大数据库系统

6.4 Hive的基本操作

6.4 Hive的基本操作

- 本节内容

  6.4.1 Hive中数据库的操作

  6.4.2 Hive中表的基本操作

HIVE中表的创建与使用

- 主要内容

创建表

删除表

清空表

查看表的信息

总结

（1）创建表

6.4.2 Hive中表的基本操作

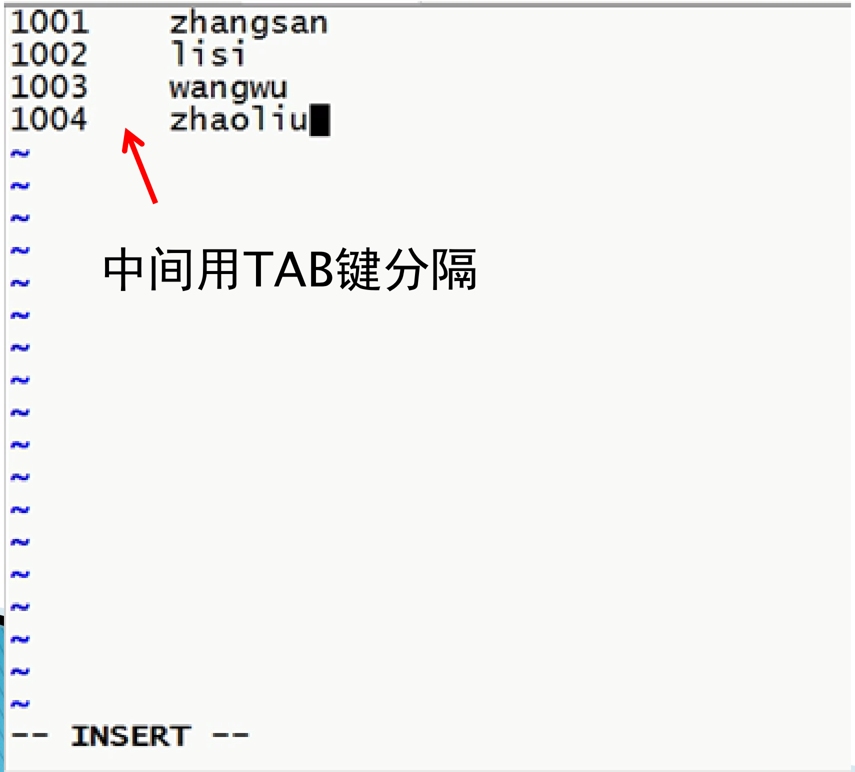
- **创建表命令格式:**

```
CREATE  [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
  [(col_name data_type [COMMENT col_comment], ...
   [constraint_specification])]
  [PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]
   [ROW FORMAT row_format]
   [STORED AS file_format]
  [LOCATION hdfs_path]
  [AS select_statement];
```

6.4.2 Hive中表的基本操作

◆ 数据准备

假定在本地/opt/modules/hive-0.13.1-bin/目录下建立一个student.txt文件，输入以下内容（学号string，姓名string两个字段）作为我们表的数据



```
1001	zhangsan
1002	lisi
1003	wangwu
1004	zhaoliu
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

中间用TAB键分隔

（1）创建表

6.4.2 Hive中表的基本操作

- 创建表tmp3_table：

```
create table if not exists tmp3_table(
number string,
name string
) row format delimited fields terminated by '\t';
stored as textfile;
load data local inpath '/opt/modules/hive-0.13.1-bin/student.txt' into table
    tmp3_table;
```

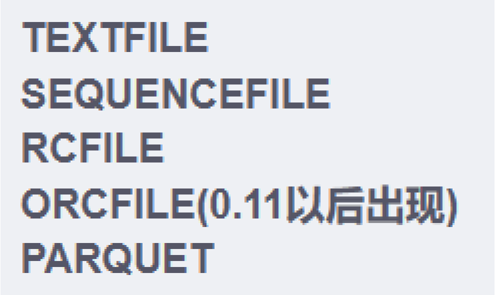
表的两个字段

- row format delimited fields terminated by '\t';

  表示字段与字段之间以“tab”分隔，分隔符由导入的数据文件内容而定；

- stored as textfile;

  表示数据存储的格式为文本文件，如果为“textfile”，这句指令可以不写；



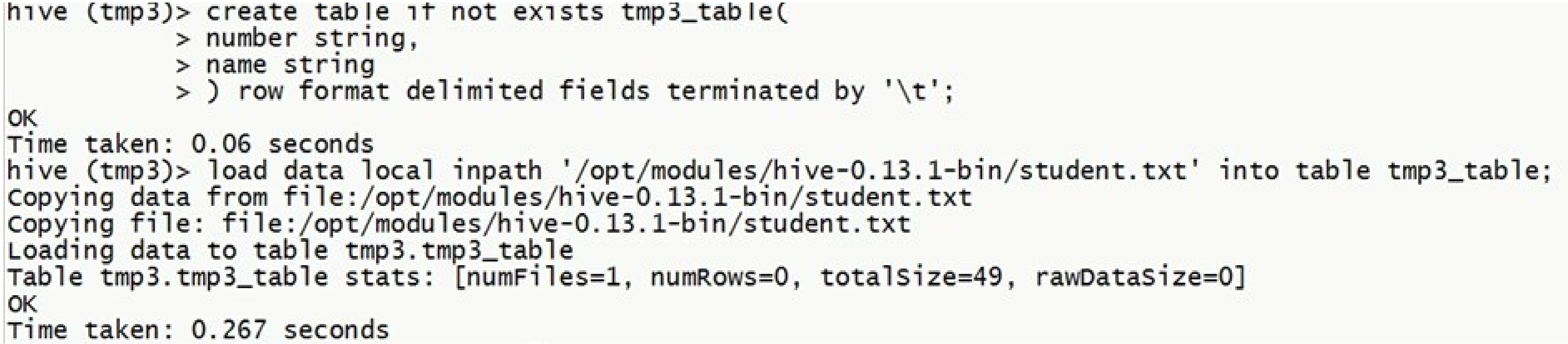
TEXTFILE
SEQUENCEFILE
RCFILE
ORCFILE(0.11以后出现)
PARQUET

- load data local inpath '/opt/modules/hive-0.13.1-bin/student.txt' into table tmp3_table;

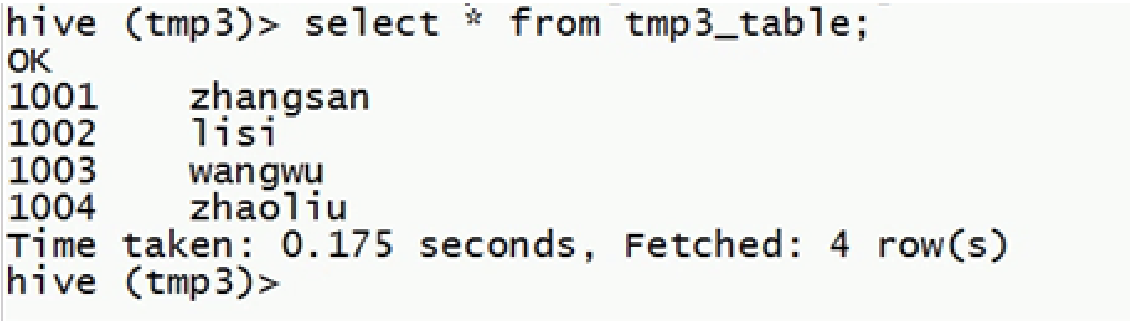
  表示数据存储的路径和文件名，注意：加local表示本地路径，不加local为hdfs路径；

（1）创建表

6.4.2 Hive中表的基本操作



```
hive (tmp3)> create table if not exists tmp3_table(
           > number string,
           > name string
           > ) row format delimited fields terminated by '\t';
OK
Time taken: 0.06 seconds
hive (tmp3)> load data local inpath '/opt/modules/hive-0.13.1-bin/student.txt' into table tmp3_table;
Copying data from file:/opt/modules/hive-0.13.1-bin/student.txt
Copying file: file:/opt/modules/hive-0.13.1-bin/student.txt
Loading data to table tmp3.tmp3_table
Table tmp3.tmp3_table stats: [numFiles=1, numRows=0, totalSize=49, rawDataSize=0]
OK
Time taken: 0.267 seconds
```



```
hive (tmp3)> select * from tmp3_table;
OK
1001    zhangsan
1002    lisi
1003    wangwu
1004    zhaoliu
Time taken: 0.175 seconds, Fetched: 4 row(s)
hive (tmp3)>
```

6.4.1 Hive中数据库的操作

◆ 注意:

- ✓ load data inpath '/student.txt' into table tmp1_table;

  意为hdfs导入， 直接将文件移动到了表的目录下，即“hive/warehouse/数据库名/表名/”下

- ✓ load data local inpath '/opt/modules/hive-0.13.1-bin/student.txt' into table tmp3_table;

  意为本地导入，将本地文件复制到了表的目录下

6.4.2 Hive中表的基本操作

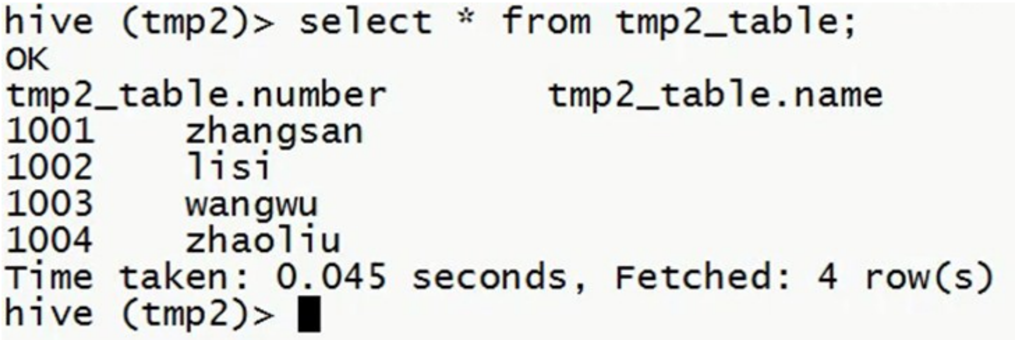
```
load data local inpath '/opt/modules/hive-0.13.1-bin/student.txt' into table tmp3_table;
load data local inpath '/opt/modules/hive-0.13.1-bin/student.txt' overwrite into table tmp3_table;
```

没加overwrite表示对表追加数据，每次追加数据都会把数据添加到tmp3_table目录下

如果使用overwrite则表示覆盖表内数据，此时会清空tmp3_table目录下的数据，将新数据添加到tmp3_table目录下

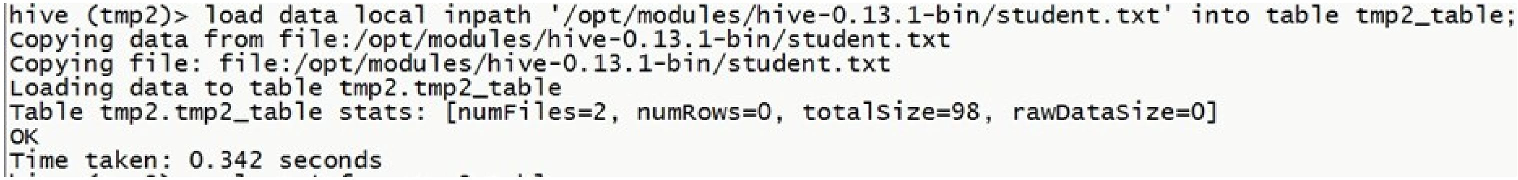
例：在tmp2_table表中追加数据

1、查询tmp2_table表的内容，发现已经存在数据



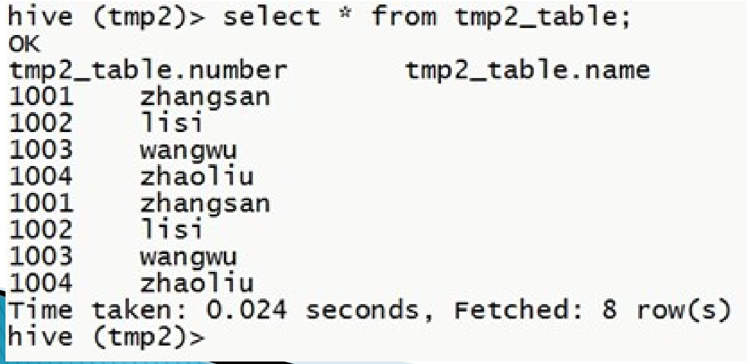
```
hive (tmp2)> select * from tmp2_table;
OK
tmp2_table.number       tmp2_table.name
1001    zhangsan
1002    lisi
1003    wangwu
1004    zhaoliu
Time taken: 0.045 seconds, Fetched: 4 row(s)
hive (tmp2)>
```

2、向tmp2_table表加载student.txt文件中的数据



```
hive (tmp2)> load data local inpath '/opt/modules/hive-0.13.1-bin/student.txt' into table tmp2_table;
Copying data from file:/opt/modules/hive-0.13.1-bin/student.txt
Copying file: file:/opt/modules/hive-0.13.1-bin/student.txt
Loading data to table tmp2.tmp2_table
Table tmp2.tmp2_table stats: [numFiles=2, numRows=0, totalSize=98, rawDataSize=0]
OK
Time taken: 0.342 seconds
```

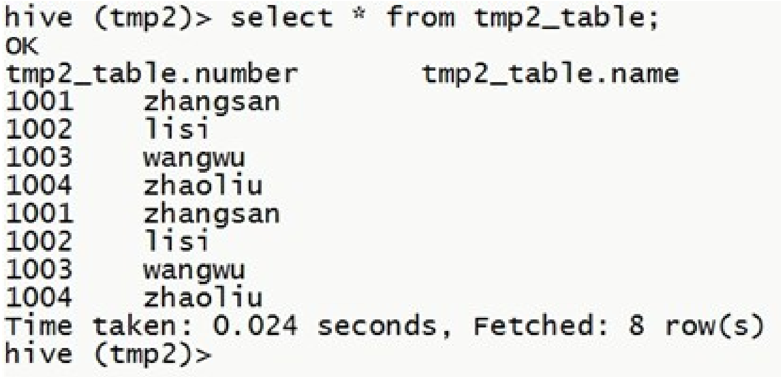
3、查询tmp2_table表的内容



```
hive (tmp2)> select * from tmp2_table;
OK
tmp2_table.number       tmp2_table.name
1001    zhangsan
1002    lisi
1003    wangwu
1004    zhaoliu
1001    zhangsan
1002    lisi
1003    wangwu
1004    zhaoliu
Time taken: 0.024 seconds, Fetched: 8 row(s)
hive (tmp2)>
```

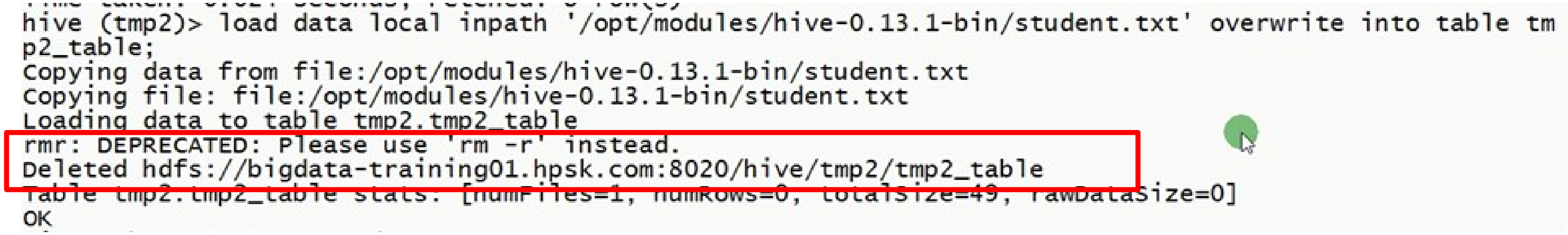
可以发现tmp2_table表中有两份数据

1、查询刚刚tmp2_table表的内容，有数据



```
hive (tmp2)> select * from tmp2_table;
OK
tmp2_table.number       tmp2_table.name
1001    zhangsan
1002    lisi
1003    wangwu
1004    zhaoliu
1001    zhangsan
1002    lisi
1003    wangwu
1004    zhaoliu
Time taken: 0.024 seconds, Fetched: 8 row(s)
hive (tmp2)>
```

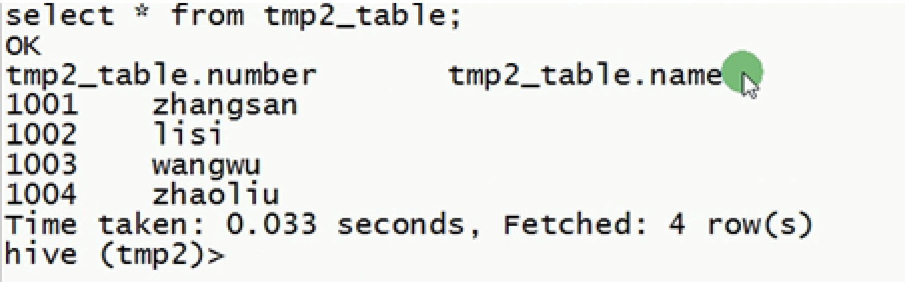
2、使用load data …overwrite into覆盖数据



```
Time taken: 0.024 seconds, Fetched: 8 row(s)
hive (tmp2)> load data local inpath '/opt/modules/hive-0.13.1-bin/student.txt' overwrite into table tm
p2_table;
Copying data from file:/opt/modules/hive-0.13.1-bin/student.txt
Copying file: file:/opt/modules/hive-0.13.1-bin/student.txt
Loading data to table tmp2.tmp2_table
rmr: DEPRECATED: Please use 'rm -r' instead.
Deleted hdfs://bigdata-training01.hpsk.com:8020/hive/tmp2/tmp2_table
Table tmp2.tmp2_table stats: [numFiles=1, numRows=0, totalSize=49, rawDataSize=0]
OK
```

观察提示可以发现，实际上是一个目录删除过程，即使用linux命令删除原表目录，再重新创建目录，放入新的数据文件

3、再查询该表，发现只有新的数据



```
select * from tmp2_table;
OK
tmp2_table.number       tmp2_table.name
1001    zhangsan
1002    lisi
1003    wangwu
1004    zhaoliu
Time taken: 0.033 seconds, Fetched: 4 row(s)
hive (tmp2)>
```

（1）创建表

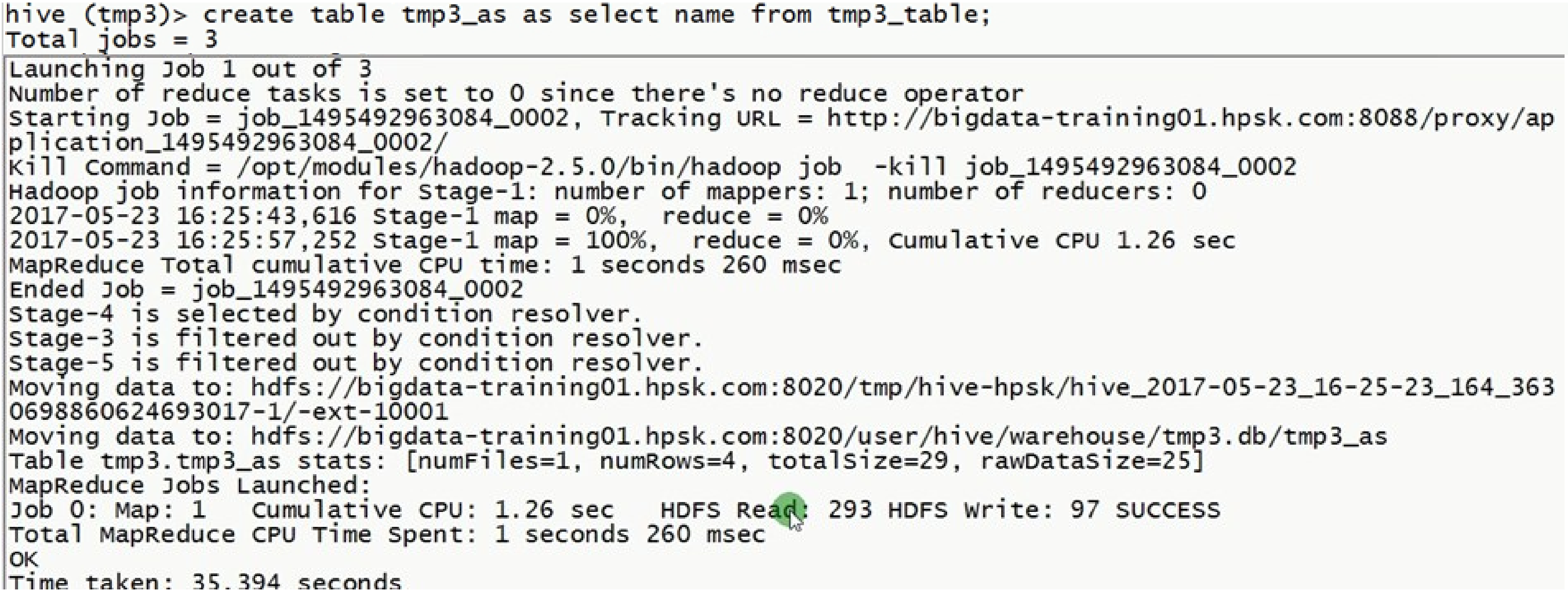
6.4.2 Hive中表的基本操作

- **添加as子查询方式创建表:**

可以通过as，将某个select语句的查询结果保存为一张表

例如：create table tmp3_as as select name from tmp3_table;

```
CREATE  [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
 [(col_name data_type [COMMENT col_comment], ...
  [constraint_specification])]
 [PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]
  [ROW FORMAT row_format]
  [STORED AS file_format]
 [LOCATION hdfs_path]
 [AS select_statement];
```



```
hive (tmp3)> create table tmp3_as as select name from tmp3_table;
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1495492963084_0002, Tracking URL = http://bigdata-training01.hpsk.com:8088/proxy/ap
plication_1495492963084_0002/
Kill Command = /opt/modules/hadoop-2.5.0/bin/hadoop job  -kill job_1495492963084_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-05-23 16:25:43,616 Stage-1 map = 0%,  reduce = 0%
2017-05-23 16:25:57,252 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.26 sec
MapReduce Total cumulative CPU time: 1 seconds 260 msec
Ended Job = job_1495492963084_0002
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://bigdata-training01.hpsk.com:8020/tmp/hive-hpsk/hive_2017-05-23_16-25-23_164_363
0698860624693017-1/-ext-10001
Moving data to: hdfs://bigdata-training01.hpsk.com:8020/user/hive/warehouse/tmp3.db/tmp3_as
Table tmp3.tmp3_as stats: [numFiles=1, numRows=4, totalSize=29, rawDataSize=25]
MapReduce Jobs Launched:
Job 0: Map: 1   Cumulative CPU: 1.26 sec   HDFS Read: 293 HDFS Write: 97 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 260 msec
OK
Time taken: 35.394 seconds
```

实际上hive通过创建MapReduce任务，从tmp3_table中取数据，再创建一个tmp3_as表，将数据存进去

（1）创建表

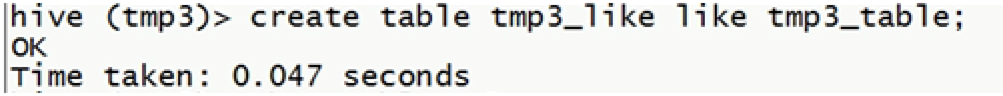
6.4.2 Hive中表的基本操作

- 通过like创建表，命令写法：

```
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS]
   [db_name.]table_name
        LIKE existing_table_or_view_name
  [LOCATION hdfs_path];
```

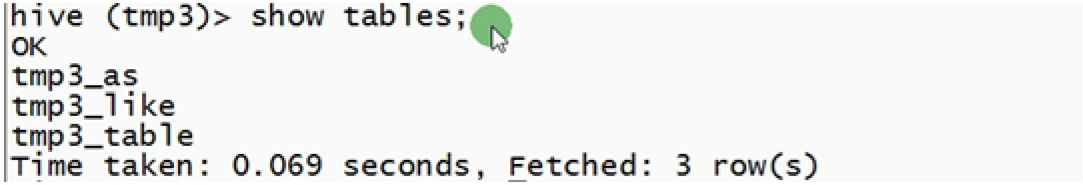
例： create table tmp3_like like tmp3_table;

1、通过like语句创建tmp3_like表:



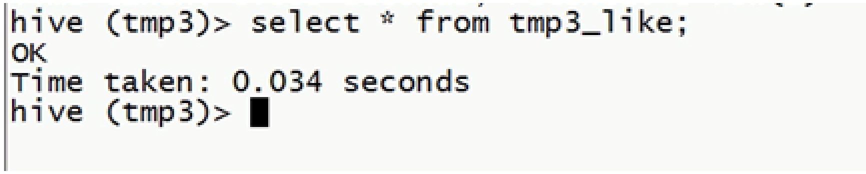
```
hive (tmp3)> create table tmp3_like like tmp3_table;
OK
Time taken: 0.047 seconds
```

2、使用show tables可以看到tmp3_like表已成功创建:



```
hive (tmp3)> show tables;
OK
tmp3_as
tmp3_like
tmp3_table
Time taken: 0.069 seconds, Fetched: 3 row(s)
```

3、使用select语句查询tmp3_like表中的信息:



```
hive (tmp3)> select * from tmp3_like;
OK
Time taken: 0.034 seconds
hive (tmp3)> █
```

结果我们发现tmp3_like表里什么都没有，tmp3_table明明有数据，为什么使用like语句创建tmp3_like表却是空的?

6.4.2 Hive中表的基本操作

- as与like的区别

as：将子查询的结果，包括数据和表结构放入的新的表中

create table tmp3_as as select name from tmp3_table;

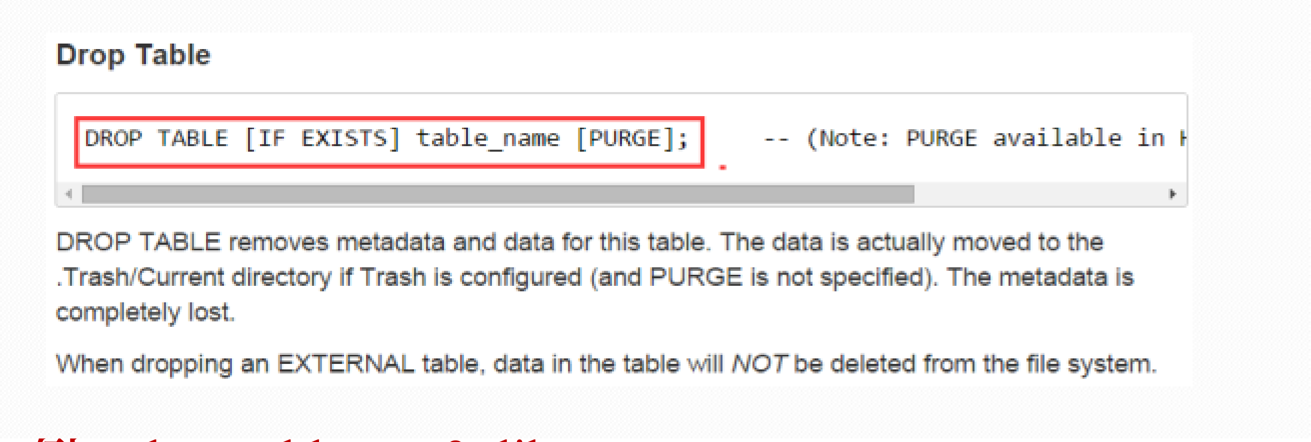
like:只是复制了表结构，即如果我们想创建一张表，但又不想要原表的数据时使用like

create table tmp3_like like tmp3_table;

（2）删除表

6.4.2 Hive中表的基本操作

◆ 删除表命令： drop table



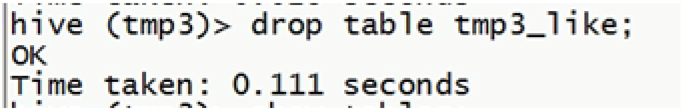
Drop Table

```
DROP TABLE [IF EXISTS] table_name [PURGE];     -- (Note: PURGE available in H
```

DROP TABLE removes metadata and data for this table. The data is actually moved to the .Trash/Current directory if Trash is configured (and PURGE is not specified). The metadata is completely lost.

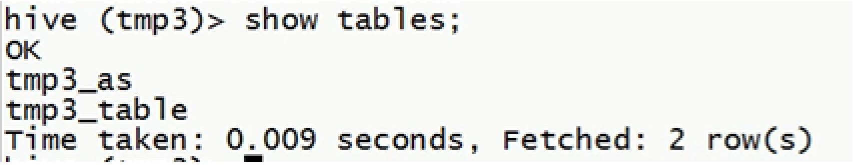
When dropping an EXTERNAL table, data in the table will *NOT* be deleted from the file system.

例： drop table tmp3_like;



```
hive (tmp3)> drop table tmp3_like;
OK
Time taken: 0.111 seconds
```

使用show tables发现已经成功删除tmp3_like表



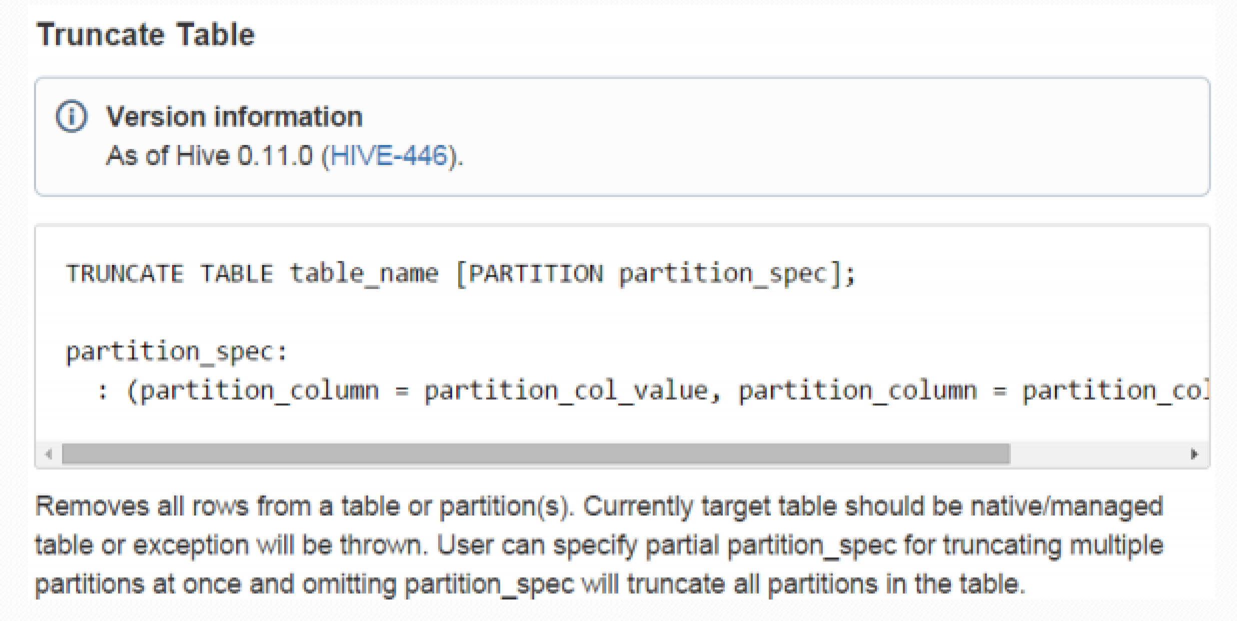
```
hive (tmp3)> show tables;
OK
tmp3_as
tmp3_table
Time taken: 0.009 seconds, Fetched: 2 row(s)
```

注意：同时删除元数据和hdfs的存储目录！

（3）清空表

6.4.2 Hive中表的基本操作

- ◆ 清空表命令：truncate table



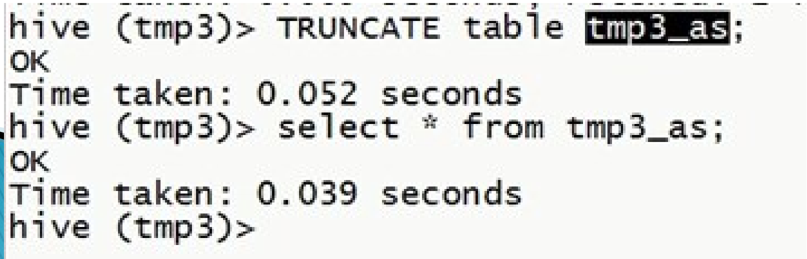
Truncate Table

ⓘ **Version information**
As of Hive 0.11.0 (HIVE-446).

```
TRUNCATE TABLE table_name [PARTITION partition_spec];

partition_spec:
  : (partition_column = partition_col_value, partition_column = partition_col
```

Removes all rows from a table or partition(s). Currently target table should be native/managed table or exception will be thrown. User can specify partial partition_spec for truncating multiple partitions at once and omitting partition_spec will truncate all partitions in the table.

例： TRUNCATE table tmp3_as;



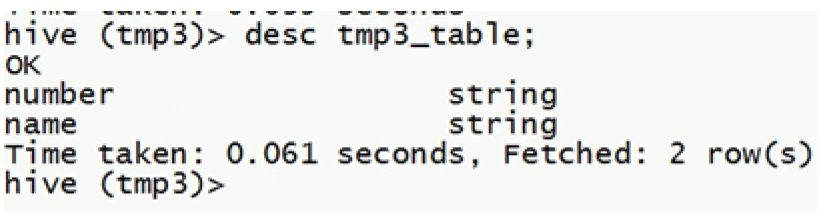
```
hive (tmp3)> TRUNCATE table tmp3_as;
OK
Time taken: 0.052 seconds
hive (tmp3)> select * from tmp3_as;
OK
Time taken: 0.039 seconds
hive (tmp3)>
```

（3）查看表的信息

6.4.2 Hive中表的基本操作

- 查看表的信息： desc tablename

例： desc tmp3_table;



```
hive (tmp3)> desc tmp3_table;
OK
number                  string
name                    string
Time taken: 0.061 seconds, Fetched: 2 row(s)
hive (tmp3)>
```

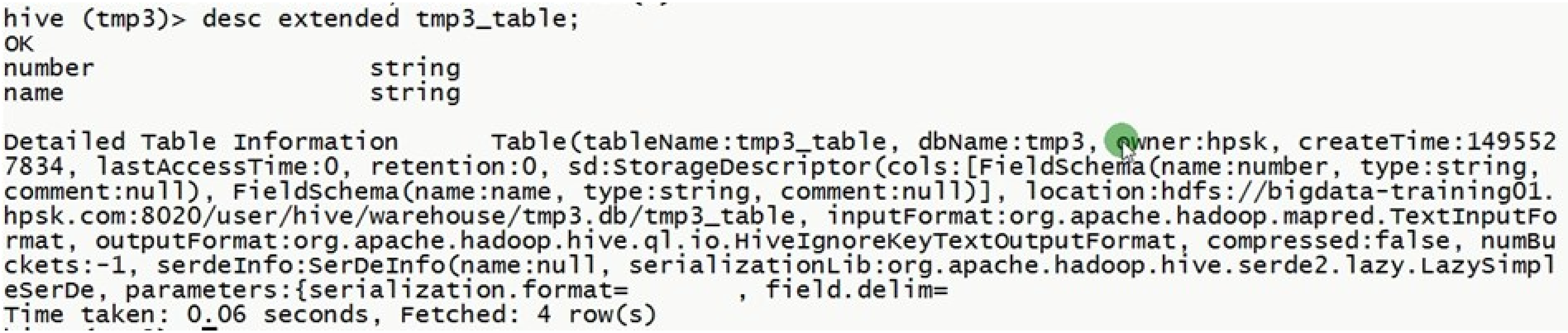
仅仅将表的信息作简单的显示

（3）查看表的信息

6.4.2 Hive中表的基本操作

- desc extended tablename;

例： desc extended tmp3_table;



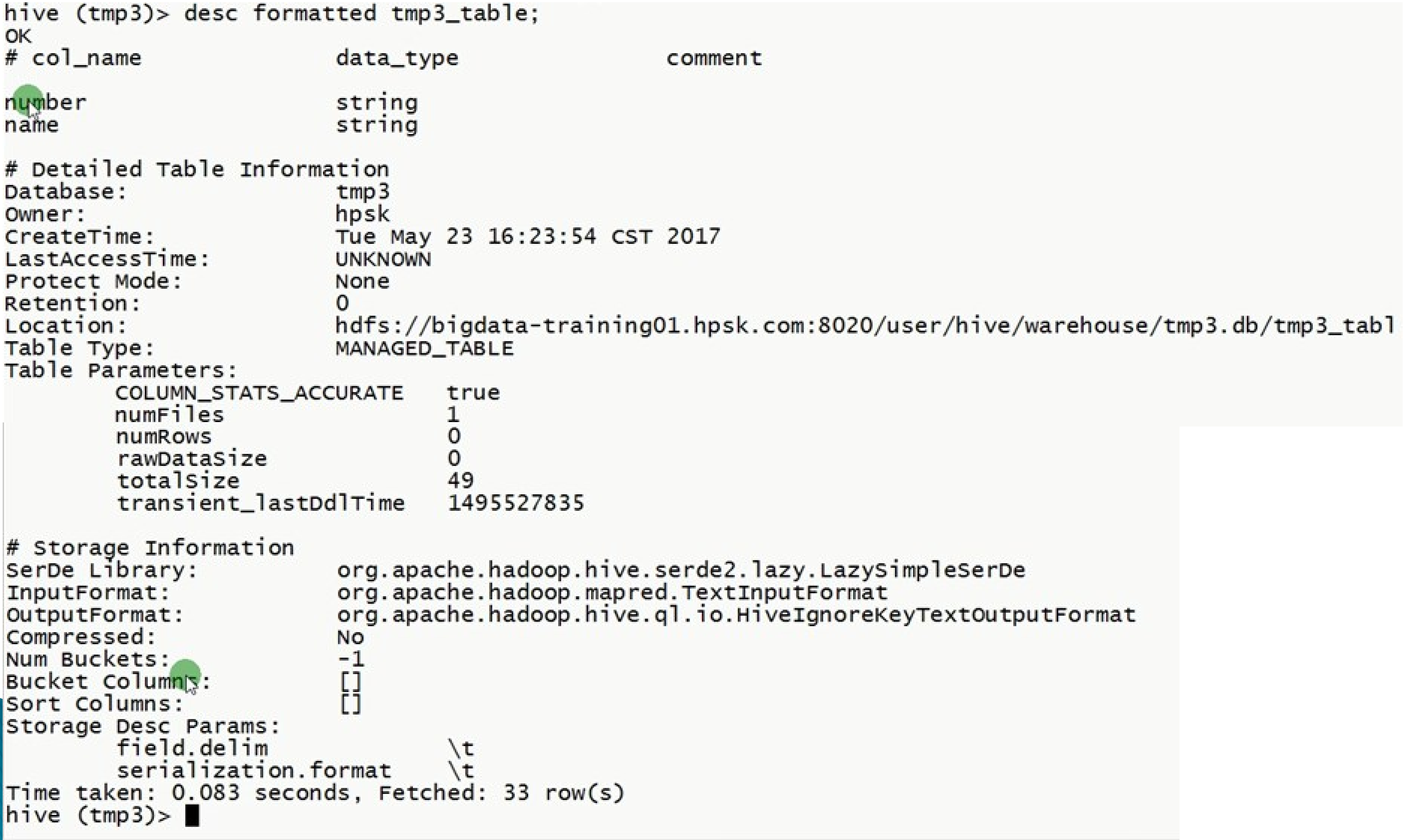
```
hive (tmp3)> desc extended tmp3_table;
OK
number                  string
name                    string

Detailed Table Information      Table(tableName:tmp3_table, dbName:tmp3, owner:hpsk, createTime:149552
7834, lastAccessTime:0, retention:0, sd:StorageDescriptor(cols:[FieldSchema(name:number, type:string,
comment:null), FieldSchema(name:name, type:string, comment:null)], location:hdfs://bigdata-training01.
hpsk.com:8020/user/hive/warehouse/tmp3.db/tmp3_table, inputFormat:org.apache.hadoop.mapred.TextInputFo
rmat, outputFormat:org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat, compressed:false, numBu
ckets:-1, serdeInfo:SerDeInfo(name:null, serializationLib:org.apache.hadoop.hive.serde2.lazy.LazySimpl
eSerDe, parameters:{serialization.format=       , field.delim=
Time taken: 0.06 seconds, Fetched: 4 row(s)
```

显示了表的名称、数据库的名称、所有者、创建时间等信息；

- desc formatted tablename;

格式化输出表的信息

例： desc formatted tmp3_table;



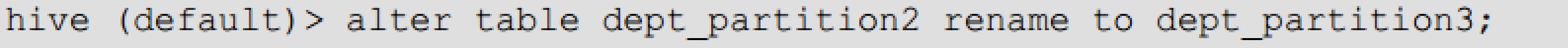
```
hive (tmp3)> desc formatted tmp3_table;
OK
# col_name              data_type               comment

number                  string
name                    string

# Detailed Table Information
Database:               tmp3
Owner:                  hpsk
CreateTime:             Tue May 23 16:23:54 CST 2017
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               hdfs://bigdata-training01.hpsk.com:8020/user/hive/warehouse/tmp3.db/tmp3_tabl
Table Type:             MANAGED_TABLE
Table Parameters:
        COLUMN_STATS_ACCURATE   true
        numFiles                1
        numRows                 0
        rawDataSize             0
        totalSize               49
        transient_lastDdlTime   1495527835

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             \t
        serialization.format    \t
Time taken: 0.083 seconds, Fetched: 33 row(s)
hive (tmp3)> █
```

6.4.2 Hive中表的基本操作

◆ 重命名表

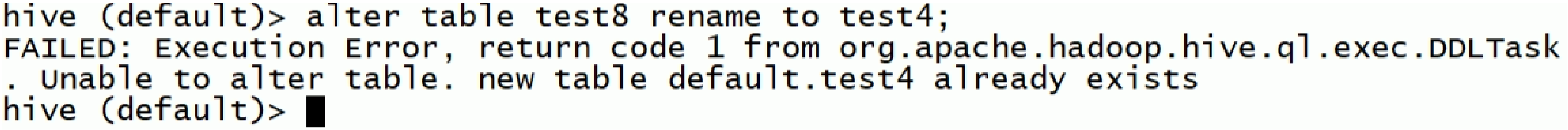
ALTER TABLE table_name RENAME TO new_table_name

例：将dept_partition2表的名称改为dept_partition3



```
hive (default)> alter table dept_partition2 rename to dept_partition3;
```

如果命名后的表名已经存在，则会报错：



```
hive (default)> alter table test8 rename to test4;
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask
. Unable to alter table. new table default.test4 already exists
hive (default)> 
```

这里的test4是已经存在的表，所以该语句报错

6.4.2 Hive中表的基本操作

- 替换（修改）列（列名列类型都可以改）

  ALTER TABLE table_name CHANGE col_old_name col_new_name column_type [COMMENT col_comment] [FIRST|AFTER column_name]

- 增加列、替换所有列

  ALTER TABLE table_name ADD|REPLACE COLUMNS(col_name data_type [COMMENT col_comment], ...)

  ADD 是代表新增一字段，字段位置在所有列后面

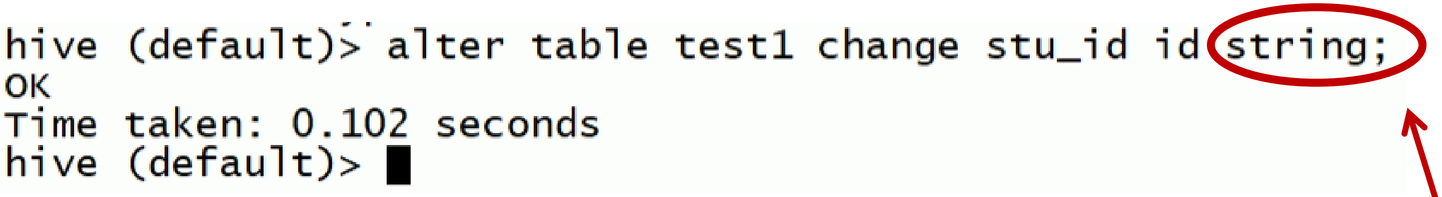
  REPLACE 则是表示替换表中所有字段

  注意：change、add、replace之后都可以跟多个字段

6.4.2 Hive中表的基本操作

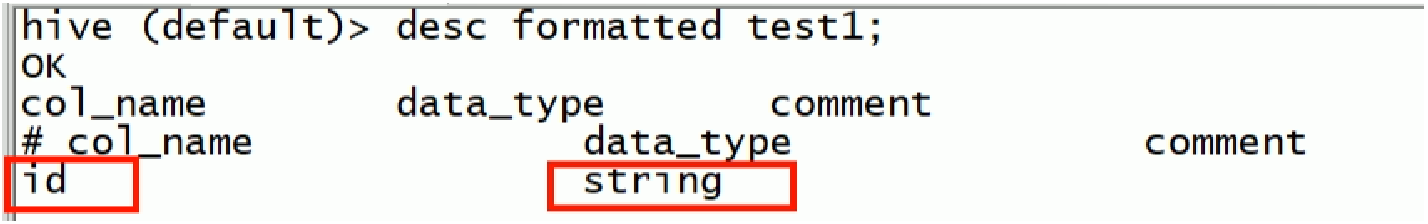
- 例1：

1、将test1表中的stu_id(string类型)替换成id（string类型）



```
hive (default)> alter table test1 change stu_id id string;
OK
Time taken: 0.102 seconds
hive (default)> █
```

2、查看test1表信息



```
hive (default)> desc formatted test1;
OK
col_name              data_type        comment
# col_name                      data_type                       comment
id                              string
```

注意：

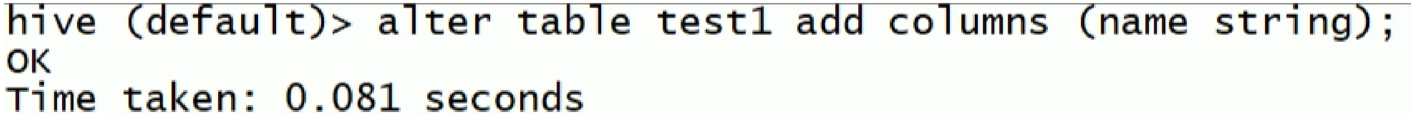
1、无论是否对字段的类型进行改动，都必须跟上改动后的数据类型

2、原类型为string无法改成int类型，而int类型可以改成string

6.4.2 Hive中表的基本操作

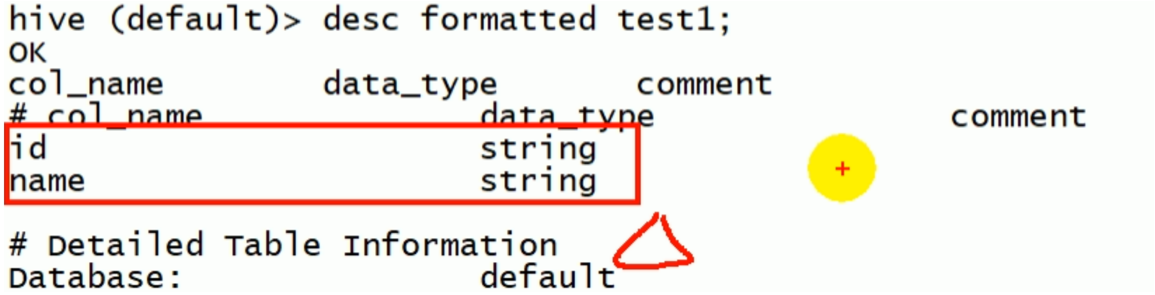
- 例2：

1、在test1中增加一个name（类型为string）字段



```
hive (default)> alter table test1 add columns (name string);
OK
Time taken: 0.081 seconds
```

2、查看test1表的信息

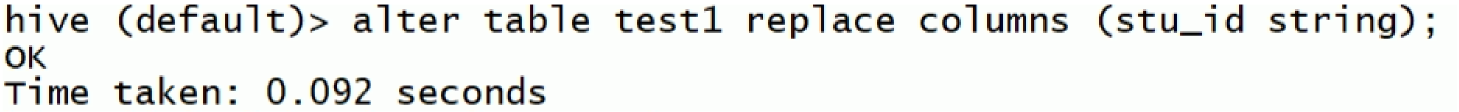


```
hive (default)> desc formatted test1;
OK
col_name            data_type           comment
# col_name                    data_type                     comment
id                            string
name                          string

# Detailed Table Information
Database:                     default
```

6.4.2 Hive中表的基本操作

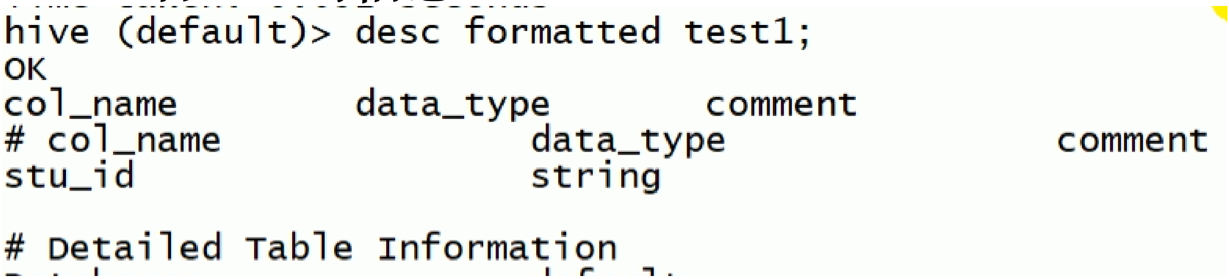
- 例3：

1、将test1表中的所有字段替换为stu_id（类型为string）



```
hive (default)> alter table test1 replace columns (stu_id string);
OK
Time taken: 0.092 seconds
```

2、查看test1的信息



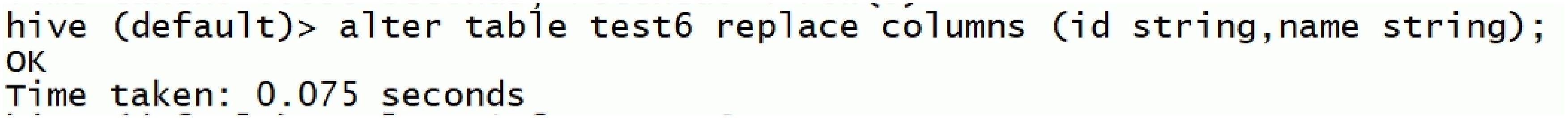
```
hive (default)> desc formatted test1;
OK
col_name                data_type               comment
# col_name                      data_type                       comment
stu_id                          string

# Detailed Table Information
```

注意replace与change的区别，change为替换指定的列，而replace替换整张表

6.4.2 Hive中表的基本操作

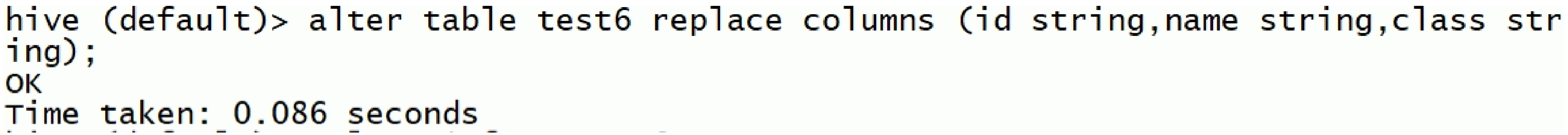
- 例4

1、将test6表中的所有字段替换为id（类型为string）、name（类型为string）



```
hive (default)> alter table test6 replace columns (id string,name string);
OK
Time taken: 0.075 seconds
```

2、将test6表中的字段替换为id（类型为string）、name（类型为string）、class（类型为string）



```
hive (default)> alter table test6 replace columns (id string,name string,class str
ing);
OK
Time taken: 0.086 seconds
```

（3）查看表的信息

6.4.2 Hive中表的基本操作

- 三种创建表的方式：

第一种：

```
create table if not exists tablename(
…
) row format delimited fields terminated by '\t';
stored as textfile;
load data local inpath ‘/…/dataname' into table tmp3_table;
```

第二种： create tablename1 as select name from tablename2;

第三种： create tablename1 like tablename2 ;

总结

- drop table tablename
- truncate table tablename
- desc tablename
- desc extended tablename;
- desc formatted tablename;
- show tables