



链滴

# Hive 基础语句 (建库, 建表, 导入数据)

作者: [sq8852161](#)

原文链接: <https://ld246.com/article/1566182758015>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

版权声明：本文为博主原创文章，遵循 [CC 4.0 by-sa](https://creativecommons.org/licenses/by-sa/4.0/) 版权协议，转载请附上原文出处链接和本声明  
本文链接：<https://blog.csdn.net/u012274449/article/details/88327067>

可能会遇到权限不足的问题:

```
hadoop fs -chmod 777 /user
```

将一个文件加载到HDFS

```
hadoop fs -mkdir /user/demo/states/
```

```
hadoop fs -put /tmp/states.txt /user/demo/states/
```

```
hadoop fs -ls /user/demo/states/
```

使用hadoop fs 文件系统命令 将权限开启

最简单的建立数据库的方法:

```
CREATE DATABASE shopping;
```

它将在hive.metastore.warehouse.dir中定义的默认顶层目录下建立一个名为shopping.db的目录

完整语法如下:

```
CREATE(DATABASE|SCHEMA) [IF NOT EXISTS] database_name
```

```
[COMMENT database_comment]
```

```
[LOCATION hdfs_path]
```

```
[WITH DBPROPERTIES (PROPERTY_NAME = property_value,...)];
```

```
CREATE DATABASE if NOT EXISTS shopping
```

```
comment 'shores all shopping baseket data'
```

```
LOCATION '/user/hive/warehouse/SHOPPING.db'
```

```
WITH DBPROPERTIES('purpose'='testing');
```

修改数据库属性:

```
ALTER DATABASE shopping SET DBPROPERTIES('department'='SALES');
```

删除数据库:

```
DROP DATABASE shopping CASCADE; CASCADE意味着级联删除,数据库和表一起删除.默认为RES  
RICT (限制)
```

创建一张表:

```
CREATE EXTERNAL TABLE shopping.customers(  
fname STRING,
```

```
lname STRING,
```

lname STRING,

address STRUCT<HOUSENO:STRING, STREET:STRING, CITY:STRING, ZIPCODE:INT,STATE:STRING,COUNTRY:STRING>,

active BOOLEAN,

created DATE)

COMMENT 'yizhangbiao';

CREATE EXTERNAL表示创建一张 外部表; 删除外部表不会删除底层数据.

将一个文件加载到HDFS

```
hadoop fs -mkdir /user/demo/states/
```

```
hadoop fs -put /tmp/states.txt /user/demo/states/
```

```
hadoop fs -ls /user/demo/states/
```

创建一个内部表访问states文件夹下的文件,也就是states.txt,如果有多个文件,他会查出所有文件内容

```
CREATE TABLE states_internal (state string) LOCATION '/user/demo/states';
```

查看表定义:

```
DESCRIBE FORMATTED states_internal;
```

查询数据:

```
SELECT * FROM states_internal;
```

创建一个外部表:

```
CREATE EXTERNAL TABLE states_external (state string) LOCATION '/user/demo/states';
```

建立第二个外部表

```
CREATE EXTERNAL TABLE states_external2(state string) LOCATION '/user/demo/states';
```

删除外部表:

```
DRop TABLE states_external;
```

再次查询:

```
SELECT * FROM states_external2;
```

说明删除外部表不影响底层数据.

删除内部表

```
DROP TABLE states_internal;
```

按照文档中所说,删除内部表会将底层数据也进行删除.

但是实际操作,数据也没有删除..这里存疑

创建不含标题的外部表:

```
CREATE EXTERNAL TABLE state3(state string) LOCATION '/user/demo/states' TBLPROPERTIES "skip.header.line.count"="2");
```

生成已有表的create table命令:

```
show create table state3;
```

创建一个带分区列的表:

```
CREATE EXTERNAL TABLE transactions(
```

```
Transdate DATE,
```

```
transid INT,
```

```
custid INT,
```

```
fname STRING,
```

```
lname STRING,
```

```
item STRING,
```

```
qty INT,
```

```
price FLOAT
```

```
)
```

```
PARTITIONED BY(store STRING); //最后一句是分区列,这个列并不一定要在表结构中存在.
```

插入数据:

```
INSERT INTO transactions PARTITION(store="woshi") values("01/25/2016",101,"A109","1111",  
SMITH","SHOES",1,11);
```

查询日期

```
SELECT * FROM transactions WHERE transdate BETWEEN '2018-11-03'and '2019-12-12'
```

使用字符串日期 如'2019-01-01'作为分区,是高效的,并且适用于很多匹配符:

如 in like between

分桶:

分桶会按照指定列,均衡的分为多少桶,不会产生新的目录及列.

桶编号最好为质数

需要链接的表,桶数必须相同,或者一个桶数为另一个桶数的因子

```
CREATE EXTERNAL TABLE customers (store string) CLUSTERED BY (store) INTO 3 BUCKETS LOCATION '/user/demo/states';
```

创建临时表:

```
CREATE TEMPORARY TABLE states(state STRING);
```

改变表名:

```
Alter table states RENAME TO states_old;
```

alter table 命令只会修改表结构,但是不会修改表数据.

将表转换为ORC文件:

```
CREATE TABLE states_orc STORED as ORC TBLPROPERTIES ("ORC.COMPRESS"="SNAPPY") as  
SELECT * from state3;
```

合并表的文件:

```
ALTER TABLE state CONCATENATE;
```

添加分区:

外部表

```
ALTER TABLE ids ADD PARTITION (datestamp='2019-03-03') LOCATION '/user/demo/ids/2019-03-03';
```

内部表:

```
MSCK REPAIR TABLE ids_internal;
```