Task1

Created ‘custom’ database

Output-

```
hive> create database custom;
OK
Time taken: 0.354 seconds
hive> show databases;
OK
custom
default
retail
Time taken: 0.153 seconds, Fetched: 3 row(s)
```

Created ‘temperature_data’ table in custom database. Loaded data from text file into the table

Output-

```
hive> CREATE TABLE temperature_data
    > ( dateformat string,zipcode bigint,temperature int) row format delimited fi
elds terminated by ',' stored as textfile;
OK
Time taken: 0.155 seconds
hive> load data local inpath '/home/acadgild/Hive/Custom/dataset_Session 14.txt'
overwrite into table temperature_data;
Loading data to table custom.temperature_data
OK
Time taken: 0.749 seconds
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
```

Task 2

---

a) Fetch date and temperature from temperature_data where zip code is greater than 300000 and less than 399999.

Script-

SELECT dateformat,temperature FROM temperature_data WHERE zipcode>300000 AND zipcode<399999;

Output-

```
hive> SELECT dateformat,temperature FROM temperature_data WHERE zipcode>300000 AN
D zipcode<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: 3.723 seconds, Fetched: 12 row(s)
hive>
```

b) Calculate maximum temperature corresponding to every year from temperature_data table.

Script –

SELECT YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'yyyy-MM-dd'))),MAX(temperature) FROM temperature_data GROUP BY YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'yyyy-MM-dd')))

```
hive> SELECT YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'
yyyy-MM-dd'))),MAX(temperature) FROM temperature_data GROUP BY YEAR(TO_DATE(from_
unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'yyyy-MM-dd')));
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futur
e versions. Consider using a different execution engine (i.e. spark, tez) or usin
g Hive 1.X releases.
Query ID = acadgild_20181125142230_10bcd4d3-3c4b-4f20-a578-baac0b270063
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:
```

Output-

```
1990    23
1991    22
1993    16
1994    23
Time taken: 74.424 secor
hive>
```

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

Script-

SELECT YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'yyyy-MM-dd'))),MAX(temperature) FROM temperature_data GROUP BY YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'yyyy-MM-dd'))) HAVING COUNT(YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'yyyy-MM-dd'))))>=2

```
hive> SELECT YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'
yyyy-MM-dd'))),MAX(temperature) FROM temperature_data GROUP BY YEAR(TO_DATE(from_
unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'yyyy-MM-dd')))
    > HAVING COUNT(YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yy
yy'),'yyyy-MM-dd'))))>=2;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futur
e versions. Consider using a different execution engine (i.e. spark, tez) or usin
```

Output-

```
1990    23
1991    22
1993    16
1994    23
Time taken: 74.424 secon
hive>
```

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

Script-

CREATE View Temperature_data_vw AS
SELECT YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'yyyy-MM-dd'))),MAX(temperature) FROM temperature_data GROUP BY YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'yyyy-MM-dd'))) HAVING COUNT(YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'yyyy-MM-dd'))))>=2;

Output-

```
hive> CREATE View Temperature_data_vw AS
    >  SELECT YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),
'yyyy-MM-dd'))),MAX(temperature) FROM temperature_data GROUP BY YEAR(TO_DATE(from
_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yyyy'),'yyyy-MM-dd')))
    > HAVING COUNT(YEAR(TO_DATE(from_unixtime(UNIX_TIMESTAMP(dateformat,'dd-MM-yy
yy'),'yyyy-MM-dd'))))>=2;
OK
Time taken: 1.093 seconds
hive> show tables;
OK
temperature_data
temperature_data_vw
Time taken: 0.147 seconds, Fetched: 2 row(s)
```

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

Script-

Insert overwrite local directory '/home/acadgild/Hive/Custom/out' row format delimited fields terminated by '|' SELECT * FROM Temperature_data_vw;

```
hive> Insert overwrite local directory '/home/acadgild/Hive/Custom/out' row forma
t delimited fields terminated by '|'  SELECT * FROM Temperature_data_vw;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futur
e versions. Consider using a different execution engine (i.e. spark, tez) or usin
g Hive 1.X releases.
Query ID = acadgild_20181125173535_3768a694-7563-4a22-951b-f3b8c2bba4aa
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>
```

Output-

```
[acadgild@localhost Retail]$ cat /home/acadgild/Hive/Custom/out/000000_0
1990|23
1991|22
1993|16
1994|23
[acadgild@localhost Retail]$
```