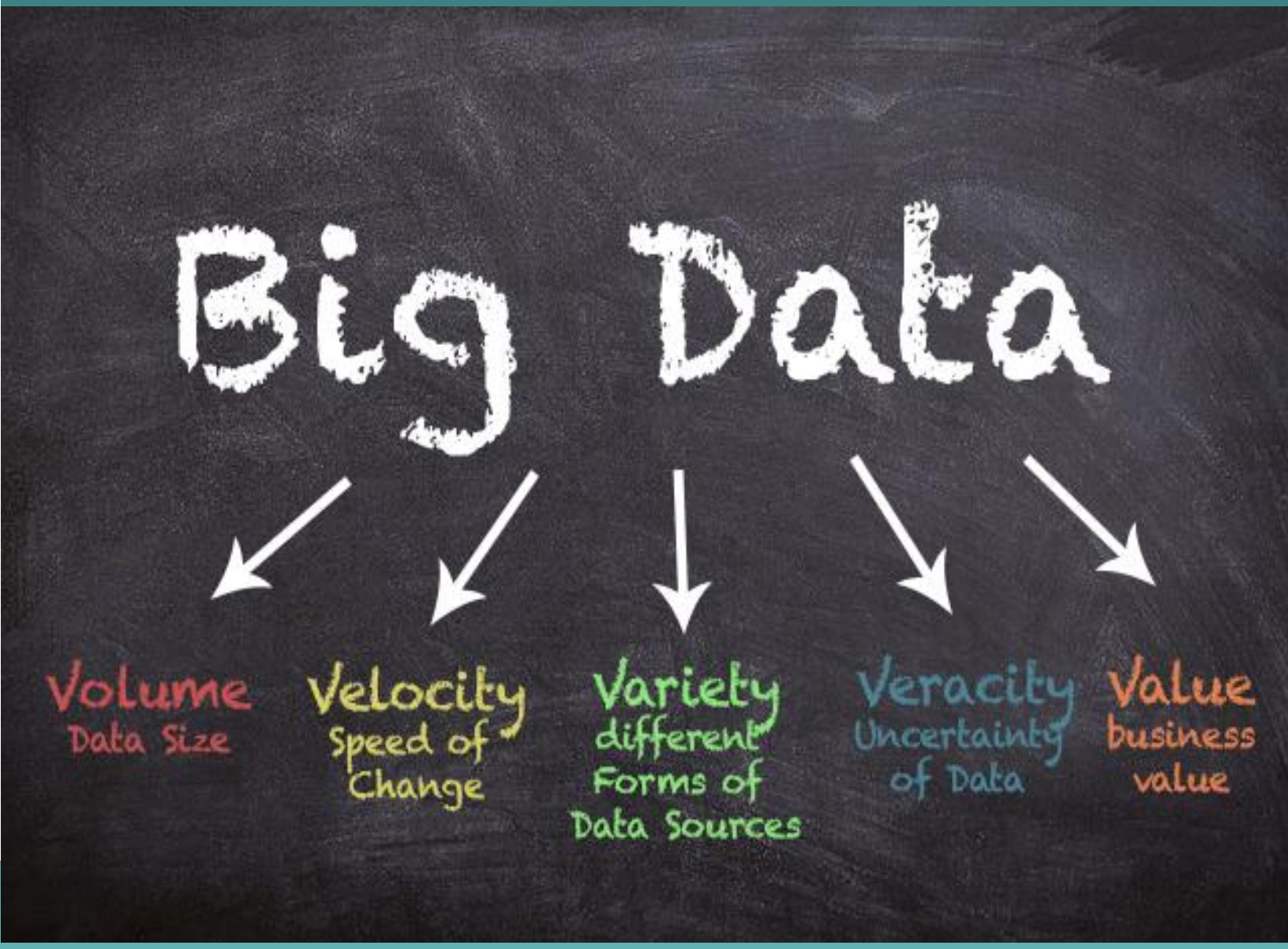
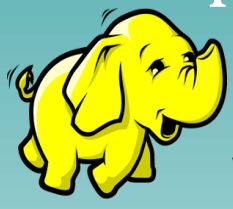
08/07/2018 | By: Ajit K Prasad



Big Data Engineering with Hadoop & Spark

Assignment on Hive Basics



Acadgild

Session 8: Assignment 8.1

This assignment is aimed at consolidating the concepts that was learnt during the Hive Basics session of the course.

Associated Data Files

Download ***"temperature_data.txt"*** from the link given below:
https://drive.google.com/file/d/0Bxr27gVaXO5sa0JBamZXdkpYUFk/view

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

Problem Statement

Task 1:

- Create a database named ***'custom'***.
- Create a table named ***temperature_data*** inside ***custom*** having below fields:
  - date (mm-dd-yyyy) format
  - zip code
  - temperature
- The table will be loaded from comma-delimited file.
- Load the ***temperature_data.txt*** (which is ',' delimited) in the table.

Solution:

- Start ***hive shell*** with the following command
  $ hive
- Execute the following commands on ***hive shell*** to complete the task
  hive> create database custom;
  hive> show databases;
  hive> use custom;
  hive> create table temperature_data(tdate string, zipcode int, temperature int)row format delimited fields terminated by ',';
  hive> show tables;
  hive> desc temperature_data;
  hive> LOAD DATA LOCAL INPATH '/home/acadgild/HiveExamples/temperature_data.txt' INTO TABLE temperature_data;
  *hive> select * from temperature_data;*

Output:

```
hive>
hive> show databases;
OK
default
simplidb
test
Time taken: 21.834 seconds, Fetched: 3 row(s)
hive> create database custom;
OK
Time taken: 0.654 seconds
hive> show databases;
OK
custom
default
simplidb
test
Time taken: 0.083 seconds, Fetched: 4 row(s)
hive> use custom;
OK
Time taken: 0.05 seconds
hive> create table temperature_data(tdate string, zipcode int, temperature int)row format delimited fields terminated by ',';
OK
Time taken: 2.198 seconds
hive> show tables;
OK
temperature_data
Time taken: 0.14 seconds, Fetched: 1 row(s)
hive> desc temperature_data;
OK
tdate                   string
zipcode                 int
temperature             int
Time taken: 0.545 seconds, Fetched: 3 row(s)
```

```
hive> LOAD DATA LOCAL INPATH '/home/acadgild/HiveExamples/temperature_data.txt' INTO TABLE temperature_data;
Loading data to table custom.temperature_data
OK
Time taken: 5.238 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: 6.165 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.

Solution:

– To perform the task use the query command:
hive> select tdate,temperature from temperature_data where zipcode>300000 and zipcode<399999;

Output:

```
hive> select tdate,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: 8.644 seconds, Fetched: 12 row(s)
hive>
```

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

Solution:

- To perform the task, use the query command:
  hive> select max(temperature), from_unixtime(unix_timestamp(tdate,'MM-dd-yyyy'),'MM-dd-yyyy') as new_date from temperature_data group by from_unixtime(unix_timestamp(tdate,'MM-dd-yyyy'),'MM-dd-yyyy');

Explanation:

- ***Max(temperature):*** this will find the maximum temperature.
- ***from_unixtime(unix_timestamp(tdate,'MM-dd-yyyy'),'yyyy'):*** this to format date in the month-date-year format and display only the year.
- This give the list of maximum temperature across all the years in the given data.

Output:

```
hive> select max(temperature), from_unixtime(unix_timestamp(tdate,'MM-dd-yyyy'),'MM-dd-yyyy') as new_date from temperature_data gro
up by from_unixtime(unix_timestamp(tdate,'MM-dd-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 e
ngine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180808130515_760f81f3-e9bb-4d3b-8e5c-1b3594858625
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_1533711610348_0001, Tracking URL = http://localhost:8088/proxy/application_1533711610348_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533711610348_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-08 13:05:54,900 Stage-1 map = 0%,  reduce = 0%
2018-08-08 13:06:17,065 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 10.06 sec
2018-08-08 13:06:36,258 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 17.89 sec
MapReduce Total cumulative CPU time: 17 seconds 890 msec
Ended Job = job_1533711610348_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 17.89 sec   HDFS Read: 9513 HDFS Write: 398 SUCCESS
Total MapReduce CPU Time Spent: 17 seconds 890 msec
OK
12      02-02-1991
11      02-02-1992
12      02-02-1995
23      10-01-1990
22      10-01-1991
11      10-01-1993
23      10-01-1994
15      10-03-1990
16      10-03-1991
16      10-03-1993
9       12-02-1990
10      12-02-1991
Time taken: 83.096 seconds, Fetched: 12 row(s)
hive>
```

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

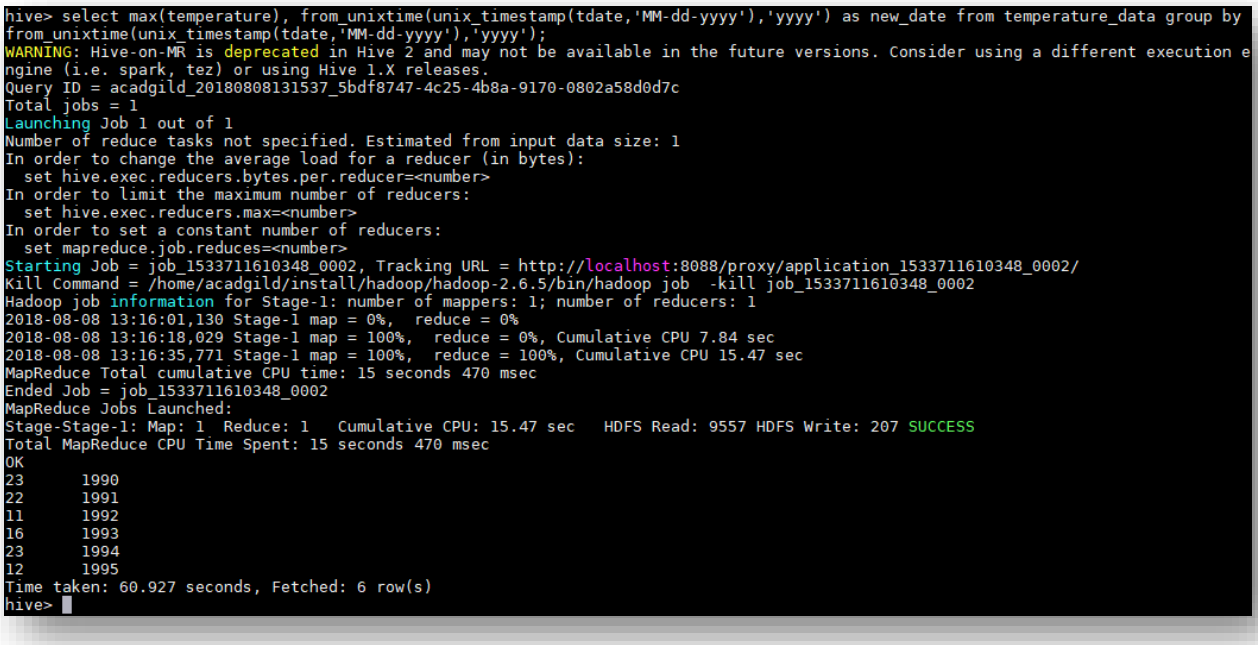
Solution:

- To perform the task, use the query command:
  hive> select max(temperature), from_unixtime(unix_timestamp(tdate,'MM-dd-yyyy'),'yyyy') as new_date from temperature_data group by from_unixtime(unix_timestamp(tdate,'MM-dd-yyyy'),'yyyy');

Explanation:

- ***Max(temperature):*** this will find the maximum temperature.
- ***from_unixtime(unix_timestamp(tdate,'MM-dd-yyyy'),'yyyy'):*** this to format date in the month-date-year format and display only the year.
- This give the list of maximum temperature from temperature_data table corresponding to those years which have at least 2 entries in the table.

Output:



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

Solution:

– To perform the task, use the query command:

hive> create view temperature_data_vw(tdate,temperature) comment 'maximum temperature from temperature_data table corresponding to those years which have at least 2 entries in the table' as select max(temperature), from_unixtime(unix_timestamp(tdate,'MM-dd-yyyy'),'yyyy') as new_date from temperature_data group by from_unixtime(unix_timestamp(tdate,'MM-dd-yyyy'),'yyyy');

hive> show tables;

*hive> select * from temperature_data_vw;*

Output:

```
hive> create view temperature_data_vw(tdate,temperature) comment 'maximum temperature from temperature_data table corresponding to
those years which have at least 2 entries in the table' as select max(temperature), from_unixtime(unix_timestamp(tdate,'MM-dd-yyyy'
),'yyyy') as new_date from temperature_data group by from_unixtime(unix_timestamp(tdate,'MM-dd-yyyy'),'yyyy');
OK
Time taken: 0.879 seconds
hive> show tables;
OK
temperature_data
temperature_data_vw
Time taken: 0.103 seconds, Fetched: 2 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 e
ngine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180808132938_0224da30-9981-4303-9f5d-d9430123bc24
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_1533711610348_0003, Tracking URL = http://localhost:8088/proxy/application_1533711610348_0003/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533711610348_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-08 13:29:59,948 Stage-1 map = 0%,  reduce = 0%
2018-08-08 13:30:21,896 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 10.39 sec
2018-08-08 13:30:40,910 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 18.83 sec
MapReduce Total cumulative CPU time: 18 seconds 830 msec
Ended Job = job_1533711610348_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 18.83 sec   HDFS Read: 9632 HDFS Write: 207 SUCCESS
Total MapReduce CPU Time Spent: 18 seconds 830 msec
OK
23      1990
22      1991
11      1992
16      1993
23      1994
12      1995
Time taken: 63.478 seconds, Fetched: 6 row(s)
hive>
```

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

Solution:

- To perform the task, use the query command on hive shell:
  *hive> insert overwrite local directory '/home/acadgild/HiveExamples/' row format delimited fields terminated by '|' select * from temperature_data_vw;*
- To check the task, if it has been successfully completed, use the query command local shell:
  $ cd HiveExamples
  $ pwd
  $ ll
  $ cat 000000_0

Output:

```
hive> insert overwrite local directory '/home/acadgild/HiveExamples/' row format delimited fields terminated by '|' select * from t
emperature_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 e
ngine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180808133910_40c7a3c0-2c0e-459f-9d4f-a1e78a0efcd9
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_1533711610348_0004, Tracking URL = http://localhost:8088/proxy/application_1533711610348_0004/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533711610348_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-08 13:39:31,550 Stage-1 map = 0%,  reduce = 0%
2018-08-08 13:39:47,779 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 7.54 sec
2018-08-08 13:40:08,062 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 15.18 sec
MapReduce Total cumulative CPU time: 15 seconds 180 msec
Ended Job = job_1533711610348_0004
Moving data to local directory /home/acadgild/HiveExamples
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 15.18 sec   HDFS Read: 9220 HDFS Write: 48 SUCCESS
Total MapReduce CPU Time Spent: 15 seconds 180 msec
OK
Time taken: 59.626 seconds
hive>
```

```
[acadgild@localhost ~]$ cd HiveExamples
[acadgild@localhost HiveExamples]$ pwd
/home/acadgild/HiveExamples
[acadgild@localhost HiveExamples]$ ll
total 4
-rw-r--r--. 1 acadgild acadgild 48 Aug  8 13:40 000000_0
[acadgild@localhost HiveExamples]$ cat 000000_0
23|1990
22|1991
11|1992
16|1993
23|1994
12|1995
[acadgild@localhost HiveExamples]$
```