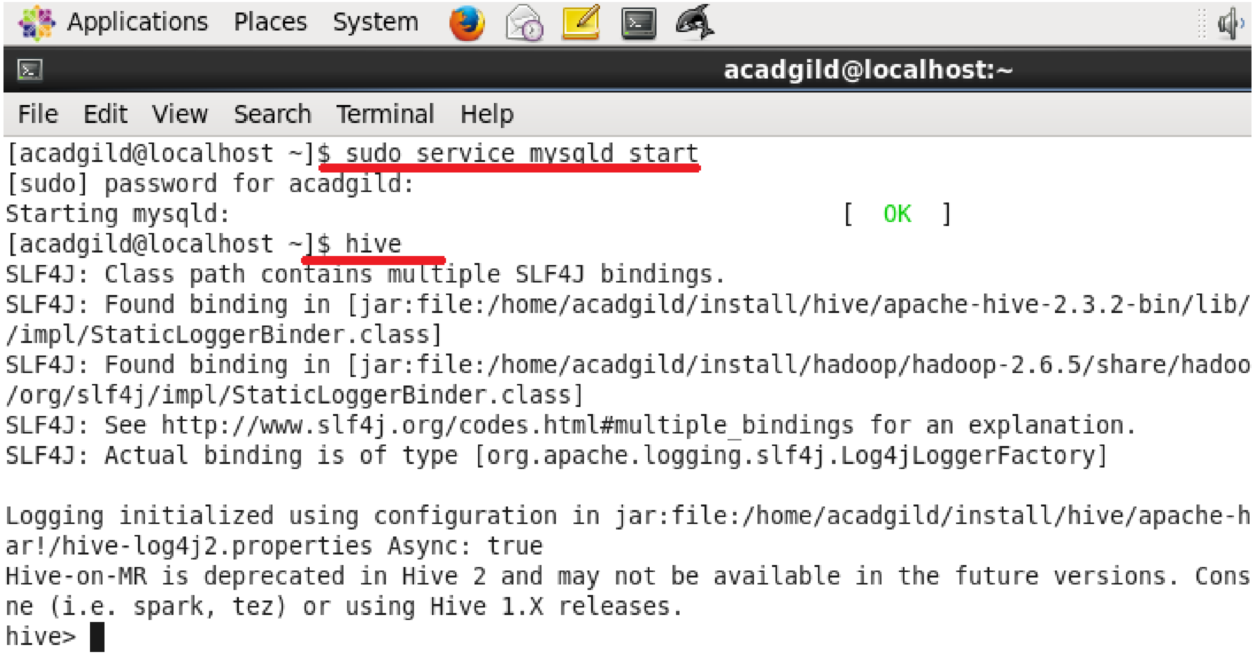
Assignment 8

Strat HIVE



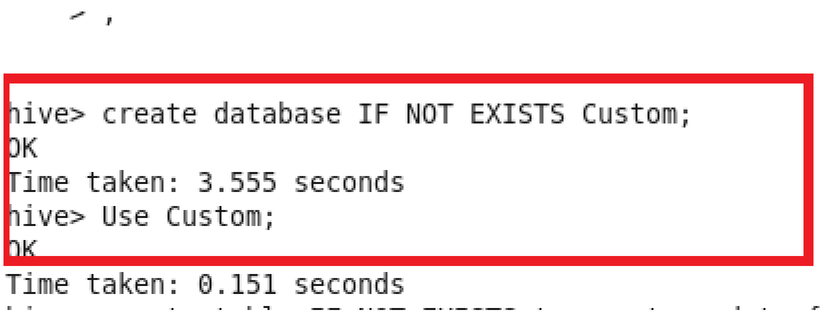
```
[acadgild@localhost ~]$ sudo service mysqld start
[sudo] password for acadgild:
Starting mysqld:                                           [  OK  ]
[acadgild@localhost ~]$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/
/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoo
/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-h
ar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Cons
ne (i.e. spark, tez) or using Hive 1.X releases.
hive>
```

Task 1

Create a database named 'custom'.

Command:
Create database IF NOT EXISTS Custom;



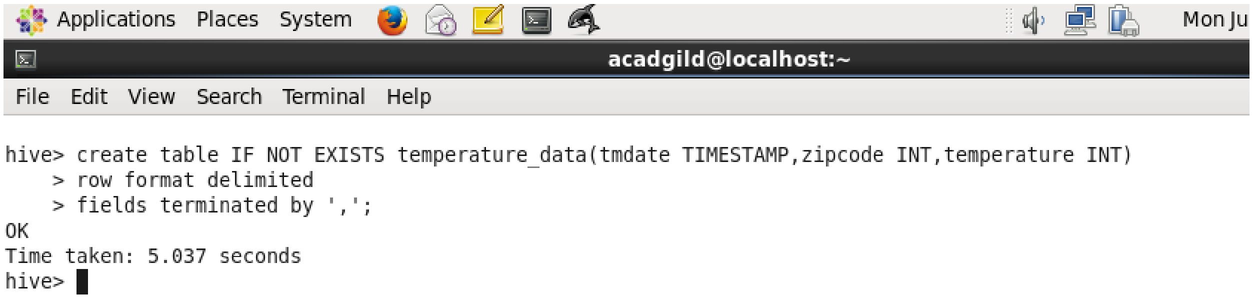
```
hive> create database IF NOT EXISTS Custom;
OK
Time taken: 3.555 seconds
hive> Use Custom;
OK
Time taken: 0.151 seconds
```

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.

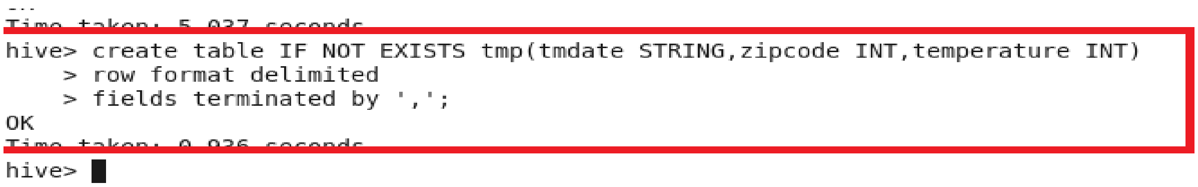
Command:

Create table IF NOT EXISTS temperature_data(tmdate TIMESTAMP,zipcode INT,temperature INT) row format delimited fields terminated by ‘,’;



```
hive> create table IF NOT EXISTS temperature_data(tmdate TIMESTAMP,zipcode INT,temperature INT)
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 5.037 seconds
hive> 
```

Create table IF NOT EXISTS tmp(tmdate STRING,zipcode INT,temperature INT) row format delimited fields terminated by ‘,’;



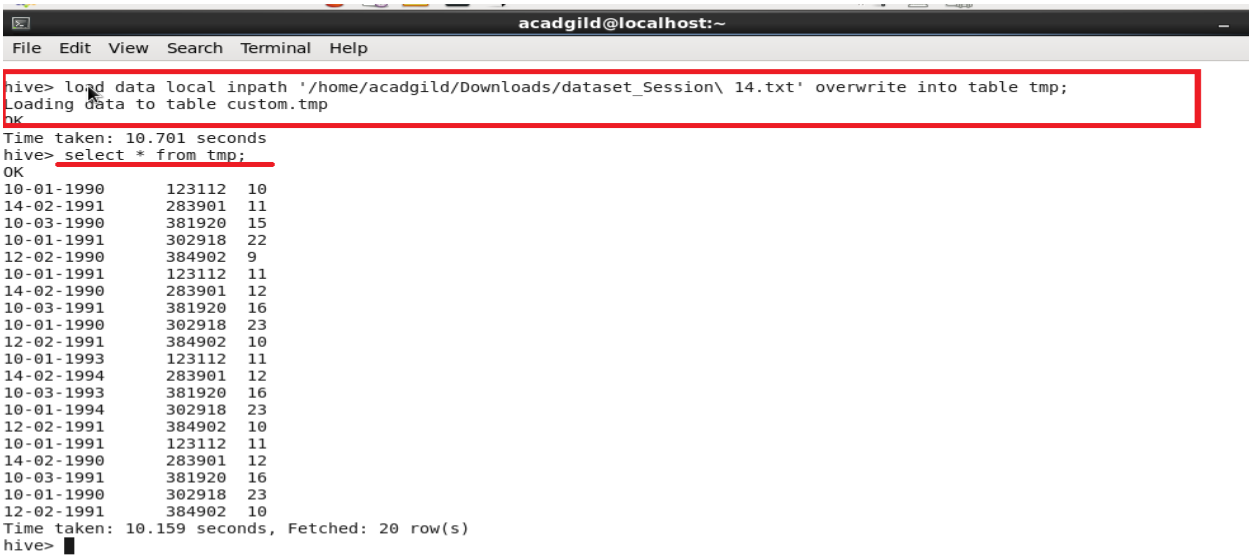
```
Time taken: 5.037 seconds
hive> create table IF NOT EXISTS tmp(tmdate STRING,zipcode INT,temperature INT)
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.036 seconds
hive> 
```

NOTE:
Tmp table is created in order to load the data from the file keeping date as STRING. The reason behind this is that, hive supports only ‘YYYY-MM-DD’ format and the format present in txt file is ‘DD-MM-YYYY’. First the data will be loaded to tmp table and then the data from tmp table will be inserted to temperature_data table.

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

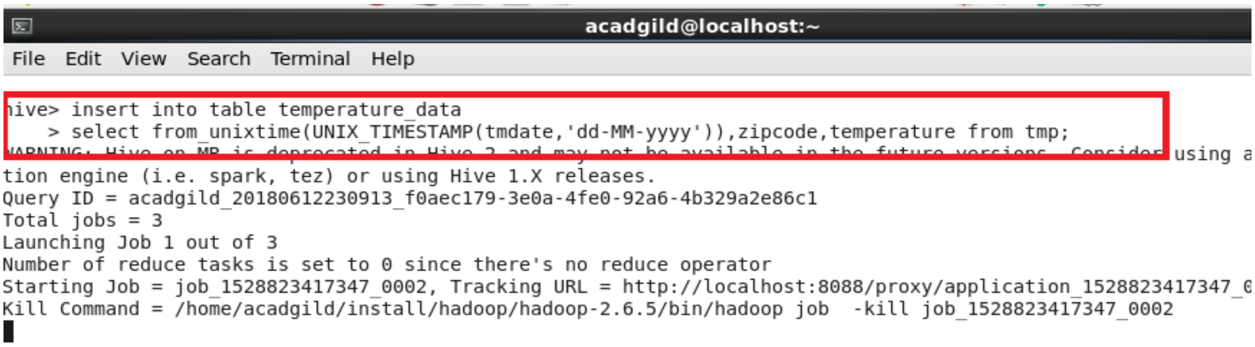
Load data local inpath ‘/home/acadgild/Downloads/dataset_session\ 14.txt’ overwrite into table tmp;

Select * from tmp;



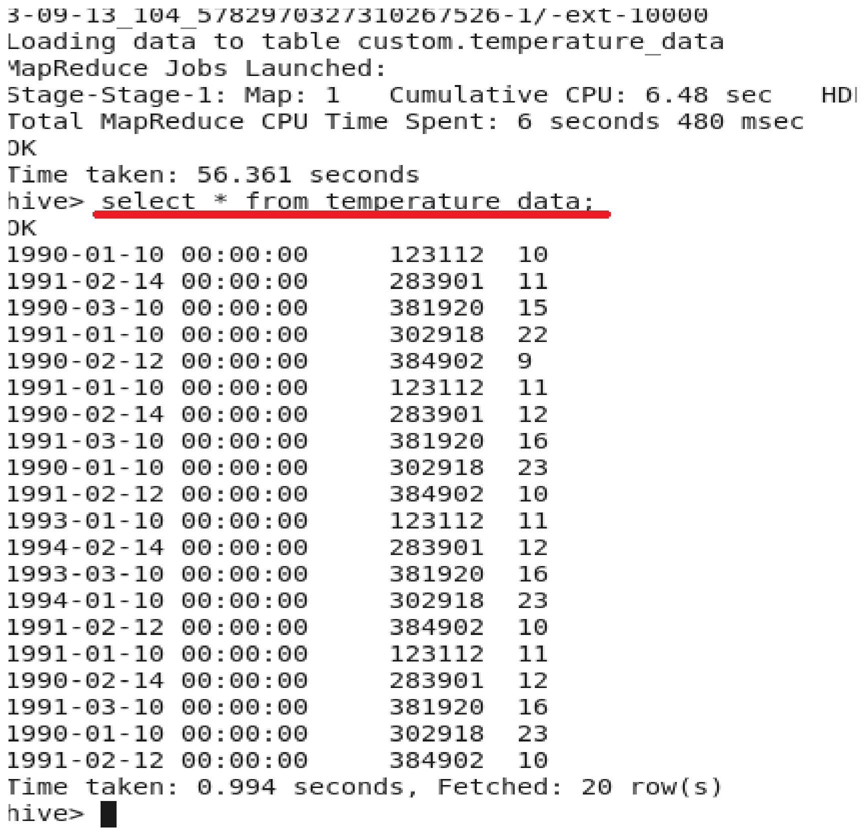
```
hive> load data local inpath '/home/acadgild/Downloads/dataset_Session\ 14.txt' overwrite into table tmp;
Loading data to table custom.tmp
OK
Time taken: 10.701 seconds
hive> select * from tmp;
OK
10-01-1990      123112  10
14-02-1991      283901  11
10-03-1990      381920  15
10-01-1991      302918  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: 10.159 seconds, Fetched: 20 row(s)
hive>
```

Insert into temperature_data select from unixtime(UNIX_TIMESTAMP(tmdate,’dd-MM-yyyy’)),zipcode,temperature from tmp;



```
hive> insert into table temperature_data
    > select from_unixtime(UNIX_TIMESTAMP(tmdate,'dd-MM-yyyy')),zipcode,temperature from tmp;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180612230913_f0aec179-3e0a-4fe0-92a6-4b329a2e86c1
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_1528823417347_0002, Tracking URL = http://localhost:8088/proxy/application_1528823417347_0
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1528823417347_0002
```

Select * from temperature_data;

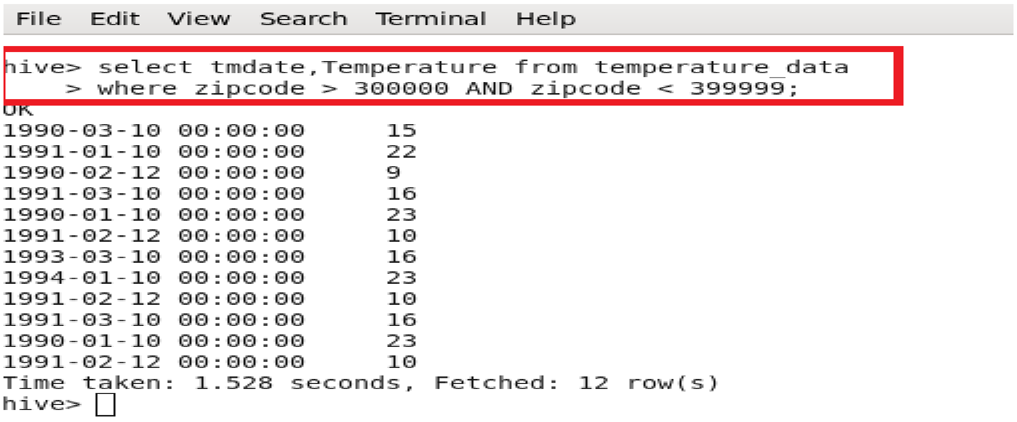


```
3-09-13_104_5782970327310267526-1/-ext-10000
Loading data to table custom.temperature_data
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 6.48 sec   HDI
Total MapReduce CPU Time Spent: 6 seconds 480 msec
OK
Time taken: 56.361 seconds
hive> select * from temperature_data;
OK
1990-01-10 00:00:00     123112  10
1991-02-14 00:00:00     283901  11
1990-03-10 00:00:00     381920  15
1991-01-10 00:00:00     302918  22
1990-02-12 00:00:00     384902  9
1991-01-10 00:00:00     123112  11
1990-02-14 00:00:00     283901  12
1991-03-10 00:00:00     381920  16
1990-01-10 00:00:00     302918  23
1991-02-12 00:00:00     384902  10
1993-01-10 00:00:00     123112  11
1994-02-14 00:00:00     283901  12
1993-03-10 00:00:00     381920  16
1994-01-10 00:00:00     302918  23
1991-02-12 00:00:00     384902  10
1991-01-10 00:00:00     123112  11
1990-02-14 00:00:00     283901  12
1991-03-10 00:00:00     381920  16
1990-01-10 00:00:00     302918  23
1991-02-12 00:00:00     384902  10
Time taken: 0.994 seconds, Fetched: 20 row(s)
hive>
```

Task 2

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

Select tmdate,Temperature from temperature_data where zipcode > 300000 AND zipcode <399999;

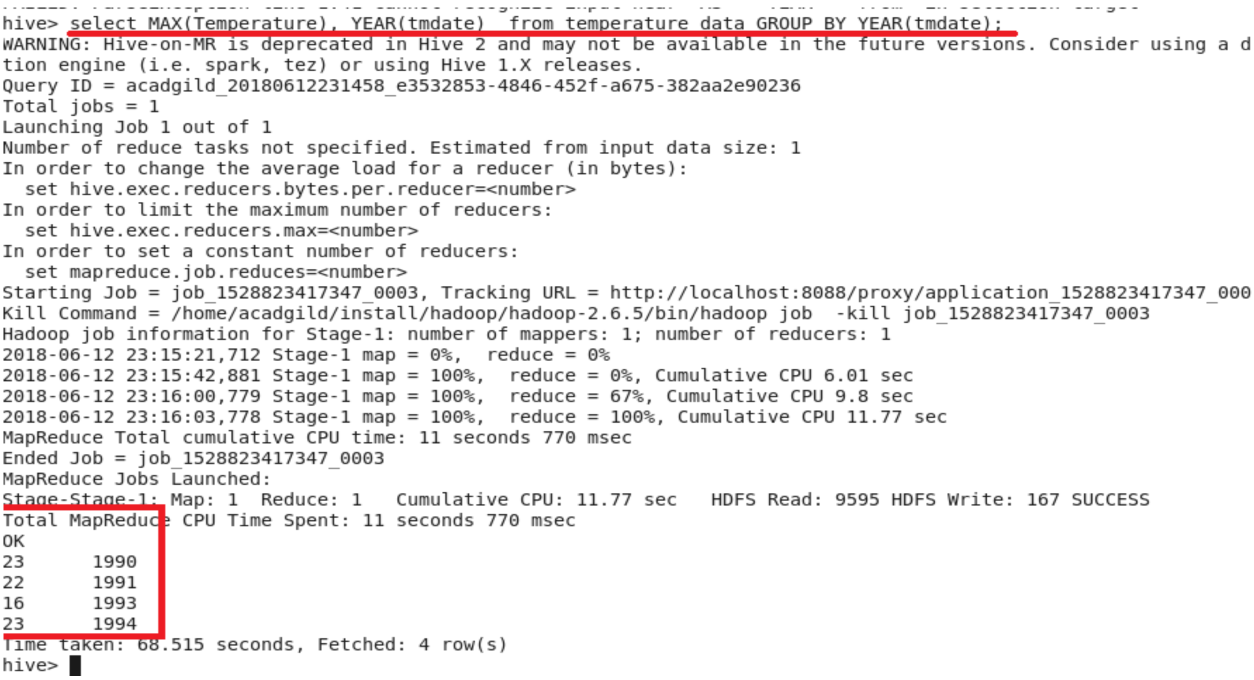


```
hive> select tmdate,Temperature from temperature_data
    > where zipcode > 300000 AND zipcode < 399999;
OK
1990-03-10 00:00:00     15
1991-01-10 00:00:00     22
1990-02-12 00:00:00     9
1991-03-10 00:00:00     16
1990-01-10 00:00:00     23
1991-02-12 00:00:00     10
1993-03-10 00:00:00     16
1994-01-10 00:00:00     23
1991-02-12 00:00:00     10
1991-03-10 00:00:00     16
1990-01-10 00:00:00     23
1991-02-12 00:00:00     10
Time taken: 1.528 seconds, Fetched: 12 row(s)
hive>
```

2. Calculate maximum temperature corresponding to every year from temperature_data

table.

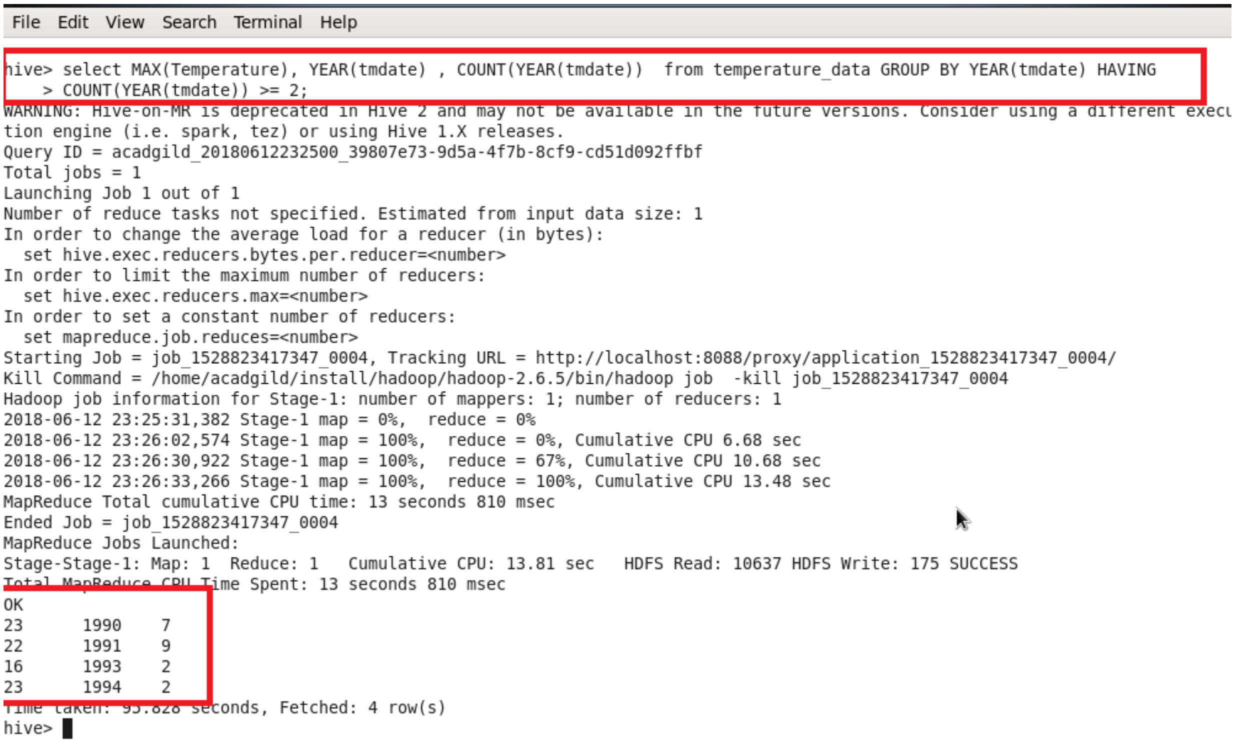
Select MAX(Temperature), YEAR(tmdate) from temperature_data GROUP BY YEAR(tmdate);



```
hive> select MAX(Temperature), YEAR(tmdate)  from temperature_data GROUP BY YEAR(tmdate);
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a d
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180612231458_e3532853-4846-452f-a675-382aa2e90236
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_1528823417347_0003, Tracking URL = http://localhost:8088/proxy/application_1528823417347_000
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1528823417347_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-06-12 23:15:21,712 Stage-1 map = 0%,  reduce = 0%
2018-06-12 23:15:42,881 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.01 sec
2018-06-12 23:16:00,779 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 9.8 sec
2018-06-12 23:16:03,778 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 11.77 sec
MapReduce Total cumulative CPU time: 11 seconds 770 msec
Ended Job = job_1528823417347_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 11.77 sec   HDFS Read: 9595 HDFS Write: 167 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 770 msec
OK
23      1990
22      1991
16      1993
23      1994
Time taken: 68.515 seconds, Fetched: 4 row(s)
hive>
```

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

Select MAX(Temperature), YEAR(tmdate) ,COUNT(YEAR(tmdate)) from temperature_data GROUP BY YEAR(tmdate) HAVIND COUNT(YEAR(tmdate)) >=2;

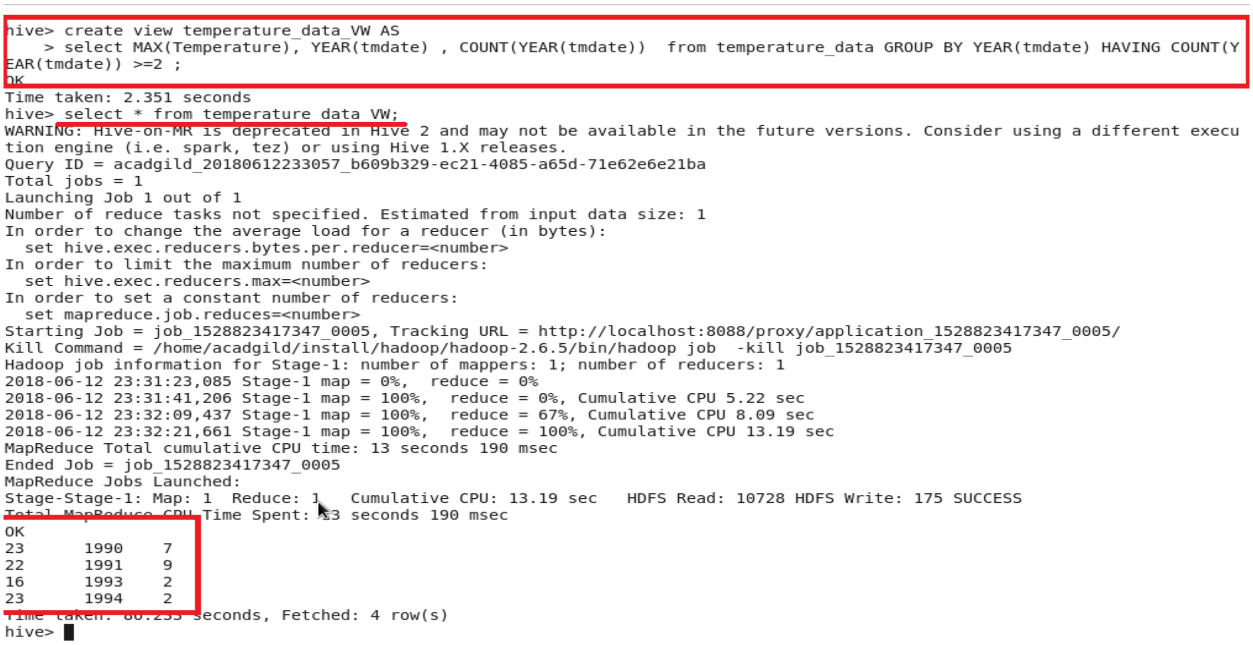


```
hive> select MAX(Temperature), YEAR(tmdate) , COUNT(YEAR(tmdate))  from temperature_data GROUP BY YEAR(tmdate) HAVING
    > COUNT(YEAR(tmdate)) >= 2;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execu
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180612232500_39807e73-9d5a-4f7b-8cf9-cd51d092ffbf
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_1528823417347_0004, Tracking URL = http://localhost:8088/proxy/application_1528823417347_0004/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1528823417347_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-06-12 23:25:31,382 Stage-1 map = 0%,  reduce = 0%
2018-06-12 23:26:02,574 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.68 sec
2018-06-12 23:26:30,922 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 10.68 sec
2018-06-12 23:26:33,266 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 13.48 sec
MapReduce Total cumulative CPU time: 13 seconds 810 msec
Ended Job = job_1528823417347_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 13.81 sec   HDFS Read: 10637 HDFS Write: 175 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 810 msec
OK
23      1990    7
22      1991    9
16      1993    2
23      1994    2
Time taken: 95.828 seconds, Fetched: 4 row(s)
hive> 
```

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

Create view temperature_data_VW AS Select MAX(Temperature), YEAR(tmdate) ,COUNT(YEAR(tmdate)) from temperature_data GROUP BY YEAR(tmdate) HAVIND COUNT(YEAR(tmdate)) >=2;

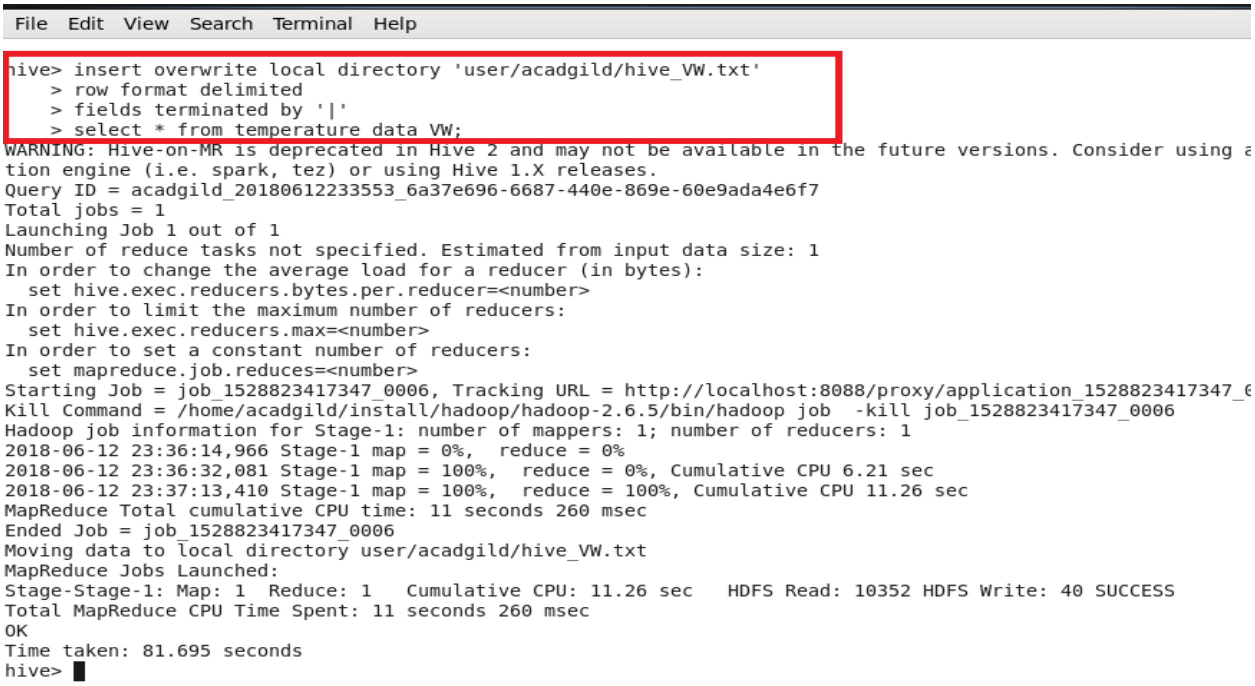
Select * from temperature_data_VW;



```
hive> create view temperature_data_VW AS
    > select MAX(Temperature), YEAR(tmdate) , COUNT(YEAR(tmdate))  from temperature_data GROUP BY YEAR(tmdate) HAVING COUNT(Y
EAR(tmdate)) >=2 ;
OK
Time taken: 2.351 seconds
hive> 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 execu
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180612233057_b609b329-ec21-4085-a65d-71e62e6e21ba
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_1528823417347_0005, Tracking URL = http://localhost:8088/proxy/application_1528823417347_0005/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1528823417347_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-06-12 23:31:23,085 Stage-1 map = 0%,  reduce = 0%
2018-06-12 23:31:41,206 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.22 sec
2018-06-12 23:32:09,437 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 8.09 sec
2018-06-12 23:32:21,661 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 13.19 sec
MapReduce Total cumulative CPU time: 13 seconds 190 msec
Ended Job = job_1528823417347_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 13.19 sec   HDFS Read: 10728 HDFS Write: 175 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 190 msec
OK
23      1990    7
22      1991    9
16      1993    2
23      1994    2
Time taken: 80.233 seconds, Fetched: 4 row(s)
hive> █
```

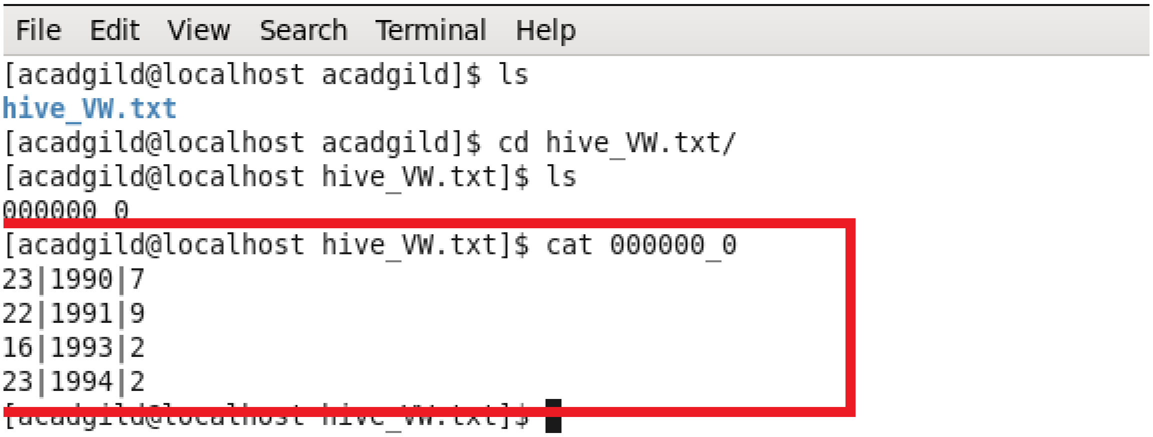
5. Export contents from temperature_data_vw to a file in local file system, such that each file is '|' delimited.

Insert overwrite local directory ‘user/acadgild/hive_VW.txt’ row format delimited fields terminated by ‘|’ select * from temperature_data_VW;



```
hive> insert overwrite local directory 'user/acadgild/hive_VW.txt'
    > 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
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180612233553_6a37e696-6687-440e-869e-60e9ada4e6f7
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_1528823417347_0006, Tracking URL = http://localhost:8088/proxy/application_1528823417347_0
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1528823417347_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-06-12 23:36:14,966 Stage-1 map = 0%,  reduce = 0%
2018-06-12 23:36:32,081 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.21 sec
2018-06-12 23:37:13,410 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 11.26 sec
MapReduce Total cumulative CPU time: 11 seconds 260 msec
Ended Job = job_1528823417347_0006
Moving data to local directory user/acadgild/hive_VW.txt
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 11.26 sec   HDFS Read: 10352 HDFS Write: 40 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 260 msec
OK
Time taken: 81.695 seconds
hive> █
```

Display the contents of output file 000000_0



```
[acadgild@localhost acadgild]$ ls
hive_VW.txt
[acadgild@localhost acadgild]$ cd hive_VW.txt/
[acadgild@localhost hive_VW.txt]$ ls
000000_0
[acadgild@localhost hive_VW.txt]$ cat 000000_0
23|1990|7
22|1991|9
16|1993|2
23|1994|2
[acadgild@localhost hive_VW.txt]$
```