# Elasticsearch Database Report

Junhui LI, Ge QIU, Zhiheng WU, Yue JI

Group : A4 IBO4

**Dataset :**

Stock exchange

**Abstract :**

Our group uses the data set named stock exchange, each item contains attributes such as id, compamy, price, date, etc. We design related queries based on these attributes.

**1. Introduction**

Elasticsearch is an open source search engine highly scalable. It allows you to keep and analyse a great volume of information practically in real time.

Elasticsearch works with JSON documents files. Using an internal structure, it can parse your data in almost real time to search for the information you need.

**2. Data prepossessing**

Elasticsearch has features like *index* and *_type*. So we need to process the data before inserting the data. We add types and indexes in bulk in the dataset. Here is the program python for adding.

```
import json
import os

DATASET_FILE = 'stocks.json'
BASE_FILE = 'movielens_usersRating.json'
TYPENAME = 'stock'
INDEXNAME = 'stocks'

if not os.path.isfile(DATASET_FILE):
    print("[#] starting json modification...")
    with open(BASE_FILE, "r") as f:
        with open(DATASET_FILE, "a+") as f2:
            for line in f:
                data = json.loads(line.strip('\n'))
                print data

                index = {"index":{"_index":INDEXNAME, "_type":TYPENAME,
"_id":data["_id"]}}
                data.pop("_id")
                fields={}
                fields["fields"] = data
                # data["field"] = data["_id"]["$oid"]
                f2.write(json.dumps(index)+'\n')
                f2.write(json.dumps(fields)+'\n')
    print("[#] modification done")
```

#### 3. Import dataset with cURL

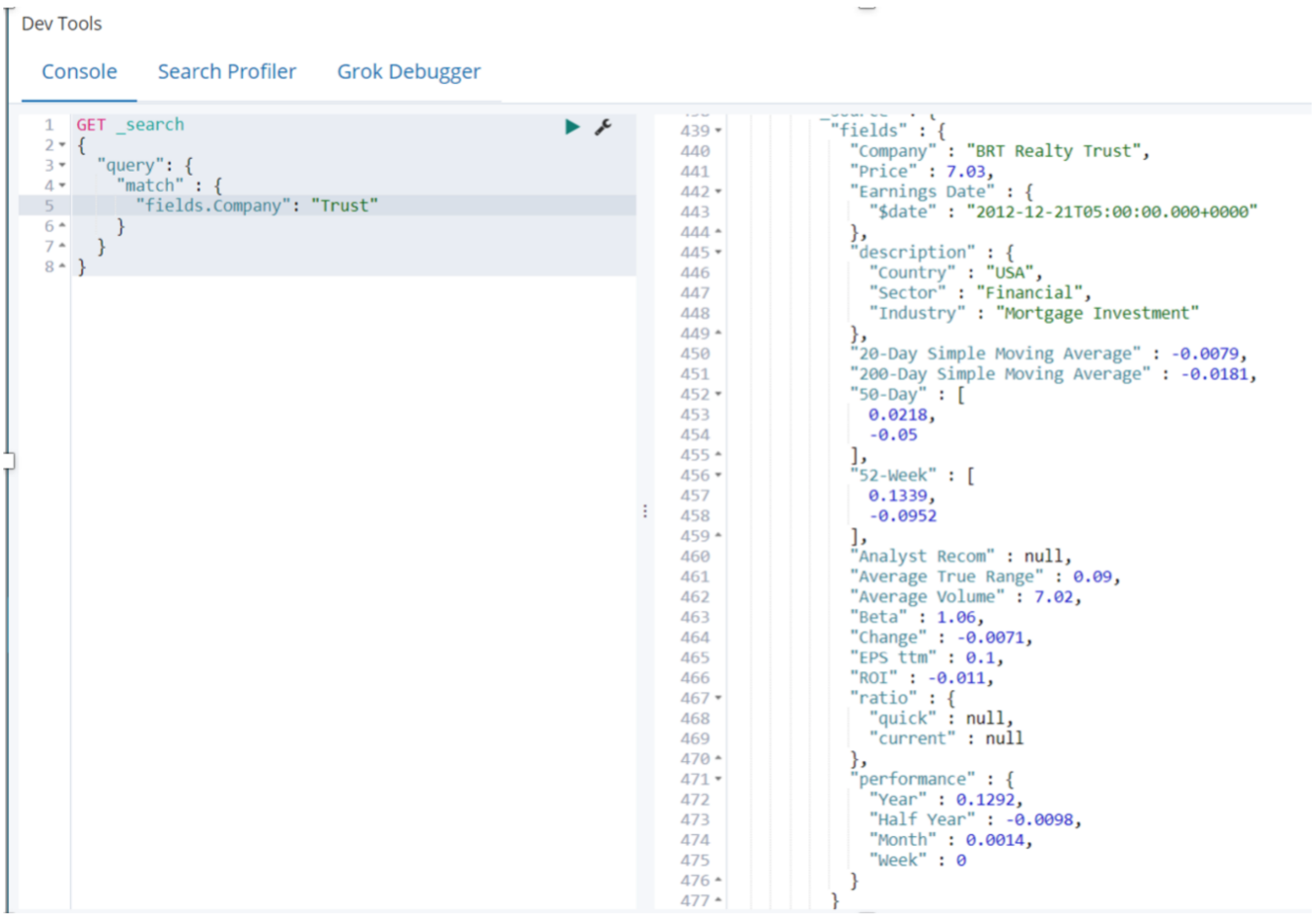
```
curl -XPUT localhost:9200/_bulk -H"Content-Type: application/json" --data-
binary @stock.json
```

#### 4. Queries

- Simple queries:

  *1. find the data of company name which contains "Trust"(simple match)*

```
GET _search
{
  "query": {
    "match" : {
      "fields.Company": "Trust"
    }
  }
}
```

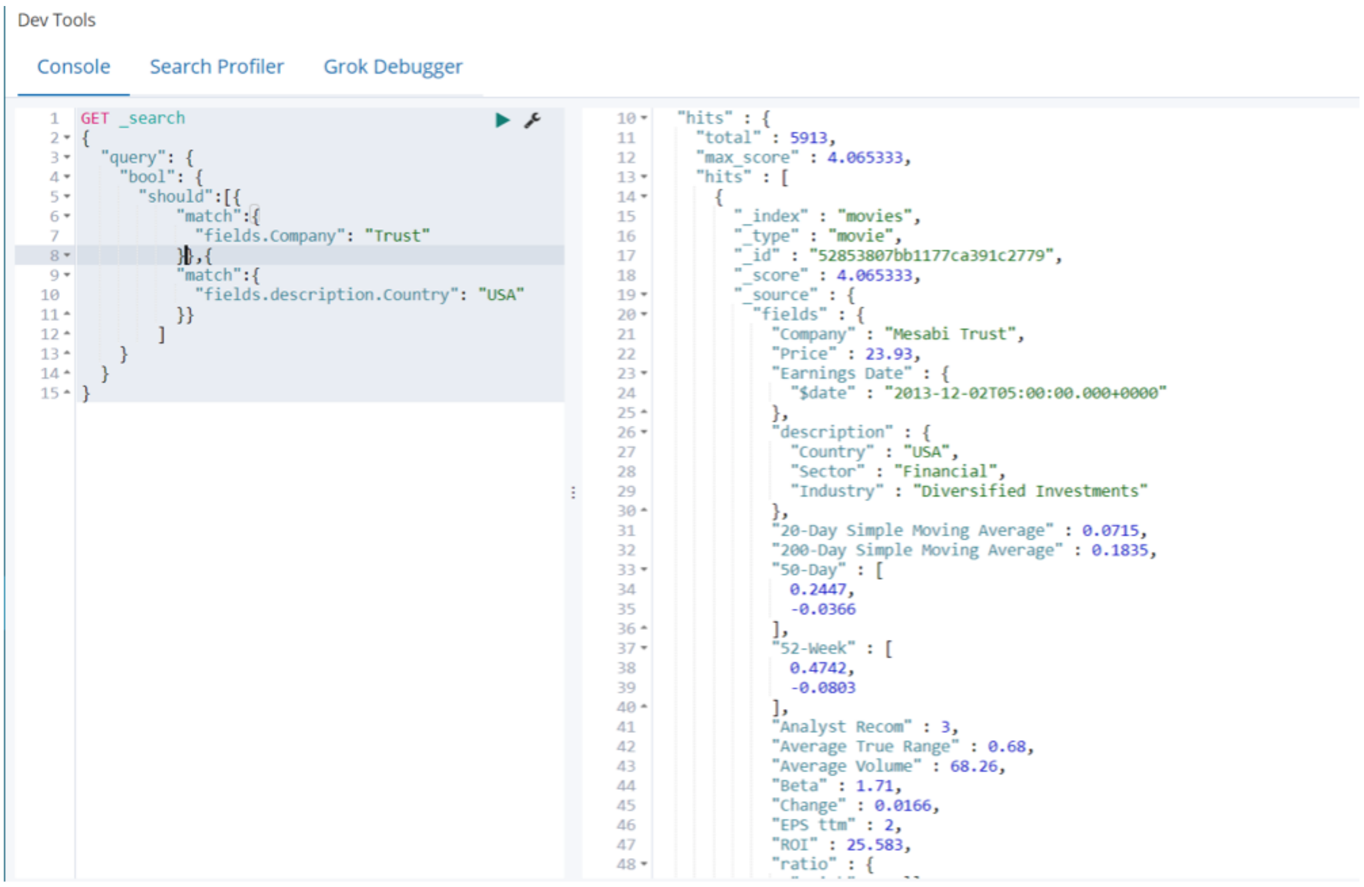


```
GET _search
{
  "query": {
    "match" : {
      "fields.Company": "Trust"
    }
  }
}
```

```
  "fields" : {
    "Company" : "BRT Realty Trust",
    "Price" : 7.03,
    "Earnings Date" : {
      "$date" : "2012-12-21T05:00:00.000+0000"
    },
    "description" : {
      "Country" : "USA",
      "Sector" : "Financial",
      "Industry" : "Mortgage Investment"
    },
    "20-Day Simple Moving Average" : -0.0079,
    "200-Day Simple Moving Average" : -0.0181,
    "50-Day" : [
      0.0218,
      -0.05
    ],
    "52-Week" : [
      0.1339,
      -0.0952
    ],
    "Analyst Recom" : null,
    "Average True Range" : 0.09,
    "Average Volume" : 7.02,
    "Beta" : 1.06,
    "Change" : -0.0071,
    "EPS ttm" : 0.1,
    "ROI" : -0.011,
    "ratio" : {
      "quick" : null,
      "current" : null
    },
    "performance" : {
      "Year" : 0.1292,
      "Half Year" : -0.0098,
      "Month" : 0.0014,
      "Week" : 0
    }
  }
```

*2. match the company contains ”Trust” and Country contains "USA"(simple should)*

```
GET _search
{
  "query": {
    "bool": {
      "should":[{
          "match":{
            "fields.Company": "Trust"
          }},{
          "match":{
            "fields.description.Country": "USA"
          }}
        ]
    }
  }
}
```

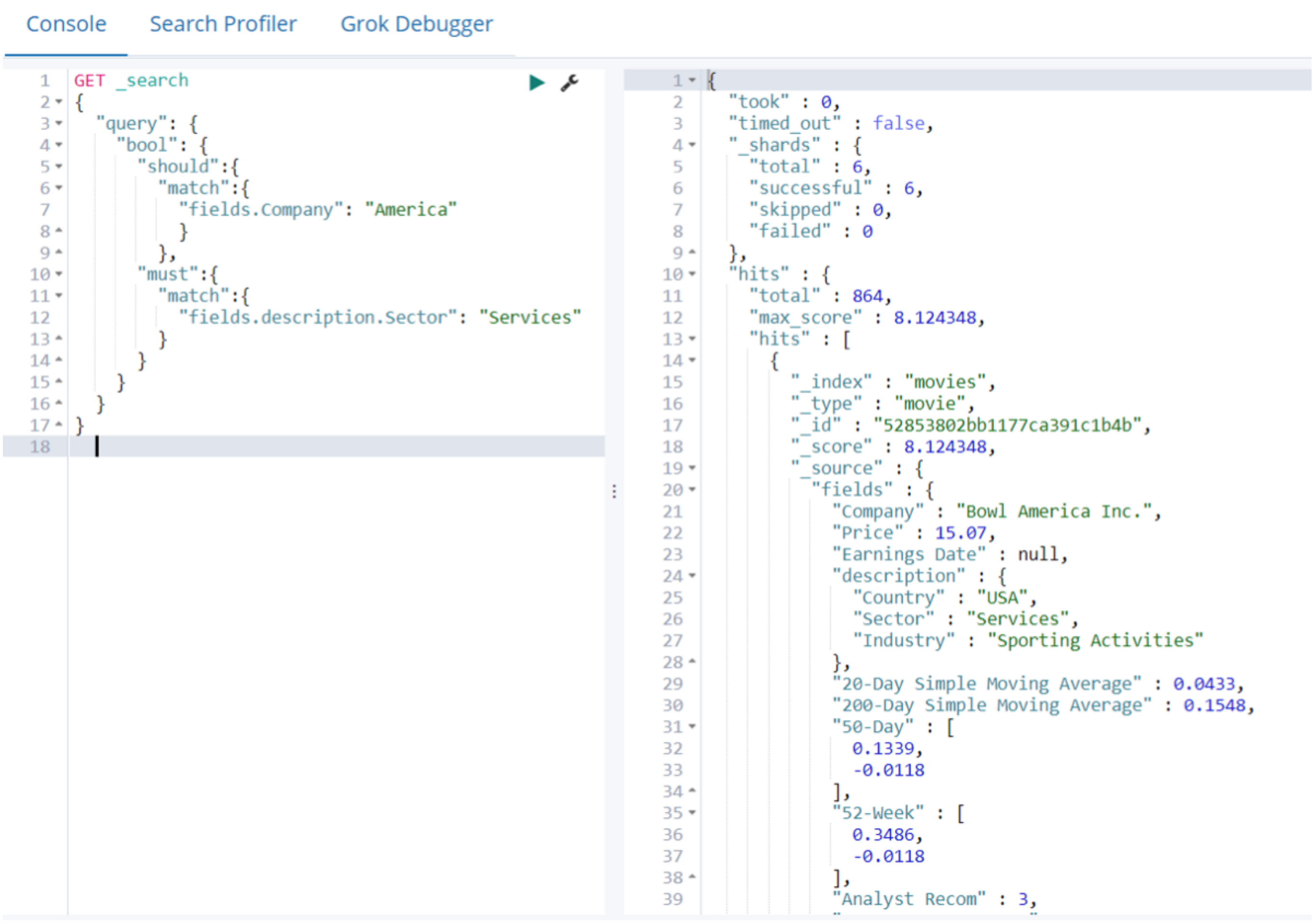


```
GET _search
{
  "query": {
    "bool": {
      "should":[{
          "match":{
            "fields.Company": "Trust"
          }},{
          "match":{
            "fields.description.Country": "USA"
          }}
        ]
    }
  }
}
```

```
"hits" : {
  "total" : 5913,
  "max_score" : 4.065333,
  "hits" : [
    {
      "_index" : "movies",
      "_type" : "movie",
      "_id" : "52853807bb1177ca391c2779",
      "_score" : 4.065333,
      "_source" : {
        "fields" : {
          "Company" : "Mesabi Trust",
          "Price" : 23.93,
          "Earnings Date" : {
            "$date" : "2013-12-02T05:00:00.000+0000"
          },
          "description" : {
            "Country" : "USA",
            "Sector" : "Financial",
            "Industry" : "Diversified Investments"
          },
          "20-Day Simple Moving Average" : 0.0715,
          "200-Day Simple Moving Average" : 0.1835,
          "50-Day" : [
            0.2447,
            -0.0366
          ],
          "52-Week" : [
            0.4742,
            -0.0803
          ],
          "Analyst Recom" : 3,
          "Average True Range" : 0.68,
          "Average Volume" : 68.26,
          "Beta" : 1.71,
          "Change" : 0.0166,
          "EPS ttm" : 2,
          "ROI" : 25.583,
          "ratio" : {
```

*3. find that the Sector must be Services and Company should contains "America"*

```
GET _search
{
  "query": {
    "bool": {
      "should":{
        "match":{
          "fields.Company": "America"
          }
        },
      "must":{
        "match":{
          "fields.description.Sector": "Services"
        }
      }
    }
  }
}
```

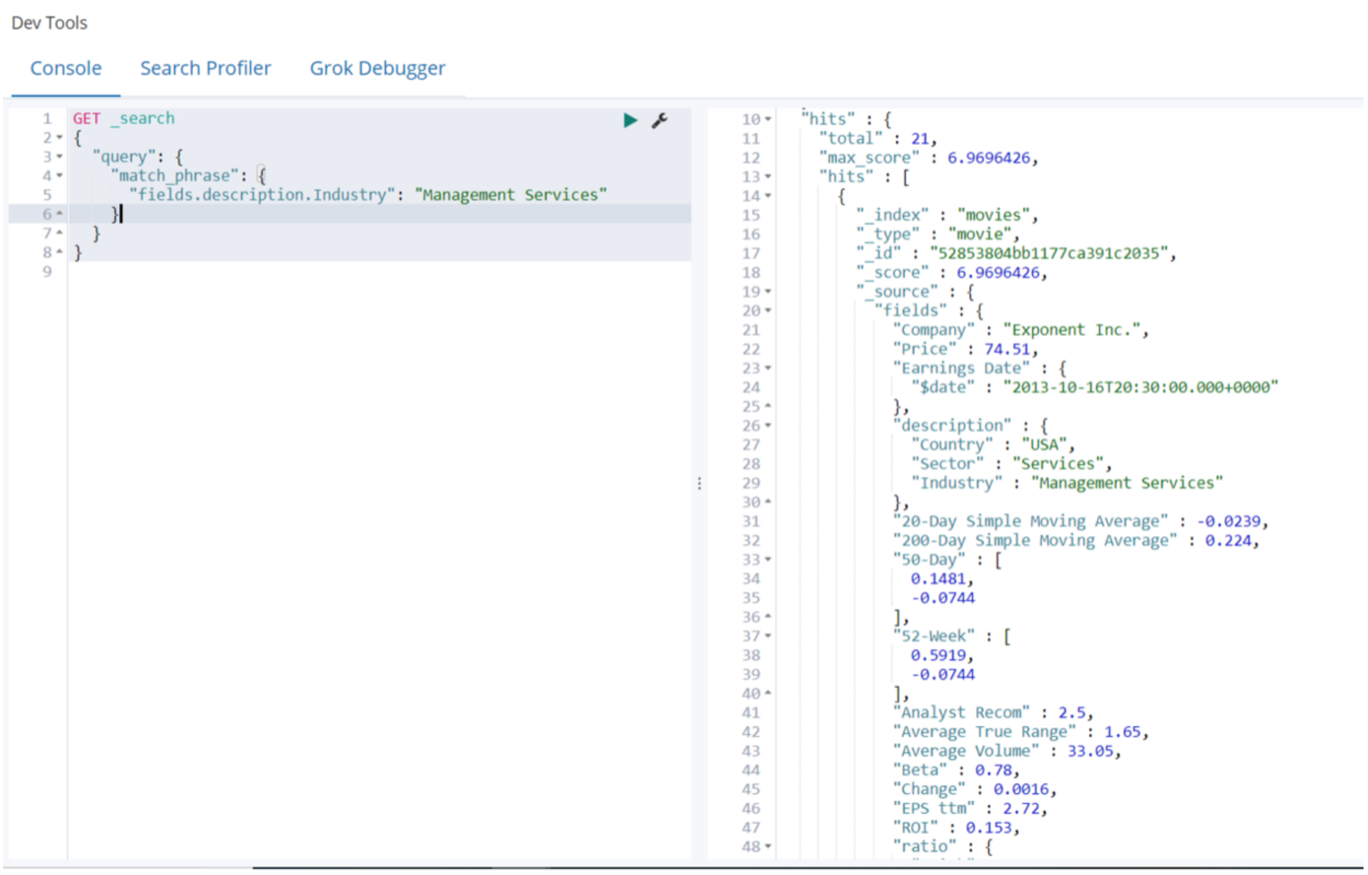


```
GET _search
{
  "query": {
    "bool": {
      "should":{
        "match":{
          "fields.Company": "America"
        }
      },
      "must":{
        "match":{
          "fields.description.Sector": "Services"
        }
      }
    }
  }
}
```

```
{
  "took" : 0,
  "timed_out" : false,
  "_shards" : {
    "total" : 6,
    "successful" : 6,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 864,
    "max_score" : 8.124348,
    "hits" : [
      {
        "_index" : "movies",
        "_type" : "movie",
        "_id" : "52853802bb1177ca391c1b4b",
        "_score" : 8.124348,
        "_source" : {
          "fields" : {
            "Company" : "Bowl America Inc.",
            "Price" : 15.07,
            "Earnings Date" : null,
            "description" : {
              "Country" : "USA",
              "Sector" : "Services",
              "Industry" : "Sporting Activities"
            },
            "20-Day Simple Moving Average" : 0.0433,
            "200-Day Simple Moving Average" : 0.1548,
            "50-Day" : [
              0.1339,
              -0.0118
            ],
            "52-Week" : [
              0.3486,
              -0.0118
            ],
            "Analyst Recom" : 3,
```

*4. find the Industry named "Management Services"(simple match_phrase)*

```
GET _search
{
  "query": {
    "match_phrase": {
      "fields.description.Industry": "Management Services"
    }
  }
}
```

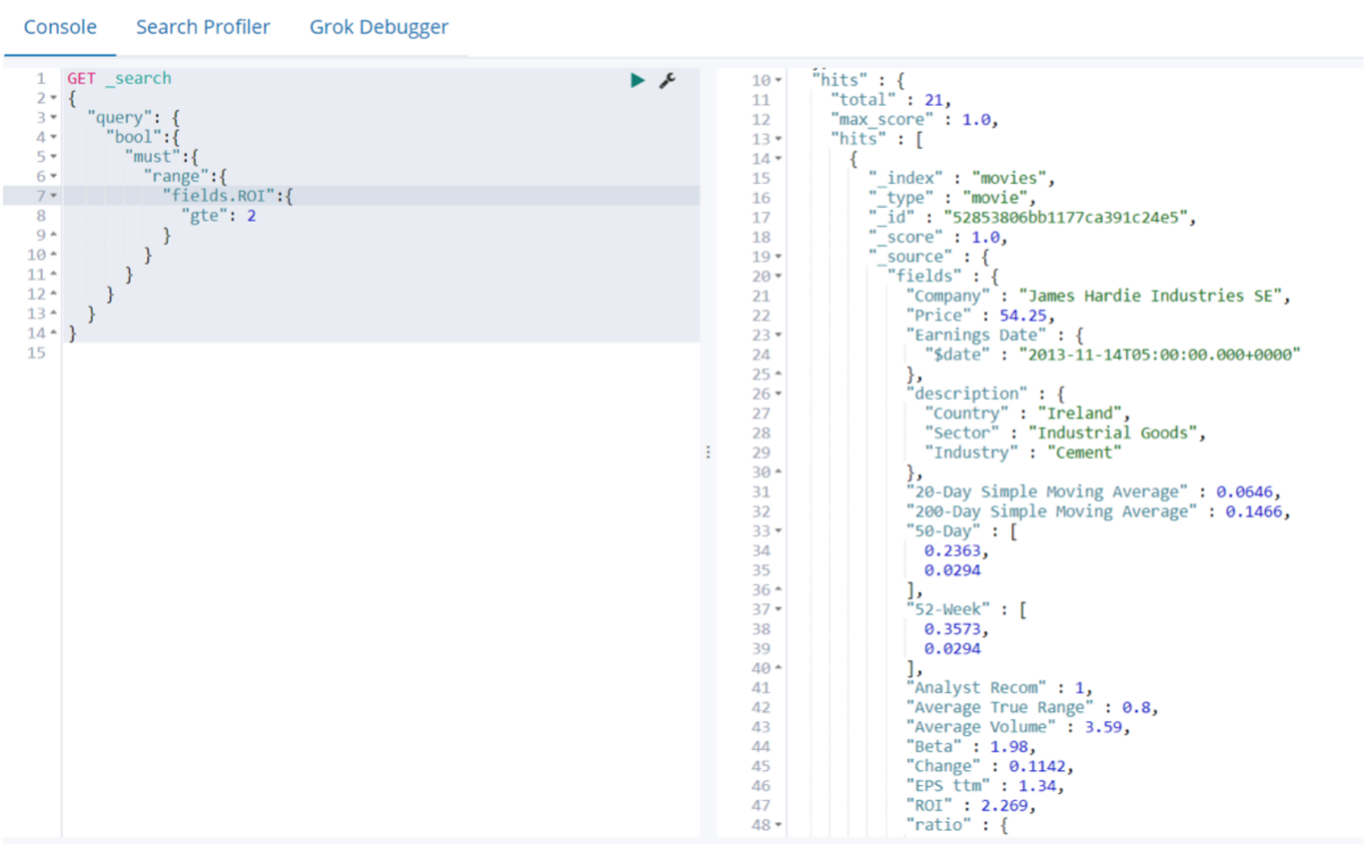


```
GET _search
{
  "query": {
    "match_phrase": {
      "fields.description.Industry": "Management Services"
    }
  }
}
```

```
"hits" : {
  "total" : 21,
  "max_score" : 6.9696426,
  "hits" : [
    {
      "_index" : "movies",
      "_type" : "movie",
      "_id" : "52853804bb1177ca391c2035",
      "_score" : 6.9696426,
      "_source" : {
        "fields" : {
          "Company" : "Exponent Inc.",
          "Price" : 74.51,
          "Earnings Date" : {
            "$date" : "2013-10-16T20:30:00.000+0000"
          },
          "description" : {
            "Country" : "USA",
            "Sector" : "Services",
            "Industry" : "Management Services"
          },
          "20-Day Simple Moving Average" : -0.0239,
          "200-Day Simple Moving Average" : 0.224,
          "50-Day" : [
            0.1481,
            -0.0744
          ],
          "52-Week" : [
            0.5919,
            -0.0744
          ],
          "Analyst Recom" : 2.5,
          "Average True Range" : 1.65,
          "Average Volume" : 33.05,
          "Beta" : 0.78,
          "Change" : 0.0016,
          "EPS ttm" : 2.72,
          "ROI" : 0.153,
          "ratio" : {
```

*5. find the ROI greater than 2(simple range)*

```
GET _search
{
  "query": {
    "bool":{
      "must":{
        "range":{
          "fields.ROI":{
            "gte": 2
          }
        }
      }
    }
  }
}
```

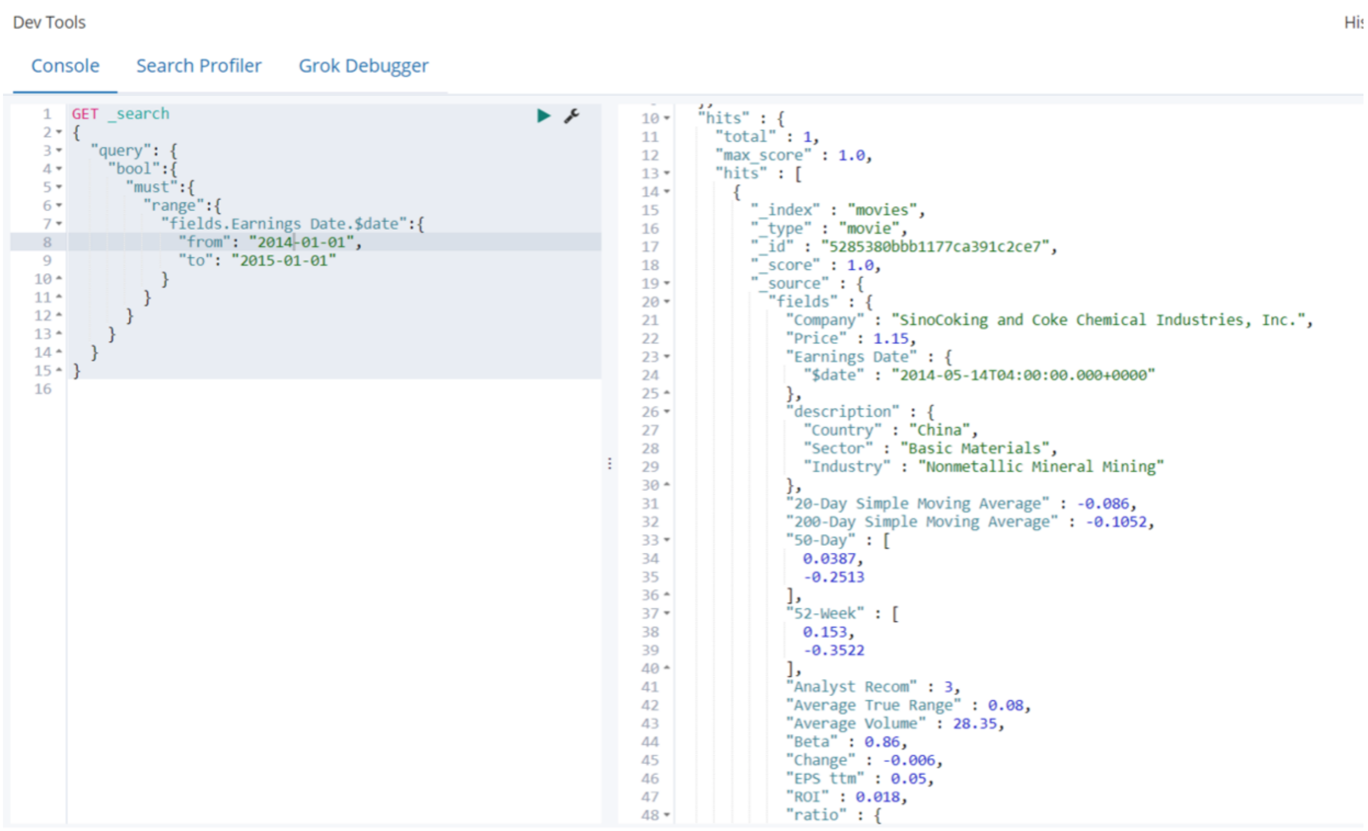


```
GET _search
{
  "query": {
    "bool":{
      "must":{
        "range":{
          "fields.ROI":{
            "gte": 2
          }
        }
      }
    }
  }
}
```

```
"hits" : {
  "total" : 21,
  "max_score" : 1.0,
  "hits" : [
    {
      "_index" : "movies",
      "_type" : "movie",
      "_id" : "52853806bb1177ca391c24e5",
      "_score" : 1.0,
      "_source" : {
        "fields" : {
          "Company" : "James Hardie Industries SE",
          "Price" : 54.25,
          "Earnings Date" : {
            "$date" : "2013-11-14T05:00:00.000+0000"
          },
          "description" : {
            "Country" : "Ireland",
            "Sector" : "Industrial Goods",
            "Industry" : "Cement"
          },
          "20-Day Simple Moving Average" : 0.0646,
          "200-Day Simple Moving Average" : 0.1466,
          "50-Day" : [
            0.2363,
            0.0294
          ],
          "52-Week" : [
            0.3573,
            0.0294
          ],
          "Analyst Recom" : 1,
          "Average True Range" : 0.8,
          "Average Volume" : 3.59,
          "Beta" : 1.98,
          "Change" : 0.1142,
          "EPS ttm" : 1.34,
          "ROI" : 2.269,
          "ratio" : {
```

*6. find the Earning Date that between 2014-01-01 and 2015-01-01(simple date)*

```
GET _search
{
  "query": {
    "bool":{
      "must":{
        "range":{
          "fields.Earnings Date.$date":{
            "from": "2014-01-01",
            "to": "2015-01-01"
          }
        }
      }
    }
  }
}
```



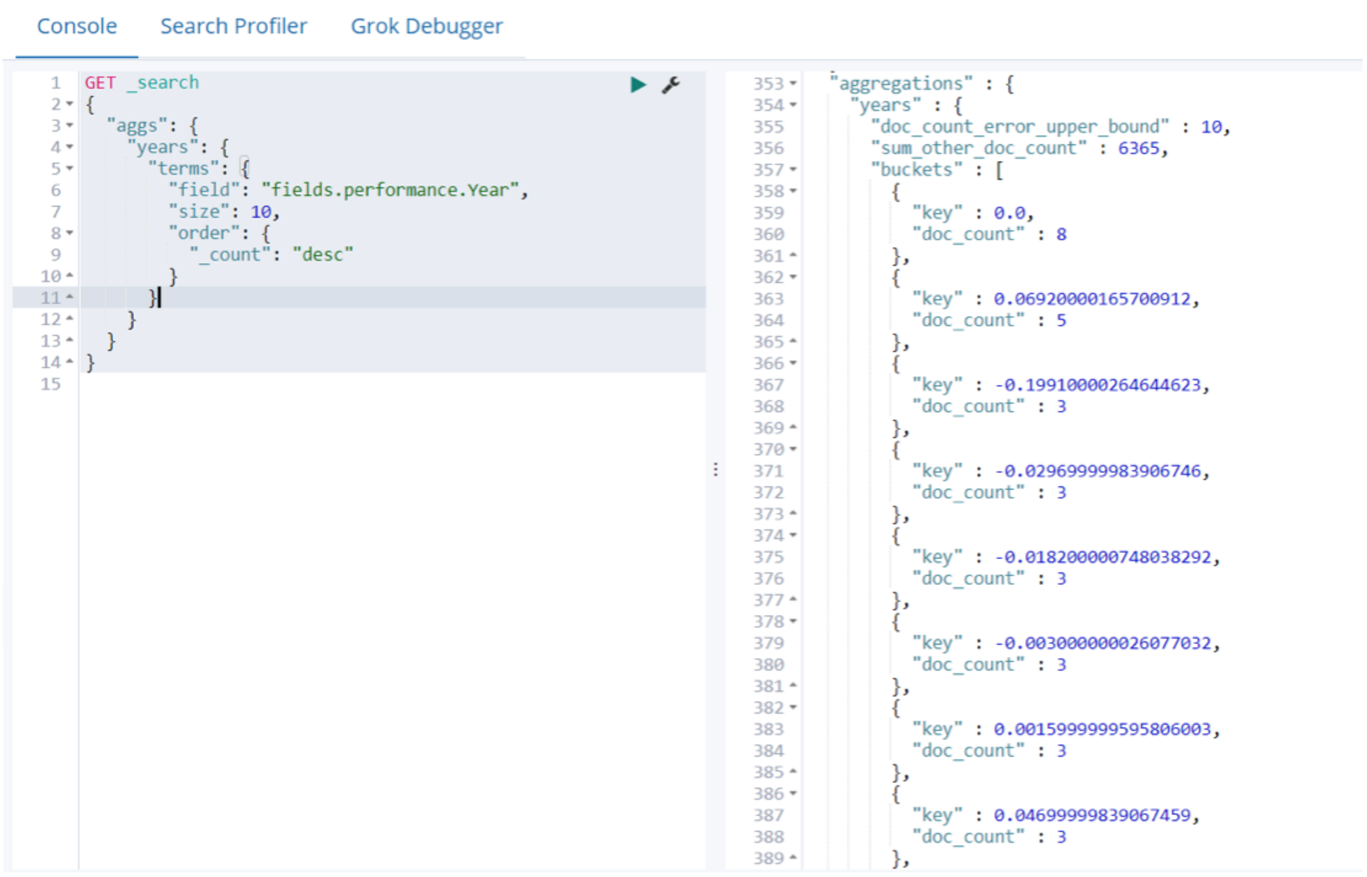
```
GET _search
{
  "query": {
    "bool":{
      "must":{
        "range":{
          "fields.Earnings Date.$date":{
            "from": "2014-01-01",
            "to": "2015-01-01"
          }
        }
      }
    }
  }
}
```

```
  "hits" : {
    "total" : 1,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "movies",
        "_type" : "movie",
        "_id" : "5285380bbb1177ca391c2ce7",
        "_score" : 1.0,
        "_source" : {
          "fields" : {
            "Company" : "SinoCoking and Coke Chemical Industries, Inc.",
            "Price" : 1.15,
            "Earnings Date" : {
              "$date" : "2014-05-14T04:00:00.000+0000"
            },
            "description" : {
              "Country" : "China",
              "Sector" : "Basic Materials",
              "Industry" : "Nonmetallic Mineral Mining"
            },
            "20-Day Simple Moving Average" : -0.086,
            "200-Day Simple Moving Average" : -0.1052,
            "50-Day" : [
              0.0387,
              -0.2513
            ],
            "52-Week" : [
              0.153,
              -0.3522
            ],
            "Analyst Recom" : 3,
            "Average True Range" : 0.08,
            "Average Volume" : 28.35,
            "Beta" : 0.86,
            "Change" : -0.006,
            "EPS ttm" : 0.05,
            "ROI" : 0.018,
            "ratio" : {
```

- Complex queries:

  *1. show the year performance aggregation in desc order(simple terms)*

```
GET _search
{
  "aggs": {
    "years": {
      "terms": {
        "field": "fields.performance.Year",
        "size": 10,
        "order": {
          "_count": "desc"
        }
      }
    }
  }
}
```

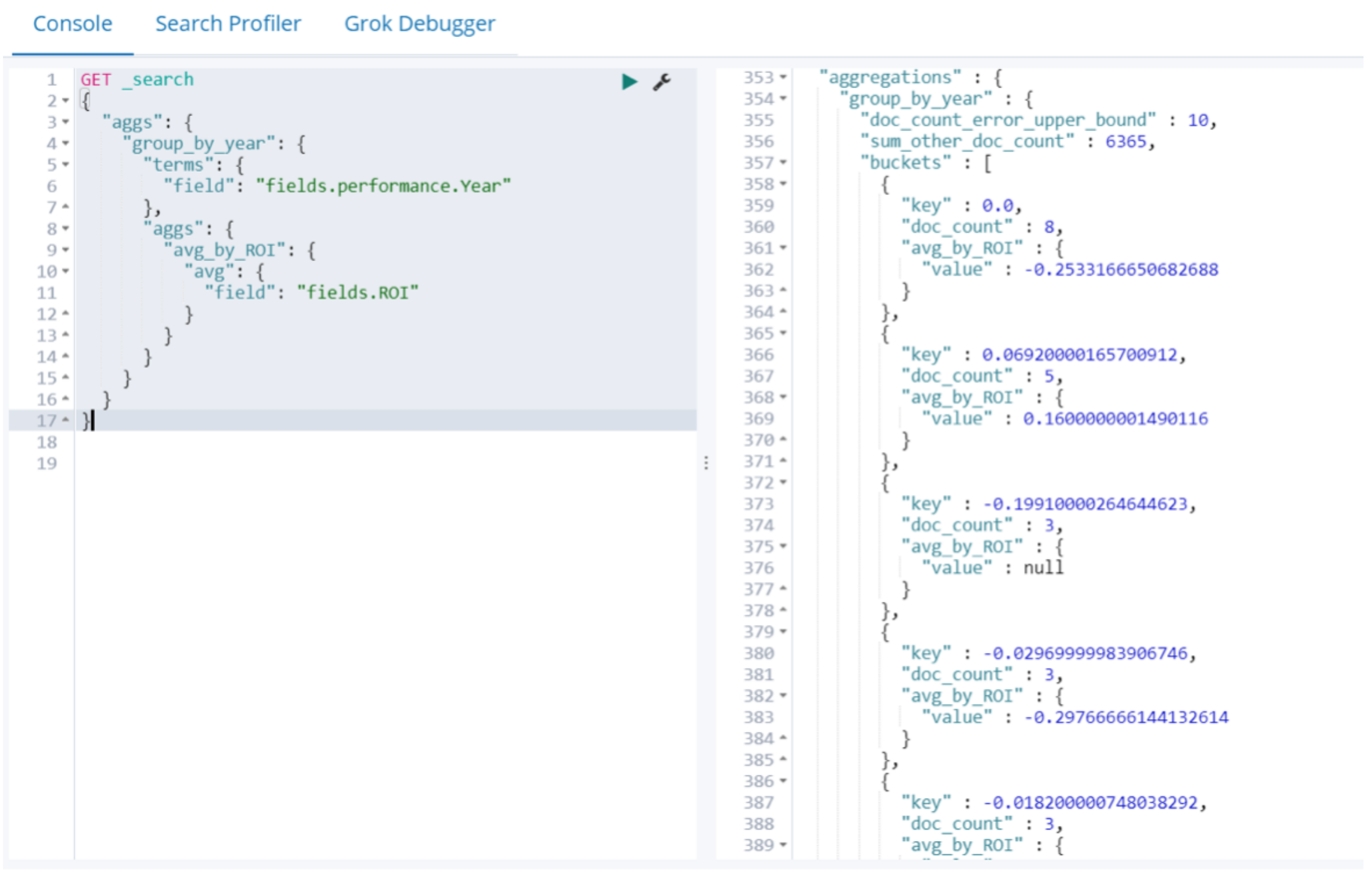


```
GET _search
{
  "aggs": {
    "years": {
      "terms": {
        "field": "fields.performance.Year",
        "size": 10,
        "order": {
          "_count": "desc"
        }
      }
    }
  }
}
```

```
"aggregations" : {
  "years" : {
    "doc_count_error_upper_bound" : 10,
    "sum_other_doc_count" : 6365,
    "buckets" : [
      {
        "key" : 0.0,
        "doc_count" : 8
      },
      {
        "key" : 0.06920000165700912,
        "doc_count" : 5
      },
      {
        "key" : -0.19910000264644623,
        "doc_count" : 3
      },
      {
        "key" : -0.02969999983906746,
        "doc_count" : 3
      },
      {
        "key" : -0.018200000748038292,
        "doc_count" : 3
      },
      {
        "key" : -0.003000000026077032,
        "doc_count" : 3
      },
      {
        "key" : 0.0015999999595806003,
        "doc_count" : 3
      },
      {
        "key" : 0.04699999839067459,
        "doc_count" : 3
      },
```

2. *Group by performance.year and find their average ROI(composite)*

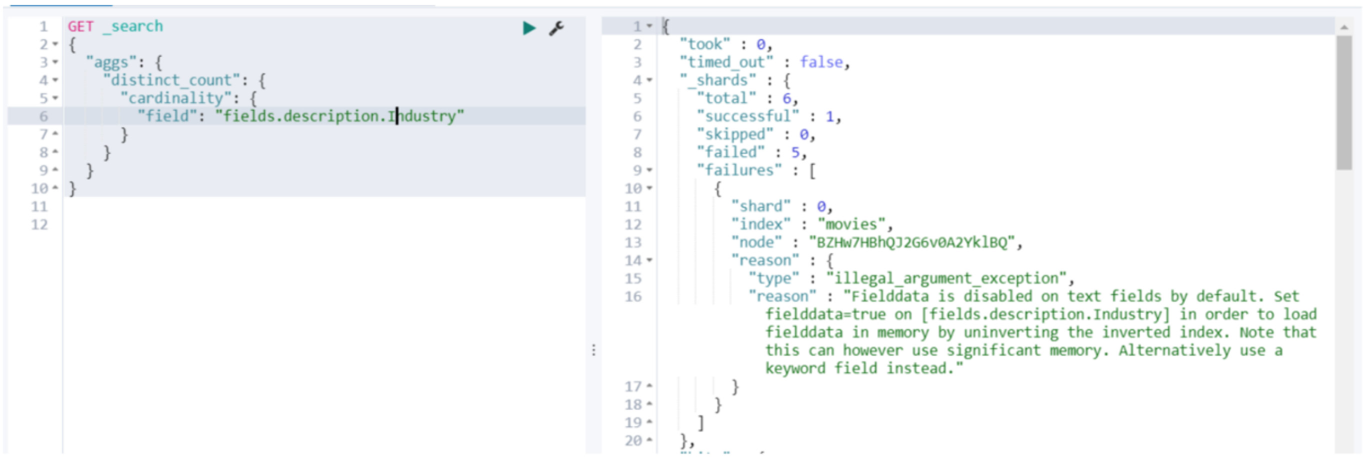
```
GET _search
{
  "aggs": {
    "group_by_year": {
      "terms": {
        "field": "fields.performance.Year"
      },
      "aggs": {
        "avg_by_ROI": {
          "avg": {
            "field": "fields.ROI"
          }
        }
      }
    }
  }
}
```



```
GET _search
{
  "aggs": {
    "group_by_year": {
      "terms": {
        "field": "fields.performance.Year"
      },
      "aggs": {
        "avg_by_ROI": {
          "avg": {
            "field": "fields.ROI"
          }
        }
      }
    }
  }
}
```

```
  "aggregations" : {
    "group_by_year" : {
      "doc_count_error_upper_bound" : 10,
      "sum_other_doc_count" : 6365,
      "buckets" : [
        {
          "key" : 0.0,
          "doc_count" : 8,
          "avg_by_ROI" : {
            "value" : -0.2533166650682688
          }
        },
        {
          "key" : 0.06920000165700912,
          "doc_count" : 5,
          "avg_by_ROI" : {
            "value" : 0.1600000001490116
          }
        },
        {
          "key" : -0.19910000264644623,
          "doc_count" : 3,
          "avg_by_ROI" : {
            "value" : null
          }
        },
        {
          "key" : -0.02969999983906746,
          "doc_count" : 3,
          "avg_by_ROI" : {
            "value" : -0.29766666144132614
          }
        },
        {
          "key" : -0.018200000748038292,
          "doc_count" : 3,
          "avg_by_ROI" : {
```

*3. number of distinct industry(cardinality)*



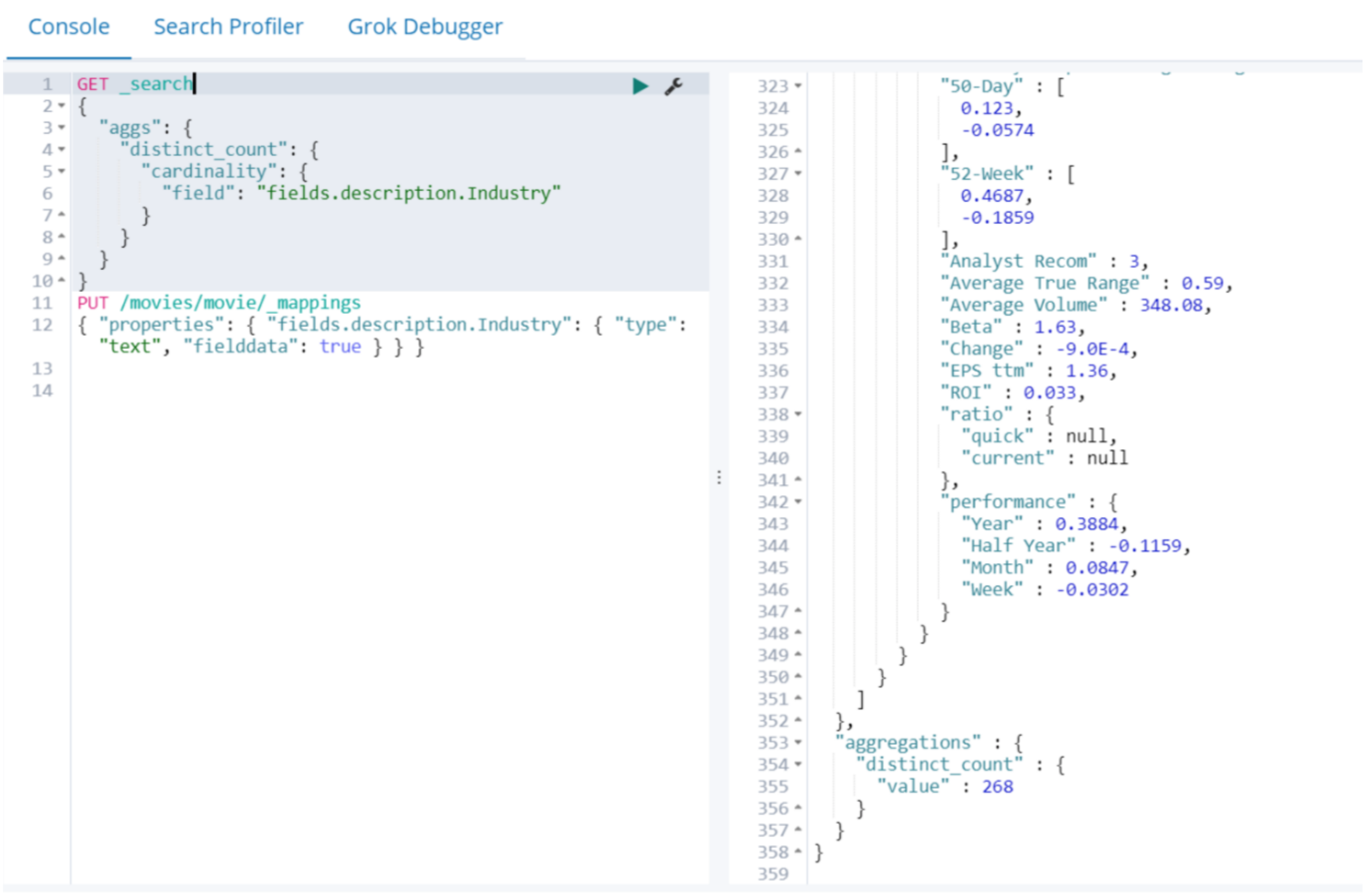
```
GET _search
{
  "aggs": {
    "distinct_count": {
      "cardinality": {
        "field": "fields.description.Industry"
      }
    }
  }
}
```

```
{
  "took" : 0,
  "timed_out" : false,
  "_shards" : {
    "total" : 6,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 5,
    "failures" : [
      {
        "shard" : 0,
        "index" : "movies",
        "node" : "BZHw7HBhQJ2G6v0A2YklBQ",
        "reason" : {
          "type" : "illegal_argument_exception",
          "reason" : "Fielddata is disabled on text fields by default. Set
            fielddata=true on [fields.description.Industry] in order to load
            fielddata in memory by uninverting the inverted index. Note that
            this can however use significant memory. Alternatively use a
            keyword field instead."
        }
      }
    ]
  },
```

So first put the industry as fielddata.

```
GET _search
{
  "aggs": {
    "distinct_count": {
      "cardinality": {
        "field": "fields.description.Industry"
      }
    }
  }
}
PUT /movies/movie/_mappings
```

```
{ "properties": { "fields.description.Industry": { "type": "text",
"fielddata": true } } }
```



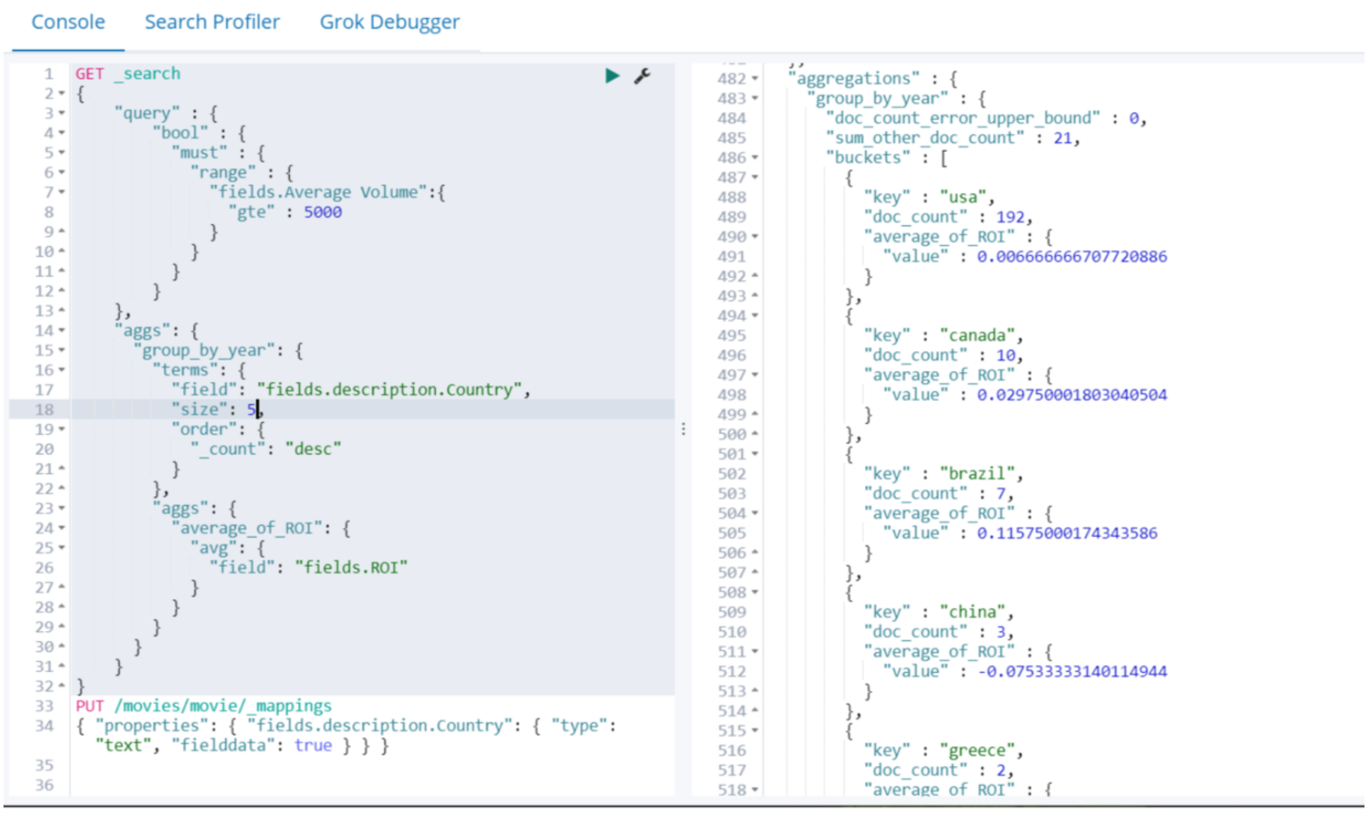
```
GET _search
{
  "aggs": {
    "distinct_count": {
      "cardinality": {
        "field": "fields.description.Industry"
      }
    }
  }
}
PUT /movies/movie/_mappings
{ "properties": { "fields.description.Industry": { "type":
  "text", "fielddata": true } } }
```

```
                    "50-Day" : [
                      0.123,
                      -0.0574
                    ],
                    "52-Week" : [
                      0.4687,
                      -0.1859
                    ],
                    "Analyst Recom" : 3,
                    "Average True Range" : 0.59,
                    "Average Volume" : 348.08,
                    "Beta" : 1.63,
                    "Change" : -9.0E-4,
                    "EPS ttm" : 1.36,
                    "ROI" : 0.033,
                    "ratio" : {
                      "quick" : null,
                      "current" : null
                    },
                    "performance" : {
                      "Year" : 0.3884,
                      "Half Year" : -0.1159,
                      "Month" : 0.0847,
                      "Week" : -0.0302
                    }
                  }
                }
              }
            }
          ]
        },
        "aggregations" : {
          "distinct_count" : {
            "value" : 268
          }
        }
      }
```

- Hard query:

  *filter the average volume that greater than 5000, then group by the performance of the year, finally calculate their average ROI*

```
GET _search
{
    "query" : {
        "bool" : {
          "must" : {
            "range" : {
              "fields.Average Volume":{
                "gte" : 5000
              }
            }
          }
        }
    },
    "aggs": {
      "group_by_year": {
        "terms": {
          "field": "fields.description.Country",
          "size": 5,
          "order": {
            "_count": "desc"
          }
        },
        "aggs": {
          "average_of_ROI": {
            "avg": {
              "field": "fields.ROI"
            }
          }
        }
      }
    }
  }
PUT /movies/movie/_mappings
{ "properties": { "fields.description.Country": { "type": "text",
"fielddata": true } } }
```

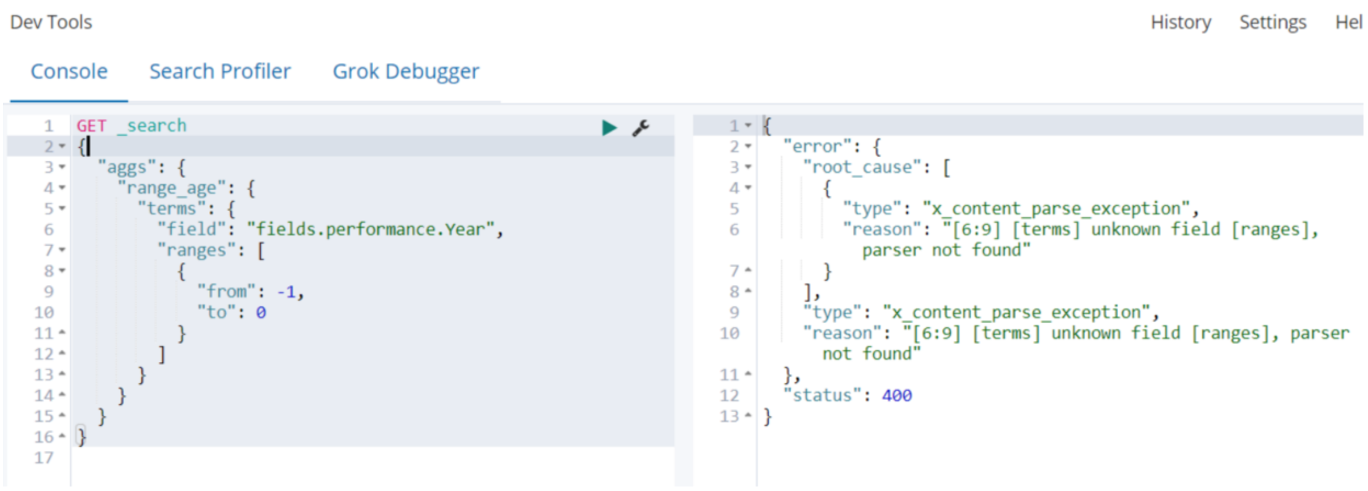


```
GET _search
{
    "query" : {
        "bool" : {
          "must" : {
            "range" : {
              "fields.Average Volume":{
                "gte" : 5000
              }
            }
          }
        }
    },
    "aggs": {
      "group_by_year": {
        "terms": {
          "field": "fields.description.Country",
          "size": 5,
          "order": {
            "_count": "desc"
          }
        },
        "aggs": {
          "average_of_ROI": {
            "avg": {
              "field": "fields.ROI"
            }
          }
        }
      }
    }
}
PUT /movies/movie/_mappings
{ "properties": { "fields.description.Country": { "type":
  "text", "fielddata": true } } }
```

```
  "aggregations" : {
    "group_by_year" : {
      "doc_count_error_upper_bound" : 0,
      "sum_other_doc_count" : 21,
      "buckets" : [
        {
          "key" : "usa",
          "doc_count" : 192,
          "average_of_ROI" : {
            "value" : 0.006666666707720886
          }
        },
        {
          "key" : "canada",
          "doc_count" : 10,
          "average_of_ROI" : {
            "value" : 0.029750001803040504
          }
        },
        {
          "key" : "brazil",
          "doc_count" : 7,
          "average_of_ROI" : {
            "value" : 0.11575000174343586
          }
        },
        {
          "key" : "china",
          "doc_count" : 3,
          "average_of_ROI" : {
            "value" : -0.07533333140114944
          }
        },
        {
          "key" : "greece",
          "doc_count" : 2,
          "average_of_ROI" : {
```

### 5. Disscussion and Conclusion

- Problems： Group by range



```
GET _search
{
  "aggs": {
    "range_age": {
      "terms": {
        "field": "fields.performance.Year",
        "ranges": [
          {
            "from": -1,
            "to": 0
          }
        ]
      }
    }
  }
}
```

```
{
  "error": {
    "root_cause": [
      {
        "type": "x_content_parse_exception",
        "reason": "[6:9] [terms] unknown field [ranges],
          parser not found"
      }
    ],
    "type": "x_content_parse_exception",
    "reason": "[6:9] [terms] unknown field [ranges], parser
      not found"
  },
  "status": 400
}
```

Cannot find proper position for the ranges within a single aggs. But we can do it by mapping or composite aggregations.

- Conclusion

This project, given the many difficulties encountered, really allowed us to understand the basics of Elasticsearch. We realized that an organization, clarity and good structure in the Elasticsearch is paramount so as not to fall into the mistakes and pitfalls of such language.

Finally, this project allowed us to concretely practice the Elasticsearch and the data structure, which may later in our journey help us better understand the logic of NoSQL.