CS 624 -- HW 1

Due date: Sunday, Feb 19th at 11:59pm (EST)

Objectives: Practice SQL in Hive.

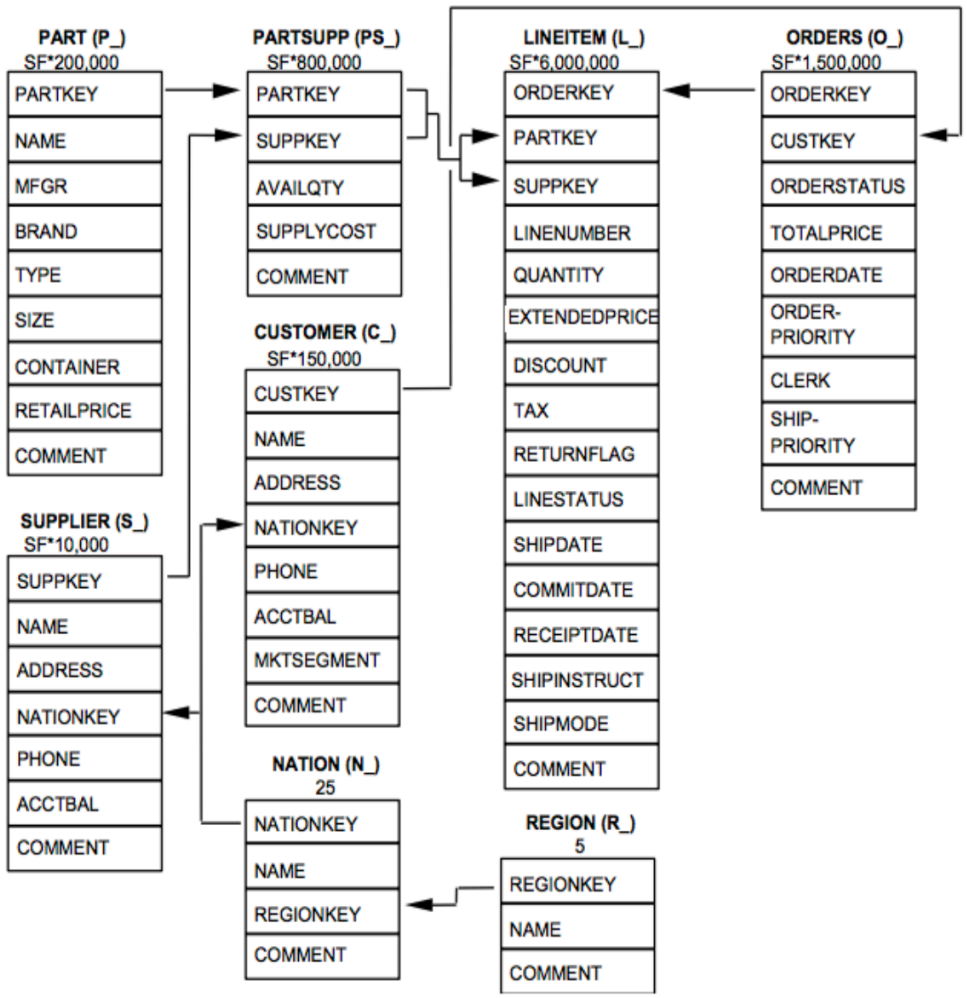
Assignment tools: Hive in ODU's Research Cloud Computing

Assignment Details

In this Assignment we will practice importing data into Hive and run various queries in Hive Interactive Shell.

1. Copy data and Load Data into Hive Tables (10 points)

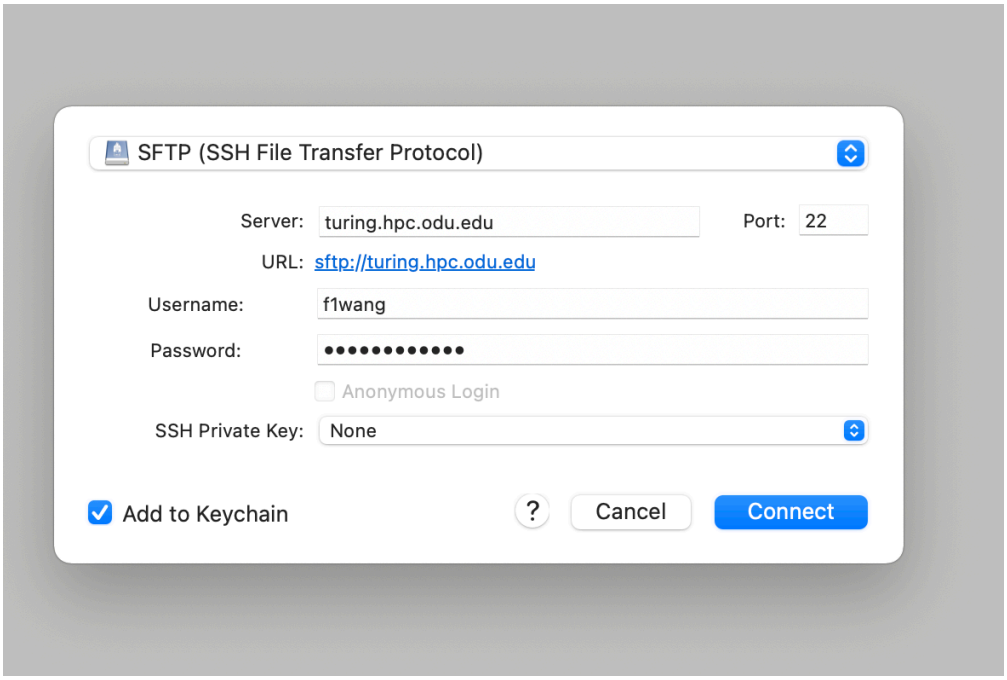
We'll work with TPC-H data. TPC-H is a standard decision support benchmark for big data analytics. It contains synthetic data and several queries representing a generalized complex analytics workload that answers critical business questions. The data size and distribution can be controlled via arguments while generating the synthetic data. Below is the schema of the benchmark database. The size of the dataset used in this assignment is 1 GB. You can learn more about TPC-H at: http://www.tpc.org/tpch/.



Dataset size:

table	number of tuples
customer	150000
lineitem	6001215
nation	25
orders	1500000
part	200000
partsupp	800000
region	5
supplier	10000

1. Download data (cs624_hw1_data.zip) from Google Drive to your local machine. Google Drive link: https://drive.google.com/file/d/1TlZBu6oH2W7qJQB7fM8XG1e8BzJXwKf3/view?usp=sharing. Please use your ODU email account to access the data.
2. Copy data from local computer to Turing cluster or Wahab cluster. There are two types of approaches:
    1. Use an app like cyberduck to transfer files from local machine to the cluster. After installing the app, the only step is to establish a connection between local machine and the server. Following figure shows what information you should input. Choose SFTP (SSH ...), input Server either turing.hpc.odu.edu or wahab.hpc.odu.edu, **your Username and password**. Then, click connect. Authentication may needed for connection.



After the successful connection, you will see something similar as the figure shown below. It shows all files or folders in your HPC account. Now, you can drag your local files to the cyberduck window to transfer files.

2. Use scp command to copy local file to the clsuter. If you use Windows system, you need to download Cygwin app and run command in Cygwin. Sample scp command: `scp cs624_hw1_data.zip netID@wahab.hpc.odu.edu:~/cs624_hw1_data.zip`.

3. Import data into Hive tables:

    1. SSH connect to the cluster. Unzip the .zip file in the cluster by running command `unzip cs624_hw1_data.zip`.

        1. If you don't want to run SSH in your local machine, you can use ondemand.wahab. Go to the website https://ondemand.wahab.hpc.odu.edu/pun/sys/dashboard, choose Clustsers/Wahab Shell Access, then you can connect to cluster and run all commands using your browser.

    2. Setup the environment. Details can be found in the second demo video in week2's content in Canvas.

    3. We will need to import table one by one into the Hive. We provide sample code in the following for imporing "customer" table, then you can figure out the rest. For this example, we will need customer.json file. The idea is to import data in customer.json file into a table "rawdata" first. "rawdata" table only has one string column. Then we will parse data in "rawdata" to insert the data into the table "customer". In the last command, we used a function "get_json_object", the second argument in this function is the attribute name used in json file. To find out what attribute names are used in json file, you can run command `head -5 customer.json` after connected to the cluster.

```
DROP TABLE IF EXISTS rawdata;
CREATE TABLE rawdata(textcol string) STORED AS TEXTFILE;
LOAD DATA LOCAL INPATH 'customer.json' INTO TABLE rawdata;
DROP TABLE IF EXISTS customer;
CREATE TABLE customer (C_CustKey INT, C_Name STRING, C_Address STRING, C_NationKey
INT, C_Phone STRING, C_AcctBal STRING, C_MktSegment STRING, C_Comment STRING);
INSERT OVERWRITE TABLE customer SELECT get_json_object(textcol, '$.C_CustKey') as
C_CustKey, get_json_object(textcol, '$.C_Name') as C_Name, get_json_object(textcol,
'$.C_Address') as C_Address, get_json_object(textcol, '$.C_NationKey') as C_NationKey,
get_json_object(textcol, '$.C_Phone') as C_Phone, get_json_object(textcol,
'$.C_AcctBal') as C_AcctBal, get_json_object(textcol, '$.C_MktSegment') as
C_MktSegment, get_json_object(textcol, '$.C_Comment') as C_Comment FROM rawdata;
```

For importing other tables, you can follow the pattern in the above code. Here we will just provide the CREATE TABLE command for all tables. Please use these command when creating tables.

```
CREATE TABLE lineitem(L_OrderKey INT, L_PartKey INT, L_SuppKey INT, L_LineNumber INT,
L_Quantity INT, L_ExtendedPrice FLOAT, L_Discount FLOAT, L_Tax FLOAT, L_ReturnFlag
STRING, L_LineStatus STRING, L_ShipDate DATE, L_CommitDate DATE, L_ReceiptDate DATE,
L_ShipInstruct STRING, L_ShipMode STRING, L_Comment STRING);
CREATE TABLE nation(N_NationKey INT, N_Name STRING, N_RegionKey INT, N_Comment
STRING);
CREATE TABLE orders(O_OrderKey INT, O_CustKey INT, O_OrderStatus STRING, O_TotalPrice
FLOAT, O_OrderDate DATE, O_OrderPriority STRING, O_Clerk STRING, O_ShipPriority INT,
O_Comment STRING);
CREATE TABLE part(P_PartKey INT, P_Name STRING, P_Mfgr STRING, P_Brand STRING, P_Type
STRING, P_Size INT, P_Container STRING, P_RetailPrice FLOAT, P_Comment STRING);
CREATE TABLE partsupp(PS_PartKey INT, PS_SuppKey INT, PS_AvailQty INT, PS_SupplyCost
FLOAt, PS_Comment STRING);
CREATE TABLE region(R_RegionKey INT, R_Name STRING, R_Comment STRING);
CREATE TABLE supplier(S_SuppKey INT, S_Name STRING, S_Address STRING, S_NationKey INT,
S_Phone STRING, S_AcctBal FLOAT, S_Comment STRING);
```

Tip: you can use command `show tables;` to verify whether all tables have been created successfully. You can also verify whether data has been imported correctly by examine the data in the tables by command `select * from customer limit 5;`

2. Query Questions (90 points)

Write and run queries on the dataset. For each question, please only use one query.

1. What is the total number of parts offered by each supplier? The query should return the name of the supplier and the total number of parts. (5 points)
2. What is the cost of the most expensive part by any supplier? The query should return the price of that most expensive part. No need to return the name of the part or the name of the supplier. (10 points)
3. What is the cost of the most expensive part for each supplier? The query should return the name of the supplier and the cost of the most expensive part. No need to return the name of that part. (10 points)
4. What is the total number of customers per nation? The query should return the name of the nation and the number of unique customers. (10 points)
5. What is number of parts shipped between Oct. 8th, 1996 and Nov. 8th, 1996 for each supplier (shipped on Oct. 8th 1996 and Nov. 8th 1996 should also be considered)? The query should return the name of the supplier and the number of parts. (10 points)
6. Find out the list of customers who ordered some parts where the retail price of those parts are smaller than 500. The query should return custkey. (10 points)
7. Who is the customer with highest total order price per nation? The query should return the name of the customer, the name of the nation and total order price. For each nation, there is only one customer who has the highest total order price. (20 points)
8. A customer is considered a *Gold* customer if they have orders with total price more than $2,000,000. Customers have orders with total price between $1,000,000 and $2,000,000 are considered *Silver* customer. Write a single SQL query to compute the number of customers in these two categories. Hint: you may want to use Case statement in SQL. Please refer to this link (https://www.w3schools.com/sql/sql_

case.asp) for examples of how to use Case statement in SQL. (15 points)

Submission

Submit everything as a single word document through Canvas assignment entry.

What to turn in:

- To earn credit for successfully load data into hive tables. Please run command `show tables;`. Please include a screenshot (like below) in your report.

```
hive (f1wang)> set hiveconf:hive.cli.print.current.db=true;
hive (f1wang)> show tables;
OK
clicks
customer
documents_meta
documents_topics
events
lineitem
nation
orders
part
partsupp
promoted_content
rawdata
region
state_part
supplier
temp2
yelp_business
yelp_review
yelp_tip
yelp_user
Time taken: 0.225 seconds, Fetched: 20 row(s)
```

- To answer each question, please include the following information:
  - SQL query
  - average runtime for each query. Make sure you run the query three times.
  - Number of rows returned
  - First 5 rows from the result (or all rows if a query returns fewer than 5 rows). You can use command `limit 5` at the end of the query to retrieve top 5 results.