Универзитет „Св. Кирил и Методиј“ во Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Проектна задача по предметот:
Неструктурирани бази на податоци и XML

Тема:
Document бази на податоци

Изработиле:
Јован Калајџиев 206013
Сара Николова 213034

Датум: 14.09.2024

Содржина

1. Вовед

Во оваа проектна задача ќе се споредат перформансите на извршување на прашалници врз неструктурирана и структурирана база на податоци, конкретно MongoDB и PostgreSQL. При што и во двете бази на податоци ќе биде вметнато истото податочно множество, со таа разлика што податочниот модел ќе биде различен. За структурираната база ќе биде направен релационен модел според кој ќе бидат креирани табелите, а во неструктурираната база податоците ќе бидат вметнати во неколку колекции од документи, при што секоја колекција ќе содржи различен тип на документи (пр.: business, user, review). Ќе бидат извршени и поедноставни и покомплексни прашалници и на крај ќе бидат споредени времињата на извршување.

2. Методологија

Податочното множество кое е искористено може да се види на следниот линк: https://www.yelp.com/dataset. Станува збор за 5 фајлови кои во нив содржат повеќе json документи и во секој фајл се претставени различен тип на ентитети. Ентитетите се следниве: business, user, review, checkin, tip. Business документите содржат детали за одредени бизниси, поточно име на биснисот, негова локација, категорија, време во кое е отворен и сл. атрибути, како и стринг кој што е уникатен за секој бизнис. User документите исто така содржат идентификациски стринг, име, датум на креирање на user-от, листа на пријатели како и број на рецензии што корисникот ги оставил за бизнисите. Review документите претставуваат оцени од страна на корисниците за бизнисите и содржат user_id, business_id, review_id, број на ѕвезди со кои корисникот го оценил бизнисот, датум и коментар. Tip документите се слични на review документите со тоа што овие не содржат број на ѕвезди и претставуваат совет за бизнисите наместо нивна оцена. Checkin документите претставуваат листа од временски печати на кои корисниците се чекирале во одреден локал.

Во неструктурираната база документите се внесени такви какви што се во фајловите, со тоа што за секој тип на ентитет има посебна колекција, а пак за структурираната база е направен релационен модел кој потоа е пресликан во табели во базата. Внесот на податоците е направен со python скрипти, исечоци од скриптите ќе бидат прикажани во продолжение. Прво ќе биде прикажан по еден пример од секој тип на ентитет:

business


```
_id: ObjectId('66d235cd194e131e630a78fa')
business_id : "mpf3x-BjTdTEA3yCZrAYPw"
name : "The UPS Store"
address : "87 Grasso Plaza Shopping Center"
city : "Affton"
state : "MO"
postal_code : "63123"
latitude : 38.551126
longitude : -90.335695
stars : 3
review_count : 15
is_open : 1
▾ attributes : Object
    BusinessAcceptsCreditCards : "True"
categories : "Shipping Centers, Local Services, Notaries, Mailbox Centers, Printing …"
▾ hours : Object
    Monday : "0:0-0:0"
    Tuesday : "8:0-18:30"
    Wednesday : "8:0-18:30"
    Thursday : "8:0-18:30"
    Friday : "8:0-18:30"
    Saturday : "8:0-14:0"
```


checkin


```
_id: ObjectId('66d23773194e131e630cc444')
business_id : "--0iUa4sNDFiZFrAdIWhZQ"
date : "2010-09-13 21:43:09, 2011-05-04 23:08:15, 2011-07-18 22:30:31, 2012-09…"
```


review


```
_id: ObjectId('66d241c4194e131e633afb49')
review_id : "KU_O5udG6zpxOg-VcAEodg"
user_id : "mh_-eMZ6K5RLWhZyISBhwA"
business_id : "XQfwVwDr-v0ZS3_CbbE5Xw"
stars : 3
useful : 0
funny : 0
cool : 0
text : "If you decide to eat here, just be aware it is going to take about 2 h…"
date : "2018-07-07 22:09:11"
```

user


```
_id: ObjectId('66d23adf194e131e631ca610')
user_id : "qVc8ODYU5SZjKXVBgXdI7w"
name : "Walker"
review_count : 585
yelping_since : "2007-01-25 16:47:26"
useful : 7217
funny : 1259
cool : 5994
elite : "2007"
friends : "NSCy54eWehBJyZdG2iE84w, pe42u7DcCH2QmI81NX-8qA, EjlCGf14tYMPJ0rsrL703w…"
fans : 267
average_stars : 3.91
compliment_hot : 250
compliment_more : 65
compliment_profile : 55
compliment_cute : 56
compliment_list : 18
compliment_note : 232
compliment_plain : 844
compliment_cool : 467
compliment_funny : 467
compliment_writer : 239
compliment_photos : 180
```


tip

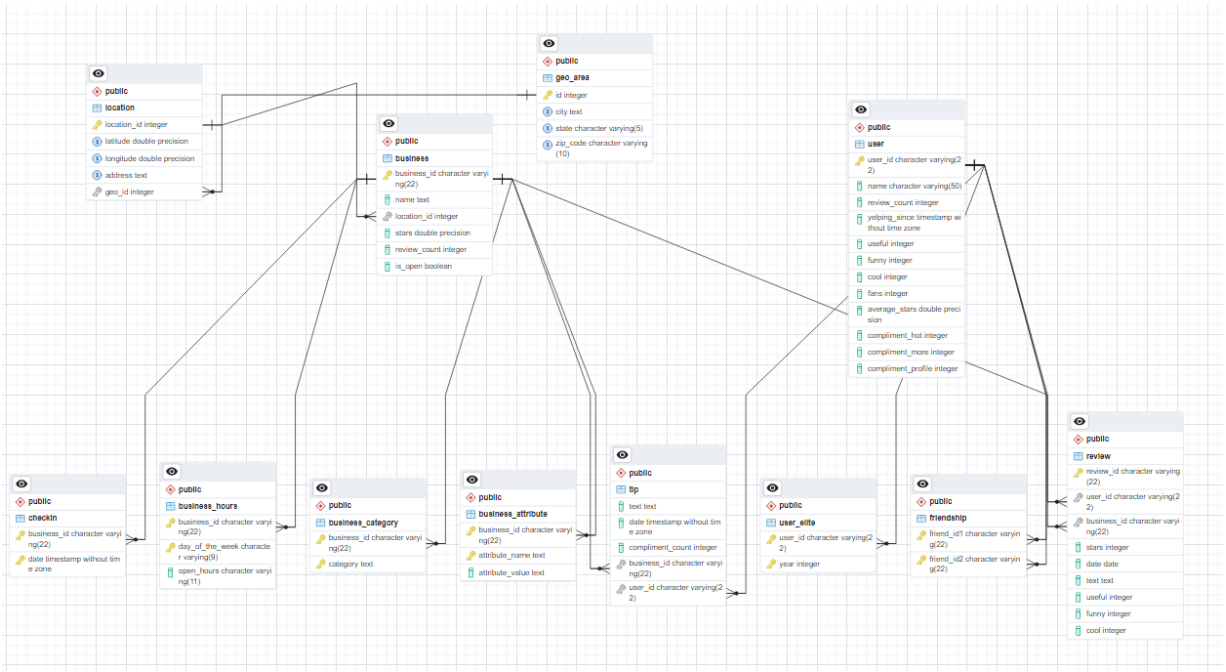

```
_id: ObjectId('66d237e7194e131e630ec79d')
user_id : "AGNUgVwnZUey3gcPCJ76iw"
business_id : "3uLgwr0qeCNMjKenHJwPGQ"
text : "Avengers time with the ladies."
date : "2012-05-18 02:17:21"
compliment_count : 0
```


Како што е спомнато погоре, во неструктурираната база документите се внесени такви какви што се, во посебни колекции, искористена е pymongo библиотеката, во продолжение дел од python скриптата во која се прави поврзување со базата и се прикажува внесот на податоците од еден фајл:

```
In [4]: import json
        from pymongo import MongoClient

        client = MongoClient("mongodb://localhost:27017/")

        db = client.business_reviews
```

```
In [5]: collection = db.business

        with open(".\\archive (2)\\yelp_academic_dataset_business.json", 'r', encoding='utf-8') as file:
            for line in file:
                document = json.loads(line.strip())
                collection.insert_one(document)
```

За структурираната база прво е направена нормализација на податоците, во продолжение е прикажан крајниот EP дијаграм:



Врз база на овој дијаграм е напишана sql скрипта за креирање на табелите која ќе биде прикачена заедно со документацијата. За внес на податоците е искористена psycopg2 библиотеката, во продолжение дел од python скриптата за воспоставување конекција со базата и вметнување на податоците во една од табелите:

In [1]:
```
import json
import psycopg2
```

In [2]:
```
conn_params = {
    'dbname': 'business_reviews',
    'user': 'postgres',
    'password': 'postgres',
    'host': 'localhost',  # Or '127.0.0.1'
    'port': '5432'  # Default PostgreSQL port
}

conn = psycopg2.connect(**conn_params)
print("Connection established")
cur = conn.cursor()
```

```
Connection established
```

```
with open(".\\archive (2)\\yelp_academic_dataset_checkin.json", 'r', encoding='utf-8') as file:
    for line in file:
        if line.strip():
            doc = json.loads(line)
            business_id = doc.get("business_id")
            dates = doc.get("date")
            for date in dates.split(", "):
                cur.execute(f"INSERT INTO checkin (business_id, date) VALUES ('{business_id}', '{date}') ON CONFLICT DO NOTHING")
                conn.commit()
```

Поради обемноста на прашалниците, особено за MongoDB базата, овде прашањата ќе бидат напишани само на македонски јазик а во посебен фајл ќе бидат испратени и самите прашата наменети за базите, заедно со нивните времиња на извршување.

1. Најди ги сите бизниси кои се сеуште отворени

Ова е едно од основните прашања со само едно филтрирање, без спојувања и агрегации.

2. Најди ги id на сите бизниси кои што се со категорија 'Pets'

Уште едно едноставно прашање со филтрирање.

3. За секој град пронајди ги трите најдобро оценети бизниси

За ова прашање во структурираната база ќе биде потребно спојување на податоци (бидејќи локациите се во посебни табели од табелата за биснис) како и користење на функцијата rank, а за неструктурираната нема да има потреба од спојување бидејќи градот се наоѓа во бизнис документите, но ќе има потреба од агрегација.

4. Најди просек од оцени за секој бизнис

За ова прашање нема потреба од спојувања но има агрегација и е со средна комплексност, целта беше да се спореди перформансот во случај кога и во двете бази нема потреба од спојување.

5. Најди ги сите корисници со повеќе од 3 пријатели и барем една оцена од 5 ѕвезди

Ова прашање е едно од прашањата со средна комплексност, и за двете бази е потребно спојување како и групирање и филтрирање на групите.

6. За секој град, најди го бројот на бизниси во него, вкупниот број на оцени како и просечниот број на ѕвезди од оцени за бизнисите во тој град.

За ова прашање во неструктурираната база се потребни едно спојување и неколку агрегации, за ова прашање времето на извршување е најголемо. За структурираната база се потребни неколку спојувања и неколку агрегации.

7. За бизнисите базирани во Santa Barbara, најди го просечниот број на ѕвезди од оцени од корисници кои имаат оставено над 50 оцени, резултатите подреди ги во опаѓачки редослед според просечниот број на ѕвезди

Ова прашање е исто така меѓу најкомплексните, и за двете бази се потребни спојувања и агрегации, како и подредување во опаѓачки редослед на резултатите.

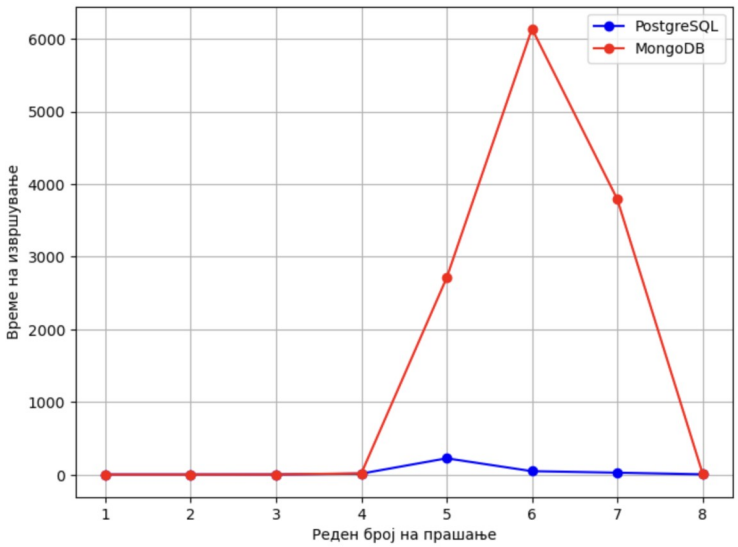
8. Најди ги заедничките пријатели на двајца корисници.

За ова прашање во структуриранта база ќе биде потребно спојување на податоци, а за неструктурираната нема да има потреба.

3. Добиени резултати

Во прилог табела и график со времињата на извршување на прашалниците во секунди:

Реден број на прашање	1	2	3	4	5	6	7	8
PostgreSQL	0.168	0.153	0.9	12.706	223.620	46.319	25.675	2.966
MongoDB	0.005	0.016	1.102	15.767	2709.189	6138.526	3791.217	13.168



Како што може да се примети, времињата на извршување во кои се потребни едно или повеќе спојувања се драстично поголеми во MongoDB, за прашањата со едноставни филтрирања пак, MongoDB покажува подобри резултати. За некои од прашањата со средна комплексност, базите имаат слични времиња на извршување.

4. Заклучок

Во оваа проектна задача беа споредени перформансите на една структурирана и една неструктурирана база на податоци, PostgreSQL и MongoDB, при што беа извршени неколку едноставни и неколку покомплексни прашалници кои вклучуваа агрегации и спојување на податоците. Според добиените резултати, PostgreSQL базата има подобри перформанси кога е потребно спојување или некаква агрегација на податоците, а MongoDB базата пак има подобри перформанси врз едноставни прашалници. Во понатамошните експериментирање би можеле да се имплементираат различни нивоа на агрегација за податочниот модел во MongoDB, а би можеле да се креираат и повеќе индекси во двете бази.

5. Користена литература

- https://www.postgresql.org/docs/
- https://www.mongodb.com/docs/