Hive Homework Tutorial

注意：以下操作路径均为本人电脑上的操作路径，同学们操作时需要对相应路径根据自己电脑情况进行修改。

1. prepare file

```
hadoop fs -mkdir /user/chenguang/hive/yahooFinance
hadoop fs -copyFromLocal Stock.csv /user/chenguang/hive/yahooFinance/
hadoop fs -ls /user/chenguang/hive/yahooFinance
```

2. Open the Hive

You can either use Hive Line, Beeline or Ambari

3. create databases

```
CREATE DATABASE IF NOT EXISTS chen_db[Database name: suggest name: username_db];
```

select database and show tables

```
USE chen_db;
SHOW TABLES;
```

4. create temp and yahooFinance table

Here we create a temp table because we need to deal with the date format (from MM/DD/YYYY to YYYY-MM-DD)

```
## Hive provides DATE and TIMESTAMP data types for date related fields
## DATE values are represented in the form YYYY-MM-DD. Date ranges allowed are 0000-01-01 to 9999-12-31
## TIMESTAMP uses the format yyyy-mm-dd hh:mm:ss

DROP TABLE IF EXISTS temp;
CREATE TABLE temp (
stockDate STRING,
Name STRING,
open FLOAT,
high FLOAT,
low FLOAT,
close FLOAT,
volume BIGINT,
adjClose FLOAT
)
ROW  FORMAT  DELIMITED
FIELDS  TERMINATED  BY  ','
LINES  TERMINATED  BY  '\n'
STORED AS TEXTFILE
tblproperties ("skip.header.line.count"="1");
```

Load data into table temp

```
LOAD DATA INPATH '/user/chenguang/hive/yahooFinance/Stock.csv'  OVERWRITE INTO TABLE temp;
LOAD DATA LOCAL INPATH '/home/chenguang/data/Stock.csv' OVERWRITE INTO TABLE temp;
```

Check schema and content

```
DESCRIBE temp;
SELECT * FROM temp limit 5;
```

Answer of Q1: Check how many rows are inserted

```
SELECT COUNT(*) FROM temp;
```

Build the table of yahooFinance and insert the data from table temp

```
DROP TABLE IF EXISTS yahooFinance;

create table yahooFinance(
stockDate DATE,
Name STRING,
open FLOAT,
high FLOAT,
low FLOAT,
close FLOAT,
volume BIGINT,
adjClose FLOAT
);
```

```
select from_unixtime(unix_timestamp('02/22/2015' ,'MM/dd/yyyy'), 'yyyy-MM-dd');
select TO_DATE(from_unixtime(unix_timestamp('02/22/2015' ,'MM/dd/yyyy'))) from temp;
```

```
insert overwrite table yahooFinance
select TO_DATE(from_unixtime(UNIX_TIMESTAMP(stockdate,'MM/dd/yy'))),
Name,
open,
high,
low,
close,
volume,
adjClose from temp;
```

5. Create a Partitioned Table and load data into it

```
hadoop fs -mkdir /user/chenguang/hive/yahooFinance/partition
```

```
DROP TABLE IF EXISTS PartitionedYahooFinance;
CREATE TABLE PartitionedYahooFinance(
stockDate DATE,
Name STRING,
open FLOAT,
high FLOAT,
low FLOAT,
close FLOAT,
volume BIGINT,
adjClose FLOAT
)
COMMENT 'This is the Partitioned Yahoo Finance Data'
PARTITIONED BY(year STRING)
```

```
ROW  FORMAT  DELIMITED
FIELDS  TERMINATED  BY  ','
STORED AS TEXTFILE
LOCATION '/user/chenguang/hive/yahooFinance/partition/';
```

check if there is any partition

warning: Table yahoofinance is not a partitioned table

```
SHOW PARTITIONS PartitionedYahooFinance;
```

Add all the data from table yahooFinance to PartitionedYahooFinance

```
INSERT OVERWRITE TABLE PartitionedYahooFinance
PARTITION (year = "Before 2003")
SELECT * FROM yahooFinance WHERE stockDate < '2003-01-01';

INSERT OVERWRITE TABLE PartitionedYahooFinance
PARTITION (year = "Between 2003 and 2009")
SELECT * FROM yahooFinance WHERE stockDate > '2002-12-31' AND stockDate < '2010-01-01';
```

Check how many rows are inserted

```
SELECT COUNT(*) FROM PartitionedYahooFinance;

INSERT OVERWRITE TABLE PartitionedYahooFinance
PARTITION (year = "After 2009")
SELECT * FROM yahooFinance WHERE stockDate > '2009-12-31';
```

Check how many rows are inserted

```
SELECT COUNT(*) FROM PartitionedYahooFinance;
```

Check how many rows are in yahooFinance

```
SELECT COUNT(*) FROM yahooFinance;

DESCRIBE PartitionedYahooFinance;

SHOW PARTITIONS PartitionedYahooFinance;
```

Add a partition

```
ALTER TABLE PartitionedYahooFinance ADD IF NOT EXISTS PARTITION (year = 'After 2016');
```

check partitions

```
SHOW PARTITIONS PartitionedYahooFinance;
```

Drop a partition

```
ALTER TABLE PartitionedYahooFinance DROP IF EXISTS PARTITION(year = 'After 2016');
```

Check partitions

```
SHOW PARTITIONS PartitionedYahooFinance;
```

Answer of Q2:

Now you know how to deal with the partition of the table. The way to answer Q2 is:

```
INSERT OVERWRITE TABLE PartitionedYahooFinance
PARTITION (year = "2008")
SELECT * FROM yahooFinance WHERE stockDate > '2007-12-31' AND stockDate < '2009-01-01';
```

所有的股票信息存在 year = "2008"这个 partition 中。

```
SELECT * FROM partitionedyahoofinance where year = '2008';
```

模糊查询：

```
INSERT OVERWRITE TABLE PartitionedYahooFinance

PARTITION (year = "blur searching 2008")

SELECT * FROM yahooFinance WHERE (stockDate like '2008%');
```

7. External vs Internal Table

## The EXTERNAL keyword lets you create a table and provide a LOCATION so that Hive does not use a default location for this table. This comes in handy if you already have data generated.

## When dropping an EXTERNAL table, data in the table is NOT deleted from the file system.

## When you drop a table, if it is managed table hive deletes both data and meta data, if it is external table Hive only deletes metadata.

```
hadoop fs -mkdir /user/chenguang/hive/yahooFinance/external
hadoop fs -copyFromLocal Stock.csv /user/chenguang/hive/yahooFinance/external/
hadoop fs -mkdir /user/chenguang/hive/yahooFinance/internal
hadoop fs -copyFromLocal Stock.csv /user/chenguang/hive/yahooFinance/internal/
```

```
DROP TABLE IF EXISTS ExternalYahooFinance;
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS ExternalYahooFinance(
stockDate STRING,
Name STRING,
open FLOAT,
high FLOAT, low
FLOAT, close
FLOAT, volume
BIGINT,
adjClose FLOAT
)
COMMENT 'This is the External Yahoo Finance Table'
ROW FORMAT DELIMITED FIELDS TERMINATED
BY ','
STORED AS textfile
LOCATION '/user/chenguang/hive/yahooFinance/external/';
## 文件夹里的.cvs 直接会导入表格中

SHOW TABLES;
```

See table type

```
DESCRIBE FORMATTED ExternalYahooFinance;
```

Drop the table ExternalYahooFinance

```
DROP TABLE IF EXISTS ExternalYahooFinance;
SHOW TABLES;
```

Check if the file is still there

```
hadoop fs -ls /user/chenguang/hive/yahooFinance/external
## Stock.csv 文件仍然存在

DROP TABLE IF EXISTS InternalYahooFinance;
CREATE TABLE IF NOT EXISTS InternalYahooFinance(
stockDate STRING,
Name STRING,
open FLOAT,
high FLOAT,
low FLOAT,
close FLOAT,
volume BIGINT,
adjClose FLOAT
)
COMMENT 'This is the External Yahoo Finance Table'
ROW FORMAT DELIMITED FIELDS TERMINATED
BY ','
STORED AS textfile
LOCATION '/user/chenguang/hive/yahooFinance/internal/';
```

See table type

```
DESCRIBE FORMATTED InternalYahooFinance;

## This table is connected with the file in the hdfs path, the table is empty if no file in the hdfs
## path
hadoop fs -rm /user/chenguang/hive/yahooFinance/internal/Stock.csv
SELECT COUNT(*) FROM internalyahoofinance;

hadoop fs -copyFromLocal Stock.csv /user/chenguang/hive/yahooFinance/internal/
```

Check file is there

```
hadoop fs -ls /user/chenguang/hive/yahooFinance/internal
SELECT COUNT(*) FROM internalyahoofinance;
```

Drop internal table

```
DROP TABLE IF EXISTS InternalYahooFinance;
SHOW TABLES;
```

The whole directory will be deleted

```
hadoop fs -ls /user/chenguang/hive/yahooFinance/internal/
```

Switch a table from internal to external.

```
ALTER TABLE table_name SET TBLPROPERTIES('EXTERNAL'='TRUE');
```

Switch a table from external to internal.

```
ALTER TABLE table_name SET TBLPROPERTIES('EXTERNAL'='FALSE');
```

1. recreate InternalYahooFinance

2. copy csv file to internal hdfs folder

```
hadoop fs -mkdir /user/chenguang/hive/yahooFinance/internal
hadoop fs -copyFromLocal Stock.csv /user/chenguang/hive/yahooFinance/internal/
```

3. Switch InternalYahooFinance from internal to external

```
ALTER TABLE InternalYahooFinance SET TBLPROPERTIES('EXTERNAL'='TRUE');
```

4. Drop InternalYahooFinance

```
DROP TABLE InternalYahooFinance;
```

5. Check csv file should still be in the internal hdfs folder

```
hadoop fs -ls /user/chenguang/hive/yahooFinance/internal
```

Answer of Q3:

From above steps you will solve this question.

Answer of Q4:

```
SELECT * FROM yahoofinance ORDER BY open DESC LIMIT 1;
SELECT CAST(MAX(open) as FLOAT) FROM yahoofinance;
```