

Assignment 8.1(Hive Basics)

Task 1 :

Create a database named 'custom'.

Create a table named temperature_data inside custom having below fields:

1. date (mm-dd-yyyy) format
2. zip code
3. temperature

The table will be loaded from comma-delimited file.

Load the dataset.txt (which is ',' delimited) in the table.

Query for table creation:

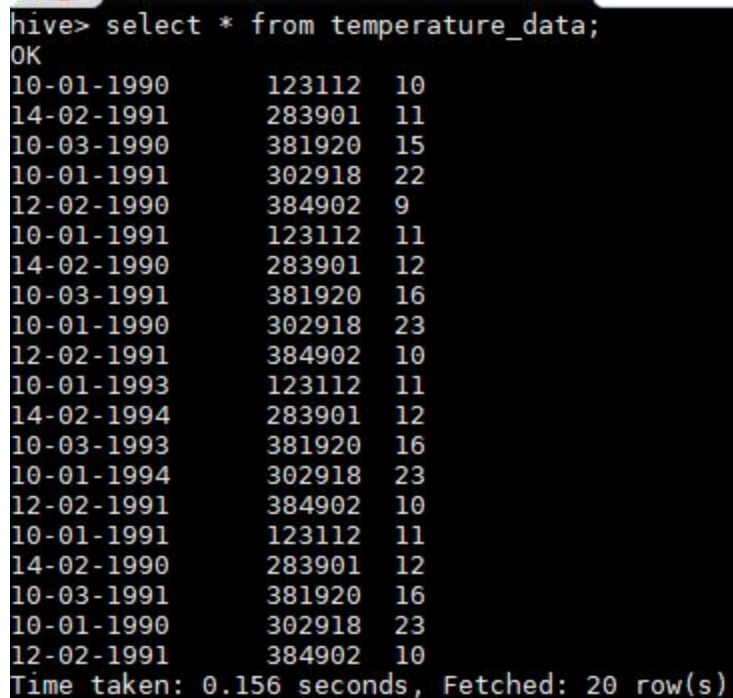
```
create table temperature_data(date String, zip Int,temp Int) row format delimited fields terminated by ',';
```

Query for loading data into the table:

```
LOAD DATA LOCAL INPATH '/home/acadgild/dataset_Session 14.txt' into table temperature_data;
```

Output Screenshot:

Table created containing the loaded data:



```
hive> select * from temperature_data;
OK
10-01-1990      123112  10
14-02-1991      283901  11
10-03-1990      381920  15
10-01-1991      302918  22
12-02-1990      384902   9
10-01-1991      123112  11
14-02-1990      283901  12
10-03-1991      381920  16
10-01-1990      302918  23
12-02-1991      384902  10
10-01-1993      123112  11
14-02-1994      283901  12
10-03-1993      381920  16
10-01-1994      302918  23
12-02-1991      384902  10
10-01-1991      123112  11
14-02-1990      283901  12
10-03-1991      381920  16
10-01-1990      302918  23
12-02-1991      384902  10
Time taken: 0.156 seconds, Fetched: 20 row(s)
```

Task 2

1. Fetch date and temp from temperature_data where zip code is greater than 300000 and less than 399999.

Query for the above task:

select dat, temperature from temperature_data where zip > 300000 and zip < 399999.

Output:

```
hive> select dat, temp from temperature_data where zip > 300000 and zip < 399999;
OK
10-03-1990      15
10-01-1991      22
12-02-1990       9
10-03-1991      16
10-01-1990      23
12-02-1991      10
10-03-1993      16
10-01-1994      23
12-02-1991      10
10-03-1991      16
10-01-1990      23
12-02-1991      10
Time taken: 0.714 seconds, Fetched: 12 row(s)
```

2. Calculate maximum temperature corresponding to every year from temperature_data table.

Query for the above task:

select SUBSTRING(dat,7,4) as year, MAX(temp) as temperature
from temperature_data
GROUP BY SUBSTRING(dat,7,4);

Output:

```
hive> select SUBSTRING(dat,7,4) as year, MAX(temp) as temperature
> from temperature_data
> GROUP BY SUBSTRING(dat,7,4);
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i
.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180509135922_94e37c2a-c559-4c2c-9758-116132ffdd0a
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1525848959469_0001, Tracking URL = http://localhost:8088/proxy/application_1525848959469_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1525848959469_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-05-09 13:59:31,461 Stage-1 map = 0%, reduce = 0%
2018-05-09 13:59:36,741 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.19 sec
2018-05-09 13:59:41,968 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.67 sec
MapReduce Total cumulative CPU time: 3 seconds 670 msec
Ended Job = job_1525848959469_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.67 sec HDFS Read: 9049 HDFS Write: 167 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 670 msec
OK
1990 23
1991 22
1993 16
1994 23
Time taken: 20.211 seconds, Fetched: 4 row(s)
```

3. Calculate maximum temperature from temperature_data table corresponding to those years which have at least 2 entries in the table.

Query for the above task:

```
SELECT dat, MAX(t1.temp) as temperature FROM
(select SUBSTRING(dat,7,4) dat, temp from temperature_data) t1 GROUP BY dat HAVING
count(t1.dat) > 2;
```

Output:

```
hive> SELECT dat, MAX(t1.temp) as temperature FROM
> (select SUBSTRING(dat,7,4) dat, temp from temperature_data) t1 GROUP BY dat HAVING count(t1.dat) > 2;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i
.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180509140244_c066394f-69d3-4a2a-9193-a68e50c8c21a
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1525848959469_0002, Tracking URL = http://localhost:8088/proxy/application_1525848959469_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1525848959469_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-05-09 14:02:50,046 Stage-1 map = 0%, reduce = 0%
2018-05-09 14:02:54,240 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.8 sec
2018-05-09 14:02:59,451 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.04 sec
MapReduce Total cumulative CPU time: 4 seconds 40 msec
Ended Job = job_1525848959469_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.04 sec HDFS Read: 10077 HDFS Write: 127 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 40 msec
OK
1990 23
1991 22
Time taken: 15.559 seconds, Fetched: 2 row(s)
```

3. Create a view on the top of last query, name it temperature_data_vw.

Query for the above task:

create view temperature_data_vw as SELECT dat,
MAX(t1.temp) as temperature FROM
(select SUBSTRING(dat,7,4) dat, temp from temperature_data) t1 GROUP BY dat HAVING
count(t1.dat) > 2;

Output:

```
-----  
hive> show tables;  
OK  
olympic_data  
temperature_data  
temperature_data_vw  
Time taken: 0.08 seconds, Fetched: 3 row(s)  
hive> █
```

4. Export contents from temperature_data_vw to a file in local file system, such that each file is '|' delimited

Query for the above task:

INSERT OVERWRITE LOCAL DIRECTORY '/home/acadgild/temperature_output' ROW
FORMAT DELIMITED FIELDS TERMINATED BY '|'

Output:

```
-----  
[acadgild@localhost temperature_output]$ ls-lrt  
-bash: ls-lrt: command not found  
[acadgild@localhost temperature_output]$ ls -lrt  
total 4  
-rw-r--r--. 1 acadgild acadgild 16 May  5 00:31 000000_0  
[acadgild@localhost temperature_output]$ cat 000000_0  
1990|23  
1991|22  
[acadgild@localhost temperature_output]$ █
```