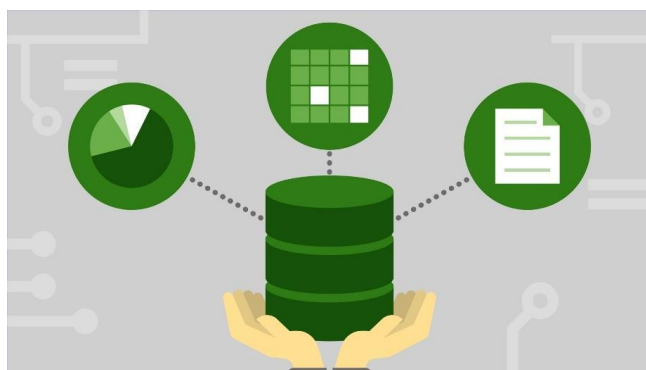


به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

دستور کار شماره دوم (کار با الستیک سرچ)

امیرعلی رایگان

۸۱۰۱۹۷۶۲۳

فروردین ۱۴۰۱

گزارش دستورکار انجام شده

گام اول. آشنایی با الاستیک سرچ

مطابق نمونه ی نشان داده شده در وبسایت موارد زیر انجام داده شد:

The screenshot shows the Elastic Dev Tools Console with a PUT request to `movies/_doc/1`. The request body is a JSON object representing a movie document. The response shows the document was successfully created with status 201.

```

1 PUT /movies/_doc/1
2 {
3   "title": "The Godfather",
4   "director": "Francis Ford Coppola",
5   "year": 1972
6 }
7
8 {
9   "_index": "movies",
10  "_id": "1",
11  "_version": 1,
12  "result": "created",
13  "shards": {
14    "total": 2,
15    "successful": 1,
16    "failed": 0
17  },
18  "_seq_no": 0,
19  "_primary_term": 1
20 }
  
```

عکس اول

The screenshot shows a series of PUT requests to create multiple movie documents in the 'movies' index. The response for the last request (document 6) is shown.

```

18 PUT /movies/_doc/4
19 {
20   "title": "Apocalypse Now",
21   "director": "Francis Ford Coppola",
22   "year": 1979,
23   "genres": ["Drama", "War"]
24 }
25
26 PUT /movies/_doc/5
27 {
28   "title": "Kill Bill: Vol. 1",
29   "director": "Quentin Tarantino",
30   "year": 2003,
31   "genres": ["Action", "Crime", "Thriller"]
32 }
33
34 PUT /movies/_doc/6
35 {
36   "title": "The Assassination of Jesse James by the Coward Robert Ford",
37   "director": "Andrew Dominik",
38   "year": 2007,
39   "genres": ["Biography", "Crime", "Drama"]
40 }
41
42 {
43   "_index": "movies",
44   "_id": "6",
45   "_version": 1,
46   "result": "created",
47   "shards": {
48     "total": 2,
49     "successful": 1,
50     "failed": 0
51   },
52   "_seq_no": 5,
53   "_primary_term": 1
54 }
  
```

عکس دوم

The screenshot shows a GET request to `movies/_doc/4`. The response returns the document for 'Apocalypse Now' with status 200.

```

1 GET /movies/_doc/4
2
3 {
4   "_index": "movies",
5   "_id": "4",
6   "_version": 1,
7   "_primary_term": 1,
8   "found": true,
9   "_source": {
10    "title": "Apocalypse Now",
11    "director": "Francis Ford Coppola",
12    "year": 1979,
13    "genres": [
14      "Drama",
15      "War"
16    ]
17  }
18 }
  
```

عکس سوم

```

1 POST movies/_search
2 {
3   "query": {
4     "bool": {
5       "must_not": [
6         {
7           "match": {
8             "genres": "drama"
9           }
10        }
11      ]
12    }
13  }
14 }

```

```

17 {
18   "_index": "movies",
19   "_id": "1",
20   "_score": 0.0,
21   "_source": {
22     "title": "The Godfather",
23     "director": "Francis Ford Coppola",
24     "year": 1972
25   }
26 },
27 {
28   "_index": "movies",
29   "_id": "5",
30   "_score": 0.0,
31   "_source": {
32     "title": "Kill Bill: Vol. 1",
33     "director": "Quentin Tarantino",
34     "year": 2003,
35     "genres": [
36       "Action",
37       "Crime",
38       "Thriller"
39     ]
40   }
41 },
42 }

```

عکس چهارم

```

1 POST movies/_search
2 {
3   "query": {
4     "match": {
5       "title": "godfather",
6       "fuzziness": "AUTO"
7     }
8   }
9 }

```

```

5 {
6   "total": 1,
7   "successful": 1,
8   "skipped": 0,
9   "failed": 0
10 },
11 {
12   "hits": {
13     "total": 1,
14     "value": 1,
15     "relation": "eq"
16   },
17   "max_score": 1.7120997,
18   "hits": [
19     {
20       "_index": "movies",
21       "_id": "1",
22       "_score": 1.7120997,
23       "_source": {
24         "title": "The Godfather",
25         "director": "Francis Ford Coppola",
26         "year": 1972
27       }
28     }
29   ]
30 }

```

عکس پنجم

```

1 POST movies/_search
2 {
3   "query": {
4     "term": {
5       "title.keyword": {
6         "value": "Apocalypse Now"
7       }
8     }
9   }
10 }

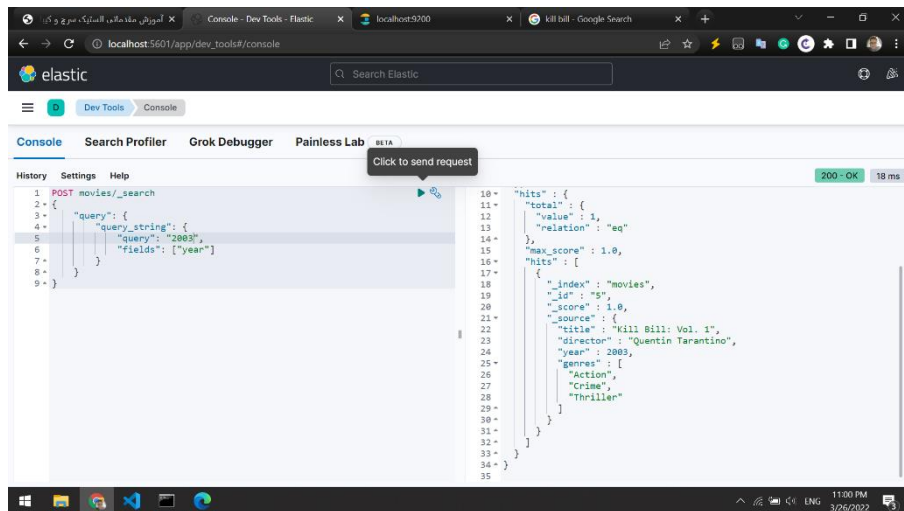
```

```

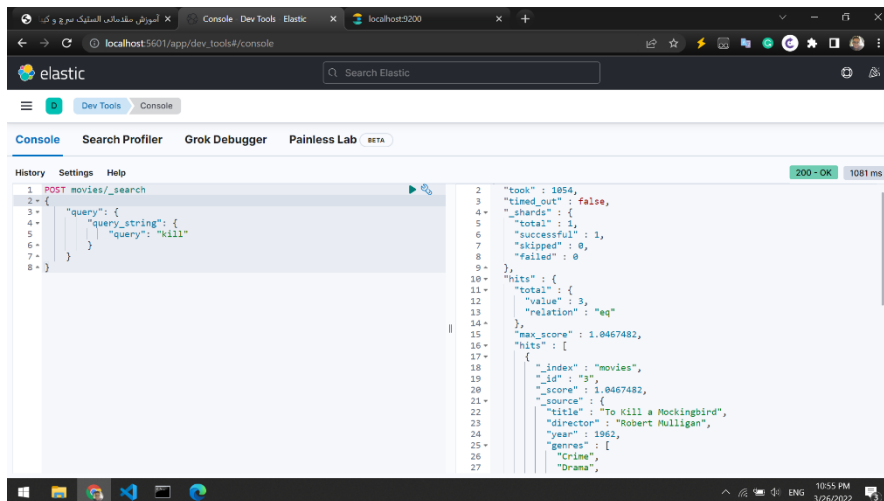
6 {
7   "successful": 1,
8   "skipped": 0,
9   "failed": 0
10 },
11 {
12   "hits": {
13     "total": 1,
14     "value": 1,
15     "relation": "eq"
16   },
17   "max_score": 1.548445,
18   "hits": [
19     {
20       "_index": "movies",
21       "_id": "4",
22       "_score": 1.548445,
23       "_source": {
24         "title": "Apocalypse Now",
25         "director": "Francis Ford Coppola",
26         "year": 1979,
27         "genres": [
28           "Drama",
29           "War"
30         ]
31       }
32     }
33 ]
34 }

```

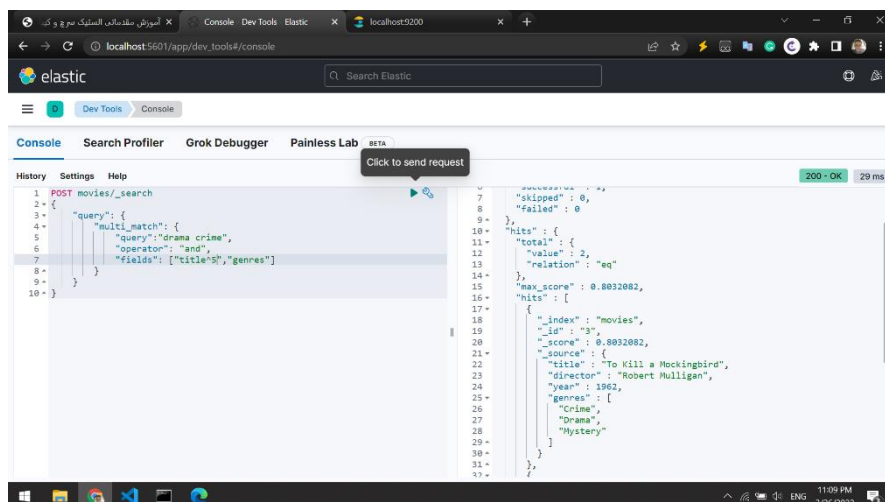
عکس ششم



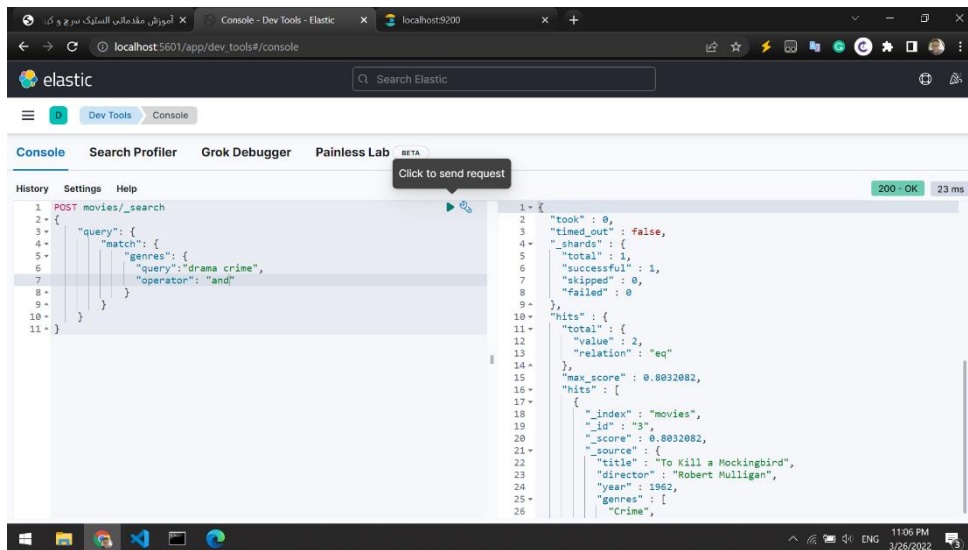
عکس هفتم



عکس هشتم



عکس نهم



عکس دهم

گام دوم . گرفتن داده ها، انتقال آنها با پایگاه داده و تصویرسازی آنها

در این گام با کمک برنامه های پایتون ابتدا داده ها با استفاده از کد های مربوطه دانلود شد. سپس با کمی تغییر در فایل pipe_line_connector.py داده ها در یک فایل json قرار داده شد و در انتها با تغییر فایل insert2elasticsearch.py آن داده ها وارد دیتابیس روی localhost ساخته بودیم شد. در پایان هم با استفاده از ابزار های الاستیک سرچ چند نوع نمودار از نحوه توزیع داده ها کشیده شد.

```

pipe_line_connector.py > on_item_update
4 import re
5 from turtle import st
6 sys.stdout.reconfigure(encoding='utf-8')
7
8 from pprint import pprint
9 def on_item_update(item_update):
10     record = item_update
11     if record["values"].get("content") != None:
12         record["values"]["content"] = record["values"]["content"].encode().decode('unicode_escape')
13         record["values"]["senderName"] = record["values"]["senderName"].encode().decode('unicode_escape')
14         if record["values"].get("parentContent") != None:
15             record["values"]["parentContent"] = record["values"]["parentContent"].encode().decode('unicode_escape')
16             record["values"]["parentSenderName"] = record["values"]["parentSenderName"].encode().decode('unicode_escape')
17
18     storing_data = {}
19     storing_data['content'] = record["values"]["content"]
20     storing_data['time'] = record["values"]["sendTimePersian"]
21     storing_data['type'] = record["values"]["type"]
22     storing_data['username'] = record["values"]["senderUsername"]
23     storing_data['hashtags'] = re.findall(r"#(\w+)", record["values"]["content"])
24
25     with open("tweets.json", "a") as outfile:
26         json.dump(storing_data, outfile)
27         print("done!")
28     #print(pprint(record))
29

```

*عکس هایی از کد های تغییر داده شده

```

insert2elasticsearch.py > ...
1 import json
2 import re
3 from elasticsearch import Elasticsearch
4
5 import warnings
6 warnings.filterwarnings("ignore")
7
8 def find_hashtags(text):
9     return re.findall(r"#(\w+)", text)
10
11 es = Elasticsearch("http://localhost:9200")
12 i = 0
13 with open('tweets.json', encoding='UTF-8') as raw_data:
14     data = raw_data.read().replace("'", '"')
15     json_docs = json.loads('[' + data + ']')
16     print(type(json_docs))
17     for json_doc in json_docs:
18         #del json_doc['id']
19         #json_doc['hashtags'] = find_hashtags(json_doc['content'])
20         i = i + 1
21         es.index(index='twitter-tweets', id=i, body=json.dumps(json_doc))
22     print(i, '-->', json_docs['hashtags'])
23

```



مشکلات و توضیحات تکمیلی

بیشترین زمانی که برای این گزارش کار صرف شد مربوط به نصب و اجرای الاستیک سرچ و کیبانا بود. در بعضی از مواقع آموزش ها مطابق با اتفاقاتی که می افتاد نبود اما در نهایت با پرسیدن از دوستان و سوال های داخل گروه مشکلات این قسمت رفع شد.

آنچه آموختم / پیشنهادات

قبل از این با این نوع پایگاه داده آشنایی نداشتم. دست گرمی بسیار مفید و کاربری ای بود و اگر در آینده بخواهم از آن استفاده کنم با مقدمات استفاده از الاستیک سرچ آشنا هستم.