23种非常有用的ElasticSearch查询例子(1)

本系列文章将展示ElasticSearch中23种非常有用的查询使用方法。由于篇幅原因，本系列文章分为六篇，
本文是此系列的第一篇文章。欢迎关注大数据技术博客微信公共账号:iteblog_hadoop。

为了展示Elasticsearch中不同查询的用法，我这里先在Elasticsearch里面创建了book相关的documents，每本书主要涉及以下字段： title, authors, summary, publish_date(发行日期),publisher以及评论条数。操作如下：

```
curl -XPUT 'https://www.iteblog.com:9200/iteblog_book_index' -d '{ "settings": { "number_of_shards": 1 }}'
```

[返回结果]

```
{"acknowledged":true}
```

```
curl -XPOST 'https://www.iteblog.com:9200/iteblog_book_index/book/_bulk' -d '
{ "index": { "_id": 1 }}
{ "title": "Elasticsearch: The Definitive Guide", "authors": ["clinton gormley", "zachary tong"], "summary" : "A distibuted real-time search and analytics engine", "publish_date" : "2015-02-07", "num_reviews": 20, "publisher": "oreilly" }
{ "index": { "_id": 2 }}
{ "title": "Taming Text: How to Find, Organize, and Manipulate It", "authors": ["grant ingersoll", "thomas morton", "drew farris"], "summary" : "organize text using approaches such as full-text search, proper name recognition, clustering, tagging, information extraction, and summarization", "publish_date" : "2013-01-24", "num_reviews": 12, "publisher": "manning" }
{ "index": { "_id": 3 }}
{ "title": "Elasticsearch in Action", "authors": ["radu gheorge", "matthew lee hinman", "roy russo"], "summary" : "build scalable search applications using Elasticsearch without having to do complex low-level programming or understand advanced data science algorithms", "publish_date" : "2015-12-03", "num_reviews": 18, "publisher": "manning" }
{ "index": { "_id": 4 }}
{ "title": "Solr in Action", "authors": ["trey grainger", "timothy potter"], "summary" : "Comprehensive guide to implementing a scalable search engine using Apache Solr", "publish_date" : "20
```

14-04-05", "num_reviews": 23, "publisher": "manning" }
'

[返回结果]


```
{
   "took": 33,
   "errors": false,
   "items": [
      {
         "index": {
            "_index": "iteblog_book_index",
            "_type": "book",
            "_id": "1",
            "_version": 1,
            "_shards": {
               "total": 2,
               "successful": 1,
               "failed": 0
            },
            "status": 201
         }
      },
      {
         "index": {
            "_index": "iteblog_book_index",
            "_type": "book",
            "_id": "2",
            "_version": 1,
            "_shards": {
               "total": 2,
               "successful": 1,
               "failed": 0
            },
            "status": 201
         }
      },
      {
         "index": {
            "_index": "iteblog_book_index",
            "_type": "book",
            "_id": "3",
            "_version": 1,
            "_shards": {
               "total": 2,
               "successful": 1,
               "failed": 0
```

```
            },
            "status": 201
        }
    },
    {
        "index": {
            "_index": "iteblog_book_index",
            "_type": "book",
            "_id": "4",
            "_version": 1,
            "_shards": {
                "total": 2,
                "successful": 1,
                "failed": 0
            },
            "status": 201
        }
    }
]
}
```

数据准备好了，现在我们可以查询ElasticSearch中的数据。

基本匹配查询(Basic Match Query)

基本匹配查询主要有两种形式：（1）、使用Search Lite API，并将所有的搜索参数都通过URL传递；（2）、使用Elasticsearch DSL，其可以通过传递一个JSON请求来获取结果。下面是在所有的字段中搜索带有"guide"的结果：

```
//////////////////////////////////////////////////////////////////////
 User: 过往记忆
 Date: 2016-08-15
 Time: 23:54
 bolg: https://www.iteblog.com
 本文地址：https://www.iteblog.com/archives/1741.html
 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
 过往记忆博客微信公共帐号：iteblog_hadoop
//////////////////////////////////////////////////////////////////////
```

```
:9200/iteblog_book_index/book/_search?q=guide
```

[返回结果]


```
{
   "took": 20,
   "timed_out": false,
   "_shards": {
      "total": 1,
      "successful": 1,
      "failed": 0
   },
   "hits": {
      "total": 2,
      "max_score": 0.24144039,
      "hits": [
         {
            "_index": "iteblog_book_index",
            "_type": "book",
            "_id": "1",
            "_score": 0.24144039,
            "_source": {
               "title": "Elasticsearch: The Definitive Guide",
               "authors": [
                  "clinton gormley",
                  "zachary tong"
               ],
               "summary": "A distibuted real-time search and analytics engine",
               "publish_date": "2015-02-07",
               "num_reviews": 20,
               "publisher": "oreilly"
            }
```

```
        },
        {
          "_index": "iteblog_book_index",
          "_type": "book",
          "_id": "4",
          "_score": 0.24144039,
          "_source": {
            "title": "Solr in Action",
            "authors": [
              "trey grainger",
              "timothy potter"
            ],
            "summary": "Comprehensive guide to implementing a scalable search engine usin
g Apache Solr",
            "publish_date": "2014-04-05",
            "num_reviews": 23,
            "publisher": "manning"
          }
        }
      ]
   }
}
```

如果我们使用Query DSL来展示出上面一样的结果可以这么来写：

```
//////////////////////////////////////////////////////////////////////
 User: 过往记忆
 Date: 2016-08-15
 Time: 23:54
 bolg: https://www.iteblog.com
 本文地址：https://www.iteblog.com/archives/1741.html
 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
 过往记忆博客微信公共帐号：iteblog_hadoop
//////////////////////////////////////////////////////////////////////
curl -XGET ':9200/iteblog_book_index/book/_search' -d '
{
   "query": {
      "multi_match" : {
         "query" : "guide",
         "fields" : ["_all"]
      }
   }
}'
```

其输出和上面使用/iteblog_book_index/book/_search?q=guide的输出一样。上面的multi_match关键字通常在查询多个fields的时候作为match关键字的简写方式。fields属性指定需要查询的字段，如果我们想查询所有的字段，这时候可以使用_all关键字，正如上面的一样。

以上两种方式都允许我们指定查询哪些字段。比如，我们想查询title中出现in Action的图书，那么我们可以这么查询：

```
//////////////////////////////////////////////////////////////////////
 User: 过往记忆
 Date: 2016-08-15
 Time: 23:54
 bolg: https://www.iteblog.com
 本文地址：https://www.iteblog.com/archives/1741.html
 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
 过往记忆博客微信公共帐号：iteblog_hadoop
//////////////////////////////////////////////////////////////////////
:9200/iteblog_book_index/book/_search?q=title:in%20action
```

[返回结果]


```
{
   "took": 27,
   "timed_out": false,
   "_shards": {
      "total": 1,
      "successful": 1,
      "failed": 0
   },
   "hits": {
      "total": 2,
      "max_score": 0.6259885,
      "hits": [
         {
            "_index": "iteblog_book_index",
            "_type": "book",
            "_id": "4",
            "_score": 0.6259885,
            "_source": {
               "title": "Solr in Action",
               "authors": [
                  "trey grainger",
                  "timothy potter"
```

```
            ],
            "summary": "Comprehensive guide to implementing a scalable search engine usin
g Apache Solr",
            "publish_date": "2014-04-05",
            "num_reviews": 23,
            "publisher": "manning"
         }
      },
      {
         "_index": "iteblog_book_index",
         "_type": "book",
         "_id": "3",
         "_score": 0.5975345,
         "_source": {
            "title": "Elasticsearch in Action",
            "authors": [
               "radu gheorge",
               "matthew lee hinman",
               "roy russo"
            ],
            "summary": "build scalable search applications using Elasticsearch without having
to do complex low-level programming or understand advanced data science algorithms",
            "publish_date": "2015-12-03",
            "num_reviews": 18,
            "publisher": "manning"
         }
      }
   ]
}
}
```


然而，DSL方式提供了更加灵活的方式来构建更加复杂的查询（我们将在后面看到），甚至指定你想要的返回结果。下面的例子中，我将指定需要返回结果的数量，开始的偏移量（这在分页的情况下非常有用），需要返回document中的哪些字段以及高亮关键字：

```
curl -XGET 'https://www.iteblog.com:9200/iteblog_book_index/book/_search' -d '
{
   "query": {
      "match" : {
         "title" : "in action"
      }
   },
   "size": 2,
```

```
    "from": 0,
    "_source": [ "title", "summary", "publish_date" ],
    "highlight": {
        "fields" : {
            "title" : {}
        }
    }
}'
```

[返回结果]

```
{
    "took": 100,
    "timed_out": false,
    "_shards": {
        "total": 1,
        "successful": 1,
        "failed": 0
    },
    "hits": {
        "total": 2,
        "max_score": 0.9105287,
        "hits": [
            {
                "_index": "iteblog_book_index",
                "_type": "book",
                "_id": "3",
                "_score": 0.9105287,
                "_source": {
                    "summary": "build scalable search applications using Elasticsearch without having
to do complex low-level programming or understand advanced data science algorithms",
                    "title": "Elasticsearch in Action",
                    "publish_date": "2015-12-03"
                },
                "highlight": {
                    "title": [
                        "Elasticsearch <em>in</em> <em>Action</em>"
                    ]
                }
            },
            {
                "_index": "iteblog_book_index",
                "_type": "book",
                "_id": "4",
                "_score": 0.9105287,
                "_source": {
```

```
            "summary": "Comprehensive guide to implementing a scalable search engine usin
g Apache Solr",
            "title": "Solr in Action",
            "publish_date": "2014-04-05"
         },
         "highlight": {
            "title": [
               "Solr <em>in</em> <em>Action</em>"
            ]
         }
      }
   ]
  }
}
```

需要注意的是
：对于查询多个关键字，match关键字允许我们使用and操作符来代替默认的or操作符。你也可以指定minimum_should_match操作符来调整返回结果的相关性(tweak
relevance)。本文就不具体介绍，更多使用情况请参见ElasticSearch官方文档。

Multi-field Search

正如我们之前所看到的，想在一个搜索中查询多个 document field（比如使用同一个查询关键字同时在title和summary中查询），你可以使用multi_match查询，使用如下：

```
curl -XGET 'https://www.iteblog.com:9200/iteblog_book_index/book/_search' -d '
{
  "query": {
    "multi_match" : {
      "query" : "elasticsearch guide",
      "fields": ["title", "summary"]
    }
  }
}'
```

[返回结果]

```
{
  "took": 13,
  "timed_out": false,
  "_shards": {
    "total": 1,
```

```
    "successful": 1,
    "failed": 0
  },
  "hits": {
    "total": 3,
    "max_score": 0.9448582,
    "hits": [
      {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "1",
        "_score": 0.9448582,
        "_source": {
          "title": "Elasticsearch: The Definitive Guide",
          "authors": [
            "clinton gormley",
            "zachary tong"
          ],
          "summary": "A distibuted real-time search and analytics engine",
          "publish_date": "2015-02-07",
          "num_reviews": 20,
          "publisher": "oreilly"
        }
      },
      {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "3",
        "_score": 0.17312013,
        "_source": {
          "title": "Elasticsearch in Action",
          "authors": [
            "radu gheorge",
            "matthew lee hinman",
            "roy russo"
          ],
          "summary": "build scalable search applications using Elasticsearch without having
to do complex low-level programming or understand advanced data science algorithms",
          "publish_date": "2015-12-03",
          "num_reviews": 18,
          "publisher": "manning"
        }
      },
      {
        "_index": "iteblog_book_index",
        "_type": "book",
```

```
            "_id": "4",
            "_score": 0.14965448,
            "_source": {
               "title": "Solr in Action",
               "authors": [
                  "trey grainger",
                  "timothy potter"
               ],
               "summary": "Comprehensive guide to implementing a scalable search engine usin
g Apache Solr",
               "publish_date": "2014-04-05",
               "num_reviews": 23,
               "publisher": "manning"
            }
         }
      ]
   }
}
```


上面的查询一共返回了三个结果。

Boosting

我们上面使用同一个搜索请求在多个field中查询，你也许想提高某个field的查询权重。在下面的例子中，我们把summary field的权重调成3，这样就提高了其在结果中的权重，这样把_id=4的文档相关性大大提高了，如下：

```
curl -XGET 'https://www.iteblog.com:9200/iteblog_book_index/book/_search' -d '
{
   "query": {
      "multi_match" : {
         "query" : "elasticsearch guide",
         "fields": ["title", "summary^3"]
      }
   },
   "_source": ["title", "summary", "publish_date"]
}'
```

[返回结果]


```
{
   "took": 8,
```

```
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "failed": 0
  },
  "hits": {
    "total": 3,
    "max_score": 0.31495273,
    "hits": [
      {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "1",
        "_score": 0.31495273,
        "_source": {
          "summary": "A distibuted real-time search and analytics engine",
          "title": "Elasticsearch: The Definitive Guide",
          "publish_date": "2015-02-07"
        }
      },
      {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "4",
        "_score": 0.14965448,
        "_source": {
          "summary": "Comprehensive guide to implementing a scalable search engine usin
g Apache Solr",
          "title": "Solr in Action",
          "publish_date": "2014-04-05"
        }
      },
      {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "3",
        "_score": 0.13094766,
        "_source": {
          "summary": "build scalable search applications using Elasticsearch without having
to do complex low-level programming or understand advanced data science algorithms",
          "title": "Elasticsearch in Action",
          "publish_date": "2015-12-03"
        }
      }
    ]
```

```
    }
}
```

大家可以对比一下这个查询结果和上面结果的不同。

需要注意的是：Boosting不仅仅意味着计算出来的分数(calculated score)直接乘以boost factor，最终的boost value会经过归一化以及其他一些内部的优化，可以参考官方文档了解更多详情。

Bool Query

我们可以在查询条件中使用AND/OR/NOT操作符，这就是布尔查询(Bool Query)。布尔查询可以接受一个must参数(等价于AND)，一个must_not参数(等价于NOT)，以及一个should参数(等价于OR)。比如，我想查询title中出现Elasticsearch或者Solr关键字的图书，图书的作者是clinton gormley，但没有radu gheorge，我们可以这么来查询：

```
curl -XGET 'https://www.iteblog.com:9200/iteblog_book_index/book/_search' -d '
{
    "query": {
        "bool": {
            "must": {
                "bool" : { "should": [
                    { "match": { "title": "Elasticsearch" }},
                    { "match": { "title": "Solr" }} ] }
            },
            "must": { "match": { "authors": "clinton gormely" }},
            "must_not": { "match": {"authors": "radu gheorge" }}
        }
    }
}'
```

[返回结果]

```
{
    "took": 26,
    "timed_out": false,
    "_shards": {
        "total": 1,
        "successful": 1,
        "failed": 0
    },
    "hits": {
```

```
    "total": 1,
    "max_score": 0.31271058,
    "hits": [
      {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "1",
        "_score": 0.31271058,
        "_source": {
          "title": "Elasticsearch: The Definitive Guide",
          "authors": [
            "clinton gormley",
            "zachary tong"
          ],
          "summary": "A distibuted real-time search and analytics engine",
          "publish_date": "2015-02-07",
          "num_reviews": 20,
          "publisher": "oreilly"
        }
      }
    ]
  }
}
```


限于篇幅的原因，本系列文章分为六部分，欢迎关注过往记忆大数据技术博客及时了解大数据相关文章，微信公共账号：iteblog_hadoop。