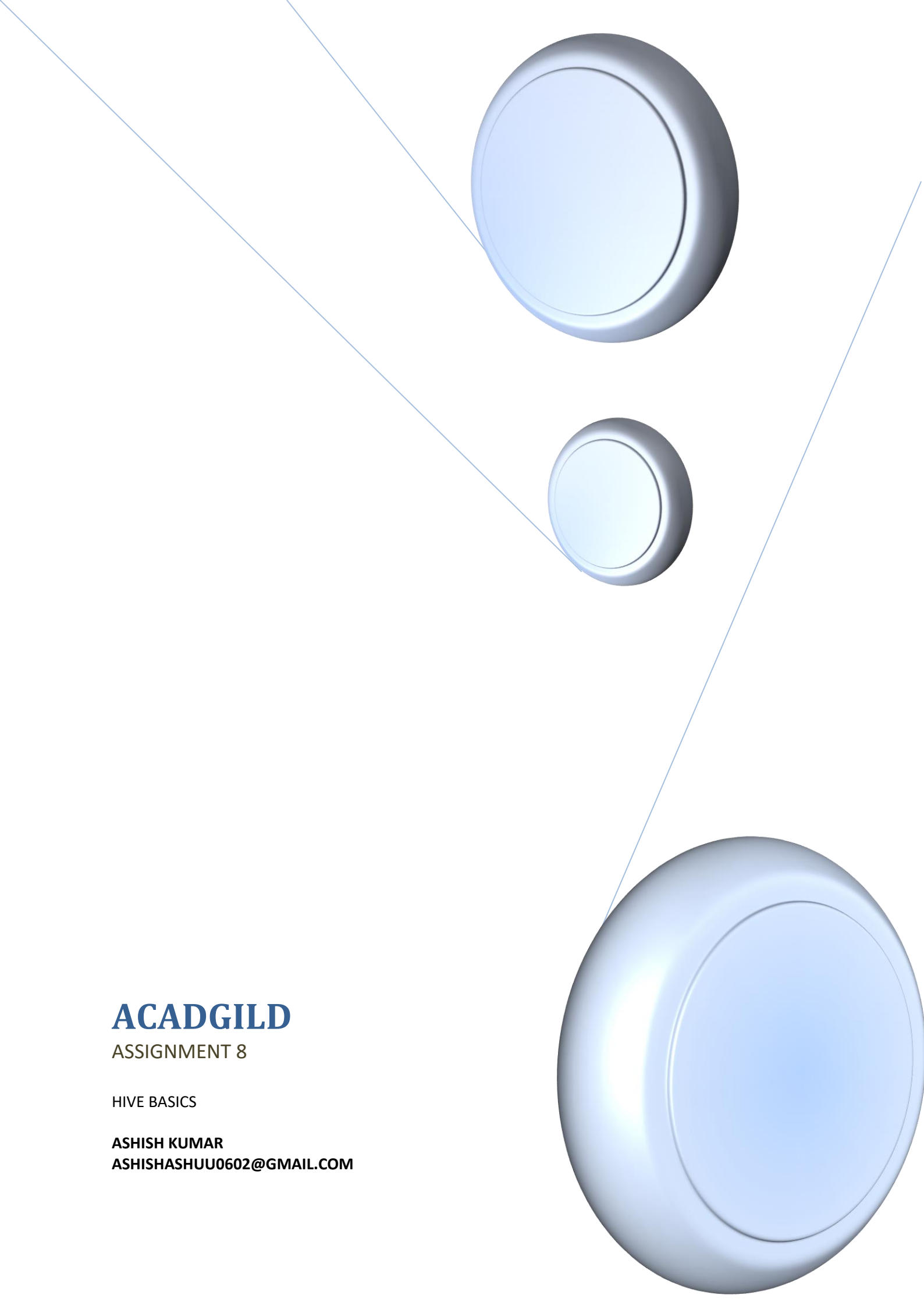


ACADGILD

ASSIGNMENT 8

HIVE BASICS

ASHISH KUMAR
ASHISHASHUU0602@GMAIL.COM

HIVE BASICS

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.

Input –

```
[acadgild@192 ~]$ cat /home/acadgild/Desktop/Assignment8/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[acadgild@192 ~]$
```

The provided Input is in DD-MM-YYYY format but the table which need to be created should have field as “MM-DD-YYYY” format.

So, to achieve this we have to use from_unixtime function.

We have to create a temporary table to store data from text input file and then we will insert this data to temperature-data table from temporary table using from_unixtime function.

Creating tempotbl table in custom database.

Commands –

HIVE BASICS

1. To create database.
   Create database custom;

```
hive> show databases;
OK
default
simplidb
Time taken: 6.379 seconds, Fetched: 2 row(s)
hive> create database custom;
OK
Time taken: 0.218 seconds
hive> show databases;
OK
custom
default
simplidb
Time taken: 0.042 seconds, Fetched: 3 row(s)
hive>
```

We have created a temporary table first and load data from dataset.txt file into this temporary table. Then we have inserted data into **'temperature_data'** table from this temporary table using insert into select statement.

temporary table created :

Command –

Create table temporary (tdate string, zipcode int, temperature int) row format delimited fields terminated by ',';

```
hive> Create table temporary (tdate string, zipcode int, temperature int) row format delimited fields terminated by ',';
OK
Time taken: 1.103 seconds
hive> select * from temporary;
OK
Time taken: 2.842 seconds
hive>
```


Loading Data from Input dataset into the temporary table:

LOAD DATA LOCAL INPATH '/home/acadgild/Desktop/Assignment8/Dataset_Session 14.txt into table temporary;

```
hive> LOAD DATA LOCAL INPATH '/home/acadgild/Desktop/Assignment8/dataset_Session 14.txt' into table temporary;
Loading data to table custom.temporary
OK
Time taken: 2.534 seconds
hive> select * from temporary;
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: 0.212 seconds, Fetched: 20 row(s)
hive>
```

Creation of temperature_data table to store values

```
hive> Create table temperature_data (date_val string, zipcode int, temperature int) row format delimited fields terminated by ',';
OK
Time taken: 0.291 seconds
hive>
```

Command to create temperature_data table –

```
Create table temperature_data (date_val string, zipcode int, temperature int) row format delimited fields terminated by ',';
```

Inserting data into **'temperature_data'** table from this **temporary** table using below insert into select statement with the help of from_unixtime and unix_timestamp functions.

Command-

```
insert into table temperature_data select from_unixtime(unix_timestamp(tdate, 'dd-mm-yyyy'), 'mm-dd-yyyy'),zipcode,temperature from temporary;
```

```
hive> insert into table temperature_data select from_unixtime(unix_timestamp(tdate, 'dd-mm-yyyy'), 'mm-dd-yyyy'),zipcode,temperature from temporary;
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 u
sing Hive 1.X releases.
Query ID = acadgild_20180808012806_aff32216-a0a1-4f86-9ba1-7a45ffffb1bb
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_1533665221456_0001, Tracking URL = http://192.168.0.11:8088/proxy/application_1533665221456_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533665221456_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2018-08-08 01:28:19,996 Stage-1 map = 0%,  reduce = 0%
2018-08-08 01:28:28,904 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.15 sec
MapReduce Total cumulative CPU time: 3 seconds 150 msec
Ended Job = job_1533665221456_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_hive_2018-08-08_01-28-06_098_743546704045639920
7-1/-ext-10000
Loading data to table custom.temperature_data
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 3.15 sec   HDFS Read: 4913 HDFS Write: 499 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 150 msec
OK
Time taken: 24.926 seconds
hive>
```

After converting date to ‘ MM-DD-YYYY’ format –

```
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: 0.193 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.**

Setting column header to TRUE so that we can have column headers along with output.

hive> **set hive.cli.print.header=true;**

```
hive> select date_val, temperature from temperature_data where zipcode > 300000 and zipcode <399999;
OK
date_val        temperature
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: 3.055 seconds, Fetched: 12 row(s)
hive>
```

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

We have used below select query by using max_temp and year as column alias for table :

Output shows Maximum temperature corresponding to every year.

select max(temperature) max_temp, date_format(from_unixtime(unix_timestamp(date_val,'mm-dd-yyyy'),'yyyy-mm-dd'),'yyyy') year from temperature_data group by date_format(from_unixtime(unix_timestamp(date_val, 'mm-dd-yyyy'),'yyyy-mm-dd'),'yyyy');

```
hive> select max(temperature) max_temp, date_format(from_unixtime(unix_timestamp(date_val,'mm-dd-yyyy'),'yyyy-mm-dd'),'yyyy') year from temperature_data group
by date_format(from_unixtime(unix_timestamp(date_val, 'mm-dd-yyyy'),'yyyy-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 execution engine (i.e. spark, tez) or u
sing Hive 1.X releases.
Query ID = acadgild_20180808033047_60f9a46d-77a8-4f3a-bf86-043e4256f1c4
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_1533665221456_0002, Tracking URL = http://192.168.0.11:8088/proxy/application_1533665221456_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533665221456_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-08 03:30:58,292 Stage-1 map = 0%,  reduce = 0%
2018-08-08 03:31:06,177 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.65 sec
2018-08-08 03:31:14,822 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.27 sec
MapReduce Total cumulative CPU time: 5 seconds 270 msec
Ended Job = job_1533665221456_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.27 sec   HDFS Read: 9757 HDFS Write: 167 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 270 msec
OK
23      1990
22      1991
16      1993
23      1994
Time taken: 28.434 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.**

We have used below select query by using max_temp and year as column alias and count function for each year for table :

Output shows Maximum temperature corresponding to every year having count of rows for each year as at least 2.

Query is –

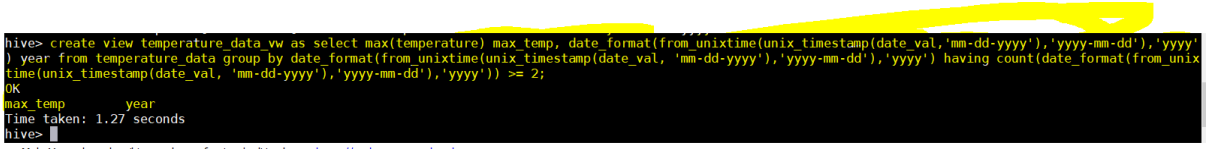
select max(temperature) max_temp, date_format(from_unixtime(unix_timestamp(date_val,'mm-dd-yyyy'),'yyyy-mm-dd'),'yyyy') year from temperature_data group by date_format(from_unixtime(unix_timestamp(date_val, 'mm-dd-yyyy'),'yyyy-mm-dd'),'yyyy') having count(date_format(from_unixtime(unix_timestamp(date_val, 'mm-dd-yyyy'),'yyyy-mm-dd'),'yyyy')) >= 2;

```
hive> select max(temperature) max_temp, date_format(from_unixtime(unix_timestamp(date_val,'mm-dd-yyyy'),'yyyy-mm-dd'),'yyyy') year from temperature_data group
by date_format(from_unixtime(unix_timestamp(date_val, 'mm-dd-yyyy'),'yyyy-mm-dd'),'yyyy') having count(date_format(from_unixtime(unix_timestamp(date_val, 'mm-d
d-yyyy'),'yyyy-mm-dd'),'yyyy')) >= 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. spark, tez) or u
sing Hive 1.X releases.
Query ID = acadgild_20180808033714_d3106fef-2631-4e8d-8b6b-73b40bf7b6b6
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_1533665221456_0004, Tracking URL = http://192.168.0.11:8088/proxy/application_1533665221456_0004/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533665221456_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-08 03:37:22,278 Stage-1 map = 0%,  reduce = 0%
2018-08-08 03:37:28,666 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.7 sec
2018-08-08 03:37:36,072 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 6.43 sec
MapReduce Total cumulative CPU time: 6 seconds 430 msec
Ended Job = job_1533665221456_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.43 sec   HDFS Read: 10668 HDFS Write: 167 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 430 msec
OK
max_temp        year
23      1990
22      1991
16      1993
23      1994
Time taken: 23.297 seconds, Fetched: 4 row(s)
hive>
```

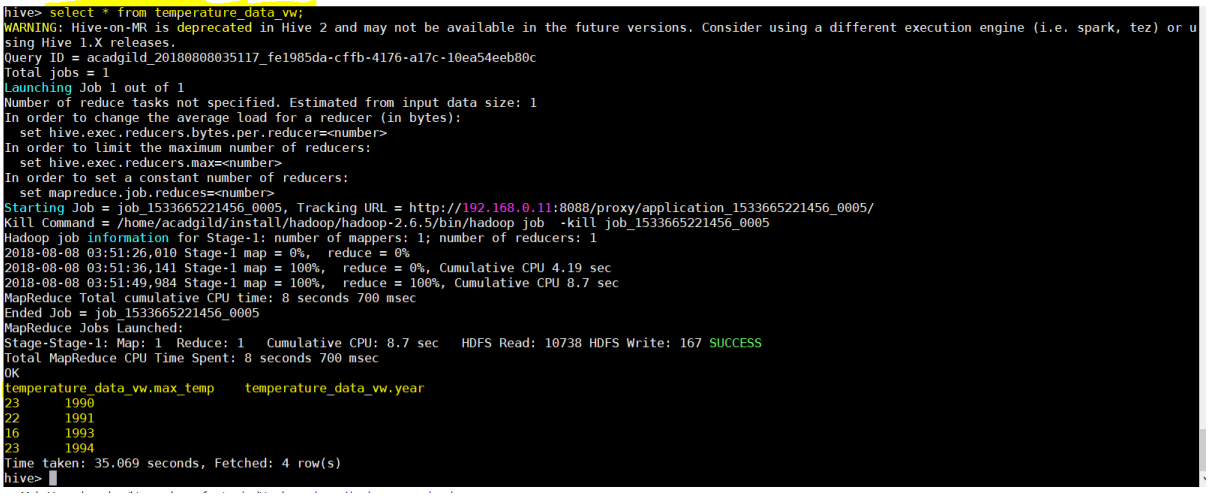
4. **Create a view on the top of last query, name it temperature_data_vw.**

Query is-

create view temperature_data_vw as select max(temperature) max_temp, date_format(from_unixtime(unix_timestamp(date_val,'mm-dd-yyyy'),'yyyy-mm-dd'),'yyyy') year from temperature_data group by date_format(from_unixtime(unix_timestamp(date_val, 'mm-dd-yyyy'),'yyyy-mm-dd'),'yyyy') having count(date_format(from_unixtime(unix_timestamp(date_val, 'mm-dd-yyyy'),'yyyy-mm-dd'),'yyyy')) >= 2;



select * from temperature_data_vw;



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

Query is-

insert overwrite local directory 'home/acadgild/Desktop/Assignment8/taskexpo' row format delimited fields terminated by '|' select * from temperature_data_vw;

```
hive> insert overwrite local directory 'home/acadgild/Desktop/Assignment8/taskexpo' row format delimited fields terminated by '|' select * from temperature_dat
a_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 u
sing Hive 1.X releases.
Query ID = acadgild_20180808035938_1f7f80f3-6a93-4155-906f-b6fbe677d54a
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_1533665221456_0006, Tracking URL = http://192.168.0.11:8088/proxy/application_1533665221456_0006/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533665221456_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-08 03:59:48,195 Stage-1 map = 0%,  reduce = 0%
2018-08-08 03:59:59,248 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.64 sec
2018-08-08 04:00:11,615 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 9.49 sec
MapReduce Total cumulative CPU time: 9 seconds 490 msec
Ended Job = job_1533665221456_0006
Moving data to local directory home/acadgild/Desktop/Assignment8/taskexpo
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 9.49 sec   HDFS Read: 10397 HDFS Write: 32 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 490 msec
OK
temperature_data_vw.max_temp    temperature_data_vw.year
Time taken: 35.374 seconds
hive>
```

Directory is created and the data is exported from view table to the **taskexpo** directory under 000000_0.

```
[acadgild@192 ~]$ ls -l /home/acadgild/Desktop/Assignment8/taskexpo
total 4
-rw-r--r--. 1 acadgild acadgild 32 Aug  8 04:10 000000_0
[acadgild@192 ~]$ cat /home/acadgild/Desktop/Assignment8/taskexpo/000000_0
23|1990
22|1991
16|1993
23|1994
[acadgild@192 ~]$
```