

## Hive几种数据导出方式

写在前面的话  
，学Hive这么久了，发现目前国内还没有一本完整的介绍Hive的书籍，而且互联网上面的资料很乱，于是我决定写一些关于[《Hive的那些事》](#)序列文章，分享给大家。我会在接下来的时间整理有关Hive的资料，如果对Hive的东西感兴趣，请关注本博客。[/archives/tag/hive的那些事](#)

在本博客的[《Hive几种数据导入方式》](#)文章中，谈到了Hive中几种数据的导入方式，不同的数据导入方式用途不一样。今天我们再谈谈Hive中的几种不同的数据导出方式。可以根据导出的地方不一样，将这些方式分为三种：

- 导出到本地文件系统；
- 导出到HDFS中；
- 导出到Hive的另一个表中。

为了避免单纯的文字，我将一步一步地用命令进行说明。

### 导出到本地文件系统

```
hive> insert overwrite local directory '/home/wyp/wyp'  
> select * from wyp;
```

这条HQL的执行需要启用Mapreduce完成，运行完这条语句之后，将会在本地文件系统的/home/wyp/wyp目录下生成文件，这个文件是Reduce产生的结果（这里生成的文件名是000000\_0），我们可以看看这个文件的内容：

```
[wyp@master ~/wyp]$ vim 000000_0  
5^Awyp1^A23^A131212121212  
6^Awyp2^A24^A134535353535  
7^Awyp3^A25^A132453535353  
8^Awyp4^A26^A154243434355  
1^Awyp^A25^A13188888888888  
2^Atest^A30^A13888888888888  
3^Azs^A34^A899314121
```

可以看出，这就是wyp表中的所有数据。数据中的列与列之间的分隔符是 ^A (ascii码是 W00001)。

和导入数据到Hive不一样，不能用 insert into 来将数据导出：

```
hive> insert into local directory '/home/wyp/wyp'
> select * from wyp;
NoViableAltException(79@[])
    at org.apache.hadoop.hive.ql.parse.HiveParser_SelectClauseParser.selectClause(HiveParser_SelectClauseParser.java:683)
    at org.apache.hadoop.hive.ql.parse.HiveParser.selectClause(HiveParser.java:30667)
    at org.apache.hadoop.hive.ql.parse.HiveParser.regular_body(HiveParser.java:28421)
    at org.apache.hadoop.hive.ql.parse.HiveParser.queryStatement(HiveParser.java:28306)
    at org.apache.hadoop.hive.ql.parse.HiveParser.queryStatementExpression(HiveParser.java:28100)
    at org.apache.hadoop.hive.ql.parse.HiveParser.execStatement(HiveParser.java:1213)
    at org.apache.hadoop.hive.ql.parse.HiveParser.statement(HiveParser.java:928)
    at org.apache.hadoop.hive.ql.parse.ParseDriver.parse(ParseDriver.java:190)
    at org.apache.hadoop.hive.ql.Driver.compile(Driver.java:418)
    at org.apache.hadoop.hive.ql.Driver.compile(Driver.java:337)
    at org.apache.hadoop.hive.ql.Driver.run(Driver.java:902)
    at org.apache.hadoop.hive.cli.CliDriver.processLocalCmd(CliDriver.java:259)
    at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:216)
    at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:413)
    at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:756)
    at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:614)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
    at java.lang.reflect.Method.invoke(Method.java:597)
    at org.apache.hadoop.util.RunJar.main(RunJar.java:212)
FAILED: ParseException line 1:12 missing TABLE at 'local' near 'local' in select clause
line 1:18 cannot recognize input near 'directory' '/home/wyp/wyp' 'select' in select clause
```

## 导出到HDFS中

和导入数据到本地文件系统一样的简单，可以用下面的语句实现：

```
hive> insert overwrite directory '/home/wyp/hdfs'
```

```
> select * from wyp;
```

将会在HDFS的 /home/wyp/hdfs 目录下保存导出来的数据。注意，和导出文件到本地文件系统的HQL少一个local，数据的存放路径就不一样了。

## 导出到Hive的另一个表中

其实这个在[《Hive几种数据导入方式》](#)文中就用到了，这也是Hive的数据导入方式，如下操作：

```
hive> insert into table test
  > partition (age='25')
  > select id, name, tel
  > from wyp;
#####
####
    这里输出了一堆Mapreduce任务信息，这里省略
#####
####
Total MapReduce CPU Time Spent: 1 seconds 310 msec
OK
Time taken: 19.125 seconds

hive> select * from test;
OK
5  wyp1  131212121212  25
6  wyp2  134535353535  25
7  wyp3  132453535353  25
8  wyp4  154243434355  25
1  wyp   13188888888888  25
2  test  13888888888888  25
3  zs    899314121    25
Time taken: 0.126 seconds, Fetched: 7 row(s)
```

细心的读者可能会问，怎么导入数据到文件中，数据的列之间为什么不是wyp表设定的列分隔符呢？其实在Hive 0.11.0版本之间，数据的导出是不能指定列之间的分隔符的，只能用默认的列分隔符，也就是上面的^A来分割，这样导出来的数据很不直观，看起来很不方便！

如果你用的Hive版本是0.11.0，那么你可以在导出数据的时候来指定列之间的分隔符（可以参见本博客的[《Hive0.11查询结果保存到文件并指定列之间的分隔符》](#)），操作如下：

```
hive> insert overwrite local directory '/home/iteblog/local'  
> row format delimited  
> fields terminated by '\t'  
> select * from wyp;
```

```
[wyp@master ~/local]$ vim 000000_0  
5   wyp1  23   131212121212  
6   wyp2  24   134535353535  
7   wyp3  25   132453535353  
8   wyp4  26   154243434355  
1   wyp   25   13188888888888  
2   test  30   13888888888888  
3   zs    34   899314121
```

这个很不错吧！

其实，我们还可以用hive的 -e 和 -f 参数来导出数据。其中-e 表示后面直接接带双引号的sql语句；而-f是接一个文件，文件的内容为一个sql语句，如下：

```
[wyp@master ~/local]$ hive -e "select * from wyp" >> local/wyp.txt  
[wyp@master ~/local]$ cat wyp.txt  
5   wyp1  23   131212121212  
6   wyp2  24   134535353535  
7   wyp3  25   132453535353  
8   wyp4  26   154243434355  
1   wyp   25   13188888888888  
2   test  30   13888888888888  
3   zs    34   899314121
```

得到的结果也是用 \t 分割的。也可以用-f参数实现：

```
[wyp@master ~/local]$ cat wyp.sql  
select * from wyp  
[wyp@master ~/local]$ hive -f wyp.sql >> local/wyp2.txt
```

上述语句得到的结果也是 Wt 分割的。

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: **【】**（**）**