



链滴

# elasticsearch-sql 示例

作者: [flowaters](#)

原文链接: <https://ld246.com/article/1502334319736>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 背景

[elasticsearch-sql](#)项目是一个非常活跃的ES SQL插件项目，能够紧跟ES官方的最新版本。

目前ES官方稳定版本为5.5.1, 开发版本为6.x; elasticsearch-sql支持从1.7.6到5.5.1的所有ES稳定版。

## SQL语法

### Query

```
SELECT * FROM bank WHERE age >30 AND gender = 'm'
```

### Aggregation

```
select COUNT(*),SUM(age),MIN(age) as m, MAX(age),AVG(age) FROM bank GROUP BY gender  
ORDER BY SUM(age), m DESC
```

### Delete

```
DELETE FROM bank WHERE age >30 AND gender = 'm'
```

### Search

```
SELECT address FROM bank WHERE address = matchQuery('880 Holmes Lane') ORDER BY score  
DESC LIMIT 3
```

## Aggregations

range age group 20-25,25-30,30-35,35-40

```
SELECT COUNT(age) FROM bank GROUP BY range(age, 20,25,30,35,40)
```

range date group by day

```
SELECT online FROM online GROUP BY date_histogram(field='insert_time','interval'='1d')
```

range date group by your config

```
SELECT online FROM online GROUP BY date_range(field='insert_time','format'='yyyy-MM-dd'  
'2014-08-18','2014-08-17','now-8d','now-7d','now-6d','now')
```

ES Geographic

```
SELECT * FROM locations WHERE GEO_BOUNDING_BOX(fieldname,100.0,1.0,101,0.0)
```

Select type

```
SELECT * FROM indexName/type
```

# SQL Features

- SQL Select
- SQL Delete
- SQL Where
- SQL Order By
- SQL Group By
- SQL AND & OR
- SQL Like
- SQL COUNT distinct
- SQL In
- SQL Between
- SQL Aliases
- SQL Not Null
- SQL(ES) Date
- SQL avg()
- SQL count()
- SQL last()
- SQL max()
- SQL min()
- SQL sum()
- SQL Nulls
- SQL isnull()
- SQL now()
- SQL floor
- SQL split
- SQL trim
- SQL log
- SQL log10
- SQL substring
- SQL round
- SQL sqrt
- SQL concat\_ws
- SQL union and minus

## Beyond sql features

- ES TopHits

- ES MISSING
- ES STATS
- ES GEO\_INTERSECTS
- ES GEO\_BOUNDING\_BOX
- ES GEO\_DISTANCE
- ES GEOHASH\_GRID aggregation

## 运行方式

[官方推荐两种运行方式](#)

1. 插件方式：通过rest api调用

```
http://localhost:9200/_sql?sql=select * from indexName limit 10
```

2. WebUI方式：通过Web UI调用

```
http://localhost:9200/_plugin/sql
```

## 安装

见[官方安装文档](#)

## 测试

```
mvn test
```

## 开发

1. 下载github代码
2. 本地启动一个ES 5.5
3. 跑单元测试

## 算子示例

### 聚合算子

#### COUNT

```
SELECT COUNT(*) FROM testindex/account
```

#### SUM

```
SELECT SUM(balance) FROM testindex/account
```

## MIN

```
SELECT MIN(age) FROM testindex/account
```

## MAX

```
SELECT MAX(age) FROM testindex/account
```

## AVG

```
SELECT AVG(age) FROM testindex/account
```

## STATS

```
SELECT STATS(age) FROM testindex/account
```

## EXTENDED\_STATS

```
SELECT EXTENDED_STATS(age) FROM testindex/account
```

## PERCENTILES

```
SELECT PERCENTILES(age) FROM testindex/account
```

```
SELECT PERCENTILES(age,25.0,75.0) x FROM testindex/account
```

## GROUP BY

```
SELECT COUNT(*) FROM testindex/account GROUP BY gender
```

```
SELECT COUNT(*) FROM testindex/account GROUP BY gender, terms('field'='age','size'=200,'alias'='age')
```

```
SELECT COUNT(*) FROM testindex/account GROUP BY gender, terms('alias'='ageAgg','field'='age','size'=3)
```

- TERMS WITH SIZE

```
SELECT COUNT(*) FROM testindex/account GROUP BY terms('alias'='ageAgg','field'='age','size'=3)
```

- TERMS WITH MISSING

```
SELECT count(*) FROM testindex/gotCharacters GROUP BY terms('alias'='nick','field'='nickname','missing'='no_nickname')
```

- TERMS WITH ORDER

```
SELECT count(*) FROM testindex/dog GROUP BY terms('field'='dog_name', 'alias'='dog_name' order='desc')
```

```
SELECT count(*) FROM testindex/dog GROUP BY terms('field'='dog_name', 'alias'='dog_name' order='asc')
```

- ORDER BY ASC

```
SELECT COUNT(*) FROM testindex/account GROUP BY age ORDER BY COUNT(*)
```

- ORDER BY DESC

```
SELECT COUNT(*) FROM testindex/account GROUP BY age ORDER BY COUNT(*) DESC
```

- LIMIT

```
SELECT COUNT(*) FROM testindex/account GROUP BY age ORDER BY COUNT(*) LIMIT 5
```

- COUNT GROUP BY

```
SELECT COUNT(age) FROM testindex/account GROUP BY range(age, 20,25,30,35,40)
```

- COUNT GROUP BY DATE

```
select insert_time from online group by date_histogram(field='insert_time','interval'='1.5h','format'='yyyy-MM','min_doc_count'=5)
```

- COUNT GROUP BY DATE WITH ALIAS

```
select insert_time from online group by date_histogram(field='insert_time','interval'='1.5h','format'='yyyy-MM','alias'='myAlias')
```

- COUNT DATE RANGE

```
select online from online group by date_range(field='insert_time','format'='yyyy-MM-dd', '2014-08-18','2014-08-17','now-8d','now-7d','now-6d','now')
```

## TOP HITS

- top hit

```
select topHits('size'=3,age='desc') from testindex/accounts group by gender
```

- top hit with include

```
select topHits('size'=3,age='desc',include=age) from testindex/account group by gender
```

- top hit with include two fields

```
select topHits('size'=3,'include'='age,firstname',age='desc') from testindex/account group by gender
```

- top hit with exclude

```
select topHits('size'=3,'exclude'='lastname',age='desc') from testindex/account group by gender
```

er

- top hit with include and exclude

```
select topHits('size'=3,'exclude'='lastname','include'='firstname,lastname',age='desc') from te
tindex/account group by gender
```

未完待续

## 从SQL到ES Query

上面啰嗦了一大堆，或许你根本对于JDBC不感兴趣，也对直接传入SQL返回结果不感兴趣，而仅仅对SQL解析为ES Query部分感兴趣。

这样，可以从SQL到ES Query，利用熟悉的SQL来探索复杂的ES语法。

## 示例

Talk is cheap, Show me the code.

pom.xml

```
<dependencies>
  <dependency>
    <groupId>org.elasticsearch</groupId>
    <artifactId>elasticsearch</artifactId>
    <version>5.5.1</version>
  </dependency>
  <dependency>
    <groupId>org.elasticsearch.plugin</groupId>
    <artifactId>parent-join-client</artifactId>
    <version>5.5.1</version>
  </dependency>
  <dependency>
    <groupId>org.elasticsearch.client</groupId>
    <artifactId>transport</artifactId>
    <version>5.5.1</version>
  </dependency>
</dependencies>
```

下载zip包放到工程的目录下

<https://github.com/NLPchina/elasticsearch-sql/releases/download/5.5.1.0/elasticsearch-sql-5.1.0.zip>

文件

```
import java.sql.SQLFeatureNotSupportedException;

import org.elasticsearch.client.transport.TransportClient;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.transport.client.PreBuiltTransportClient;
import org.nlpcn.es4sql.SearchDao;
```

```

import org.nlpcn.es4sql.exception.SqlParseException;
import org.nlpcn.es4sql.query.SqlElasticSearchRequestBuilder;

public class MainElasticsearchSql {
    public static void main(String[] args) throws SQLFeatureNotSupportedException, SqlParseException {
        Settings settings = Settings.builder().put("client.transport.ignore_cluster_name", true).build();

        TransportClient client = new PreBuiltTransportClient(settings);
        SearchDao searchDao = new SearchDao(client);

        String query = String.format("SELECT COUNT(*) as count FROM index/type");

        SqlElasticSearchRequestBuilder select = (SqlElasticSearchRequestBuilder)searchDao.explain(query).explain();
        System.out.println(select);
    }
}

```

### 执行结果

```

{
  "from" : 0,
  "size" : 0,
  "_source" : {
    "includes" : [
      "COUNT"
    ],
    "excludes" : [ ]
  },
  "aggregations" : {
    "count" : {
      "value_count" : {
        "field" : "_index"
      }
    }
  }
}

```