DISTINCT p.id) > 100
```

Результат

display_name	count
1800 INFORMATION	1
Adam Bellaire	1
Adam Davis	1
Adam Liss	1

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.

```

1 SELECT CAST(DATE_TRUNC('month', p.creation_date) AS date) AS month_posts,
2         COUNT (DISTINCT id) AS user_count
3 FROM stackoverflow.posts AS p
4 WHERE user_id IN (
5     SELECT DISTINCT u.id
6     FROM stackoverflow.users AS u
7     JOIN stackoverflow.posts AS p
8     ON p.user_id = u.id
9     WHERE u.creation_date :: date BETWEEN '2008-09-01' AND '2008-09-30'
10         AND p.creation_date :: date BETWEEN '2008-12-01' AND '2008-12-31'
11 )
12 AND p.creation_date :: date BETWEEN '2008-01-01' AND '2008-12-31'
13 GROUP BY month_posts
14 ORDER BY month_posts DESC

```

Результат

month_posts	user_count
2008-12-01	17641
2008-11-01	18294
2008-10-01	27171

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.

```

1 SELECT user_id,
2         creation_date,
3         views_count,
4         SUM(views_count) OVER (PARTITION BY user_id ORDER BY creation_date)
5 FROM stackoverflow.posts
6 ORDER BY user_id,
7         creation_date;

```

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

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.

```

1 SELECT ROUND(AVG (active_days))
2 FROM (
3     SELECT user_id,
4            COUNT(DISTINCT creation_date :: date) AS active_days
5     FROM stackoverflow.posts
6     WHERE creation_date :: date BETWEEN '2008-12-01' AND '2008-12-07'
7     GROUP BY user_id
8 ) AS cnt;

```

Результат

round
2

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*.

```
1 WITH posts_month AS (  
2     SELECT EXTRACT (MONTH FROM creation_date) AS month_number,  
3           COUNT(id) AS posts_cnt  
4     FROM stackoverflow.posts  
5     WHERE creation_date :: date BETWEEN '2008-09-01' AND '2008-12-31'  
6     GROUP BY EXTRACT (MONTH FROM creation_date)  
7     ORDER BY month_number  
8 )  
9  
10 SELECT *,  
11     ROUND(( posts_cnt - LAG(posts_cnt) OVER () ) * 100 / LAG(posts_cnt) OVER () ::  
12     numeric, 2)  
12 FROM posts_month
```

Результат

month_number	posts_cnt	round
9	70371	
10	63102	-10.33
11	46975	-25.56
12	44592	-5.07

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.

```

1 WITH week AS (
2     SELECT EXTRACT(WEEK FROM creation_date) AS week_number,
3           MAX(creation_date) OVER (ORDER BY EXTRACT(WEEK FROM creation_date))
4     FROM stackoverflow.posts
5     WHERE user_id = (
6         SELECT user_id
7         FROM stackoverflow.posts
8         GROUP BY user_id
9         ORDER BY COUNT(id) DESC
10        LIMIT 1
11    )
12     AND creation_date :: date BETWEEN '2008-10-01' AND '2008-10-31'
13     ORDER BY creation_date
14 )
15
16 SELECT DISTINCT *
17 FROM week
18 ORDER BY week_number

```

Результат

week_number	max
40	2008-10-05 09:00:58
41	2008-10-12 21:22:23