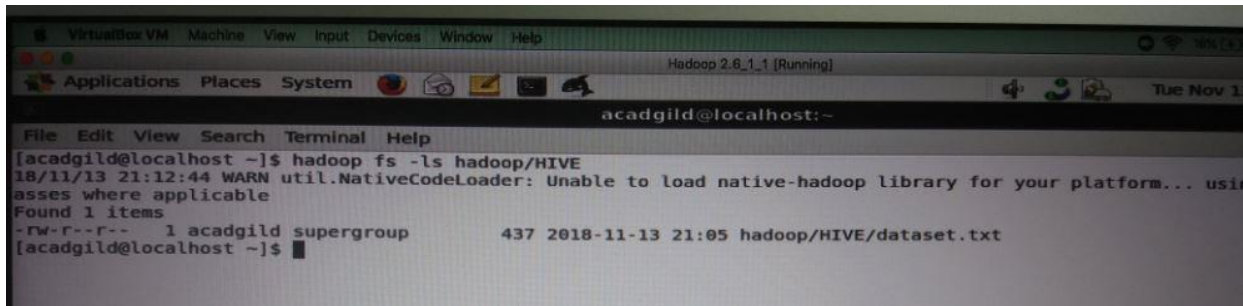


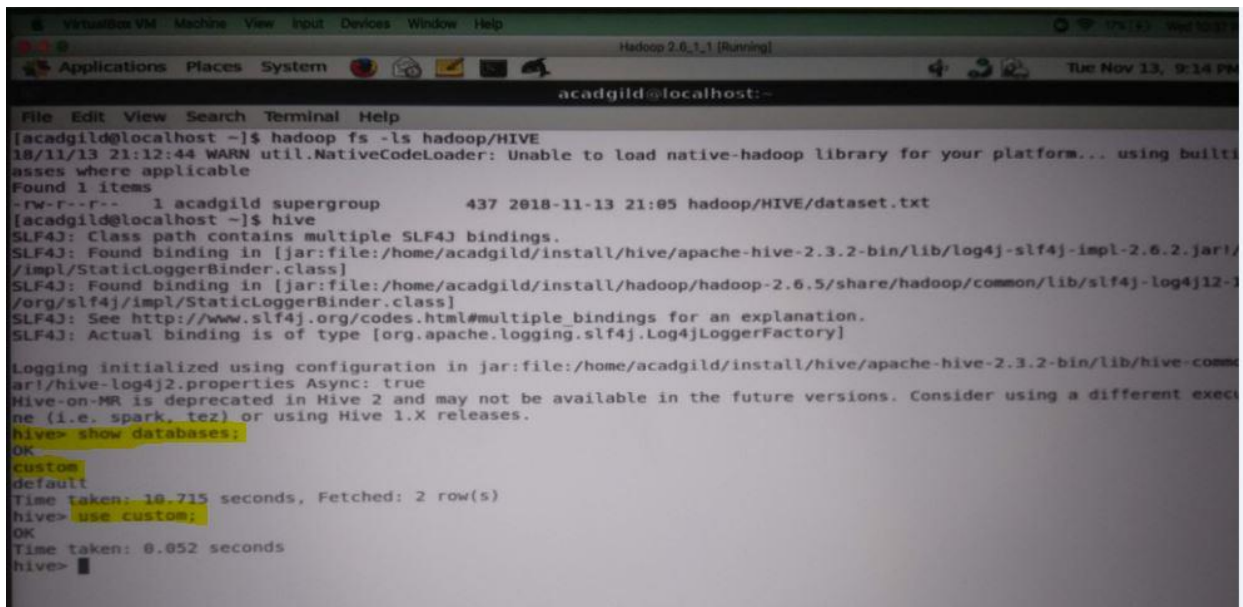
Assignment 8.1

1. Copied the dataset file to hadoop on path /hadoop/HIVE/dataset.txt.



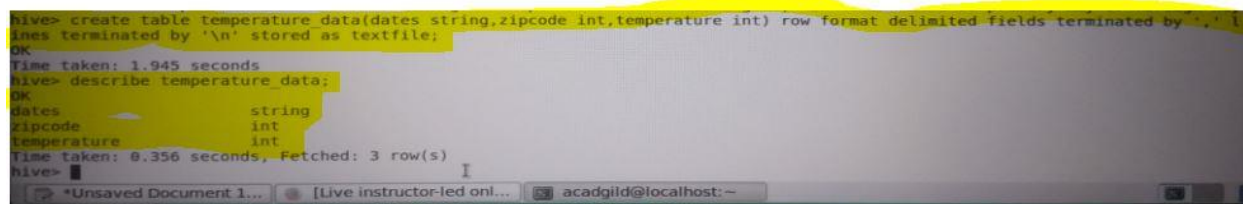
```
acadmild@localhost:~  
[acadmild@localhost ~]$ hadoop fs -ls hadoop/HIVE  
18/11/13 21:12:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using built-in  
classes where applicable  
Found 1 items  
-rw-r--r-- 1 acadmild supergroup          437 2018-11-13 21:05 hadoop/HIVE/dataset.txt  
[acadmild@localhost ~]$
```

2. Started hive on command prompt by using command hive.
3. Created data custom using command 'create database custom'.
4. Verified that the database has been created successfully by using command show databases.



```
acadmild@localhost:~  
[acadmild@localhost ~]$ hadoop fs -ls hadoop/HIVE  
18/11/13 21:12:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using built-in  
classes where applicable  
Found 1 items  
-rw-r--r-- 1 acadmild supergroup          437 2018-11-13 21:05 hadoop/HIVE/dataset.txt  
[acadmild@localhost ~]$ hive  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/acadmild/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/acadmild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12.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/acadmild/install/hive/apache-hive-2.3.2-bin/lib/hive-common.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> show databases;  
OK  
custom  
default  
Time taken: 10.715 seconds, Fetched: 2 row(s)  
hive> use custom;  
OK  
Time taken: 0.052 seconds  
hive>
```

5. Created table temperature_data in custom database using the command 'create table if not exists temperature_data (dates string, zip_code int, temperature int) row format delimited fields terminated by ',' lines terminated by '\n' stored as textfile ;'



```
hive> create table if not exists temperature_data (dates string, zip_code int, temperature int) row format delimited fields terminated by ',' lines terminated by '\n' stored as textfile ;  
OK  
Time taken: 1.945 seconds  
hive> describe temperature_data;  
OK  
dates          string  
zip_code       int  
temperature    int  
Time taken: 0.356 seconds, Fetched: 3 row(s)  
hive>
```

6. Inserted data into the table temperature_data from hadoop using command load data inpath '/hadoop/HIVE/dataset' into table temperature_data.

```
hive> load data inpath 'hadoop/HIVE/dataset.txt' into table temperature_data;
Loading data to table custom.temperature_data
OK
Time taken: 2.673 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
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: 0.555 seconds, Fetched: 20 row(s)
hive>
```

7. To fetch date and temperature from temperature_data where zip code is greater than 300000 and less than 399999 used query 'select dates, temperature from temperature_data where zipcode > 300000 and zipcode < 399999;'.

```
Time taken: 0.555 seconds, Fetched: 20 row(s)
hive> select dates,temperature from temperature_data where zipcode > 300000 and 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: 0.512 seconds, Fetched: 12 row(s)
hive>
```

8. To calculate maximum temperature corresponding to every year from temperature_data used query 'select year(from_unixtime(unix_timestamp(dates,'mm-dd-yyyy'))), max(temperature) from temperature_data group by year(from_unixtime(unix_timestamp(dates,'mm-dd-yyyy')));' but due to some issue it gets stuck and does not move forward. Everytime I run this query it gets stuck and I am not able to proceed further.

```
Time taken: 0.513 seconds, Fetched: 20 row(s)
hive> select year(from_unixtime(unix_timestamp(dates,'mm-dd-yyyy'))),max(temperature) from temperature_data group by year(fro
m_unixtime(unix_timestamp(dates,'mm-dd-yyyy')));
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execu
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20181113220306_2a1fe97e-0d8b-45d8-b95a-6e53e76cd26b
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_1542114883196_0004, Tracking URL = http://localhost:8088/proxy/application_1542114883196_0004/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1542114883196_0004
```

9. Below are the queries that can be used to generate the desired output. Due to the above issue I was not able to execute the below queries.
10. To calculate max temperature from temperature_data table corresponding to those years which have at least 2 entries in the table used query **'select year(from_unixtime(unix_timestamp(dates,'mm-dd-yyyy'))), max(temperature) from temperature_data group by year(from_unixtime(unix_timestamp(dates,'mm-dd-yyyy')) having count(*) > 1;'**
11. To create view on the top of last query with name temperature_data_vw used query **'create view temperature_data_vw as select year(from_unixtime(unix_timestamp(dates,'mm-dd-yyyy'))), max(temperature) from temperature_data group by year(from_unixtime(unix_timestamp(dates,'mm-dd-yyyy')) having count(*) > 1;'**
12. To export the contents from temperature_data_vw to a file in local file system, such that each file is '|' delimited used the query **'insert overwrite local directory '/home/acadgild/HIVE/output' row format delimited fields terminated by '|' select * from temperature_data_vw;'**