Transwarp Data Hub v4.5

ESdrive SQL使用手册

修订历史记录
修改记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。
文档版本T00145-01-010 (2016-03) 第一次发布。

2016-3-17

目录

1. ESdrive SQL简介

ESdrive实现了Inceptor over ElasticSearch的逻辑，让用户可以用SQL通过Inceptor操作ElasticSearch。目前ElasticSearch支持的数据类型如下：BOOLEAN，TINYINT，SMALLINT，INTEGER，BIGINT，FLOAT，DOUBLE，STRING。

ElasticSearch中的对象和Inceptor中的对象

在ESdrive SQL中，ElasticSearch的对象和Inceptor的对象之间逻辑上的对应关系为：

ElasticSearch	Inceptor
Index	表(Table)
Document	行(Row)
Field	列(Column)

在ESdrive SQL中，为了表达方便，我们会统一使用Inceptor的对象名称，也就是Table，Row和Column。

2. ESdrive SQL DDL

ESdrive SQL中的DDL包括创建（CREATE）/编辑（ALTER）/删除（DROP）/清空（TRUNCATE）表。

2.1. 建表：CREATE

简化建ES表语法

```
CREATE [EXTERNAL] TABLE <table> (
    <id> STRING,  ❶
    <column> <data_type>,
    <column> <data_type>,
    ...
)
STORED AS ES  ❷
[WITH SHARD NUMBER <m>] ❸
[REPLICATION <n>] ❹
[TBLPROPERTIES('elasticsearch.tablename'='<esdrive_table>')]; ❺
```

❶ ES表的第一列为ES表的id，必须是STRING类型的。

❷ 指定表的格式为 ES。

❸ 可选项，指定ES表的分片数m。如果不指定，使用默认值5。一旦建表即不能更改。这里需要用户预估计索引数据的量，**一个SHARD上的数据不要超过25GB**。超过25GB可能会有比较严重的性能问题。

❹ 可选项，指定每个分片的副本数n，如果不指定，使用默认值1。建表后还可以使用 ALTER 来更改。

❺ 可选项，指定新建的表在ElasticSearch中的表名。建外表时必须指定新建的表在ElasticSearch中对应的表名。创建内表时可以指定，也可以忽略，当忽略时，默认的ElasticSearch表名是elasticsearch_<dbName：tableName>。

普通建ES表语法

```
CREATE [EXTERNAL] TABLE <table> (
    <id> STRING,
    <column> <data_type>,
    <column> <data_type>,
    ...
)
STORED BY 'io.transwarp.esdrive.ElasticSearchStorageHandler'  ❶
[WITH SERDEPROPERTIES('elasticsearch.columns.mapping'='_id,<cl1>,<cl2>, ...')]  ❷
[WITH SHARD NUMBER <m>]
[REPLICATION <n>]
[TBLPROPERTIES('elasticsearch.tablename'='<esdrive_table>')];
```

❶ 指定使用 `'io.transwarp.esdrive.ElasticSearchStorageHandler'` 作为storage handler。

❷ 可选项，指定新建的表和ElasticSearch中的对应表的列的映射关系。如果使用这个选项指定列的映射关系，等号右边的 `_id` 为固定用法，不能改为其他名字。

建表语法的选择

显然，简化建ES表语法步骤简单，易于记忆，所以我们 **鼓励用户在建内表时使用简化建ES表语法**。如果新建的表和ElasticSearch中已经有的表重名，需要使用 `TBLPROPERTIES('elasticsearch.tablename'='<esdrive_table>')` 来指定新建表在ElasticSearch中的表名。建外表时，ES表必须映射到ElasticSearch中一张已经存在的源表。如果ES表和它的源表在ElasticSearch中的对应列列名相同，可以使用简化建ES表语法建表；如果ES表和它的源表在ElasticSearch中的对应列列名不同，需要使用普通建ES表语法来建表，通过建表语句中的 `WITH SERDEPROPERTIES('elasticsearch.columns.mapping'='_id,<cl1>,<cl2>,…')` 来指定ES表和ElasticSearch中源表之间的列名映射。**注意**，ElasticSearch中表的列名 **区分大小写**。

例 1. 建ES内表

```
CREATE TABLE employee (
    eid STRING,
    name STRING,
    age TINYINT,
    height DOUBLE,
    about STRING
)
STORED AS ES;
```

例 2. 建ES外表，ES表列名和源表列名相同

```
CREATE EXTERNAL  TABLE esdrive_test(
```

```
  key STRING,
  content STRING,
  tint INT,
  tfloat FLOAT,
  tbool BOOLEAN)
STORED AS ES
TBLPROPERTIES('elasticsearch.tablename'='esdrive_test');
```

例 3. 建ES外表，ES表列名和源表列名不同

```
CREATE EXTERNAL TABLE esdrive_test3(
  key3 STRING,
  content3 STRING,
  tint3 INT,
  tfloat3 FLOAT,
  tbool3 BOOLEAN)
STORED BY 'io.transwarp.esdrive.ElasticSearchStorageHandler'
WITH SERDEPROPERTIES('elasticsearch.columns.mapping'='_id, content, tint,
 tfloat, tbool')
TBLPROPERTIES('elasticsearch.tablename'='esdrive_test');
```

建表后可以用 DESCRIBE 查看表的元数据：

```
DESCRIBE FORMATTED employee;
```

2.2. 修改表：ALTER

使用 ALTER 可以修改ES表的TBLPROPERTIES。**目前只能用于修改ES表settings中可动态改变的配置项**，例如replication。关于ES表settings的细节请参考《ElasticSearch使用手册》。

修改ES表TBLPROPERTIES的语法

```
ALTER TABLE <table> SET TBLPROPERTIES
 ('<property_name>'='<property_value>', '<property_name>'='<property_value>', ...);
```

其中， property_name 为属性名，property_value 为属性值，它们都需要放在引号中间。该语法可以用于修改ES表的副本数。

修改ES表副本数的语法

```
ALTER TABLE <table> SET TBLPROPERTIES ('number_of_replicas'='<n>');
```

例 4. 修改ES表副本数

```
ALTER TABLE employee SET TBLPROPERTIES ('number_of_replicas'='8');
```

TBLPROPERTIES 可以使用 DESCRIBE FORMATTED 来查看：

```
DESCRIBE FORMATTED employee;
```

2.3. 删除表：DROP

删除ES表的语法

```
DROP TABLE <table>;
```

2.4. 清空表：TRUNCATE

清空ES表的语法

```
TRUNCATE TABLE <table>;
```

注意，不能 TRUNCATE 外表。

3. ESdrive SQL DML

ESdrive SQL中的DML（Data Manipulation Language）包含插入数据（INSERT），和删除表中记录（DELETE）。ESdrive SQL目前不支持 UPDATE。

3.1. 插入：INSERT

ESdrive SQL支持向ES表中单条插入数据或者批量插入查询结果。

单条插入的语法

```
INSERT INTO TABLE <table> [(<column1>, <column2>, ...)] VALUES (<value1>,
 <value2>, ...);
```

例 5. 单条插入ES表

向例 2 “建ES外表，ES表列名和源表列名相同”中创建的表esdrive_test插入数据：

```
INSERT INTO TABLE esdrive_test(key, content, tint, tfloat, tbool) VALUES ("1",
 "mysql is database", 1 ,1.1, true);
INSERT INTO TABLE esdrive_test(key, content, tint, tfloat, tbool) VALUES ("2",
 "oracle is database",2,2.2, false);
INSERT INTO TABLE esdrive_test(key, content, tint, tfloat, tbool) VALUES ("3",
 "db2 is datatbase",3,3.3, true);
INSERT INTO TABLE esdrive_test(key, content, tint, tfloat, tbool) VALUES ("4",
 "oracle and mysql are databases",4,4.4, false);
INSERT INTO TABLE esdrive_test(key, content, tint, tfloat, tbool) VALUES ("5",
 "contains test !!!", 5,5.5, false);
INSERT INTO TABLE esdrive_test(key, content, tint, tfloat, tbool) VALUES ("6",
 "test", 6,6.6, true);
INSERT INTO TABLE esdrive_test(key, content, tint, tfloat, tbool) VALUES ("7",
 "first second third",7,7.7, false);
```

例 6. 向ES表批量插入查询记录

向例 1 “建ES内表”中创建的表employee插入数据：

```
INSERT INTO TABLE employee SELECT * FROM employee_2;
```

3.2. 删除表中记录：DELETE

删除ES表中记录的语法

```
DELETE FROM <table> WHERE <filter_conditions>;
```

例 7. 删除ES表中记录

```
DELETE FROM employee WHERE name = 'Alice';
```

4. ESdrive SQL DQL

ESdrive SQL支持所有Hyperdrive SQL中的DQL（除 USE_INDEX 以外），包括 WHERE、GROUP BY、 JOIN、子查询和集合运算等，具体细节请参考《Hyperdrive SQL使用手册》。本章我们将详细介绍ESdrive SQL中特有的DQL：使用 CONTAINS 和 MATCHES 这两个UDF做查询。

ElasticSearch中的分词

下面即将介绍的 CONTAINS 和 MATCHES 函数的必须ES表的 **分词列** 使用。目前，ESdrive SQL还没有分词语法，对表的分词必须通过ElasticSearch Query DSL在建表时设置，例如：

建esdrive_test表的ElasticSearch Query DSL Query Body

```
{
  "settings": {
     "refresh_interval": "5s",
     "number_of_shards" :   4,
     "number_of_replicas" : 1
  },
  "mappings": {
    "_default_":{
      "_all": { "enabled":  true }
    },
    "resource": {
      "dynamic": false,
      "properties": {
```

```
        "key": {
          "type": "string",
          "index": "analyzed"  ❶
        },
        "content": {
          "type": "string",
          "analyzer": "english"  ❷
        },
        "tint": {
          "type": "integer"
        },
         "tfloat": {
          "type": "float"
        },
        "tbool": {
          "type": "boolean"
        }
      }
    }
  }
}
```

❶ 表示对该列分词，使用ElasticSearch默认的分词插件english。

❷ 显式指定对该列用分词插件english分词。ElasticSearch对不同语言提供不同的分词插件，分词插件的安装和配置请参考《ElasticSearch使用手册》。

在ElasticSearch中建表完成以后，用户可以通过ESdrive SQL建外表映射到它，然后进行查询。目前在ESdrive SQL中还不支持对内表使用 `CONTAINS` 和 `MATCHES` 函数（不会报错，但是不会得到预期结果）。

其他ElasticSearch Query DSL的使用请参考《ElasticSearch使用手册》。

4.1. CONTAINS

`CONTAINS` 函数用于搜索字符列的单个词和短语。目前可支持：

- 精确匹配词或短语。
- 模糊匹配词或短语。
- 精确匹配附近的（`NEAR` 语法）词的短语。

`CONTAINS` 函数使用语法

```
CONTAINS([<table>.]<column>, '<text_query>')
```

`CONTAINS` 操作符在 `<column>` 中匹配 `<text_query>`，如果匹配成功，返回TRUE，不成功则返回FALSE。第二个参数 `<text_query>` 必须是字符类型，并且要放在引号之间。

表 1. `CONTAINS` 中支持的运算符

ESdrive中的表示形式	描述
AND, &	逻辑操作符“与”
OR, |	逻辑操作符“或”

ESdrive中的表示形式	描述
()	用于提高表达式的优先级
NEAR	间隔词匹配函数“near”（具体见下）

NEAR 语法

```
NEAR((<term1>, <term2>, ...), <slop>, [<in_order>])
```

NEAR 函数用于检索附近的一组词（词之间有一定间隔）。需要至少两个参数。第一个参数 `(<term1>, <term2>, …)` 是一组匹配项，可以有一个或多个。第二个参数 `<slop>` 是匹配项之间最多能间隔的词数，该参数必须是一个正整数。第三个参数 `<in_order>` 是可选项，用于指定是否按第一个参数中匹配项出现的顺序匹配，该参数必须是TRUE或FALSE，默认值是FALSE。

下面，我们对例 2 “建ES外表，ES表列名和源表列名相同”中的esdrive_test表进行一系列查询。

例 8. 单个词匹配

```
SELECT * FROM esdrive_test WHERE CONTAINS(content, 'oracle');

2 oracle is database 2 2.2 false
4 oracle and mysql are databases 4 4.4 false
```

例 9. 在 CONTAINS 中使用“与”

```
SELECT * FROM esdrive_test WHERE CONTAINS(content, 'oracle AND mysql AND db2');
SELECT * FROM esdrive_test WHERE CONTAINS(content, 'oracle & mysql & db2');
```

这两个查询都没有输出，表示没有匹配成功的记录。

例 10. 在 CONTAINS 中使用“或”

```
SELECT * FROM esdrive_test WHERE CONTAINS(content, 'oracle OR mysql OR db2');
SELECT * FROM esdrive_test WHERE CONTAINS(content, 'oracle | mysql | db2');

1 mysql is database 1 1.1 true
2 oracle is database 2 2.2 false
4 oracle and mysql are databases 4 4.4 false
3 db2 is datatbase 3 3.3 true
```

例 11. 在查询中使用多个运算符

```
SELECT * FROM esdrive_test WHERE CONTAINS(content, '(oracle & mysql) | db2');
```

```
4 oracle and mysql are databases 4 4.4 false
3 db2 is datatbase 3 3.3 true
```

例 12. 在查询中使用“near”

不指定按匹配项出现顺序匹配

```
SELECT * FROM esdrive_test WHERE CONTAINS(content, 'NEAR((database, oracle),
 2)');

2 oracle is database 2 2.2 false
```

指定按匹配项出现顺序匹配

```
SELECT * FROM esdrive_test WHERE CONTAINS(content, 'NEAR((database, oracle), 2,
 TRUE)');
```

没有输出

表 2. CONTAINS 中支持的模糊查询符号

ESdrive中的表示形式	描述
%	匹配任意个
_	匹配1个

例 13. 模糊查询符 %

```
SELECT * FROM esdrive_test WHERE CONTAINS(content, 'test%');

6 test 6 6.6 true
5 contains test !!! 5 5.5 false
```

例 14. 模糊查询符 _

```
SELECT * FROM esdrive_test WHERE CONTAINS(content, 'tes_');

6 test 6 6.6 true
5 contains test !!! 5 5.5 false
```

4.2. MATCHES

MATCHES 函数使用语法

```
MATCHES([<table>.]<column>, '<text_query>')
```

MATCHES 进行 **不区分大小写，匹配固定位置的字符和空格** 的精确匹配。MATCHES 在 <column> 中匹配 <text_query>，如果匹配成功，返回TRUE，不成功则返回FALSE。第二个参数 <text_query> 必须是字符类型，并且要放在引号之间。

例 15. 在查询中使用 MATCHES

```
SELECT * FROM esdrive_test WHERE MATCHES(content, 'mysql are databases');

4 oracle and mysql are databases 4 4.4 false
```

因为 MATCHES 不区分大小写，所以下面查询返回和上面相同的结果：

```
SELECT * FROM esdrive_test WHERE MATCHES(content, 'mysql ARe DaTabases');

4 oracle and mysql are databases 4 4.4 false
```

因为 MATCHES 匹配固定位置的字符和空格，词序会影响匹配结果。例如下面的查询没有输出：

```
SELECT * FROM esdrive_test WHERE MATCHES(content, 'databases are mysql');
```