

# Hive 入门

周维萍

2013 年 4 月

## 一.Hive 简介

Hive 是基于 hadoop 分布式文件系统的一种数据库，它的数据都是以文件形式存在的。Hive 中的每一条记录对应于文件中的一行，各个字段的值是被指定的分隔符分隔的。在读数据的时候，会将文件行以分隔符分隔字段值，并将各个值按顺序给字段；现有的 hive 的权限基于文件的，如果某个用户对表对应的文件有读的权限，那么用户就对表有读的权限。

当前 hive 运用最多的是分区，hive 会将各个分区的数据分别放在不同的文件夹下；

在用 hive 执行 SQL 语句时，是将语句处理成 mapreduce 程序运行的。

## 二.数据类型

整型 int 4 字节 smallint 2 字节 Tinyint 1 字节 bigint 8 字节
浮点数 float double
字符串 string
布尔型 boolean
不支持日期时间型
不支持二进制串
其它数据类型 ARRAY MAP STRUCT CREATE TABLE complex ( col1 ARRAY<INT>, col2 MAP<STRING, INT>, col3 STRUCT<a:STRING, b:INT, c:DOUBLE> ); hive> SELECT col1[0], col2['b'], col3.c FROM complex; 1 2 1.0

### 三.支持各种内建函数

详见 [hive 函数大全.pdf](#)

### 四.DDL(数据定义)

#### 1. 创建和删除建数据库

```
create database if not exists db_test
comment '用于测试';
drop database if exists db_test;
```

#### 2. 建表

```
create external table order_joined_extend(
addr_id bigint comment 'address id' ,
alliance_id int ,
allot_quantity int ,
city_ship_type_desc string
)
comment 'order_joined_extend'
partitioned by (create_date string,type string)
row format delimited fields terminated by '\001'
lines terminated by '\n'
stored as textfile
location '/home/zhouweiping/order_joined_extend/';
```

**external** 建立外部表。外部表的好处：**a.**可以直接将数据文件放到 **location** 指定的目录，在 **hive** 中即可查询出数据；**b.**可以多个表使用一份数据，只需将 **location** 指向同一个目录；

**partitioned by** 建立分区表。分区表是将分区列值一样的放到一个文件中，如果该分区列下还有子分区，会在该文件夹下再分小文件夹；如图：

```
[zhouweiping@h252021 ~]$ hdfs ls /share/comm/ddclick/ | head -3
Found 239 items
drwxrwxr-x - maze supergroup 0 2013-02-01 10:24 /share/comm/ddclick/2012-08-30
drwxrwxr-x - maze supergroup 0 2013-01-21 21:17 /share/comm/ddclick/2012-09-01
[zhouweiping@h252021 ~]$ hdfs ls /share/comm/ddclick/2012-09-01 | head -3
Found 39 items
drwxrwxr-x - maze supergroup 0 2013-01-21 20:05 /share/comm/ddclick/2012-09-01/ddclick_account
drwxrwxr-x - maze supergroup 0 2013-01-21 19:38 /share/comm/ddclick/2012-09-01/ddclick_bang
[zhouweiping@h252021 ~]$ hdfs ls /share/comm/ddclick/2012-09-01/ddclick_bang | head -10
Found 12 items
-rw-rw-r-- 3 maze supergroup 0 2013-01-21 19:38 /share/comm/ddclick/2012-09-01/ddclick_bang/_SUCCESS
drwxrwxr-x - maze supergroup 0 2013-01-21 19:34 /share/comm/ddclick/2012-09-01/ddclick_bang/_logs
-rw-rw-r-- 3 maze supergroup 22286078 2013-01-21 19:35 /share/comm/ddclick/2012-09-01/ddclick_bang/part-m-00000
-rw-rw-r-- 3 maze supergroup 21978171 2013-01-21 19:35 /share/comm/ddclick/2012-09-01/ddclick_bang/part-m-00001
-rw-rw-r-- 3 maze supergroup 22185937 2013-01-21 19:35 /share/comm/ddclick/2012-09-01/ddclick_bang/part-m-00002
-rw-rw-r-- 3 maze supergroup 22392740 2013-01-21 19:35 /share/comm/ddclick/2012-09-01/ddclick_bang/part-m-00003
-rw-rw-r-- 3 maze supergroup 22196028 2013-01-21 19:35 /share/comm/ddclick/2012-09-01/ddclick_bang/part-m-00004
```

row format 指定表中行列分隔符。

Stored as 文件存储的格式，此处的 textfile。

Location 指定表中数据文件存放的 hdfs 目录。该参数默认为：

/user/hive/warehouse/dbname.db/tablename

也可以用 `create table table_name like old_table_name`, 但是这个只能建内表，不能建外表，就是加了 `external`，所建的表任然是内表；而且在建表时如果原表是分区表，新建的表也只是一般的表，原表中的分区字段成了新表中的一般字段

### 3. 建表的时候可以同时插入数据

```
create table order_joined_extend1
comment 'order_joined_extend'
row format delimited fields terminated by '\001'
lines terminated by '\n'
stored as textfile
location '/home/zhouweiping/order_joined_extend1/'
as
select * from order_joined_extend;
```

但是这种方法不支持外部表和分区表，并且在建表时不能指定详细的列

### 4. 删除表

```
drop table if exists order_joined_extend1;
```

删除的表可能是外部表或者内表，在删除外部表时只是删除了表结构，数据文件依然存在

### 5. 修改表

增加删除分区

```
alter table order_joined_extend
add partition(create_date='2012-09-01', type='ddclick_bang')
location '/share/comm/ddclick/2012-09-01/ddclick_bang/';

alter table order_joined_extend
drop if exists partition(create_date='2012-09-01', type='ddclick_bang');
```

重命名

```
alter table order_joined_extend rename to order_joined_extend_rename;
```

替换原有的列，替换时只是在分区列之前，分区列不变

```
ALTER TABLE order_joined_extend REPLACE COLUMNS
(
    product_id string,
```

```
product_name string,  
bd_name string  
)
```

增加列，之后在分区之前的最后一列加，不能指定到某列之后

```
alter table order_joined_extend  
add columns (add_col_test string)
```

内部表转外部表

```
alter table tablePartition set TBLPROPERTIES ('EXTERNAL'='TRUE');
```

外部表转内部表

```
alter table tablePartition set TBLPROPERTIES ('EXTERNAL'='FALSE');
```

## 6. Show/describe

```
show databases;
```

```
show tables;
```

```
show tables '*tianzhao*';
```

显示表中中含有tianzhao的表名

```
show partitions table_name;
```

展示表中现有的分区

```
desc formatted table_name;
```

可以描述出很多信息，包括字段，location，分区字段，是内表或者外表等；

```
show functions;
```

显示可以用的函数列表，包括可用的udf函数。

```
describe function length;
```

返回length函数的说明

```
show table extended like order_joined_extend partition(create_date='2012-09-01',type='ddclick_bang');
```

显示指定分区的一些信息

## 五.DML（数据操作）

**Hive 只支持 select、insert，不支持 delete、update**

### 1. load 数据

Load**本地数据**到hive，最好指定本地文件的绝对路径

**追加**导入数据：load data **local** inpath '/home/zhouweiping/d.dat' into table order\_joined\_extend1;

```
覆盖导入数据: load data local inpath '/home/zhouweiping/d.dat' overwrite into
table order_joined_extend1;
```

加载hdfs上的数据到hive表

如果是外表可以直接将数据文件拷贝到location的目录

```
Hadoop fs -cp from location
```

内表或者外表都可以用load的方法

```
load data inpath '/home/zhouweiping/d.dat' into table order_joined_extend1;
```

load 数据时:

如果数据在**本地**, 会将本地数据**复制**一份到 hdfs 中表的 location;

如果是 **hdfs** 是的数据, 会直接**移动**到 location; **所以如果 load 数据的数据文件跟 location 相同, 会报错;**

## 2. Insert

插入数据到非分区表

```
Insert overwrite table table1
```

```
Select * from table2
```

**插入数据到分区表, 需要指定分区值**

```
insert overwrite table order_joined_extend partition (create_date='2012-09-01',type='ddclick_bang')
```

```
select addr_id,alliance_id,allot_quantity,city_ship_type_desc, from
order_joined_extend1;
```

**一个输入, 多个输出**

**from table2**

```
Insert overwrite table table1 Select *
```

```
Insert overwrite table table3 Select *
```

**动态分区**

```
set hive.exec.dynamic.partition=true;
```

```
set hive.exec.dynamic.partition.mode=nostrict;
```

```
INSERT OVERWRITE TABLE order_joined_extend PARTITION(createdate,type)
```

```
SELECT *
```

```
FROM order_joined_extend1 ;
```

Hive将会以select的**最后两列作为动态分区的值**, 将createdate,type相同的列插入到一个partition中

**将query的结果写入文件**

写到本地文件

```
insert overwrite local directory '/home/zhouweiping/directory.dat'
```

```
select * from order_joined_extend limit 10;
```

```
写到hdfs
insert overwrite directory '/home/zhouweiping/directory.dat'
select * from order_joined_extend limit 10;
```

### 3. select

一般的 SQL 语句都支持

```
SELECT [ALL | DISTINCT] select_expr, select_expr, ...
FROM table_reference
[WHERE where_condition]
[GROUP BY col_list]
[CLUSTER BY col_list
 | [DISTRIBUTE BY col_list] [SORT BY col_list]
]
[LIMIT number]
```

在使用聚合函数时，select 的列必须是 group by 后面的字段或者只用了聚合函数的；

### 4. Join

Hive 只支持等值连接（equality joins）、外连接（outer joins）和（left semi join）。Hive 不支持所有非等值的连接，因为非等值连接非常难转化到 map/reduce 任务；hive 也不支持 in 子查询，但是可以用 left semi join 实现 in 操作。另外，Hive 支持多于 2 个表的连接。

有些数据库，例如 MySQL 和 Oracle，允许在 SELECT 语句的 FROM 子句中列出要连接的表，而在 WHERE 子句中指定连接条件。但是 Hive 并不支持这种语法，所以下面的语句会由于解析出错而导致运行失败：

```
SELECT sales.*, things.*
FROM sales, things
WHERE sales.id = things.id;
```

Hive 只允许在 FROM 子句中出现一个表。要进行连接操作，必须遵循其他的等值连接跟关系型数据库一样；

**JOIN 子句中表的顺序很重要：一般最好将最大的表在最后。**

```
hive> SELECT *
      > FROM things LEFT SEMI JOIN sales ON (sales.id = things.id);
2   Tie
3   Hat
4   Coat
```

写 LEFT SEMI JOIN 查询时必须遵循一个限制：右表(sales)只能在 ON 子句中出现。例如，我们不能在 SELECT 表达式中引用右表。

并运算 union all

```
INSERT OVERWRITE TABLE actions_users
SELECT u.id, actions.date
FROM (
  SELECT av.uid AS uid
  FROM action_video av
  WHERE av.date = '2008-06-03'
  UNION ALL
  SELECT ac.uid AS uid
  FROM action_comment ac
  WHERE ac.date = '2008-06-03'
) actions JOIN users u ON(u.id = actions.uid);
```