

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:

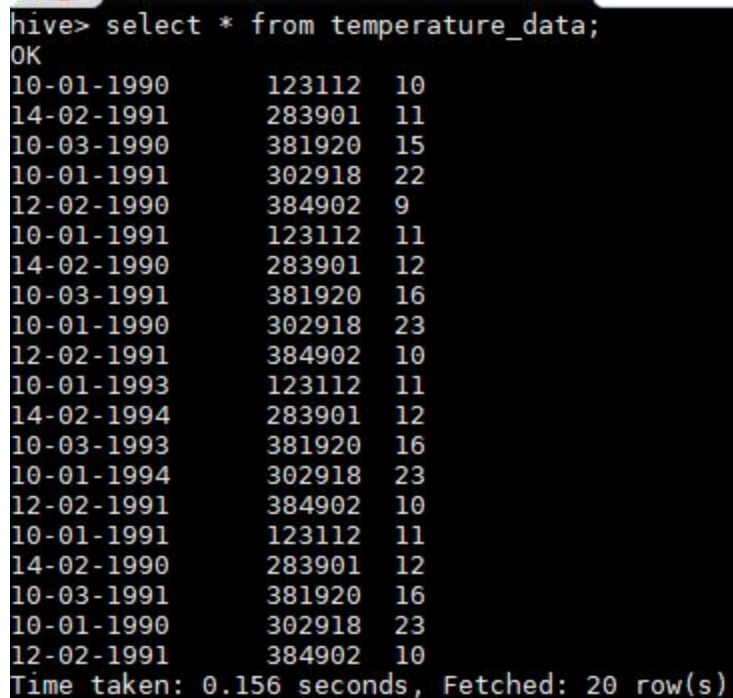
```
create table temperature_data(date String, zip Int,temp Int) row format delimited fields terminated by ',';
```

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:

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:

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:

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:
