23种非常有用的ElasticSearch查询例子(2)

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，
本文是此系列的第二篇文章。欢迎关注大数据技术博客微信公共账号:iteblog_hadoop。

Fuzzy Queries（模糊查询）

模糊查询可以在Match和 Multi-Match查询中使用以便解决拼写的错误，模糊度是基于Levenshtein distance计算与原单词的距离。使用如下：

```
curl -XGET 'https://www.iteblog.com:9200/iteblog_book_index/book/_search' -d '
{
   "query": {
      "multi_match" : {
         "query" : "comprihensiv guide",
         "fields": ["title", "summary"],
         "fuzziness": "AUTO"
      }
   },
   "_source": ["title", "summary", "publish_date"],
   "size": 1
}'
```

[返回结果]

```
{
   "took": 208,
   "timed_out": false,
   "_shards": {
      "total": 1,
      "successful": 1,
      "failed": 0
   },
```

```
  "hits": {
    "total": 2,
    "max_score": 0.5961596,
    "hits": [
      {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "4",
        "_score": 0.5961596,
        "_source": {
          "summary": "Comprehensive guide to implementing a scalable search engine usin
g Apache Solr",
          "title": "Solr in Action",
          "publish_date": "2014-04-05"
        }
      }
    ]
  }
}
```


需要注意：上面我们将fuzziness的值指定为AUTO，其在term的长度大于5的时候相当于指定值为2。然而80%的人拼写错误的编辑距离(edit distance)为1，所有如果你将fuzziness设置为1可能会提高你的搜索性能。具体的可以参考Elasticsearch权威指南相关章节。

Wildcard Query(通配符查询)

通配符查询允许我们指定一个模式来匹配，而不需要指定完整的trem。?将会匹配如何字符；*将会匹配零个或者多个字符。比如我们想查找所有作者名字中以t字符开始的记录，我们可以如下使用：


```
curl -XGET 'https://www.iteblog.com:9200/iteblog_book_index/book/_search' -d '
{
   "query": {
      "wildcard" : {
         "authors" : "t*"
      }
   },
   "_source": ["title", "authors"],
   "highlight": {
      "fields" : {
         "authors" : {}
      }
   }
}'
```

[返回结果]

```
{
   "took": 37,
   "timed_out": false,
   "_shards": {
      "total": 1,
      "successful": 1,
      "failed": 0
   },
   "hits": {
      "total": 3,
      "max_score": 1,
```

```
"hits": [
    {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "1",
        "_score": 1,
        "_source": {
            "authors": [
                "clinton gormley",
                "zachary tong"
            ],
            "title": "Elasticsearch: The Definitive Guide"
        },
        "highlight": {
            "authors": [
                "zachary <em>tong</em>"
            ]
        }
    },
    {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "2",
        "_score": 1,
        "_source": {
            "authors": [
                "grant ingersoll",
                "thomas morton",
                "drew farris"
            ],
            "title": "Taming Text: How to Find, Organize, and Manipulate It"
        },
        "highlight": {
            "authors": [
                "<em>thomas</em> morton"
            ]
        }
    },
    {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "4",
        "_score": 1,
        "_source": {
            "authors": [
                "trey grainger",
```

```
                "timothy potter"
            ],
            "title": "Solr in Action"
         },
         "highlight": {
            "authors": [
               "<em>trey</em> grainger",
               "<em>timothy</em> potter"
            ]
         }
      }
   ]
  }
}
```

Regexp Query(正则表达式查询)

ElasticSearch还支持正则表达式查询，此方式提供了比通配符查询更加复杂的模式。比如我们先查找作者名字以t字符开头，中间是若干个a-z之间的字符，并且以字符y结束的记录，可以如下查询：

```
curl -XGET 'https://www.iteblog.com:9200/iteblog_book_index/book/_search' -d '
{
   "query": {
      "regexp" : {
         "authors" : "t[a-z]*y"
      }
   },
   "_source": ["title", "authors"],
   "highlight": {
      "fields" : {
         "authors" : {}
      }
   }
}'
```

```
{
   "took": 25,
   "timed_out": false,
   "_shards": {
      "total": 1,
      "successful": 1,
      "failed": 0
```

```
  },
  "hits": {
    "total": 1,
    "max_score": 1,
    "hits": [
      {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "4",
        "_score": 1,
        "_source": {
          "authors": [
            "trey grainger",
            "timothy potter"
          ],
          "title": "Solr in Action"
        },
        "highlight": {
          "authors": [
            "<em>trey</em> grainger",
            "<em>timothy</em> potter"
          ]
        }
      }
    ]
  }
}
```


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。