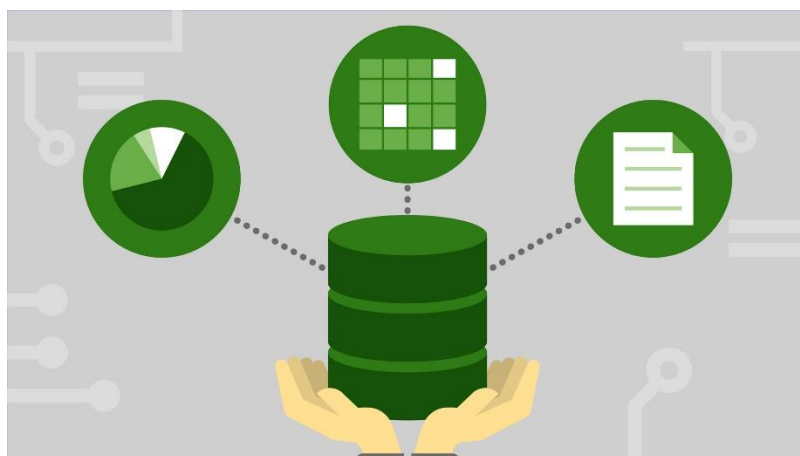


به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

دستورکار شماره ۹

شماره دانشجویی

۸۱۰۱۹۶۴۴۳

تیر ۱۴۰۰

هومان چمنی

گزارش فعالیت‌های انجام شده

منطقاً طبق کوورهایی که قرار است زده شود و توضیحاتی که در داک پروژه آمده است، نیاز به طراحی دو جدول داریم. یکی برای نگه‌داشتن tweet_by_hashtag و دیگری نیز tweet_by_user تا کارهایی که لازم داریم انجام شوند. هر دو جدول نیز شامل فیلد زمان ارسال می‌شوند و برای نیازهایی که تعریف کرده‌ایم در آخر هم یک ترتیب‌بندی ای طبق همان زمان ارسال روی دو جدول تعریف می‌کنیم. طبیعتاً قبل از شروع تعریف کردن این دو جدول می‌بایست یک فضای کلید مربوط به توینتر در کاساندار تعریف کنیم. برای سادگی و خب اینکه روی یک دستگاه کار می‌کنیم و مواردی مثل بازیابی داده‌ها نیز فعلاً برایمان مهم نیست از استراتژی ساده استفاده کرده و دفعات تکرار را هم ۱ می‌گذاریم:

```
houmch@houmch-VirtualBox:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> create KEYSPACE tweet WITH replication = {'class': 'SimpleStrategy', 're
plication_factor': 1};
cqlsh>
```

کد مربوط به ساختن جداول نیز به صورت زیر است:

```
cqlsh:tweet> CREATE TABLE if NOT EXISTS t_by_hashtag (
... tweet_id int,
... send_time timestamp,
... hashtag text,
... PRIMARY KEY (hashtag, send_time)
... ) WITH CLUSTERING ORDER BY (send_time ASC);
```

```
cqlsh:tweet> CREATE TABLE if NOT EXISTS t_by_username (
... tweet_id int,
... send_time timestamp,
... username text,
... PRIMARY KEY (username, send_time)
... ) WITH CLUSTERING ORDER BY (send_time ASC);
cqlsh:tweet> S
```

حال به توضیح کد پایتون بخش اول می‌پردازیم. در این بخش با استفاده از سشن و بررسی هشگ‌هایی که در توینت وجود دارند اقدام به اضافه کردن می‌کنیم. دو جدول داریم که باید پر شوند پس دوباره از session.execute استفاده می‌کنیم و اطلاعات هر کدام رو مطابق ساختار جدولی که تعریف کرده‌ایم هندل می‌کنیم. برای کار کردن با سشن هم ابتدا باید آن را به کلاستر متصل کنیم که ساختار آن به صورت زیر می‌باشد.

```
session = cluster.connect(cluster_name, wait_for_all_pools=True)
```

```
def make_int(inp):
    return int(inp)

for hash in tweet['hashtags']:
    session.execute(
        """
        INSERT INTO t_by_hashtag (tweet_id, send_time, hashtag)
        VALUES (%(tweet_id)s, %(send_time)s, %(hashtag)s)
        """,
        {
            'tweet_id' : make_int(tweet['id']),
            'send_time' : tweet['sendTime'],
            'hashtag' : hash
        }
    )

session.execute(
    """
    INSERT INTO t_by_username (tweet_id, send_time, username)
    VALUES (%(tweet_id)s, %(send_time)s, %(username)s)
    """,
    {
        'tweet_id' : make_int(tweet['id']),
        'send_time' : tweet['sendTime'],
        'username' : tweet['senderUsername']
    }
)
```

برنامه دومی هم که نوشته شد مربوط به زدن کووری می‌شود. برای این قسمت ابتدا تایم شروع و پایان را به صورت year/month/day hour:minute به عنوان ورودی از کاربر می‌گیریم. سپس باید متن کووری را بسازیم. طبیعتاً بازهم از session استفاده می‌کنیم و دو حالت خواهیم داشت که هر دو حالت هم آیدی توییت یا توییت‌ها به ما رسیده و با آن آیدی یکتا می‌توان توییت را شناسایی کرد.

در صورتی که بخواهیم با هشتگ کار کنیم:

```
hash_query = "select tweet_id from t_by_hashtag WHERE hashtag=%s AND send_time <= '{ }' AND send_time >= '{ }'"
hash_query.format(end_input_time, start_input_time)
selected_elements = session.execute(hash_query, [input_hash])
```

در صورتی هم که بخواهیم با نام کاربری کار کنیم:

```
user_query = "select tweet_id from t_by_username WHERE username=%s AND send_time <= '{ }' AND send_time >= '{ }'"
user_query.format(end_input_time, start_input_time)
selected_elements = session.execute(user_query, [input_uname])
```

در صورتی که بخواهیم مثلاً تعداد توییت‌هایی که شرایط مذکور را رعایت می‌کنند بدست بیاوریم دوتا راه داریم. یا می‌توان تعداد توییت آیدی‌هایی که برگردانده شده را شمارش کرده و با طی کردن آنان مثلاً یک count ای را زیاد کنیم. راه دیگر هم این است که مستقیم از خروجی‌ای که از سشن گرفته‌ایم از توابع درونی آن استفاده کنیم تا به جای خود المان‌ها تعداد را برگرداند.

البته لازم به ذکر است که برگرداندن تعداد توییت‌ها در یک بازه دلخواه قابل انجام نمیباشد زیرا طبق طراحی جدول‌هایی که انجام داده‌ایم لازم است در کووری‌ای که می‌زنیم (یعنی چیزی که در کووری بررسی می‌کنیم) باید بخشی از partition key باشد ولی ما جداولمان را این شکلی طراحی نکرده‌ایم و با اجرای دستور warning می‌خوریم. برای انجام این کار لازم است که جدول مورد نیاز آن از اول طراحی شود تا بتوان این کار را هم انجام داد.

مشکلات و توضیحات تکمیلی

مشکل خاصی برخورد نکردم.

آنچه آموختم

کار کردن اولیه با کاساندرا و نحوه ساختن جداول