Hive几种数据导入方式

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。https://www.iteblog.com/archives/tag/hive-technology/

好久没写Hive的那些事了，今天开始写点吧。今天的话题是总结Hive的几种常见的数据导入方式，我总结为四种：

- 从本地文件系统中导入数据到Hive表；
- 从HDFS上导入数据到Hive表；
- 从别的表中查询出相应的数据并导入到Hive表中；
- 在创建表的时候通过从别的表中查询出相应的记录并插入到所创建的表中。

我会对每一种数据的导入进行实际的操作，因为纯粹的文字让人看起来很枯燥，而且学起来也很抽象。好了，开始操作！

一、从本地文件系统中导入数据到Hive表

先在Hive里面创建好表，如下：

```
hive> create table wyp
    > (id int, name string,
    > age int, tel string)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '₩t'
    > STORED AS TEXTFILE;
OK
Time taken: 2.832 seconds
```

这个表很简单，只有四个字段，具体含义我就不解释了。本地文件系统里面有个/home/wyp/wyp.txt 文件，内容如下：

```
[wyp@master ~]$ cat wyp.txt
1       wyp     25      13188888888888
2       test    30      13888888888888
3       zs      34      899314121
```

wyp.txt文件中的数据列之间是使用 ₩t
分割的，可以通过下面的语句将这个文件里面的数据导入到wyp表里面，操作如下：

```
hive> load data local inpath 'wyp.txt' into table wyp;
Copying data from file:/home/wyp/wyp.txt
Copying file: file:/home/wyp/wyp.txt
Loading data to table default.wyp
Table default.wyp stats:
[num_partitions: 0, num_files: 1, num_rows: 0, total_size: 67]
OK
Time taken: 5.967 seconds
```

这样就将wyp.txt里面的内容导入到wyp表里面去了（关于这里面的执行过程大家可以参见本博客的《Hive表与外部表》），可以到wyp表的数据目录下查看，如下命令：

```
hive> dfs -ls /user/hive/warehouse/wyp ;
Found 1 items
-rw-r--r--3 wyp supergroup 67 2014-02-19 18:23 /hive/warehouse/wyp/wyp.txt
```

数据的确导入到wyp表里面去了。

和我们熟悉的关系型数据库不一样，Hive现在还不支持在insert语句里面直接给出一组记录的文字形式，也就是说，Hive并不支持INSERT INTO .... VALUES形式的语句。

二、HDFS上导入数据到Hive表

从本地文件系统中将数据导入到Hive表的过程中，其实是先将数据临时复制到HDFS的一个目录下（典型的情况是复制到上传用户的HDFS home目录下,比如/home/wyp/），然后再将数据从那个临时目录下移动（注意，这里说的是移动，不是复制！）到对应的Hive表的数据目录里面。既然如此，那么Hive肯定支持将数据直接从HDFS上的一个目录移动到相应Hive表的数据目录下，假设有下面这个文件/home/wyp/add.txt，具体的操作如下：

```
[wyp@master /home/q/hadoop-2.2.0]$ bin/hadoop fs -cat /home/wyp/add.txt
5	wyp1	23	131212121212
6	wyp2	24	134535353535
7	wyp3	25	132453535353
8	wyp4	26	154243434355
```

上面是需要插入数据的内容，这个文件是存放在HDFS上/home/wyp目录（和一中提到的不同，一中提到的文件是存放在本地文件系统上）里面，我们可以通过下面的命令将这个文件里面的内容导入到Hive表中，具体操作如下：

```
hive> load data inpath '/home/wyp/add.txt' into table wyp;
Loading data to table default.wyp
Table default.wyp stats:
[num_partitions: 0, num_files: 2, num_rows: 0, total_size: 215]
OK
Time taken: 0.47 seconds

hive> select * from wyp;
OK
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
Time taken: 0.096 seconds, Fetched: 7 row(s)
```

从上面的执行结果我们可以看到，数据的确导入到wyp表中了！请注意load data inpath '/home/wyp/add.txt' into table wyp;里面是没有local这个单词的，这个是和一中的区别。

三、从别的表中查询出相应的数据并导入到Hive表中

假设Hive中有test表，其建表语句如下所示：

```
hive> create table test(
    > id int, name string
    > ,tel string)
    > partitioned by
    > (age int)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '₩t'
    > STORED AS TEXTFILE;
OK
```

```
Time taken: 0.261 seconds
```

大体和wyp表的建表语句类似，只不过test表里面用age作为了分区字段（关于什么是分区字段，请参见本博客的《Hive的数据存储模式》
中的介绍，其详细的介绍本博客将会在接下来的时间内介绍，请关注本博客！）。下面语句就是将wyp表中的查询结果并插入到test表中：

```
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)
```

通过上面的输出，我们可以看到从wyp表中查询出来的东西已经成功插入到test表中去了！如果目标表（test）中不存在分区字段，可以去掉partition
(age='25')语句。当然，我们也可以在select语句里面通过使用分区值来动态指明分区：

```
hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> insert into table test
    > partition (age)
    > select id, name,
    > tel, age
```

```
    > 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 510 msec
OK
Time taken: 17.712 seconds


hive> select * from test;
OK
5       wyp1    131212121212    23
6       wyp2    134535353535    24
7       wyp3    132453535353    25
1       wyp     13188888888888  25
8       wyp4    154243434355    26
2       test    13888888888888  30
3       zs      899314121       34
Time taken: 0.399 seconds, Fetched: 7 row(s)
```

这种方法叫做动态分区插入，但是Hive中默认是关闭的，所以在使用前需要先把hive.exec.dynamic.partition.mode设置为nonstrict。当然，Hive也支持insert overwrite方式来插入数据，从字面我们就可以看出，overwrite是覆盖的意思，是的，执行完这条语句的时候，相应数据目录下的数据将会被覆盖！而insert into则不会，注意两者之间的区别。例子如下：

```
hive> insert overwrite table test
    > PARTITION (age)
    > select id, name, tel, age
    > from wyp;
```

更可喜的是，Hive还支持多表插入，什么意思呢？在Hive中，我们可以把insert语句倒过来，把from放在最前面，它的执行效果和放在后面是一样的，如下：

```
hive> show create table test3;
OK
CREATE  TABLE test3(
  id int,
  name string)
```

```
Time taken: 0.277 seconds, Fetched: 18 row(s)

hive> from wyp
    > insert into table test
    > partition(age)
    > select id, name, tel, age
    > insert into table test3
    > select id, name
    > where age>25;

hive> select * from test3;
OK
8       wyp4
2       test
3       zs
Time taken: 4.308 seconds, Fetched: 3 row(s)
```

可以在同一个查询中使用多个insert子句，这样的好处是我们只需要扫描一遍源表就可以生成多个不相交的输出。这个很酷吧！

四、在创建表的时候通过从别的表中查询出相应的记录并插入到所创建的表中

在实际情况中，表的输出结果可能太多，不适于显示在控制台上，这时候，将Hive的查询输出结果直接存在一个新的表中是非常方便的，我们称这种情况为CTAS（create table .. as select）如下：

```
hive> create table test4
    > as
    > select id, name, tel
    > from wyp;

hive> select * from test4;
OK
5       wyp1    131212121212
6       wyp2    134535353535
7       wyp3    132453535353
8       wyp4    154243434355
1       wyp     13188888888888
2       test    13888888888888
3       zs      899314121
Time taken: 0.089 seconds, Fetched: 7 row(s)
```

数据就插入到test4表中去了，CTAS操作是原子的，因此如果select查询由于某种原因而失败，新表是不会创建的！
好了，很晚了，今天就到这，洗洗睡！2014年2月20日 00:59:17