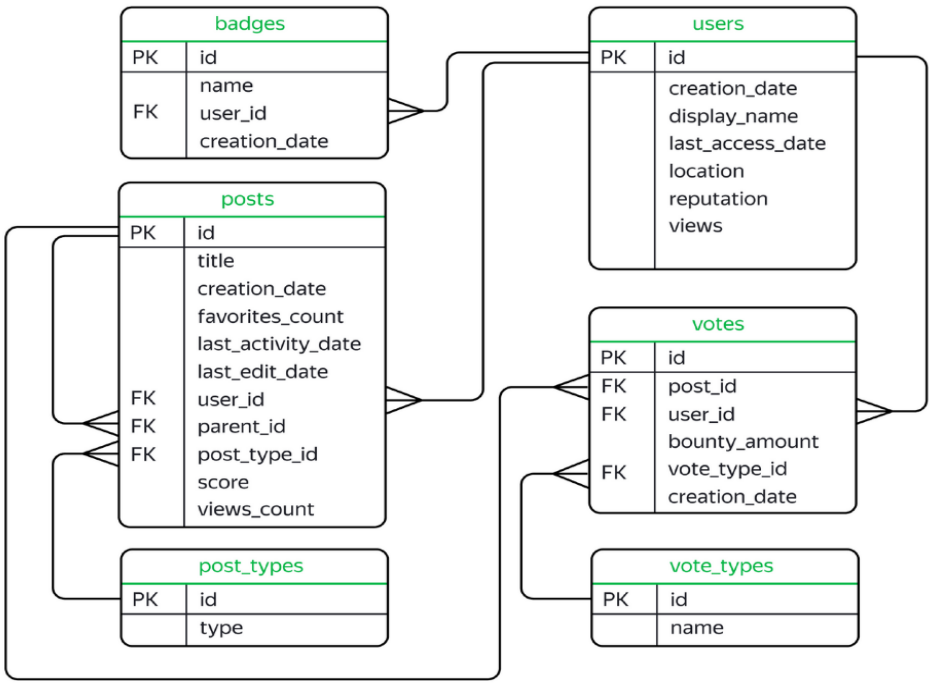
Учебный проект №8
Продвинутый SQL

Описание данных



Данные таблиц:

stackoverflow.badges - хранит информацию о значках, которые присуждаются за разные достижения

stackoverflow.post_types - содержит информацию о типе постов.

stackoverflow.posts - содержит информацию о постах.

stackoverflow.users - содержит информацию о пользователях

stackoverflow.vote_types - содержит информацию о типах голосов

stackoverflow.votes - содержит информацию о голосах за посты

Задания (первая часть)

Задача 1/13
Найдите количество вопросов, которые набрали больше 300 очков или как минимум 100 раз были добавлены в «Закладки».

Решение:

```
SELECT COUNT (DISTINCT p.id)
FROM stackoverflow.posts p
LEFT JOIN stackoverflow.post_types pt ON p.post_type_id = pt.id
WHERE pt.type = 'Question' AND (p.score > 300 OR p.favorites_count >= 100);
```

Результат:

count
1355

Задача 2/13
Сколько в среднем в день задавали вопросов с 1 по 18 ноября 2008 включительно? Результат округлите до целого числа.

Решение:

```
WITH dcnt AS
   (SELECT COUNT(DISTINCT p.id) dc
    FROM stackoverflow.posts p
    JOIN stackoverflow.post_types pt ON p.post_type_id=pt.id
    WHERE pt.type = 'Question' AND p.creation_date::DATE BETWEEN '2008-11-01' AND '2008-11-
18'
    GROUP BY p.creation_date::DATE)
SELECT ROUND(AVG(dcnt.dc))
FROM dcnt;
```

Результат

round
383

Задача 3/13
Сколько пользователей получили значки сразу в день регистрации? Выведите количество уникальных пользователей.

Решение:

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

Результат:

count
7047

Задача 4/13
Сколько уникальных постов пользователя с именем Joel Coehoorn получили хотя бы один голос?

Решение:
```
SELECT COUNT(DISTINCT p.id)
FROM stackoverflow.posts p
JOIN stackoverflow.votes v on p.id=v.post_id
WHERE p.user_id IN
    (SELECT u.id
     FROM stackoverflow.users u
     WHERE u.display_name = 'Joel Coehoorn');
```

-- считать голоса для поста не нужно, достаточно внутреннего джйона с таблицей votes, если ид поста по ней проходит - значит хоть один голос есть, а остальные в иннер джойн не войдут

Результат

count
12

Задача 5/13
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.

Решение:
```
SELECT *
     ,ROW_NUMBER() OVER(ORDER BY vt.id DESC) rank
FROM stackoverflow.vote_types vt
ORDER BY vt.id;
```

Вывод:
Результат

id	name	rank
1	AcceptedByOriginator	15
2	UpMod	14
…		
14	ModeratorReview	2
15	ApproveEditSuggestion	1

Задача 6/13
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.

Решение:
```
SELECT DISTINCT v.user_id
     ,COUNT(v.id) OVER (PARTITION BY v.user_id) AS close_cnt
FROM stackoverflow.votes v
JOIN stackoverflow.vote_types vt ON v.vote_type_id=vt.id
```

```
WHERE vt.name = 'Close'
ORDER BY close_cnt DESC, v.user_id DESC
LIMIT 10;
```

Результат

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

Задача 7/13

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.

Решение:

```
SELECT *
     ,DENSE_RANK() OVER (ORDER BY badge_cnt DESC) rating
FROM
   (SELECT u.id
        ,COUNT(b.id) badge_cnt
   FROM stackoverflow.users u
   JOIN stackoverflow.badges b ON u.id=b.user_id
   WHERE b.creation_date::DATE BETWEEN '2008-11-15' AND '2008-12-15'
   GROUP BY 1
   ORDER BY 2 DESC, 1 ASC
   LIMIT 10) AS cnt;
```

Результат

id	badge_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

id	badge_cnt	rating
25222	20	9

Задача 8/13

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.

Решение:

```
SELECT p.title
    ,p.user_id
    ,p.score
    ,ROUND(AVG(p.score) OVER (PARTITION BY p.user_id)) avg_score
FROM stackoverflow.posts p
WHERE p.score <> 0 AND p.title IS NOT NULL
```

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
…			
Converting ARBG to RGB with alpha blending	45	19	93
Programmatically access a Microsoft Project (MPP) file from C#	45	18	93
Limits on number of Rows in a SQL Server Table	45	4	93

Задача 9/13

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

Решение:

```
SELECT p.title
FROM stackoverflow.posts p
WHERE p.title IS NOT NULL AND
    p.user_id IN
        (SELECT b.user_id
        FROM stackoverflow.badges b
        GROUP BY b.user_id
        HAVING COUNT(b.id) > 1000);
```

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#?

title
Project management to go with GitHub

Задача 10/13
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.

Решение:
```
SELECT u.id
     ,u.views
     ,CASE
         WHEN u.views >= 350 THEN 1
         WHEN u.views >= 100 THEN 2
         WHEN u.views < 100 THEN 3
      END as cat
FROM stackoverflow.users u
WHERE u.location LIKE '%United States%' AND u.views > 0;
```

Результат

id	views	cat
3	24396	1
13	35414	1
23	757	1
…	…	…
1366	1078	1
1376	255	2
1377	57	3

Задача 11/13
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.

Решение:
```
WITH cnt AS
   (SELECT u.id
        ,u.views
        ,CASE
            WHEN u.views >= 350 THEN 1
            WHEN u.views >= 100 THEN 2
            WHEN u.views < 100 THEN 3
         END as cat
   FROM stackoverflow.users u
```

```
   WHERE u.location LIKE '%United States%' AND u.views > 0),
cnt_max AS
  (SELECT *
       ,MAX(cnt.views) OVER(PARTITION BY cnt.cat) max_view
   FROM cnt)
SELECT id
     ,cat
     ,views
FROM cnt_max
WHERE cnt_max.views = cnt_max.max_view
ORDER BY 3 DESC, 1 ASC;
```

Результат

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

Задача 12/13

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.

Решение

```
WITH cnt AS
  (SELECT DISTINCT EXTRACT(DAY FROM u.creation_date::DATE) dt
       ,COUNT(u.id) OVER(PARTITION BY EXTRACT(DAY FROM u.creation_date::DATE))
user_cnt
   FROM stackoverflow.users u
   WHERE u.creation_date::DATE BETWEEN '2008-11-01' AND '2008-11-30')
SELECT *
     ,SUM(user_cnt) OVER (ORDER BY dt) cum_sum
FROM cnt;
```

Результат

dt	user_cnt	cum_sum
1	34	34
2	48	82
3	75	157
4	192	349
5	122	471
6	132	603
7	104	707

dt	user_cnt	cum_sum
8	42	749
9	45	794
10	93	887
11	113	1000
12	113	1113
13	96	1209
14	89	1298
15	42	1340
16	32	1372
17	84	1456
18	89	1545
19	107	1652
20	95	1747
21	81	1828
22	40	1868
23	50	1918
24	84	2002
25	104	2106
26	98	2204
27	71	2275
28	56	2331
29	44	2375
30	33	2408

Задача 13/13

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.

Решение:

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.

Результат

id	time_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
…	…
265	1 day, 22:25:54
266	2:00:19

Задания (Вторая часть)

Задача 1/7
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.

Решение:
```
SELECT DISTINCT CAST(DATE_TRUNC('month', p.creation_date) AS DATE) month_dt
     ,SUM(p.views_count) OVER (PARTITION BY CAST(DATE_TRUNC('month', p.creation_date) AS
DATE)) sum_view
FROM stackoverflow.posts p
WHERE p.creation_date BETWEEN '2008-01-01' AND '2008-12-31'
ORDER BY 2 DESC;
```

Результат

month_dt	**sum_view**
2008-09-01	452928568
2008-10-01	365400138
2008-11-01	221759651
2008-12-01	197792841
2008-08-01	131367083
2008-07-01	669895

Обратите внимание, что данные отличаются. Возможно, повышенная активность в сентябре и октябре связана с началом учебного года. Малая активность в июле может свидетельствовать о неполноте данных.

Задача 2/7
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.

Решение:
```
SELECT u.display_name
     ,COUNT(DISTINCT u.id) cnt
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 1
HAVING COUNT(pt.type) > 100
ORDER BY 1;
```

Результат

display_name	cnt
1800 INFORMATION	1
Adam Bellaire	1
Adam Davis	1
Adam Liss	1
aku	1
Alan	8
…	…
TheSmurf	1
Tom	19
tvanfosson	1
tzot	1
Vilx-	1
Vinko Vrsalovic	1

Кажется, что одному имени пользователя должен соответствовать один user_id. Но это не так: многим популярным именам вроде Alan, Dan или Chris соответствует несколько значений user_id. Данные лучше не анализировать по имени, иначе результаты будут некорректными.

Задача 3/7

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.

Решение:

```
WITH pp AS
   (SELECT u.id
         ,COUNT(p.id)
    FROM stackoverflow.users u
    JOIN stackoverflow.posts p ON u.id=p.user_id
    WHERE (p.creation_date::DATE BETWEEN '2008-12-01' AND '2008-12-31')
       AND (u.creation_date::DATE BETWEEN '2008-09-01' AND '2008-09-30')
    GROUP BY 1
    HAVING COUNT(p.id) > 0)

SELECT CAST(DATE_TRUNC('month', p.creation_date) AS DATE)
     ,COUNT(p.id)
FROM stackoverflow.posts p
JOIN pp ON p.user_id=pp.id
WHERE p.creation_date BETWEEN '2008-01-01' AND '2008-12-31'
GROUP BY 1
ORDER BY 1 DESC;
```

Результат

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

В итоговой таблице встречаются аномальные значения: пользователи, зарегистрированные в сентябре, были активны и в августе. Возможно, это ошибка в данных.

Задача 4/7

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.

Решение:

```
SELECT p.user_id
     ,p.creation_date
     ,p.views_count
     ,SUM(p.views_count) OVER (PARTITION BY p.user_id ORDER BY p.creation_date) sum_view
FROM stackoverflow.posts p
ORDER by 1;
```

Результат

user_id	creation_date	views_count	sum_view
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
…			
5	2008-09-11 08:21:49	0	75605
5	2008-09-12 16:01:55	0	75605
5	2008-09-12 16:20:24	3839	79444

В теории все расчёты с оконной функцией можно выполнить и без неё. Но размер запроса имеет значение.

Задача 5/7

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.

Решение:

```
WITH post_count AS
  (SELECT p.user_id
        ,p.creation_date::DATE dt
        ,COUNT(p.id) cnt
   FROM stackoverflow.posts p
   WHERE p.creation_date BETWEEN '2008-12-01' AND '2008-12-07'
   GROUP BY 1,2
  ),
avg_days AS
   (SELECT pc.user_id
```

```
        ,COUNT(pc.cnt) OVER (PARTITION BY pc.user_id) day_cnt
    FROM post_count pc)
SELECT ROUND(AVG(day_cnt))
FROM avg_days;
```

Результат

round
2

Попробуйте проанализировать итоговую таблицу: какие выводы можно сделать?

Задача 6/7

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.

Решение:

```
WITH month_count AS
   (SELECT DISTINCT EXTRACT(month FROM p.creation_date) mdt
        ,COUNT(p.id) OVER (PARTITION BY EXTRACT(month FROM p.creation_date)) mcnt
   FROM stackoverflow.posts p
   WHERE p.creation_date BETWEEN '2008-09-01' AND '2008-12-31')
SELECT mc.mdt
     ,mc.mcnt
     ,ROUND(((mc.mcnt::NUMERIC / LAG(mc.mcnt) OVER (ORDER BY mc.mdt)) - 1) * 100, 2) AS
diff
FROM month_count mc;
```

Результат

mdt	mcnt	diff
9	70371	
10	63102	-10.33
11	46975	-25.56
12	44592	-5.07

В SQL нет инструментов визуализации. А жаль, можно бы было сразу построить диаграмму для более информативного отображения результата.

Задача 7/7

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.

Решение:

```
WITH week_div AS
    (SELECT p.creation_date
          ,p.id
          ,p.user_id
          ,EXTRACT(week FROM p.creation_date) week_num
          ,MAX(p.creation_date) OVER (PARTITION BY EXTRACT(week FROM p.creation_date))
max_dt
     FROM stackoverflow.posts p
     WHERE p.creation_date::DATE BETWEEN '2008-10-01' AND '2008-10-31'
     AND p.user_id IN
                (SELECT p.user_id
                 FROM stackoverflow.posts p
                 GROUP BY 1
                 ORDER BY COUNT(p.id) DESC
                 LIMIT 1)
     ORDER BY 1)
SELECT week_num
     ,creation_date
FROM week_div
WHERE creation_date = max_dt;
```

Результат

week_num	creation_date
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

Поздравляем! Вы выполнили все задания проекта.