ASSIGNMENT 8.1

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.

SOLUTION - 1

```
                        • MobaXterm 10.4 •
            (SSH client, X-server and networking tools)

→ SSH session to acadgild@192.168.56.2
  • SSH compression : v
  • SSH-browser     : v
  • X11-forwarding  : v  (remote display is forwarded through SSH)
  • DISPLAY         : v  (automatically set on remote server)

→ For more info, ctrl+click on help or visit our website
```

#Launch HIVE

```
Last login: Tue Jul 24 11:57:05 2018 from 192.168.56.1
[acadgild.mmisra ~]$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in
[jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2
.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in
[jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log
4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in
jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.jar!/
hive-log4j2.properties Async: true
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.
hive>
```

# Creating database called custom

```
    >
    > create database custom;
OK
Time taken: 5.028 seconds
```

# switching to custom database

```
hive> use custom;
OK
Time taken: 0.027 seconds
```

#Creating a temporary table temp_temp to load the data into as the data has date in the DD-MM-YY format and we need date in MM-DD-YY format in table temperature_data

```
hive>
    > CREATE TABLE temp_temp(
    > ds STRING,
    > zip STRING,
    > temperature DECIMAL)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.63 seconds
```

#Loading data into the temporary file from the data set file

```
hive> LOAD DATA LOCAL INPATH 'dataset_Session 14.txt' INTO TABLE temp_temp;
Loading data to table custom.temp_temp
OK
Time taken: 2.0 seconds
```

# Now create the table temperature_data to hold the date in MM-DD-YYYY format

```
hive> CREATE TABLE temperature_data(
    > ds STRING,
    > zip STRING,
    > temperature DECIMAL)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.111 seconds
```

#Copy data from temp_temp into table temperature_data while doing the conversion of the date format. We convert the date in DD-MM-YYYY format to unixtime first and then we convert back unixtime to MM-DD-YYYY format

```
hive>
    > FROM temp_temp INSERT OVERWRITE TABLE temperature_data select
FROM_UNIXTIME(UNIX_TIMESTAMP(ds,'dd-mm-yyyy'),'mm-dd-yyyy'),zip, temperature;

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_20180724120158_a2db0c63-b4ab-4a8b-b5a4-d9e8e9f13f11
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1532413643255_0001, Tracking URL =
http://localhost:8088/proxy/application_1532413643255_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1532413643255_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2018-07-24 12:02:13,515 Stage-1 map = 0%,  reduce = 0%
2018-07-24 12:02:21,199 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.34 sec
MapReduce Total cumulative CPU time: 3 seconds 340 msec
Ended Job = job_1532413643255_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory
```

```
hdfs://localhost:8020/user/hive/warehouse/custom.db/temperature_data/.hive-staging_hiv
e_2018-07-24_12-01-58_261_6116726392548351416-1/-ext-10000
Loading data to table custom.temperature_data
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 3.34 sec   HDFS Read: 5041 HDFS Write: 499 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 340 msec
OK
Time taken: 24.732 seconds
```

#dump the data set file first and see date is in DD-MM-YYYY format

```
[acadgild.mmisra ~]$ cat dataset_Session\ 14.txt
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
```

#dump the table temperature_data and see that the data is loaded and date format is in MM-DD-YYYY

```
hive> select * from temperature_data;
OK
01-10-1990      123112  10
02-14-1991      283901  11
03-10-1990      381920  15
01-10-1991      302918  22
02-12-1990      384902  9
01-10-1991      123112  11
02-14-1990      283901  12
03-10-1991      381920  16
01-10-1990      302918  23
02-12-1991      384902  10
01-10-1993      123112  11
02-14-1994      283901  12
03-10-1993      381920  16
01-10-1994      302918  23
02-12-1991      384902  10
01-10-1991      123112  11
02-14-1990      283901  12
03-10-1991      381920  16
01-10-1990      302918  23
02-12-1991      384902  10
Time taken: 2.458 seconds, Fetched: 20 row(s)
hive>
```

Task 2

- Fetch date and temperature from temperature_data where zip code is greater than 300000 and less than 399999.
- Calculate maximum temperature corresponding to every year from temperature_data table.
- Calculate maximum temperature from temperature_data table corresponding to those years which have at least 2 entries in the table.
- Create a view on the top of last query, name it temperature_data_vw.
- Export contents from temperature_data_vw to a file in local file system, such that each file is '|' delimited.

SOLUTION 2

# to fetch date and temperature where zip code is greater than 300000 and less than 399999 is we use the WHERE clause

```
hive>
    >
    >
    >
    > select ds,temperature from temperature_data where zip>300000 AND zip<399999;
OK
03-10-1990      15
01-10-1991      22
02-12-1990      9
03-10-1991      16
01-10-1990      23
02-12-1991      10
03-10-1993      16
01-10-1994      23
02-12-1991      10
03-10-1991      16
01-10-1990      23
02-12-1991      10
Time taken: 0.332 seconds, Fetched: 12 row(s)
hive>
    >
    >
```

# to find the maximum temperature for each year, we create a view based on the year and temperature. We define a new column year and populate it by manipulating the date field using ‘year’ function.

```
    >
    > create VIEW temp_year AS select
year(FROM_UNIXTIME(UNIX_TIMESTAMP(ds,'mm-dd-yyyy'),'yyyy-mm-dd')) as year,temperature
FROM temperature_data;
OK
Time taken: 0.236 seconds
```

#dump the content of the temp_year to see data organized by year

```
hive>
    >
    >
    >
    > select * from temp_year;
OK
1990    10
1991    11
1990    15
1991    22
1990    9
1991    11
1990    12
1991    16
1990    23
1991    10
1993    11
1994    12
1993    16
1994    23
1991    10
1991    11
1990    12
1991    16
1990    23
1991    10
Time taken: 2.979 seconds, Fetched: 20 row(s)
```

# now we find the maximum temperature for each year by using GROUP BY clause for the year column

```
hive> select year,MAX(temperature) from temp_year group by year;
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_20180724120236_ccdeca22-eebd-4f31-afde-b9bacac58b09
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_1532413643255_0002, Tracking URL =
http://localhost:8088/proxy/application_1532413643255_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1532413643255_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-07-24 12:02:45,119 Stage-1 map = 0%,  reduce = 0%
2018-07-24 12:02:51,617 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.1 sec
2018-07-24 12:02:58,066 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.53 sec
MapReduce Total cumulative CPU time: 5 seconds 530 msec
Ended Job = job_1532413643255_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.53 sec   HDFS Read: 9896 HDFS Write:
167 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 530 msec
OK
```

```
1990    23
1991    22
1993    16
1994    23
Time taken: 22.163 seconds, Fetched: 4 row(s)
hive>
```

# to calculate Calculate maximum temperature from temperature_data table corresponding to those years which have at least 2 entries in the table, we create a view called temperature_data_vw and filter the results using the HAVING CLAUSE

```
    >
    > create view temperature_data_vw AS select year,MAX(temperature) from temp_year group
by year HAVING year>1;
OK
Time taken: 0.382 seconds
hive>
```

# now we dump the results for the above query

```
    > select * from temperature_data_vw;
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_20180724120400_2776423b-cf91-4554-9e15-c32066838fc6
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_1532413643255_0003, Tracking URL =
http://localhost:8088/proxy/application_1532413643255_0003/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1532413643255_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-07-24 12:04:08,769 Stage-1 map = 0%,  reduce = 0%
2018-07-24 12:04:15,261 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.85 sec
2018-07-24 12:04:21,614 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.27 sec
MapReduce Total cumulative CPU time: 5 seconds 270 msec
Ended Job = job_1532413643255_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.27 sec   HDFS Read: 10773 HDFS Write:
167 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 270 msec
OK
1990    23
1991    22
1993    16
1994    23
Time taken: 23.53 seconds, Fetched: 4 row(s)
```

# To write the table temperature_data_vw content into a file we use the INSERT OVERWRITE command as below

```
 hive>
    > insert overwrite local directory 'ans' row format 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 future versions.
Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180724120431_cbc37894-91fb-44ce-948c-6e0fb1f0ee24
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_1532413643255_0004, Tracking URL =
http://localhost:8088/proxy/application_1532413643255_0004/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
job_1532413643255_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-07-24 12:04:39,018 Stage-1 map = 0%,  reduce = 0%
2018-07-24 12:04:45,604 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.31 sec
2018-07-24 12:04:53,075 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.83 sec
MapReduce Total cumulative CPU time: 5 seconds 830 msec
Ended Job = job_1532413643255_0004
Moving data to local directory ans
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.83 sec   HDFS Read: 10355 HDFS Write:
32 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 830 msec
OK
Time taken: 22.279 seconds
hive>
    >
    >
    >
    > quit;
You have new mail in /var/spool/mail/acadgild
```

#Now we check the contents of the output file to see in the required format

```
[acadgild.mmisra ~]$ cd ans
[acadgild.mmisra ans]$ ls -la
total 16
drwxrwxr-x.  2 acadgild acadgild 4096 Jul 24 12:04 .
drwx------. 45 acadgild acadgild 4096 Jul 24 12:04 ..
-rw-r--r--.  1 acadgild acadgild   32 Jul 24 12:04 000000_0
-rw-r--r--.  1 acadgild acadgild   12 Jul 24 12:04 .000000_0.crc
[acadgild.mmisra ans]$ cat 000000_0
1990|23
1991|22
1993|16
1994|23
[acadgild.mmisra ans]$
```