Hive sql基本语法

本文档以FusionInsight版本的Hive主件进行讲解，不包含所有的语法。

一．DDL

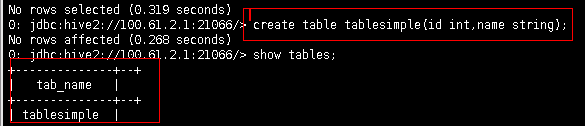
数据库定义语言(DDL),主要用来定义和管理SQL数据库中所有对象的语言,主要操作包括create(创建)、修改（alter）、显示（show或者describe）、删除（drop）。

1. **创建操作create**

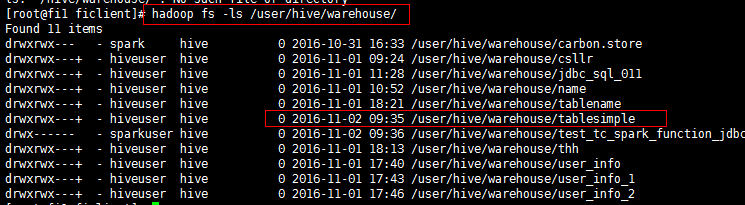
**1.1 创建普通表**

CREATE TABLE [IF NOT EXIST] table\_name(col1 type,col2 type …..)；

默认存储类型为：RCfile，默认存储类型在FI的管理页面是可以修改的。表的存储位置在hdfs中：/user/hive/warehouse/表名/数据文件。示例：



存储路径:



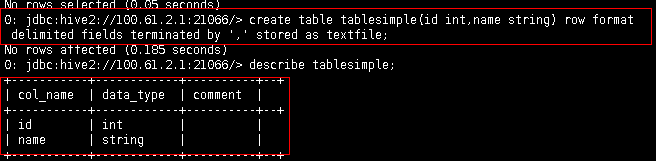
**1.2 创建指定存储类型的表**

CREATE TABLE stored\_table(col1 type,col2 type …..) ROW FORMAT DELIMITED ‘\t’ FIELDS TERMINATED BY '\n' STORED AS TEXTFILE;

row format delimited ‘\t’:表示数据行的分隔符为‘\t’;

fields terminated by '\n':表示数据列的分隔符为‘\n’;

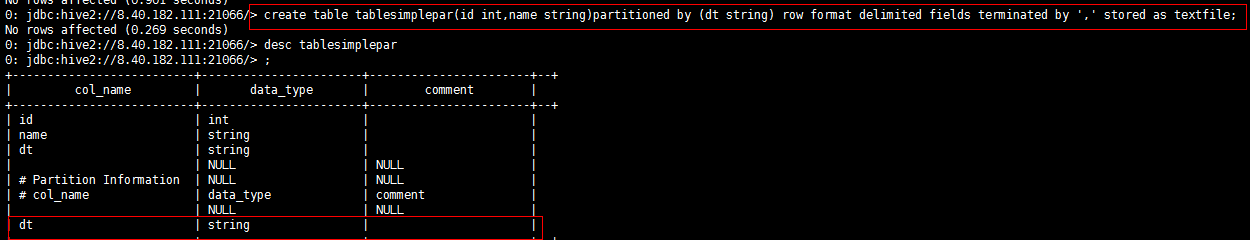
stored as textfile:表示该表指定的存储类型为文本类型（textfile）;系统支持的文件类型有：textfile、Sequencefile、RCfile、自定义类型。



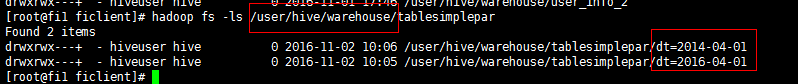
**1.3 创建分区表**

CREATE TABLE par\_table(col1 type,col2 type …..) PARTITIONED BY(date STRING, pos STRING) ROW FORMAT DELIMITED ‘\t’ FIELDS TERMINATED BY '\n' STORED AS SEQUENCEFILE;

partitioned by (date STRING,pos STRING)；表示该表有两个分区，分别是date和pos。分区表的存储位置跟一般表是有区别的，存储路径为：/user/hive/warehouse/表名/分区1/分区2/数据文件。示例如下：



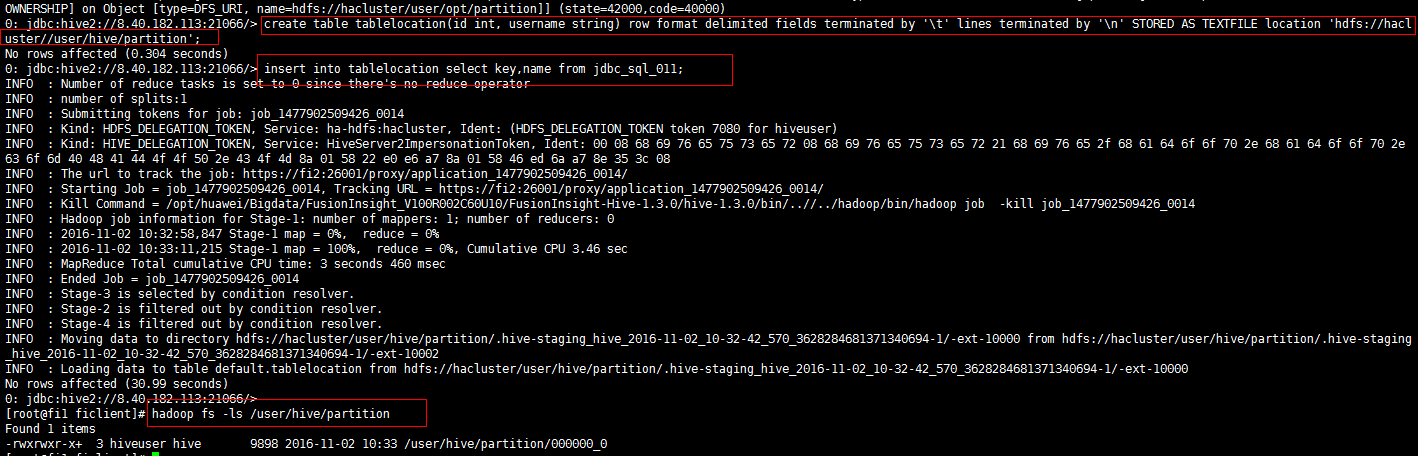
存储路径:



**1.4 创建指定路径的外部表**

CREATE EXTERNAL TABLE page\_view(col1 type,col2 type …..) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\054' STORED AS TEXTFILE LOCATION '<hdfs\_location>';

LOCATION ‘<HDFS\_location>’:通过此语句指定外部表的存储路径，该路径是在hdfs中，不是则本地。使用该语句时会涉及权限问题，location后面的路径建表用户有权限操作才能创建成功。表的存储位置为location指定的路径，示例如下：



**1.5 创建视图**

CREATE VIEW [IF NOT EXISTS] view\_name [ (column\_name [COMMENT column\_comment], ...) ];

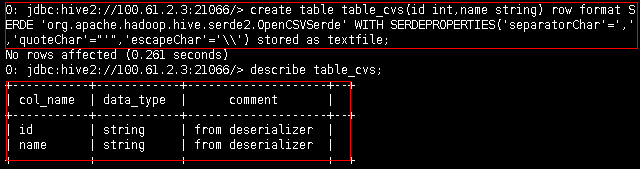
示例如下:

****

**1.6 创建CVS表**

CREATE TABLE tableName(col1 type,col2 type …..) ROW FORMAT SERDE

'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES ('separatorChar' = ',','quoteChar' = "'",'escapeChar' = '\\') STORED AS TEXTFILE;

****

**1.7 创建hive on hbase表**

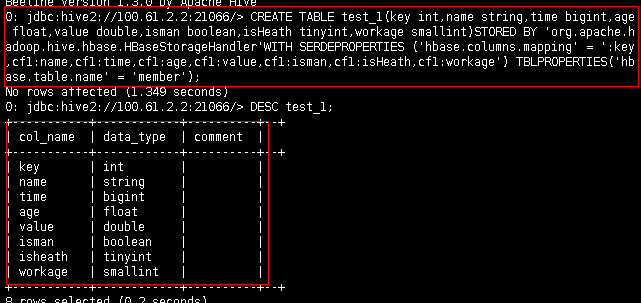
CREATE TABLE tableNam(col1 type,col2 type …..) STORED BY

'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES ("hbase.columns.mapping"="info:city") TBLPROPERTIES("hbase.table.name" = "hbseTableName");

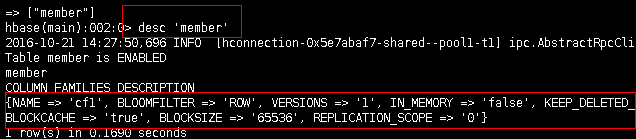
info:对应hbase表的列族

city:对应hbase表中info列族中具体的列

hbseTableName:对一个的hbase表名。此语句在创建hive表的同时会创建相关联的hbase表。在hive表对应的目录下不会生成文件（这是与一般的hive表不一样的地方）。



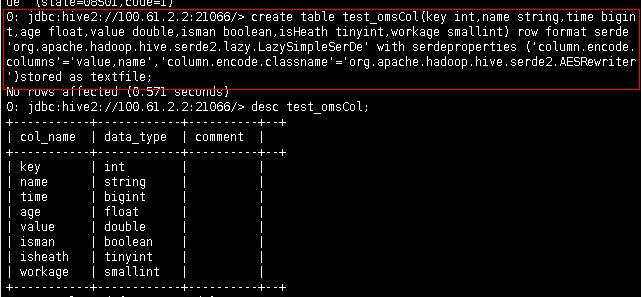
Hbase中表的信息：

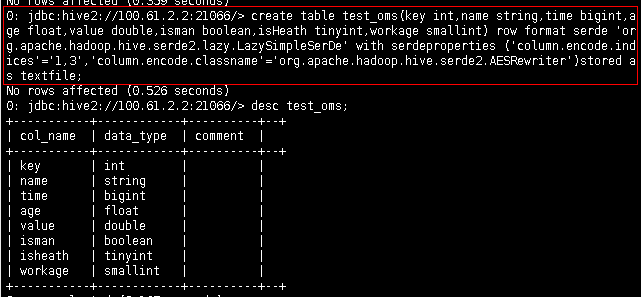
**1.7** **创建加密表**

创建加密列表有两种方式，一是指定加密列的列名，二是直接指定加密列的列号（从0

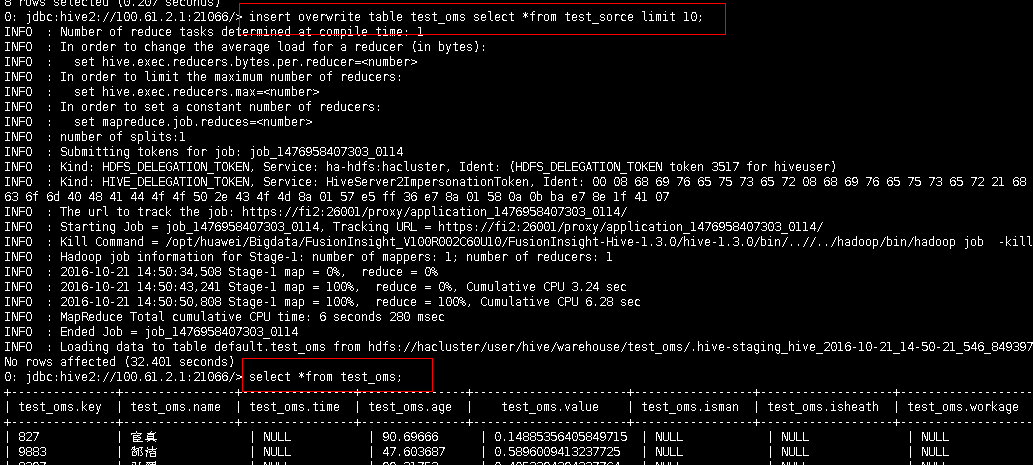
开始）。同时hive的加密方式也有2种，AES(对应加密类名称为：org.apache.hadoop.hive.serde2.AESRewriter)和SMS4(对应加密类名称为：org.apache.hadoop.hive.serde2.SMS4Rewriter)。目前列加密只支持存储在HDFS上的TextFile和SequenceFile文件格式的表。

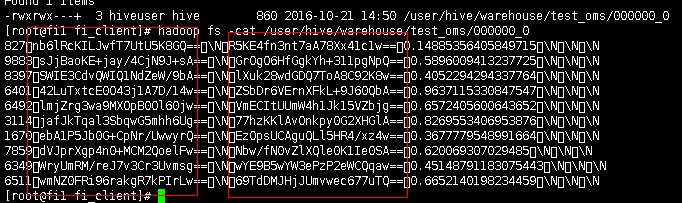
**使用列名创建列加密表：**

create table table\_name(col1 type,col2 type …..) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe' WITH SERDEPROPERTIES ('column.encode.columns'='col1,col2', 'column.encode.classname'='org.apache.hadoop.hive.serde2.AESRewriter')STORED AS TEXTFILE;**** **使用列号创建列加密表：**

create table table\_name(col1 type,col2 type …..) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe' WITH SERDEPROPERTIES ('column.encode.columns'='0,2', 'column.encode.classname'='org.apache.hadoop.hive.serde2.AESRewriter')STORED AS TEXTFILE;****

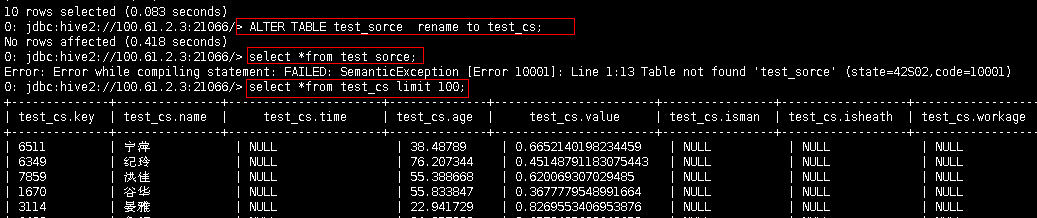
设置了加密列的表，使用select是不能判断是否加密的，只能通过数据文件判断加密是否生效。





1. **修改操作alter**

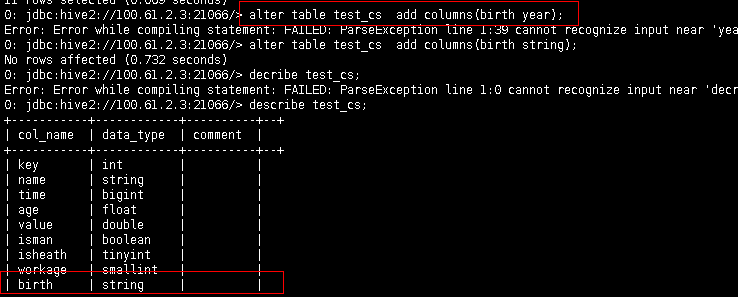
**2.1 修改表名**

ALTER TABLE table\_name RENAME TO new\_table\_name；**2.1 新增或者更新列**

ALTER TABLE table\_name ADD|REPLACE COLUMNS (col\_name data\_type

[COMMENT col\_comment], ...) ；

ADD是代表新增一字段，新增字段的默认位置在所有列后面(partition列前)；REPLACE则是表示替换表中所有字段。

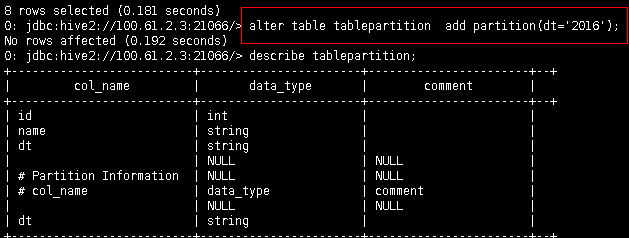


**2.3 增加和删除分区**

ALTER TABLE table\_name ADD/DROP PARTITION (partition\_col =partition\_col\_value,partition\_col = partiton\_col\_value, ...)；

ADD表示新增分区，DROP表示删除分区。新增分区和删除分区都是建立在分区表基

础上进行操作的，普通的表是不能增加或者删除分区的。



**2.4 修改列的名字、类型、位置、注释**

ALTER TABLE table\_name CHANGE col\_old\_name col\_new\_name column\_type [COMMENT col\_comment] [FIRST|AFTER column\_name]；

col\_old\_name：原始列名

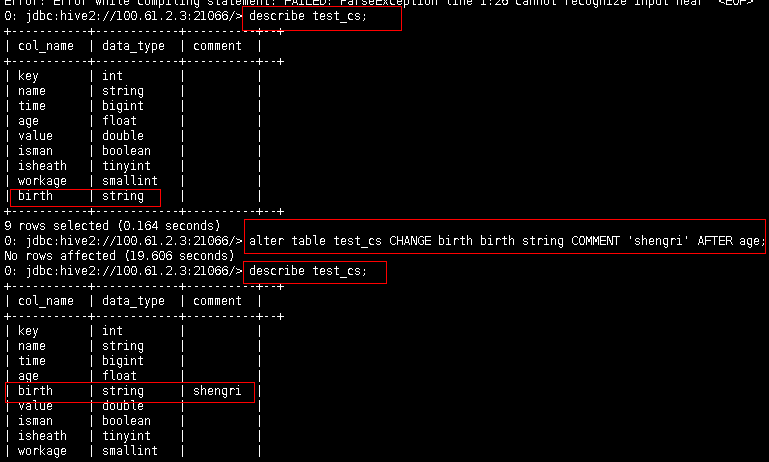
col\_new\_name：新的列名

column\_tyep：列的数据类型

COMMENT col\_comment：列的注释，这个可以有可以没有

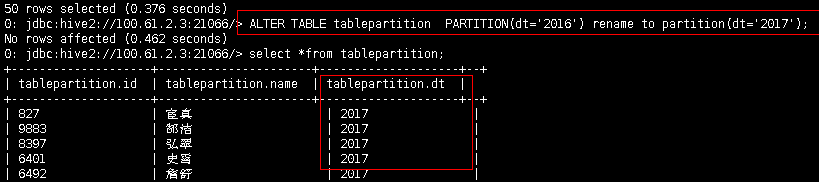
FIRST|AFTER column\_name：定义列的位置

上边这个命令可以允许改变列名、数据类型、注释、列位置或者它们的任意组合。



**2.5 重命名分区**

ALTER TABLE table\_name PARTITION (dt=old\_value) RENAME TO PARTITION (dt=new\_value);

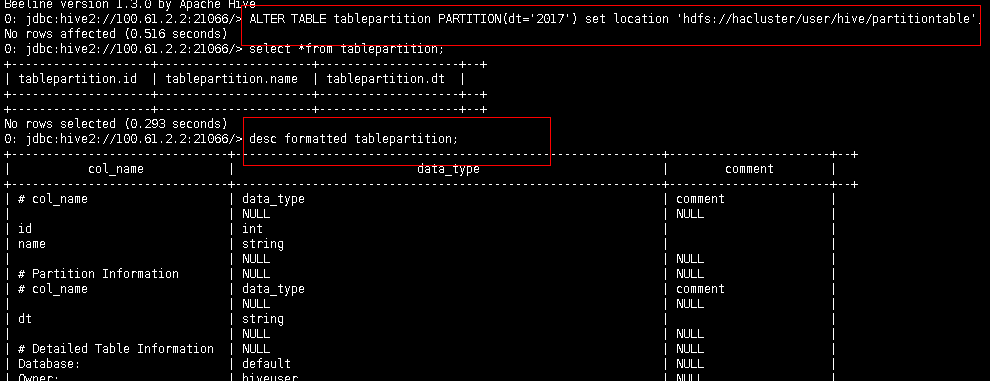
****

**2.6 修改分区的位置**

ALTER TABLE table\_name PARTITION (dt=old\_value) SET LOCATION "new location";

New location:默认实在hdfs中的，一般不建议使用，修改分区的位置之后，新导入的

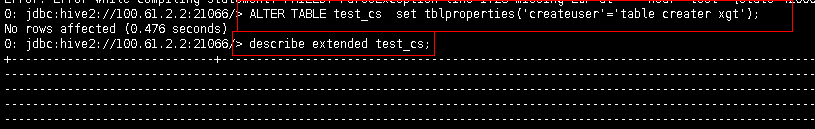
数据的存储在new location中

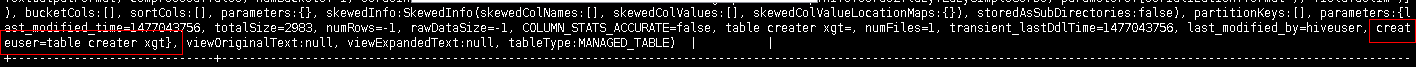


**2.7 增加表的元数据信息**

ALTER TABLE table\_name SET TBLPROPERTIES [property\_name = property\_value....]

这个新增表的描述信息，可以通过describe extended table来查看自己新增的表信息





**2.8 alter table/partition touch**

ALTER TABLE table\_name TOUCH [PARTITION partition\_apec];

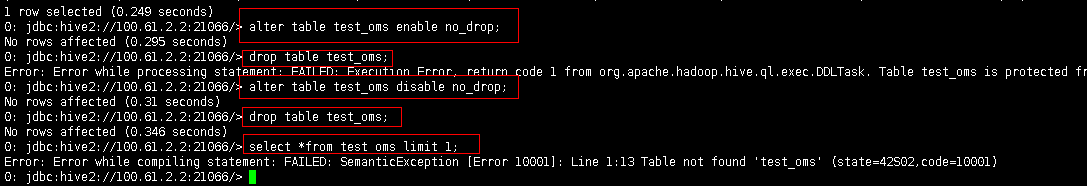
可以通过describe table来查看信息



**2.9 alter table/partition protections**

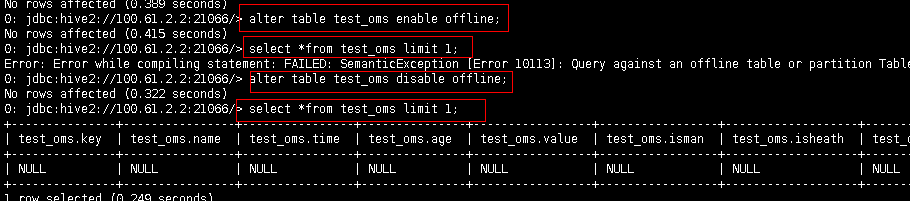
ALTER TABLE table\_name [PARTITION partition\_spec] ENABLE|DISABLE NO\_DROP;

//表和分区能否被删除



ALTER TABLE table\_name [PARTITION partition\_spec] ENABLE|DISABLE OFFLINE;

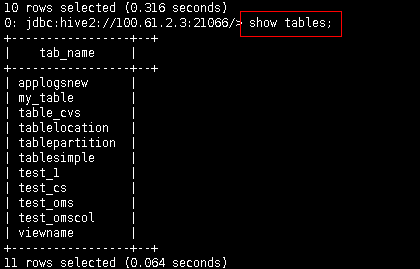
//表和分区能否查询

****

**3. 显示操作show**

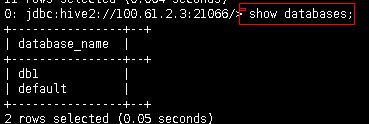
**3.1 显示表**

SHOW tables;

****

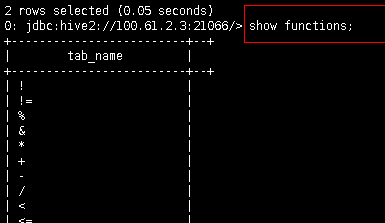
* 1. **显示数据库**

SHOW databases

****

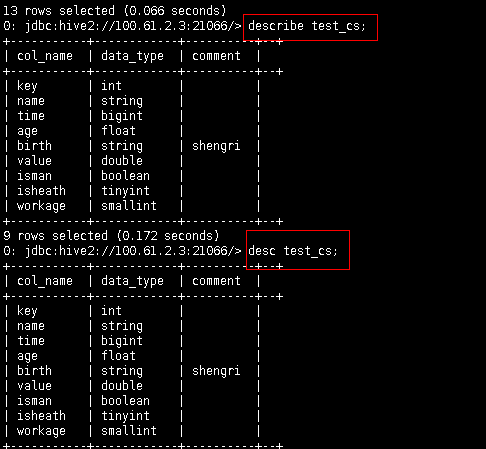
**3.3 显示内部函数**

SHOW functions;

****

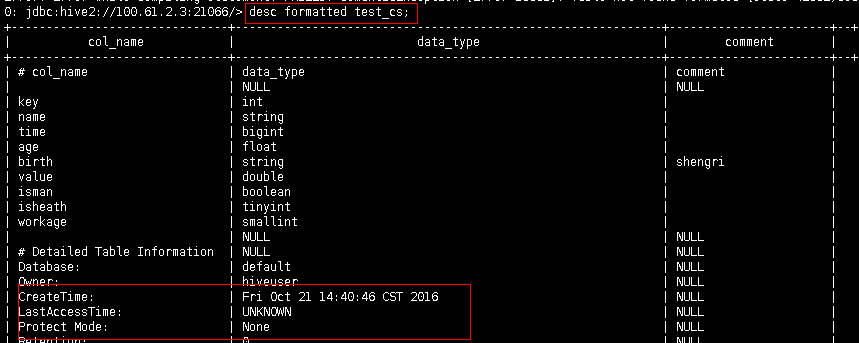
**3.4 获取表的描述信息**

DESCRIBE/DESC tablename;

****

**3.5 获取详细的表的描述**

Desc/describe formatted table\_name;

****

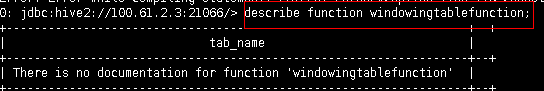
**3.6 获取数据库的描述信息**

describe database databaseName;

****

**3.7 获取内置函数的描述信息**

describe function function\_name;

****

二．DML

DML主要是数据库操作语言。主要操作有：查询（SELECT）、插入（INSERT）、

删除(REMOVE).

1. **查询操作(SELECT)**

查询只做简单介绍,关于group by/order by/join等复杂操作的查询,专门的查询文档做记录.

SELECT [ALL | DISTINCT] select\_expr, select\_expr, ... FROM table\_reference [WHERE where\_condition];

****

1. **数据操作(INSERT)**

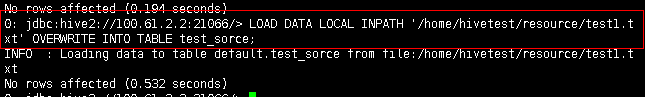
Hive支持两种数据操作,一是load导入数据,二是insert插入数据。两者的区别是：load是直接给一个文件的路径，把这个文件直接导入到表的存储位置，目前只有存储类型为textfile的表可以导入数据；insert不支持直接insert数据操作，它需要结合select一起完成数据的插入操作。

**2.1 load数据**

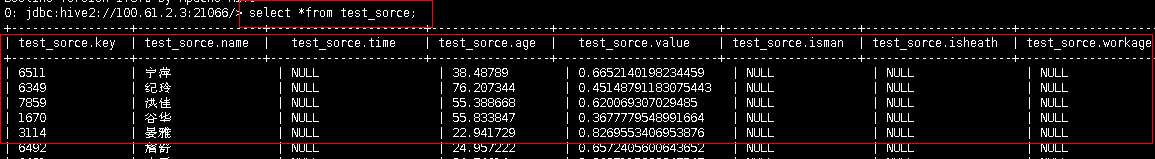
LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE tablename [PARTITION (partcol1=val1, partcol2=val2 ...)]

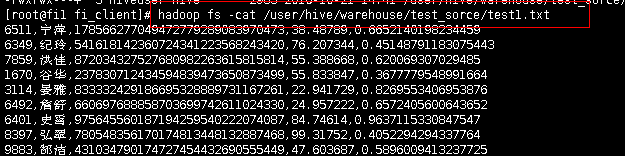
LOCAL表示节点本地的文件系统;没有LOCAL的路径为：/user，即hdfs的文件路径

OVERWRITE: 也就是说相同文件名上传，是否覆盖之前已上传了的文件。

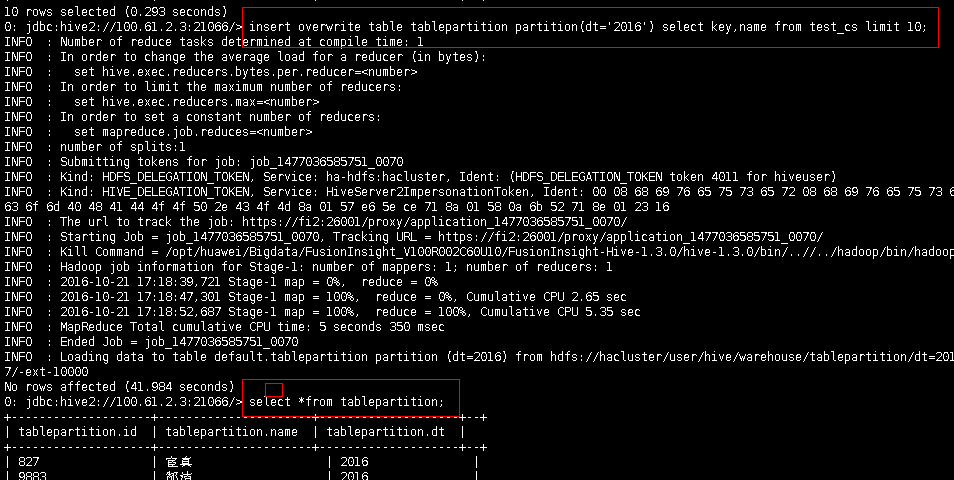


Load的数据可以通过select查询，也可以直接查看数据文件：



**2.2 insert overwrite数据**

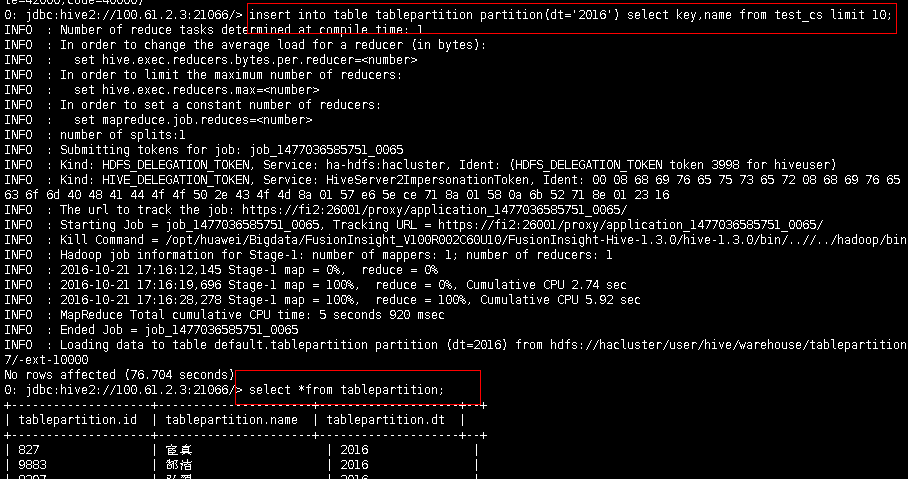
INSERT OVERWRITE TABLE tablename [PARTITION (part1=val1, part2=val2 ...)] select col1,col2.. FROM select\_table;

****

**2.3 insert into数据**

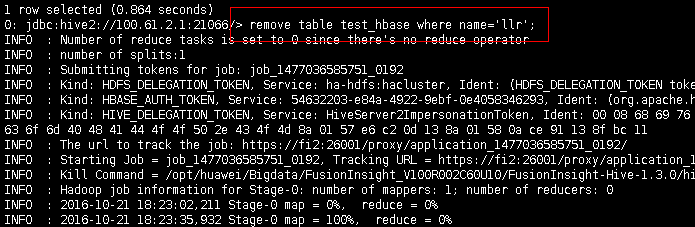
INSERT INTO TABLE tablename [PARTITION (part1=val1, part2=val2 ...)] select col1,col2.. FROM select\_table;

insert into是将select查询出来的数据直接插入表中。

**3. 删除操作(remove)**

Hive表不支持删除数据操作，但是hive on hbase提供remove数据操作。语法格式：

remove table tableName where 条件;

****