FIIT STU

DBS zadanie 2

Dokumentácia Databázové Systémy

Norbert Matuška

3-9-2024

Contents

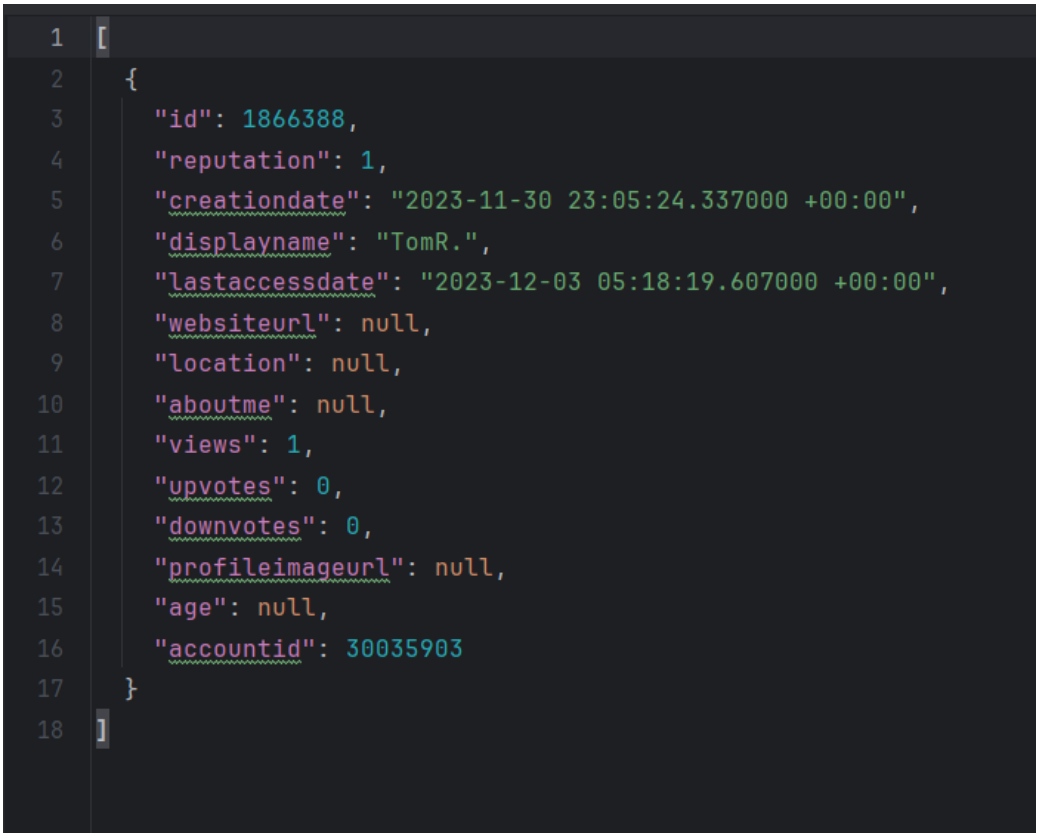
GET /v2/posts/:post id/users

```
cur.execute("""
SELECT
    u.id,
    u.reputation,
    u.creationdate,
    u.displayname,
    u.lastaccessdate,
    u.websiteurl,
    u.location,
    u.aboutme,
    u.views,
    u.upvotes,
    u.downvotes,
    u.profileimageurl,
    u.age,
    u.accountid
FROM
    users u
INNER JOIN comments c ON u.id = c.userid
WHERE
    c.postid = %s
ORDER BY
    c.creationdate DESC
""", (post_id,))

users_data = cur.fetchall()
cur.close()
conn.close()
return {"items": users_data}
```

Najprv SELECTnem všetky dôležité údaje, ktoré sa od nás vyžadujú, spravím INNER JOIN usera a comments kde sa users ID zhodujú v oboch a nakoniec iba porovnávam vo WHERE či sa post ID zhodujé s požadovaným post ID.

Example postID 1819157




```
[
  {
    "id": 1866388,
    "reputation": 1,
    "creationdate": "2023-11-30 23:05:24.337000 +00:00",
    "displayname": "TomR.",
    "lastaccessdate": "2023-12-03 05:18:19.607000 +00:00",
    "websiteurl": null,
    "location": null,
    "aboutme": null,
    "views": 1,
    "upvotes": 0,
    "downvotes": 0,
    "profileimageurl": null,
    "age": null,
    "accountid": 30035903
  }
]
```

GET /v2/users/:user id/friends

```
cur.execute("""
SELECT
    u.id,
    u.reputation,
    u.creationdate,
    u.displayname,
    u.lastaccessdate,
    u.websiteurl,
    u.location,
    u.aboutme,
    u.views,
    u.upvotes,
    u.downvotes,
    u.profileimageurl,
    u.age,
    u.accountid
FROM
    users u
WHERE
    EXISTS (
        SELECT 1
        FROM comments c
        INNER JOIN posts p ON c.postid = p.id
        WHERE (p.owneruserid = %s OR c.userid = %s)
        AND u.id = c.userid
    )
ORDER BY
    u.creationdate;
""", (user_id, user_id))

friends_data = cur.fetchall()
cur.close()
conn.close()
return {"items": friends_data}
```

V podmienke pri tejto query kontrolujeme pomocou subquery, či existujú nejaké záznamy v tabuľke comments kde user je buď vlastníkom príspevku alebo k nemu pridal komentár. Ak existuje, zahrnieme ho do výsledku.

Example user ID 1076348


```
{
  "id": 482362,
  "reputation": 10581,
  "creationdate": "2015-08-11 15:42:36.267000 +00:00",
  "displayname": "DrZoo",
  "lastaccessdate": "2023-12-03 05:41:11.750000 +00:00",
  "websiteurl": null,
  "location": null,
  "aboutme": null,
  "views": 1442,
  "upvotes": 555,
  "downvotes": 46,
  "profileimageurl": null,
  "age": null,
  "accountid": 2968677
},
{
  "id": 1076348,
  "reputation": 1,
  "creationdate": "2019-08-15 14:00:28.473000 +00:00",
  "displayname": "Richard",
  "lastaccessdate": "2019-09-10 14:57:48.527000 +00:00",
  "websiteurl": null,
  "location": null,
  "aboutme": null,
  "views": 0,
  "upvotes": 0,
  "downvotes": 0,
  "profileimageurl": null,
  "age": null,
  "accountid": 16514661
}
```

GET /v2/tags/:tagname/stats

```
cur.execute("""
WITH PostDay AS (
    SELECT
        EXTRACT(DOW FROM p.creationdate) AS day_of_week,
        COUNT(DISTINCT p.id) AS day_count_tag
    FROM posts p
    JOIN post_tags pt ON p.id = pt.post_id
    JOIN tags t ON pt.tag_id = t.id
    WHERE t.tagname = %s
    GROUP BY day_of_week
),
TotalPosts AS (
    SELECT
        EXTRACT(DOW FROM p.creationdate) AS day_of_week,
        COUNT(DISTINCT p.id) AS day_count
    FROM posts p
    JOIN post_tags pt ON p.id = pt.post_id
    JOIN tags t ON pt.tag_id = t.id
    GROUP BY day_of_week
)
SELECT
    CASE
        WHEN pd.day_of_week = 0 THEN 'sunday'
        WHEN pd.day_of_week = 1 THEN 'monday'
        WHEN pd.day_of_week = 2 THEN 'tuesday'
        WHEN pd.day_of_week = 3 THEN 'wednesday'
        WHEN pd.day_of_week = 4 THEN 'thursday'
        WHEN pd.day_of_week = 5 THEN 'friday'
        WHEN pd.day_of_week = 6 THEN 'saturday'
    END AS day,
    ROUND((pd.day_count_tag::NUMERIC/tp.day_count::NUMERIC) * 100, 2) AS
day_percentage
FROM PostDay pd
JOIN TotalPosts tp ON tp.day_of_week = pd.day_of_week
ORDER BY pd.day_of_week;
        """, (tagname,))

        stats_data = cur.fetchall()
        cur.close()
        conn.close()

        stats_result = {"result": {day['day']: day['day_percentage'] for day
in stats_data}}
        return stats_result
```

V prvom CTE (Common Table Expression) PostDay extractujem deň v týždni (DOW) a počítam počet postov podľa daného tagu. Potom v CTE TotalPosts už počítam všetky posty. Nakoniec iba predelím pre každý deň tagnute posty/všetky posty.

Example tag name ´linux´

	day	day_percentage
1	sunday	11.78
2	monday	11.57
3	tuesday	11.6
4	wednesday	11.51
5	thursday	11.35
6	friday	11.72
7	saturday	11.99

GET /v2/posts/?duration=:duration in minutes&limit=:limit

```
cur.execute("""
SELECT
    id,
    creationdate,
    viewcount,
    lasteditdate,
    lastactivitydate,
    title,
    closeddate,
    ROUND(EXTRACT(EPOCH FROM (closeddate - creationdate)) / 60, 2) AS
duration
FROM
    posts
WHERE
    closeddate IS NOT NULL
    AND EXTRACT(EPOCH FROM (closeddate - creationdate)) / 60 <= %s
ORDER BY
    creationdate DESC
LIMIT
    %s;
        """, (duration, limit))

        posts_data = cur.fetchall()
        cur.close()
        conn.close()
        return {"items": posts_data}
```

Pomocou EPOCH konvertujem čas na sekundy, predelím ho 60 aby z neho boli minúty. V podmienke iba pozerám či bol post zatvorený a či trval menej ako požadovaný čas.

Example duration 5 limit 2

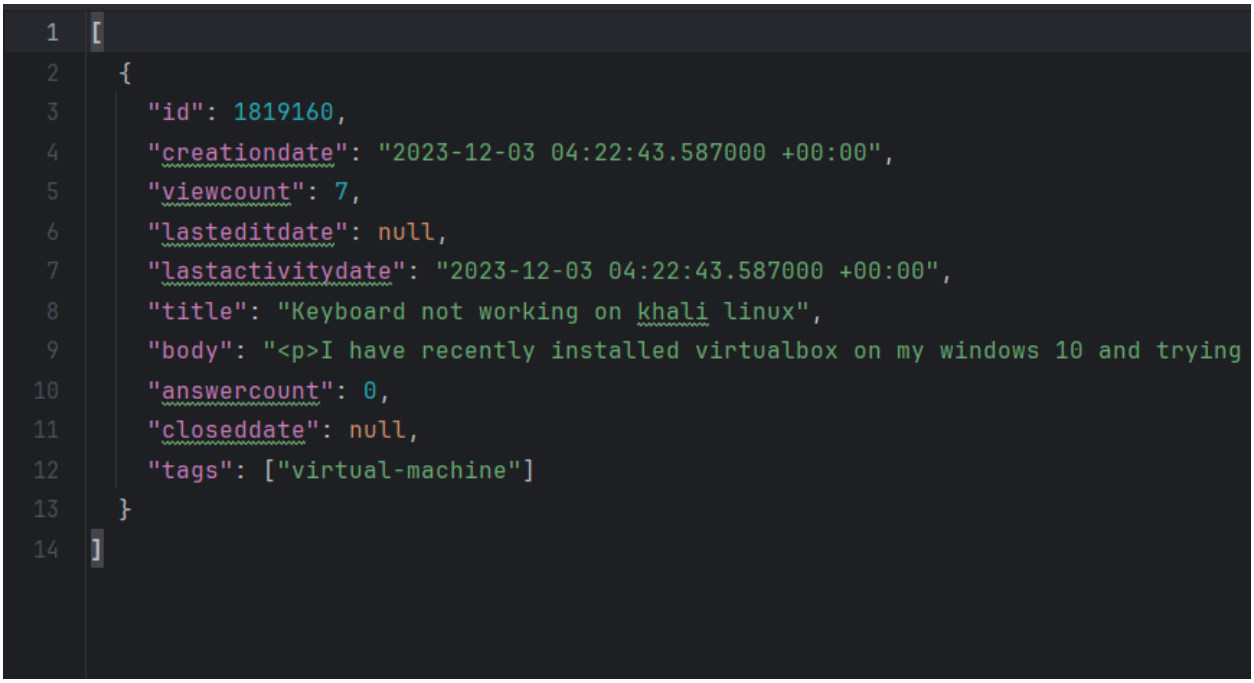

```
[
  {
    "id": 1818849,
    "creationdate": "2023-11-30 15:55:32.137000 +00:00",
    "viewcount": 22924,
    "lasteditdate": null,
    "lastactivitydate": "2023-11-30 15:55:32.137000 +00:00",
    "title": "Why is my home router address is 10.x.x.x and not 100.x.x.x which is properly reserved and widely accepted for CGNAT?
    "closeddate": "2023-11-30 15:59:23.560000 +00:00",
    "duration": 3.86
  },
  {
    "id": 1818386,
    "creationdate": "2023-11-27 17:26:57.617000 +00:00",
    "viewcount": 19,
    "lasteditdate": null,
    "lastactivitydate": "2023-11-27 17:26:57.617000 +00:00",
    "title": "Are there any libraries for parsing DWG files with LGPL, MIT, Apache, BSD?",
    "closeddate": "2023-11-27 17:29:18.947000 +00:00",
    "duration": 2.36
  }
]
```

GET /v2/posts?limit=:limit&query=:query

```
cur.execute("""
SELECT
    p.id,
    p.creationdate,
    p.viewcount,
    p.lasteditdate,
    p.lastactivitydate,
    p.title,
    p.body,
    p.answercount,
    p.closeddate,
    ARRAY_AGG(t.tagname) AS tags
FROM
    posts p
JOIN
    post_tags pt ON p.id = pt.post_id
JOIN
    tags t ON pt.tag_id = t.id
WHERE
    (LOWER(UNACCENT(p.title)) LIKE %s OR
    LOWER(UNACCENT(p.body)) LIKE %s)
GROUP BY
        p.id, p.creationdate, p.viewcount, p.lasteditdate,
p.lastactivitydate,
        p.title, p.body, p.answercount, p.closeddate
ORDER BY
    p.creationdate DESC
LIMIT
    %s;
```

Podľa zadaného stringu hľadám, či sa objavuje niekde v nadpise alebo v body postu. LOWER a UNACCENT používam aby hľadanie bolo case-insensitive a accent-insensitive. ARRAY_AGG používam preto, aby posty s viacerými tagmi neboli zobrazené viackrát.

Example query ´linux´ a limit 1




```
[
  {
    "id": 1819160,
    "creationdate": "2023-12-03 04:22:43.587000 +00:00",
    "viewcount": 7,
    "lasteditdate": null,
    "lastactivitydate": "2023-12-03 04:22:43.587000 +00:00",
    "title": "Keyboard not working on khali linux",
    "body": "<p>I have recently installed virtualbox on my windows 10 and trying
    "answercount": 0,
    "closeddate": null,
    "tags": ["virtual-machine"]
  }
]
```