

Creating and using a database:

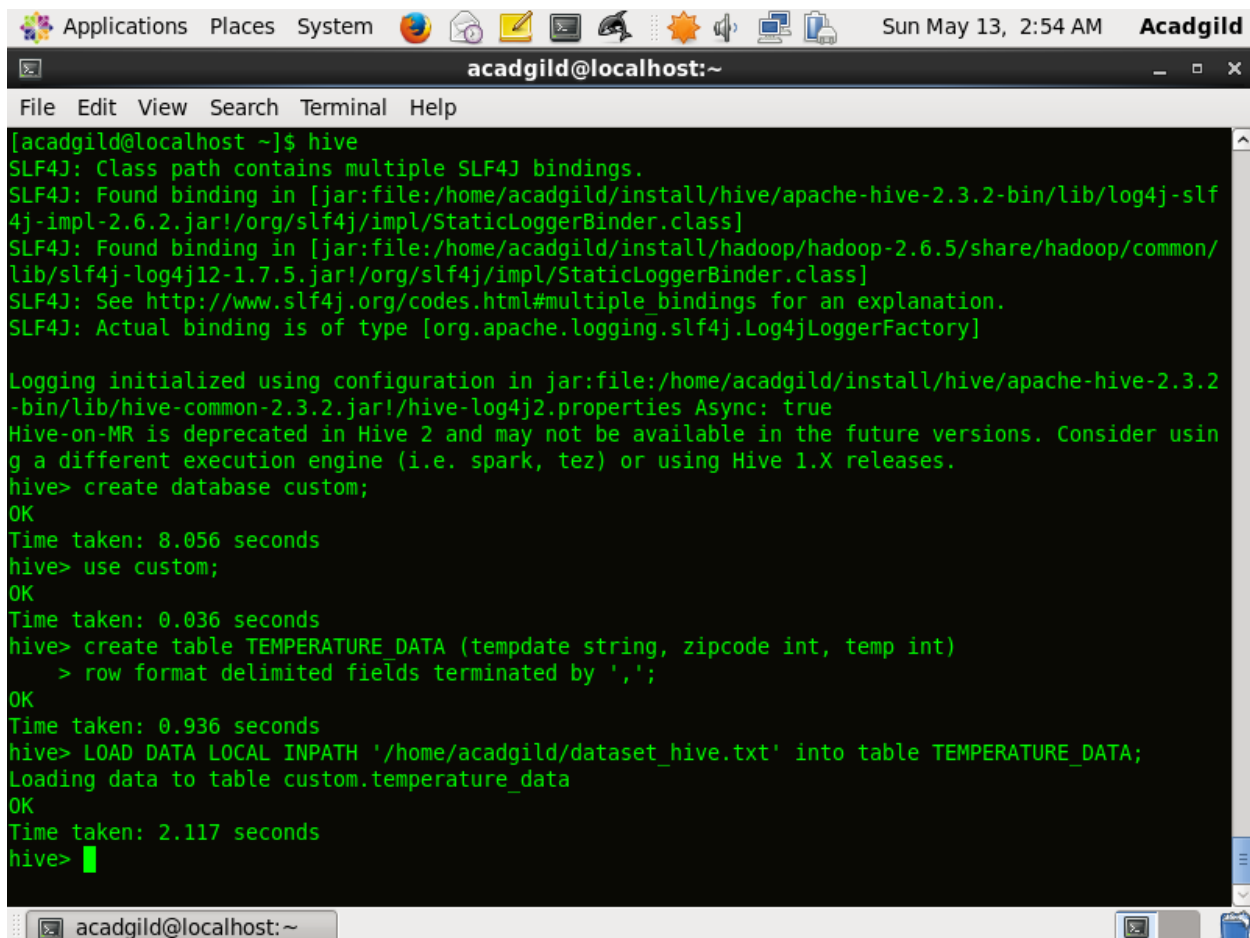
Create database 'custom';

Use 'custom';

Creating the specified table:

Create table temperature_data (tempdate string, zipcode int, temp int) row
format delimited filelds terminated by ',';

Use command LOAD DATA LOCAL INPATH '<path>' into <table-name>;

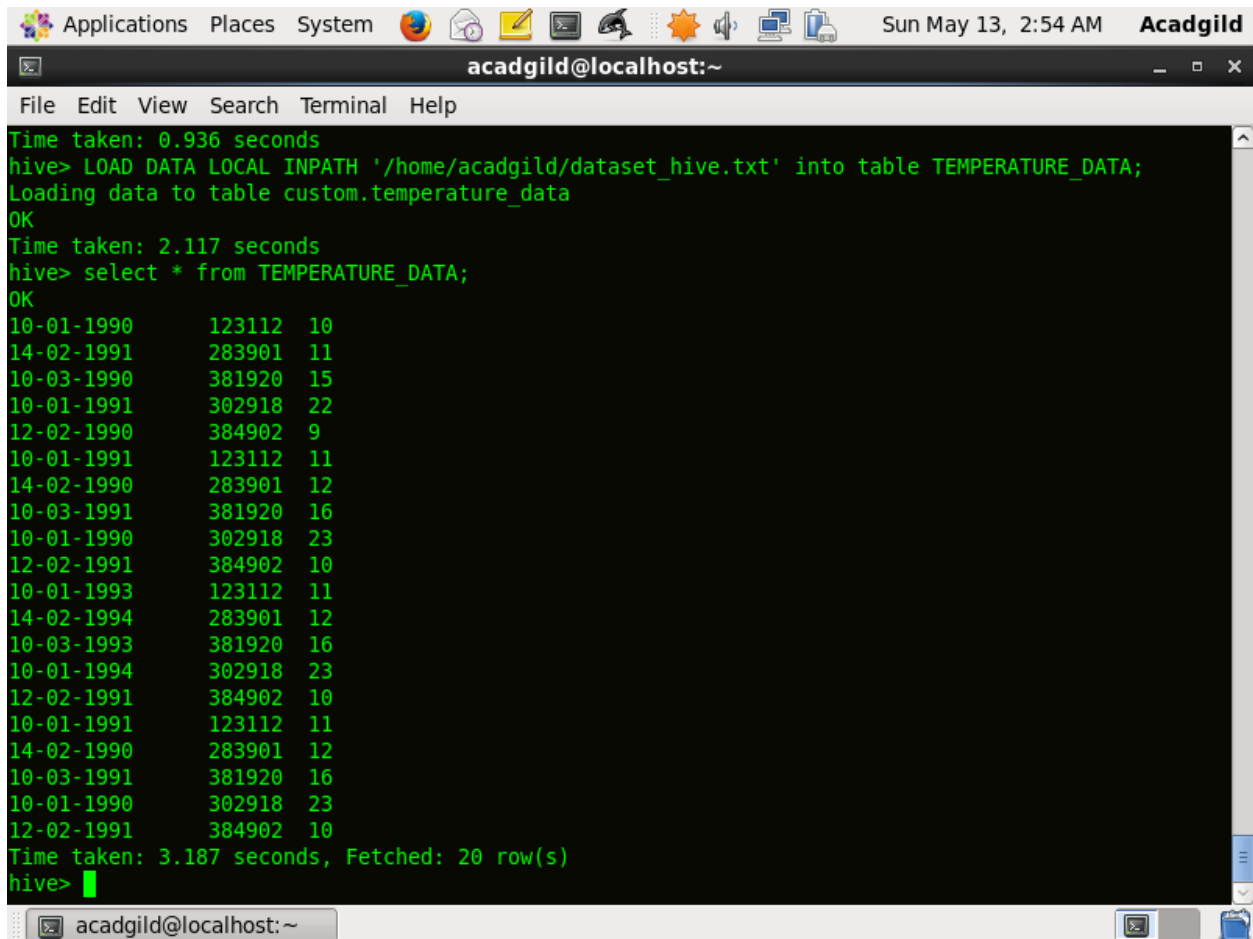
A screenshot of a terminal window titled 'acadgild@localhost:~'. The window shows the execution of Hive commands. The output includes SLF4J warnings about multiple bindings, logging initialization messages, and successful execution of 'create database custom;', 'use custom;', 'create table TEMPERATURE_DATA (tempdate string, zipcode int, temp int) row format delimited fields terminated by ',';', and 'LOAD DATA LOCAL INPATH '/home/acadgild/dataset_hive.txt' into table TEMPERATURE_DATA;'. The terminal shows time taken for each command and ends with a prompt 'hive>' and a cursor.

```
[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/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-log4j12-1.7.5.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> create database custom;
OK
Time taken: 8.056 seconds
hive> use custom;
OK
Time taken: 0.036 seconds
hive> create table TEMPERATURE_DATA (tempdate string, zipcode int, temp int)
> row format delimited fields terminated by ',';
OK
Time taken: 0.936 seconds
hive> LOAD DATA LOCAL INPATH '/home/acadgild/dataset_hive.txt' into table TEMPERATURE_DATA;
Loading data to table custom.temperature_data
OK
Time taken: 2.117 seconds
hive> █
```

Check contents of the table after loading data by using command :

Select * from temperature_data



The screenshot shows a terminal window titled 'acadgild@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help) and a system tray at the top. The terminal displays the following text:

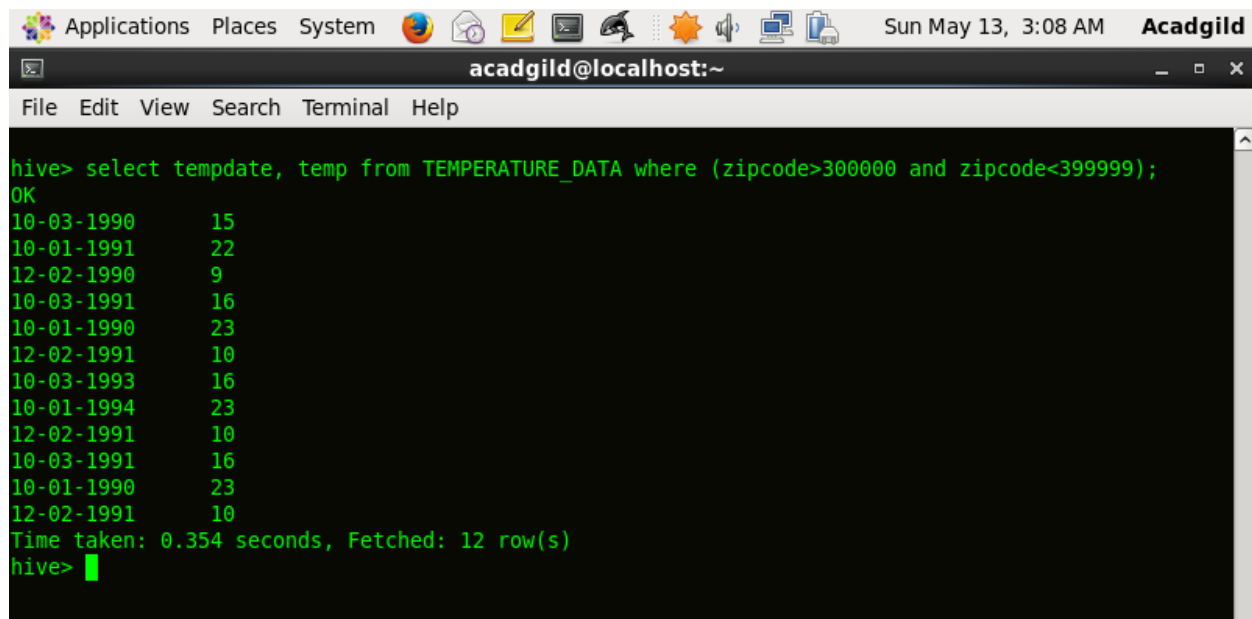
```
Time taken: 0.936 seconds
hive> LOAD DATA LOCAL INPATH '/home/acadgild/dataset_hive.txt' into table TEMPERATURE_DATA;
Loading data to table custom.temperature_data
OK
Time taken: 2.117 seconds
hive> select * from TEMPERATURE_DATA;
OK
10-01-1990      123112  10
14-02-1991      283901  11
10-03-1990      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: 3.187 seconds, Fetched: 20 row(s)
hive>
```

To Fetch date and temperature from temperature_data where zip code is greater than 300000 and less than 399999 use the where keyword to filter the records by using the condition as mentioned

```
hive> select tempdate, temp from TEMPERATURE_DATA where (zipcode>300000 and zipcode<399999);
```

I am selecting tempdate and temp because I want only those fields in the output.

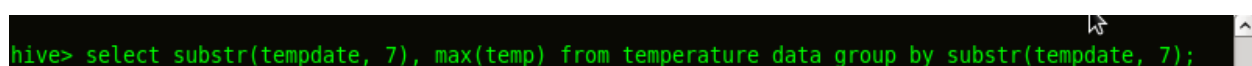
Output:



```
Applications Places System acadgild@localhost:~ Sun May 13, 3:08 AM Acadgild
File Edit View Search Terminal Help
hive> select tempdate, temp 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: 0.354 seconds, Fetched: 12 row(s)
hive>
```

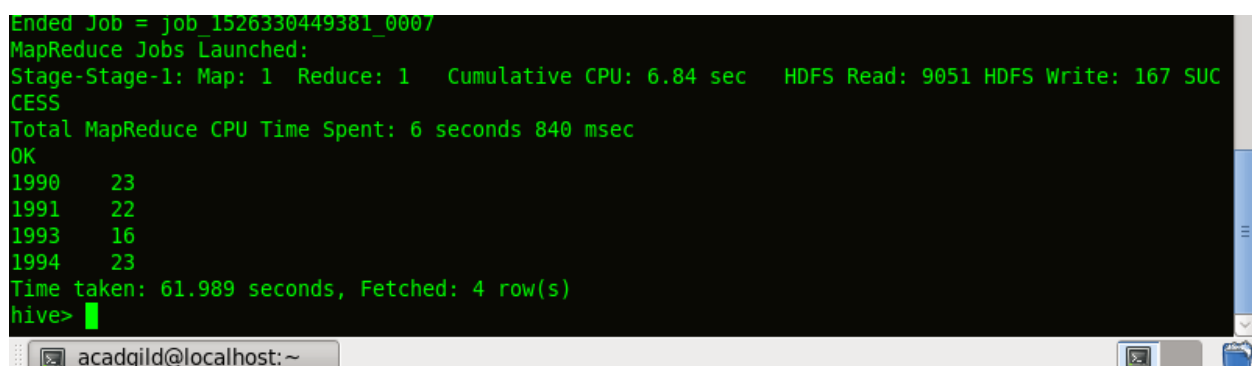
To find maximum temperature corresponding to every year from temperature_data table we have to group tempdate column and find the maximum temperature from each group.

We will use built in function substr() for returning a part of the tempdate string which will give the year and max() function to find the max value from temp column. Use the below command.



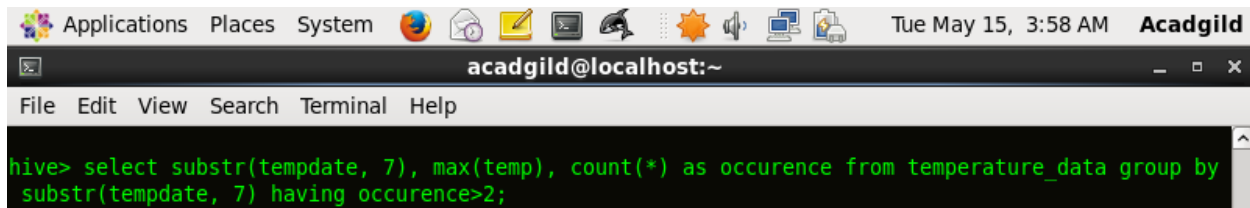
```
hive> select substr(tempdate, 7), max(temp) from temperature_data group by substr(tempdate, 7);
```

Output:



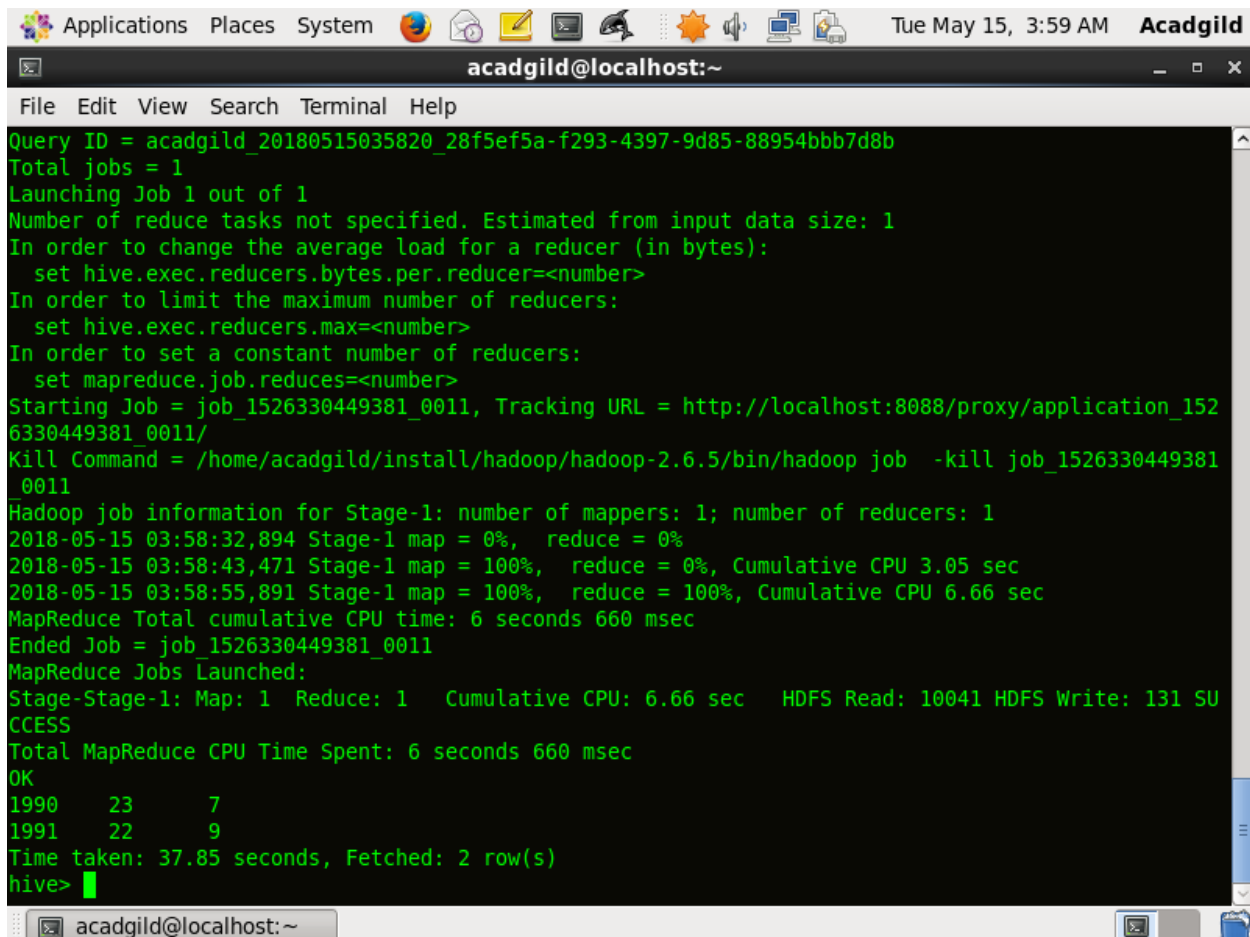
```
Ended Job = job_1526330449381_0007
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 6.84 sec HDFS Read: 9051 HDFS Write: 167 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 840 msec
OK
1990      23
1991      22
1993      16
1994      23
Time taken: 61.989 seconds, Fetched: 4 row(s)
hive>
```

To Calculate maximum temperature from temperature_data table corresponding to those years which have at least 2 entries in the table, I add the having clause which is used to filter the grouped output. Here I assign the number of entries to occurrence variable which will store the output of the coun() function and having condition will check whether the number of entries are more than 2.



```
Applications Places System acadgild@localhost:~
File Edit View Search Terminal Help
hive> select substr(tempdate, 7), max(temp), count(*) as occurrence from temperature_data group by
substr(tempdate, 7) having occurrence>2;
```

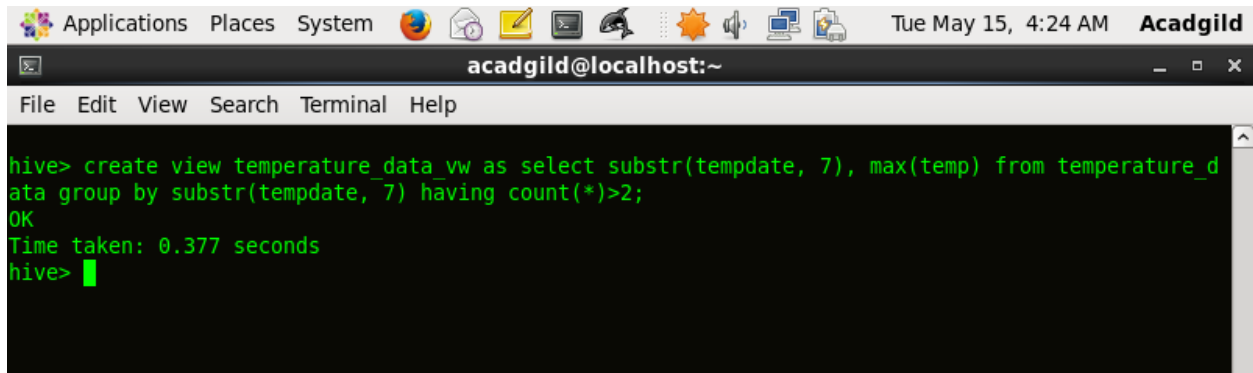
Output:



```
Applications Places System acadgild@localhost:~
File Edit View Search Terminal Help
Query ID = acadgild_20180515035820_28f5ef5a-f293-4397-9d85-88954bbb7d8b
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_1526330449381_0011, Tracking URL = http://localhost:8088/proxy/application_152
6330449381_0011/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1526330449381
_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-05-15 03:58:32,894 Stage-1 map = 0%, reduce = 0%
2018-05-15 03:58:43,471 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.05 sec
2018-05-15 03:58:55,891 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 6.66 sec
MapReduce Total cumulative CPU time: 6 seconds 660 msec
Ended Job = job_1526330449381_0011
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 6.66 sec HDFS Read: 10041 HDFS Write: 131 SU
CCESS
Total MapReduce CPU Time Spent: 6 seconds 660 msec
OK
1990      23      7
1991      22      9
Time taken: 37.85 seconds, Fetched: 2 row(s)
hive>
```

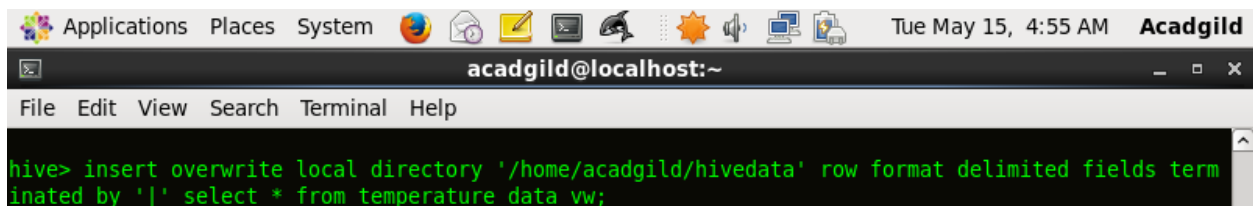
Create view <view name> as <query>

So temperature_data_vw is my view name and using my last query to store in the view.

A terminal window titled 'acadgild@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help) and a toolbar. The terminal shows the following commands and output:

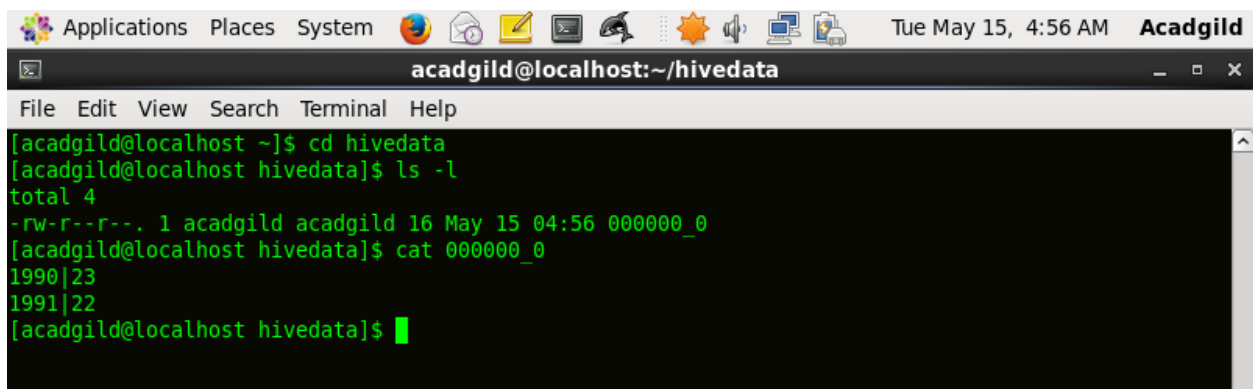
```
hive> create view temperature_data_vw as select substr(tempdate, 7), max(temp) from temperature_data group by substr(tempdate, 7) having count(*)>2;
OK
Time taken: 0.377 seconds
hive> █
```

To export content to a local file system with '|' delimiters use command :

A terminal window titled 'acadgild@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help) and a toolbar. The terminal shows the following command and output:

```
hive> insert overwrite local directory '/home/acadgild/hivedata' row format delimited fields terminated by '|' select * from temperature_data_vw;
```

Lets check the contents of our hivedata local directory:

A terminal window titled 'acadgild@localhost:~/hivedata' with a menu bar (File, Edit, View, Search, Terminal, Help) and a toolbar. The terminal shows the following commands and output:

```
[acadgild@localhost ~]$ cd hivedata
[acadgild@localhost hivedata]$ ls -l
total 4
-rw-r--r--. 1 acadgild acadgild 16 May 15 04:56 000000_0
[acadgild@localhost hivedata]$ cat 000000_0
1990|23
1991|22
[acadgild@localhost hivedata]$ █
```