BIG DATA HADOOP AND SPARK DEVLOPMENT

ASSIGNMENT 8

Table of Contents:

1. Introduction 1
2. Objective 1
3. Problem Statement 1
4. Expected Output
    - Task 1 3
    - Task 2 7

BIG DATA HADOOPAND SPARK DEVELOPMENT

1. Introduction

In this assignment, the given tasks are performed and Output of the tasks are recorded in the form of Screenshots.

2. Objective

This Assignment consolidates the deeper understanding of the Session – 8 Introduction to the HIVE

3. Problem Statement

- 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.

- 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.

4. Expected Output
    - Task 1
- **Create a database named 'custom'**.

Create Database Statement

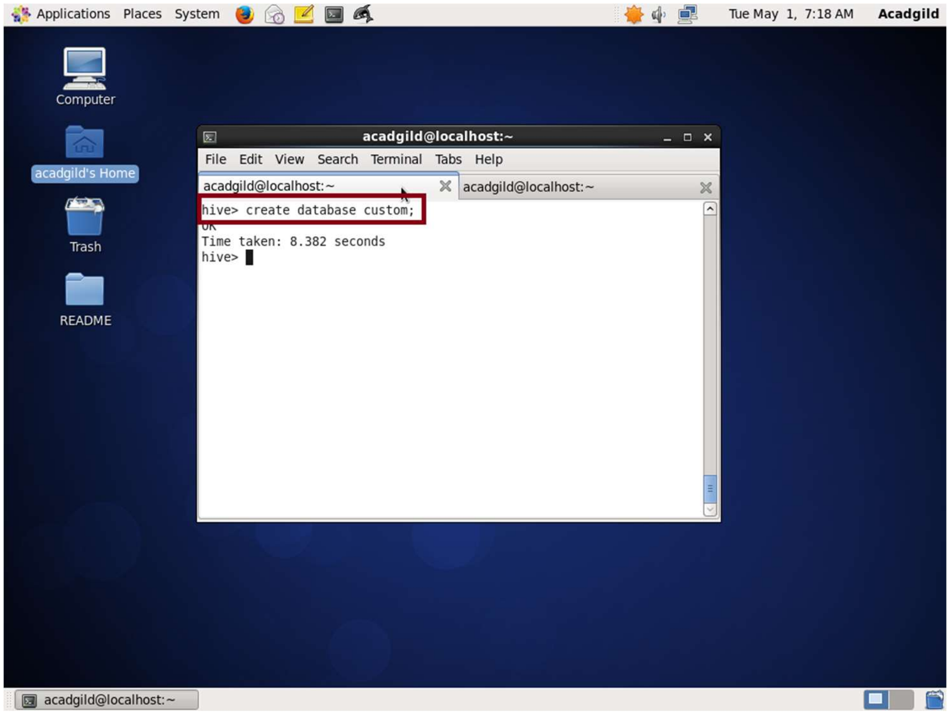
Create Database is a statement used to create a database in Hive. A database in Hive is a namespace or a collection of tables.

Syntax:

```
CREATE DATABASE|SCHEMA [IF NOT EXISTS] <database name>
```

Here the following command is used to create the database custom

create database custom;



```
hive> create database custom;
OK
Time taken: 8.382 seconds
hive> 
```

- **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.

Create Table Statement

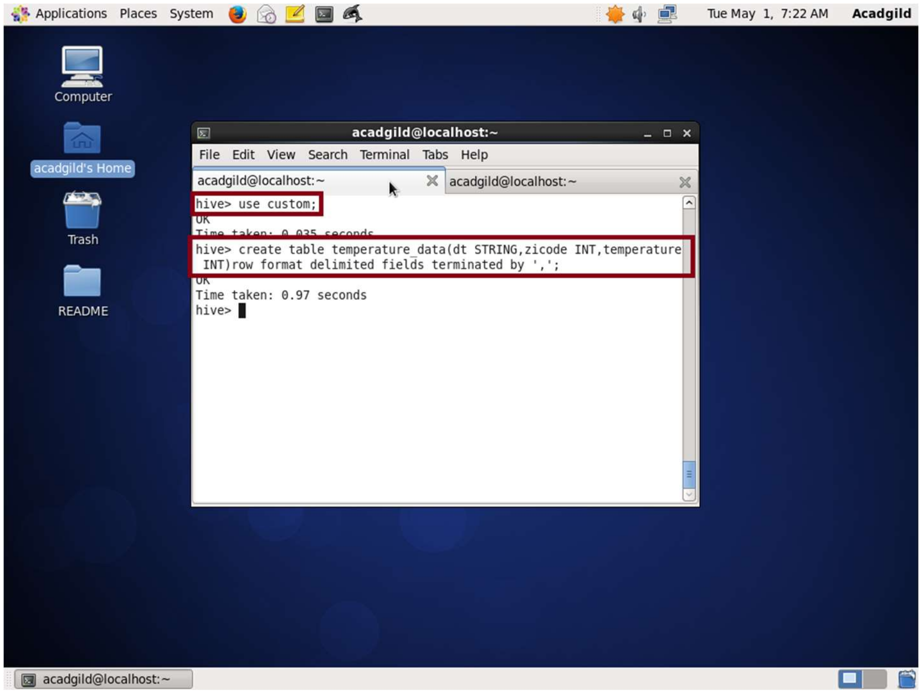
Create Table is a statement used to create a table in Hive.

Syntax:

```
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.] table_name
```

The following command is used to create the table temperature_data with date,zipcode,temperature.

create table temperature_data(dt STRING, zipcode INT, temperature INT) row format delimited fields terminated by ‘,’;



```
hive> use custom;
OK
Time taken: 0.035 seconds
hive> create table temperature_data(dt STRING,zicode INT,temperature
 INT)row format delimited fields terminated by ',';
OK
Time taken: 0.97 seconds
hive> 
```

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

Load Data Statement

Generally, after creating a table in SQL, we can insert data using the Insert statement. But in Hive, we can insert data using the LOAD DATA statement.

While inserting data into Hive, it is better to use LOAD DATA to store bulk records. There are two ways to load data: one is from local file system and second is from Hadoop file system.

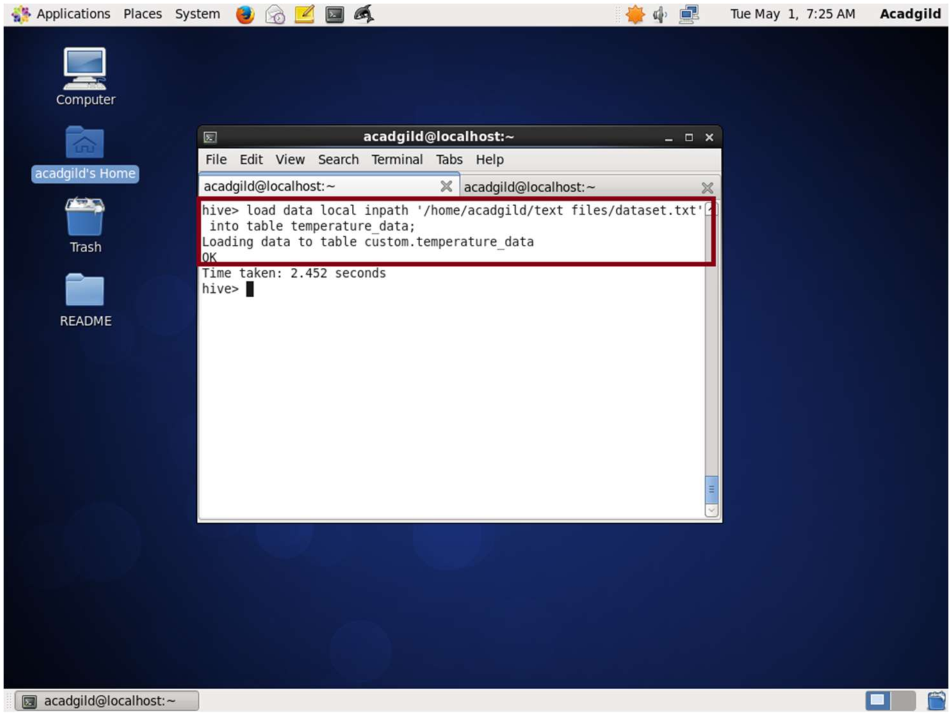
Syntax:

```
LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE tablename
[PARTITION (partcol1=val1, partcol2=val2 ...)]
```

- LOCAL is identifier to specify the local path. It is optional.
- OVERWRITE is optional to overwrite the data in the table.
- PARTITION is optional.

By using the following command, the data for the table temperature_data is loaded.

Load data local inpath '/home/acadgild/text files/dataset.txt' into table temperature_data;

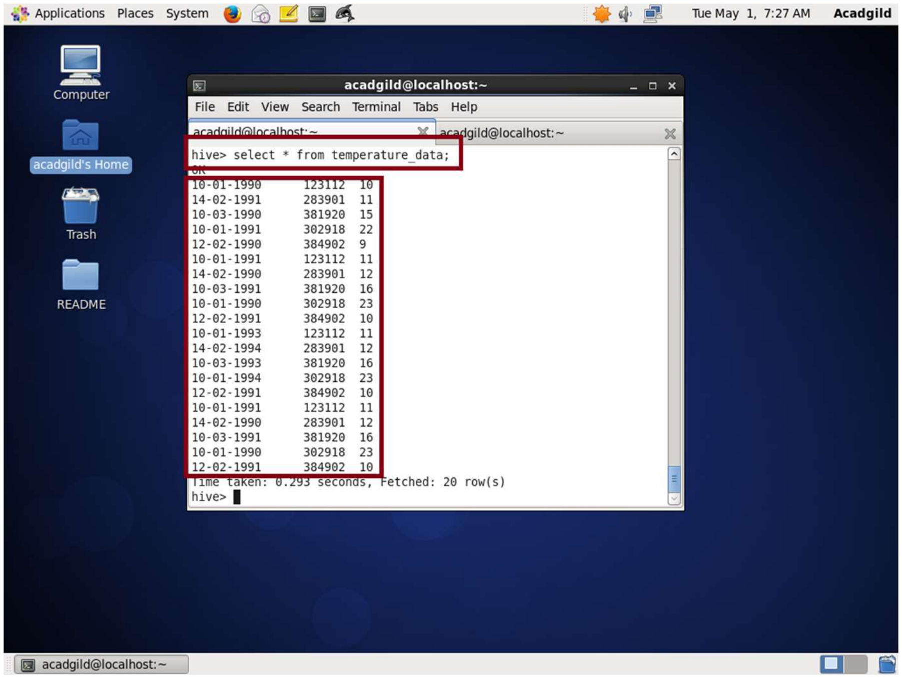


```
hive> load data local inpath '/home/acadgild/text files/dataset.txt'
 into table temperature_data;
Loading data to table custom.temperature_data
OK
Time taken: 2.452 seconds
hive>
```

SELECT statement is used to retrieve the data from a table.

By the below command the data in the table are shown in the console by select command.

select * from temperature_data;



```
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: 0.293 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.**

By using the following command, we can fetch the date and temperature from the table temperature_data where zip code is greater than 300000 and less than 399999.

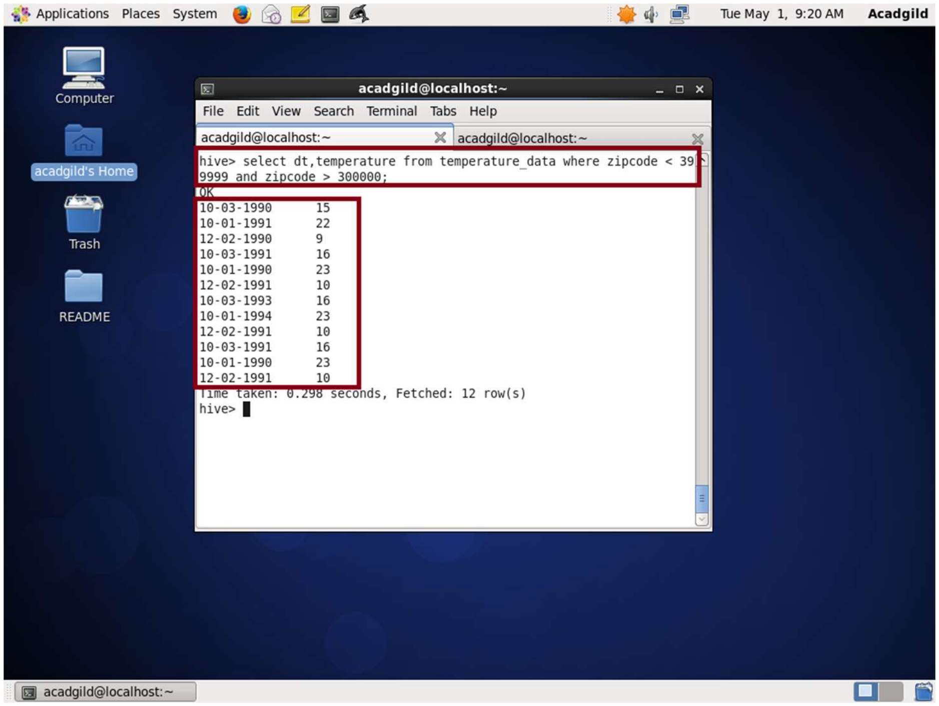
SELECT statement with WHERE clause

SELECT statement is used to retrieve the data from a table. **WHERE** clause works similar to a condition. It filters the data using the condition and gives you a finite result.

Syntax:

```
SELECT [ALL | DISTINCT] select_expr, select_expr, ...

FROM table_reference

[WHERE where_condition]

[GROUP BY col_list]

[HAVING having_condition]

[CLUSTER BY col_list | [DISTRIBUTE BY col_list] [SORT BY col_list]]

[LIMIT number];
```

select dt,temperature from temperature_data where zipcode < 399999 and zipcode > 300000;

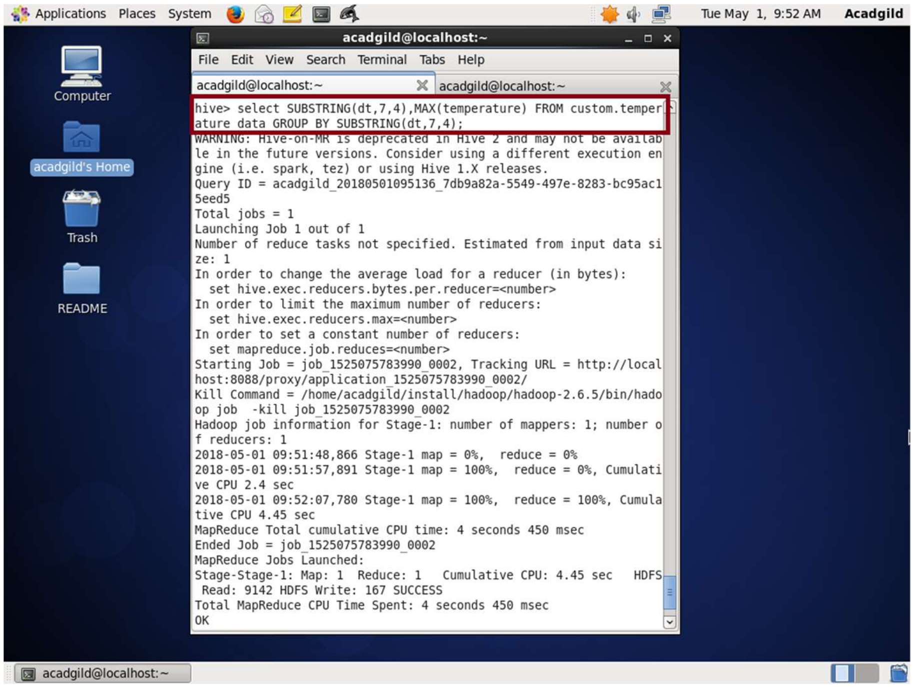


```
hive> select dt,temperature from temperature_data where zipcode < 39
9999 and zipcode > 300000;
OK
10-03-1990      15
10-01-1991      22
12-02-1990      9
10-03-1991      16
10-01-1990      23
12-02-1991      10
10-03-1993      16
10-01-1994      23
12-02-1991      10
10-03-1991      16
10-01-1990      23
12-02-1991      10
Time taken: 0.298 seconds, Fetched: 12 row(s)
hive>
```

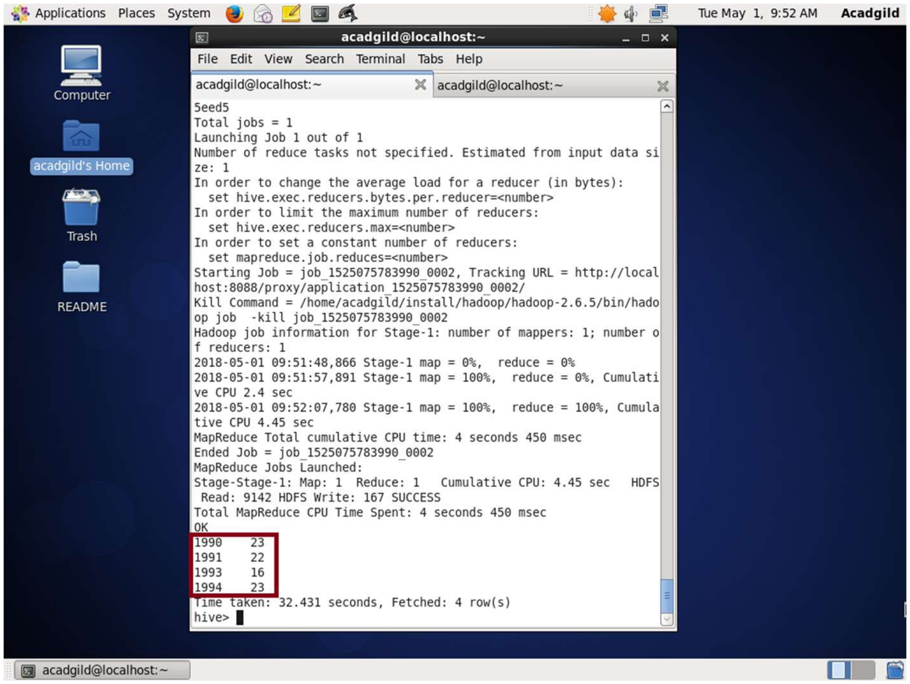
- **Calculate maximum temperature corresponding to every year from temperature_data table.**

By using the following command, Maximum temperature is displayed by using Max function and year is displayed by using Substring function as dt is saved as string format.

Select SUBSTRING(dt,7,4), MAX(temperature) FROM custom.temperature_data GROUP BY SUBSTRING(dt,7,4);



```
hive> select SUBSTRING(dt,7,4),MAX(temperature) FROM custom.temper
ature data GROUP BY SUBSTRING(dt,7,4);
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be availab
le in the future versions. Consider using a different execution en
gine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180501095136_7db9a82a-5549-497e-8283-bc95ac1
5eed5
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data si
ze: 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_1525075783990_0002, Tracking URL = http://local
host:8088/proxy/application_1525075783990_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hado
op job  -kill job_1525075783990_0002
Hadoop job information for Stage-1: number of mappers: 1; number o
f reducers: 1
2018-05-01 09:51:48,866 Stage-1 map = 0%,  reduce = 0%
2018-05-01 09:51:57,891 Stage-1 map = 100%,  reduce = 0%, Cumulati
ve CPU 2.4 sec
2018-05-01 09:52:07,780 Stage-1 map = 100%,  reduce = 100%, Cumula
tive CPU 4.45 sec
MapReduce Total cumulative CPU time: 4 seconds 450 msec
Ended Job = job_1525075783990_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.45 sec   HDFS
 Read: 9142 HDFS Write: 167 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 450 msec
OK
```

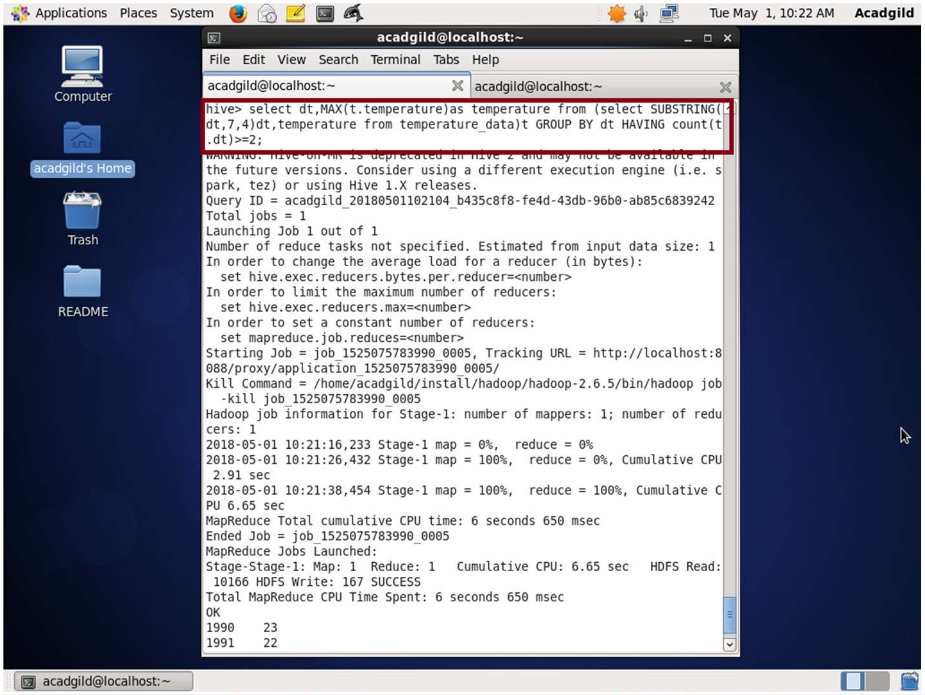


```
5eed5
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data si
ze: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1525075783990_0002, Tracking URL = http://local
host:8088/proxy/application_1525075783990_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hado
op job  -kill job_1525075783990_0002
Hadoop job information for Stage-1: number of mappers: 1; number o
f reducers: 1
2018-05-01 09:51:48,866 Stage-1 map = 0%,  reduce = 0%
2018-05-01 09:51:57,891 Stage-1 map = 100%,  reduce = 0%, Cumulati
ve CPU 2.4 sec
2018-05-01 09:52:07,780 Stage-1 map = 100%,  reduce = 100%, Cumula
tive CPU 4.45 sec
MapReduce Total cumulative CPU time: 4 seconds 450 msec
Ended Job = job_1525075783990_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.45 sec   HDFS
 Read: 9142 HDFS Write: 167 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 450 msec
OK
1990    23
1991    22
1993    16
1994    23
Time taken: 32.431 seconds, Fetched: 4 row(s)
hive>
```

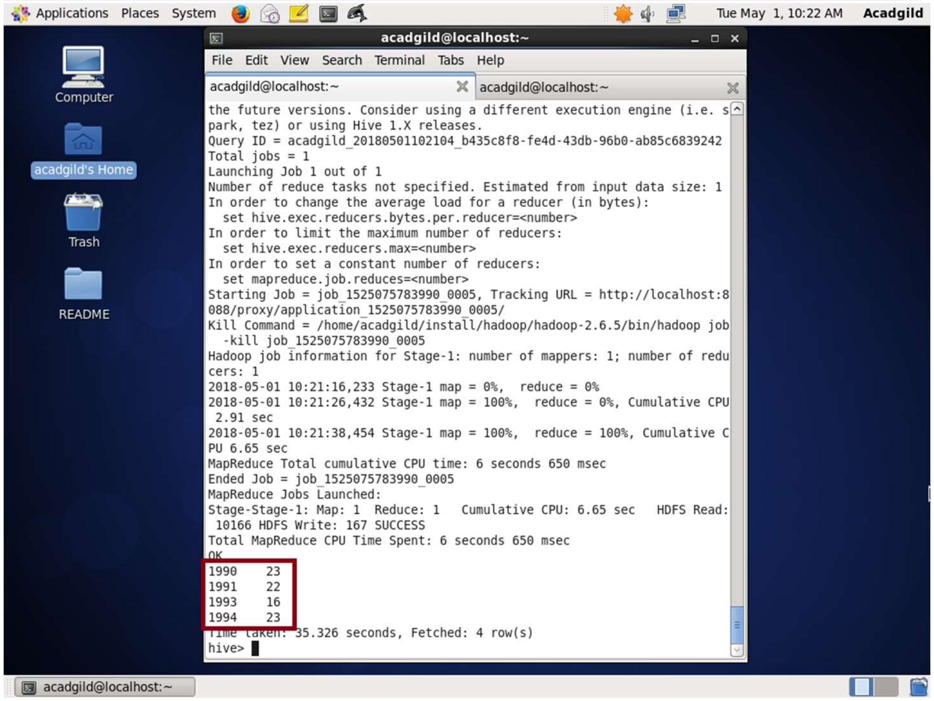
- **Calculate maximum temperature from temperature_data table corresponding to those years which have at least 2 entries in the table.**

By using following command, the maximum temperature is calculated from temperature_data table corresponding to those years which have at least 2 entries in the table.

select dt, MAX(t.temperature)as temperature from (select SUBSTRING(dt,7,4)dt,temperature from temperature_data)t GROUP BY HAVING count (t.dt)>=2;



```
hive> select dt,MAX(t.temperature)as temperature from (select SUBSTRING(
dt,7,4)dt,temperature from temperature_data)t GROUP BY dt HAVING count(t
.dt)>=2;
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. s
park, tez) or using Hive 1.X releases.
Query ID = acadgild_20180501102104_b435c8f8-fe4d-43db-96b0-ab85c6839242
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_1525075783990_0005, Tracking URL = http://localhost:8
088/proxy/application_1525075783990_0005/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job
  -kill job_1525075783990_0005
Hadoop job information for Stage-1: number of mappers: 1; number of redu
cers: 1
2018-05-01 10:21:16,233 Stage-1 map = 0%,  reduce = 0%
2018-05-01 10:21:26,432 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU
 2.91 sec
2018-05-01 10:21:38,454 Stage-1 map = 100%,  reduce = 100%, Cumulative C
PU 6.65 sec
MapReduce Total cumulative CPU time: 6 seconds 650 msec
Ended Job = job_1525075783990_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.65 sec   HDFS Read:
 10166 HDFS Write: 167 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 650 msec
OK
1990    23
1991    22
```



```
the future versions. Consider using a different execution engine (i.e. s
park, tez) or using Hive 1.X releases.
Query ID = acadgild_20180501102104_b435c8f8-fe4d-43db-96b0-ab85c6839242
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_1525075783990_0005, Tracking URL = http://localhost:8
088/proxy/application_1525075783990_0005/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job
  -kill job_1525075783990_0005
Hadoop job information for Stage-1: number of mappers: 1; number of redu
cers: 1
2018-05-01 10:21:16,233 Stage-1 map = 0%,  reduce = 0%
2018-05-01 10:21:26,432 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU
 2.91 sec
2018-05-01 10:21:38,454 Stage-1 map = 100%,  reduce = 100%, Cumulative C
PU 6.65 sec
MapReduce Total cumulative CPU time: 6 seconds 650 msec
Ended Job = job_1525075783990_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.65 sec   HDFS Read:
 10166 HDFS Write: 167 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 650 msec
OK
1990    23
1991    22
1993    16
1994    23
Time taken: 35.326 seconds, Fetched: 4 row(s)
hive> 
```

- **Create a view on the top of last query, name it as temperature_data_vw.**

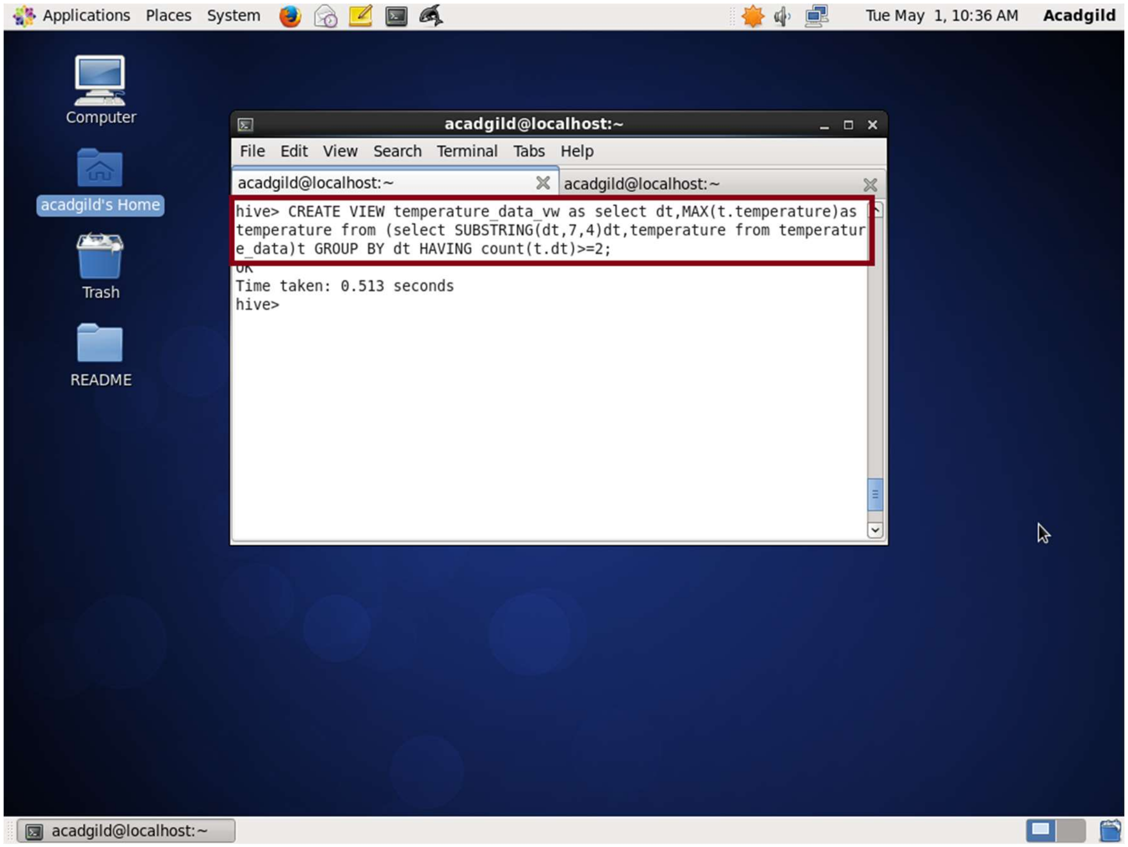
Creating a View

You can create a view at the time of executing a SELECT statement. The syntax is as follows:

```
CREATE VIEW [IF NOT EXISTS] view_name [(column_name [COMMENT column_comment], ...) ]

[COMMENT table_comment]

AS SELECT ...
```

By using the following command, we can create a view as temperature_data_vw on last query

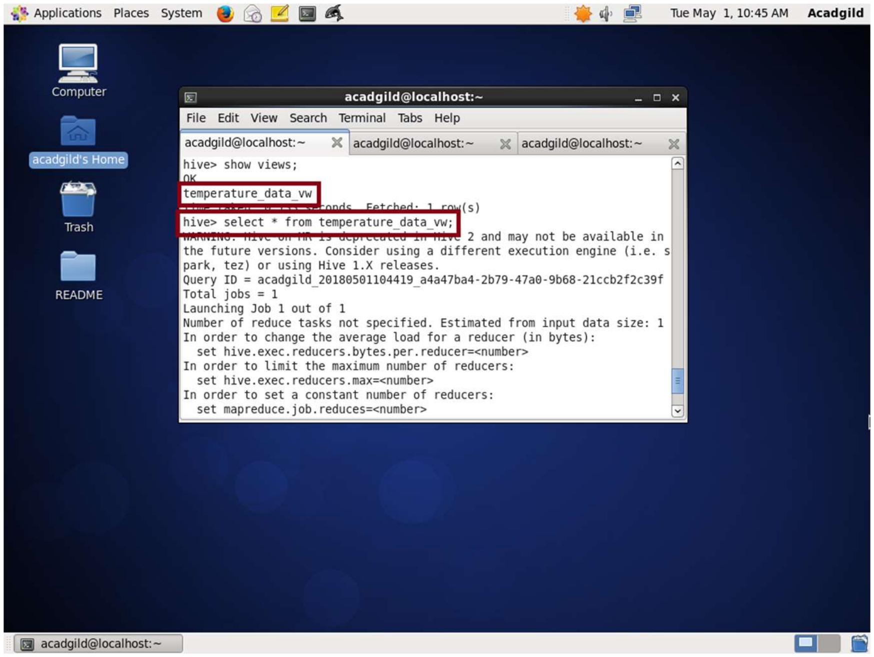
CREATE VIEW temperature_data_vw as select dt,MAX(t.temperature)as temperature from (select SUBSTRING(dt,7,4),temperature from temperature_data)t GROUP BY dt HAVING count(t.dt)>=2;



```
hive> CREATE VIEW temperature_data_vw as select dt,MAX(t.temperature)as
temperature from (select SUBSTRING(dt,7,4)dt,temperature from temperatur
e_data)t GROUP BY dt HAVING count(t.dt)>=2;
OK
Time taken: 0.513 seconds
hive>
```

By using the show views following command, the views are listed which are created under the custom database

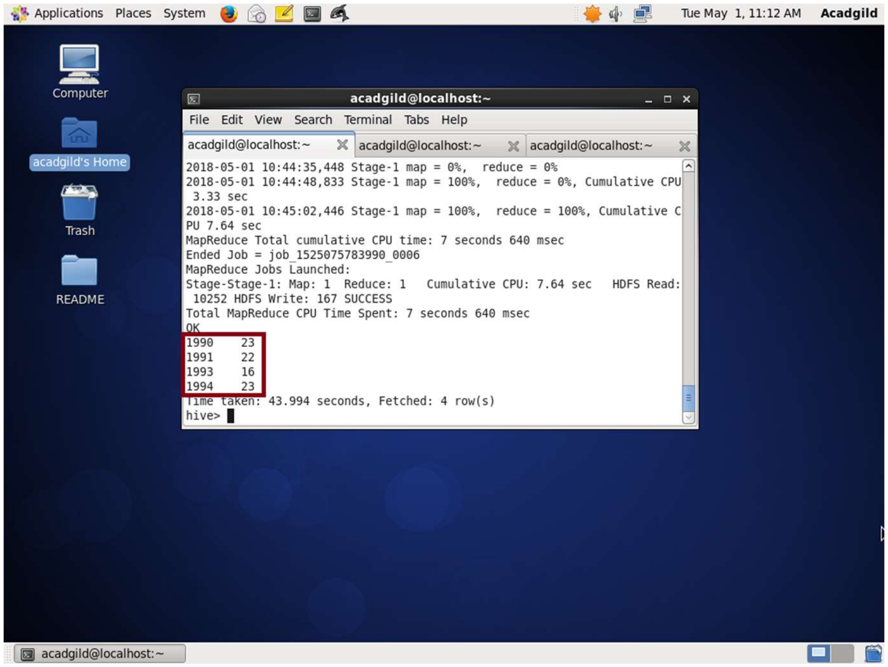
show views;



```
hive> show views;
OK
temperature_data_vw
Time taken: 0.1[illegible] seconds, Fetched: 1 row(s)
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 execution engine (i.e. s
park, tez) or using Hive 1.X releases.
Query ID = acadgild_20180501104419_a4a47ba4-2b79-47a0-9b68-21ccb2f2c39f
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>
```

To check the saved view, we are using select command to list the data in the view.

select * from temperature_data_vw;

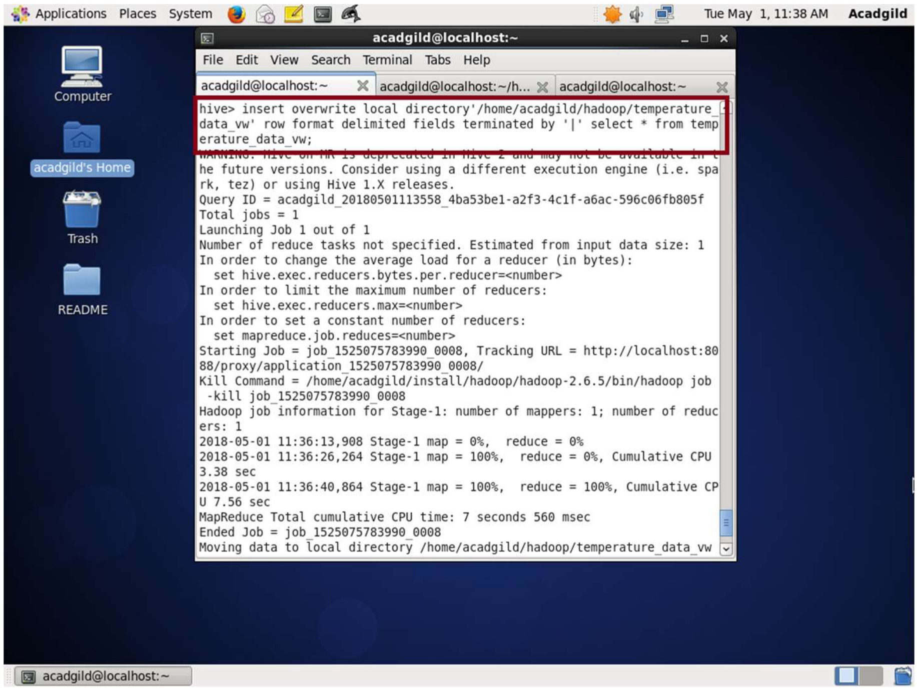


```
2018-05-01 10:44:35,448 Stage-1 map = 0%,  reduce = 0%
2018-05-01 10:44:48,833 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU
 3.33 sec
2018-05-01 10:45:02,446 Stage-1 map = 100%,  reduce = 100%, Cumulative C
PU 7.64 sec
MapReduce Total cumulative CPU time: 7 seconds 640 msec
Ended Job = job_1525075783990_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 7.64 sec   HDFS Read:
 10252 HDFS Write: 167 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 640 msec
OK
1990    23
1991    22
1993    16
1994    23
Time taken: 43.994 seconds, Fetched: 4 row(s)
hive>
```

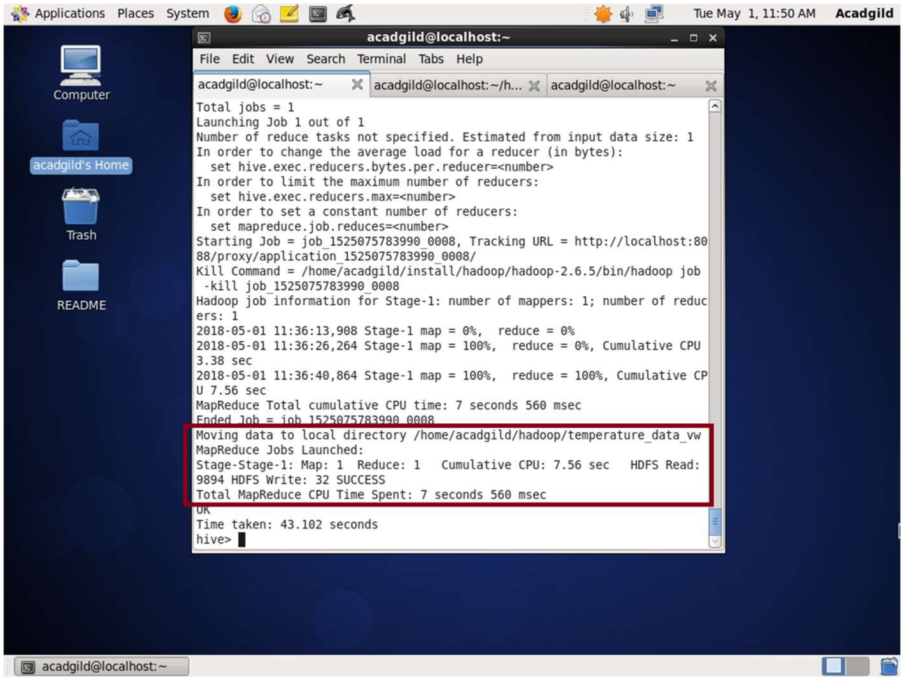
- **Export contents from temperature_data_vw to a file in local file system, such that each file is '|' delimited.**

By using the following command, we can create a new directory temperature_data_vw and we can create a text file and save the data with delimited fields terminated by '|'

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



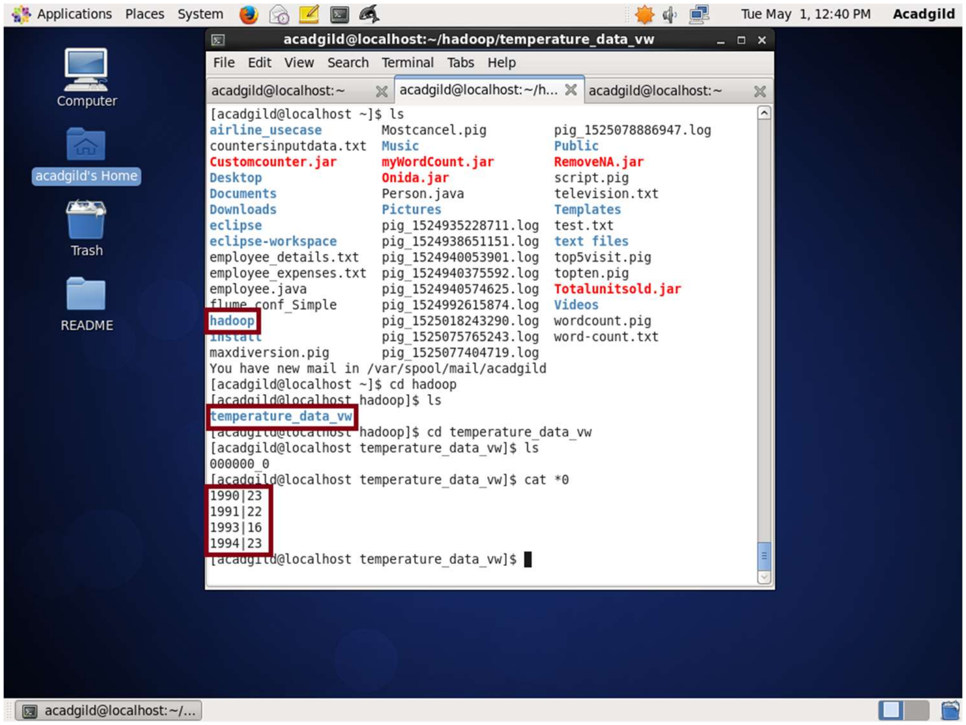
```
hive> insert overwrite local directory'/home/acadgild/hadoop/temperature_
data_vw' row format delimited fields terminated by '|' select * from temp
erature_data_vw;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in t
he future versions. Consider using a different execution engine (i.e. spa
rk, tez) or using Hive 1.X releases.
Query ID = acadgild_20180501113558_4ba53be1-a2f3-4c1f-a6ac-596c06fb805f
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_1525075783990_0008, Tracking URL = http://localhost:80
88/proxy/application_1525075783990_0008/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job
 -kill job_1525075783990_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reduc
ers: 1
2018-05-01 11:36:13,908 Stage-1 map = 0%,  reduce = 0%
2018-05-01 11:36:26,264 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU
3.38 sec
2018-05-01 11:36:40,864 Stage-1 map = 100%,  reduce = 100%, Cumulative CP
U 7.56 sec
MapReduce Total cumulative CPU time: 7 seconds 560 msec
Ended Job = job_1525075783990_0008
Moving data to local directory /home/acadgild/hadoop/temperature_data_vw
```



```
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_1525075783990_0008, Tracking URL = http://localhost:80
88/proxy/application_1525075783990_0008/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job
 -kill job_1525075783990_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reduc
ers: 1
2018-05-01 11:36:13,908 Stage-1 map = 0%,  reduce = 0%
2018-05-01 11:36:26,264 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU
3.38 sec
2018-05-01 11:36:40,864 Stage-1 map = 100%,  reduce = 100%, Cumulative CP
U 7.56 sec
MapReduce Total cumulative CPU time: 7 seconds 560 msec
Ended Job = job_1525075783990_0008
Moving data to local directory /home/acadgild/hadoop/temperature_data_vw
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 7.56 sec   HDFS Read:
9894 HDFS Write: 32 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 560 msec
OK
Time taken: 43.102 seconds
hive>
```

Contents are exported from temperature_data_vw to a file in local file system, such that each file is '|' delimited.

The location of the file is /home/acadgild/Hadoop/temperature_data_vw



```
[acadgild@localhost ~]$ ls
airline_usecase         Mostcancel.pig         pig_1525078886947.log
countersinputdata.txt   Music                  Public
Customcounter.jar       myWordCount.jar        RemoveNA.jar
Desktop                 Onida.jar              script.pig
Documents               Person.java            television.txt
Downloads               Pictures               Templates
eclipse                 pig_1524935228711.log  test.txt
eclipse-workspace       pig_1524938651151.log  text files
employee_details.txt    pig_1524940053901.log  top5visit.pig
employee_expenses.txt   pig_1524940375592.log  topten.pig
employee.java           pig_1524940574625.log  Totalunitsold.jar
flume_conf_Simple       pig_1524992615874.log  Videos
hadoop                  pig_1525018243290.log  wordcount.pig
install                 pig_1525075765243.log  word-count.txt
maxdiversion.pig        pig_1525077404719.log
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ cd hadoop
[acadgild@localhost hadoop]$ ls
temperature_data_vw
[acadgild@localhost hadoop]$ cd temperature_data_vw
[acadgild@localhost temperature_data_vw]$ ls
000000_0
[acadgild@localhost temperature_data_vw]$ cat *0
1990|23
1991|22
1993|16
1994|23
[acadgild@localhost temperature_data_vw]$ 
```

By using,

Ls command, the files are listed.

Cat command, the contents in the file are displayed in the console.