CASE STUDY

CASE STUDY:

Take or create a database (any) and use it in hadoop and run basic commands on it.

MY DATASET:

The dataset is taken from Kaggle and name is **FINALDATASET.csv** and in Hadoop we are creating its directory as **dataset** .

It has 100 rows

It has 3 columns:

1.ID

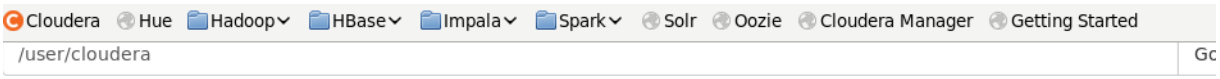
2. QUANTITY

3. ITEM

A PART OF THE DATASET:

```
ID,QUANTITY,ITEM
1808,500,tropical fruit
2552,400,whole milk
2300,50,pip fruit
1187,322,other vegetables
3037,44,whole milk
4941,666,rolls/buns
4501,555,other vegetables
3803,666,pot plants
2762,777,whole milk
4119,888,tropical fruit
1340,999,citrus fruit
2193,133,beef
1997,33,frankfurter
4546,433,chicken
4736,55,butter
1959,777,fruit/vegetable juice
1974,777,packaged fruit/vegetables
2421,332,chocolate
1513,444,specialty bar
1905,500,other vegetables
2810,455,butter milk
2867,800,whole milk
3962,900,tropical fruit
1088,766,tropical fruit
4976,777,bottled water
4056,444,yogurt
3611,111,sausage
1420,44,other vegetables
4286,500,brown bread
```

```
[cloudera@quickstart Desktop]$ hdfs dfs -mkdir dataset
```



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	cloudera	cloudera	0 B	Mon Nov 20 06:58:03 -0800 2023	0	0 B	company
-rw-r--r--	cloudera	cloudera	82 B	Mon Mar 16 22:40:50 -0700 2020	1	128 MB	dante1
drwxr-xr-x	cloudera	cloudera	0 B	Mon Nov 20 12:43:50 -0800 2023	0	0 B	dataset
drwxr-xr-x	cloudera	cloudera	0 B	Mon Nov 20 07:46:11 -0800 2023	0	0 B	grocery_shop
drwxr-xr-x	cloudera	cloudera	0 B	Mon Nov 20 07:57:40 -0800 2023	0	0 B	my_dataset_grocery
-rw-r--r--	cloudera	cloudera	75 B	Mon Mar 16 22:54:15 -0700 2020	1	128 MB	nameone
drwxr-xr-x	cloudera	cloudera	0 B	Sun Nov 19 23:47:30 -0800 2023	0	0 B	office
drwxr-xr-x	cloudera	cloudera	0 B	Mon Nov 20 07:19:35 -0800 2023	0	0 B	shagun

```
[cloudera@quickstart Desktop]$ hdfs dfs -copyFromLocal /home/cloudera/Desktop/FINALDATASET.csv /user/cloudera/dataset;
```

Browse Directory

/user/cloudera/dataset Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	cloudera	cloudera	2.5 KB	Mon Nov 20 12:49:52 -0800 2023	1	128 MB	FINALDATASET.csv

Hadoop, 2017.

COMMANDS ON THE DATASET:

CREATING DATABASE AND THEN TABLE-

```
hive> create database shop;
OK
Time taken: 0.64 seconds
hive> use shop;
OK
Time taken: 0.047 seconds
```

```
hive> create table grocery(
    > ID int,
    > Quantity int,
    > Item string
    > )
    > row format delimited fields terminated by ',' stored as TEXTFILE;
OK
Time taken: 0.604 seconds
```

```
hive> load data inpath '/user/cloudera/dataset/FINALDATASET.csv' OVERWRITE into table grocery;
Loading data to table shop.grocery
chgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehouse/shop.db/grocery/FINA
Table shop.grocery stats: [numFiles=1, numRows=0, totalSize=2555, rawDataSize=0]
OK
Time taken: 1.289 seconds
```

BASIC COMMANDS:

```
hive> select * from grocery where (ID=1808);
OK
1808    500     tropical fruit
Time taken: 1.064 seconds, Fetched: 1 row(s)
hive> select max(Quantity) from grocery;
Query ID = cloudera_20231120130404_559f839c-7d0a-43ec-8fbc-17715984cf05
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 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_1700491316189_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/ap
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1700491316189_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-11-20 13:04:33,166 Stage-1 map = 0%,  reduce = 0%
2023-11-20 13:04:47,705 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.14 sec
2023-11-20 13:05:02,406 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.37 sec
MapReduce Total cumulative CPU time: 4 seconds 370 msec
Ended Job = job_1700491316189_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.37 sec   HDFS Read: 10006 HDFS Write  4 S
Total MapReduce CPU Time Spent: 4 seconds 370 msec
OK
999
Time taken: 57.253 seconds, Fetched: 1 row(s)
```

```
hive> select ID,Item from grocery where(Quantity=500);
OK
1808    tropical fruit
1905    other vegetables
4286    brown bread
1495    root vegetables
Time taken: 0.138 seconds, Fetched: 4 row(s)
```

```
hive> select sum(Quantity) from grocery where (Quantity<500);
Query ID = cloudera_20231120130707_efb37011-81df-43fd-93ff-278f30dea656
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 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_1700491316189_0002, Tracking URL = http://quickstart.clou
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1700491316189_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers
2023-11-20 13:07:35,324 Stage-1 map = 0%,  reduce = 0%
2023-11-20 13:07:48,103 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.5
2023-11-20 13:08:01,302 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4
MapReduce Total cumulative CPU time: 4 seconds 710 msec
Ended Job = job_1700491316189_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.71 sec   HDFS Read: 102
Total MapReduce CPU Time Spent: 4 seconds 710 msec
OK
20440
Time taken: 39.192 seconds, Fetched: 1 row(s)
```

```
hive> select *from grocery where(2000<ID<3000);
OK
1808    500     tropical fruit
2552    400     whole milk
2300    50      pip fruit
1187    322     other vegetables
3037    44      whole milk
4941    666     rolls/buns
4501    555     other vegetables
3803    666     pot plants
2762    777     whole milk
4119    888     tropical fruit
1340    999     citrus fruit
2193    133     beef
1997    33      frankfurter
4546    433     chicken
4736    55      butter
1959    777     fruit/vegetable juice
1974    777     packaged fruit/vegetables
2421    332     chocolate
1513    444     specialty bar
1905    500     other vegetables
2810    455     butter milk
2867    800     whole milk
3962    900     tropical fruit
1088    766     tropical fruit
4976    777     bottled water
4056    444     yogurt
3611    111     sausage
1420    44      other vegetables
4286    500     brown bread
4918    66      yogurt
4783    700     hamburger meat
3709    444     root vegetables
4289    122     pork
1559    455     beef
2900    245     pastry
1905    567     fruit/vegetable juice
3527    766     canned beer

hive> Select ID,ITEM from grocery where (Quantity=500);
OK
1808    tropical fruit
1905    other vegetables
4286    brown bread
1495    root vegetables
Time taken: 0.071 seconds, Fetched: 4 row(s)
```

```
hive> Select ID,ITEM from grocery where (Quantity>500);
OK
4941    rolls/buns
4501    other vegetables
3803    pot plants
2762    whole milk
4119    tropical fruit
1340    citrus fruit
1959    fruit/vegetable juice
1974    packaged fruit/vegetables
2867    whole milk
3962    tropical fruit
1088    tropical fruit
4976    bottled water
4783    hamburger meat
1905    fruit/vegetable juice
3527    canned beer
1863    tropical fruit
4708    sausage
2874    sausage
4177    frankfurter
1663    rolls/buns
2632    whole milk
1377    curd cheese
4162    red/blush wine
2270    sausage
4829    tropical fruit
3811    red/blush wine
4766    whole milk
2436    frankfurter
3860    whole milk
4875    frozen potato products
4152    fruit/vegetable juice
4155    citrus fruit
4010    pork
4389    detergent
3746    grapes
2560    sausage
1503    chicken
```

```
Time taken: 0.174 seconds, Fetched: 39 row(s)
hive> Select ID,ITEM from grocery where (Quantity<500);
OK
2552    whole milk
2300    pip fruit
1187    other vegetables
3037    whole milk
2193    beef
1997    frankfurter
4546    chicken
4736    butter
2421    chocolate
1513    specialty bar
2810    butter milk
4056    yogurt
3611    sausage
1420    other vegetables
4918    yogurt
3709    root vegetables
4289    pork
1559    beef
2900    pastry
3558    citrus fruit
3128    sausage
3841    berries
3903    canned beer
2658    butter milk
4272    coffee
1120    pastry
2676    rolls/buns
1697    misc. beverages
2507    root vegetables
4620    sausage
3365    canned beer
2978    ham
2910    turkey
1061    whole milk
3276    whole milk
```

```
hive> Select sum(Quantity) from Grocery where (Item='tropical fruit);
FAILED: ParseException line 1:62 character '<EOF>' not supported here
hive> Select sum(Quantity) from Grocery where (Item='tropical fruit');
Query ID = cloudera_20231120131212_130c6367-13a2-41a1-80cb-957885981ab8
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 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_1700491316189_0003, Tracking URL = http://quickstart.cloudera:8088/pro
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1700491316189_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-11-20 13:12:45,527 Stage-1 map = 0%,  reduce = 0%
2023-11-20 13:12:56,865 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.34 sec
2023-11-20 13:13:10,685 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.68 sec
MapReduce Total cumulative CPU time: 4 seconds 680 msec
Ended Job = job_1700491316189_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.68 sec   HDFS Read: 11158 HDFS Write
Total MapReduce CPU Time Spent: 4 seconds 680 msec
OK
4509
Time taken: 36.081 seconds, Fetched: 1 row(s)
```

CONCLUSION:

At the end we were able to import entire dataset to Hadoop and hdfs .

Also we were able to take out some important insights about the large dataset very quickly.

This way Hadoop is helpful in big data analytics and it seems an easy task while working on Hadoop, hive, sqoop, hdfs etc