密级：机密（版权所有，翻版必究）

分册名称：　　　　　　　　　　　　　　　　　　　　　第　册/共　册

大数据高端人才培养计划

Hive 复杂数据类型使用

沈阳昊宸科技有限公司

2017 年 3 月 18 日

变更履历

修改编号	版本	修改内容	修改人	修改日期
			dulm	

目　　录

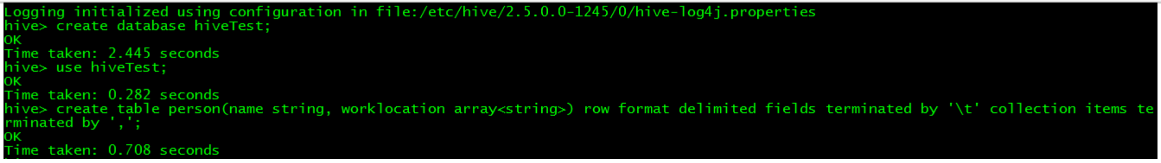
1 Hive 复杂数据类型使用

Hive 复杂数据类型有 Array、Map、Struct 三种。下面介绍这三种复杂数据类型的使用方法。

1.1 Array 类型

1.1.1 创建数据库表，以 Array 做为数据类型

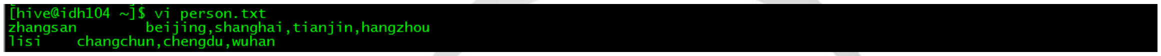
```
Create table person(name string, worklocation array<string> ) row format delimited fields terminated by ‘\t’ collection items terminated by ‘,’;
```



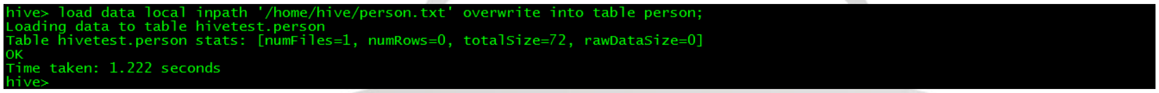
```
Logging initialized using configuration in file:/etc/hive/2.5.0.0-1245/0/hive-log4j.properties
hive> create database hiveTest;
OK
Time taken: 2.445 seconds
hive> use hiveTest;
OK
Time taken: 0.282 seconds
hive> create table person(name string, worklocation array<string>) row format delimited fields terminated by '\t' collection items te
rminated by ',';
OK
Time taken: 0.708 seconds
```

```
vi person.txt
zhangsan    beijing,shanghai,tianjin,hangzhou
lisi    changchu,chengdu,wuhan

load data local inpath '/home/hive/person.txt'
OVERWRITE INTO TABLE person;
```

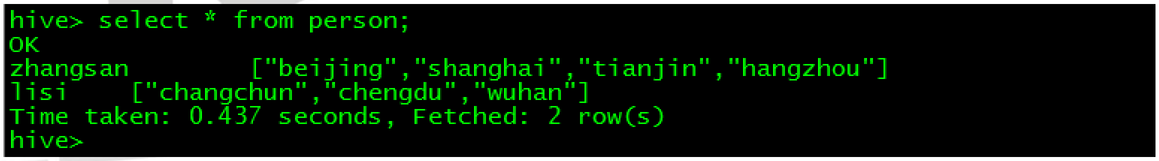


```
[hive@idh104 ~]$ vi person.txt
zhangsan        beijing,shanghai,tianjin,hangzhou
lisi    changchun,chengdu,wuhan
```



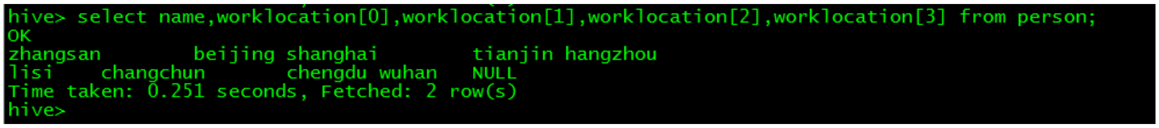
```
hive> load data local inpath '/home/hive/person.txt' overwrite into table person;
Loading data to table hivetest.person
Table hivetest.person stats: [numFiles=1, numRows=0, totalSize=72, rawDataSize=0]
OK
Time taken: 1.222 seconds
hive>
```

1.1.2 查询数据库表



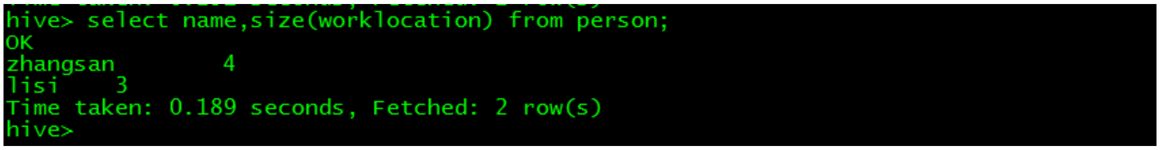
```
hive> select * from person;
OK
zhangsan        ["beijing","shanghai","tianjin","hangzhou"]
lisi    ["changchun","chengdu","wuhan"]
Time taken: 0.437 seconds, Fetched: 2 row(s)
hive>
```

查询 person 表 array 数据类型字段指定列。



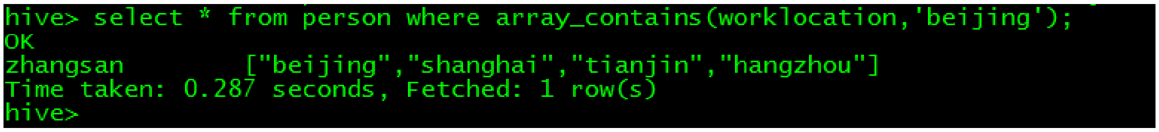
```
hive> select name,worklocation[0],worklocation[1],worklocation[2],worklocation[3] from person;
OK
zhangsan        beijing shanghai        tianjin hangzhou
lisi    changchun       chengdu wuhan   NULL
Time taken: 0.251 seconds, Fetched: 2 row(s)
hive>
```

查询 array 数据类型字段的长度。



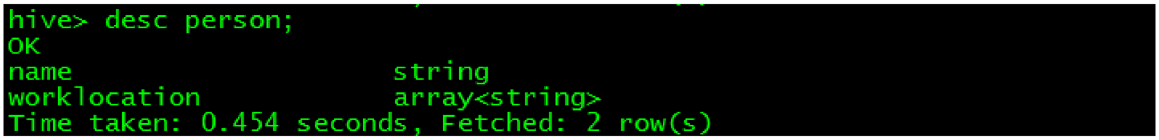
```
hive> select name,size(worklocation) from person;
OK
zhangsan        4
lisi    3
Time taken: 0.189 seconds, Fetched: 2 row(s)
hive>
```

查询 array 数据类型字段指定列的一行数据。



```
hive> select * from person where array_contains(worklocation,'beijing');
OK
zhangsan        ["beijing","shanghai","tianjin","hangzhou"]
Time taken: 0.287 seconds, Fetched: 1 row(s)
hive>
```

查看表结构

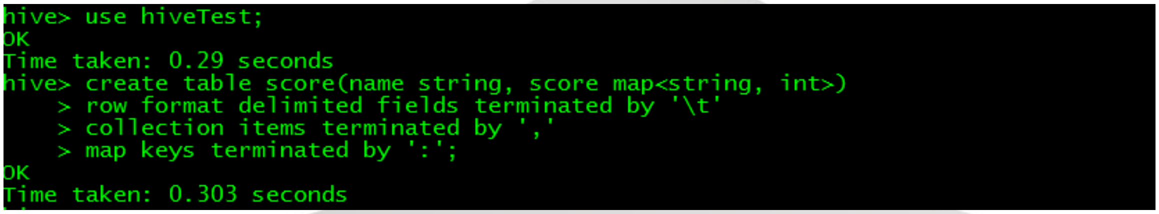


```
hive> desc person;
OK
name                    string
worklocation            array<string>
Time taken: 0.454 seconds, Fetched: 2 row(s)
```

2 Map 类型

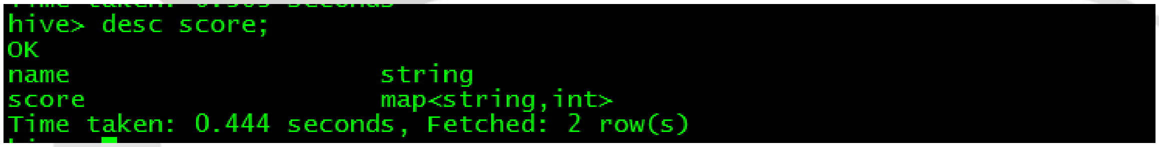
2.1 创建数据库表，以 Map 做为数据类型

create table score(name string, score map<string, int>) row format delimited fields terminated by ‘\t’ collection items terminated by ‘,’ map keys terminated by ‘:’;



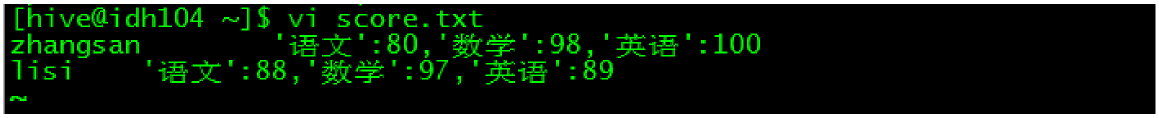
```
hive> use hiveTest;
OK
Time taken: 0.29 seconds
hive> create table score(name string, score map<string, int>)
    > row format delimited fields terminated by '\t'
    > collection items terminated by ','
    > map keys terminated by ':';
OK
Time taken: 0.303 seconds
```

查看表结构

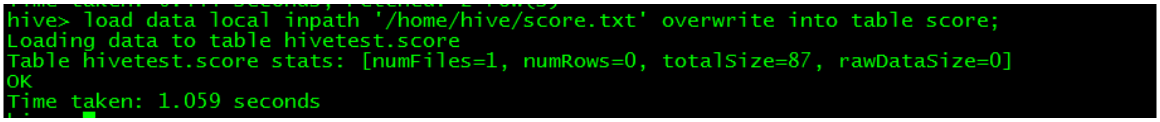


```
hive> desc score;
OK
name                    string
score                   map<string,int>
Time taken: 0.444 seconds, Fetched: 2 row(s)
```

在本地创建 score.txt 文件，并将文件内容导入到 score 表中。



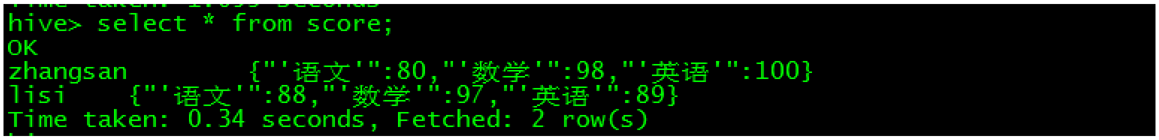
```
[hive@idh104 ~]$ vi score.txt
zhangsan        '语文':80,'数学':98,'英语':100
lisi    '语文':88,'数学':97,'英语':89
~
```



```
hive> load data local inpath '/home/hive/score.txt' overwrite into table score;
Loading data to table hivetest.score
Table hivetest.score stats: [numFiles=1, numRows=0, totalSize=87, rawDataSize=0]
OK
Time taken: 1.059 seconds
```

2.2 查询数据库表

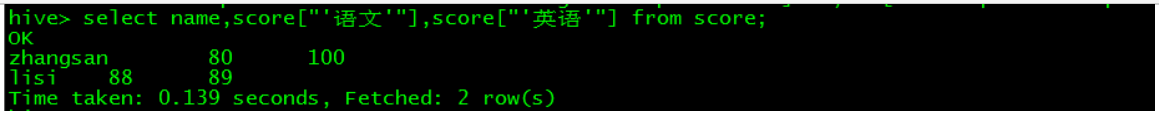
查询 score 表的所有数据。



```
hive> select * from score;
OK
zhangsan        {"'语文'":80,"'数学'":98,"'英语'":100}
lisi    {"'语文'":88,"'数学'":97,"'英语'":89}
Time taken: 0.34 seconds, Fetched: 2 row(s)
```

查询所有的语文，英语成绩。

```
select name,score["'语文'"],score["'英语'"] from score;
```

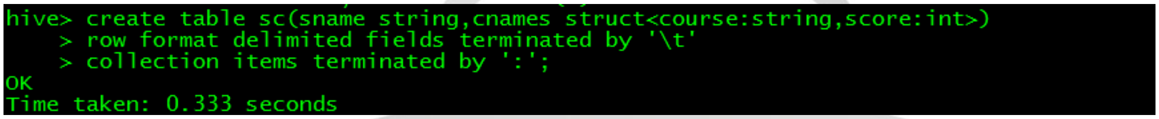


```
hive> select name,score["'语文'"],score["'英语'"] from score;
OK
zhangsan        80      100
lisi    88      89
Time taken: 0.139 seconds, Fetched: 2 row(s)
```

3 Struct 类型

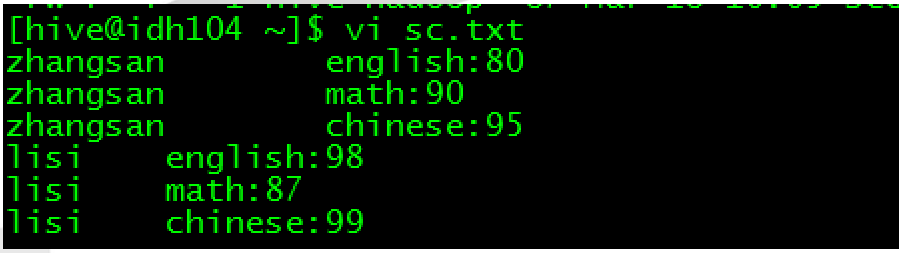
3.1 创建数据库表，以 Struct 作为数据类型

create table sc(sname string,cnames struct<course:string,score:int>) row format delimited fields terminated by '\t' collection items terminated by ':';

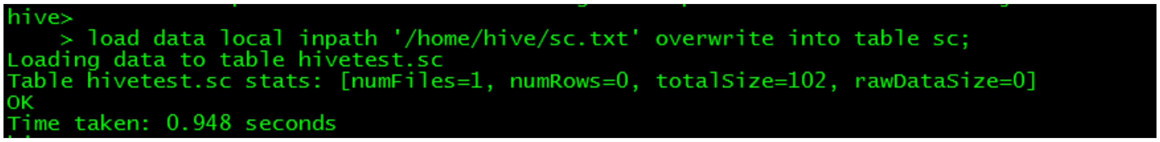


```
hive> create table sc(sname string,cnames struct<course:string,score:int>)
    > row format delimited fields terminated by '\t'
    > collection items terminated by ':';
OK
Time taken: 0.333 seconds
```

本地新建文件 sc.txt，并将文件导入到 sc 表中。



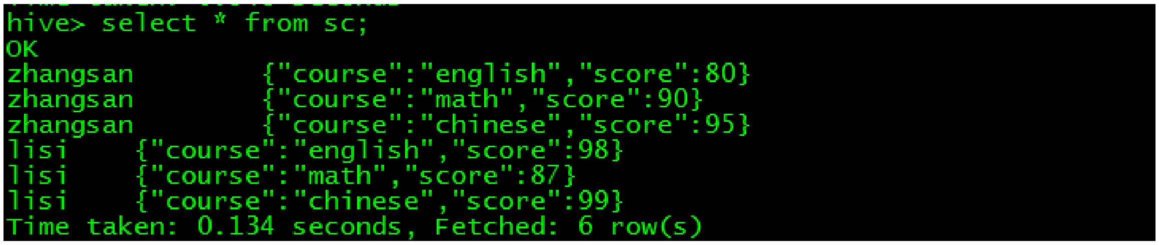
```
[hive@idh104 ~]$ vi sc.txt
zhangsan        english:80
zhangsan        math:90
zhangsan        chinese:95
lisi    english:98
lisi    math:87
lisi    chinese:99
```



```
hive>
    > load data local inpath '/home/hive/sc.txt' overwrite into table sc;
Loading data to table hivetest.sc
Table hivetest.sc stats: [numFiles=1, numRows=0, totalSize=102, rawDataSize=0]
OK
Time taken: 0.948 seconds
```

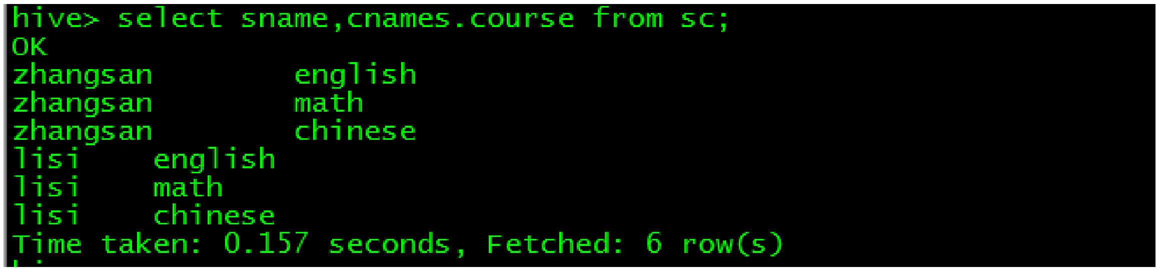
3.2 查询数据库表

查询 sc 表的所有数据。



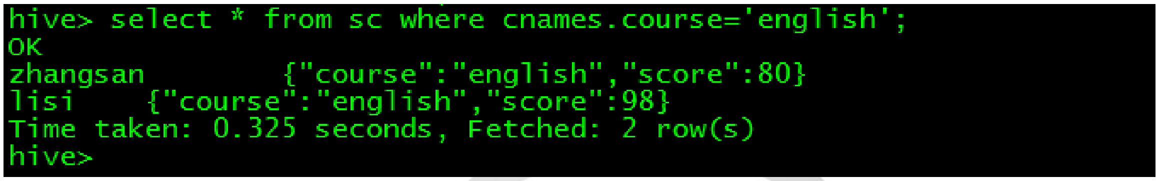
```
hive> select * from sc;
OK
zhangsan        {"course":"english","score":80}
zhangsan        {"course":"math","score":90}
zhangsan        {"course":"chinese","score":95}
lisi    {"course":"english","score":98}
lisi    {"course":"math","score":87}
lisi    {"course":"chinese","score":99}
Time taken: 0.134 seconds, Fetched: 6 row(s)
```

查询 SC 表的学生名称和选课名称。



```
hive> select sname,cnames.course from sc;
OK
zhangsan        english
zhangsan        math
zhangsan        chinese
lisi    english
lisi    math
lisi    chinese
Time taken: 0.157 seconds, Fetched: 6 row(s)
```

查询 SC 表所有学生英语课成绩。



```
hive> select * from sc where cnames.course='english';
OK
zhangsan        {"course":"english","score":80}
lisi    {"course":"english","score":98}
Time taken: 0.325 seconds, Fetched: 2 row(s)
hive>
```