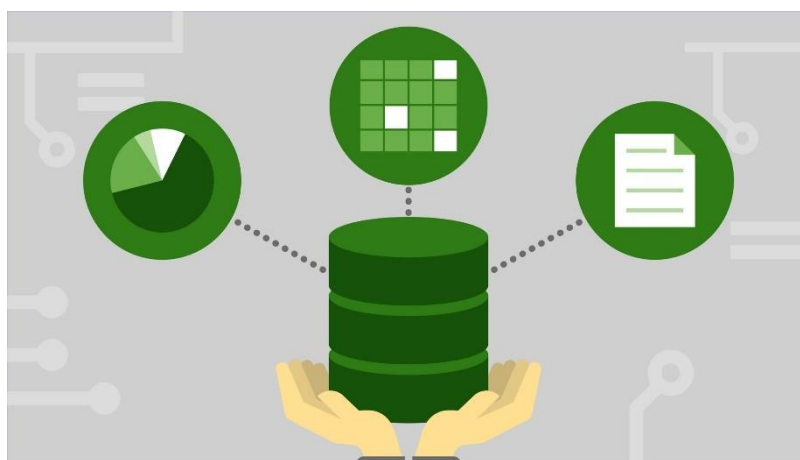


به نام خدا



دانشگاه تهران  
پردیس دانشکده‌های فنی  
دانشکده برق و کامپیوتر



## آزمایشگاه پایگاه داده

دستور کار شماره ۷

سجاد علی‌زاده

۸۱۰۱۹۷۵۴۷

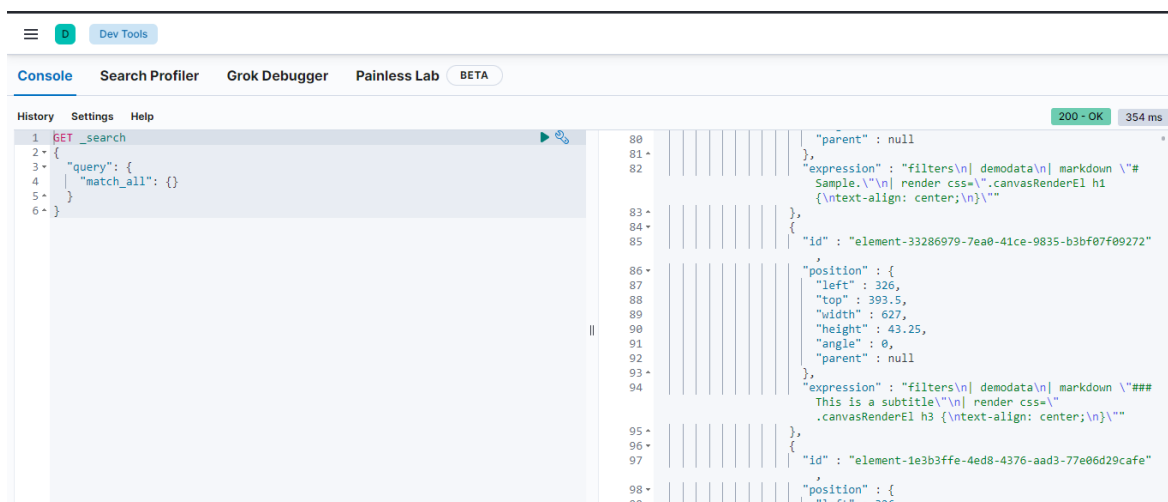
آذرماه ۱۴۰۰

## گزارش دستورکار انجام شده

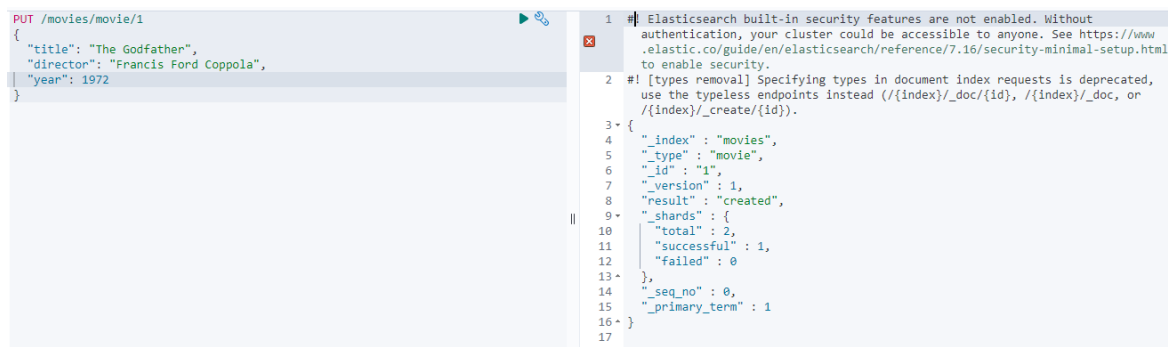
## گام اول:

ابتدا kibana و elastic search را نصب می‌کنیم:

```
{
  "name" : "DESKTOP-C49049U",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "8-hWtU9JT6HVnSpKHx_WQ",
  "version" : {
    "number" : "7.16.2",
    "build_flavor" : "default",
    "build_type" : "zip",
    "build_hash" : "2b937c44140b6559905130a8650c64dbd0879cfb",
    "build_date" : "2021-12-18T19:42:46.604893745Z",
    "build_snapshot" : false,
    "lucene_version" : "8.10.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```



اضافه کردن داده: یک داده به شکل زیر اضافه می‌کنیم:



همانطور که مشاهده می‌شود به ما گفته میشود type منسوخ شده است (یک وارنینگ سکيوریتی هم میدهد که میگوید باید تنظیمات سکيوریتی را فعال کنیم. اهمیتی نمیدهیم) این زین پس از \_doc استفاده میکنیم. همچنین یک سری اطلاعات اضافه برگردانده می‌شود. حال اگر یک بار دیگر این دستور را اجرا کنیم نتیجه زیر حاصل می‌شود:

| History   | Settings  | Help           |
|---|---|----------------|
| <pre> 1 PUT /movies/movie/1 2 { 3   "title": "The Godfather", 4   "director": "Francis Ford Coppola", 5   "year": 1972 6 } </pre> | <pre> 1 #! Elasticsearch built-in security features are not enabled. Without 2 authentication, your cluster could be accessible to anyone. See https://www 3 .elastic.co/guide/en/elasticsearch/reference/7.16/security-minimal-setup.html 4 to enable security. 5 6 #! [types removal] Specifying types in document index requests is deprecated, 7 use the typeless endpoints instead ({index}/{_doc/{id}}, {index}/{_doc, or 8 {index}/{_create/{id}}). 9 10 { 11   "_index": "movies", 12   "_type": "movie", 13   "_id": "1", 14   "_version": 2, 15   "result": "updated", 16   "_shards": { 17     "total": 2, 18     "successful": 1, 19     "failed": 0 20   }, 21   "_seq_no": 1, 22   "_primary_term": 1 23 } </pre> | 200 - OK 38 ms |

همانطور که مشاهده می‌شود مقدار ورژن به ۲ تغییر یافت و مقدار result نیز updated شد. بهتر است برای ایجاد داده از PUT و برای بروزرسانی داده از POST استفاده کنیم. حال چند نمونه اضافه میکنیم تا در ادامه با آنها کار کنیم:

|   |   |  |
|---|---|--|
| <pre> PUT movies/_doc/2 {   "title": "Lawrence of Arabia",   "director": "David Lean",   "year": 1962,   "genres": ["Adventure", "Biography", "Drama"] } </pre>   | <pre> 1 #! Elasticsearch built-in security features are not enabled. 2 Without authentication, your cluster could be accessible to 3 anyone. See https://www.elastic.co/guide/en/elasticsearch 4 /reference/7.16/security-minimal-setup.html to enable 5 security. 6 7 { 8   "_index": "movies", 9   "_type": "_doc", 10  "_id": "2", 11  "_version": 1, 12  "result": "created", 13  "_shards": { 14    "total": 2, 15    "successful": 1, 16    "failed": 0 17  }, 18  "_seq_no": 2, 19  "_primary_term": 1 20 } </pre> |  |
| <pre> PUT movies/_doc/3 {   "title": "To Kill a Mockingbird",   "director": "Robert Mulligan",   "year": 1962,   "genres": ["Crime", "Drama", "Mystery"] } </pre> | <pre> 1 #! Elasticsearch built-in security features are not enabled. 2 Without authentication, your cluster could be accessible to 3 anyone. See https://www.elastic.co/guide/en/elasticsearch 4 /reference/7.16/security-minimal-setup.html to enable 5 security. 6 7 { 8   "_index": "movies", 9   "_type": "_doc", 10  "_id": "3", 11  "_version": 1, 12  "result": "created", 13  "_shards": { 14    "total": 2, 15    "successful": 1, 16    "failed": 0 17  }, 18  "_seq_no": 3, 19  "_primary_term": 1 20 } </pre> |  |

|   |  |
|---|--|
| <pre> 1 PUT movies/_doc/4 {   "title": "Apocalypse Now",   "director": "Francis Ford Coppola",   "year": 1979,   "genres": ["Drama", "War"] } </pre>  | <pre> 1 #! Elasticsearch built-in security features are not enabled.   Without authentication, your cluster could be accessible to   anyone. See https://www.elastic.co/guide/en/elasticsearch   /reference/7.16/security-minimal-setup.html to enable   security. 2 { 3   "_index" : "movies", 4   "_type" : "_doc", 5   "_id" : "4", 6   "_version" : 1, 7   "result" : "created", 8   "_shards" : { 9     "total" : 2, 10    "successful" : 1, 11    "failed" : 0 12  }, 13  "_seq_no" : 4, 14  "_primary_term" : 1 15 } </pre> |
| <pre> 1 PUT movies/_doc/5 2 { 3   "title": "Kill Bill: Vol. 1", 4   "director": "Quentin Tarantino", 5   "year": 2003, 6   "genres": ["Action", "Crime", "Thriller"] 7 } </pre>                             | <pre> 1 #! Elasticsearch built-in security features are not enabled.   Without authentication, your cluster could be accessible to   anyone. See https://www.elastic.co/guide/en/elasticsearch   /reference/7.16/security-minimal-setup.html to enable   security. 2 { 3   "_index" : "movies", 4   "_type" : "_doc", 5   "_id" : "5", 6   "_version" : 1, 7   "result" : "created", 8   "_shards" : { 9     "total" : 2, 10    "successful" : 1, 11    "failed" : 0 12  }, 13  "_seq_no" : 5, 14  "_primary_term" : 1 15 } </pre> |
| <pre> 1 PUT movies/_doc/6 {   "title": "The Assassination of Jesse James by the Coward   Robert Ford",   "director": "Andrew Dominik",   "year": 2007,   "genres": ["Biography", "Crime", "Drama"] } </pre> | <pre> 1 #! Elasticsearch built-in security features are not enabled.   Without authentication, your cluster could be accessible to   anyone. See https://www.elastic.co/guide/en/elasticsearch   /reference/7.16/security-minimal-setup.html to enable   security. 2 { 3   "_index" : "movies", 4   "_type" : "_doc", 5   "_id" : "6", 6   "_version" : 1, 7   "result" : "created", 8   "_shards" : { 9     "total" : 2, 10    "successful" : 1, 11    "failed" : 0 12  }, 13  "_seq_no" : 6, 14  "_primary_term" : 1 15 } </pre> |

با دستور زیر می‌توانیم یک داده را با استفاده از آیدی جستجو و دریافت کنیم:

|                                   |   |
|-----------------------------------|---|
| <pre> 1 GET /movies/_doc/1 </pre> | <pre> 1 #! Elasticsearch built-in security features are not enabled.   Without authentication, your cluster could be accessible to   anyone. See https://www.elastic.co/guide/en/elasticsearch   /reference/7.16/security-minimal-setup.html to enable   security. 2 { 3   "_index" : "movies", 4   "_type" : "_doc", 5   "_id" : "1", 6   "_version" : 2, 7   "_seq_no" : 1, 8   "_primary_term" : 1, 9   "found" : true, 10  "source" : { 11    "title" : "The Godfather", 12    "director" : "Francis Ford Coppola", 13    "year" : 1972 14  } 15 } </pre> |
|-----------------------------------|---|

در اینجا سند با آیدی ۱ را دریافت کردیم.

حال با استفاده از کوئری استرینگ یک جستجو انجام می‌دهیم:

```
POST movies/_doc/_search
{
  "query": {
    "query_string": {
      "query": "kill"
    }
  }
}
```

```
2 #! [types removal] Specifying types in search requests is deprecated.
3 {
4   "took" : 29,
5   "timed_out" : false,
6   "_shards" : {
7     "total" : 1,
8     "successful" : 1,
9     "skipped" : 0,
10    "failed" : 0
11  },
12  "hits" : {
13    "total" : {
14      "value" : 0,
15      "relation" : "eq"
16    },
17    "max_score" : null,
18    "hits" : [ ]
19  }
20 }
```

از آنجا که جستجو با تایپ منسوخ شده هیچ جوابی دریافت نمیشود. حال با حذف تایپ به جواب مورد نظر میرسیم:

```
POST movies/_search
{
  "query": {
    "query_string": {
      "query": "kill"
    }
  }
}
```

```
8   "skipped" : 0,
9   "failed" : 0
10 },
11 "hits" : {
12   "total" : {
13     "value" : 2,
14     "relation" : "eq"
15   },
16   "max_score" : 1.0467482,
17   "hits" : [
18     {
19       "_index" : "movies",
20       "_type" : "movie",
21       "_id" : "3",
22       "_score" : 1.0467482,
23       "_source" : {
24         "title" : "To Kill a Mockingbird",
25         "director" : "Robert Mulligan",
26         "year" : 1962,
27         "genres" : [
28           "Crime",
```

هر سندی که در یک فیلد آن کلمه kill آمده باشد بازگردانده شده است.

حال جستجو در یک فیلد را انجام می‌دهیم:

```
GET /movies/_search
{
  "query": {
    "query_string": {
      "query": "ford",
      "fields": ["title"]
    }
  }
}
```

```
15 },
16 "max_score" : 0.9794736,
17 "hits" : [
18   {
19     "_index" : "movies",
20     "_type" : "movie",
21     "_id" : "6",
22     "_score" : 0.9794736,
23     "_source" : {
24       "title" : "The Assassination of Jesse James by
the Coward Robert Ford",
25       "director" : "Andrew Dominik",
26       "year" : 2007,
27       "genres" : [
28         "Biography",
29         "Crime",
30         "Drama"
31       ]
32     }
33   }
34 ]
```

اسنادی که در فیلد title آن‌ها کلمه ford آمده باشد در خروجی مشاهده می‌شود.

نکته بی‌ربط: فیلم The assassination of Jesse James by the Coward Robert Ford خیلی جالبه. کل داستان فیلم رو

تو اسم فیلم اسپویل کرده. چرا آخه؟ ☹️

حال چند کوئری پیچیده‌تر مینویسیم:

تمام فیلمهایی که در نام آنها kill یا ford به کار رفته است:

```
GET /movies/_search
{
  "query": {
    "query_string": { "query": "title:ford OR kill" }
  }
}
```

```
18 {
19   "_index": "movies",
20   "_type": "movie",
21   "_id": "3",
22   "_score": 1.0467482,
23   "_source": {
24     "title": "To Kill a Mockingbird",
25     "director": "Robert Mulligan",
26     "year": 1962,
27     "genres": [
28       "Crime",
29       "Drama",
30       "Mystery"
31     ]
32   }
33 },
34 {
35   "_index": "movies",
36   "_type": "movie",
37   "_id": "5",
38   "_score": 1.0467482,
```

تمام فیلم هایی که عنوان kill داشته و کارگردان آنها Tarantino است:

```
GET /movies/_search
{
  "query": {
    "query_string": { "query": "(title:kill) AND (director:Tarantino)" }
  }
}
```

```
14 relation : eq
15 },
16 "max_score" : 2.682807,
17 "hits" : [
18 {
19   "_index": "movies",
20   "_type": "movie",
21   "_id": "5",
22   "_score": 2.682807,
23   "_source": {
24     "title": "Kill Bill: Vol. 1",
25     "director": "Quentin Tarantino",
26     "year": 2003,
27     "genres": [
28       "Action",
29       "Crime",
30       "Thriller"
31     ]
32   }
33 }
34 ]
```

تمام فیلم های ژانر درام در سال ۱۹۶۲:

```
GET /movies/_search
{
  "query": {
    "query_string": { "query": "(genres:Drama) AND (year:1962)" }
  }
}
```

```
22 "_score": 1.2795144,
23 "_source": {
24   "title": "Lawrence of Arabia",
25   "director": "David Lean",
26   "year": 1962,
27   "genres": [
28     "Adventure",
29     "Biography",
30     "Drama"
31   ]
32 },
33 },
34 {
35   "_index": "movies",
36   "_type": "movie",
37   "_id": "3",
38   "_score": 1.2795144,
39   "_source": {
40     "title": "To Kill a Mockingbird",
41     "director": "Robert Mulligan",
42     "year": 1962,
```

تمام فیلم های ژانر درام تا سال ۲۰۰۵:

```
GET /movies/_search
{
  "query": {
    "query_string": { "query": "(genres:Drama) AND (year:[
      * TO 2005] )" }
  }
}
```

```

9      "failed" : 0
10    },
11    "hits" : {
12      "total" : {
13        "value" : 3,
14        "relation" : "eq"
15      },
16      "max_score" : 1.3257576,
17      "hits" : [
18        {
19          "_index" : "movies",
20          "_type" : "movie",
21          "_id" : "4",
22          "_score" : 1.3257576,
23          "_source" : {
24            "title" : "Apocalypse Now",
25            "director" : "Francis Ford Coppola",
26            "year" : 1979,
27            "genres" : [
28              "Drama",
29              "War"

```

تمام فیلم های درام یا فیلم هایی که در نام آنها کلمه kill به کار رفته باشد:

```
GET /movies/_search
{
  "query": {
    "query_string": { "query": "(genres:Drama) OR (title
      :kill)" }
  }
}
```

```

6      "total" : 1,
7      "successful" : 1,
8      "skipped" : 0,
9      "failed" : 0
10    },
11    "hits" : {
12      "total" : {
13        "value" : 5,
14        "relation" : "eq"
15      },
16      "max_score" : 1.3262626,
17      "hits" : [
18        {
19          "_index" : "movies",
20          "_type" : "movie",
21          "_id" : "3",
22          "_score" : 1.3262626,
23          "_source" : {
24            "title" : "To Kill a Mockingbird",
25            "director" : "Robert Mulligan",
26            "year" : 1962,

```

با دستور زیر اسنادی که در نام آنها کلمه kill استفاده شده باشد اولویت بالاتری میگیرند:

```
GET /movies/_search
{
  "query": {
    "query_string": { "query": "(genres:Drama) OR (title
      :kill^2)" }
  }
}
```

```

37      "total" : 2,
38      "_score" : 2.0934963,
39      "_source" : {
40        "title" : "Kill Bill: Vol. 1",
41        "director" : "Quentin Tarantino",
42        "year" : 2003,
43        "genres" : [
44          "Action",
45          "Crime",
46          "Thriller"
47        ]
48      },
49    },
50    {
51      "_index" : "movies",
52      "_type" : "movie",
53      "_id" : "4",
54      "_score" : 0.32575765,
55      "_source" : {
56        "title" : "Apocalypse Now",
57        "director" : "Francis Ford Coppola",
58        "year" : 1979,

```

همانطور که مشاهده می شود ابتدا اسنادی آمده است که در عنوان آنها کلمه kill است و سپس فیلم های ژانر درام.

با دستور زیر یک درخت جستجو میسازیم که در آن فیلم‌هایی که ژانر آنها درام است بازگردانده میشود:

```
POST /movies/_search
{
  "query": {
    "match": {
      "genres": {
        "query": "drama"
      }
    }
  }
}
```

```

10- },
11- "hits": {
12-   "total": {
13-     "value": 4,
14-     "relation": "eq"
15-   },
16-   "max_score": 0.32575765,
17-   "hits": [
18-     {
19-       "_index": "movies",
20-       "_type": "movie",
21-       "_id": "4",
22-       "_score": 0.32575765,
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- }
35- }
```

میتوان از اپراتورهای مختلف استفاده کرد. در کوئری زیر فیلم‌هایی بازگردانده میشود که kill یا bill در نام آنها باشد.

```
POST /movies/_search
{
  "query": {
    "match": {
      "title": {
        "query": "Kill Bill",
        "operator": "or"
      }
    }
  }
}
```

```

23-   "_source": {
24-     "title": "Kill Bill: Vol. 1",
25-     "director": "Quentin Tarantino",
26-     "year": 2003,
27-     "genres": [
28-       "Action",
29-       "Crime",
30-       "Thriller"
31-     ]
32-   }
33- },
34- {
35-   "_index": "movies",
36-   "_type": "movie",
37-   "_id": "3",
38-   "_score": 1.0467482,
39-   "_source": {
40-     "title": "To Kill a Mockingbird",
41-     "director": "Robert Mulligan",
42-     "year": 1962,
43-     "genres": [
44-       "Drama",
45-       "Crime"
46-     ]
47-   }
48- }
49- ]
50- }
```

استفاده از `match_multi` و جستجو در چند فیلد در این قسمت انجام شده است، با استفاده از توان اولویت بالاتری به `title` داده‌ایم:

```
POST /movies/_search
{
  "query": {
    "multi_match": {
      "query": "ford",
      "fields": [
        "title^3",
        "director"
      ]
    }
  }
}
```

```

11- "hits": {
12-   "total": {
13-     "value": 3,
14-     "relation": "eq"
15-   },
16-   "max_score": 2.9384212,
17-   "hits": [
18-     {
19-       "_index": "movies",
20-       "_type": "movie",
21-       "_id": "6",
22-       "_score": 2.9384212,
23-       "_source": {
24-         "title": "The Assassination of Jesse James by
25-           the Coward Robert Ford",
26-         "director": "Andrew Dominik",
27-         "year": 2007,
28-         "genres": [
29-           "Biography",
30-           "Crime",
31-           "Drama"
32-         ]
33-       }
34-     }
35-   ]
36- }
37- }
```



می‌توان چند match داشت. And آنها در پاسخ می‌آید:

```
POST /movies/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "genres": "drama"
          }
        },
        {
          "match": {
            "title": {
              "query": "kill",
              "boost": 3
            }
          }
        }
      ]
    }
  }
}
```

```
13   "value" : 1,
14   "relation" : "eq"
15 },
16   "max_score" : 3.4197588,
17   "hits" : [
18     {
19       "_index" : "movies",
20       "_type" : "movie",
21       "_id" : "3",
22       "_score" : 3.4197588,
23       "_source" : {
24         "title" : "To Kill a Mockingbird",
25         "director" : "Robert Mulligan",
26         "year" : 1962,
27         "genres" : [
28           "Crime",
29           "Drama",
30           "Mystery"
31         ]
32       }
33     }
34   ]
35 }
```

میتوان از کلیدواژه‌های دیگر نیز استفاده کرد. کوئری زیر فیلم‌هایی را برمیگرداند که در عنوان آن‌ها کلمه ford یا kill به کار رفته است و ژانر آن‌ها اسرارآمیز نیست:

```
{
  "query": {
    "bool": {
      "must": {
        "bool": {
          "should": [
            {
              "match": {
                "title": "ford"
              }
            },
            {
              "match": {
                "title": "kill"
              }
            }
          ]
        }
      },
      "must_not": {
        "match": {
          "genres": "Mystery"
        }
      }
    }
  }
}
```

```
7   "successful" : 1,
8   "skipped" : 0,
9   "failed" : 0
10 },
11   "hits" : {
12     "total" : {
13       "value" : 2,
14       "relation" : "eq"
15     },
16     "max_score" : 1.0467482,
17     "hits" : [
18       {
19         "_index" : "movies",
20         "_type" : "movie",
21         "_id" : "5",
22         "_score" : 1.0467482,
23         "_source" : {
24           "title" : "Kill Bill: Vol. 1",
25           "director" : "Quentin Tarantino",
26           "year" : 2003,
27           "genres" : [
28             "Action",
29             "Crime",
30             "Thriller"
31           ]
32         }
33       }
34     ]
35   }
36 }
```

برای جستجو در فیلدهای غیر متنی از کلیدواژه term استفاده میکنیم. در این کوئری فیلم‌هایی که سال ساخت آنها ۲۰۰۳ است بازگردانده می‌شود:

```
GET /movies/_search
{
  "query": {
    "term": {
      "year": {
        "value": "2003",
        "boost": 1.0
      }
    }
  }
}
```

```

7    "successful" : 1,
8    "skipped" : 0,
9    "failed" : 0
10  },
11  "hits" : {
12    "total" : {
13      "value" : 1,
14      "relation" : "eq"
15    },
16    "max_score" : 1.0,
17    "hits" : [
18      {
19        "_index" : "movies",
20        "_type" : "movie",
21        "_id" : "5",
22        "_score" : 1.0,
23        "_source" : {
24          "title" : "Kill Bill: Vol. 1",
25          "director" : "Quentin Tarantino",
26          "year" : 2003,
27          "genres" : [
28            "Action",
29            "Crime",
30            "Thriller"

```

برای افزایش سرعت میتوان از فیلترینگ استفاده کرد. مثلاً دستور زیر ابتدا تمام فیلم‌های تولید شده بعد از سال ۱۹۶۰ و از ژانر درام را فیلتر کرده، سپس به دنبال فیلم‌هایی که عنوان آنها شامل kill باشد، می‌گردد.

```
GET /movies/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "title": "kill"
          }
        }
      ],
      "filter": {
        "bool": {
          "must": [
            {
              "range": {
                "year": {
                  "gte": 1960
                }
              }
            },
            {
              "term": {
                "genres": {
                  "value": "drama"
                }
              }
            }
          ]
        }
      ]
    }
  }
}
```

```

5    "_shards" : {
6      "total" : 1,
7      "successful" : 1,
8      "skipped" : 0,
9      "failed" : 0
10  },
11  "hits" : {
12    "total" : {
13      "value" : 1,
14      "relation" : "eq"
15    },
16    "max_score" : 1.0,
17    "hits" : [
18      {
19        "_index" : "movies",
20        "_type" : "movie",
21        "_id" : "5",
22        "_score" : 1.0,
23        "_source" : {
24          "title" : "Kill Bill: Vol. 1",
25          "director" : "Quentin Tarantino",
26          "year" : 2003,
27          "genres" : [
28            "Action",
29            "Crime",
30            "Thriller"
31        ]
32      }
33    ]
34  }
35 }
36 }
37 }
```

میتوان خلاصه شده این کوئری یعنی بدون filter را بدین شکل نیز نوشت:

```
GET /movies/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "title": "kill"
          }
        },
        {
          "range": {
            "year": {
              "gte": 1960
            }
          }
        },
        {
          "term": {
            "genres": {
              "value": "drama"
            }
          }
        }
      ]
    }
  }
}
```

```
5  "shards": {
6    "total": 1,
7    "successful": 1,
8    "skipped": 0,
9    "failed": 0
10 },
11 "hits": {
12   "total": {
13     "value": 1,
14     "relation": "eq"
15   },
16   "max_score": 2.3262625,
17   "hits": [
18     {
19       "_index": "movies",
20       "_type": "movie",
21       "_id": "3",
22       "_score": 2.3262625,
23       "_source": {
24         "title": "To Kill a Mockingbird",
25         "director": "Robert Mulligan",
26         "year": 1962,
27         "genres": [
28           "Crime",
29           "Drama",
30           "Mystery"
31         ]
32       }
33     }
34   ]
35 }
36 }
```

حال اگر بخواهیم تنها بر اساس ژانر و سال یعنی بخش داخل فیلتر، داده‌ها را جستجو کنیم و جستجوی متن، انجام ندهیم میتوان از `match_all` استفاده کرد:

```
GET /movies/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match_all": {}
        },
        {
          "range": {
            "year": {
              "gte": 1960
            }
          }
        },
        {
          "term": {
            "genres": {
              "value": "drama"
            }
          }
        }
      ]
    }
  }
}
```

```
2  {
3    "took": 1,
4    "timed_out": false,
5    "shards": {
6      "total": 1,
7      "successful": 1,
8      "skipped": 0,
9      "failed": 0
10   },
11   "hits": {
12     "total": {
13       "value": 4,
14       "relation": "eq"
15     },
16     "max_score": 2.3257577,
17     "hits": [
18       {
19         "_index": "movies",
20         "_type": "movie",
21         "_id": "4",
22         "_score": 2.3257577,
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   }
35 }
```

معادل کوئری قبل بدون استفاده از فیلتر و با جستجو به صورت زیر می‌شود:

```
GET /movies/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "range": {
            "year": {
              "gte": 2000
            }
          }
        },
        {
          "term": {
            "genres": {
              "value": "drama"
            }
          }
        }
      ]
    }
  }
}
```

```
2  {
3    "took": 0,
4    "timed_out": false,
5    "shards": {
6      "total": 1,
7      "successful": 1,
8      "skipped": 0,
9      "failed": 0
10   },
11   "hits": {
12     "total": {
13       "value": 1,
14       "relation": "eq"
15     },
16     "max_score": 1.2795144,
17     "hits": [
18       {
19         "_index": "movies",
20         "_type": "movie",
21         "_id": "6",
22         "_score": 1.2795144,
23         "_source": {
24           "title": "The Assassination of Jesse James by the Coward Robert Ford",
25           "director": "Andrew Dominik",
26           "year": 2007,
27           "genres": [
28             "Biography",
29             "Crime",
30             "Drama"
31           ]
32         }
33       }
34     ]
35 }
```

با fuzziness می‌توانیم کلماتی را که غلط املایی دارند پیدا کنیم. در مثال زیر با اینکه godfather غلط املایی دارد اما باز هم پیدا میشود:

```
GET /movies/_search
{
  "query": {
    "match": {
      "title": {
        "query": "godfater",
        "fuzziness": "AUTO"
      }
    }
  }
}
```

```
cluster could be accessible to anyone. See http
/reference/7.16/security-minimal-setup.html to

2- {
3   "took" : 44,
4   "timed_out" : false,
5   "_shards" : {
6     "total" : 1,
7     "successful" : 1,
8     "skipped" : 0,
9     "failed" : 0
10  },
11  "hits" : {
12    "total" : {
13      "value" : 1,
14      "relation" : "eq"
15    },
16    "max_score" : 1.7120997,
17    "hits" : [
18      {
19        "_index" : "movies",
20        "_type" : "movie",
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  }
31 }
```

بعضی اوقات می‌خواهیم بعضی کلمات دقیقاً پشت سر هم باشند. با استفاده از slops این کار را انجام می‌دهیم:

```
GET /movies/_search
{
  "query": {
    "match_phrase": {
      "title": {
        "query": "Kill Bill",
        "slop": 2
      }
    }
  }
}
```

```
5-   "_shards" : {
6     "total" : 1,
7     "successful" : 1,
8     "skipped" : 0,
9     "failed" : 0
10  },
11  "hits" : {
12    "total" : {
13      "value" : 1,
14      "relation" : "eq"
15    },
16    "max_score" : 2.6128197,
17    "hits" : [
18      {
19        "_index" : "movies",
20        "_type" : "movie",
21        "_id" : "5",
22        "_score" : 2.6128197,
23        "_source" : {
24          "title" : "Kill Bill: Vol. 1",
25          "director" : "Quentin Tarantino",
26          "year" : 2003,
27          "genres" : [
28            "Action",
29            "Crime",
30            "Thriller"
31          ]
32        }
33      }
34    ]
35  }
36 }
```

در اینجا حداکثر دو فضای خالی بین هر کلمه تا بعدی را می‌توانیم قبول کنیم.

یک اشتباه رایج کوئری به شکل زیر است:

```
GET /movies/_search
{
  "query": {
    "term": {
      "director": {
        "value": "Francis Ford Coppola"
      }
    }
  }
}
```

```
1  #! Elasticsearch built-in s
2  cluster could be accessib
3  /reference/7.16/security-
4  {
5    "took" : 1,
6    "timed_out" : false,
7    "_shards" : {
8      "total" : 1,
9      "successful" : 1,
10     "skipped" : 0,
11     "failed" : 0
12   },
13   "hits" : {
14     "total" : {
15       "value" : 0,
16       "relation" : "eq"
17     },
18     "max_score" : null,
19     "hits" : [ ]
20   }
```

چون الاستیک کلمه‌ها را جدا کرده و هر کدام را ایندکس میکند برای کوئری بالا جواب پیدا نمیشود. برای حل این مشکل از keyword استفاده میکنیم:

```
GET movies/_search
{
  "query": {
    "term": {
      "director.keyword": "Francis Ford Coppola"
    }
  }
}
```

```
6    "total" : 1,
7    "successful" : 1,
8    "skipped" : 0,
9    "failed" : 0
10  },
11  "hits" : {
12    "total" : {
13      "value" : 2,
14      "relation" : "eq"
15    },
16    "max_score" : 1.0296195,
17    "hits" : [
18      {
19        "_index" : "movies",
20        "_type" : "movie",
21        "_id" : "1",
22        "_score" : 1.0296195,
23        "_source" : {
24          "title" : "The Godfather",
25          "director" : "Francis Ford Coppola",
26          "year" : 1972
27        }
28      },
29      {
30        "_index" : "movies",
31        "_type" : "movie",
32        "_id" : "4",
33        "_score" : 1.0296195,
34        "_source" : {
35          "title" : "Apocalypse Now",
36          "director" : "Francis Ford Coppola",
37          "year" : 1979,
38          "genres" : [

```

مشاهده میکنیم که با استفاده از keyword جواب مورد نظر ما پیدا شده است.

## گام دوم:

ابتدا با کد داده شده بیش از ۵۰۰ توییت را ذخیره کردیم:

```

Fetched and inserted 492 tweets so far
Waiting for 30 seconds...
Fetched and inserted 502 tweets so far
Waiting for 30 seconds...
PS C:\>

```

ابتدا dashboard و index pattern را میسازیم.

حال ابر کلمات را میسازیم:

شاخص\_بورس  
زقیام  
عکورش  
قنابت شستا  
برکت  
وایران نوری  
خاور فولاد  
کبافی  
های\_وب

شاخص\_بورس  
زقیام  
عکورش  
قنابت شستا  
برکت  
وایران نوری  
خاور فولاد  
کبافی  
های\_وب

twitter\*

Data Options

Metrics

> Tag size Count

Buckets

Tags

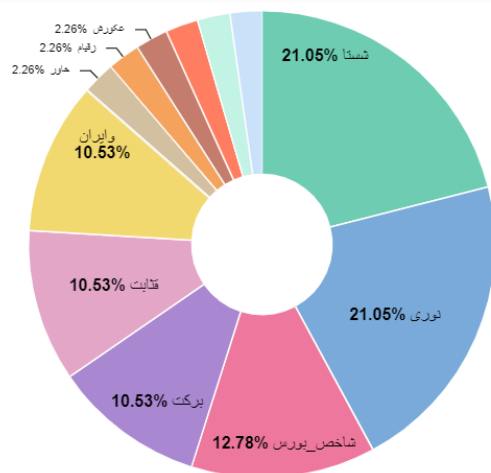
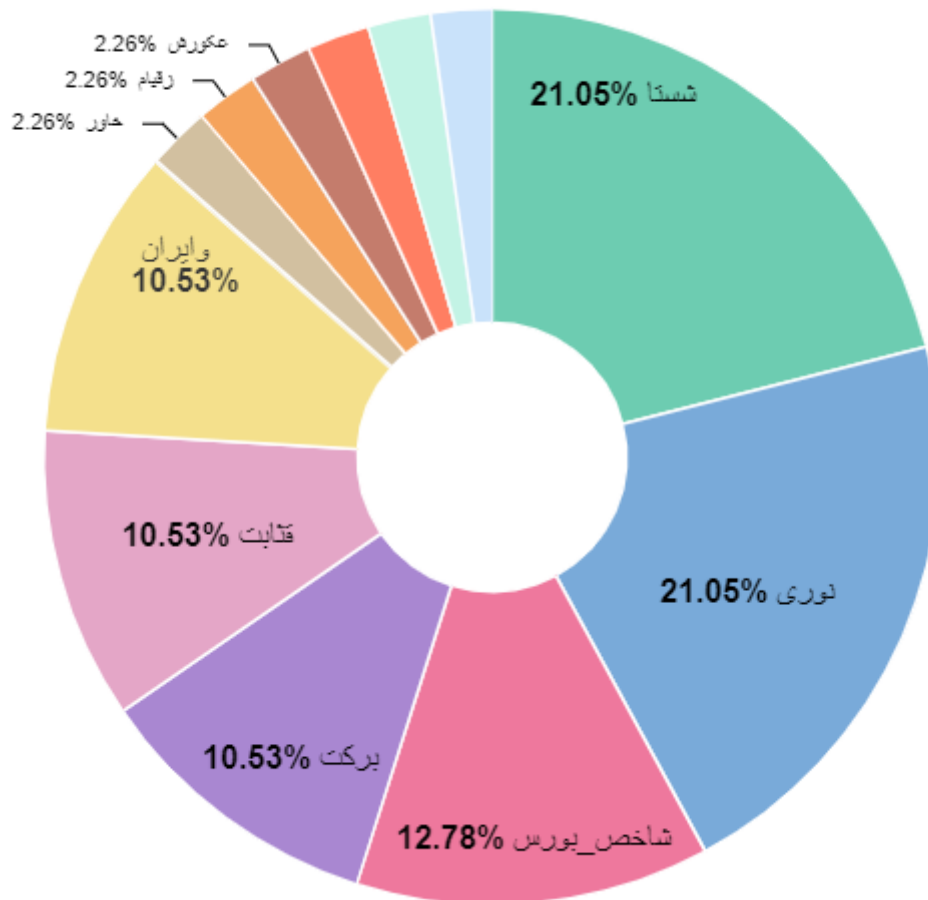
Aggregation [Terms help](#)

Terms

Field

hashtags.keyword

و بعد از آن نمودار Pie:



twitter\*

Data

Options

Metric: Count

Order

Descending

Size

50

☐ Group other values in separate bucket

☐ Show missing values

Custom label

> Advanced

+ Add

×

 Discard
 

⌂

▶ Update

## مشکلات و توضیحات تکمیلی

---

در کد داده شده برای دریافت توییت یک اشکال ریز وجود داشت که با کمک دوستان رفع کردم.



## آنچه آموختم / پیشنهادات

---

گزارش کار در عین خلاصه بودن بسیار مفید بود و باعث شد دید مناسبی نسبت به این پایگاه داده پیدا کنم.