**ESQL语法指南**

**V3.0**

**公开**

**杭州合众数据技术有限公司**

**2016年5月**

|  |  |  |
| --- | --- | --- |
| 修改日期 | 修改版本 | 修改内容 |
| **2016/3/10** | Esql3.0 | 支持 like 、 通配符？ \* 等模糊查询 |
| In（xx,xx,xx）in默认支持1024个条件 |
| 增加后台日志 |
| 页面不再提示504错误 |
| **2016/5/5** | Esql3.0 | 增加权限控制功能，可以为用户授权到表级别。可以控制用户访问ip. |
| 增加语法功能：show tables;create table;create index;create user;  desc tables 语法也在权限控制范围内，没有权限的表不能执行 |
| 增加语法树可视化，便于语法分析。 |
| **2016/5/12** | Esql3.0 | 增加语法功能：bulk into;Insert into;Flush;Cat;  权限控制：root用户可修改;用户修改自己密码；Root用户可设置关闭运行远程访问 |
| **2016/6/2** | Esql3.0.5 | 增加红名单功能，增加root查询sys\_meta功能 |
| **2016/6/8** | Esql3.0.5 | 增加更新语法：Update,upsert |
| **2016/6/8** | Esql3.0.5 | 实现 create table 时, index 中不同 doc\_type 的同名字段的类型一致性检查 |

目录

[1. 引言 5](#_Toc452131537)

[2. 术语介绍 5](#_Toc452131538)

[2.1表或索引(\_index) 5](#_Toc452131539)

[2.2文档类型(\_type) 6](#_Toc452131540)

[2.3文档记录(Document) 6](#_Toc452131541)

[2.4分片索引(Shard) 7](#_Toc452131542)

[2.5索引副本(Replica) 7](#_Toc452131543)

[3. 数据建模 7](#_Toc452131544)

[3.1创建索引 7](#_Toc452131545)

[3.2字段类型 8](#_Toc452131546)

[3.3查看表结构 9](#_Toc452131547)

[4. 数据操作 10](#_Toc452131548)

[4.1插入数据 10](#_Toc452131549)

[4.2批量操作 10](#_Toc452131550)

[5. 搜索查询 11](#_Toc452131551)

[5.1基本语法 11](#_Toc452131552)

[5.2搜索语法 11](#_Toc452131553)

[5.2.1跨表搜索 11](#_Toc452131554)

[5.2.2整库搜索 12](#_Toc452131555)

[5.2.3全文搜索 13](#_Toc452131556)

[5.2.4分页(limit) 14](#_Toc452131557)

[5.2.5高亮(highlight) 15](#_Toc452131558)

[5.2.6排序(order by) 16](#_Toc452131559)

[5.2.7排名(score by) 19](#_Toc452131560)

[5.3查询条件 22](#_Toc452131561)

[5.3.1嵌套表查询(insideq) 22](#_Toc452131562)

[5.3.2 in 条件查询 22](#_Toc452131563)

[5.3.3 Between to 22](#_Toc452131564)

[5.3.4 !=,>=,>,<=,< 23](#_Toc452131565)

[5.3.5 is null和is not null 23](#_Toc452131566)

[5.3.6 Id查询 23](#_Toc452131567)

[5.4 高级搜索 24](#_Toc452131568)

[5.4.1 DQ语法 24](#_Toc452131569)

[5.4.2模糊查询 24](#_Toc452131570)

[5.4.3 多个字段中搜索 24](#_Toc452131571)

[6. 碰撞比对 25](#_Toc452131572)

[6.1快速比对（fastjoin） 26](#_Toc452131573)

[6.2逻辑关系 26](#_Toc452131574)

[7. 符号使用 28](#_Toc452131575)

[7.1 占位符号使用 28](#_Toc452131576)

[7.2 引号使用 29](#_Toc452131577)

[8. 权限控制 29](#_Toc452131578)

[8.1. 创建用户 29](#_Toc452131579)

[8.2. 赋予用户权限 30](#_Toc452131580)

[8.3. 查看用户权限 30](#_Toc452131581)

[8.4. 查看用户非权限内的表 30](#_Toc452131582)

[8.5. 收回用户授权 30](#_Toc452131583)

[8.6. 删除用户 30](#_Toc452131584)

[8.7. 修改用户密码 31](#_Toc452131585)

[8.8. 关闭root用户远程登录功能 31](#_Toc452131586)

[9. 监控管理 32](#_Toc452131587)

[9.1查看所有表 32](#_Toc452131588)

[9.2查看版本信息 33](#_Toc452131589)

[9.3查看分片情况 33](#_Toc452131590)

[9.4查看主节点 33](#_Toc452131591)

[9.5查看集群节点 33](#_Toc452131592)

[9.6查看索引状态 34](#_Toc452131593)

[9.7查看分段信息 34](#_Toc452131594)

[9.8查看索引统计 35](#_Toc452131595)

[9.9查看恢复情况 35](#_Toc452131596)

[9.10查看集群状态 35](#_Toc452131597)

[9.11查看执行中的任务 35](#_Toc452131598)

[9.12 查看别名 35](#_Toc452131599)

[9.13显示线程情况 35](#_Toc452131600)

[9.14查看插件 36](#_Toc452131601)

[9.15查看cache情况 36](#_Toc452131602)

[9.16查看磁盘空间 36](#_Toc452131603)

[9.17刷新 37](#_Toc452131604)

[10. 可视化语法树 37](#_Toc452131605)

# 引言

《ESQL使用指南》，是针对合众全文数据库UDB来进行ESQL开发的指导手册。通过该手册，您可以全面的了解合众分步式数据库的功能和开发接口，为您开发相关的搜索应用提供关键支持。

ESQL能做什么？

ESQL面向数据库执行查询

ESQL可从数据库取回数据

ESQL可在数据库中插入新的记录

ESQL可更新数据库中的数据

ESQL可从数据库删除记录

ESQL可在数据库中创建新表

# 术语介绍

## 2.1表或索引(\_index)

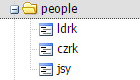
一个数据库通常包含一张或多张表（索引）。每张表有一个名字标识（例如“人员”或者“轨迹”）。表包含带有数据的记录（行）。

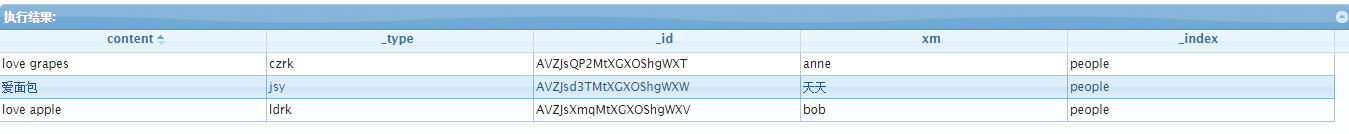
UDB把数据存放到一张或者多张表（索引）中。如果用关系型数据库模型对比，表（索引）的地位与数据库表（table)相当。索引存放和读取的基本单元是文档(Document)，由字段（fields）组成。

在分布式架构中，表（索引）是由1个或多个分片索引（Shard）组成的（见2.4）。

## 2.2文档类型(\_type)

文档类型（\_type）是对表中记录的一种归类。默认情况下\_type=base。用户可以按需求自己指定数据的文档类型，进行分类存放。例如：在people表下可以设置3个\_type：czrk常住人口、ldrk流动人口、jsy驾驶员。将这3类数据集中存放在people表中，以\_type来进行区分。





## 2.3 文档记录(Document)

在UDB中，文档(Document)是主要的存在实体，一条记录就是一个文档(Document)。所有的UDB应用需求到最后都可以统一建模成一个检索模型：检索相关文档。文档(Document)由一个或者多个字段(Field)组成，每个字段(Field)由一个字段名和一个或者多个值组成(有多个值的值称为多值字段(multi-valued))。在UDB中，每个文档(Document)都可能会有不同的字段(Field)集合；也就是说文档(Document)是没有固定的模式和统一的结构。实际上，从客户端的角度来看，文档(Document)就是一个JSON对象。

参数映射(Mapping)，所有的文档(Document)在存储之前都必须经过分析(analyze)流程。用户可以配置输入文本分解成Token的方式；哪些Token应该被过滤掉；或者其它的的处理流程。保存上述的配置信息，这就是参数映射(Mapping)在UDB中扮演的角色。在生产应用中， UDB的mapping的配置与数据库的表结构一样的重要，是根据用户的需求来设计的。

## 2.4分片索引(Shard)

在UDB集群中，一张表是由一个或多个分片索引(Shard)组成的。所有分片索引(Shard)是以一种整体的形式：表（索引）呈现给用户的。需要注意的是，尽管索引分片这个过程是自动的，但是在应用中需要事先调整好参数。因为集群中分片的数量需要在索引创建前配置好，而且服务器启动后是无法修改的。

## 2.5索引副本(Replica)

当集群负载增长，用户搜索请求阻塞在单个节点上时，通过索引副本(Replica)机制就可以解决这个问题。索引副本(Replica)机制的的思路很简单：为索引分片创建一份新的拷贝，它可以像原来的主分片一样处理用户搜索请求。同时也顺便保证了数据的安全性。即如果主分片数据丢失，UDB通过索引副本使得数据不丢失。索引副本可以随时添加或者删除，所以用户可以在需要的时候动态调整其数量。

# 数据建模

## 3.1创建索引

合众全文库本身支持动态结构，不用建表也可以根据数据类型自动建表结构，但如果有一些需要特殊指定的字段，也可以通过建表语句来配置的，同时每张表可以单独定义分片数和副本数。

语法：(sql语法不区分大小写，此处大写表示语法结构。UDB中字段区分大小写)

CREATE TABLE table\_name.type (

FieldName field\_type { analyzer=ik},

FieldName field\_type { doc\_values=true,index=not\_analyzed }

) WITH pri, rep

注：

索引字段默认是否分词, 在esql安装目录的配置文件中设置，如设置为ture默认分词，则建表语句字段都为分词，如个别字段想设置为整词，string类型字段，当添加其他属性时，必须增加index=not\_analyzed整词属性语法。其他类型字段不需增加index=not\_analyzed属性。

**其他参数设置**

CREATE TABLE TABLE\_NAME.TYPE (FieldName date{各种属性 })；

例：

1）CREATE TABLE template\_test1.base(

KSSJ DATE {format='yyyy-MM-dd HH:mm:ss.SSS'},

XM STRING,

DD STRING {include\_in\_all=false, index=not\_analyzed},

\_all SET ture);

注：有 \_all SET enable 的时候，如果某个字段不想被包含在\_all 中是 设置 include\_in\_all = false；反之，有 \_all SET disable 的时候，如果某个字段想被包含在\_all 中是 设置 include\_in\_all = true

2）CREATE TABLE zzz11.base(

ZJHM STRING {analyzer=mmseg},

TAG\_NUM LONG {doc\_values=true},\_all SET {analyzer=mmseg});

3）CREATE TABLE test\_ttl.base(

name STRING {index=not\_analyzed},

\_ttl SET 780s);

\_ttl: 表示过期时间, 时间到，当系统刷新时，将删掉索引。

**修改索引字段属性**

索引创建后可以修改整词字段属性为分词并指定分词器（ik,mmseg）和修改字段是否分词。如由分词修改为整词，由ik分词器修改为mmseg分词器。

语法：

CREATE INDEX ik STRING {索引属性 } ON table\_name.type (FieldName1 , FieldName 2);

例： create index ik string {analyzed=ik} on unimas\_test.info(xm,dd)

create index raw string {index=not\_analyzed} on name.info(姓名)

说明：unimas\_test.info; name索引已经创建，修改字段xm,dd,姓名的分词属性：ik分词器，

not\_analyzed整词。红色字体ik,raw为自定义标记。

**修改后查询语法：**

例： select \* from name  where names = gjw group by names.raw

使用属性标记查询为修改后的效果，查询没有标记的字段names为原来的属性效果。

## 3.2创建父子模型索引

创建子索引：CREATE TABLE table\_name.type (\_parent SET father\_table\_name )

创建父索引：CREATE TABLE table\_name. father\_table\_name()

**例**:

create table unimas01.info(\_parent  set people)

create table unimas01.people()

**创建object（对象）索引**

create table people\_trail.lg(XM string,DZ object (XXQH int例er,XXDZ string)) with 10,1；

create table people\_trail.wb(XM string,DZ object (XXQH int例er,XXDZ string), \_parent set people) with 10,1 父子模型的对象子表,父表名称：people

**创建nested（嵌套）索引**

create table people\_all.base(XM string,LG nested (ZJHM string,RZSJ date)) with 10,1

**说明：**

同一个索引下，多个\_type类型中的相同字段，字段的属性需一致。Esql在创建语句提交前会进行检查，如相同字段属性不一致，将无法成功创建表。

如： test.lg（name string,age string）

test.wb(name string,age integer)

test索引下，lg,wb都有age字段，字段属性不一致将不能创建成功。

## 3.3字段类型

|  |  |  |
| --- | --- | --- |
| **字段类型** |  | **说明** |
| 字符串 | string | 存储通用的中文和英文文本数据。 |
| 数字 | float, double, int例er, long, short, byte | 存储数字类数据，查询统计性能较好。 |
| 日期 | date | 存储日期类数据。默认格式dateOptionalTime（输入的数据应为格式化后日期的字符串） |
| 二进制 | binary | 存储二进制数据。（输入的数据应为base64的字符串） |
| IP | IP（0.0.0.0） | 存储IP地址 |
| 地理位置 | geo\_point | 存储地理位置坐标 |

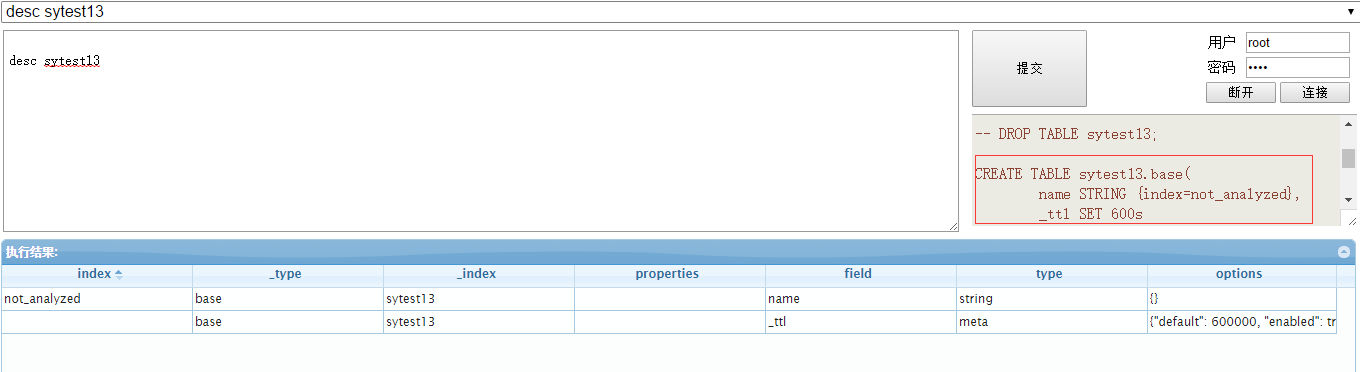
|  |  |  |
| --- | --- | --- |
| **分词器种类** |  | **说明** |
| 中文分词器 | ik、mmesg例 | 常用的中文分词器。 |
| 拼音分词器 | pinyin | 拼音分词器，用来拼音搜索和同音搜索。 |

## 3.4查看表结构

语法：desc table\_name

例：（1）desc people\_trail

图：



查看表结构，同时可以显示建表语句(红框处)。

\_index: 索引名称

\_type: 索引类型

Field:字段名称

Type:字段类型

Options: 其他配置

## 3.5创建元数据（注释）

语法：COMMENT ON TABLE INDEX\_NAME IS ‘string’; 索引添加注释

COMMENT ON COLUMN Field IS ‘string’；字段添加注释

例：

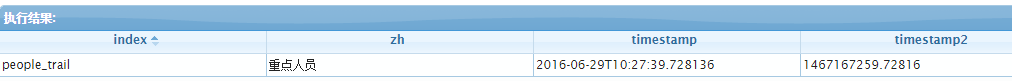
comment on table people\_taril is '人口轨迹'

comment on table people\_trail.zdry is '重点人员'

comment on column people\_trail.zdry.CSRQ is ‘出生日期’

**查询元数据：**

get meta people\_trail 查询索引



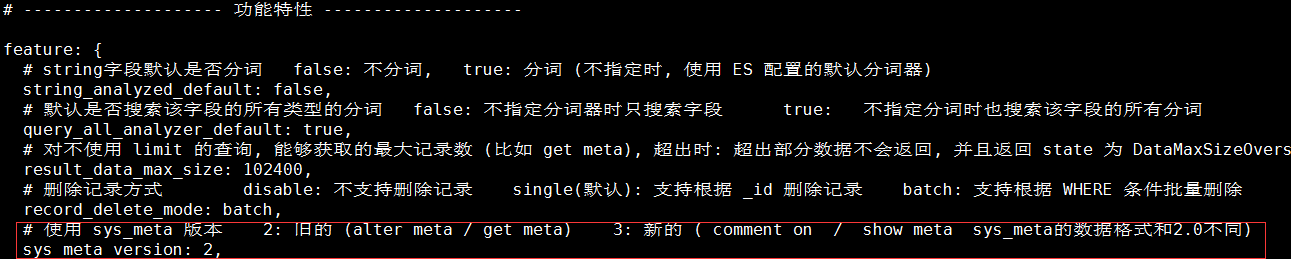
get meta people\_trail.xdry; 查询类型

get meta people\_trail.xdry.CSRQ; 查询字段

**说明**：

创建元数据语法与esql2.0版本完全不同，在esql3.0版本中需在配置文件中设置才能使用2.0语法，即2.0与3.0元数据的语法不能同时兼容，需选择一个使用。

修改方法：修改配置文件esql.yml的内容：



## 3.6索引别名操作

创建索引别名

语法：CREATE ALIAS alias\_name ON table\_name

CREATE ALIAS alias\_name ON table\_name\*

CREATE ALIAS alias\_name ON table\_name1, table\_name2, table\_name3

删除索引别名

语法 ：DROP ALIAS alias\_name ON table\_name

查看索引别名

语法： SHOW ALIAS

## 3.7关系人模型

通过统计语法查询人与人的关系

示例1：

创建索引：

CREATE TABLE relation.base(

    xm STRING {index=not\_analyzed},

    zzxz STRING {index=not\_analyzed},

    mz STRING {index=not\_analyzed}

) with 12, 0;

创建字段索引：为统计字段创建索引

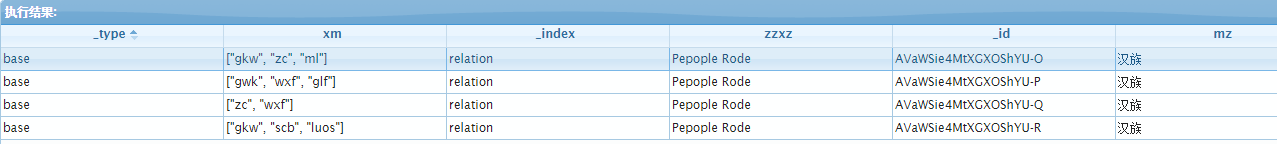
CREATE INDEX la string ON relation.base( xm );

写入数据：

BULK INTO relation.base(xm(),zzxz,mz) values ('["gkw","zc","ml"]','Pepople Rode','汉族'),('["gwk","wxf","glf"]','Pepople Rode','汉族'), ('["zc","wxf"]','Pepople Rode','汉族'), ('["gkw","scb","luos"]','Pepople Rode','汉族')

查询索引数据：

Select \* from relation



查询xm字段关系：

select \* from relation where xm = zc group by xm.la,xm



说明：

查询索引中与zc有关系的人，如上图显示，查询结果同时显示其他人员关系。索引 la 为自定义名称，被统计关系的字段xm 必须为整词，因此如索引创建时是分词，为xm字段创建索引时需给出整词属性，语法如示例2。

示例2：

CREATE TABLE relation.base(

    xm STRING,

    zzxz STRING {index=not\_analyzed},

    mz STRING {index=not\_analyzed}

) with 12, 0;

CREATE INDEX raw string {index=not\_analyzed} ON relation.base( xm );

# 数据操作

## 4.1插入数据

INSERT INTO table\_name.type (insert\_columns) VALUES (insert\_values)

**例**:

insert into qs\_test01.base (a,b,c) values (1,'2015-01-01T00:00:00.000Z','hello');

insert into qs\_test01.base (\_id,a,b,c) values (100,1,'2015-01-01T00:00:00.000Z','world');

**\_id字段可以缺省，缺省时由系统自动生成，唯一且在结果中会返回。**

插入父表数据

insert into unimas01.people (\_id, a, b) values ('1','aa','bb')

插入子表数据

insert into unimas01.info (\_parent, C) values ('1','cc')

\_parent 表示关联的父表，C 表示父表字段，values中的1，表示关联父表的\_id=1的数据

插入对象多值

insert into unimas01.base (e()) values ('{"e1":123,"e2":"123"}')

e 为多值字段，值是"e1":123,"e2":"123"

insert into unimas01.base (e(),f) values('{"e1":woe,"e2":"678"}',10001)

## 4.2批量操作

**语法：**BULK INTO index\_name.type VALUES (value1,value2,value3), (value1,value2,value3), (value1,value2,value3)

例: bulk into unimas01.base (a,b,c) values (ss,2d,3f),(se,gr,gr),(7gs,dd,bird)；

bulk into my\_mh.base(names()) values ('["zjl","gjw","hyl"]'),('["zjl","gjw","hyl"]'), ('["zjl","gjw","hyl"]'),('["zjl","gjw","hyl"]'), ('["gjw","zzy"]'), ('["zhy","scb","zzh"]')

## 更新操作

UPDATE index.type SET field1=’xxx’, field2=’xxx’ WHERE \_id =’xxxxxxxxxx’

注意：\_id=‘’的引号不能省略

例：

update test.base set username=’zry22’, nickname='合众101' where \_id= 'AVUov2rYb9GONmiBZ3oY'

UPSERT INTO index.type(\_id,field1,field2) values (‘xxxxxxx’,’xxx’,’xxxx’)

例：

upsert into test.base(\_id, username, nickname)  values('AVUov2rYb9GONmiBZ3oZ', 'zry0U', '中国0U’)

## 删除操作

**4.4.1 删除数据**

语法：DELETE index\_name.type WHERE \_id =

用户需有delete权限，where条件只支持\_id.

例：

delete people\_trail.xdry where \_id = '110101196312224033'；

delete account\_12306 where \_id = AVMxHNsCjZvNnPkdkx5j；

说明：等号右侧引号可省略。

* + 1. **删除索引**

语法：DROP TABLE index\_name

将comment 元数据一并删除。

## 复制索引

语法：COPY index\_name TO new\_index\_name

复制索引结构及数据,先创建索引再复制数据也是可以的，旧索引上的字段在新索引里更新属性，也是支持复制的。

例：copy people\_trail to new\_trail;

# 搜索查询

## 5.1基本语法

SELECT \* FROM table [WHERE field= ‘ ‘ AND field BETWEEN 10 TO 100 limit 10,10 HIGHLIGHT field1,field2 ORDER BY filed1,filed2 DESC]

说明：所有关键字(select ,from ,where, and, or, between to,limit highlight等)不区分大小写,[]括弧内的语句为可选。where field支持 “=”，“ >”，“ <”，“<=”号

**例**：

select \* from people\_trail01 where ZZXZ = 浙江省杭州市 and CSRQ >= '2007-10-01' and KSSJ between '2010-04-30'to '2012-04-30' limit 10,10 highlight \* order by CSRQ

## 5.2搜索语法

### 5.2.1跨表搜索

需要在多个表中进行搜索时，可以在from语句后面添加多个表名，分别用逗号隔开。

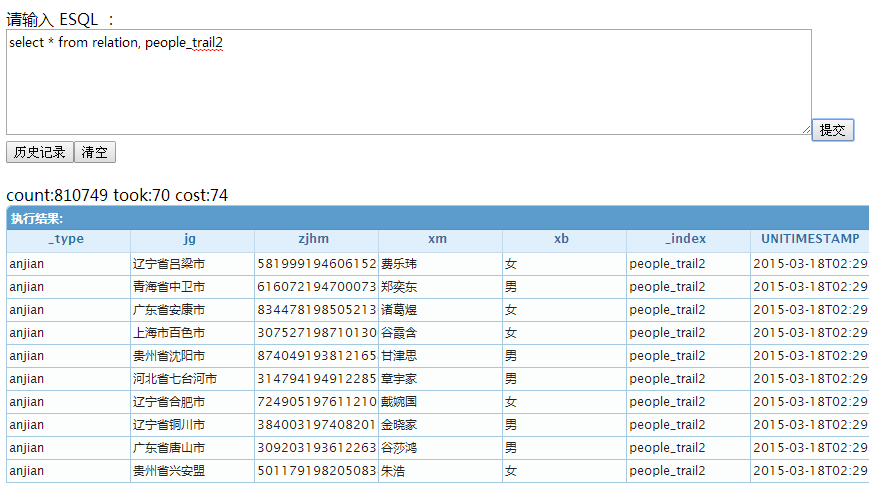
**例**：（1）select \* from people\_trail2，结果集数为800000。



（2）select \* from relation，结果集数为10749。



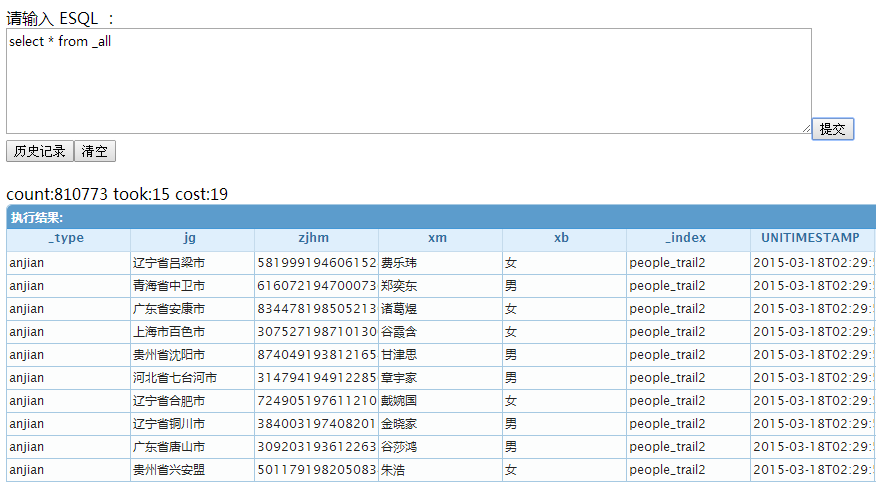
（3）select \* from relation, people\_trail2，结果集为（1）（2）的和。



### 5.2.2整库搜索

需要在整个库中进行搜索时，可以使用\_all关键字替代表名。

**例**：select \* from \_all



### 5.2.3全文搜索

全文搜索，又叫全字段搜索,是指对表中的所有字段进行搜索，不指定某个字段，可以使用\_all关键词。

**例**：select \* from people\_trail2 where \_all = 康思晓

另：下面两个语句等同。

语句1：select \* from people\_all where \_all=伟 女 中国 和

语句2：select \* from people\_all where \_all=伟 and \_all=女 and \_all=中国 and \_all=和

至于两个语句，谁的性能更好，可以在实际环境中进行测试，以便选择使用。

### 5.2.4分页(limit)

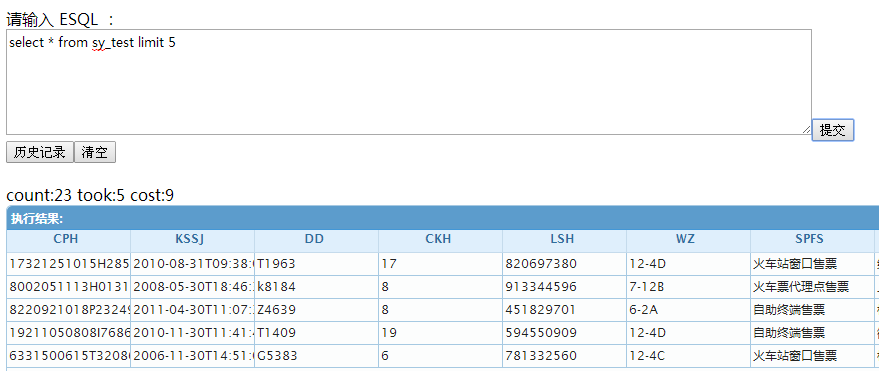
语法：limit语句，支持对搜索结果的分页。

limit pagesize（pagesize：分页大小）

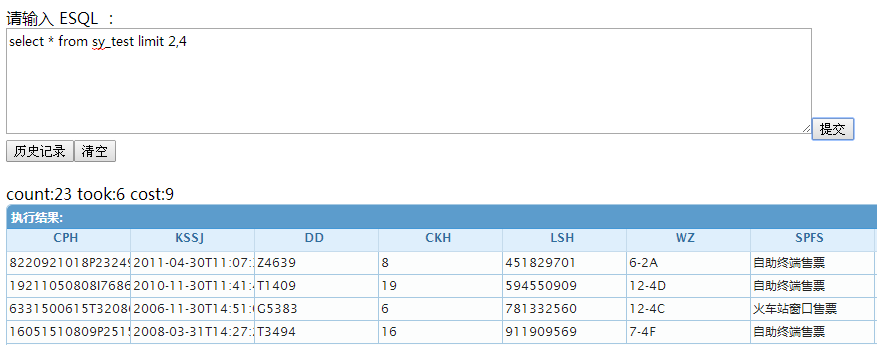
limit start,pagesize（start：起始位置，pagesize：分页大小）

**例**：

（1）select \* from sy\_test limit 5



（2）select \* from sy\_test limit 2,4

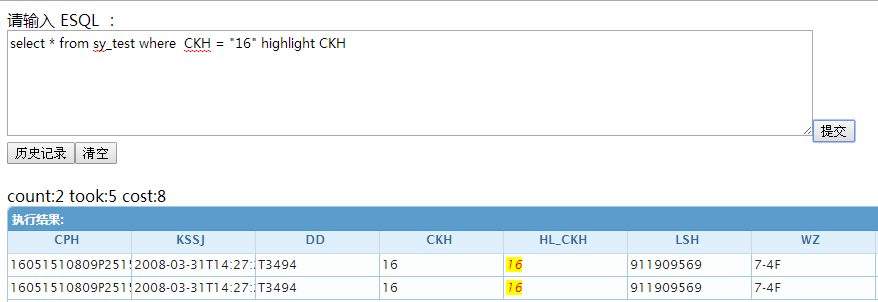


### 5.2.5高亮(highlight)

对搜索结果可以进行高亮显示，使用<em></em>标识。具体如何显示，可以根据用户需求在html页面中指定em的样式表决定。

高亮搜索一定要配合where子句，要不然就不知道哪些信息需要高亮。

**例**: （1）select \* from sy\_test where CKH = "16" highlight CKH

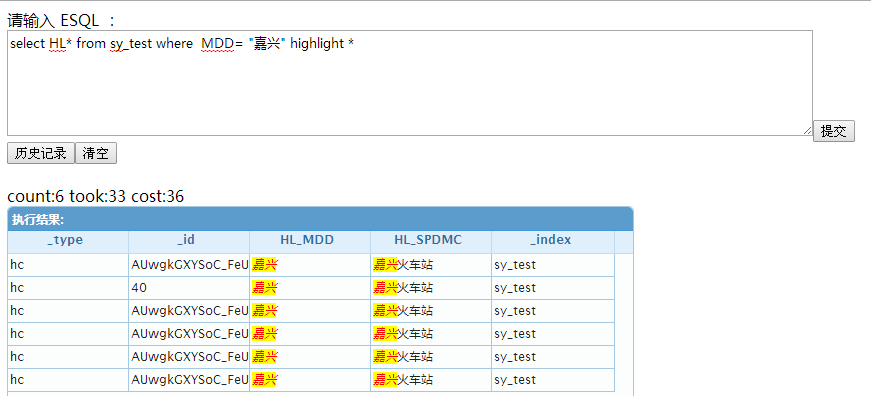


注意：

select \* from table where all= 李四 highlight all是没有用的。\_all不会出现在结果中。

高亮字段也支持\*,这样只要满足条件的字段都会被高亮

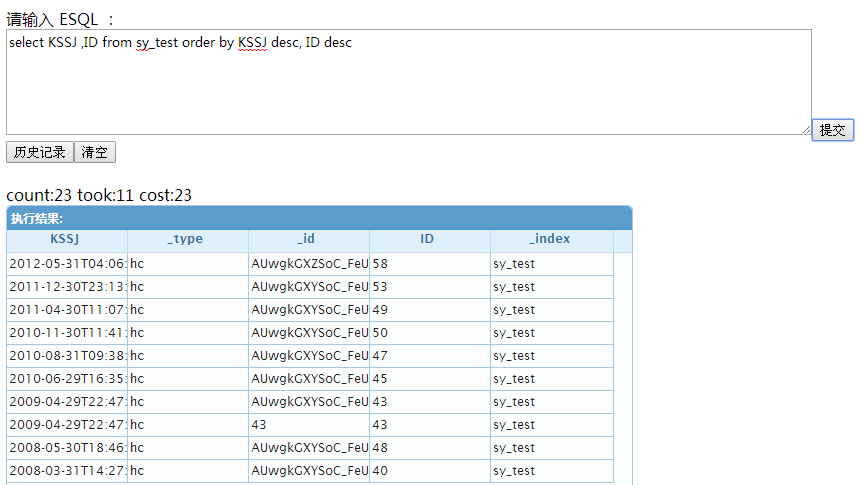
（2）select HL\* from sy\_test where MDD= "嘉兴" highlight \*



### 5.2.6排序(order by)

语法：order by field1 [asc],fileld2 desc,filed3…

**例**: select KSSJ ,ID from sy\_test order by KSSJ desc, ID desc



默认字段按升序排序，支持多字段排序。

支持不带条件的整表order by ,一定要注意数据量或者使用doc\_value属性,否则容易内存溢出。

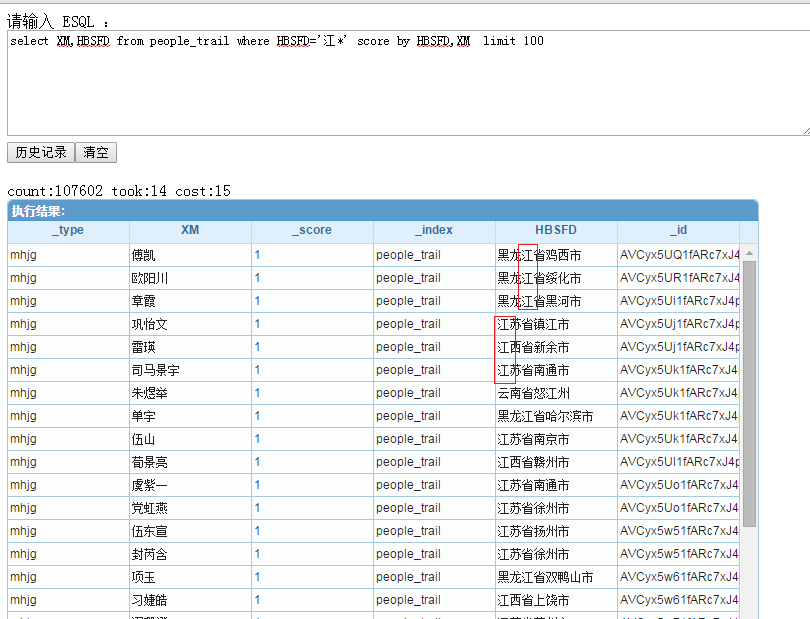
**支持字段得分权重的设置**

**使用场景：**查询条件在多个字段中存在，通过score by 控制某个字段优先返回结果或直接 给出权重。

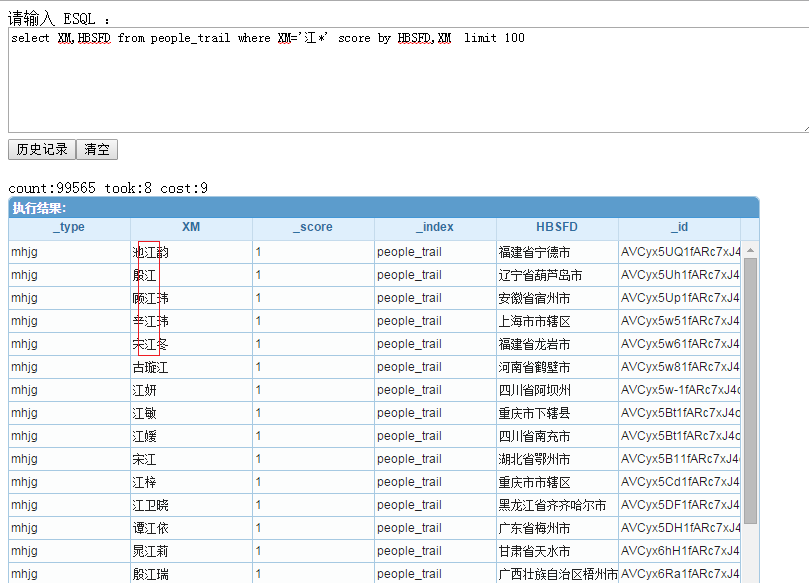
说明：返回结果中\_score字段表示得分

**例：**

select XM,HBSFD from people\_trail where HBSFD='江\*' **score by** HBSFD,XM limit 100



select XM,HBSFD from people\_trail where XM='江\*' **score by** HBSFD,XM limit 100



select XM,HBSFD from people\_trail where HBSFD^10='江\*'or XM^100='江' 

select XM,HBSFD from people\_trail where HBSFD^100='江\*'or XM^10='江'



### 5.2.7排名(score by)

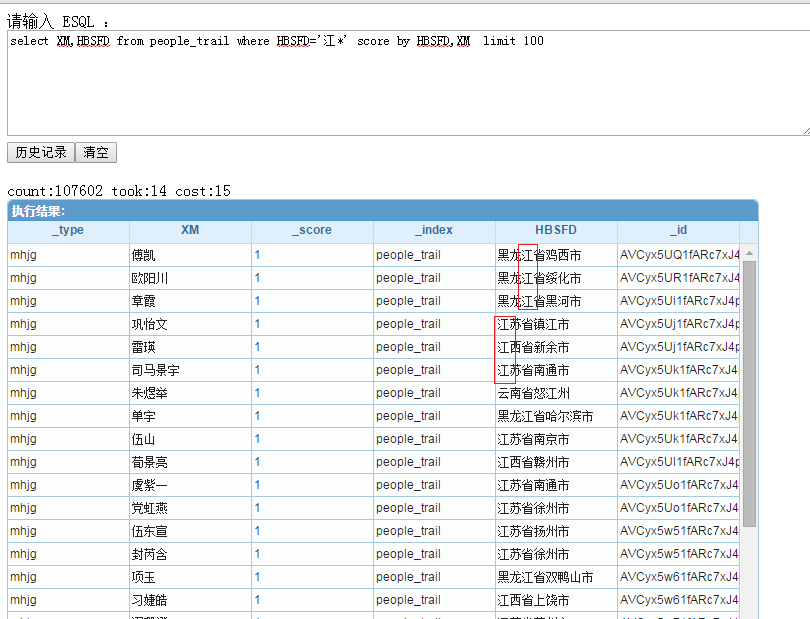
**支持字段得分权重的设置**

**使用场景：**查询条件在多个字段中存在，通过score by 控制某个字段优先返回结果或直接 给出权重。

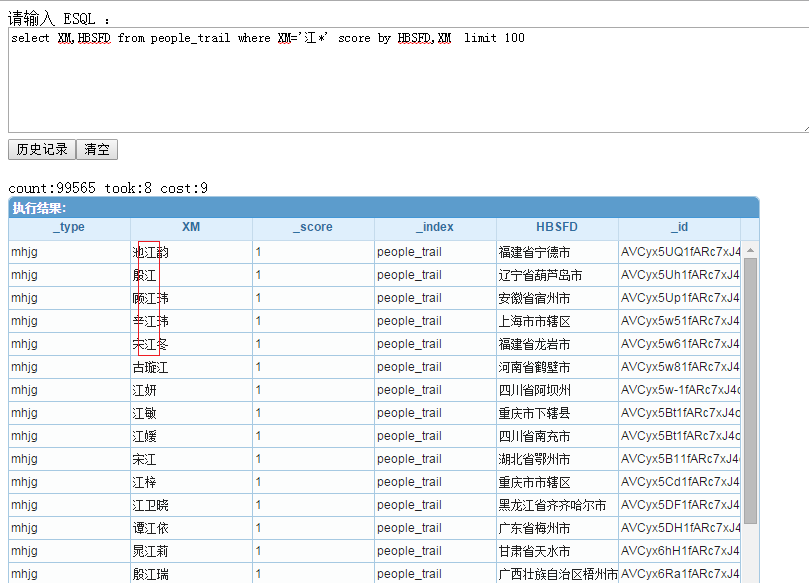
说明：返回结果中\_score字段表示得分

**例：**

select XM,HBSFD from people\_trail where HBSFD='江\*' **score by** HBSFD,XM limit 100



select XM,HBSFD from people\_trail where XM='江\*' **score by** HBSFD,XM limit 100



select XM,HBSFD from people\_trail where HBSFD^10='江\*'or XM^100='江' 

select XM,HBSFD from people\_trail where HBSFD^100='江\*'or XM^10='江'



## 5.3查询条件

### 5.3.1嵌套表查询(insideq)

语法：SELECTE \* FROM table\_name.type WHERE INSIDEQ(SELECT子句)

例:

select \* from qs\_test.people where insideq(select \* from jl where jl.info=zhejiang) or insideq(select \* from jl where jl.info=unimas)

### in 条件查询

语法：select \* from index\_name where field\_name in (xxx,xxx,xxx,…..)

例: select \* from people\_test where MZ in (回，蒙，苗)

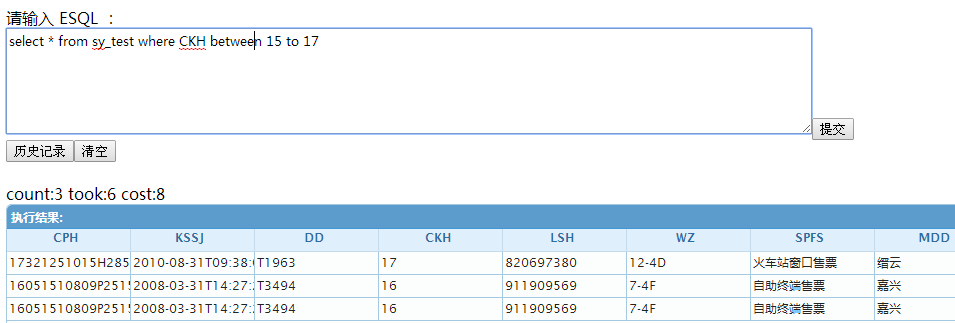
in()中的条件默认支持1024个

### Between to

对于日期、数字等类型，可以使用范围搜索，使用between to 子句。

**例**：

（1）select \* from sy\_test where CKH between 15 to 17



### !=,>=,>,<=,<

语法1：字段 != 值。

**例**：select \* from mytest where \_all="张三" and ZJHM != ‘3210101010010100101010’

语法2：数字或日期类型 [>=,>,<=,<] 值

**例**：select \* from sy\_test where KSSJ > ‘2010-08-31’

### 5.3.5 is null和is not null

语法：字段 is null。

**例**：select \* from passport where PASSPORTNAME = ‘卡口’ and CREATOR is null

select \* from passport where PASSPORTNAME = 卡口 and CREATOR is not null

汉字可以省略单引号，但是为了统一规则，建议都使用单引号。

### 5.3.6 Id查询

所谓id查询，就是直接根据id来得到相应的记录，在全文库中使用这种方法得到数据最快，随机查询时间可以达到5ms以下。

**例:** select \* from relation where \_id =210912198801109875\_816478197212015774

## 5.4 高级搜索

### 5.4.1 DQ语法

相当于2.0版本的q=查询，对于高级的检索，使用普通的SQL表述较繁琐的可以采用DQ语法， DQ支持AND,OR的复杂逻辑关系。AND,OR 需大写。括号中需加单引号。AND不能省略。

**例**：

Select \* from my\_table where DQ(‘\_type:people AND age:[20 TO 30]’)

select \* from people\_all where DQ('拐卖 AND 175\* AND短头发 AND \_type:base') highlight \* limit 0,10

### 5.4.2模糊查询

**支持 like 、 通配符？ \* 等模糊查询。**

**例**： select \* from vechile\_trail where hphm like ‘浙A88?8?’

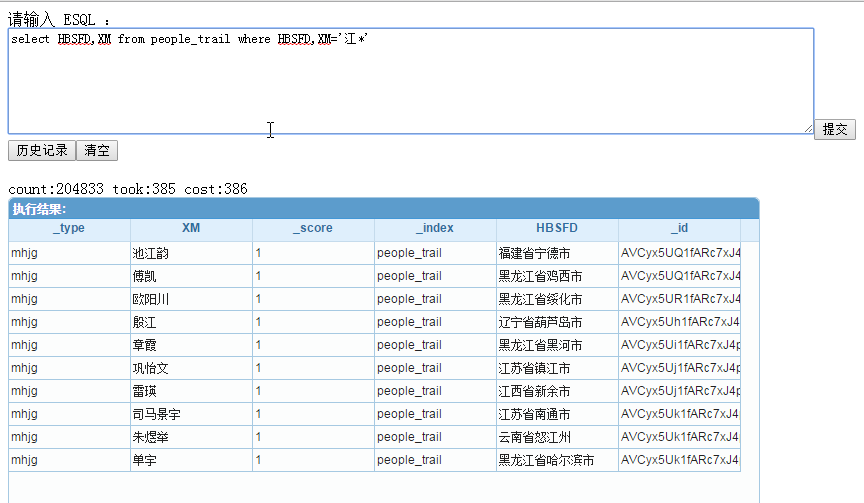
select count(\*) from record where TransportID like in (‘33060251\*’，‘330607031\*’)

注意： like 右侧必须使用 ‘’ 单引号。使用like时查询字段属性应为整词not\_analyzed.

### 5.4.3 多个字段中搜索

**例：**select HBSFD,XM from people\_trail where HBSFD,XM='江\*'

搜索结果没一条都有“江”



# 统计分析

语法：SELECT COUNT(\*) FROM table\_name where field=”” group by field1，field2。

* 蓝色为语法结构，大小写不敏感。

函数：支持count，min，max，sum，avg统计函数。

count(\*)为固定格式，对应的统计结果为count，在任何统计中都会包含此结果，不管你是否使用了count(\*)。

select count(\*) from table

select count(\*) from table group by wayid

**例:**

1. select min(speed) as min\_speed， max(speed) as maxSpeed from table group by wayid
2. SELECT SUM(c3) as c5,avg(c2)as c4 FROM statistic group by c1
3. select avg(c2)as c4 from statistic

注意事项：目前group by的使用尽量控制在2个字段以内，多余2个字段会造成内存溢出。

**Doc\_values属性设置**

在没有对相关字段进行Doc\_value设置的情况下，排序（order by）和分组统计（group by）的计算时会消耗大量内存，容易造成集群内存溢出。因此，在建表时可以将一些经常进行排序和分组统计的字段设置为Doc\_value。这样就可以在支持大数据量操作的情况下避免出现上述的问题。

语法：

create table tablename.type (

fieldname string {index=not\_analyzed, doc\_values=true},

)

设置doc\_values的对应字段，必须为整词（yes），否则会建表失败。

设置doc\_values的字段在建表成功后，可以通过desc tablename命令（见3.3）来查看表结构信息。如果成功，能看到doc\_values字段的值为true；否则为false。

同时可以调用监控命令cat fielddata（查看cache）和管理命令clear cache（清除缓存）来验证doc\_values设置是否起作用。

## 6.1默认分组

group by 后面只写字段名时，代表使用默认分组规则，对字段的分词结果进行统计。专业表示为trems。

比如，你有一个颜色字段，字段的内容有”黄色”，”黑色”，”蓝色”，”白色”，分词的结果可能是”色”，”黄”，”黑”，”白”，”蓝”。如果该字段按trems分组，会出现以下结果:

色:1000

黄:200

黑:100

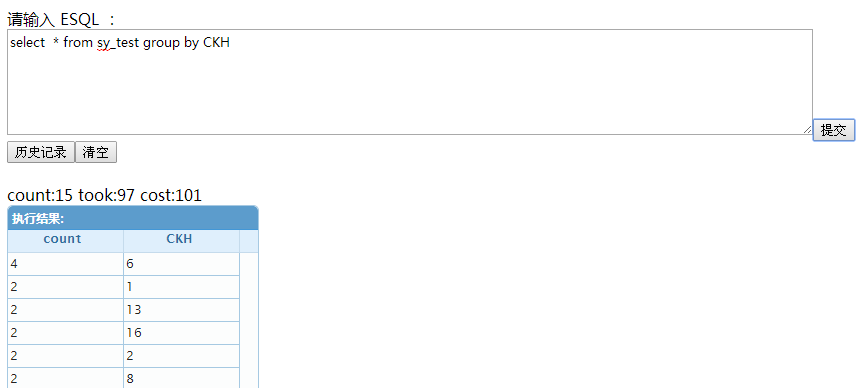
白:500

蓝:200

共计1000条，和我们一般意义上的理解有偏差，但结果也能引用。还有一种方法可以和通常意义上的结果保持一致，就是在入库时指定该字段不分词，做整词规则。

例子：

select \* from sy\_test group by CKH



## 6.2 range分组

数字等类型可以使用range分组来自定义数据统计区间。

例子：

select count(\*) from table speed.range[(0-40)(40-100)(100-)]

说明：指定speed字段采用range分组，

0-40：大于等于0，小于40；

40-100：大于等于40，小于100；

100-：大于等于100；

(-x)和(x-)都支持。

## 6.3 ip\_range分组

专门针对ip类型的分组。

例子：

select sum( flows ) as total\_flows from bb group by speed.ip\_range[(192.168.1.1-192.168.1.23)(192.168.1.23-)]

## 6.4 histogram分组

数字间隔分组，比如一个速度字段speed，我们以10km/h为间隔进行分组统计。

例子：

select count(\*) from car201110 group by speed.histogram[{interval:10}]

结果如下:

count : 738 speed : 20

count : 774 speed : 30

count : 786 speed : 40

count : 757 speed : 50

count : 740 speed : 60

count : 760 speed : 70

count : 779 speed : 80

count : 746 speed : 90

count : 803 speed : 100

count : 800 speed : 110

count : 759 speed : 120

count : 711 speed : 130

count : 777 speed : 140

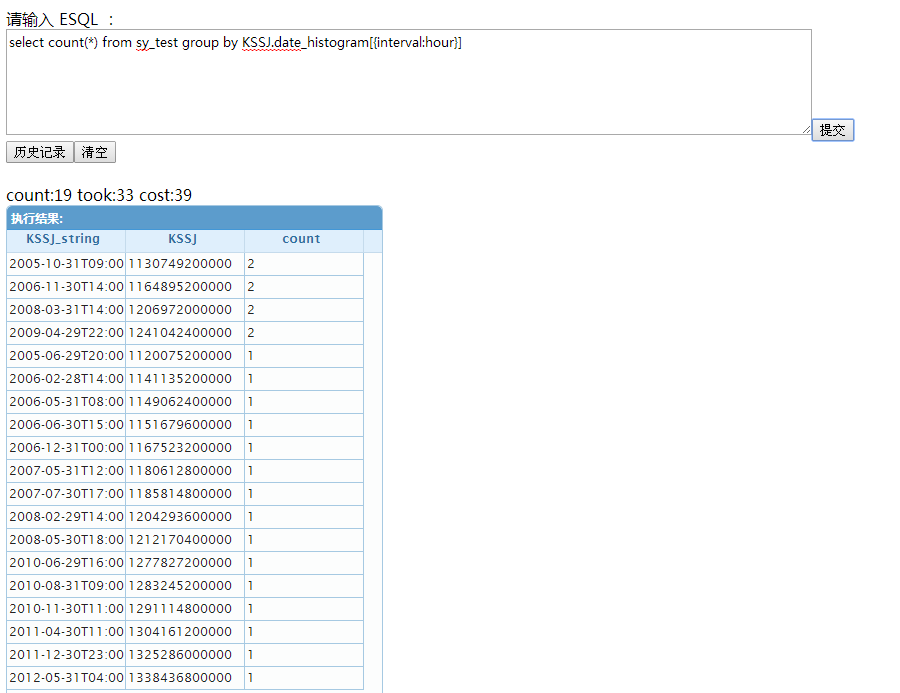
count : 70 speed : 150

说明：Speed:20 代表 20>speed>=10，以此类推。

## 6.5 date\_histogram分组

与上相同，不过date\_histogram是针对日期类型的，可以使用year，quarter，month，week，day，hour，minute，second为单位。

例子：

select count(\*) from sy\_test group by KSSJ.date\_histogram[{interval:hour}]

# 碰撞比对

适用于碰撞比对使用场景。

通过对需要进行碰撞比对的数据对象进行分析，将一张基础索引表作为父表，其他多张分类数据表作为子表。在明确父表和子表之间的主外键关联后，建立一个特殊的父子数据模型（一对多，一个父表可以对应多个分类子表），父子表复杂对象模型。该模型的用途在于，同时指向一个父表的多个子表之间依据各自的检索结果取交集后，来迅速并准确定位父表中对应的记录。这就是UDB提供的碰撞比对功能，我们称之为fastjoin。

比如：搜索一类人。他们入住在如家酒店，同时也在沸蓝网吧上过网，又在xx月xx天坐过火车。

上述数据在父子表复杂对象模型中是以一对多的方式存储的，即：人基本信息作为父信息，旅馆、网吧、火车这3个轨迹信息作为子信息以数组的形式和父信息进行组装，在UDB中看成一条记录。

因此，当需要同时满足如家酒店、沸蓝网吧、坐过火车，这3个子搜索条件时，便对这3类子表信息进行了取交集的运算，而在外部看来他们父子1对3的组合是一条记录，所以能迅速并准确定位到命中的记录。

## 7.1快速比对（fastjoin）

通过fastjoin技术，在最大程度保持数据灵活性的基础上获取高性能的碰撞比对功能，实现上述应用描述的需求。支持fastjoin搜索的前提是，索引必须以父子关系建立模型。例如：people\_all索引，将people基本信息作为F父表，将lg（旅馆）、wb（网吧）等作为多个子表。子表通过指定的外键与父表进行关联。

进行搜索时，可以通过lg、wb等多个子表之间的join查询来获取父表的对应记录。

**例**：

Select \* from people\_trail where MZ ='汉'and fastjoin(select \* from lg where DD='布丁酒店')

**注意：udb2.2（ES2.0）版本以上，创建父子表模型时，必须先创建子表，再创建父表。**

## 7.2逻辑关系

**语法支持and 和 or，not的关系**

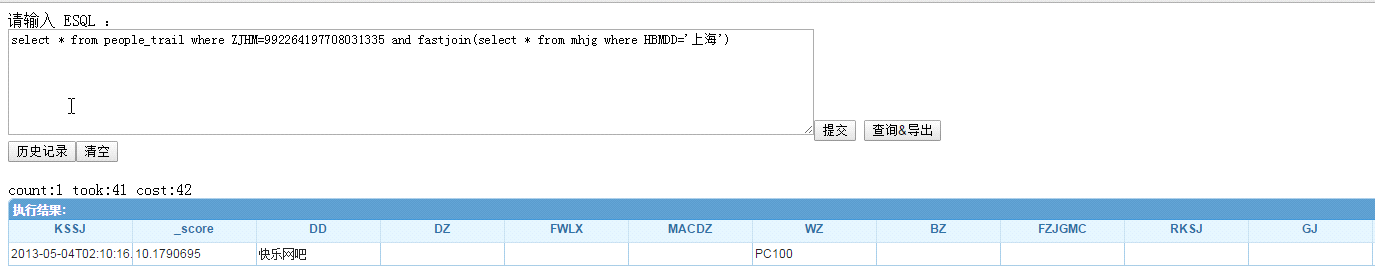
**例：**

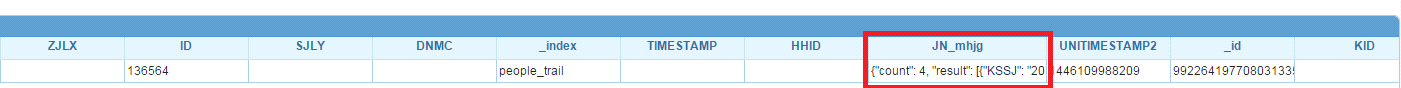
1）Select \* from people\_trail where MZ ='汉'and fastjoin(select \* from lg where DD='布丁酒店') and fastjoin(select \* from wb where DD='星光网吧') or fastjoin(select \* from mhjg where HBMDD='杭州')

上面语句意思是：一个汉族人乘飞机到杭州，且住在布丁酒店，或在星光网吧上过网。

2）Select \* from people\_trail where MZ ='汉'and fastjoin(select \* from lg where DD='布丁酒店') and fastjoin(select \* from wb where DD='星光网吧') and not fastjoin(select \* from mhjg where HBMDD='杭州')

3）select \* from people\_trail where ZJHM=992264197708031335 and fastjoin(select \* from mhjg where HBMDD='上海')





查询结果中JN\_mhjg字段为子表匹配的结果。

**JN\_mhjg字段内容：**

{

"count": 4,

"result": [

{

"KSSJ": "2011-10-31T11:50:00.000Z",

"DHHM": null,

"HBRQ": "2011-10-31T11:50:00.000Z",

"\_score": 5.2724576,

"DJPH": null,

"BH": null,

"ZHSJ": null,

"YF": null,

"HJGSDDM": null,

"YWM": null,

"WZ": "93",

"ZJHM": "992264197708031335",

"HBMDD": "上海",

"CW": "经济仓",

"GJ": null,

"XB": null,

"SJBZID": null,

"DD": "T36",

"CSRQ": null,

"\_type": "mhjg",

"XLJS": null,

"XM": "黎琳奕",

"CRSJ": null,

"HZH": null,

"UNITIMESTAMP": "2015-10-29T08:56:15.808Z",

"ZZGBSJ": null,

"DDH": null,

"ZJLX": null,

"ID": "110524",

"\_index": "people\_trail",

"TIMESTAMP": null,

"JSSJ": null,

"HBSFD": "绍兴",

"UNITIMESTAMP2": 1446108975808,

"\_id": "AVCyy-O81fARc7xJ8cAj",

"HJGSD": null

},

{

......

},

{

...... }

]

}

**where条件逻辑关系支持：and、or、not**

**例:** select \* from my\_table where ((a=1 or b = 2) and c = 1) **and** (d = 1 **or not** ( e = 1 and f = 1 ) )

# 符号使用

## 8.1 占位符号使用

**？**占位 1个字符

**\*** 占位多个字符

需要加 ''来标识是一个完整的整体

使用占位符的字段需要字段属性为整词（not\_analyzed）

## 8.2 引号使用

**‘’** 单引号： 标识完整的字符串，ESQL3.0中，除数字类型外，其他类型都行要加单引号。

**例：**

1. select \* from '12306'.'320gc';
2. select \* from people\_trail01 where \_all='王\*';

\_all = something 必须写成 \_all = ‘something’ ； \_all = ‘192.168.1.100’ \_all = ‘2015-10-11’

\_all = 汉字，**中文汉字**可以不使用单引号查询，但是为了统一规则，建议都使用单引号。

如果字段名中带以下字符的，也需要带单引号

如： DZS.常驻人口地址=杭州, 写成: DZS.’常驻人口地址’=’杭州’ , 千万别写成‘DZS.常驻人口地址’=’杭州’ 这样 DZS.常驻人口地址会变成一个整体，将无法检索。

“”双引号： 用于检索是表示进行整词检索；

与esql2的区别：esql2右值不需要单引号，而esql3必须加单引号

# 权限控制

Esql3.1支持不同用户有不同访问权限，支持特定用户特定ip访问。目前授权用户访问级别控制在表级别,可以授权用户访问特定表。

Esql初始化默认账号： root/密码;（管理员密码查看邮件不在此文档中说明） 管理员用户，拥有所有访问权限，创建其他用户并为其授予各种权限，目前支持的权限有select、create 。登录esql页面首先连接用户。

图：



## 9.1. 创建用户

root用户下创建普通用户：

**语法：**CREATE USER username IDENTIFIED BY 'password';

**例:** create user sy identified by 'ys'

## 9.2. 查看用户

语法：show users

Root用户查看系统所有其他用户。

## 查看系统权限

语法：SHOW PRIVILEGES

查看有本系统有哪些权限可以设置。

## 赋予用户权限

Root用户下为用户赋权限：

**语法：**GRANT ROLES ON TABLES TO NAME@STRING （ROLES: create,select,drop）

**例：**grant select on sjrz\_qm\*.\* to test@'10.68.11.%'

grant select on account\*.\* to sy@'%'

grant select on account\*.base to sy@'%'

grant drop on people\_trail\*.base to username@'%';

索引名支持通配符。如：授予用户sjrz\_qm\_20121101、sjrz\_qm\_20121101索引权限，可以用通配符代替sjrz\_qm\*.\*。点后的通配符代表索引的所有\_type。

@符号后为指定的访问ip地址。%代表所有地址，[即不设置ip。@符号为语法一部分不能省略](mailto:即不设置ip.@符号为语法一部分不能省略)。

**为使用红名单的用户赋权：**

语法：GRANT ROLES ON TABLES TO NAME@STRING with redkeys

例：grant select on people\_trail.\* to test@'%' with redkeys;

说明：如果用户授权时不加with redkeys，红名单功能对此用户不生效。其他用户仍然可见红名单内容

## 9.5. 查看用户权限

Root用户查看其他用户权限

语法： SHOW GRANT WITH users;

例： 显示test用户的所有权限

show grant with test

show grant with test@‘%'

显示test用户在本机的所有权限

show grant with test@localhost

show grant with test@'127.0.0.1'

显示test用户在某网段的所有权限

show grant with test@’10.68.11.%'

登录创建的用户，新权限只有重新登录后才生效。

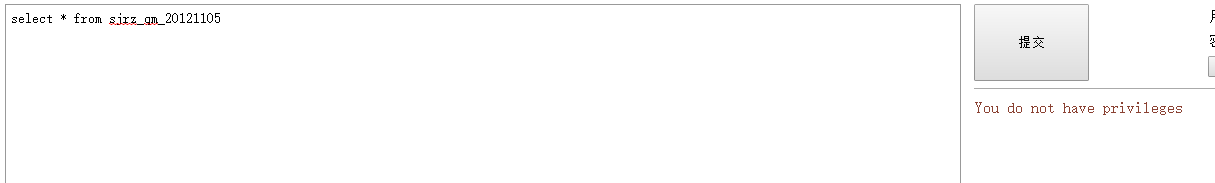
**语法：**SHOW tables 查看用户名下有权访问的所有表。

DESC tables 如果查看没有权限的表结构，会提示没有权限。

**例：** select \* from table sjrz\_qm\_20121101

## 9.6. 查看用户非权限内的表

访问没有权限的表，会提示没有权限。



## 9.7. 收回用户授权

**语法：**REVOKE roles ON tables FROM NAME@STRING; （ROLES: create,select）

**例**：revoke select on account\*.\* from sy@'10.68.12.%'

注意revoke时用 from 不是 to

收回权限需要一个一个收回，即收回select ,create等权限，分别执行。

## 9.8. 删除用户

Drop user xxx

Root用户执行

## 9.9. 修改用户密码

语法：ALTER USER NAME IDENTIFIED BY STRING

例： alter user sy identified by ‘1234sy’

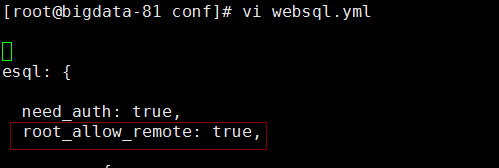
说明：root用户可以修改所有用户密码，用户可以修改自己的登录密码。

root用户可修改root密码。

## 9.10. root用户远程登录功能

软件包默认设置是关闭root远程登录，即非本地ip的web页面上不能登录root用户。

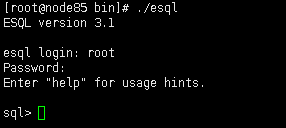
安装时可修改配置：设置为true,即可非本地IP使用root用户

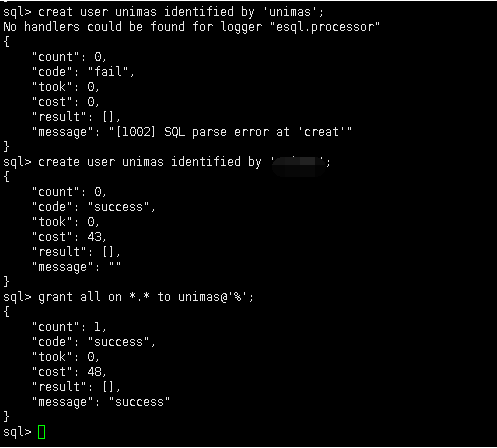


出于安全考虑，此参数不建议设置为true。Root用户保持本地ip登录比较安全。

管理员可以通过ssh工具远程连接esql服务器到安装目录的/bin下使用esql shell创建用户，

然后在其他ip地址web页面使用非管理员用户查询数据。具体使用方法如下：





此例中的赋权语法是给用户unimas所有权限，目前所有权限包括可以创建其他用户，但是还不支持为所创建的用户赋权限。

## 9.11 版本更新保留授权

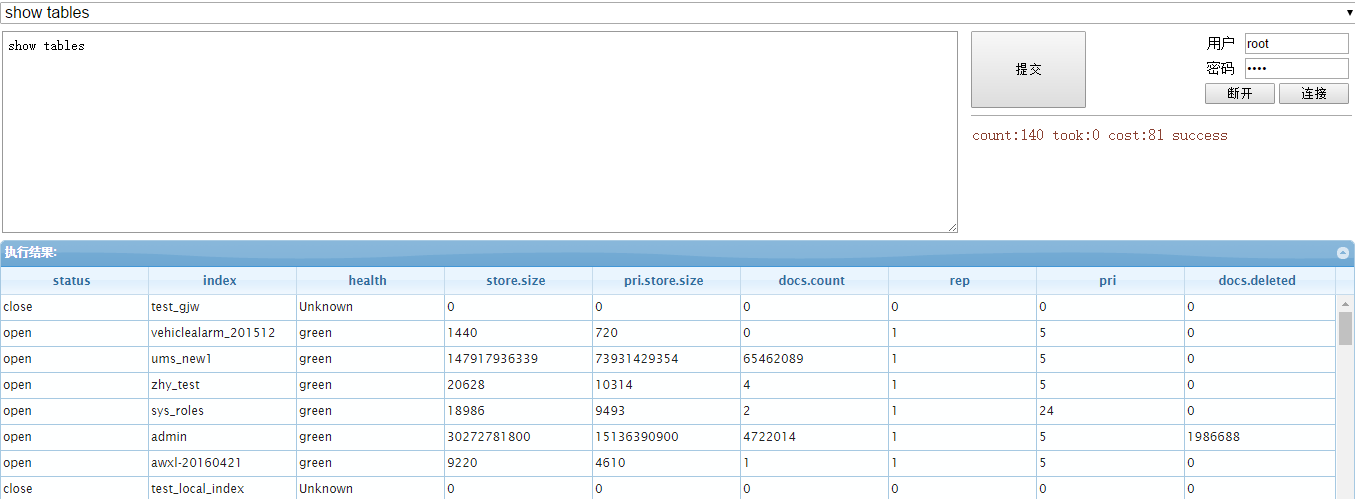
由于版本更新采用重新安装的方式，因此所有设置会清除，为避免权限重新设置，更新版本前将../db/esql.dat文件保留，覆盖到新集群中即可。Esql.dat文件存储用户和权限数据。

# 监控管理

可以查看集群状态的各种参数

### 10.1查看所有表

语法：show tables

图：

说明：

status：索引状态。Open=打开，能够搜索的工作状态。

index：索引名称。

Health：代表健康状态。Green=正常，yellow=恢复中，red=错误，prue=分片迁移（除了red，其他均不影响搜索功能）。

sore.size：存储占用的总空间。

pri.store.size：主存储数据占用的空间，不包括副本。

docs.count 代表是表中的记录数。

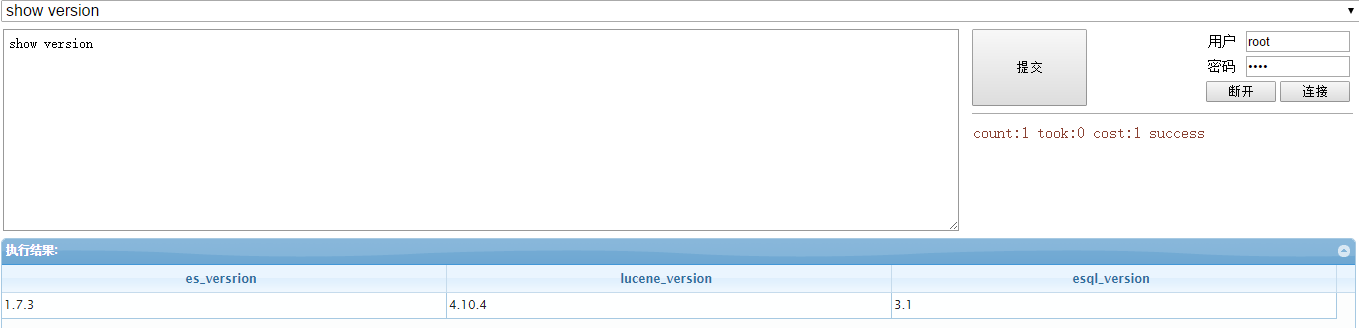
rep：副本数，每个表可以单独指定。

pri：分片数，每个表可以单独指定。

docs.deleted：代表打了删除标记的记录数，在表优化时会才会真正删除掉。

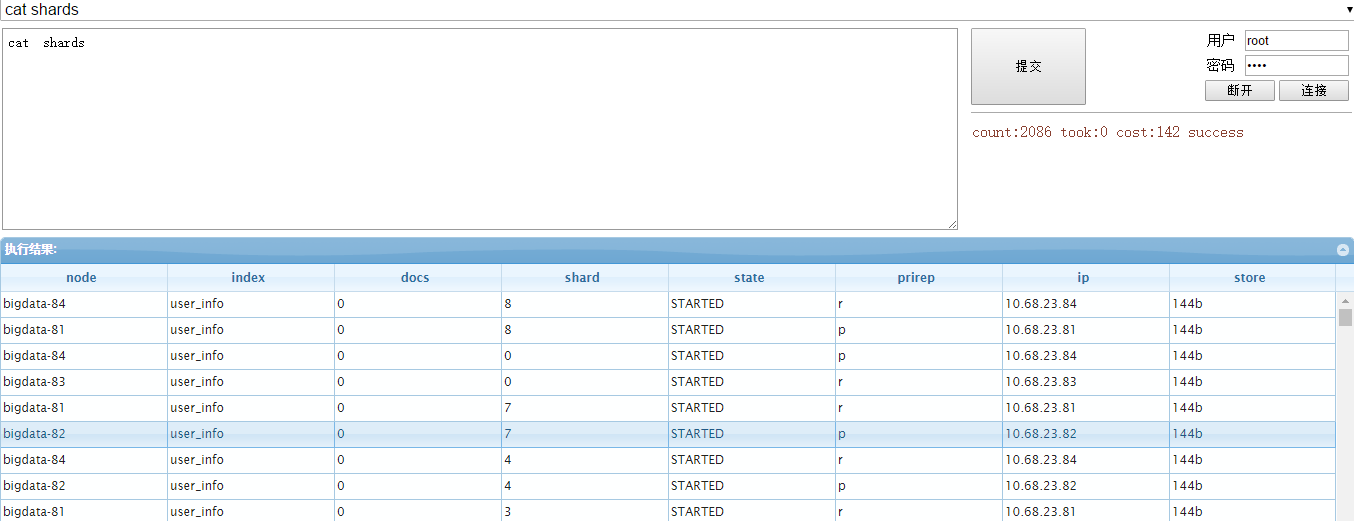
### 10.2查看版本信息

语法： show version

图：

### 10.3查看分片情况

语法：cat  shards/ cat shards  index\_name

图：

说明：

node:节点名称信息hostname。

index:索引名称。

docs:该shard上存储的docs记录数。

shard:分片名称。

state:状态。started=运行状态。

prirep:主副分片标志。p=主，r=副。

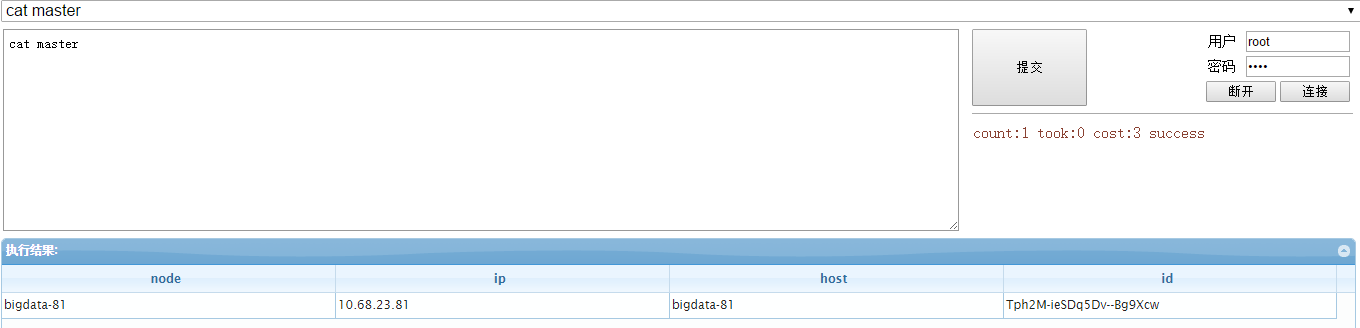
ip:shard所在ip。

store:存储容量。

例: cat shards  sjrz\_qm\_20160419

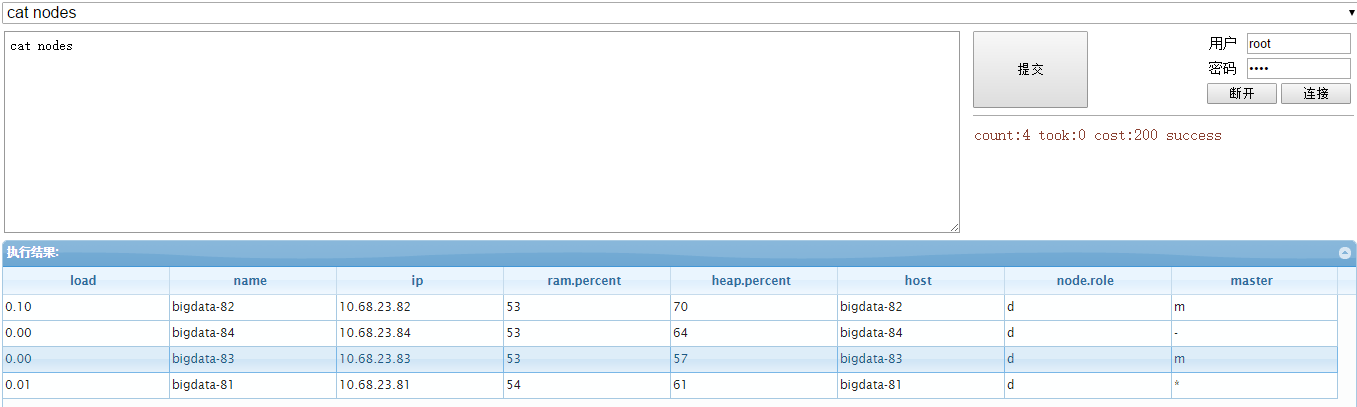
### 10.4查看主节点

语法：cat master

图：

### 10.5查看集群节点

语法：cat nodes

图：

说明：

load: 系统负载指标，cpu信息

name: 节点信息hostname。

ip: 节点ip信息。

ram.percent: 系统剩余内存

heap.percent: 集群使用内存

host:hostname信息。

node.role: d 表示数据节点

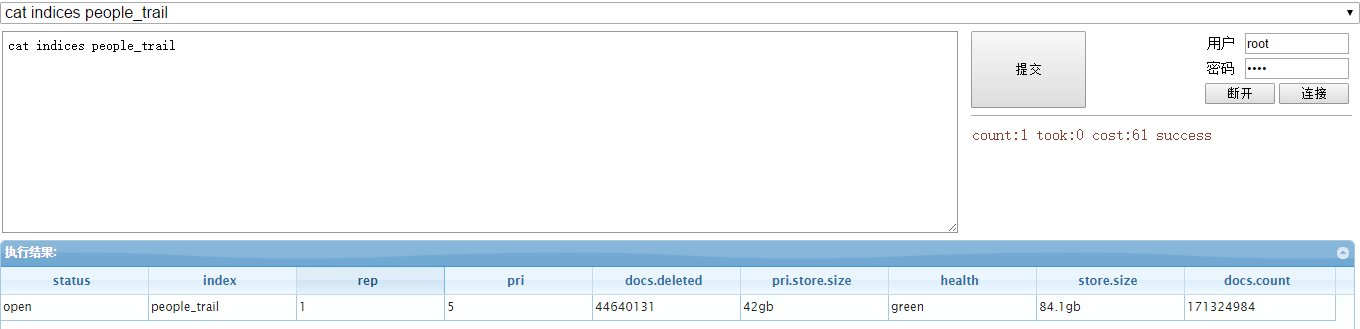
master: \* 表示主节点

例: cat nodes with ip,sm (sm= s例ment memory查看节点IP及数据段使用的内存)

### 10.6查看索引状态

语法：cat indices

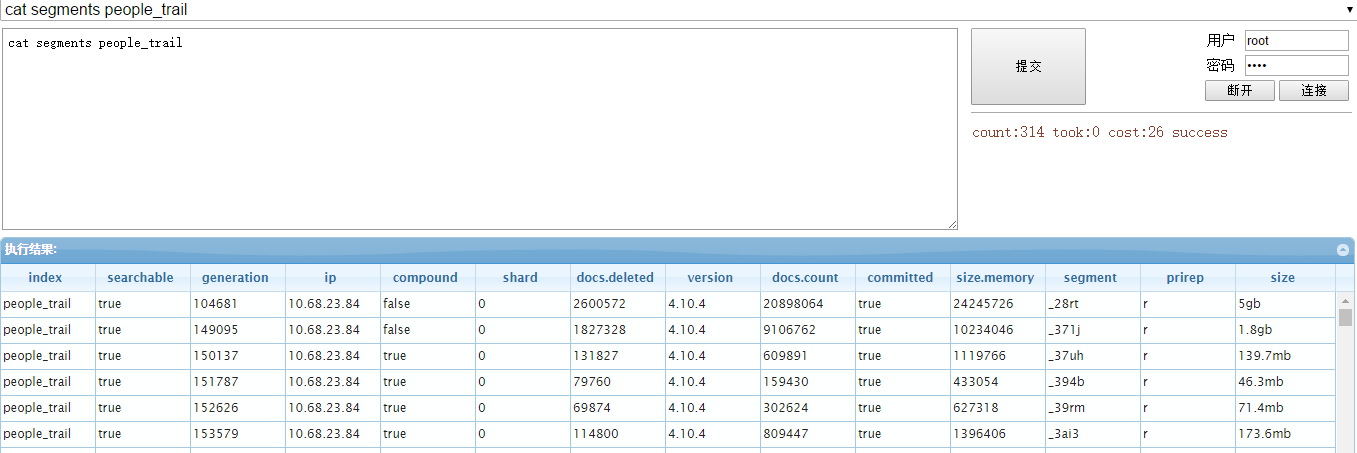
查看索引状态，如主分片数和副本数、文档数、删除文档数，大小等等。

图：

### 10.7查看分段信息

语法：cat segments /cat segments index\_name

注意：没有with，esql2有with.

图：

说明：

index:索引名称。

searchable:是否可查询。true=是。:

ip:分段所在ip。

shard:分段所在分片位置。

docs.deleted:代表打了删除标记的记录数，在表优化时会才会真正删除掉。

version:lucene版本（索引数据基础版本）。

docs.count:存储的docs记录数。

committed:是否提交。true=是。

size.memory:代表段在内存中的缓存。

s例ment: 代表在硬盘上存储的一个段(一组文件)。

prirep: 主副分片标志。p=主，r=副。

size:代表段(文件)的大小。

### 10.8查看索引统计

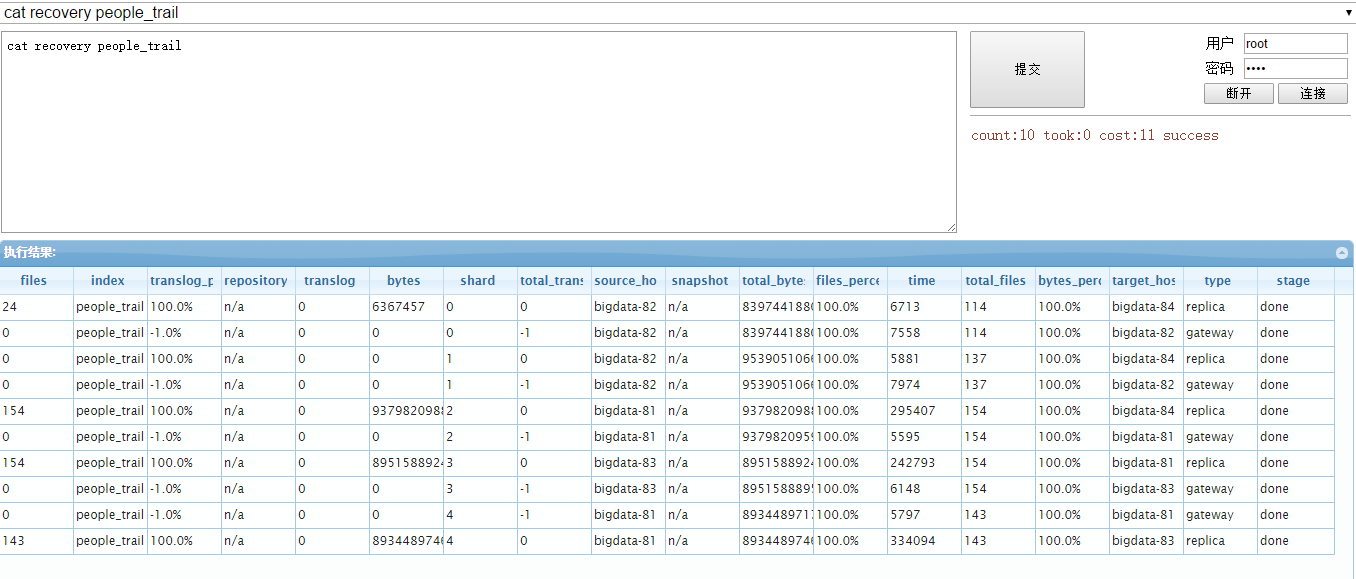
语法：cat count/cat count index\_name

查看整个集群文档数或单个索引

### 10.9查看恢复情况

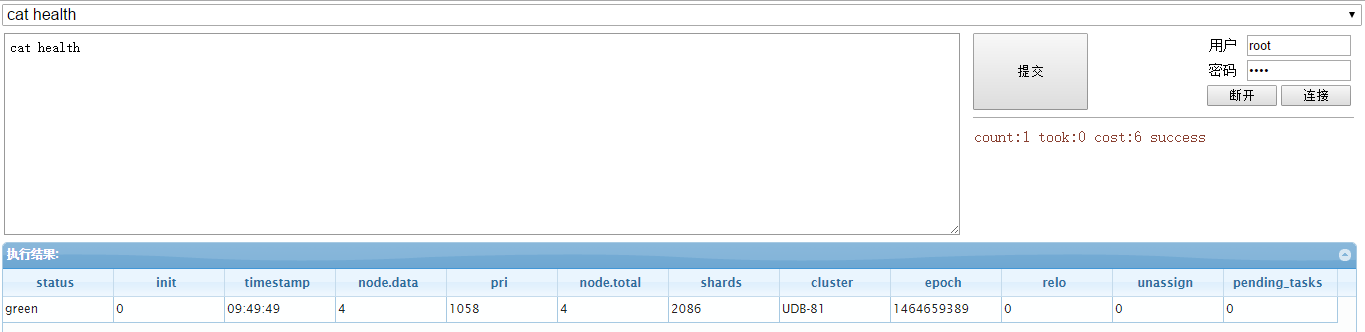
语法：cat recovery/cat recovery index

显示正在进行的和以前完成的索引分片恢复状态。恢复事件随时可能发生，在集群中一个索引分片转移到其它节点上。可能发生在快照恢复、改变副本级别、节点故障或者启动节点。

图：

### 10.10查看集群状态

语法：cat health

图：

### 10.11查看执行中的任务

语法：cat pending\_tasks

### 10.12 查看别名

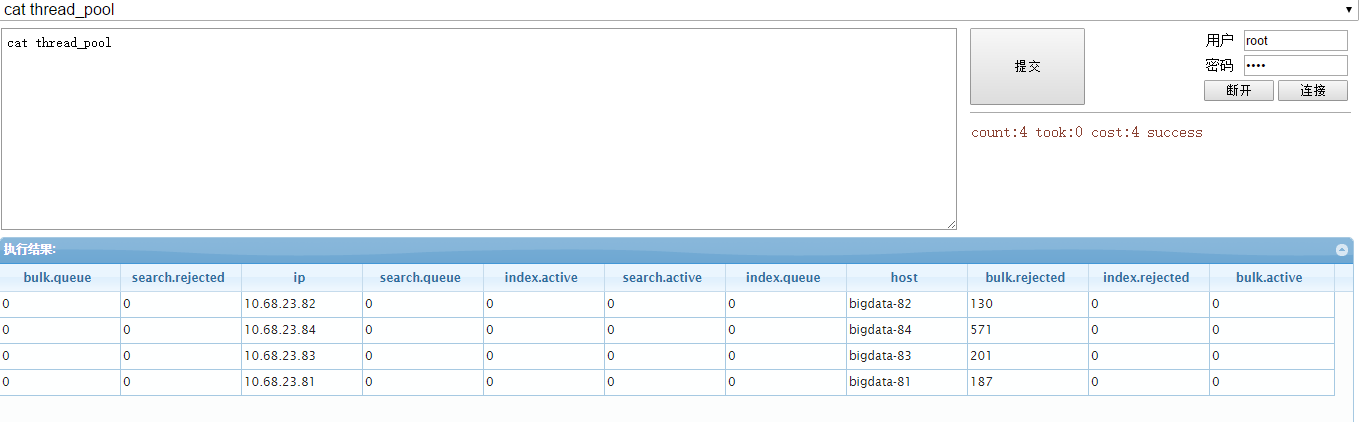
语法：cat aliases

查看当前配置的索引索引的别名，包括过滤和路由相关信息

### 10.13显示线程情况

语法：cat thread\_pool

显示线程池状态

图：****

### 10.14查看插件

语法：cat plugins

图：

显示集群中各种插件

### 10.15查看cache情况

语法：cat fielddata

Fielddata 是执行order by 和 group by 时所占用的内存。

图：

### 10.16查看磁盘空间

语法：cat  allocation

每个数据节点分配了多少分片和使用了多少磁盘空间

图：

此语法对应开源es查看集群语法：

#  curl 10.68.23.81:9200/\_cat

/\_cat/allocation

/\_cat/shards

/\_cat/shards/{index}

/\_cat/master

/\_cat/nodes

/\_cat/indices/{index}

/\_cat/s例ments

/\_cat/s例ments/{index}

/\_cat/count

/\_cat/count/{index}

/\_cat/recovery

/\_cat/recovery/{index}

/\_cat/health

/\_cat/pending\_tasks

/\_cat/aliases

/\_cat/aliases/{alias}

/\_cat/thread\_pool

/\_cat/plugins

/\_cat/fielddata

/\_cat/fielddata/{fields}

### 10.17刷新

语法： flush index\_name

插入值后需要执行刷新语法才能看到结果，否则要等集群默认刷新时才能看到，时间比较长。

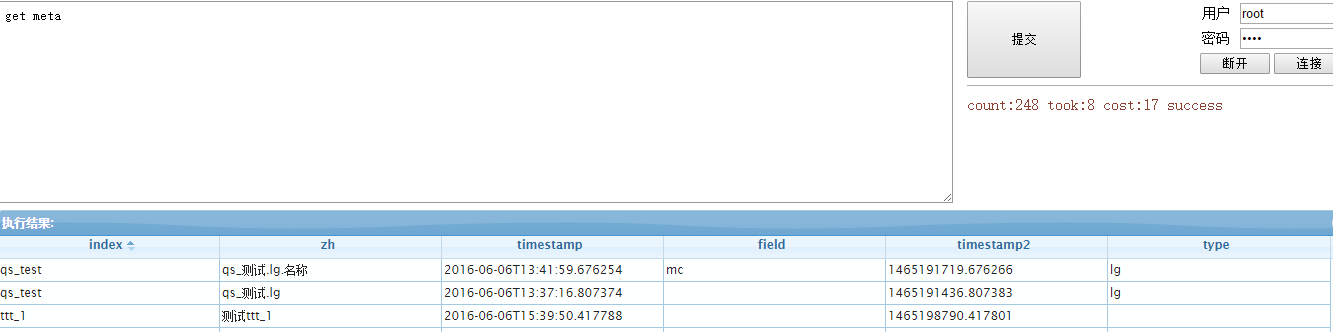
默认设置为：每 30 分钟主动进行一次flush，或者当 translog 文件大小大于 2048MB时，主动进行一次flush。

### 10.18 查看元数据

语法： GET MATE

此语法查看sys\_meta索引下的数据，sys\_meta索引为系统存储元数据信息索引。现在需要手动创建此索引。

例：



### 10.19 控制台使用

**10.19.1导出索引结构**

../bin/下执行esql脚本

语法：./esql –u root –p toor –f index\_name –d 自定义名称.sql

-u：表示user

-p: 表示password

-f: 表示过滤条件filter

-d: 表示dump，如-d后的路径省略，默认存储路径在../esql3.0.11-1/websql下

-h: 表示查看帮助

-e: 表示要执行sql语句

–h：查看帮助

例：

./esql -u root -p toor -f zhy -d /home/zhy.sql；

./esql -u root -p toor -f people\_test\* -d /home/people\_test.sql；

./esql –h

注：索引名称支持通配符；

**10.19.2 导入索引结构**

**例：**

默认将sql文件放在当前目录下

./esql -u root -p toor < people\_trail.sql

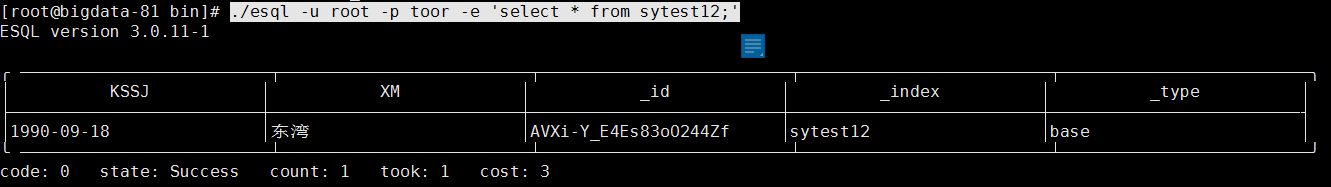
./esql -u root -p toor < /Esql/esql3.0.11-3/websql/people\_trail.sql

**10.19.3 执行sql**

可以在控制台执行sql语句，与web界面查询功能相同。

例：

./esql -u root -p toor -e 'select \* from sytest12;'



./esql -u root -p toor -e 'desc sytest12;’

可以显示建表语句。

### 10.20清除缓存

语法：clear cache /clear cache with index\_name

# 红名单管理

针对敏感人群数据的高要求，将敏感人群数据放置在红名单中，一般级别的用户在数据搜索和查询时，系统会自动过滤掉相应信息，达到对敏感数据的保护目的。

操作过程：首先使用红名单功能的用户需要root为用户授权时指定红名单，具体用法可参考8.2章。同时参考10.5章

### 11.1创建红名单

Create table redkeys.base(secret\_key string) with 10,1

### 11.2写入数据到红名单

Insert into redkeys.base (secret\_key) values (‘证件号码或姓名或其他内容’)

Flush redkeys 刷新红名单

例:

Insert into redkeys.base (secret\_key) values ('361028196210315833')

Flush redkeys

注意：单引号

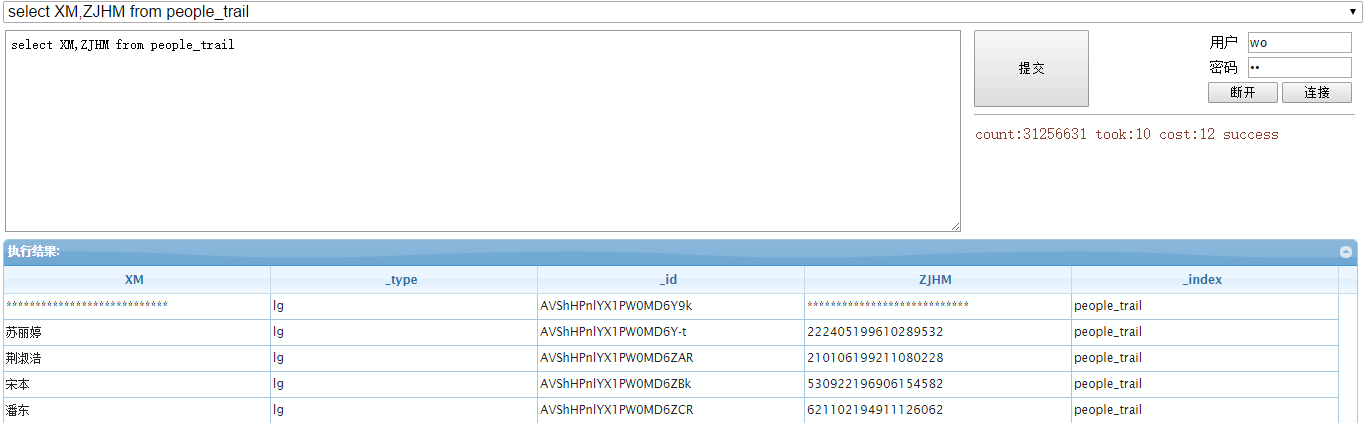
### 11.3重启esql:

语法： reload

### 11.4检验是否生效

如查询：select zjhm from account where \_all = '361028196210315833'

返回结果： 红名单中的数据内容显示\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



### 11.5查看用户红名单权限

查看test用户目前已有的权限

show grant with test

如果已经有所有表的select权限，则清除之

revoke select on \*.\* from test@'%'

设置非redkeys的select权限

grant select on qs\_test.\* to test@'%'

设置附加redkeys的select权限

grant select on qs\_test.\* to test@'%' with redkeys;

查看设置后的结果

show grant with test

 例： 红名单授权用户test过程

### 11.6检验红名单数据

用test帐号查询写入红名单数据

select \* from qs\_test;

（插入数据语句参考10.2语法，插入红名单数据后，执行reload，再登录授权用户检查是否生效）

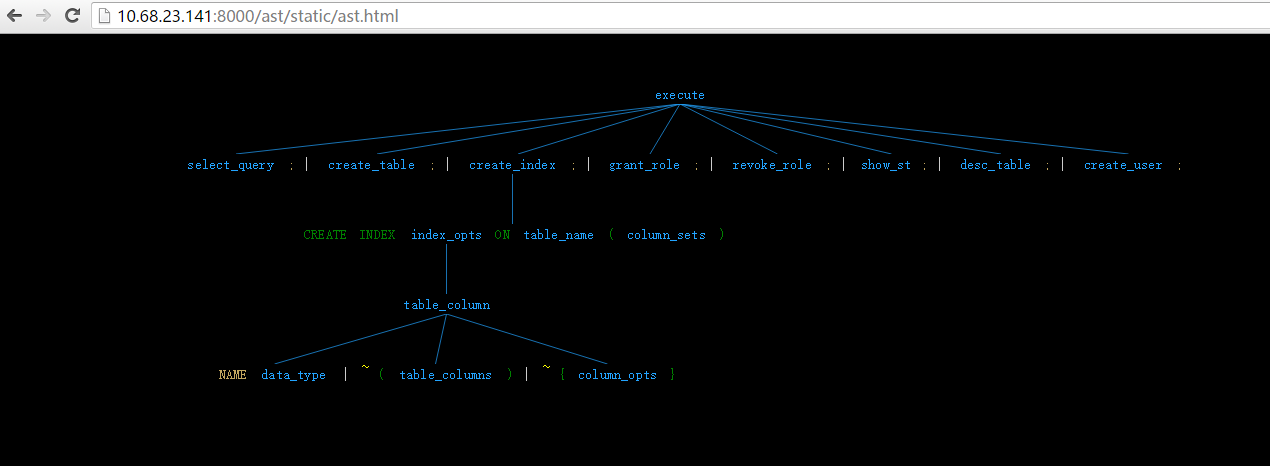
# 可视化语法树

通过可视化语法树可以清晰看到语法解析，便于语法分析。因目前3版本很多语法还没有完全支持，语法树中有些语法是不可执行的。

访问地址：esql访问地址/ast

例：10.68.23.187:8000 esql地址

10.68.23.187:8000/ast 语法树地址



# 其他功能

支持浏览器IE6,IE8，chrome；

WEB界面捷键使用：alt+l: 执行选中部分或当前行； alt+k: 执行选中部分或多行语句