Slovenská Technická Univerzita v Bratislave
Fakulta Informatiky a Informačných Technológií

Pokročilé databázové technológie

Zadanie 6 – MongoDB

Marek Adamovič

Cvičenie: Piatok 11:00 – 12:40
Ing. Ján Balažia, PhD.

2022/2023

Slovenská Technická Univerzita v Bratislave
Fakulta Informatiky a Informačných Technológií

Obsah

1. Dátový model

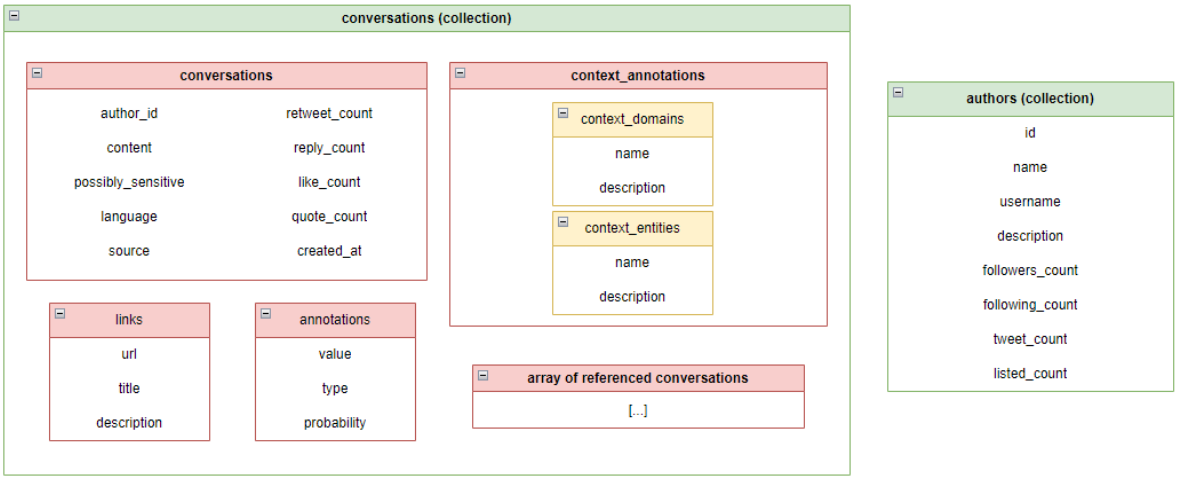
Otázka:

Navrhnite dátový model (kolekcie a formát dokumentov) v MongoDB pre dataset tweetov, ktorý bude využívaný mobilnou aplikáciou, ktorá bude:

a. Zobrazovať tweety jednotlivých používateľov vo forme feedov

b. Zobrazovať jednotlivé tweety a ich retweets

Odpoveď:



Obrázok 1 Dátový model Mongo

Pri tvorbe dátového modelu (obrázok č.1) sme zvolili prístup, v ktorom väčšina dát bude obsiahnutých v jednej collection (embedded prístup), a to v conversations. To hlavne z dôvodu, že tieto dáta sa v našej pomyselnej aplikácií nebudú (často) meniť a bez samotných tweetov by nemali veľký zmysel. To sa však netýka autorov-užívateľov, ktorých ukladáme (referenced prístup) do vlastnej collection authors. Je to najmä kvôli tomu, aby sme pri zmene údajov autora (čo je prípad, ktorý nastáva veľmi často) nemuseli meniť tieto údaje v každom jednom tweete, ktorý tento autor postol. Taktiež majú dáta autorov zmysel aj bez dát o tweetoch, čo je ďalší dôvod, prečo je dobré mať ich vo vlastnej collection. Čo sa týka referencovaných tweetov, tiež sme zvolili embedded prístup, keďže MongoDB neumožňuje referenced prístup v rámci jednej collection. Teda ak by sme chceli využiť referenced prístup, museli by sme mať vlastnú kolekciu pre referencované tweety. To by veľmi nedávalo zmysel, mať dve kolekcie na tweety, kvôli čomu sme zvolili embedded prístup (s tým, že referencované tweety nebudú mať len základné informácie o nich, aby nám napríklad nevznikali nekonečný embedding v rámci jednej collection). Na obrázku č.2,3 si vieme pozrieť príklad štruktúry importovaných dát.

```
_id: 224
name: "Dave  Pell"
username: "davepell"
description: "Managing  Editor,  Internet."
followers_count: 60064
following_count: 1054
tweet_count: 54618
listed_count: 2016
```


Obrázok 2 Príklad štruktúry dát z collection authors


```
_id: 1496779955958603778
content: "RT  @dlepeska:  #Ukraina"
possibly_sensitive: false
language: "qht"
source: "Twitter  for  Android"
retweet_count: 2
reply_count: 0
like_count: 0
quote_count: 0
created_at: "2022-02-24T10:32:04+01:00"
author_id: 182981546
links: Array
annotations: Array
context_domains: Array
  0: Object
      id: 123
      name: "Ongoing  News  Story"
      description: "Ongoing  News  Stories  like  'Brexit'"
  1: Object
      id: 156
      name: "Cities"
      description: "Cities  around  the  world"
context_entities: Array
conversation_references: Array
  0: Object
      id: 96560
      type: "retweeted"
      parent_conversation_author_id: 19939989
      parent_conversation_author_name: "David  Lepeska"
      parent_conversation_author_username: "dlepeska"
      parent_conversation_id: 1496779491594874881
      parent_conversation_content: "#Ukraina  https://t.co/5SwNvpqIhQ"
      parent_conversation_created_at: "2022-02-24T10:30:14+01:00"
```


Obrázok 3 Príklad štruktúry dát z collection covnersations

2. Import

Otázka:

Nainštalujte alebo využite online inštanciu MongoDB servera, do ktorého importujte všetky tweets (a s nimi spojené data – anotácie, referencie, odkazy a informácie o kontexte) zo dňa 24.02.2022.

Odpoveď:

Na túto úlohu sme využili online inštanciu MongoDB servera. Najskôr sme exportovali dáta v požadovanej štruktúre pomocou query (obrázok č.4 -> authors collection, obrázok č.5 -> conversations collection) z postgres databázy.

```
COPY(
    EXPLAIN ANALYZE SELECT json_build_object(
        '_id', author.id,
        'name', author.name,
        'username', author.username,
        'description', author.description,
        'followers_count', author.followers_count,
        'following_count', author.following_count,
        'tweet_count', author.tweet_count,
        'listed_count', author.listed_count
    )
    FROM conversations c
    JOIN authors author ON author.id = c.author_id
    WHERE c.created_at >= '2022-02-24 00:00:00' AND c.created_at < '2022-02-25 00:00:00'
    GROUP BY author.id
    --LIMIT 50
) TO 'D:\skola2022_2023\PDT\zadanie6\authors.json' WITH (FORMAT CSV, QUOTE ' ');

-- Create date index
--CREATE INDEX conversation_created ON conversations USING btree(created_at);
```

Obrázok 4 Export dát potrebných pre author collection

```
COPY(
    SELECT json_build_object(
        '_id', c.id,
        'content', c.content,
        'possibly_sensitive', c.possibly_sensitive,
        'language', c.language,
        'source', c.source,
        'retweet_count', c.retweet_count,
        'reply_count', c.reply_count,
        'like_count', c.like_count,
        'quote_count', c.quote_count,
        'created_at', c.created_at,
        'author_id', c.author_id,
        'links', l_l,
        'annotations', a_l,
        'context_domains', cd_l , -- sometimes null
        'context_entities', ce_l, -- sometimes null
        'conversation_references', cr_l
    )
    FROM conversations c
    LEFT JOIN LATERAL(
        SELECT COALESCE(json_agg(json_build_object('id', l.id, 'url', l.url, 'title', l.title, 'description', l.description)), '[]'::json) l_l
        FROM links l
        WHERE l.conversation_id = c.id
    ) AS l_l ON true
    LEFT JOIN LATERAL(
        SELECT COALESCE(json_agg(json_build_object('id', a.id, 'value', a.value, 'type', a.type, 'probability', a.probability)), '[]'::json) a_l
        FROM annotations a
        WHERE a.conversation_id = c.id
    ) AS a_l ON true
    LEFT JOIN LATERAL(
        SELECT COALESCE(json_agg(json_build_object('id', cd.id, 'name', cd.name, 'description', cd.description)), '[]'::json) cd_l
        FROM context_domains cd
        JOIN context_annotations ca ON ca.context_domain_id = cd.id AND ca.conversation_id = c.id
    ) AS cd_l ON true
    LEFT JOIN LATERAL(
        SELECT COALESCE(json_agg(json_build_object('id', ce.id, 'name', ce.name, 'description', ce.description)), '[]'::json) ce_l
        FROM context_entities ce
        JOIN context_annotations ca ON ca.context_domain_id = ce.id AND ca.conversation_id = c.id
    ) AS ce_l ON true
    LEFT JOIN LATERAL(
        SELECT COALESCE(json_agg(json_build_object(
            'id', cr.id,
            'type', cr.type,
            'parent_conversation_author_id', author2.id,
            'parent_conversation_author_name', author2.name,
            'parent_conversation_author_username', author2.username,
            'parent_conversation_id', c2.id,
            'parent_conversation_content', c2.content,
            'parent_conversation_created_at', c2.created_at
            )
        ), '[]'::json) cr_l
        FROM conversation_references cr
        JOIN conversations c2 ON c2.id = cr.parent_id
        JOIN authors author2 ON author2.id = c2.author_id
        WHERE (c2.created_at >= '2022-02-24 00:00:00' AND c2.created_at < '2022-02-25 00:00:00') AND cr.conversation_id = c.id
    ) AS cr_l ON true
    WHERE c.created_at >= '2022-02-24 00:00:00' AND c.created_at < '2022-02-25 00:00:00'
) TO 'D:\skola2022_2023\PDT\zadanie6\conversations.json' WITH (FORMAT CSV, QUOTE ' ');
```

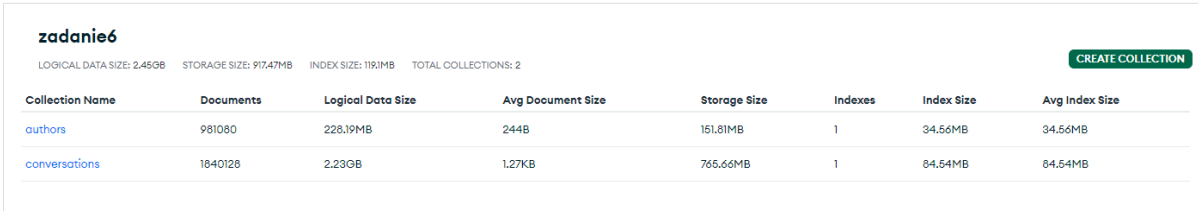
Obrázok 5 Export dát potrebných pre conversations collection

Keď sme dokončili export dát z postgresu, pomocou python scriptu (obrázok č.6) sme nahrali dáta do online MongoDB clusteru. Stav databázy po importe si vieme pozrieť na obrázku č.7.

mongo.py

```python
from pymongo import MongoClient
import json
import time

BATCHSIZE = 5000
client = MongoClient('mongodb+srv://admin:admin@cluster0.fqbgwci.mongodb.net/test')

db = client['zadanie6']


def import_conversations():
    collection = db['conversations']
    with open('conversations.json', 'r', encoding='utf8') as f:
        x = 0
        start = time.time()
        documents = []
        for line in f:
            x += 1
            documents.append(json.loads(line))

            if len(documents) == BATCHSIZE:
                collection.insert_many(documents)
                print(x)
                documents = []

        #send final data
        collection.insert_many(documents)
        print('Final time:')
        print(time.time() - start)
    return

def import_authors():
    collection = db['authors']
    with open('authors.json', 'r', encoding='utf8') as f:
        x = 0
        start = time.time()
        documents = []
        for line in f:
            x += 1
            documents.append(json.loads(line))

            if len(documents) == BATCHSIZE:
                collection.insert_many(documents)
                print(x)
                documents = []

        #send final data
        collection.insert_many(documents)
        print(x)
        print('Final time:')
        print(time.time() - start)
    return

import_conversations()
import_authors()
```

Obrázok 6 Python script pre import dát na MongoDB cluster



zadanie6

LOGICAL DATA SIZE: 2.45GB STORAGE SIZE: 917.47MB INDEX SIZE: 119.1MB TOTAL COLLECTIONS: 2

CREATE COLLECTION

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
authors	981080	228.19MB	244B	151.81MB	1	34.56MB	34.56MB
conversations	1840128	2.23GB	1.27KB	765.66MB	1	84.54MB	84.54MB

Obrázok 7 Stav databázy po importe

3. Dopyty

Otázka:

Napíšte dotaz, ktorý nad importovanou databázou:

a. Vypíše posledných 10 tweetov pre autora, ktorý má username Newnews_eu

b. Vypíše posledných 10 retweetov pre tweet, ktorý má id 1496830803736731649.

Odpoveď:

Pre úlohu a) sme vytvorili dopyt (obrázok č.8), ktorý najskôr spojí collection conversation s collection authors na základe autorovho id (podobne ako join pri relačných databázach). Následne pomocou unwind nahradí pole autorov (v tomto prípade je autor vždy len jeden, takže je to skôr kozmetická úprava, vďaka ktorej vieme pracovať priamo so záznamom a nie s poľom jedného prvku) prvkom autora z tohto poľa. Pomocou match ponecháme záznamy daného autora (Newnews_eu) a v závere usporiadame výsledky podľa dátumu created_at a ponecháme 10 výsledkov.

```
first_query = '''[
    {
        "$lookup": {
            "from": "authors",
            "localField": "author_id",
            "foreignField": "_id",
            "as": "author"
        }
    },
    {"$unwind": "$author"},
    {"$match": {"author.username": "Newnews_eu"}},
    {"$sort": {"created_at": -1}},
    {"$limit": 10}
]'''
results1 = collection.aggregate(json.loads(first_query))
for r in results1:
    print(r)
```

Obrázok 8 Query pre vypísanie posledných 10 tweetov autora Newnews_eu

Pre úlohu b) sme vytvorili dopyt (obrázok č.9), ktorý rozbalí pole s referenciami pomocou unwind a pozrie sa, či v týchto referenciách je retweet konverzácie s požadovaným id. Následne tieto výsledky zoradí podľa času created_at a vráti prvých 10.

```
second_query = '''[
    {"$unwind": "$conversation_references"},
    {"$match":{
        "$and":[
        {"conversation_references.parent_conversation_id": 1496830803736731649},
        {"conversation_references.type": "retweeted"}
        ]
    }},
    {"$sort": {"created_at": -1}},
    {"$limit": 10}
]'''
results2 = collection.aggregate(json.loads(second_query))
for r in results2:
    print(r)
```

Obrázok 9 Posledných 10 retweetov tweetu s ID 1496830803736731649