PDT protokol 6

Jakub Povinec

https://github.com/kuko6/tweets-mongo

(tento protokol som písal ako markdown a teda export do pdf nie je ideálny. Odporúčam radšej originál, ktorý je v `docs/protokol.md` a taktiež aj na `docs/protokol.md` alebo na githube)

Úloha 1

Môj dátový model obsahuje 2 kolekcie - **Authors** a **Tweets**. Tieto objekty som sa rozhodol rozdeliť, pretože k autorovi zvyčajne patrí veľké množstvo tweetov a teda sa neoplatí aby boli tieto dokumenty vnorené.

Dokumenty majú medzi sebou vzťah **one-to-many** s tým, že dokument tweetu obsahuje referenciu na autora. Druhá možnosť by bola aby samotný autor obsahoval list odkazov na jeho tweety, čo by ale podľa mňa nebolo ideálne, keďže používatelia majú na Twittery bežne veľké množstvo tweetov, ktoré sa taktiež pomerne často zväčšuje a aj podľa mongodb dokumentácie je v takomto prípade lepšie uchovávať referenciu v druhom dokumente. V tomto prípade je stále jednoduché nájsť tweety prislúchajúce danému autorovi.

Dokument Author

Tento dokument obsahuje rovnaké polia ako tabuľka `authors` v postgresql:

- `_id` - unikátny identifikátor autora, ktorý je uložený ako `string` namiesto `int64`
  - hodnota je ale rovnaká ako v postgresql
- `name` uložený ako `string`
- `username` uložený ako `string`
- `description` uložený ako `string`
- `followers_count` uložený ako `int`
- `following_count` uložený ako `int`
- `tweet_count` uložený ako `int`
- `listed_count` uložený ako `int`

```
{
  "_id": "846391998",
  "name": "Dr. Malcolm Davis",
  "username": "Dr_M_Davis",
  "description": "Senior Analyst – Australian Strategic Policy Institute – focus on Defence Strategy and
Capability issues including Space Policy and Space Security",
  "followers_count": 10525,
  "following_count": 7830,
  "tweet_count": 19949,
  "listed_count": 204
}
```

Dokument Tweet

Dokument pre tweety taktiež obsahuje do veľkej miery rovnaké polia ako tabuľka `conversations` v postgresql:

- `_id` - unikátny identifikátor tweetu, ktorý je uložený ako `string` namiesto `int64`
  - hodnota je ale rovnaká ako v postgresql
- `author_id` referencia na authora tweetu (vzťah **many-to-one**)
- `content` uložený ako `string`
- `possibly_sensitive` uložený ako `bool`

- `language` uložený ako `string`
- `source` uložený ako `string`
- `retweet_count` uložený ako `int`
- `reply_count` uložený ako `int`
- `like_count` uložený ako `int`
- `quote_count` uložený ako `int`
- `created_at` uložený ako `timestamp`
- `hashtags` uložený ako `list` stringov (hashtagov)
  - aj keď tweety môžu mať viac hashtagov, ktoré môžu byť rovnaké pre viaceré tweety, v tomto prípade sa ich neoplatí mať v samostatnej kolekcií, keďže ich je väčšinou len niekoľko
  - v tomto prípade je to vzťah **one-to-few**
- `links` uložený ako `list` objektov reprezentujúce odkazy (nazov, popis a url)
  - rovnako ako pri hashtagoch aj toto je vzťah **one-to-few**
- `context_annotations` uložený ako `list` objektov, ktoré obsahujú ďalšie dva objekty (`entity` a `domain`), kde každý má meno a popis
  - objekty `entity` aj `domain` môžu byť rovnaké vo viacerých tweetoch
  - na rozdiel od vzťahu medzi autormi a tweetmi sa v tomto prípade neoplatí tieto dokumenty rozdelovať, pretože aj keď ich môže byť veľké množstvo ich počet sa nemení
  - vzťah **one-to-few**
- `annotations` uložený ako `list` objektov reprezentujúcich anotácie
  - anotácie sú jedinečné pre každý tweet
  - vzťah **one-to-few**
- `conversation_references` uložený ako `list` objektov reprezentujúcich referencie
  - každý objekt obsahuje typ referencie a `_id` referencovaného tweetu
  - vzťah **one-to-many**

```
{
  "_id": "1496682988486414347",
  "author_id": "846391998",
  "content": "This is key - its a direct threat to #NATO of a #Russian military response against NATO,
including the implicit threat of use of #nuclear weapons. https://t.co/JrXUsmtNqZ",
  "possibly_sensitive": false,
  "language": "en",
  "source": "Twitter Web App",
  "retweet_count": 9,
  "reply_count": 1,
  "like_count": 13,
  "quote_count": 2,
  "created_at": {
    "$date": {
      "$numberLong": "1645672006000"
    }
  },
  "context_annotations": [
    {
      "entity": {
        "name": "North Atlantic Treaty Organization",
        "description": "North Atlantic Treaty Organization"
      },
      "domain": {
        "name": "Political Body",
        "description": "A section of a government, like The Supreme Court"
      }
    },
    ...
  ],
  "conversation_hashtags": [
    "NATO",
    "Russian",
    "nuclear"
  ],
  "annotations": [
    {
      "value": "NATO",
      "probability": 0.714,
      "type": "Organization"
    }
  ],
  "links": [
    {
      "url": "https://twitter.com/samagreene/status/1496679943689883649",
      "title": null,
      "description": null
    }
  ],
  "conversation_references": [
    {
      "type": "quoted",
      "id": "1496679943689883649"
    }
  ]
}
```


Úloha 2

MongoDB som spúštal cez docker, pomocou:

```
docker run \
    -p 27017:27017 \
    --name pdt-mongo \
    -v mongodata:/data/db \
    mongo:latest
```

Importovanie dát

Na denormalizovanie dát z postgresql som použil viacmenej rovnaké sql ako v 5. zadaní. Jediný rozdiel je hlavne v použití podmienky na vyfiltrovanie tweetov z **24.2.2022** a v `conversation_references`, z ktorých ma v tomto prípade zaujímal iba typ a id referencovaného tweetu. Taktiež som zmenil `id` tweetu, autora a referencovaného tweetu na `text`, keďže s pôvodnými hodnotami som mal problém pri vyhľadávaní.

```sql
SELECT
    c.id::text as _id, c.author_id::text, c."content", c.possibly_sensitive, c."language", c."source",
c.retweet_count, c.reply_count, c.like_count, c.quote_count, c.created_at,
    json_build_object(
        '_id', a.id::text, 'name', a."name", 'username', a.username,
        'description', a.description, 'followers_count', a.followers_count,
        'following_count', a.following_count, 'tweet_count', a.tweet_count, 'listed_count', a.listed_count
    ) author,
    COALESCE(ca.jsons, '[]') context_annotations,
    COALESCE(ch.jsons, '[]') conversation_hashtags,
    COALESCE(an.jsons, '[]') annotations,
    COALESCE(l.jsons, '[]') links,
    COALESCE(cr.jsons, '[]') conversation_references
FROM conversations c
JOIN authors a ON c.author_id = a.id
LEFT JOIN (
    SELECT ca.conversation_id,
        json_agg(json_build_object(
            'entity', json_build_object('name', ce."name", 'description', ce.description),
            'domain', json_build_object('name', cd."name", 'description', cd.description))
        ) jsons
    FROM context_annotations ca
    JOIN context_entities ce ON ca.context_entity_id = ce.id
    JOIN context_domains cd ON ca.context_domain_id = cd.id
    GROUP BY ca.conversation_id
) ca ON ca.conversation_id = c.id
LEFT JOIN (
    SELECT ch.conversation_id, json_agg(h.tag) jsons
    FROM conversation_hashtags ch
    JOIN hashtags h ON ch.hashtag_id = h.id
    GROUP BY ch.conversation_id
) ch ON ch.conversation_id = c.id
LEFT JOIN (
    SELECT an.conversation_id, json_agg(json_build_object('value', an."value", 'probability',
an.probability, 'type', an."type")) jsons
    FROM annotations an
    GROUP BY an.conversation_id
) an ON an.conversation_id = c.id
LEFT JOIN (
    SELECT l.conversation_id, json_agg(json_build_object('url', l.url, 'title', l.title, 'description',
l.description)) jsons
    FROM links l
    GROUP BY l.conversation_id
) l ON l.conversation_id = c.id
LEFT JOIN (
    SELECT
        cr.conversation_id,
        json_agg(json_build_object('type', cr."type", 'id', p.id::text)) jsons
    FROM conversation_references cr
    JOIN conversations p ON cr.parent_id = p.id
    WHERE timezone('UTC', p.created_at)::date = '2022-02-24'
    GROUP BY cr.conversation_id
) cr ON cr.conversation_id = c.id
WHERE timezone('UTC', c.created_at)::date = '2022-02-24';
```

Táto query vráti celkovo 1 924 682 záznamov.

Dáta som do mongodb importoval cez python skript. Tento skript spočíval v tom, že sa po 10000 vyberali z postgresql (pomocou `cursor.fetchmany(10000)`) záznamy, ktoré sa následne pomocou `collection.insert_many(data)` zapísali do mongo. Keďže sú oba dokumenty obsiahnuté v jednom riadku, bolo taktiež potrebné jednotlivé riadky prejsť a oddeliť autora od tweetu, na čo stačilo použiť len metódu `pop()`, keďže autor už bol uložený ako json.

Celé importovanie trvalo okolo 435s.

```python
def main():
    mongo = MongoClient('localhost', port=27017)
    mongo_db = mongo.get_database('pdt')
    tweets = mongo_db['tweets']
    authors = mongo_db['authors']

    conn = psycopg.connect(host='127.0.0.1', dbname='pdt', user='mac', password='', row_factory=dict_row)
    cur = query_data(conn)

    authors_data = []
    inserted_authors = set()
    tweets_data = []
    while True:
        rows = cur.fetchmany(10000)
        if len(rows) == 0: break

        for row in rows:
            author = row.pop('author')
            tweet = row

            # To avoid errors for inserting duplicate documents (authors)
            if not author['_id'] in inserted_authors:
                inserted_authors.add(author['_id'])
                authors_data.append(author)

            tweets_data.append(tweet)

        tweets.insert_many(tweets_data)
        authors.insert_many(authors_data)

        authors_data.clear()
        tweets_data.clear()

    cur.close()
    conn.close()
    mongo.close()
```

Úloha 3

Časť a - vypíšte posledných 10 tweetov pre autora, ktorý má username Newnews_eu

Keďže sú kolekcie pre autorov a tweety oddelené je potrebné ich najskôr spojiť. Na "joinovanie" kolekcií slúži v mongodb príkaz `lookup`, v ktorom je potrebné špecifikovať druhú tabuľku, polia na základe ktorých sa spájajú a alias. Ako ďalšie sa pomocou `match` vyfiltruje autor so správnym username. Ďalej sa pomocou `project` vyberú polia, ktoré sa majú vypísať. Pole `created_at` som zmenil na string, aby sa dal dátum exportovať v čitatelnejšom tvare. Nakoniec sa výsledky zoradia zostupne podľa `created_at` a vyberie sa len prvých 10.

```
db.tweets.aggregate([
  {
    '$lookup': {
      'from': 'authors',
      'localField': 'author_id',
      'foreignField': '_id',
      'as': 'author'
    }
  }, {
    '$match': {
      'author.username': 'Newnews_eu'
    }
  }, {
    '$project': {
      '_id': 1,
      'author_id': 1,
      'author.username': 1,
      'content': 1,
      'created_at': {
        '$dateToString': {
          'date': '$created_at',
          'timezone': 'UTC'
        }
      }
    }
  }, {
    '$sort': {
      'created_at': -1
    }
  }, {
    '$limit': 10
  }
])
```

Prvých 5 tweetov pre časť a

Celý výstup je v `outputs/uloha3_a.json`.

```
{
  "_id": "1496994513444913152",
  "author_id": "1495377954045665283",
  "content": "⚠️🧯#BREAKING | Clashes continue in Sum.\n\n#Ukraine #UkraineRussiaCrisis #UkraineRussiaCrisis
#Russia #RussiaUkraineConflict #RussiaUkraineCrisis #Putin #nowarinukraine https://t.co/bwzDx3zTLT",
  "author": [
    {
      "username": "Newnews_eu"
    }
  ],
  "created_at": "2022-02-24T23:44:39.000Z"
},
{
  "_id": "1496992794925322246",
  "author_id": "1495377954045665283",
  "content": "⚠️🧯#BREAKING | Russian soldier captured in the city of Sum.\n\n#Ukrayna #Ukraine
#RussiaUkraineConflict #Putin #UkraineRussiaCrisis #NoWar #nowarinukraine #UkraineRussie https://t.co/
gtIGWdHzOF",
  "author": [
    {
      "username": "Newnews_eu"
    }
  ],
  "created_at": "2022-02-24T23:37:49.000Z"
},
{
  "_id": "1496988583009603587",
  "author_id": "1495377954045665283",
  "content": "⚠️#BREAKING | President of Ukraine Zelensky: \"Ukrainian soldiers on Snake Island were all
killed because they refused to surrender. Our soldiers on Snake Island died fighting
heroically.\"\n\n#Ukraine #UkraineRussie #worldwar3 #NoWar #nowarinukraine #RussiaUkraineConflict https://
t.co/vxbA7dmqXm",
  "author": [
    {
      "username": "Newnews_eu"
    }
  ],
  "created_at": "2022-02-24T23:21:05.000Z"
},
{
  "_id": "1496986599812485121",
  "author_id": "1495377954045665283",
  "content": "⚠️#BREAKING | Zelensky: The enemy has set me as target number one and my family as target
number two.\n\n#Ukraine #RussiaInvadedUkraine #worldwar3 #NoWar #nowarinukraine #Ukraina
#RussiaUkraineConflict #UkraineRussie",
  "author": [
    {
      "username": "Newnews_eu"
    }
  ],
  "created_at": "2022-02-24T23:13:12.000Z"
},
{
  "_id": "1496985447423045648",
  "author_id": "1495377954045665283",
  "content": "⚠️#BREAKING | President of Ukraine Zelensky: "Today I asked 27 European leaders whether
Ukraine will join NATO, I directly asked. Everyone is afraid. They are not answering.\"\n\n#Ukraine #Ukraina
#UkraineRussie #worldwar3 #NoWar #nowarinukraine #RussiaInvadedUkraine https://t.co/QK17mXrWfy",
  "author": [
    {
      "username": "Newnews_eu"
    }
  ],
  "created_at": "2022-02-24T23:08:37.000Z"
}
```

Časť b - vypíše posledných 10 retweetov pre tweet, ktorý má id 1496830803736731649

V tomto prípade stačí len pomocou `match` vybrať tweety, ktoré majú v poli `conversation_references`, id vybraného tweetu a daná referencia je typu `retweeted`. Ďalej je už tento dopyt rovnaký ako ten predchádzajúci.

```
db.tweets.aggregate([
  {
    '$match': {
      'conversation_references.id': '1496830803736731649',
      'conversation_references.type': 'retweeted'
    }
  }, {
    '$project': {
      '_id': 1,
      'content': 1,
      'created_at': {
        '$dateToString': {
          'date': '$created_at',
          'timezone': 'UTC'
        }
      },
      'conversation_references': 1
    }
  }, {
    '$sort': {
      'created_at': -1
    }
  }, {
    '$limit': 10
  }
])
```

Prvých 5 tweetov pre časť b

Celý výstup je v `outputs/uloha3_b.json`.

```
{
  "_id": "1496997912890458112",
  "content": "RT @Newnews_eu: ⚠️#BREAKING | A father who sent his family to a safe zone bid farewell to his
little girl and stayed behind to fight .....…",
  "conversation_references": [
    {
      "type": "retweeted",
      "id": "1496830803736731649"
    }
  ],
  "created_at": "2022-02-24T23:58:09.000Z"
},
{
  "_id": "1496997818246254593",
  "content": "RT @Newnews_eu: ⚠️#BREAKING | A father who sent his family to a safe zone bid farewell to his
little girl and stayed behind to fight .....…",
  "conversation_references": [
    {
      "type": "retweeted",
      "id": "1496830803736731649"
    }
  ],
  "created_at": "2022-02-24T23:57:47.000Z"
},
{
  "_id": "1496997095584067590",
  "content": "RT @Newnews_eu: ⚠️#BREAKING | A father who sent his family to a safe zone bid farewell to his
little girl and stayed behind to fight .....…",
  "conversation_references": [
    {
      "type": "retweeted",
      "id": "1496830803736731649"
    }
  ],
  "created_at": "2022-02-24T23:54:55.000Z"
},
{
  "_id": "1496997095936561161",
  "content": "RT @Newnews_eu: ⚠️#BREAKING | A father who sent his family to a safe zone bid farewell to his
little girl and stayed behind to fight .....…",
  "conversation_references": [
    {
      "type": "retweeted",
      "id": "1496830803736731649"
    }
  ],
  "created_at": "2022-02-24T23:54:55.000Z"
},
{
  "_id": "1496997060637302787",
  "content": "RT @Newnews_eu: ⚠️#BREAKING | A father who sent his family to a safe zone bid farewell to his
little girl and stayed behind to fight .....…",
  "conversation_references": [
    {
      "type": "retweeted",
      "id": "1496830803736731649"
    }
  ],
  "created_at": "2022-02-24T23:54:46.000Z"
}
```