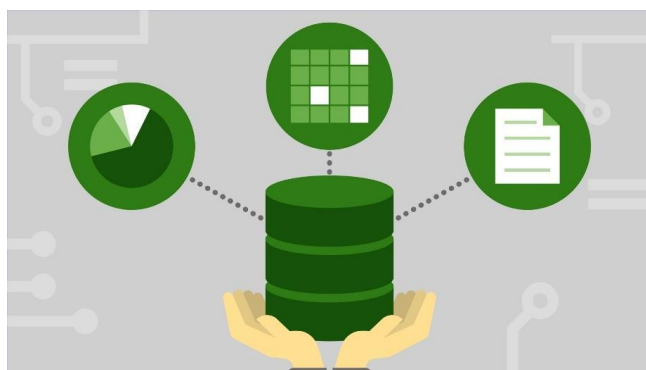


به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

دستور کار شماره نه (کار با ردیس)

امیرعلی رایگان

۸۱۰۱۹۷۶۲۳

بهار ۱۴۰۱

گزارش دستورکار انجام شده

آشنایی با ردیس

در ابتدا با خواندن و انجام مراحل گام به گام پست آموزش ردیس در نشانی ذکر شده در صورت پروژه در سایت ویرگول به آشنایی با این پایگاه داده پرداخته شد. اسکرین شات هایی از این فرآیند در ادامه آمده است:

```

amirali@DESKTOP-2P0IDPU: ~
127.0.0.1:6379> lpush newList newItem
(integer) 1
127.0.0.1:6379> rpush newList newItem2
(integer) 2
127.0.0.1:6379> lpush newList newItem3
(integer) 3
127.0.0.1:6379> lpush newList newItem4
(integer) 4
127.0.0.1:6379> lrange newList 0 3
1) "newItem4"
2) "newItem3"
3) "newItem"
4) "newItem2"
127.0.0.1:6379> lpop newList
"newItem4"
127.0.0.1:6379> lrange newList 0 3
1) "newItem3"
2) "newItem"
3) "newItem2"
127.0.0.1:6379>
  
```

شاید اگر لیستی از رشته ها را بخواهید بسازید از List استفاده کنید ولی اگر لیستی بخواهید که

```

amirali@DESKTOP-2P0IDPU: ~
127.0.0.1:6379> lpush newList newItem
(integer) 1
127.0.0.1:6379> rpush newList newItem2
(integer) 2
127.0.0.1:6379> lpush newList newItem3
(integer) 3
127.0.0.1:6379> lpush newList newItem4
(integer) 4
127.0.0.1:6379> lrange newList 0 3
1) "newItem4"
2) "newItem3"
3) "newItem"
4) "newItem2"
127.0.0.1:6379> lpop newList
"newItem4"
127.0.0.1:6379> lrange newList 0 3
1) "newItem3"
2) "newItem"
3) "newItem2"
127.0.0.1:6379>
  
```

شاید اگر لیستی از رشته ها را بخواهید بسازید از List استفاده کنید ولی اگر لیستی بخواهید که

```

127.0.0.1:6379> sadd newSet newMember
(integer) 1
127.0.0.1:6379> sadd newSet newMember2
(integer) 1
127.0.0.1:6379> sadd newSet newMember23
(integer) 1
127.0.0.1:6379> sadd newSet newMember23
(integer) 0
127.0.0.1:6379> scard newSet
(integer) 3
127.0.0.1:6379> smembers newSet
1) "newMember2"
2) "newMember"
3) "newMember23"
127.0.0.1:6379> sismember newSet newMember
(integer) 1
127.0.0.1:6379> sismember newSet newMemberssss
(integer) 0
127.0.0.1:6379> srem newSet newMember
(integer) 1
127.0.0.1:6379> smembers newSet
1) "newMember2"
2) "newMember23"
127.0.0.1:6379>

```

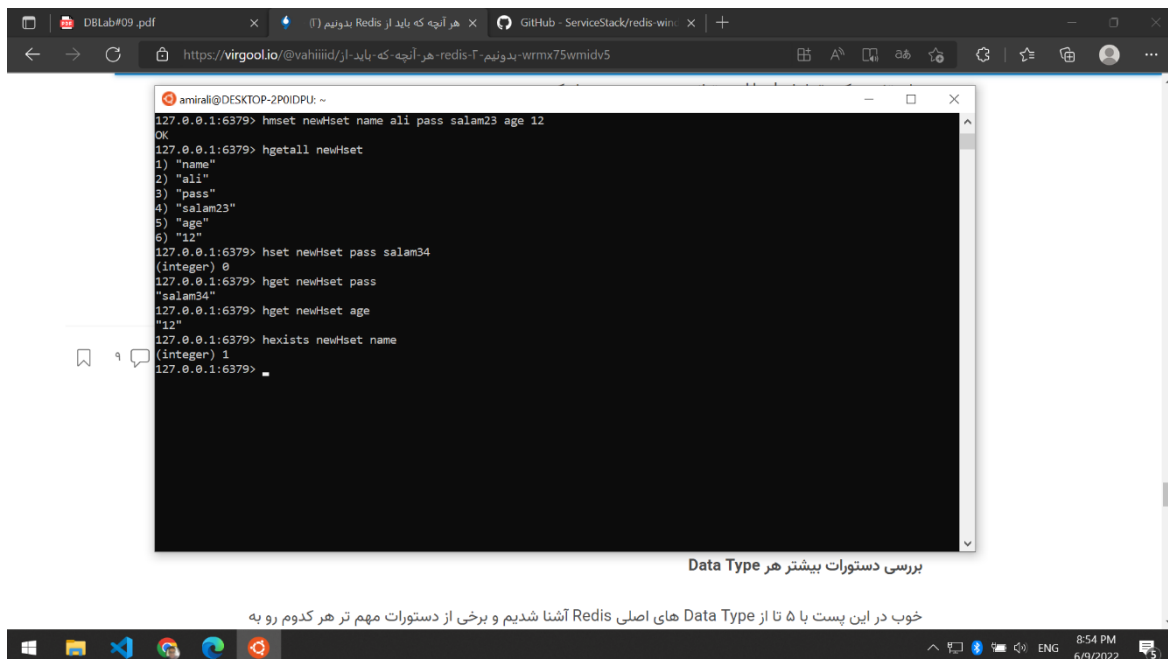
بدون تکرار، اما بر خلاف Set می توان در Sorted Set یک مقدار به عنوان Score به هر المان داد.

با این مقدار می توانیم به بیشترین Score المان های خودمون رو مرتب کنیم و این نکته را هم باید

```

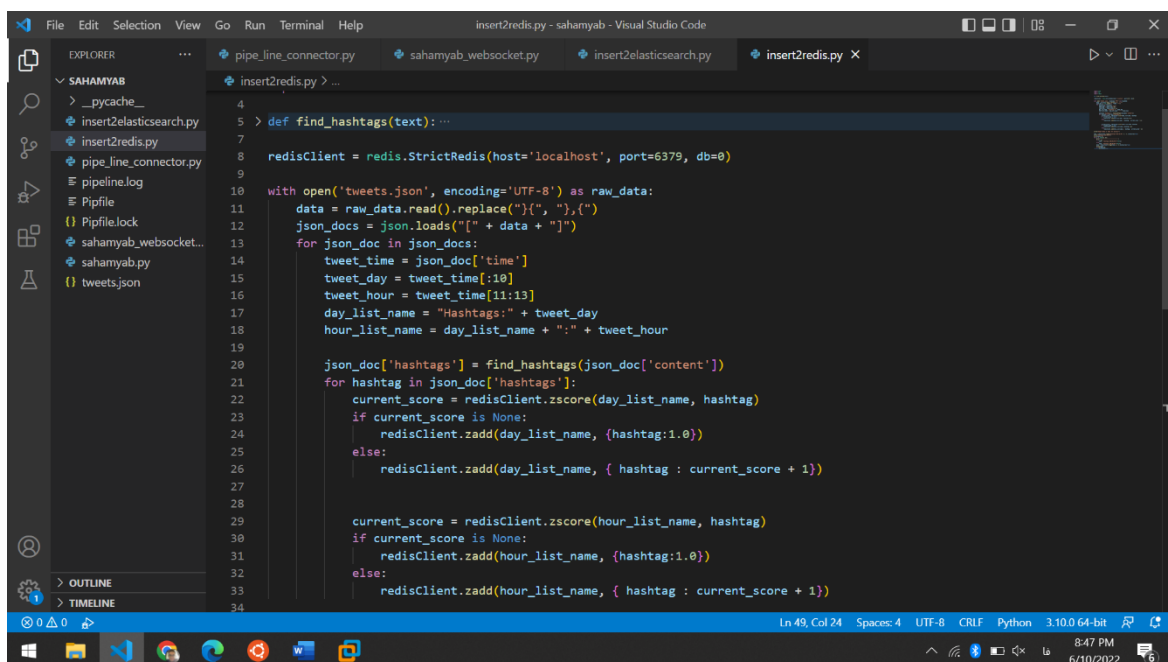
127.0.0.1:6379> zadd newZset 1 member1
(integer) 1
127.0.0.1:6379> zadd newZset 2 member2
(integer) 1
127.0.0.1:6379> zadd newZset 4 member3
(integer) 1
127.0.0.1:6379> zscore newZset member3
"4"
127.0.0.1:6379> zrange newZset 0 3
1) "member1"
2) "member2"
3) "member3"
127.0.0.1:6379> zrange newZset 0 2
1) "member1"
2) "member2"
3) "member3"
127.0.0.1:6379> zrange newZset 0 1
1) "member1"
2) "member2"
127.0.0.1:6379> zscore newZset member3
"4"
127.0.0.1:6379> zrem newZset member3
(integer) 1
127.0.0.1:6379> zrange newZset 0 3
1) "member1"
2) "member2"
127.0.0.1:6379>

```



نوشتن کد

با استفاده از کد سهام یاب که برای دستور کار الستیک سرچ در اختیارمان قرار گرفته بود، توییت هایی که در حال تولید هستند را گرفته، آنها را مرتب می کنیم، هشتگ هایشان را پیدا می کنیم و در پایگاه داده ردیس با نام متناسب ذخیره می کنیم. (البته در این دستور کار چون تعداد توییت هایی که تولید می شد خیلی خیلی کم بود از همان توییت هایی که در الستیک سرچ استفاده شد استفاده کردم. به همین دلیل تاریخ توییت ها عموماً در فروردین ماه است.) تصویری از کد ذخیره کردن داده ها در ادامه آمده است:



در ادامه عکسی از نمونه ای از استخراج داده ها در روز ۷ خرداد آمده است.

```

34
35 print("Sample data in redis for one day:")
36
37 data = redisClient.zrange("hashtags:1401/01/07", 0, -1, withscores=True)
38 print("hashtags:1401/01/07")
39 print(data[-1])
40 for i in range(1, 24):
41     print('-----')
42     if i < 10:
43         time = "hashtags:1401/01/07:0"+str(i)
44     else:
45         time = "hashtags:1401/01/07:"+str(i)
46     data = redisClient.zrange(time, 0, -1, withscores=True)
47     print(time)
48     if len(data)>0:
49         print(data[-1])
50

```

```

Sample data in redis for one day:
hashtags:1401/01/07
(b'\xd8\xae\xd9\x88\xd8\xaf\xd8\xb1\xd9\x88', 168.0)
-----
hashtags:1401/01/07:01
-----
hashtags:1401/01/07:02
(b'\xd8\xa8\xda\xa9\xd8\xa7\xd8\xa8', 7.0)
-----
hashtags:1401/01/07:03
-----
hashtags:1401/01/07:04
-----
hashtags:1401/01/07:05
-----
hashtags:1401/01/07:06
-----
hashtags:1401/01/07:07
-----
hashtags:1401/01/07:08
-----
hashtags:1401/01/07:09
-----
hashtags:1401/01/07:10
-----
hashtags:1401/01/07:11
(b'\xd9\x86\xd9\x88\xd8\xb1\xdb\x8c', 14.0)
-----
hashtags:1401/01/07:12
(b'\xd8\xae\xd9\x88\xd8\xaf\xd8\xb1\xd9\x88', 21.0)
-----
hashtags:1401/01/07:13
-----

```

سوالات:

• سوال اول:

برای این کار می توانیم از دستوری شبیه دستور زیر در زمان ذخیره کردن داده استفاده کنیم.

```
r.set('name:' + str(i), i, ex=time_to_expire_s)
```

• سوال دوم:

می‌توانیم از دستوری به شکل زیر استفاده کنیم. در این دستور عدد اول نشان دهنده ی تعداد ثانیه های بین ذخیره کردن داده در دیسک و عدد دوم حداقل تغییرات برای ذخیره سازی است. به عنوان مثال با این دستور هر ۶۰ ثانیه اگر ۱۰۰۰ تغییر در داده ها رخ داده بود داده ها به دیسک انتقال داده می شوند.

```
save 60 1000
```

به این کار به اصطلاح snapshoting نیز می گویند. برای خواسته سوال می توانیم چنین دستوری را وارد کنیم.

```
save 600 100
```

- سوال سوم:

Redis برای نگه داری دائمی داده ها آنها را با توجه به تنظیماتی که ما برای آن مشخص می کنیم به دیسک اصلی سیستم منتقل می کند و بعد از پاک شدن RAM دوباره می تواند آنها را منتقل کند و کار را از سر بگیرد.

این ویژگی باعث شده اصطلاحاً به آن **on-disk persistence** بگویند و این کار را می تواند در سطوح مختلفی انجام دهد.

مشکلات و توضیحات تکمیلی

مشکلی به وجود نیامد.

آنچه آموختم / پیشنهادات

آشنایی خوبی با ردیس بی پیدا کردم.