Meno: Zaťovič Dominik (ID: 121058)
Predmet: Databázové systémy
Ročník: LS 2023/2024

ZADANIE 3

Popis zadania:

Zadanie je zamerané na vytvorenie základných SQL dotazov nad priloženou PostgreSQL databázou, ktorá vznikla z Stack Exchange Data Dump Superuser datasetu. Cieľom je realizovať nižšie uvedené úlohy ako RESTful endpointy, ktoré sú realizované ako SQL dotazy, transformované do JSON výstupu. Výstup je opísaný ako JSON Schema. Vstupy na pripojenie k databázovému serveru budú poskytované rovnako ako v zadani 1 (pomocou environment premenných).

Poradie vo výstupe musí byť zhodné s jeho definíciou pri jednotlivých end-pointoch. Pri realizácii je možné používať iba čisté SQL dopyty a nie je dovolené používať žiadne ORM.

Okrem implementovania samotných endpointov je potrebné vyhotoviť dokumentáciu, ktorá bude obsahovať:

SQL dopyty s ich popisom, príklady volania HTTP end-pointu (pre každý endpoint).

Začiatok end-pointu:

```python
from fastapi import APIRouter, FastAPI, HTTPException, Query
import psycopg2
import os
from dotenv import load_dotenv

router = APIRouter()

def connect_to_postgres():
    try:
        # Connect to PostgreSQL database
        load_dotenv()
        db_name=os.getenv('DATABASE_NAME')
        db_user=os.getenv('DATABASE_USER')
        db_pass=os.getenv('DATABASE_PASSWORD')
        db_host=os.getenv('DATABASE_HOST')
        db_port=os.getenv('DATABASE_PORT')

        conn = psycopg2.connect(
            database=db_name,
            user=db_user,
            password=db_pass,
            host=db_host,
            port=db_port
        )
        return conn
    except psycopg2.Error as e:
        raise HTTPException(status_code=500, detail=str(e))
```


FIIT STU
Ilkovičova 8, Bratislava

V tejto funkcii sa pripojím na databázu, a zadám do nej informácie potrebné k jej pripojeniu.

Tieto informácie mám uložené v súbore „.env“.

HTTP Volania

GET /v3/users/:user id/badge history

V tejto funkcií máme za úlohu vypísať všetky odznaky, ktoré získal užívateľ s ID user_id. Vypísať sa majú len také znaky, pred ktorými užívateľ poslal aspoň 1 správu. Taktiež ak viac odznakov ma rovnakú poslednú správu, tak to vráti len prvý získaný odznak. Odznaky nemôžu byť duplicitné, teda vypíšeme iba prvé získanie pre každý rôzny odznak.

```
FROM (
    SELECT *,
           ROW_NUMBER() OVER(ORDER BY post_created_at) AS position
    FROM (
        SELECT badge_posts.id AS badge_id,
               badge_posts.date AS badge_created_at,
               last_post_id_before_badge,
               post_title,
               badge_posts.name AS badge_type,
               posts.creationdate AS post_created_at,
               ROW_NUMBER() OVER(PARTITION BY last_post_id_before_badge ORDER BY badge_posts.date) AS badge_rank
        FROM (
            SELECT DISTINCT b.*,
                (SELECT p.id
                 FROM posts p
                 WHERE p.owneruserid = b.userid
                 AND p.creationdate < b.date
                 ORDER BY p.creationdate DESC
                 LIMIT 1) AS last_post_id_before_badge,
                (SELECT p.title
                 FROM posts p
                 WHERE p.owneruserid = b.userid
                 AND p.creationdate < b.date
                 ORDER BY p.creationdate DESC
                 LIMIT 1) AS post_title
            FROM badges b
            JOIN users u ON b.userid = u.id
            WHERE u.id = %s
        ) AS badge_posts
        JOIN posts ON posts.id = last_post_id_before_badge
        WHERE last_post_id_before_badge IS NOT NULL -- Filter out badges where there is no last post
    ) AS ranked_badges
    WHERE badge_rank = 1
) AS filtered_ranks;
```

Subquery „badge_posts“ vyberie potrebné atribúty z tabuľky badges a spolu s nimi aj „last_post_id_before_badge“, čo je subquery, ktorá mi vráti ID posledného postu pred získanim odznaku, a post_title, ktorá mi vráti názov tohto postu.

Subquery „ranked_badges“ pracuje na predošlej subquery, kde pridá kolónku „badge_rank“ ktorá hovorí o tom, koľkatý v poradí bol odznak získaný v prípade, že majú rovnaký „last_post_id_before_badge“ (tieto badges sú zoradené podľa dátumu získania)

Subquery „filtered_ranks“ vyfiltruje len tie badges, ktoré majú badge_rank rovný 1, teda boli získané ako prvé po odoslaní príspevku. Taktiež pridám kolónku „position“, ktorá vypočíta riadok na ktorom sa nachádzajú vyfiltrované odznaky (potrebné pre párovanie a výpis v outpute)

```
query = """
    SELECT last_post_id_before_badge,
       post_title,
       post_created_at,
       position,
       badge_id,
       badge_type,
       badge_created_at,
       position
FROM (
    SELECT *,
           ROW_NUMBER() OVER(ORDER BY post_created_at) AS position
    FROM (
```

Vonkajší SELECT vyberie potrebné atribúty zo subquery „filtered_ranks“. Výsledok sa mi potom uloží do pola, s ktorým narábam na potrebný výpis.

Príklad:

GET /v3/users/120 /badge history


```
{
  "items:": [
    {
        "last_post_id_before_badge": 7744,
        "post_title": "How do I make Firefox remember its window size?",
        "post_created_at": "2009-07-18T05:33:08.597000+02:00",
        "position": 1
    },
    {
        "badge_id": 5453,
        "badge_type": "Student",
        "badge_created_at": "2009-07-18T05:47:30.730000+02:00",
        "position": 1
    },
    {
        "last_post_id_before_badge": 8957,
        "post_title": null,
        "post_created_at": "2009-07-20T04:27:58.430000+02:00",
        "position": 2
    },
    {
        "badge_id": 6095,
        "badge_type": "Teacher",
        "badge_created_at": "2009-07-20T04:32:30.713000+02:00",
        "position": 2
    },
    {
```

Meno: Zaťovič Dominik (ID: 121058)
Predmet: Databázové systémy
Ročník: LS 2023/2024

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

V tejto funkcii chceme pre zadaný :tag vypočítať pre jednotlivé posty, ktoré maju viac ako :count komentárov, priemernú dobu odpovede medzi jednotlivými komentármi v rámci daného príspevku. Taktiež aj dĺžku odpovede medzi jednotlivými odpoveďami.

```
FROM (
    SELECT
        p.id AS post_id,
        p.title AS post_title,
        u.displayname AS user_displayname,
        c.text AS comment_text,
        p.creationdate AS post_created_at,
        c.creationdate AS created_at,
        EXTRACT(EPOCH FROM (c.creationdate - LAG(c.creationdate, 1, p.creationdate) OVER (PARTITION BY post_id ORDER BY c.creationdate))) AS time_difference

    FROM
        posts p
    JOIN
        post_tags pt ON p.id = pt.post_id
    JOIN
        tags t ON pt.tag_id = t.id
    JOIN
        comments c ON p.id = c.postid
    JOIN
        users u ON c.userid = u.id
    WHERE
        t.tagname = %s
        AND p.id IN (
            SELECT p.id
            FROM posts p
            JOIN comments c ON p.id = c.postid
            GROUP BY p.id
            HAVING COUNT(c.id) > %s
            ORDER BY MIN(c.creationdate)
        )
) AS Subquery
```

V subquery „Subquery“ vyberiem rôzne atribúty, ktoré budem neskôr potrebovať, a taktiež vypočítam dĺžku času medzi vytvorením daného komentáru a komentáru pred ním (time_difference) pomocou window function LAG, ktorá mi vráti hodnotu c.creationdate z predchádzajúceho riadku (komentáre su zoradené poďla času vytvorenia).

```
WHERE
    t.tagname = %s
    AND p.id IN (
        SELECT p.id
        FROM posts p
        JOIN comments c ON p.id = c.postid
        GROUP BY p.id
        HAVING COUNT(c.id) > %s
        ORDER BY MIN(c.creationdate)
    )
```

Táto subquery sa ešte filtuje podľa tagname (WHERE t.tagname = %s) a toho, či post.id sa nachádza v tom, čo nám vrátila ďalšia subquery, ktorá slúži na filtrovanie príspevkov s viac ako :count komentármi.

FIIT STU
Ilkovičova 8, Bratislava

```
sql_query = """
    SELECT
        post_id,
        post_title,
        user_displayname,
        comment_text,
        post_created_at,
        created_at,
        TO_CHAR(AGE(created_at, LAG(created_at, 1, post_created_at) OVER (PARTITION BY post_id ORDER BY created_at)), 'HH24:MI:SS') AS diff,
        TO_CHAR(INTERVAL '1 minute' * AVG(time_difference) OVER (PARTITION BY post_id ORDER BY created_at ROWS BETWEEN UNBOUNDED PRECEDING A
    FROM (
        SELECT
```

Nakoniec vonkajší SELECT nám vyberie rôzne atribúty, vrátane upravenia sekúnd time_difference do viac čiaťelného formátu (diff) a vypočítanie priemernej doby odpovede tiež v čitateľnom formáte (HH24:MI:SS).

Príklad:

GET /v3/tags/networking/comments?count=40


```
{
  "items": [
    {
      "post_id": 1034137,
      "post_title": "Did I just get hacked?",
      "user_displayname": "Jonno",
      "comment_text": "Yeah that doesn't look too good. I'm not an expert in Linux by any means, but somethings definitely t
      and failed. Are there any other logs in your auth.log? Any other means of remote admin? I've seen Mac's with VNC serve
      was downloading from are hosted in China somewhere.",
      "post_created_at": "2016-02-01T11:21:48.690000+01:00",
      "created_at": "2016-02-01T11:25:02.610000+01:00",
      "diff": "00:03:13",
      "avg": "00:03:13"
    },
    {
      "post_id": 1034137,
      "post_title": "Did I just get hacked?",
      "user_displayname": "David Schwartz",
      "comment_text": "The attack actually came from China.",
      "post_created_at": "2016-02-01T11:21:48.690000+01:00",
      "created_at": "2016-02-01T11:30:45.310000+01:00",
      "diff": "00:05:42",
      "avg": "00:04:28"
    },
    {
      "post_id": 1034137,
      "post_title": "Did I just get hacked?",
      "user_displayname": "vaid",
      "comment_text": "Yes but what is a Microsoft owned IP doing trying to breach a device across the internet?",
      "post_created_at": "2016-02-01T11:21:48.690000+01:00",
```

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

Táto funkcia slúži na získanie komentárov pre príspevky s tagom :tagname. Komentáre boli vytvorené ako k-te v poradí zoradených podľa dátumu vytvorenia pre :limit príspevkov zoradených podľa dátumu vytvorenia.

```
sql_query = """
    SELECT
        c.id AS comment_id,
        u.displayname AS displayname,
        p.body AS body,
        c.text AS text,
        c.score AS score
    FROM (
        SELECT
            p.id AS post_id,
            c.id AS comment_id,
            c.text AS comment_text,
            ROW_NUMBER() OVER (PARTITION BY p.id ORDER BY c.creationdate) AS comment_rank
        FROM
            posts p
        JOIN
            post_tags pt ON p.id = pt.post_id
        JOIN
            tags t ON pt.tag_id = t.id
        JOIN
            comments c ON p.id = c.postid
        JOIN
            users u ON c.userid = u.id
        WHERE
            t.tagname = %s
        ORDER BY
            p.creationdate
    ) AS RankedPosts
    JOIN
        comments c ON RankedPosts.comment_id = c.id
    JOIN
        posts p ON RankedPosts.post_id = p.id
    JOIN
        users u ON c.userid = u.id
    WHERE
        comment_rank = %s
    LIMIT %s
"""
```

Najprv tu mám subquery „RankedPosts," v ktorej vyberiem rôzne atribúty, ktoré budem potrebovať a taktiež každému príspevku vypočítam číslo riadku pre každý komentár (comment_rank) v zoradenom zozname podľa dátumu vytvorenia.

Táto subquery taktiež usporiadá príspevky podľa dátumu založenia a vyfiltruje len také, ktorých tag je ako ten, čo sme zadali

Vonkajší SELECT vyberie potrebné atribúty zo subquery „RankedPosts“ a spojí ich s ďalšími tabuľkami na základe ich prislúchajúcim ID.

Taktiež filtrujeme len tie, ktorých „comment_rank“ (ktorý je vypočítaný v subquery) sa rovná nám zadanému číslu.

Týmto účinne získame k-ty komentár v poradí pre každý príspevok zoradený podľa dátumu.

Na konci je LIMIT, ktorý je rovný počtu, pre koľko prvých postov chceme hlaďat ich k-ty komentár v poradí.

Priklad:

GET /v3/tags/linux/comments/2?limit=2

```
{
  "items": [
    {
      "id": 745427,
      "displayname": "Oliver Salzburg",
      "body": "<p>I am running Kubuntu Hardy Heron, with a dual monitor setup, and have VirtualBox on it running Windows XP in seamless mo
      monitor. \nHow can this be achieved?</p>\n",
      "text": "http://ubuntuforums.org/showthread.php?t=433359",
      "score": 0,
      "position": 2
    },
    {
      "id": 68008,
      "displayname": "Kyle Cronin",
      "body": "<p>I've been looking into tiling window managers because I'm tired of manually positioning and resizing all my windows wher
      I've investigated so far seem rather minimalist and seem mainly for rearranging terminal windows with clumsy keyboard navigation.</p
      on each partition. Bonus points if it works on Gnome. </p>\n",
      "text": "@bedwyr: not sure why Rich M tagged it \"osx\", I switched it to \"gnome\"",
      "score": 0,
      "position": 2
    }
  ]
}
```

Meno: Zaťovič Dominik (ID: 121058)
Predmet: Databázové systémy
Ročník: LS 2023/2024

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

Cieľom tejto funkcie je vypísať zoznam príspevkov pre post s ID postid. Zoznam začína samotným príspevkov nasledovaný :limit príspevkami pre ktoré platí, že náš prvotný post je parent týchto postov.

```
sql_query = """
    SELECT
        u.displayname,
        p.body,
        to_char(p.creationdate AT TIME ZONE 'UTC', 'YYYY-MM-DD"T"HH24:MI:SS"Z"') AS created_at_utc
    FROM
        posts p
    JOIN
        users u ON p.owneruserid = u.id
    WHERE
        p.id = %s OR p.parentid = %s
    ORDER BY
        p.creationdate
    LIMIT %s;
"""

cur.execute(sql_query, (postid, postid, limit))
```

Spravil som jednoduchú query, ktorá mi vyberie user displayname, text príspevku a čas vytvorenia príspevku z tabuľky posts, ktoré spojím s tabuľkou user na základe owneruserid v tabuľke „posts“ ktorým vyhovuje „id“ v tabuľke users.

Výpis platí pre také posty, kde ID postu je rovnaká ako ID, ktoré sme zadali alebo ID parent postu je také, aké sme zadali, a zoradíme ich podľa dátumu založenia.

Príklad:

GET /v3/posts/2154?limit=2


```
{
  "items": [
    {
      "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 sure t
      after that point I just reset the router( and possibly the cable modem) and that fixes things most of the time.</p>\n\n<p
      router.</p>\n\n<p>EDIT: Just to clarify, I was speaking more about reset as in turning the router off and on. Still, any
      probably be restarting </p>\n\n<p>Also, personally I usually have to deal with D-Link or Linksys home routers. I generall
      "created_at": "2009-07-15T12:51:57Z"
    },
    {
      "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 overv
      that the config is overwritten with the original one. But in some cases you have an updated router that isn't working for
      "created_at": "2009-07-15T12:54:48Z"
    }
  ]
}
```


FIIT STU
Ilkovičova 8, Bratislava

Záver

Tento projekt bol celkom náročný, keďže obsahoval zloźitejšie queriny a subqueriny, a zo zadania nie vždy bolo jasné, čo je od nás očakávané. Všetky queriny mi však fungujú, viacmenej tak ako majú, jediný problém som mal pri prvej úlohe, kde som nevypísal „type“ pre post a badge, keďže som to nevedel spraviť, okrem toho by malo byť všetko tak, ako má.