Hubble数据导入示例

1.hubble导入普通表，数据处理流程

1.1 示例数据

```
#文本字段含义
第一列：rowid（行号）
第二列：name（姓名）
第三列：age（年龄）
第四列：gender（性别）

#文件名称
simple_example.dat

#文件内容
1,张三,24,男
2,李四,26,男
3,王五,56,男
4,寒梅,18,女
5,李蕾,15,女
```

1.2 数据上传到hdfs操作

```
1.创建hdfs，数据目录
hdfs dfs -mkdir -p /data/example/simple

2.上传本地数据文件到hdfs
hdfs dfs -put /data/example/simple/simple_example.dat /data/example/simple/

3.查看文件是否上传成功
hdfs dfs -ls /data/example/simple
```

1.3 加载数据操作

1.在hive数据库中创建数据库（hive cli下执行）

```
create database example;

#验证是否创建成功
show databases;
```

2.在hive数据库中创建外部表（外部表删表不删数据）

```
drop table if exists example.simple_example;
create external table example.simple_example
(
    rowid int,
    name string,
    age int,
    gender string
)
row format delimited fields terminated by ','--指定字段分隔符
STORED AS textfile;--指定 存储格式

#验证建表是否成功
use example;
#查看表是否被创建
show tables;
#查看表结构
desc simple_example;
#查看建表语句
show create table simple_example;
#删除表
drop table example.simple_example;
```

注：外部表手动需要删除hdfs的数据

3.将数据加载到hive表中

```
LOAD DATA INPATH '/data/example/simple/' INTO TABLE example.simple_example;
```

4.验证是否加载成功

```
select * from simple_example limit 1;
hive> select * from simple_example limit 1;
OK
1   张三  24  男
2   李四  26  男
3   王五  56  男
4   寒梅  18  女
5   李蕾  15  女
```

数据在hive中能够正常查询，数据加载完毕

1.4 导入数据到hubble操作

1.创建hubble对应的orc库

```
create database example_orc;
```

2.创建Hubble对应的orc表

```
    drop table if exists example_orc.simple_example_orc;
    create external table example_orc.simple_example_orc
    (
        rowid int,
        name string,
        age int,
        gender string
    )
    row format delimited fields terminated by '/001'--指定字段分隔符
    STORED AS orc;--指定 存储格式

    #验证建表是否成功
    use example_orc;
    #查看表是否被创建
    show tables;
    #查看表结构
    desc simple_example_orc;
    #查看建表语句
    show create table simple_example_orc;
    #删除表
    drop table example.simple_example_orc;
```

3.查看表结构，是否转换成

```
    desc simple_example_orc;
```

##########################下面使用HUBBLE数据库进行数据操作##########################

备注：通过hubble-sql.sh进入到hubble cli模式进行操作

3.数据转换成orc格式，转换操作

```
    insert into example_orc.simple_example_orc select * from example.simple_example;
```

4.统计数据条数，是否正确导入

```
    select count(*) from hive.example_orc.simple_example_orc;
```

2.hubble导入分区表，数据处理流程

2.1 示例数据

```
#文件名称
DEMO-20180217101348.txt
DEMO-20180218101449.txt
```

```
DEMO-20180219101550.txt
DEMO-20180220101651.txt
DEMO-20180221101752.txt
DEMO-20180222101853.txt

#文本字段含义
第一列：rowid（行号）
第二列：name（姓名）
第三列：age（年龄）
第四列：gender（性别）

#文件内容
1,张三,24,男
2,李四,26,男
3,王五,56,男
4,寒梅,18,女
5,李蕾,15,女
6,李涵,18,男
```

2.2 数据上传到hdfs操作

```
1.创建hdfs，数据目录
hdfs dfs -mkdir -p /data/example/partition

2.上传本地数据文件到hdfs
hdfs dfs -put /data/example/partition/DEMO-20180217101348.txt /data/example/partition/
hdfs dfs -put /data/example/partition/DEMO-20180218101449.txt /data/example/partition/
hdfs dfs -put /data/example/partition/DEMO-20180219101550.txt /data/example/partition/
hdfs dfs -put /data/example/partition/DEMO-20180220101651.txt /data/example/partition/
hdfs dfs -put /data/example/partition/DEMO-20180221101752.txt /data/example/partition/
hdfs dfs -put /data/example/partition/DEMO-20180222101853.txt /data/example/partition/

3.查看文件是否上传成功
hdfs dfs -ls /data/example/partition
```

2.3 加载数据操作

```
1.在hive数据库中创建按数据库（hive cli下执行）

    create database example;

    #验证是否创建成功
    show databases;

2.在hive数据库中创建外部表（外部表删表不删数据）

    drop table if exists example.partition_example;
    create external table example.partition_example
    (
        rowid int,
        name string,
```

```
        age int,
        gender string
    )
    partitioned by(partition_date string)  --指定分区
    row format delimited fields terminated by ','--指定字段分隔符
    STORED AS textfile;--指定 存储格式

    #进入操作数据库
    use example;

    #查看表是否被创建
    show tables;

    #查看表结构
    desc partition_example;

    #查看建表语句
    show create table partition_example;
```

3.将数据加载到hive表中

```
LOAD DATA INPATH '/data/example/partition/DEMO-20180217101348.txt' INTO TABLE
example.partition_example PARTITION(partition_date='date-20180217');
LOAD DATA INPATH '/data/example/partition/DEMO-20180218101449.txt' INTO TABLE
example.partition_example PARTITION(partition_date='date-20180218');
LOAD DATA INPATH '/data/example/partition/DEMO-20180219101550.txt' INTO TABLE
example.partition_example PARTITION(partition_date='date-20180219');
LOAD DATA INPATH '/data/example/partition/DEMO-20180220101651.txt' INTO TABLE
example.partition_example PARTITION(partition_date='date-20180220');
LOAD DATA INPATH '/data/example/partition/DEMO-20180221101752.txt' INTO TABLE
example.partition_example PARTITION(partition_date='date-20180221');
LOAD DATA INPATH '/data/example/partition/DEMO-20180222101853.txt' INTO TABLE
example.partition_example PARTITION(partition_date='date-20180222');
```

4. 查看分区

```
    #此操作在hive中执行
    show partitions example.partition_example;

    hive> show partitions partition_example;
    OK
    partition_date=date-20180217
    partition_date=date-20180218
    partition_date=date-20180219
    partition_date=date-20180220
    partition_date=date-20180221
    partition_date=date-20180222
```

5.验证分区数据是否加载成功

```
    #根据分区查找数据
    select * from partition_example where partition_date='date-20180217' limit 1;
```

```
    #随机查询返回数据
    select * from partition_example limit 10;


##########################下面使用HUBBLE数据库进行数据操作##########################

备注：通过hubble-sql.sh进入到hubble cli模式进行操作

3.数据转换成orc格式，转换操作

    create table hive.example_orc.simple_example_orc with(partitioned_by =
ARRAY['partition_date']) as select * from hive.example_orc.simple_example;

    备注：建表并全量导入分区数据。

4.统计数据条数，是否正确导入

    select count(*) from hive.example_orc.simple_example_orc;ss

6.删除表操作

    drop table example.partition_example;
    注：删除表结构并没有删除表数据，外部表需要手动删除hdfs对应的表数据
```