hduser@rootuser:~$ hive

Logging initialized using configuration in jar:[file:/usr/local/hive/lib/hive-common-1.2.1.jar!/hive-log4j.properties](file:///usr/local/hive/lib/hive-common-1.2.1.jar!/hive-log4j.properties)

There are two ways to know the current database. One temporary in cli and second one is persistently.

1) in CLI just enter this command: set hive.cli.print.current.db=true;

2) In hive-site.xml paste this code:

<property>

<name>hive.cli.print.current.db</name>

<value>true</value>

</property>

hive (default)> show databases;

OK

database\_name

default

Time taken: 0.611 seconds, Fetched: 1 row(s)

hive (default)> create database niit;

OK

Time taken: 0.154 seconds

hive (default)> use niit;

OK

Time taken: 0.016 seconds

hive (niit)> create table customer(custno INT, firstname STRING, lastname STRING, age INT, profession STRING)

> row format delimited

> fields terminated by ','

> stored as textfile;

OK

Time taken: 0.182 seconds

hive (niit)> show tables;

OK

tab\_name

customer

Time taken: 0.034 seconds, Fetched: 1 row(s)

Describing metadata or schema of the table

hive (niit)> describe customer;

OK

col\_name data\_type comment

custno int

firstname string

lastname string

age int

profession string

Time taken: 0.269 seconds, Fetched: 5 row(s)

Describing detailed metadata or schema of the table

describe extended txnrecords;

or

describe formatted txnrecords;

hive (niit)> describe formatted customer;

OK

col\_name data\_type comment

# col\_name data\_type comment

custno int

firstname string

lastname string

age int

profession string

# Detailed Table Information

Database: niit

Owner: hduser

CreateTime: Fri Dec 22 18:29:15 IST 2017

LastAccessTime: UNKNOWN

Protect Mode: None

Retention: 0

Location: hdfs://localhost:54310/user/hive/warehouse/niit.db/customer

Table Type: MANAGED\_TABLE

Table Parameters:

transient\_lastDdlTime 1513947555

# Storage Information

SerDe Library: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe

InputFormat: org.apache.hadoop.mapred.TextInputFormat

OutputFormat: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat

Compressed: No

Num Buckets: -1

Bucket Columns: []

Sort Columns: []

Storage Desc Params:

field.delim ,

serialization.format ,

Time taken: 0.062 seconds, Fetched: 31 row(s)

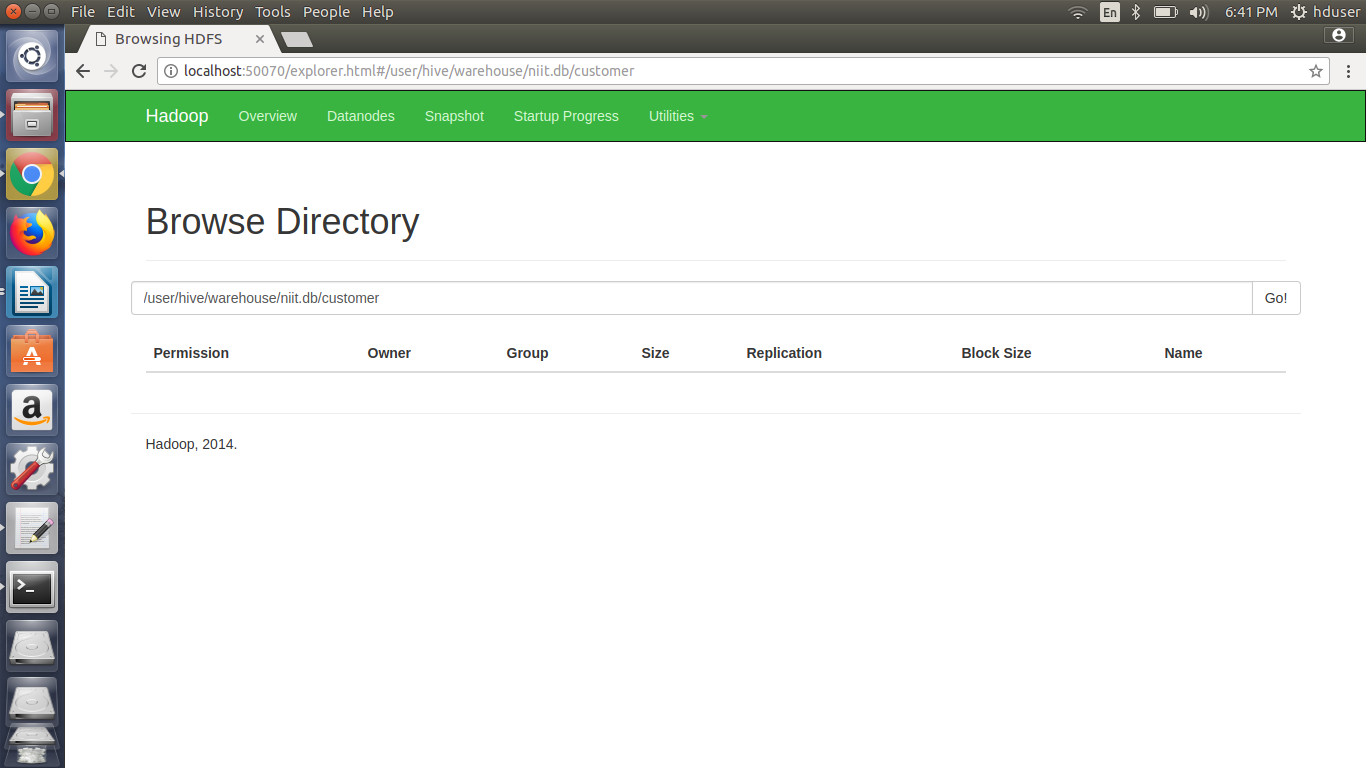
hive (niit)> select \* from customer;

OK

customer.custno customer.firstname customer.lastname customer.age customer.profession

Time taken: 0.381 seconds

0 records because customer folder does not have any data ---- therefore you can’t fetch the data



First we need to put our data file into hive customer table folder.

hive (niit)> quit;

hduser@rootuser:~$ hadoop fs -put /home/hduser/InputData/custs.txt /user/hive/warehouse/niit.db/customer;

17/12/22 18:48:02 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hadoop fs -ls /user/hive/warehouse/niit.db/customer;

17/12/22 19:07:15 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 1 items

-rw-r--r-- 1 hduser supergroup 391894 2017-12-22 18:48 /user/hive/warehouse/niit.db/customer/custs.txt

Now when we run the select query we can see and read the data in a structured way.

How? The data file must be present inside the table folder …. Only then it will work.

hive (niit)> select \* from customer;

OK

customer.custno customer.firstname customer.lastname customer.age customer.profession

4000001 Kristina Chung 55 Pilot

4000002 Paige Chen 74 Teacher

4000003 Sherri Melton 34 Firefighter

4000004 Gretchen Hill 66 Computer hardware engineer

4000005 Karen Puckett 74 Lawyer

4000006 Patrick Song 42 Veterinarian

......

......

Counting number of Records in Table:

hive (niit)> select count(\*) from customer;

Query ID = hduser\_20171222192155\_52666e0c-4de3-4bb2-82e9-f237e4d40a4c

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\_1513946090631\_0001, Tracking URL = http://rootuser:8088/proxy/application\_1513946090631\_0001/

Kill Command = /usr/local/hadoop/bin/hadoop job -kill job\_1513946090631\_0001

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1

2017-12-22 19:22:03,372 Stage-1 map = 0%, reduce = 0%

2017-12-22 19:22:06,613 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.1 sec

2017-12-22 19:22:12,844 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.01 sec

MapReduce Total cumulative CPU time: 3 seconds 10 msec

Ended Job = job\_1513946090631\_0001

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.01 sec HDFS Read: 398695 HDFS Write: 5 SUCCESS

Total MapReduce CPU Time Spent: 3 seconds 10 msec

OK

\_c0

9999

Time taken: 18.001 seconds, Fetched: 1 row(s)

hduser@rootuser:~$ nano custs\_add;

4010000,John,Mathews,50,Lawyer

hduser@rootuser:~$ hadoop fs -put /home/hduser/custs\_add /user/hive/warehouse/niit.db/customer;

17/12/22 19:32:08 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Now there will be 2 files in the customer folder present in hive

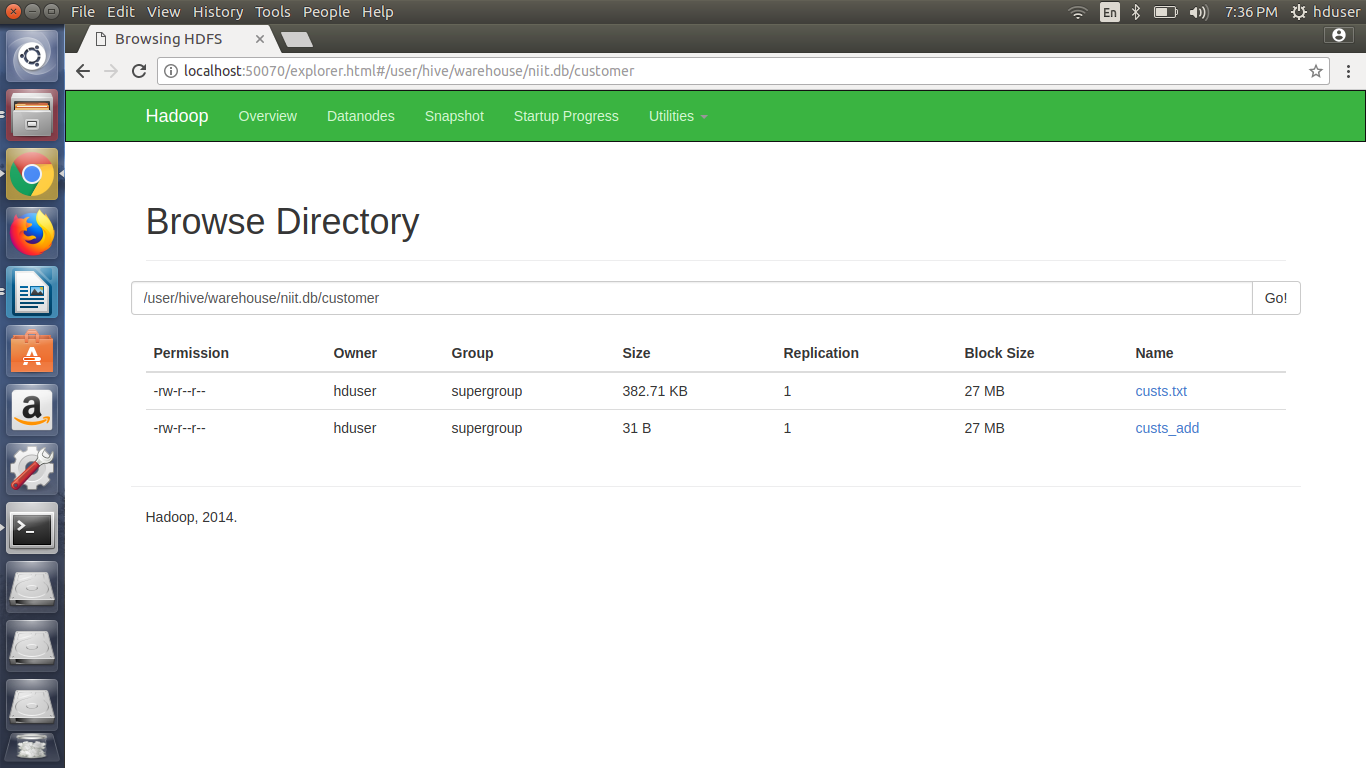
hduser@rootuser:~$ hadoop fs -ls /user/hive/warehouse/niit.db/customer;

17/12/22 19:32:23 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 2 items

-rw-r--r-- 1 hduser supergroup 391894 2017-12-22 18:48 /user/hive/warehouse/niit.db/customer/custs.txt

-rw-r--r-- 1 hduser supergroup 31 2017-12-22 19:32 /user/hive/warehouse/niit.db/customer/custs\_add



Now in customer table which files data gets selected?

Ans: Both data file wil be used. So total number of records fetched=10000. custs.txt(99999)+custs\_add(1)

hive (niit)> select count(\*) from customer;

Query ID = hduser\_20171222193713\_960ff9f6-55ba-4202-9daf-c05cfe2076c2

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\_1513946090631\_0002, Tracking URL = http://rootuser:8088/proxy/application\_1513946090631\_0002/

Kill Command = /usr/local/hadoop/bin/hadoop job -kill job\_1513946090631\_0002

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1

2017-12-22 19:37:18,549 Stage-1 map = 0%, reduce = 0%

2017-12-22 19:37:22,702 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.18 sec

2017-12-22 19:37:25,805 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 2.43 sec

MapReduce Total cumulative CPU time: 2 seconds 430 msec

Ended Job = job\_1513946090631\_0002

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 2.43 sec HDFS Read: 398812 HDFS Write: 6 SUCCESS

Total MapReduce CPU Time Spent: 2 seconds 430 msec

OK

\_c0

10000

Time taken: 13.254 seconds, Fetched: 1 row(s)

Find Count of customer for each profession from customer table.

hive (niit)> select profession, count(\*) AS Total\_Records from customer group by profession;

Query ID = hduser\_20171222194253\_0a3c80bf-bf03-4584-bcdd-030c29f7701a

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\_1513946090631\_0003, Tracking URL = http://rootuser:8088/proxy/application\_1513946090631\_0003/

Kill Command = /usr/local/hadoop/bin/hadoop job -kill job\_1513946090631\_0003

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1

2017-12-22 19:42:57,073 Stage-1 map = 0%, reduce = 0%

2017-12-22 19:43:01,221 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.27 sec

2017-12-22 19:43:06,357 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 2.57 sec

MapReduce Total cumulative CPU time: 2 seconds 570 msec

Ended Job = job\_1513946090631\_0003

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 2.57 sec HDFS Read: 399387 HDFS Write: 885 SUCCESS

Total MapReduce CPU Time Spent: 2 seconds 570 msec

OK

profession total\_records

Accountant 199

Actor 202

Agricultural and food scientist 195

Architect 203

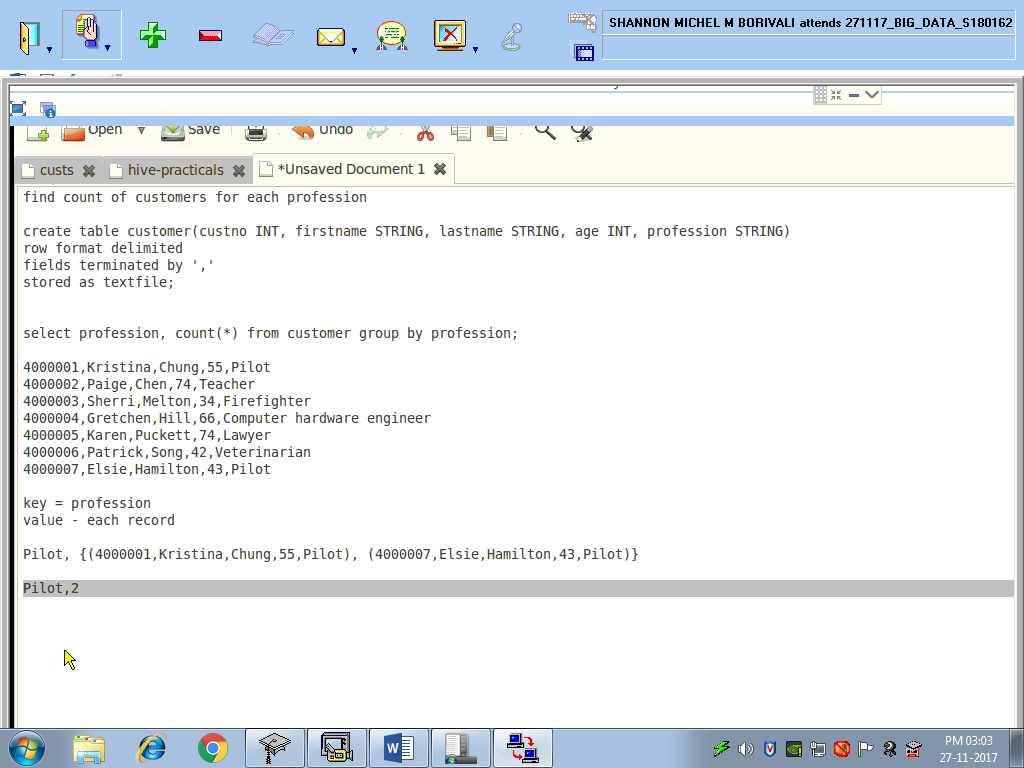
Artist 175

Athlete 196

.....

......

Data is coming in sorted form because of the internal mapred job.



Order by desc ----- One more mapred job

**hive (niit)> select profession, count(\*) AS Total\_Records from customer group by profession order by Total\_Records desc;**

Query ID = hduser\_20171222194956\_8ba61b1f-b671-4743-b062-bcc862d94910

**Total jobs = 2**

Launching Job 1 out of 2

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\_1513946090631\_0004, Tracking URL = http://rootuser:8088/proxy/application\_1513946090631\_0004/

Kill Command = /usr/local/hadoop/bin/hadoop job -kill job\_1513946090631\_0004

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1

2017-12-22 19:49:59,962 Stage-1 map = 0%, reduce = 0%

2017-12-22 19:50:04,081 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.51 sec

2017-12-22 19:50:09,221 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 2.8 sec

MapReduce Total cumulative CPU time: 2 seconds 800 msec

Ended Job = job\_1513946090631\_0004

Launching Job 2 out of 2

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\_1513946090631\_0005, Tracking URL = http://rootuser:8088/proxy/application\_1513946090631\_0005/

Kill Command = /usr/local/hadoop/bin/hadoop job -kill job\_1513946090631\_0005

Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

2017-12-22 19:50:17,772 Stage-2 map = 0%, reduce = 0%

2017-12-22 19:50:21,916 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 0.67 sec

2017-12-22 19:50:26,061 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 1.92 sec

MapReduce Total cumulative CPU time: 1 seconds 920 msec

Ended Job = job\_1513946090631\_0005

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 2.8 sec HDFS Read: 398815 HDFS Write: 1746 SUCCESS

Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 1.92 sec HDFS Read: 6284 HDFS Write: 885 SUCCESS

Total MapReduce CPU Time Spent: 4 seconds 720 msec

OK

profession total\_records

Politician 228

Computer support specialist 222

Photographer 222

Loan officer 221

Librarian 218

Firefighter 217

Computer software engineer 216

.....

.....

**Now the data is sorted on the 2nd column.**

**Compare the difference using chain mapreduce and sql.**

**Mapred --- lines of code**

**SQL – 3 words**

**to save the file on hdfs (It will create the specified directory on HDFS if not exists.**

**-------------------------**

**hive (niit)> INSERT OVERWRITE DIRECTORY '/niit/custcount' row format delimited fields terminated by ',' select profession, count(\*) from customer group by profession;**

**Query ID = hduser\_20171222195653\_d9d51f1f-f3ba-42db-950e-a3a85e9f5455**

**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\_1513946090631\_0006, Tracking URL = http://rootuser:8088/proxy/application\_1513946090631\_0006/**

**Kill Command = /usr/local/hadoop/bin/hadoop job -kill job\_1513946090631\_0006**

**Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1**

**2017-12-22 19:56:57,124 Stage-1 map = 0%, reduce = 0%**

**2017-12-22 19:57:01,237 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.21 sec**

**2017-12-22 19:57:06,392 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 2.53 sec**

**MapReduce Total cumulative CPU time: 2 seconds 530 msec**

**Ended Job = job\_1513946090631\_0006**

**Moving data to: /niit/custcount**

**MapReduce Jobs Launched:**

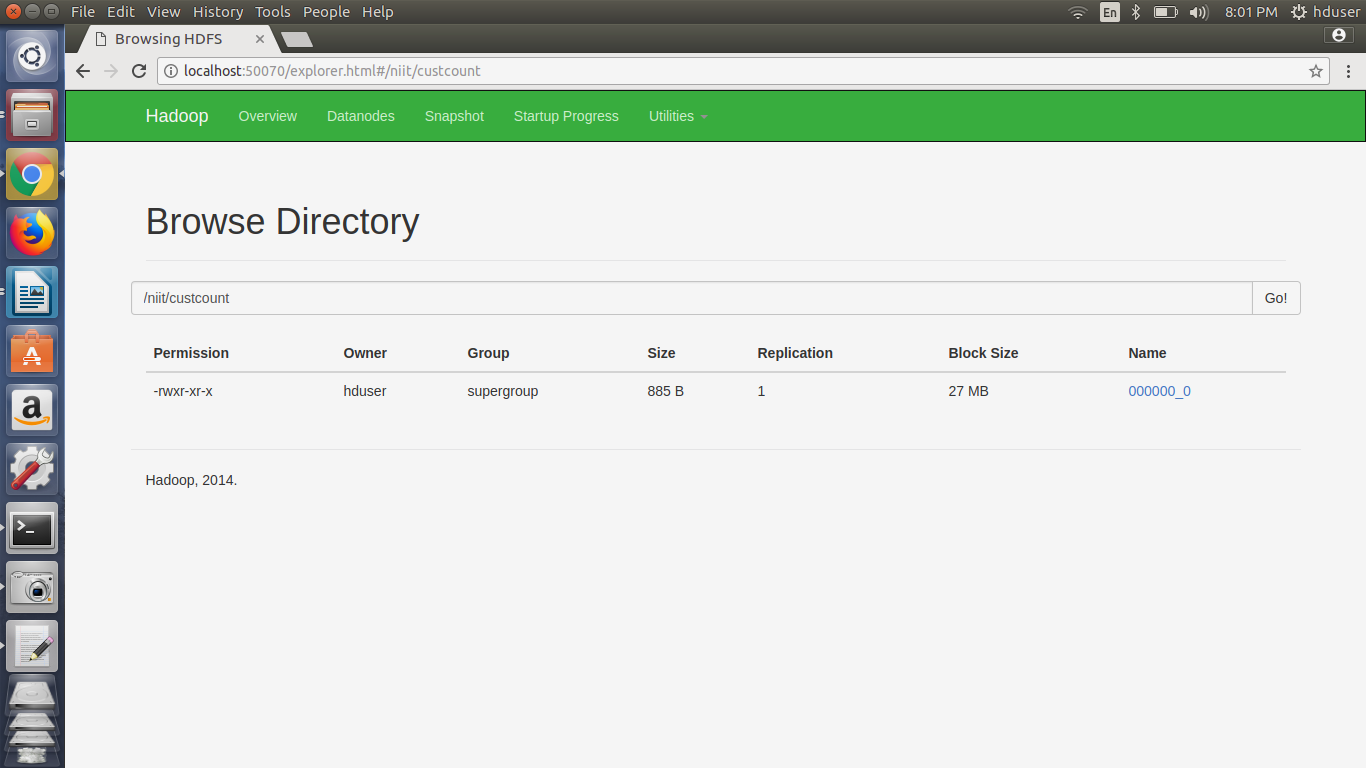
**Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 2.53 sec HDFS Read: 399138 HDFS Write: 885 SUCCESS**

**Total MapReduce CPU Time Spent: 2 seconds 530 msec**

**OK**

**profession \_c1**

**Time taken: 13.992 seconds**

****

**Now How to load data directly from local file system to hive on hdfs?**

**Load the data into the table (from local file system)**

**-----------------------------------------------------**

**LOAD DATA LOCAL INPATH '/home/hduser/txns1.txt' OVERWRITE INTO TABLE txnrecords;**

**hive (niit)> create table txnrecords(txnno INT, txndate STRING, custno INT, amount DOUBLE, category STRING, product STRING, city STRING, state STRING,**

**spendby STRING)**

**> row format delimited**

**> fields terminated by ','**

**> stored as textfile;**

**OK**

**Time taken: 0.146 seconds**

**hive (niit)> LOAD DATA LOCAL INPATH '/home/hduser/txns1.txt' OVERWRITE INTO TABLE txnrecords;**

**Loading data to table niit.txnrecords**

**Table niit.txnrecords stats: [numFiles=1, numRows=0, totalSize=4418144, rawDataSize=0]**

**OK**

**Time taken: 0.175 seconds**

**hive (niit)> select count(\*) from txnrecords;**

Query ID = hduser\_20171222201001\_b4153f52-8583-421d-b53a-c00f424725f3

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\_1513946090631\_0007, Tracking URL = http://rootuser:8088/proxy/application\_1513946090631\_0007/

Kill Command = /usr/local/hadoop/bin/hadoop job -kill job\_1513946090631\_0007

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1

2017-12-22 20:10:05,034 Stage-1 map = 0%, reduce = 0%

2017-12-22 20:10:09,136 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.29 sec

2017-12-22 20:10:13,254 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 2.83 sec

MapReduce Total cumulative CPU time: 2 seconds 830 msec

Ended Job = job\_1513946090631\_0007

MapReduce Jobs Launched:

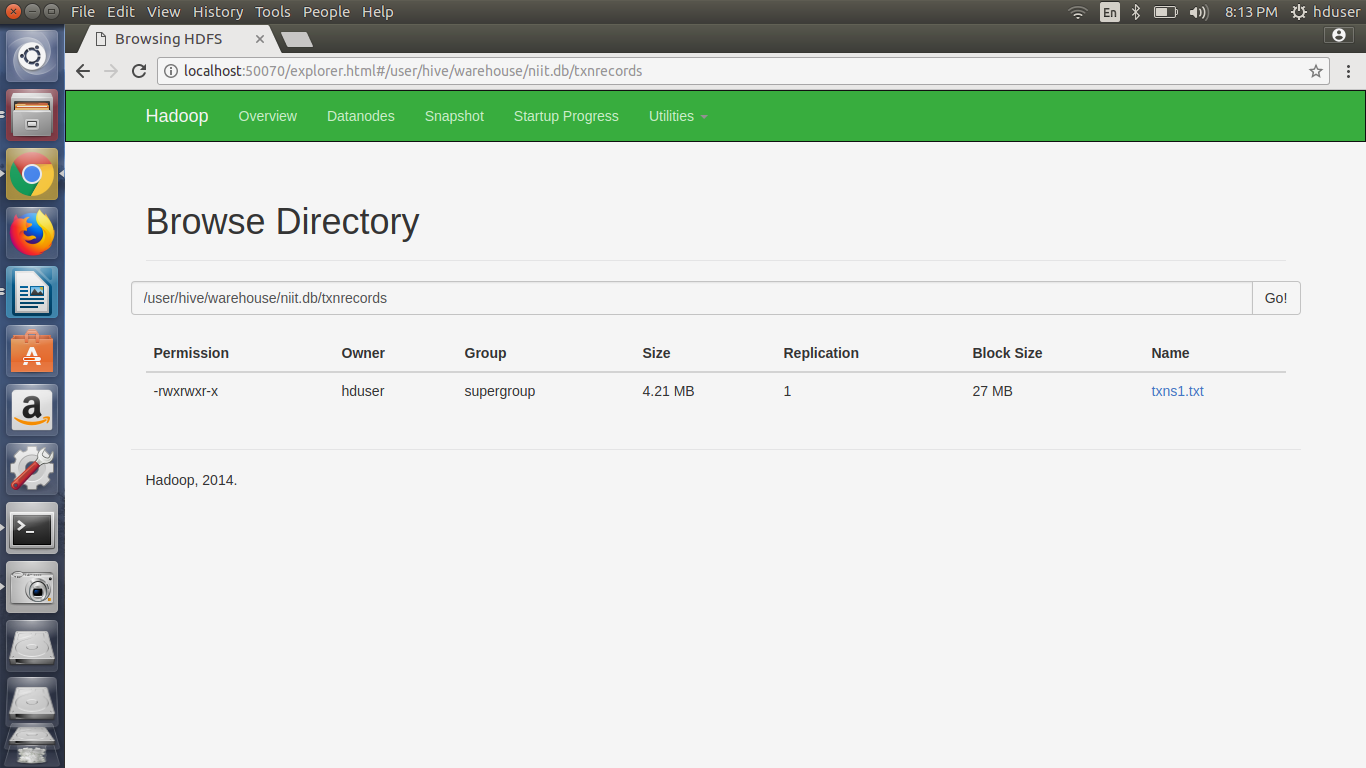
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 2.83 sec HDFS Read: 4425660 HDFS Write: 6 SUCCESS

Total MapReduce CPU Time Spent: 2 seconds 830 msec

OK

**\_c0**

**50000**

****

**hive (niit)> select \* from txnrecords;**

**OK**

**txnrecords.txnno txnrecords.txndate txnrecords.custno txnrecords.amount txnrecords.category txnrecords.product txnrecords.city txnrecords.state txnrecords.spendby**

**0 06-26-2011 4007024 40.33 Exercise & Fitness Cardio Machine Accessories Clarksville Tennessee credit**

**1 05-26-2011 4006742 198.44 Exercise & Fitness Weightlifting Gloves Long Beach California credit**

**2 06-01-2011 4009775 5.58 Exercise & Fitness Weightlifting Machine Accessories Anaheim California credit**

**3 06-05-2011 4002199 198.19 Gymnastics Gymnastics Rings Milwaukee Wisconsin credit**

**.......**

**.......**

**Time taken: 0.052 seconds, Fetched: 50000 row(s)**