FIIT STU

DBS zadanie 3

Dokumentácia Databázové systémy

Norbert Matuška
3-23-2024

Contents

GET /v3/tags/:tag/comments?count=:count

```
cur.execute("""
SELECT
    TO_CHAR('00:00:00'::time + (comments_data.sum_diff /
comments_data.count_diff) * '1 second'::interval, 'HH24:MI:SS.US') AS avg,
    comments_data.created_at,
    comments_data.diff,
    comments_data.displayname,
    comments_data.post_created_at,
    comments_data.post_id,
    comments_data.text,
    comments_data.title
FROM (
    SELECT
        post_data.post_id,
        post_data.title,
        post_data.displayname,
        post_data.text,
        post_data.post_created_at,
        post_data.created_at,
        TO_CHAR(post_data.created_at - LAG(post_data.created_at) OVER (
            PARTITION BY post_data.post_id ORDER BY post_data.created_at
        ), 'HH24:MI:SS.US') AS diff,
        SUM(post_data.diff) OVER (
            PARTITION BY post_data.post_id ORDER BY post_data.created_at
            ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
        ) as sum_diff,
        COUNT(post_data.diff) OVER (
            PARTITION BY post_data.post_id ORDER BY post_data.created_at
            ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
        ) as count_diff
    FROM (
        SELECT
            posts.id AS post_id,
            posts.title,
            users.displayname,
            comments.text,
            posts.creationdate AS post_created_at,
            comments.creationdate AS created_at,
            EXTRACT(EPOCH FROM (comments.creationdate -
LAG(comments.creationdate) OVER (
                PARTITION BY comments.postid ORDER BY comments.creationdate
            ))) AS diff
        FROM
            comments
        JOIN
            posts ON comments.postid = posts.id
        JOIN
            users ON comments.userid = users.id
        JOIN
            post_tags ON posts.id = post_tags.post_id
        JOIN
            tags ON post_tags.tag_id = tags.id
        WHERE
            tags.tagname = %s
```

```
    ) AS post_data
    JOIN (
        SELECT
            postid
        FROM
            comments
        GROUP BY
            postid
        HAVING
            COUNT(*) > %s
    ) AS popular_posts ON post_data.post_id = popular_posts.postid
    WHERE
        post_data.diff IS NOT NULL
) AS comments_data
ORDER BY
    comments_data.post_id, comments_data.created_at;
""", (tag, count))
```

Subequery post_data joinne viacero tables aby sme mohli vyhľadávať podľa špecifického tagname a vypočítame časový rozdiel pomocou funkcie LAG. Táto window function môže accessovať dáta z predošlého riadka. Používame partition cez ID aby sme počítali každý post ako separátny subset dát. Ďalej v subquery popular posts filtrujeme posty, ktoré majú aspoň x počet komentárov. Na základe post_data si tiež vypočítavam sum_diff a count_diff pre každý príspevok až po aktuálny riadok (ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW). Nakoniec vypočítam avg pre komentáre každého postu (sum_diff/countdiff) na formát HH24:MI:SS.US.

Example output pre tag ´networking´ a count > 40

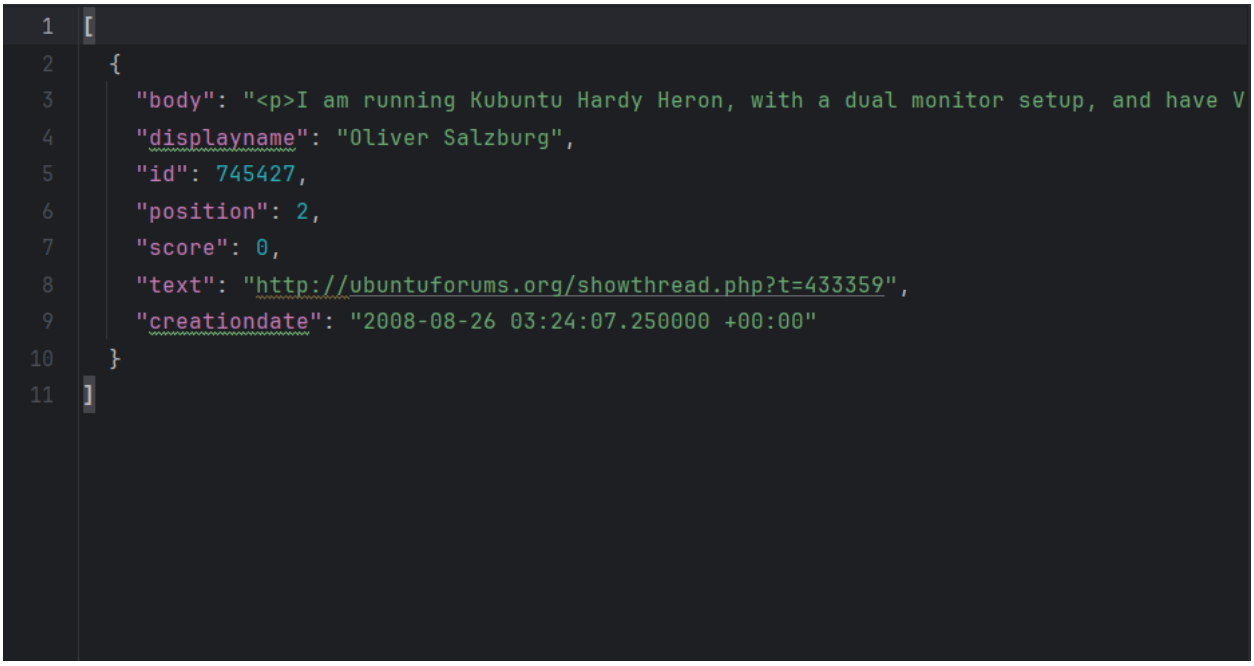

```
[
  {
    "post_id": 1034137,
    "title": "Did I just get hacked?",
    "displayname": "David Schwartz",
    "text": "The attack actually came from China.",
    "post_created_at": "2016-02-01 10:21:48.690000 +00:00",
    "created_at": "2016-02-01 10:30:45.310000 +00:00",
    "diff": null,
    "avg": "00:05:42.700000"
  },
  {
    "post_id": 1034137,
    "title": "Did I just get hacked?",
    "displayname": "vaid",
    "text": "Yes but what is a Microsoft owned IP doing trying to breach a device acros
    "post_created_at": "2016-02-01 10:21:48.690000 +00:00",
    "created_at": "2016-02-01 10:37:58.037000 +00:00",
    "diff": "00:07:12.727000",
    "avg": "00:06:27.713500"
  },
  {
    "post_id": 1034137,
    "title": "Did I just get hacked?",
    "displayname": "Journeyman Geek",
    "text": "You got brute forced. This is why one does not leave a ssh server on the :
    "post_created_at": "2016-02-01 10:21:48.690000 +00:00",
    "created_at": "2016-02-01 10:41:17.843000 +00:00",
    "diff": "00:03:19.806000",
    "avg": "00:05:25.077667"
  },
```

GET /v3/tags/:tagname/comments/:position?limit=:limit

```
cur.execute("""
SELECT
    sub.body,
    sub.displayname,
    sub.id,
    sub.position,
    sub.score,
    sub.text,
    sub.creationdate
FROM (
    SELECT
        p.id as post_id,
        c.id as id,
        u.displayname,
        p.body as body,
        c.text as text,
        c.score,
        p.creationdate,
        ROW_NUMBER() OVER (PARTITION BY p.id ORDER BY c.creationdate ASC) as
position
    FROM
        comments c
    JOIN
        posts p ON p.id = c.postid
    JOIN
        users u ON u.id = c.userid
    JOIN
        post_tags pt ON p.id = pt.post_id
    JOIN
        tags t ON pt.tag_id = t.id
    WHERE
        t.tagname = %s
) sub
WHERE
    sub.position = %s
ORDER BY
    sub.creationdate ASC
LIMIT %s;
""", (tag_name, position, limit))
```

Pomocou ROW_NUMBER() v subquery sub, si očíslujem komentáre zoradené chronologicky podľa creationdate a reštartuje sa poďla každého unikátneho postu (PARTITION BY p.id). Outer query len vyfiltruje komentáre a posty ktorých position je práve tá ktorá prišla do požiadavky.

Example output pre tagname ´linux´, position 2 a limit 1




```
[
  {
    "body": "<p>I am running Kubuntu Hardy Heron, with a dual monitor setup, and have V
    "displayname": "Oliver Salzburg",
    "id": 745427,
    "position": 2,
    "score": 0,
    "text": "http://ubuntuforums.org/showthread.php?t=433359",
    "creationdate": "2008-08-26 03:24:07.250000 +00:00"
  }
]
```

GET /v3/posts/:postid?limit=:limit

```
cur.execute("""
SELECT
    u.displayname,
    p.body,
    p.creationdate as created_at
FROM
    posts p
JOIN
    users u ON p.owneruserid = u.id
WHERE
    p.id = %s OR p.parentid = %s
ORDER BY
    p.creationdate ASC
LIMIT %s;
""", (post_id, post_id, limit))
```

Keďže zoradím posty podľa creationdate, môžem si byť istý, že na začiatku bude originálny post a ďalej budú nasledovať subposty.

Example output pre id 2154 a limit 2


```
[
  {
    "displayname": "Eugene M",
    "body": "<p>So, I'm a technology guy and sometimes I have to troubleshoot a home network, including my own. I make s
    "created_at": "2009-07-15 12:51:57.340000 +00:00"
  },
  {
    "displayname": "Ólafur Waage",
    "body": "<p>Every router has it's original firmware stored somewhere on it.</p>\n\n<p>When you reset the router you
    "created_at": "2009-07-15 12:54:48.507000 +00:00"
  }
]
```