PRACTICAL 7 A

Querying, Sorting, Aggregating data using HiveQL

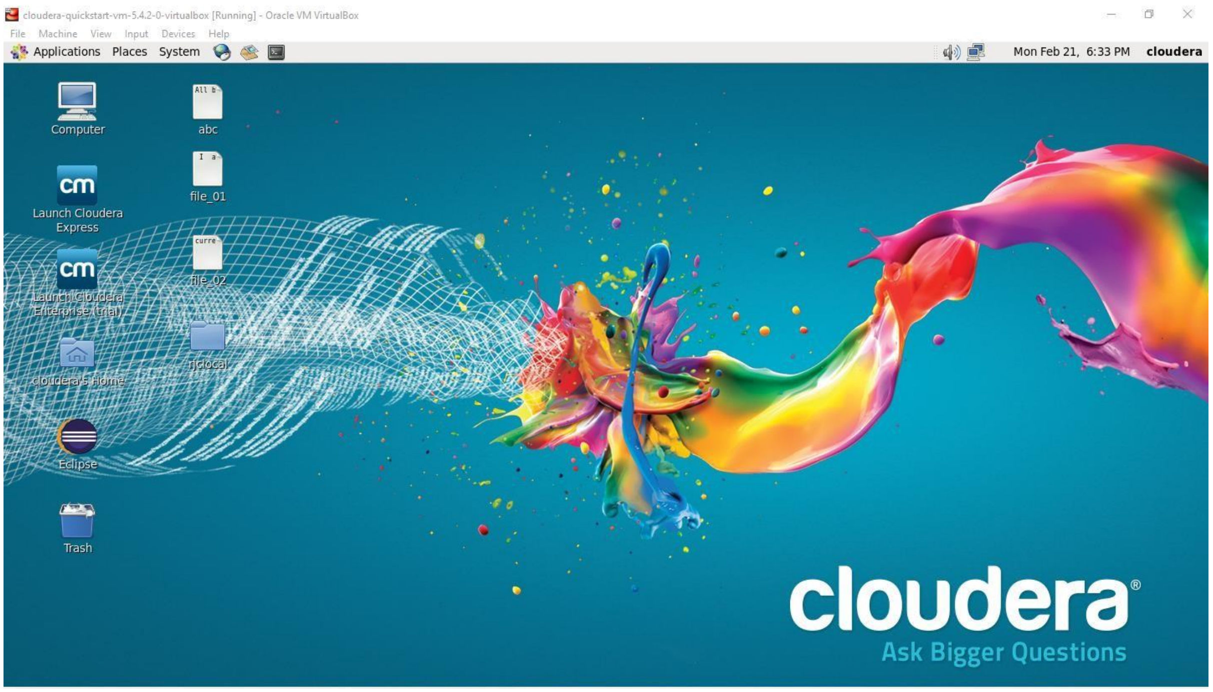
What is HIVE:

Hive is a data warehouse system which is used to analyse structured data. It is built on the top of Hadoop. It was developed by Facebook.

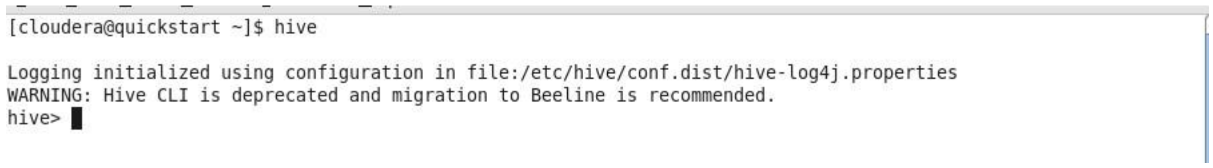
Hive provides the functionality of reading, writing, and managing large datasets residing in distributed storage. It runs SQL like queries called HQL (Hive query language) which gets internally converted to MapReduce jobs.

Using Hive, we can skip the requirement of the traditional approach of writing complex MapReduce programs. Hive supports Data Definition Language (DDL), Data Manipulation Language (DML), and User Defined Functions (UDF).

<u>Steps: Querying, Sorting, Aggregating data using HiveQL</u> 1. Open the cloudera.

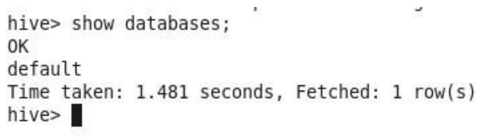

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Mon Feb 21, 6:33 PM cloudera
Computer
abc
file_01
Launch Cloudera Express
file_02
Launch Cloudera Enterprise (trial)
cloudera's Home
Eclipse
Trash
cloudera
Ask Bigger Questions

2. Open the terminal, Now we use hive command to enter the hive shell prompt and in hive shell we could execute all of the hive commands.



```
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> 
```

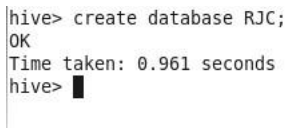
3. Now we will see the databases which are already existing using below command. Show databases;



```
hive> show databases;
OK
default
Time taken: 1.481 seconds, Fetched: 1 row(s)
hive> 
```

4. Creating a database name 'RJC' using below command.

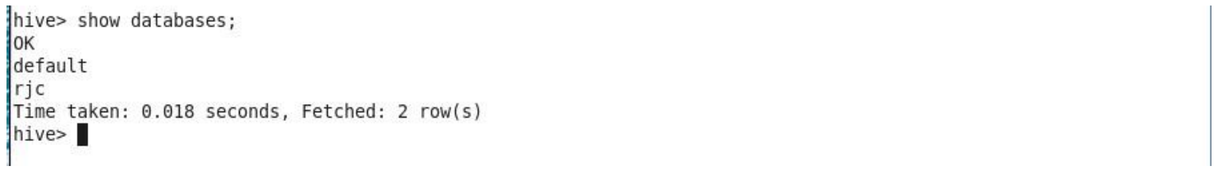
Create database RJC;



```
hive> create database RJC;
OK
Time taken: 0.961 seconds
hive> 
```

5. So if we want to see whether this RJC database is created or not we will use below command,

show databases;



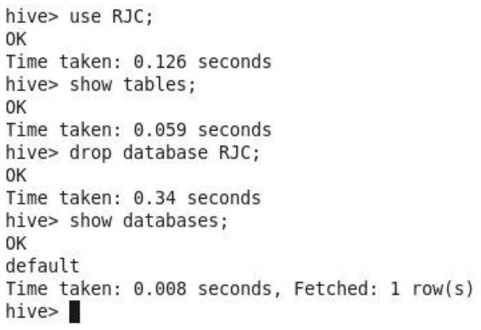
```
hive> show databases;
OK
default
rjc
Time taken: 0.018 seconds, Fetched: 2 row(s)
hive> 
```

6. Now we want to check whether we have any tables inside this rjc database or not. So first we will move to this databse rjc using below command,

Use rjc;

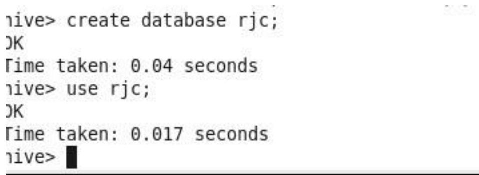
Now we have moved inside this rjc database. Now we will check out which are the tables available using below command, show tables;

No will drop this 'rjc' database using below command drop database rjc; and to check whether the data base dropped or not using below command, show databases;



```
hive> use RJC;
OK
Time taken: 0.126 seconds
hive> show tables;
OK
Time taken: 0.059 seconds
hive> drop database RJC;
OK
Time taken: 0.34 seconds
hive> show databases;
OK
default
Time taken: 0.008 seconds, Fetched: 1 row(s)
hive> 
```

7. Creating a database “rjc”. create database rjc;



```
ıive> create database rjc;
ƆK
Time taken: 0.04 seconds
ıive> use rjc;
ƆK
Time taken: 0.017 seconds
ıive> 
```

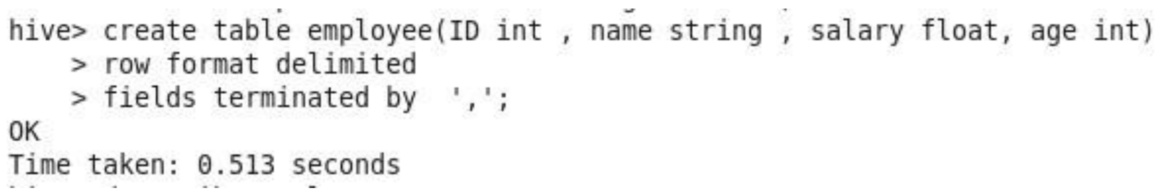
8. Now we will create a table inside this rjc database named as employee using below command,

create table employee(ID int, name string, salary float, age int)

After this we will not put semicolon , When we will be loading the data from some existing csv file or maybe some other text files so we have to mention that how that data has to be loaded here. We are simply creating the schema of the table with some certain fields or attributes along with their datatypes and then mention

row format delimited

fields terminated by ‘,’;



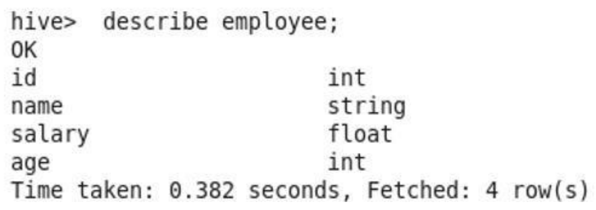
```
hive> create table employee(ID int , name string , salary float, age int)
    > row format delimited
    > fields terminated by  ',';
OK
Time taken: 0.513 seconds
```

row format delimited means , every record is present in one row andfields terminated by ‘,’ and fields are terminated by comma. So as soon as it encountersone comma so that means that one is the value of some field and after comma it isencountering abc so that abc is the value of some another field. By default it is a “tab” character that means fields are separated by “tab”.

9. So now we will see the schema of the table using below command,

describe employee;

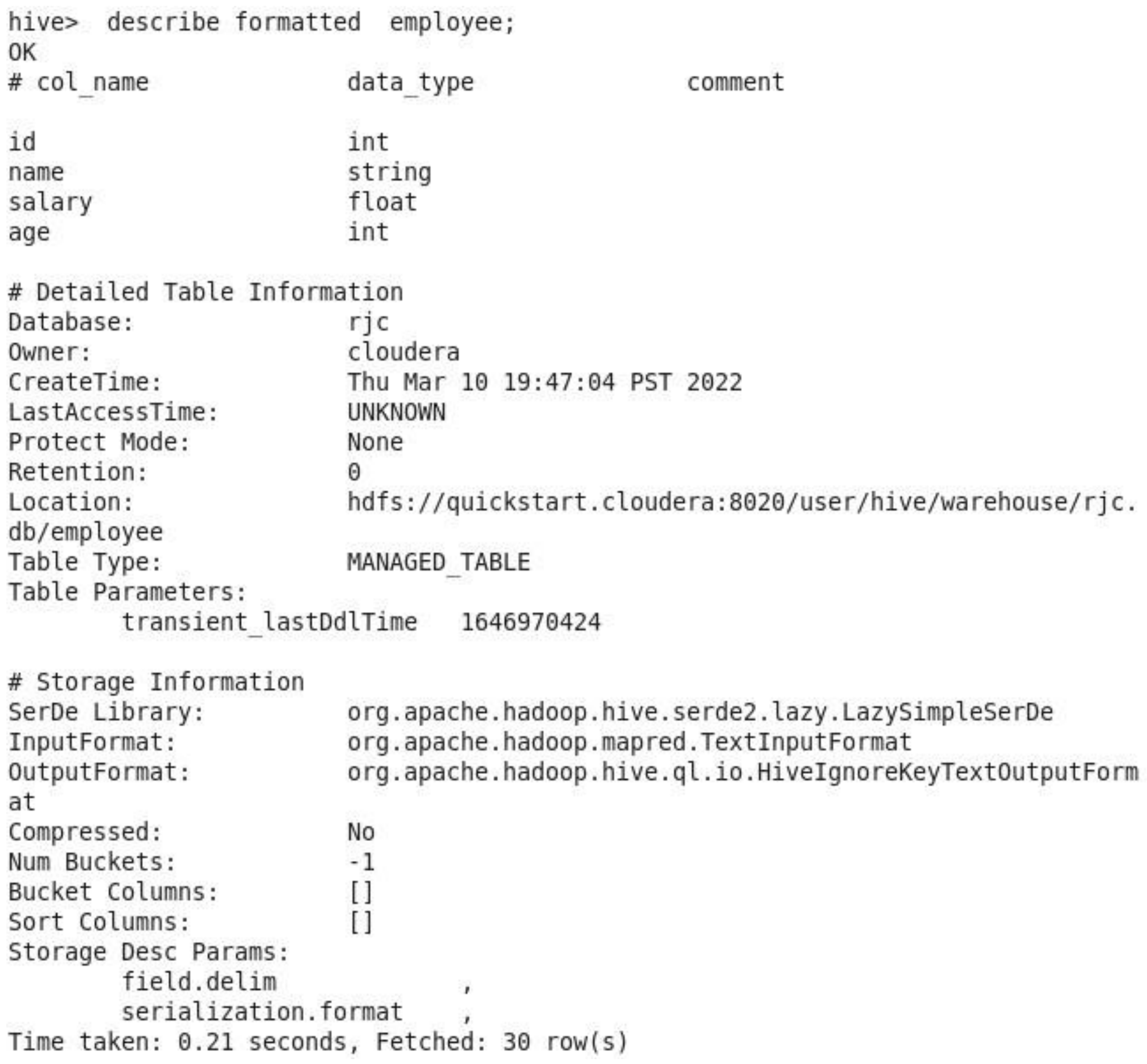
It will give different fields of table employee along with their respective datatypes.



```
hive>  describe employee;
OK
id                      int
name                    string
salary                  float
age                     int
Time taken: 0.382 seconds, Fetched: 4 row(s)
```

10. By default the internal table would store in the warehouse directory of hive. Whereas the external tables are available in the hdfs. And if we drop the internal table so then the table data and the metadata associated with that table will be deleted from the hdfs. Whereas when we drop the external table then only the metadata associated with that table will be deleted whereas the table data will be untouched by hive as it would be residing in the hdfs and it would be outside the warehouse directory of the hive.

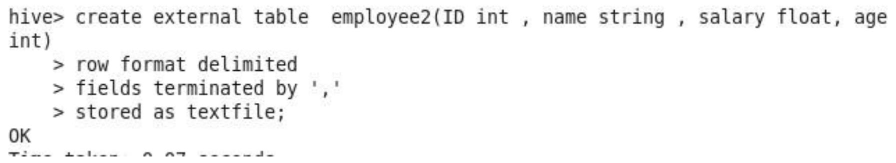
So Now we will check how the table which we will be created is internal table or external table using below command,



```
hive>  describe formatted  employee;
OK
# col_name              data_type               comment

id                      int
name                    string
salary                  float
age                     int

# Detailed Table Information
Database:               rjc
Owner:                  cloudera
CreateTime:             Thu Mar 10 19:47:04 PST 2022
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               hdfs://quickstart.cloudera:8020/user/hive/warehouse/rjc.
db/employee
Table Type:             MANAGED_TABLE
Table Parameters:
        transient_lastDdlTime   1646970424

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputForm
at
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.21 seconds, Fetched: 30 row(s)
```

By default hive createsInternal table or Managed Table.

11. Now we will create the external table using below command,



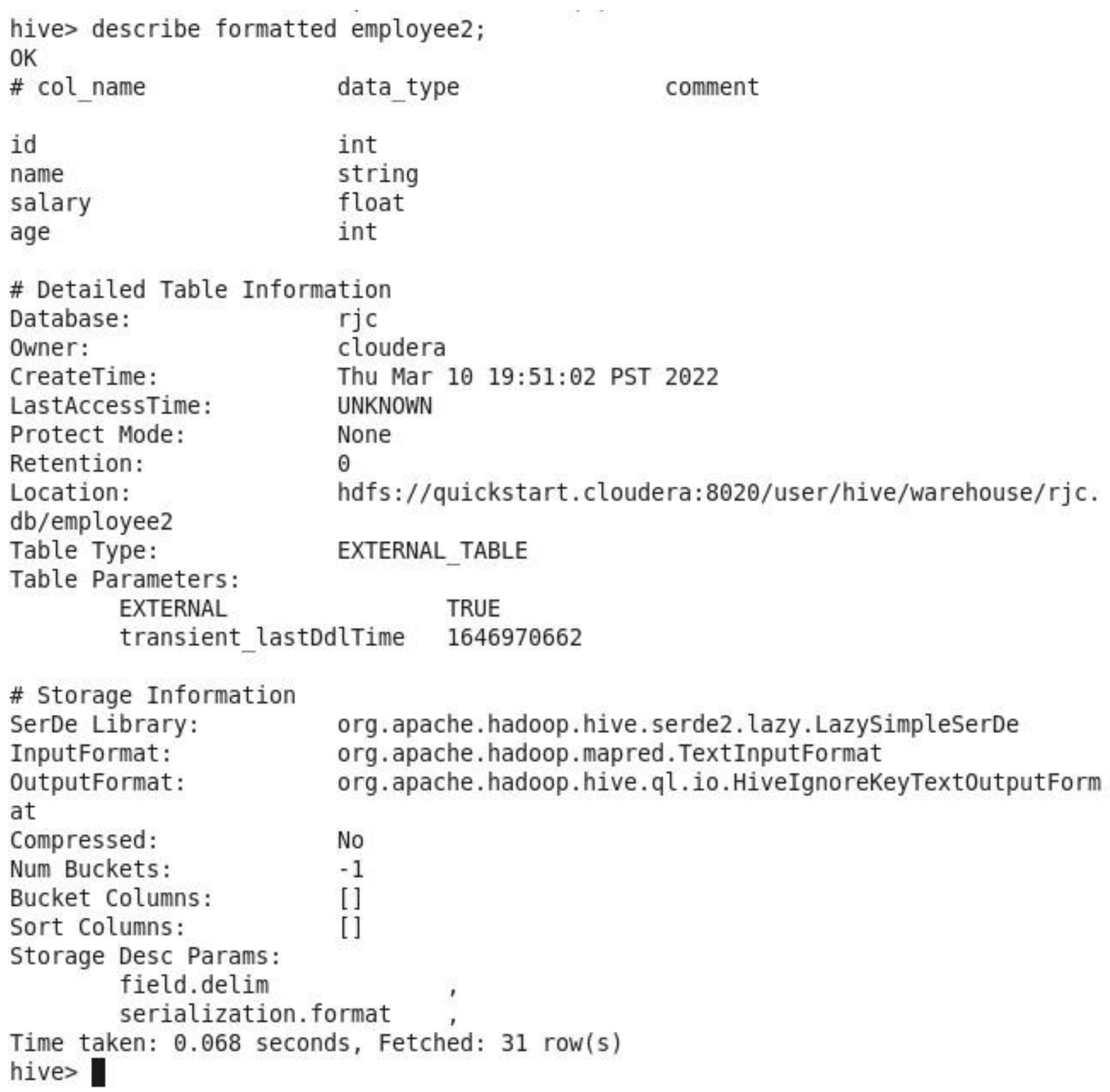
```
hive> create external table  employee2(ID int , name string , salary float, age
int)
    > row format delimited
    > fields terminated by ','
    > stored as textfile;
OK
```

12. Checking the schema of the table using below command,

create external table emloyee2 (ID int, name string, salary float, age int)
➢ row format delimited

➢ **fields terminated by ‘,’**
➢ **stored as textfile;**

13. So Now we will check how the table which we will be created is internal table or external table using below command,



```
hive> describe formatted employee2;
OK
# col_name              data_type               comment

id                      int
name                    string
salary                  float
age                     int

# Detailed Table Information
Database:               rjc
Owner:                  cloudera
CreateTime:             Thu Mar 10 19:51:02 PST 2022
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               hdfs://quickstart.cloudera:8020/user/hive/warehouse/rjc.
db/employee2
Table Type:             EXTERNAL_TABLE
Table Parameters:
        EXTERNAL                TRUE
        transient_lastDdlTime   1646970662

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputForm
at
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.068 seconds, Fetched: 31 row(s)
hive> █
```

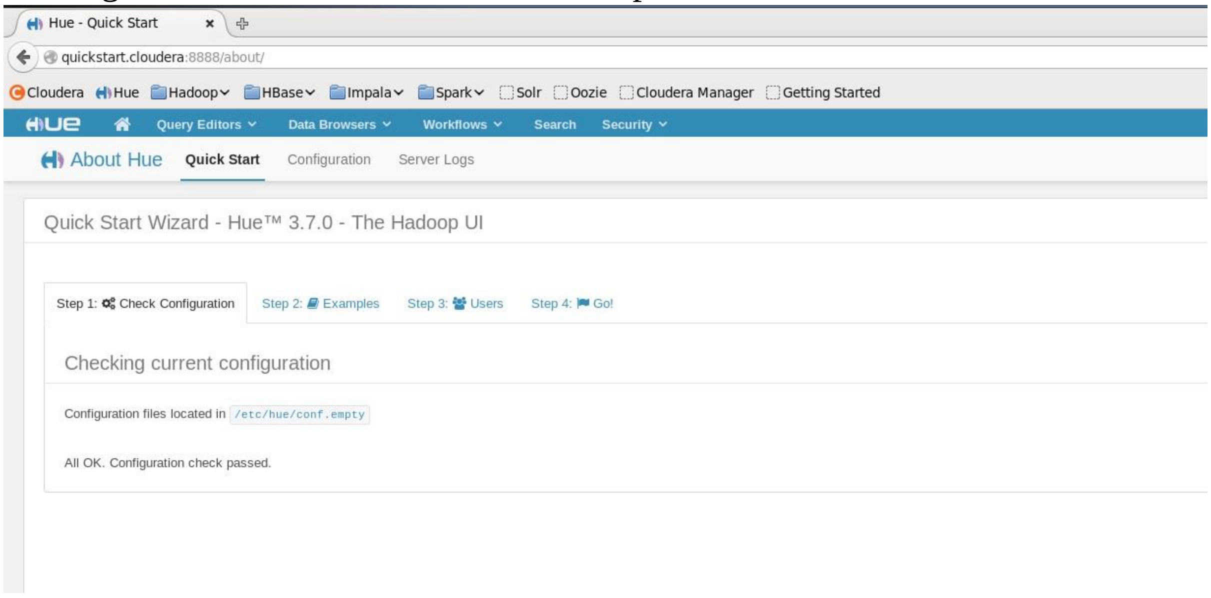
14. So whatever we have done in terminal the same thing we can see in the browser as well.

Open the browser → click on Hue

Then click on Query Editor → select Hive

Write the query in query editor. Here we are showing the databases by using below command, show databases;

It will give the list of databases which are present.


Hue - Quick Start
quickstart.cloudera:8888/about/
Cloudera Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started
Query Editors Data Browsers Workflows Search Security
About Hue Quick Start Configuration Server Logs

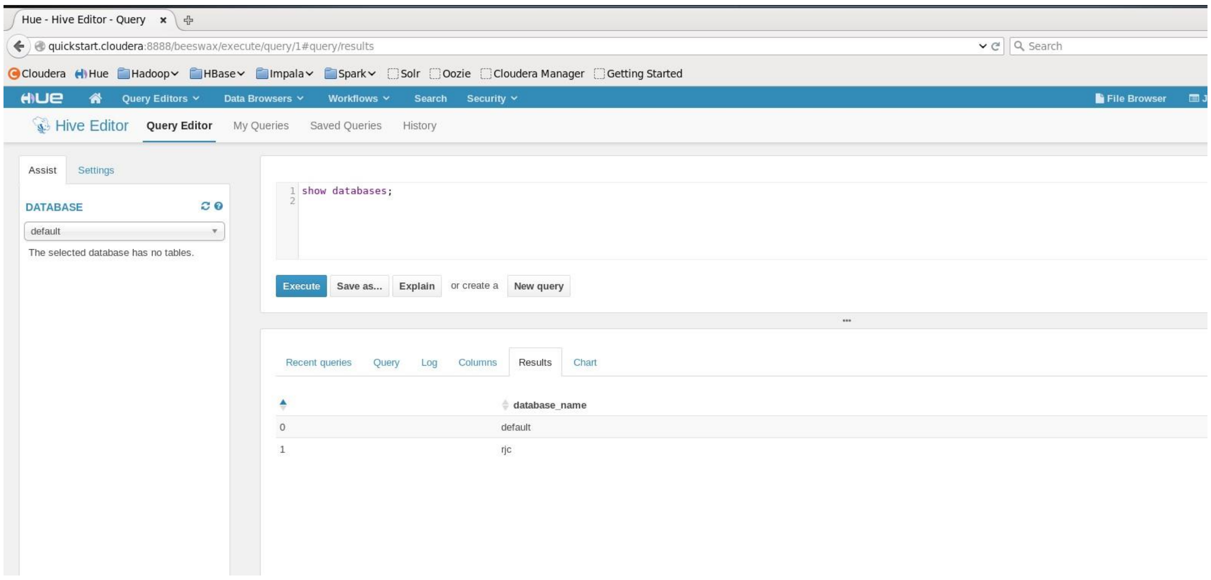
Quick Start Wizard - Hue™ 3.7.0 - The Hadoop UI

Step 1: Check Configuration Step 2: Examples Step 3: Users Step 4: Go!

Checking current configuration

Configuration files located in `/etc/hue/conf.empty`

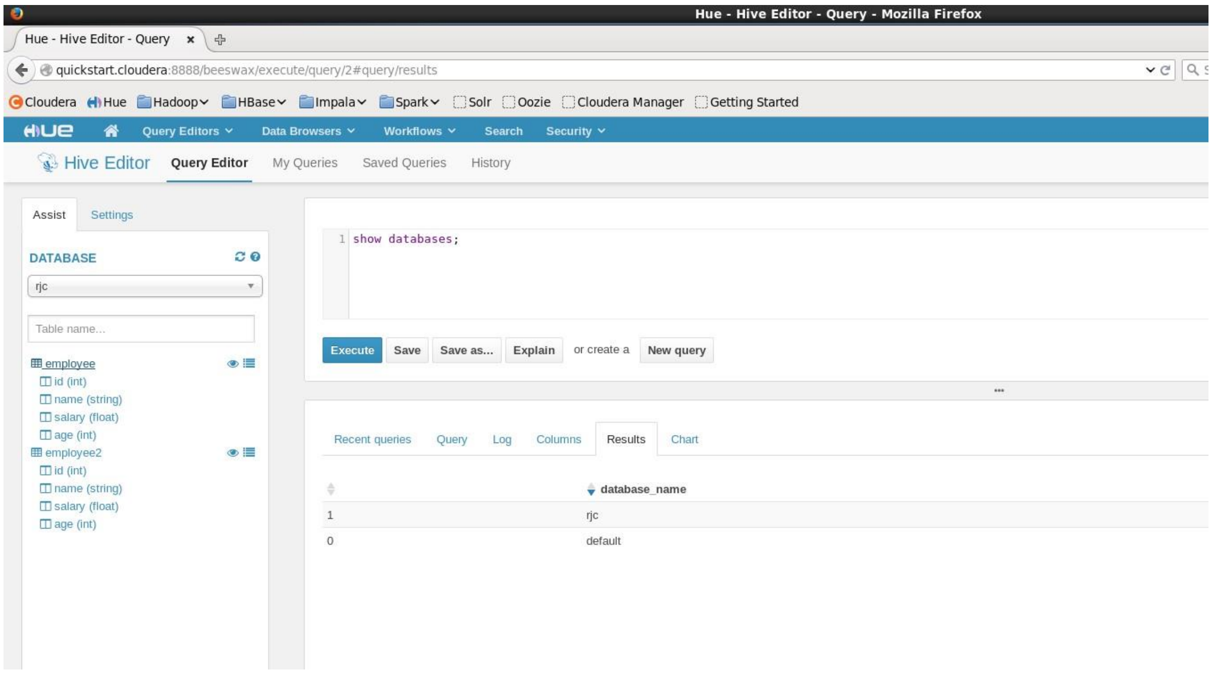
All OK. Configuration check passed.


Hue - Hive Editor - Query
quickstart.cloudera:8888/beeswax/execute/query/1#query/results
Cloudera Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started
Query Editors Data Browsers Workflows Search Security File Browser
Hive Editor Query Editor My Queries Saved Queries History
Assist Settings
DATABASE
default
Execute Save as... Explain or create a New query
Recent queries Query Log Columns Results Chart

The selected database has no tables.

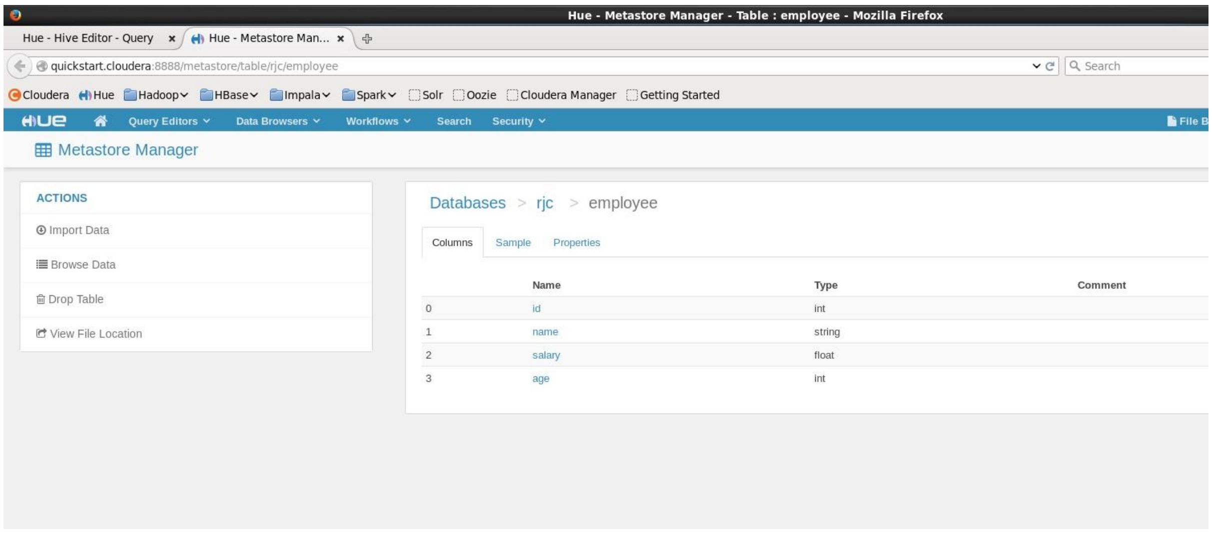
```
show databases;
```

```
    database_name
0   default
1   rjc
```



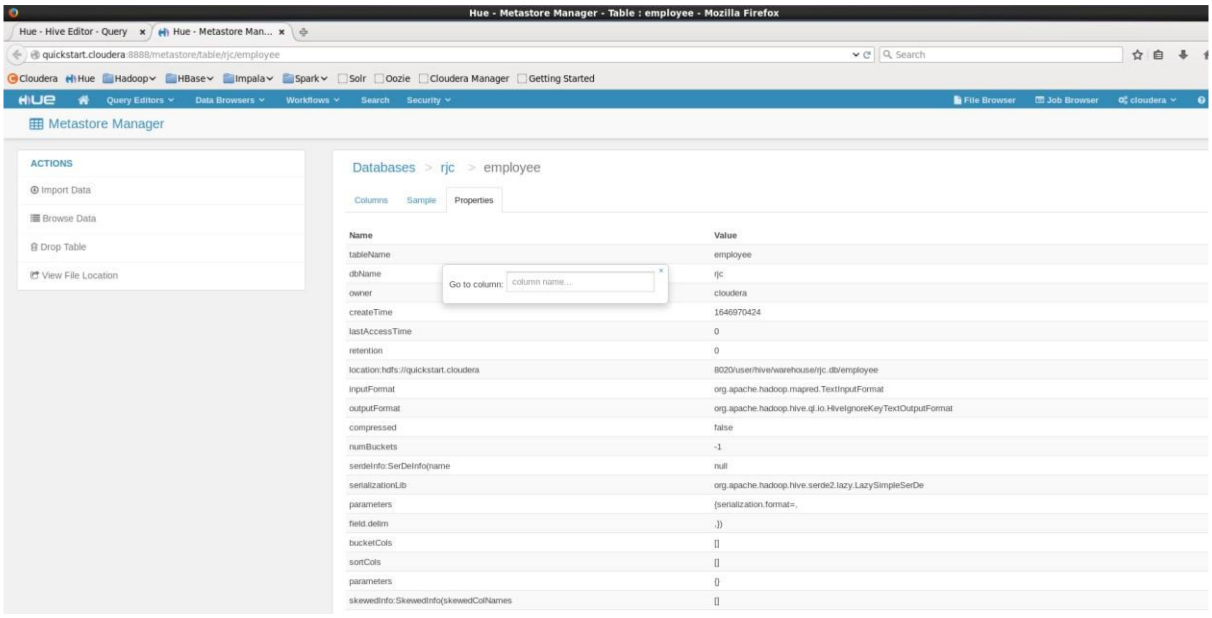
```
show databases;
```

	database_name
1	rjc
0	default



Databases > rjc > employee

	Name	Type	Comment
0	id	int	
1	name	string	
2	salary	float	
3	age	int	


Hue - Metastore Manager - Table : employee - Mozilla Firefox
Hue - Hive Editor - Query
Hue - Metastore Man...
quickstart.cloudera:8888/metastore/table/rjc/employee
Cloudera Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started
Query Editors Data Browsers Workflows Search Security
File Browser Job Browser cloudera
Metastore Manager
ACTIONS
Import Data
Browse Data
Drop Table
View File Location
Databases > rjc > employee
Columns Sample Properties
Go to column: column name...

Name	Value
tableName	employee
dbName	rjc
owner	cloudera
createTime	1646970424
lastAccessTime	0
retention	0
location:hdfs://quickstart.cloudera	8020/user/hive/warehouse/rjc.db/employee
inputFormat	org.apache.hadoop.mapred.TextInputFormat
outputFormat	org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
compressed	false
numBuckets	-1
serdeInfo:SerDeInfo(name	null
serializationLib	org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
parameters	{serialization.format=,
field.delim	,})
bucketCols	[]
sortCols	[]
parameters	{}
skewedInfo:SkewedInfo(skewedColNames	[]

15. Creating a new external table named as employee3 in the specific location using below command,

```
hive> create external table  employee3(ID int , name string , salary float, age int)
    > row format delimited
    > fields terminated by ','
    > location '/user/cloudera/vj';
OK
Time taken: 1.208 seconds
hive> █
```

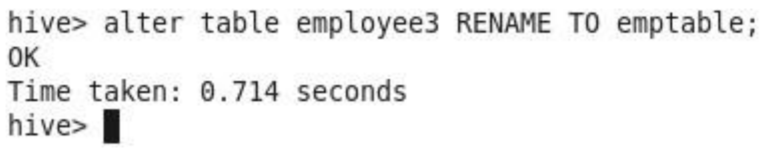
16. To see the schema of the employee3 table we use below command,

```
hive> describe employee3;
OK
id                      int
name                    string
salary                  float
age                     int
Time taken: 0.128 seconds, Fetched: 4 row(s)
hive> █
```

17. Then switch to browser and Refresh and select the rjc database and refresh, Now it will display the employee3 table along with other two tables which we have created earlier. 18. Now move to terminal and listing out all the tables using below command;

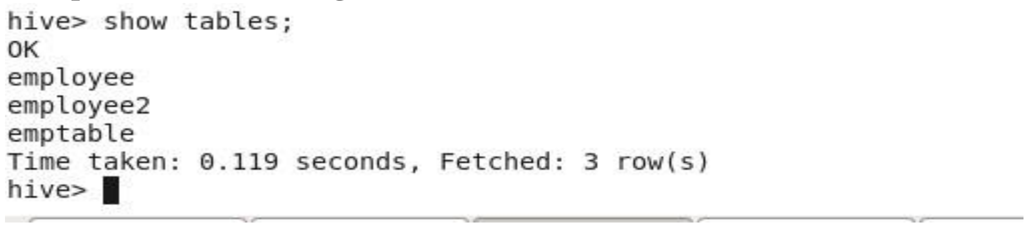
```
hive> show tables;
OK
employee
employee2
employee3
Time taken: 0.088 seconds, Fetched: 3 row(s)
hive> █
```

19. ALTER COMMANDS



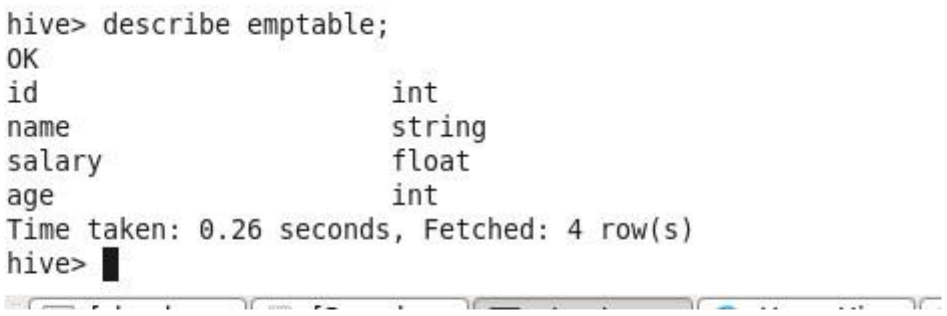
```
hive> alter table employee3 RENAME TO emptable;
OK
Time taken: 0.714 seconds
hive> █
```

20. Now we will check whether the name of the employee3 table changes to emptable or not using below command,



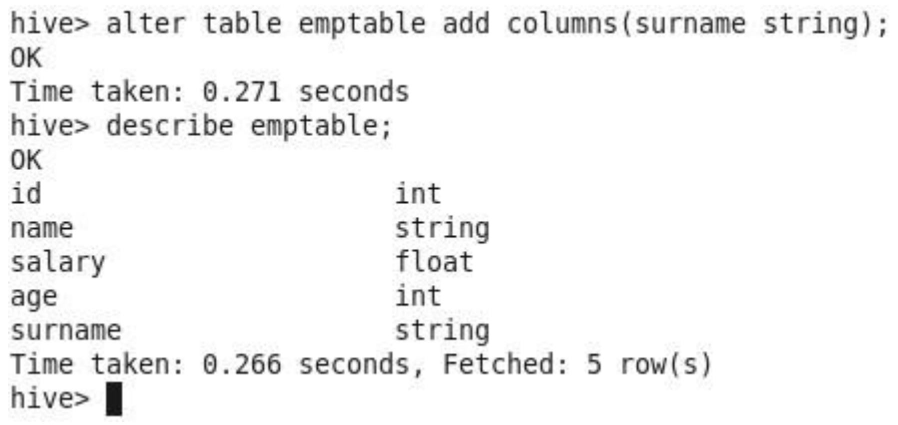
```
hive> show tables;
OK
employee
employee2
emptable
Time taken: 0.119 seconds, Fetched: 3 row(s)
hive> █
```

21. Describe emptable



```
hive> describe emptable;
OK
id                      int
name                    string
salary                  float
age                     int
Time taken: 0.26 seconds, Fetched: 4 row(s)
hive> █
```

22. First we will see the fields of emptable then we will add new column as surname in emptable using below command,



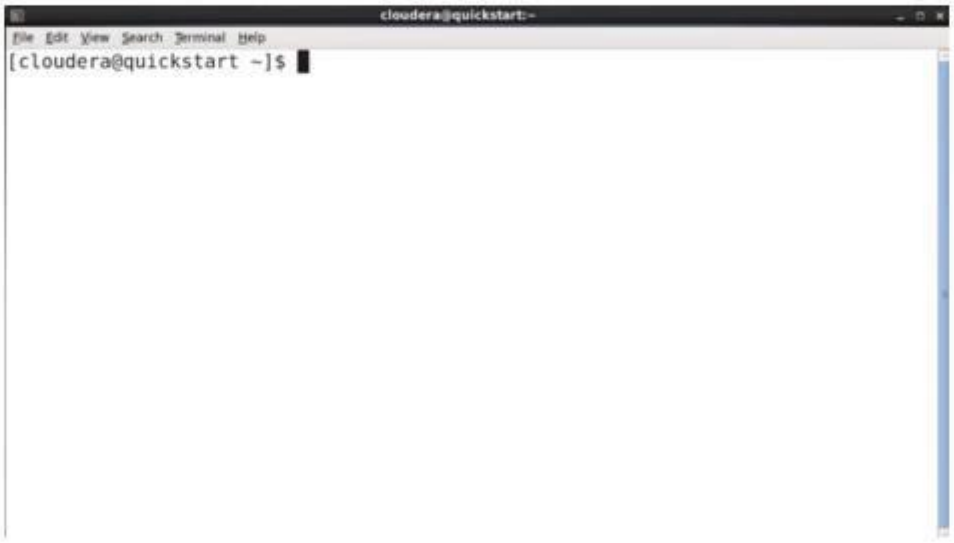
```
hive> alter table emptable add columns(surname string);
OK
Time taken: 0.271 seconds
hive> describe emptable;
OK
id                      int
name                    string
salary                  float
age                     int
surname                 string
Time taken: 0.266 seconds, Fetched: 5 row(s)
hive> █
```

23. Now we will change field name of the emptable to first_name using alter command,

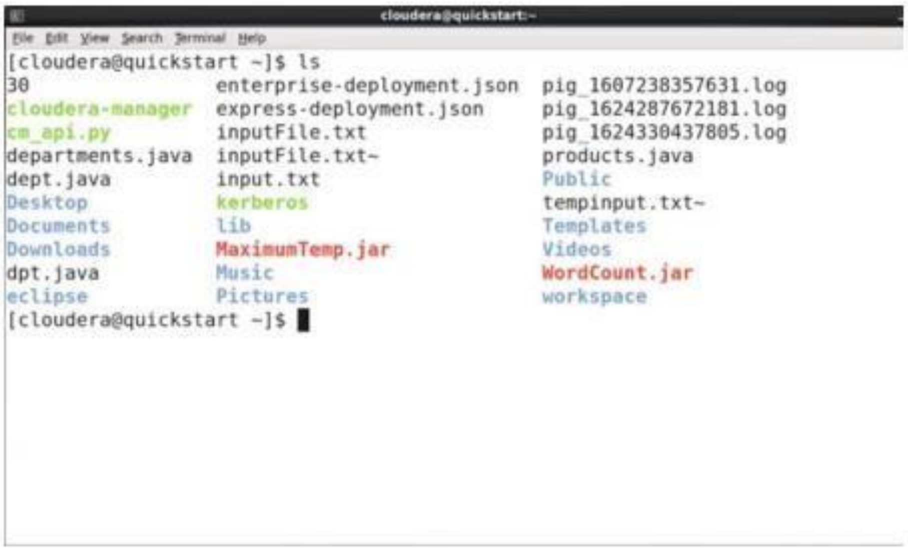
Loading the data in the table

24. Before loading the data in the table we will first create the csv file. Now open the new terminal ,using ls command list out all the directories --> change the directory to document directory

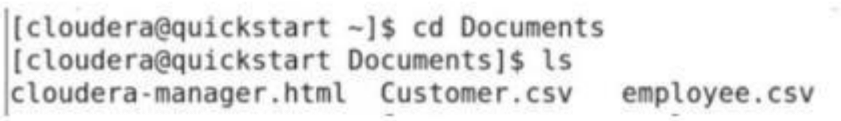
-->use ls command to list all the files present inside the document folder or directory



```
[cloudera@quickstart ~]$ 
```

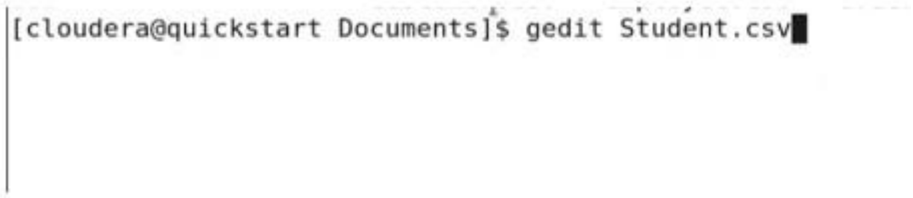


```
[cloudera@quickstart ~]$ ls
30                 enterprise-deployment.json  pig_1607238357631.log
cloudera-manager   express-deployment.json     pig_1624287672181.log
cm_api.py          inputFile.txt               pig_1624330437805.log
departments.java   inputFile.txt~              products.java
dept.java          input.txt                   Public
Desktop            kerberos                    tempinput.txt~
Documents          lib                         Templates
Downloads          MaximumTemp.jar             Videos
dpt.java           Music                       WordCount.jar
eclipse            Pictures                    workspace
[cloudera@quickstart ~]$ 
```



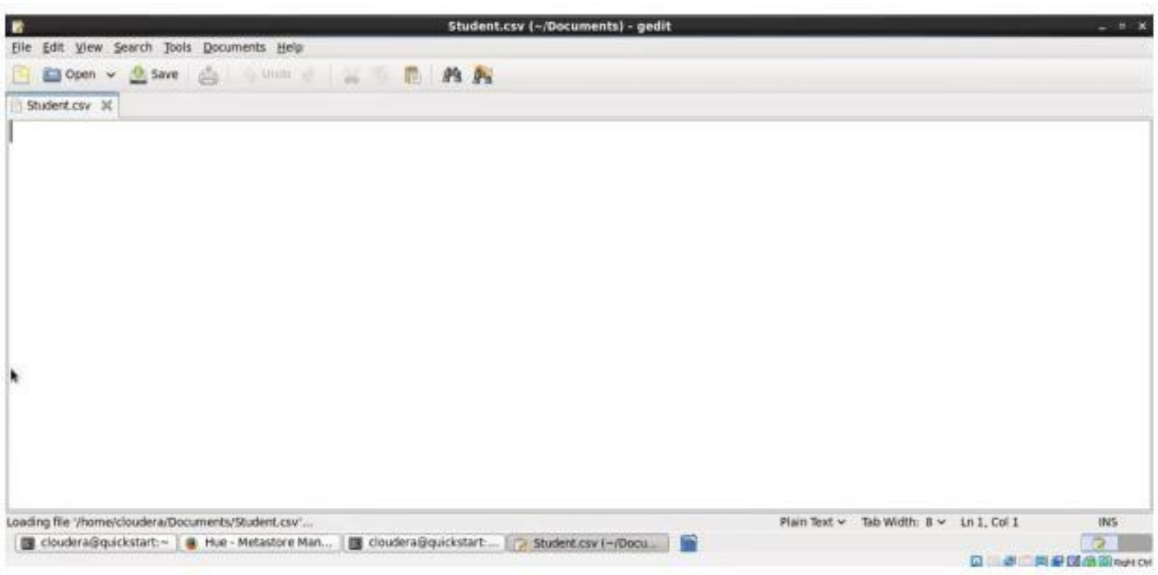
```
[cloudera@quickstart ~]$ cd Documents
[cloudera@quickstart Documents]$ ls
cloudera-manager.html  Customer.csv   employee.csv
```

25. Now creating new file as Student.csv using below command,



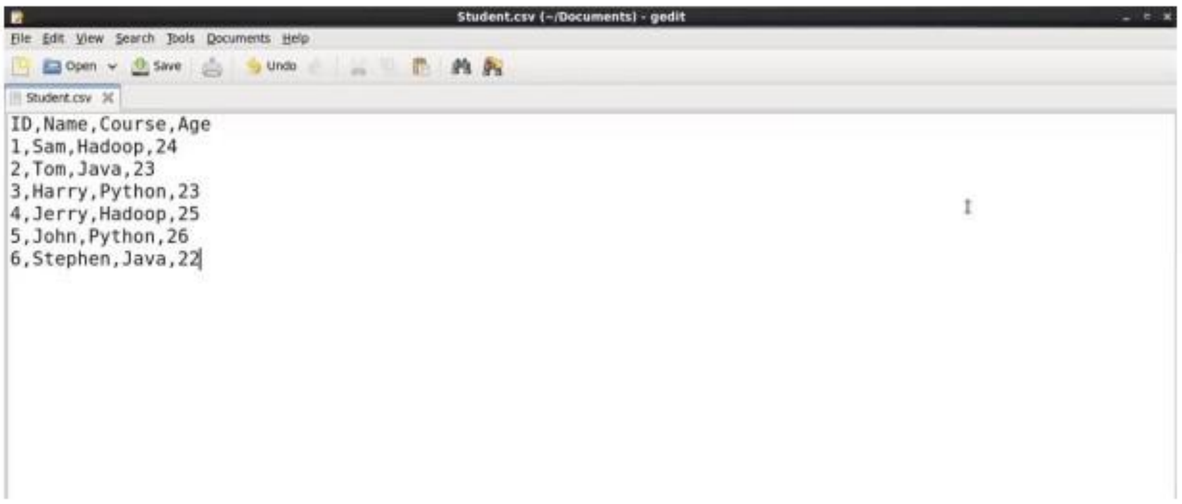
```
[cloudera@quickstart Documents]$ gedit Student.csv
```

As soon as we hit enter it will create a Student.csv file as it was not existing earlier.


Student.csv (~/Documents) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
Student.csv
Plain Text Tab Width: 8 Ln 1, Col 1 INS

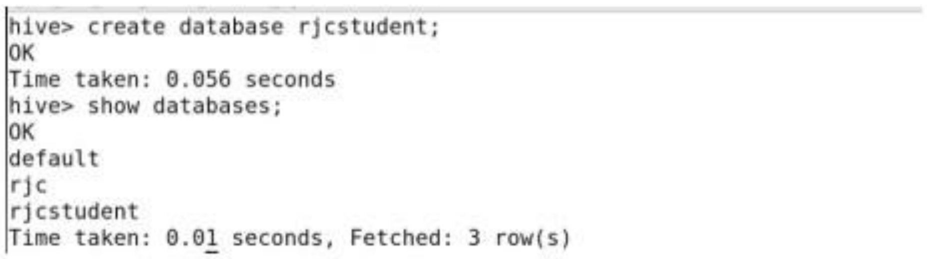
```
Loading file '/home/cloudera/Documents/Student.csv'...
```

Now we are populating the Student.csv file with some data and save this file.


Student.csv (~/Documents) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
Student.csv

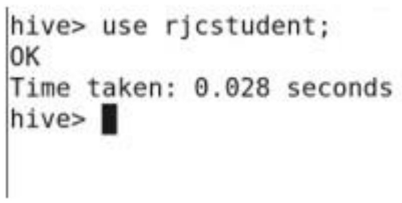
```
ID,Name,Course,Age
1,Sam,Hadoop,24
2,Tom,Java,23
3,Harry,Python,23
4,Jerry,Hadoop,25
5,John,Python,26
6,Stephen,Java,22
```

26. Creating a new database as rjcstudent.



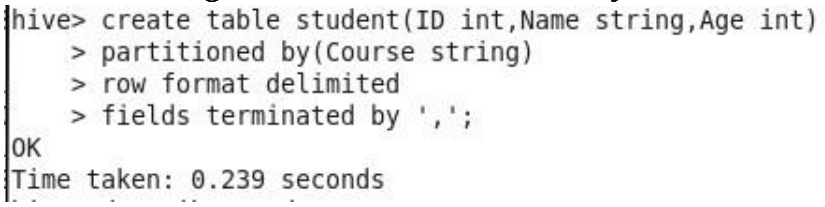
```
hive> create database rjcstudent;
OK
Time taken: 0.056 seconds
hive> show databases;
OK
default
rjc
rjcstudent
Time taken: 0.01 seconds, Fetched: 3 row(s)
```

27. Using rjcstudent database.



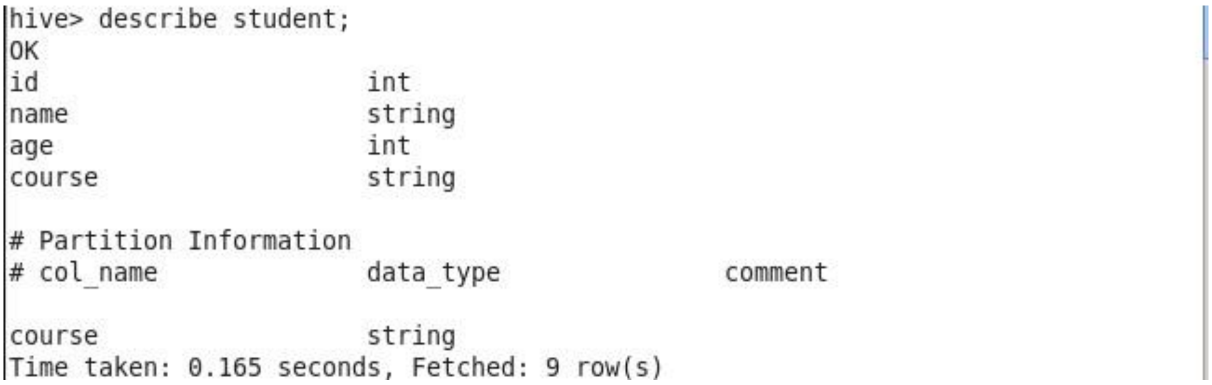
```
hive> use rjcstudent;
OK
Time taken: 0.028 seconds
hive> █
```

28. Creating new table student inside rjcstudent database.



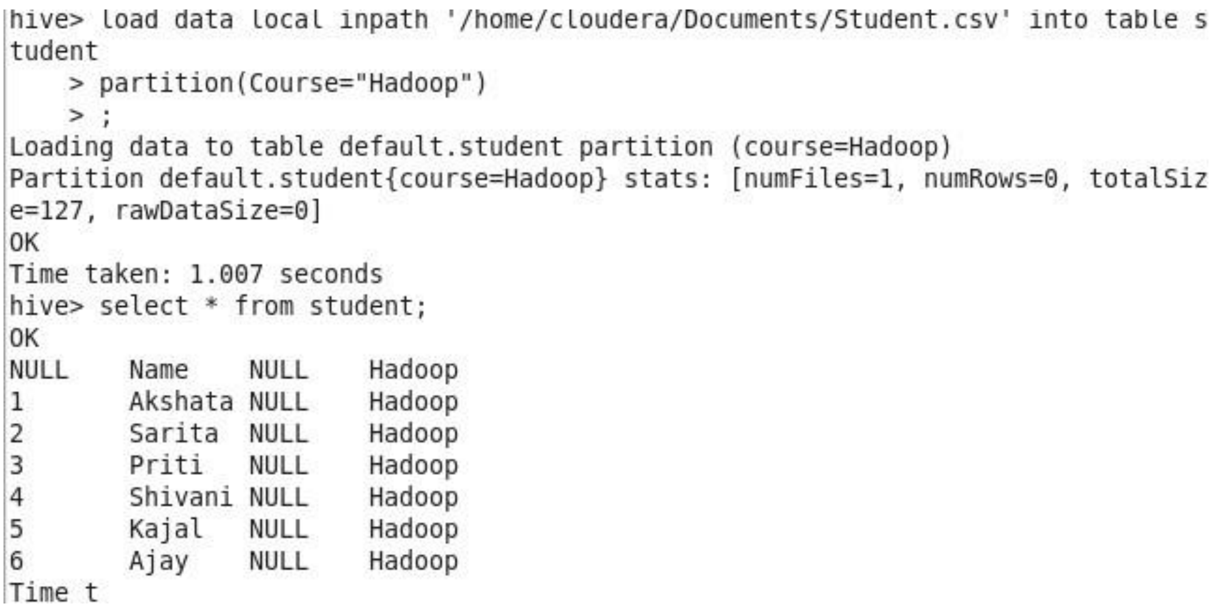
```
hive> create table student(ID int,Name string,Age int)
    > partitioned by(Course string)
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.239 seconds
```

29. To see the structure or schema of the table,



```
hive> describe student;
OK
id                      int
name                    string
age                     int
course                  string

# Partition Information
# col_name              data_type               comment

course                  string
Time taken: 0.165 seconds, Fetched: 9 row(s)
```

30. Loading data in the student table from Student.csv file which we have created in document directory. Here we are partitioning based on course = 'Hadoop'.

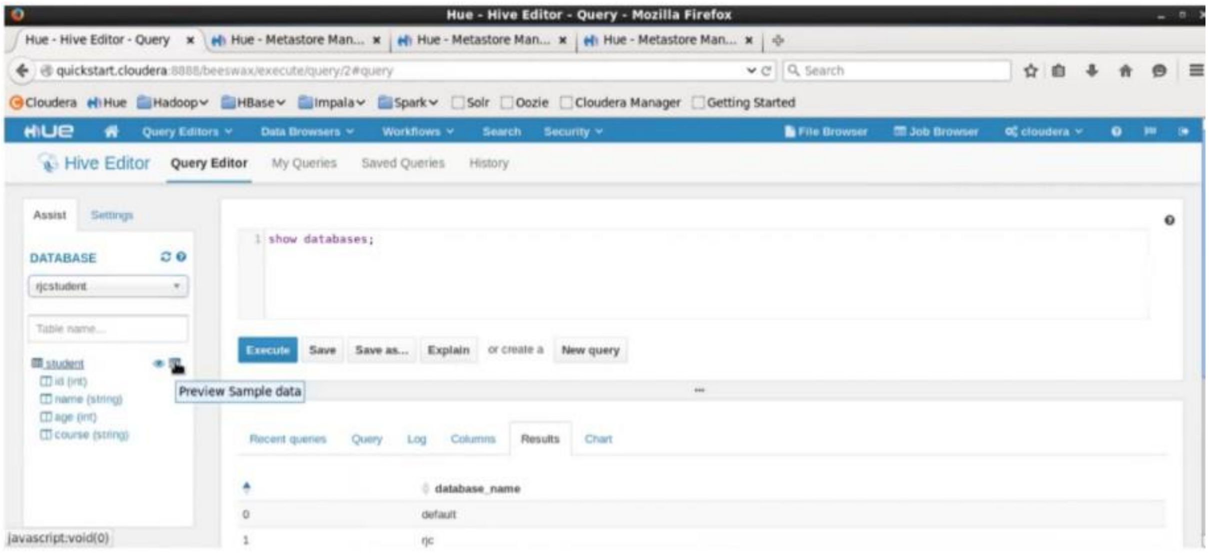


```
hive> load data local inpath '/home/cloudera/Documents/Student.csv' into table s
tudent
    > partition(Course="Hadoop")
    > ;
Loading data to table default.student partition (course=Hadoop)
Partition default.student{course=Hadoop} stats: [numFiles=1, numRows=0, totalSiz
e=127, rawDataSize=0]
OK
Time taken: 1.007 seconds
hive> select * from student;
OK
NULL    Name    NULL    Hadoop
1       Akshata NULL    Hadoop
2       Sarita  NULL    Hadoop
3       Priti   NULL    Hadoop
4       Shivani NULL    Hadoop
5       Kajal   NULL    Hadoop
6       Ajay    NULL    Hadoop
Time t
```

31. Now we similarly partition for course = Java and course = Python.

```
hive> load data local inpath '/home/cloudera/Documents/Student.csv' into table s
tudent
    > partition(Course="Java");;
Loading data to table default.student partition (course=Java)
Partition default.student{course=Java} stats: [numFiles=1, numRows=0, totalSize=
127, rawDataSize=0]
OK
Time taken: 0.42 seconds
hive> load data local inpath '/home/cloudera/Documents/Student.csv' into table s
tudentpartition(Course="Python");
Loading data to table default.student partition (course=Python)
Partition default.student{course=Python} stats: [numFiles=1, numRows=0, totalSiz
e=127, rawDataSize=0]
OK
Time taken: 0.558 seconds
```

32. Now go to browser refresh the page and select database as rjcstudent and click in preview student table.



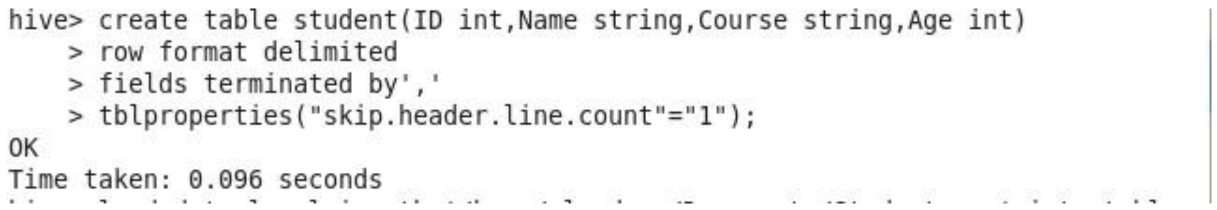
```
1 show databases;
```

```
    database_name
0   default
1   rjc
```

33. Now dropping the table student and creating the student table again normally (i.e. without

partitioning).

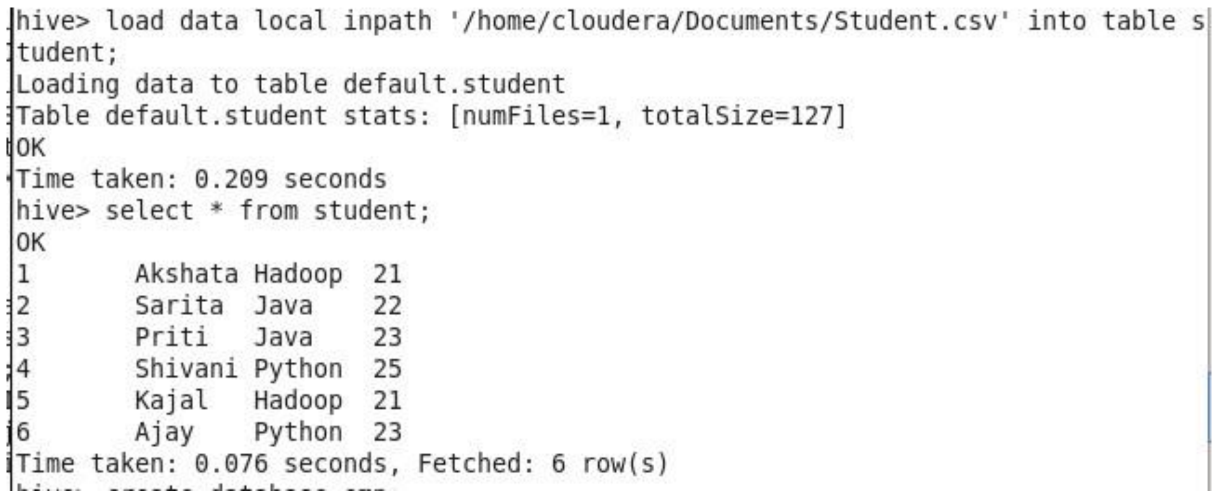
```
hive> drop table student;
OK
Time t;
OK
Time taken: 0.053 seconds
```



```
hive> create table student(ID int,Name string,Course string,Age int)
    > row format delimited
    > fields terminated by','
    > tblproperties("skip.header.line.count"="1");
OK
Time taken: 0.096 seconds
```

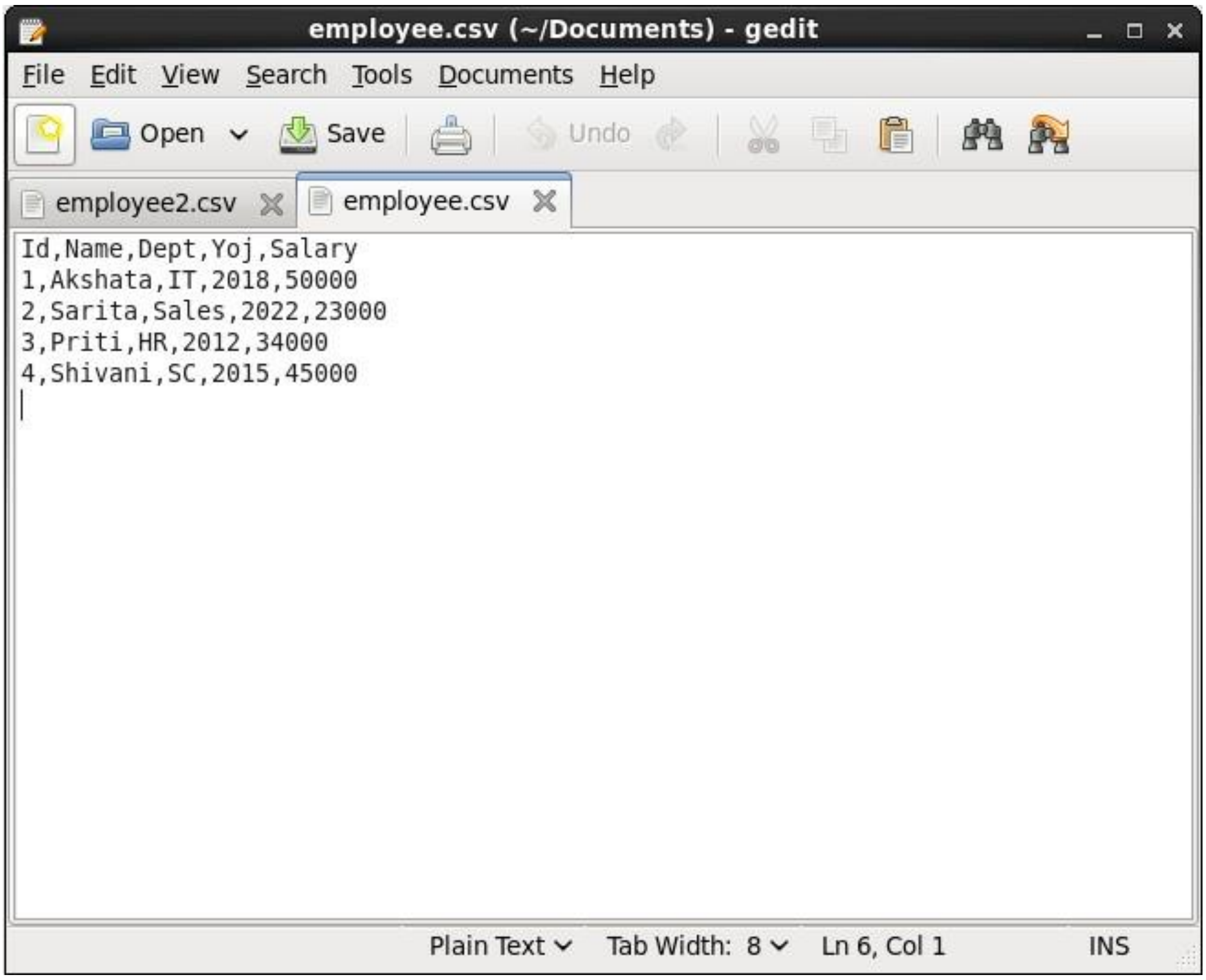
34. Loading data in the student table from Student.csv file which present inside

/home/cloudera/Documents directory.



```
hive> load data local inpath '/home/cloudera/Documents/Student.csv' into table s
tudent;
Loading data to table default.student
Table default.student stats: [numFiles=1, totalSize=127]
OK
Time taken: 0.209 seconds
hive> select * from student;
OK
1       Akshata Hadoop  21
2       Sarita  Java    22
3       Priti   Java    23
4       Shivani Python  25
5       Kajal   Hadoop  21
6       Ajay    Python  23
Time taken: 0.076 seconds, Fetched: 6 row(s)
```

35. Now creating employee.csv file.


employee.csv (~/Documents) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
employee2.csv
employee.csv
Plain Text
Tab Width: 8
Ln 6, Col 1
INS

```
Id,Name,Dept,Yoj,Salary
1,Akshata,IT,2018,50000
2,Sarita,Sales,2022,23000
3,Priti,HR,2012,34000
4,Shivani,SC,2015,45000
```

Querying :

36. Creating new database for performing querying operations.

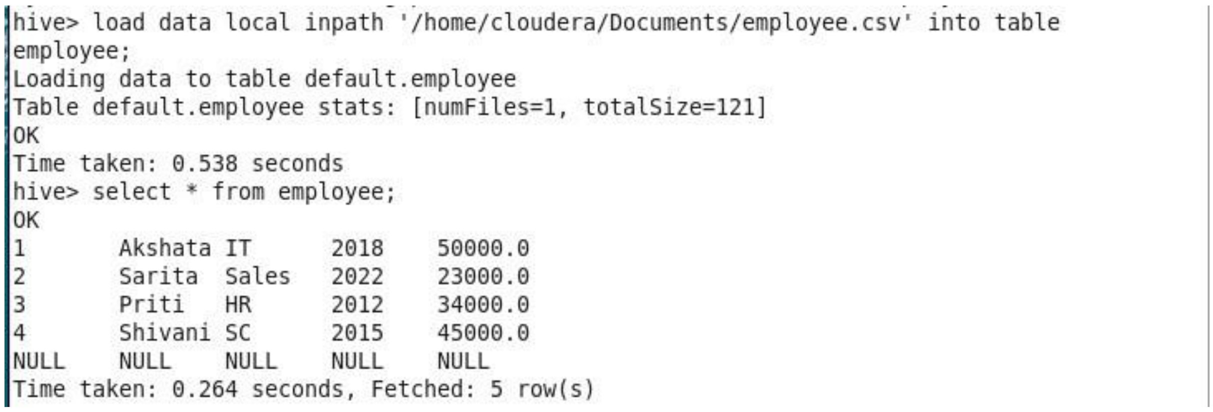
```
hive> create database hiveql;
OK
Time taken: 0.154 seconds
```

37. Using database hiveql and creating table employee inside the hiveql database.

```
hive> create table employee(ID int,Name string,Department string,YOJ int,Salary
float)
    > ROW FORMAT DELIMITED
    >  FIELDS TERMINATED BY ','
    > tblproperties("skip.header.line.count"="1");
OK
Time taken: 0.299 seconds
hive> describe employee;
OK
id                      int
name                    string
department              string
yoj                     int
salary                  float
Time taken: 0.182 seconds, Fetched: 5 row(s)
```

38. Loading the data into employee table from employee.csv file which we have created

earlier and it is present in /home/cloudera/Documents directory.

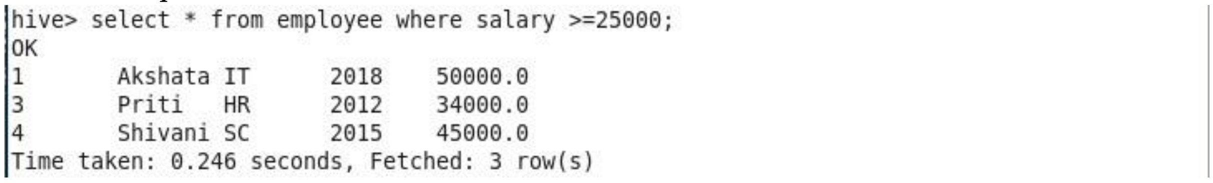


```
hive> load data local inpath '/home/cloudera/Documents/employee.csv' into table
employee;
Loading data to table default.employee
Table default.employee stats: [numFiles=1, totalSize=121]
OK
Time taken: 0.538 seconds
hive> select * from employee;
OK
1       Akshata IT      2018    50000.0
2       Sarita  Sales   2022    23000.0
3       Priti   HR      2012    34000.0
4       Shivani SC      2015    45000.0
NULL    NULL    NULL    NULL    NULL
Time taken: 0.264 seconds, Fetched: 5 row(s)
```

Now we Perform some operations on this employee table.

39. We are printing or displaying the rows or records of employees whose salary is greater

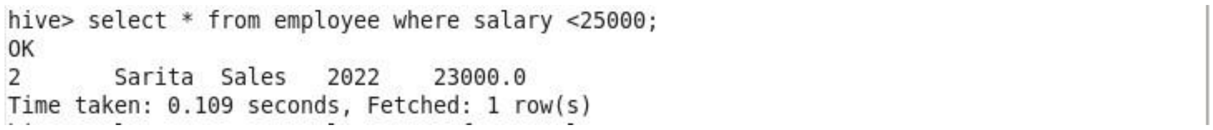
than and equal to 25000.



```
hive> select * from employee where salary >=25000;
OK
1       Akshata IT      2018    50000.0
3       Priti   HR      2012    34000.0
4       Shivani SC      2015    45000.0
Time taken: 0.246 seconds, Fetched: 3 row(s)
```

40. Now we are printing or displaying the rows or records of employees

whose salary is less than 25000.



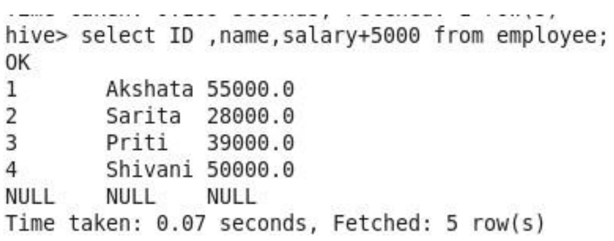
```
hive> select * from employee where salary <25000;
OK
2       Sarita  Sales   2022    23000.0
Time taken: 0.109 seconds, Fetched: 1 row(s)
```

Aggregating

41.Arithmetic operations:

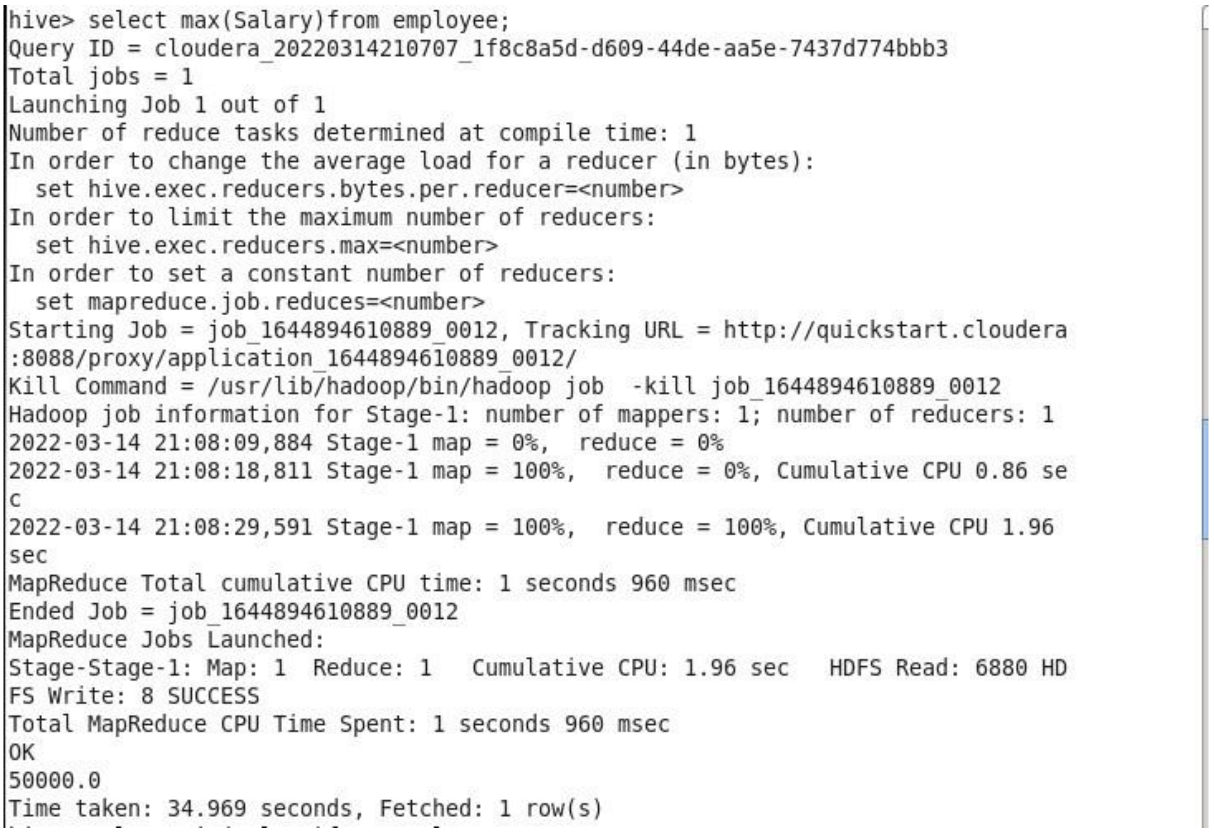
We are adding 5000 in existing salary and displaying specific columns

using below command,



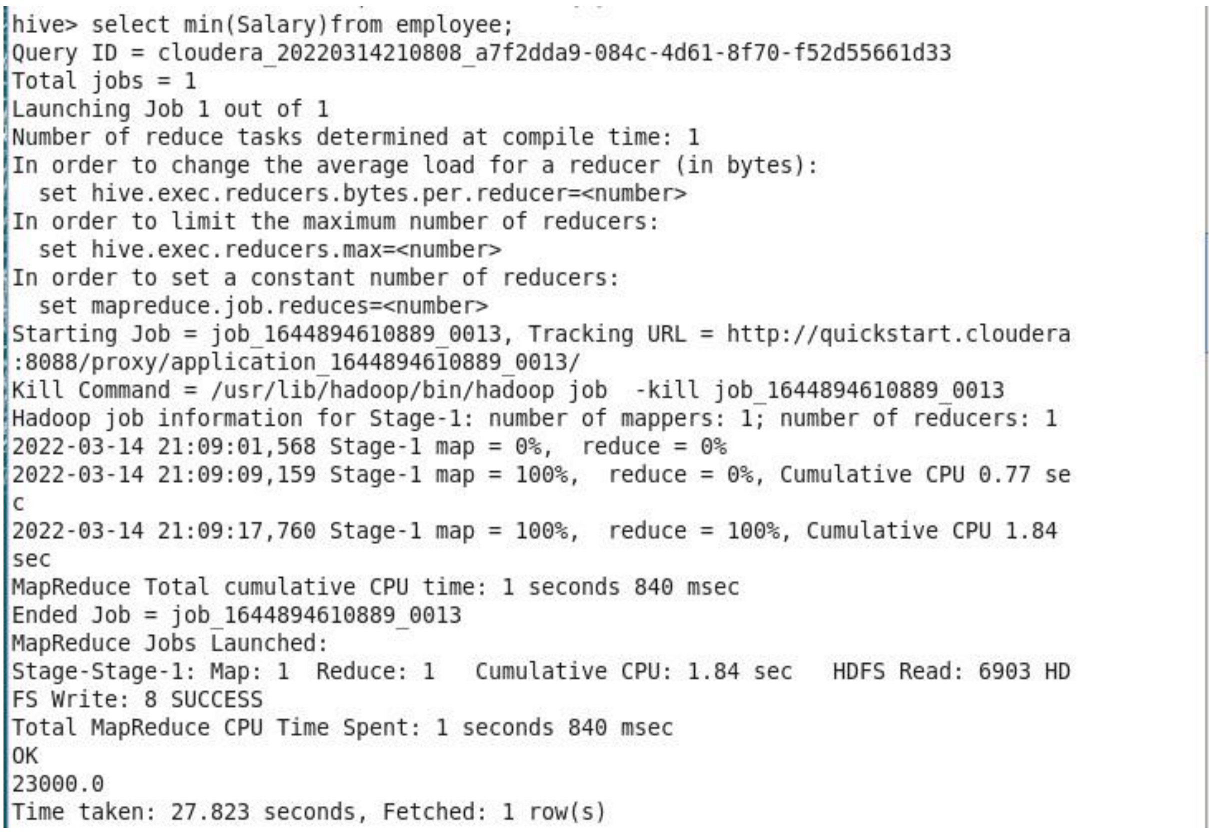
```
hive> select ID ,name,salary+5000 from employee;
OK
1       Akshata 55000.0
2       Sarita  28000.0
3       Priti   39000.0
4       Shivani 50000.0
NULL    NULL    NULL
Time taken: 0.07 seconds, Fetched: 5 row(s)
```

42. Displaying the maximum salary using below command, select max(salary) from employee;



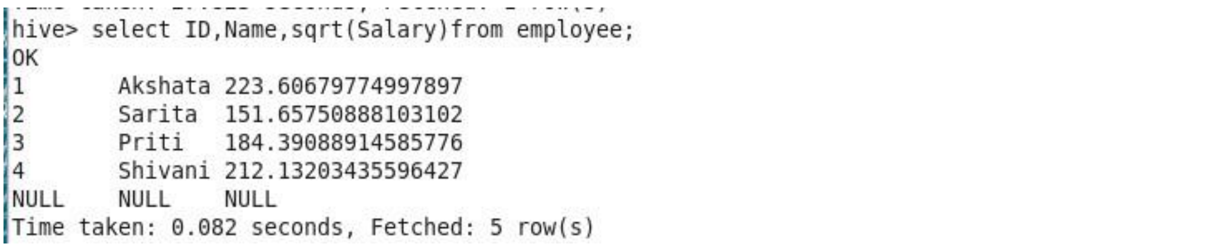
```
hive> select max(Salary)from employee;
Query ID = cloudera_20220314210707_1f8c8a5d-d609-44de-aa5e-7437d774bbb3
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<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_1644894610889_0012, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1644894610889_0012/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1644894610889_0012
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-03-14 21:08:09,884 Stage-1 map = 0%,  reduce = 0%
2022-03-14 21:08:18,811 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.86 se
c
2022-03-14 21:08:29,591 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 1.96
sec
MapReduce Total cumulative CPU time: 1 seconds 960 msec
Ended Job = job_1644894610889_0012
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.96 sec   HDFS Read: 6880 HD
FS Write: 8 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 960 msec
OK
50000.0
Time taken: 34.969 seconds, Fetched: 1 row(s)
```

43. Displaying the minimum salary using below command,



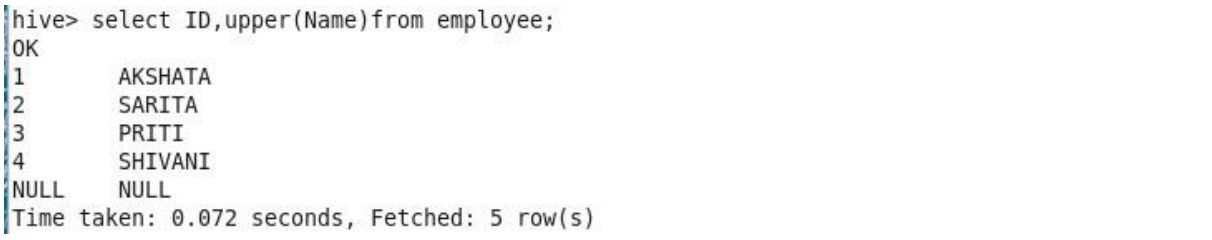
```
hive> select min(Salary)from employee;
Query ID = cloudera_20220314210808_a7f2dda9-084c-4d61-8f70-f52d55661d33
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<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_1644894610889_0013, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1644894610889_0013/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1644894610889_0013
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-03-14 21:09:01,568 Stage-1 map = 0%,  reduce = 0%
2022-03-14 21:09:09,159 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.77 se
c
2022-03-14 21:09:17,760 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 1.84
sec
MapReduce Total cumulative CPU time: 1 seconds 840 msec
Ended Job = job_1644894610889_0013
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.84 sec   HDFS Read: 6903 HD
FS Write: 8 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 840 msec
OK
23000.0
Time taken: 27.823 seconds, Fetched: 1 row(s)
```

44. Now finding the square root of salary using below command,



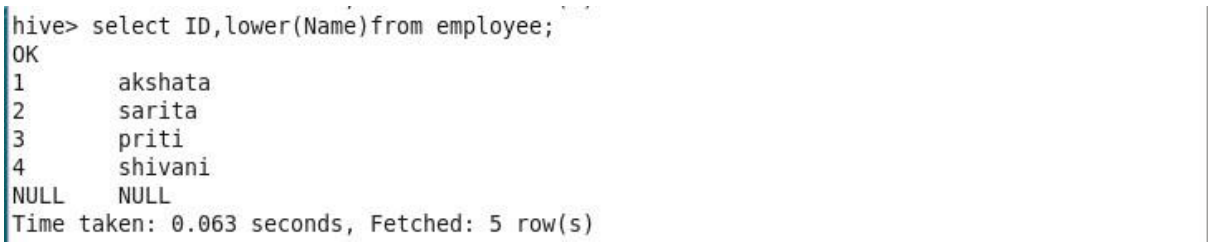
```
hive> select ID,Name,sqrt(Salary)from employee;
OK
1       Akshata 223.60679774997897
2       Sarita  151.65750888103102
3       Priti   184.39088914585776
4       Shivani 212.13203435596427
NULL    NULL    NULL
Time taken: 0.082 seconds, Fetched: 5 row(s)
```

45. Now we are showing the name column in upper case using below command,



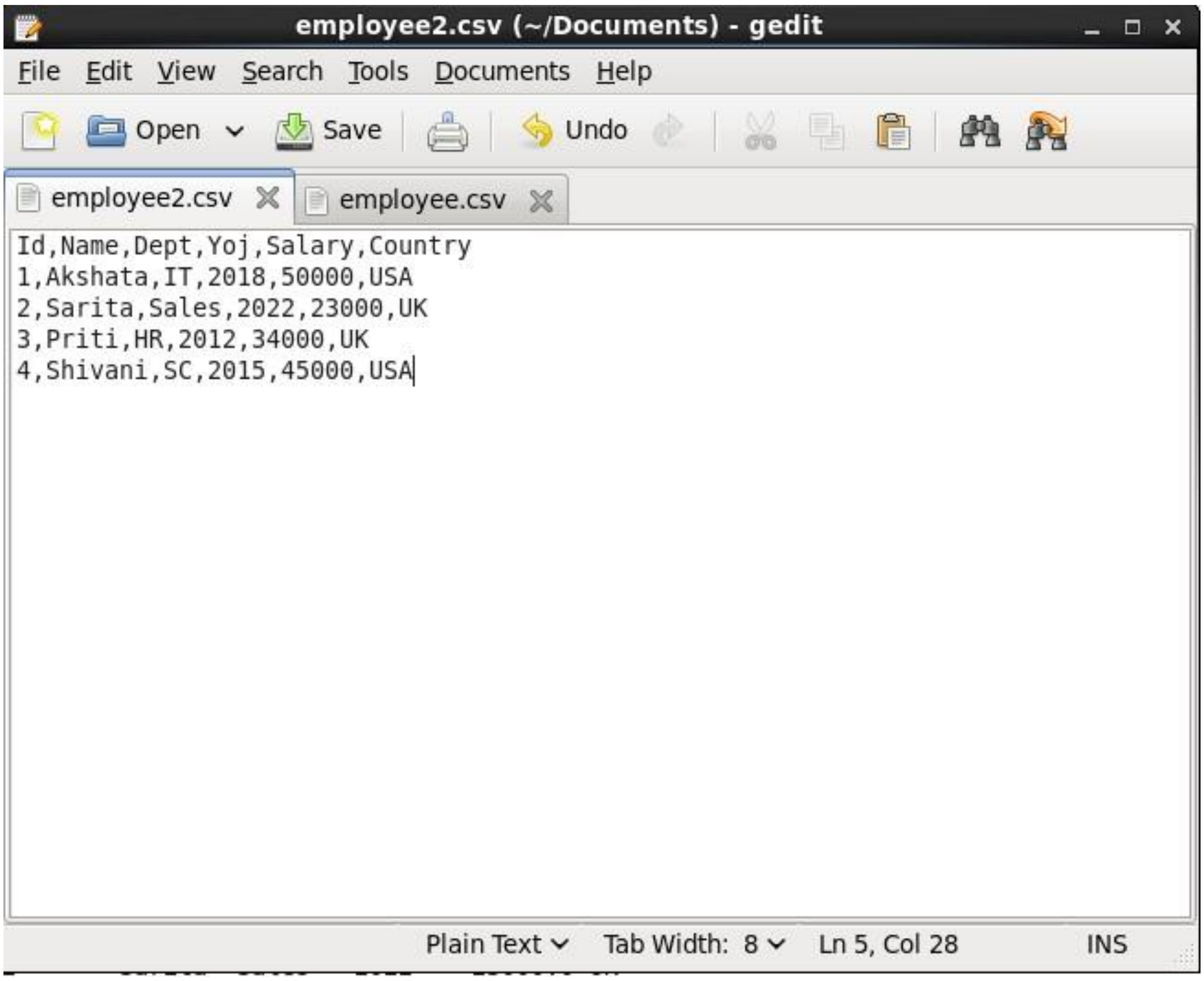
```
hive> select ID,upper(Name)from employee;
OK
1       AKSHATA
2       SARITA
3       PRITI
4       SHIVANI
NULL    NULL
Time taken: 0.072 seconds, Fetched: 5 row(s)
```

46. Now we are showing the name column in lower case using below command,



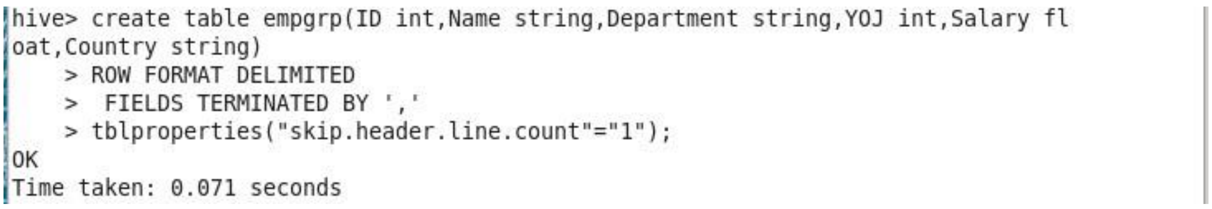
```
hive> select ID,lower(Name)from employee;
OK
1       akshata
2       sarita
3       priti
4       shivani
NULL    NULL
Time taken: 0.063 seconds, Fetched: 5 row(s)
```

47. Creating another employee2.csv file with the same data as employee.csv but adding one more column as Country to it.


employee2.csv (~/Documents) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
employee2.csv
employee.csv
Plain Text
Tab Width: 8
Ln 5, Col 28
INS

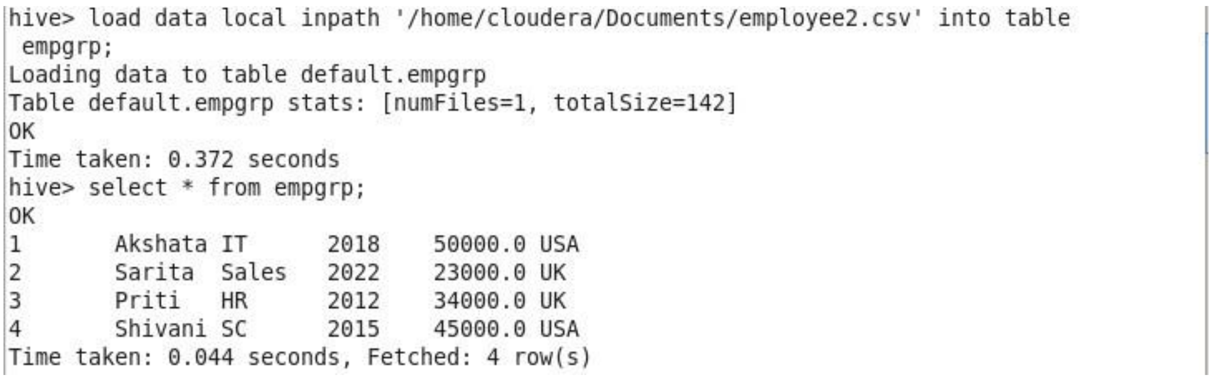
```
Id,Name,Dept,Yoj,Salary,Country
1,Akshata,IT,2018,50000,USA
2,Sarita,Sales,2022,23000,UK
3,Priti,HR,2012,34000,UK
4,Shivani,SC,2015,45000,USA
```

48. Creating a new table as empgroup.



```
hive> create table empgrp(ID int,Name string,Department string,YOJ int,Salary fl
oat,Country string)
    > ROW FORMAT DELIMITED
    >  FIELDS TERMINATED BY ','
    > tblproperties("skip.header.line.count"="1");
OK
Time taken: 0.071 seconds
```

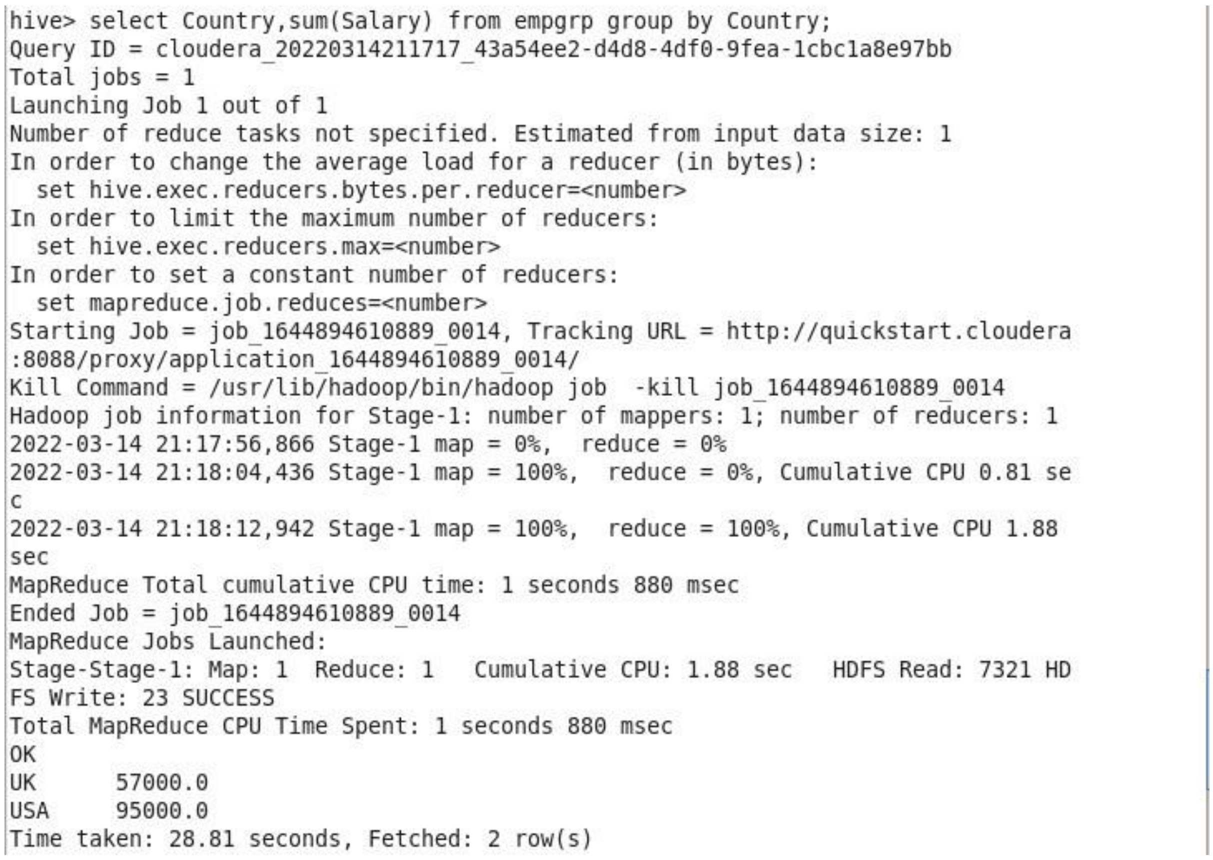
49. Loading the data into empgroup table from employee2.csv file which we have created and it is present in /home/cloudera/Documents directory.



```
hive> load data local inpath '/home/cloudera/Documents/employee2.csv' into table
 empgrp;
Loading data to table default.empgrp
Table default.empgrp stats: [numFiles=1, totalSize=142]
OK
Time taken: 0.372 seconds
hive> select * from empgrp;
OK
1       Akshata IT      2018    50000.0 USA
2       Sarita  Sales   2022    23000.0 UK
3       Priti   HR      2012    34000.0 UK
4       Shivani SC      2015    45000.0 USA
Time taken: 0.044 seconds, Fetched: 4 row(s)
```

50.Groupby clause

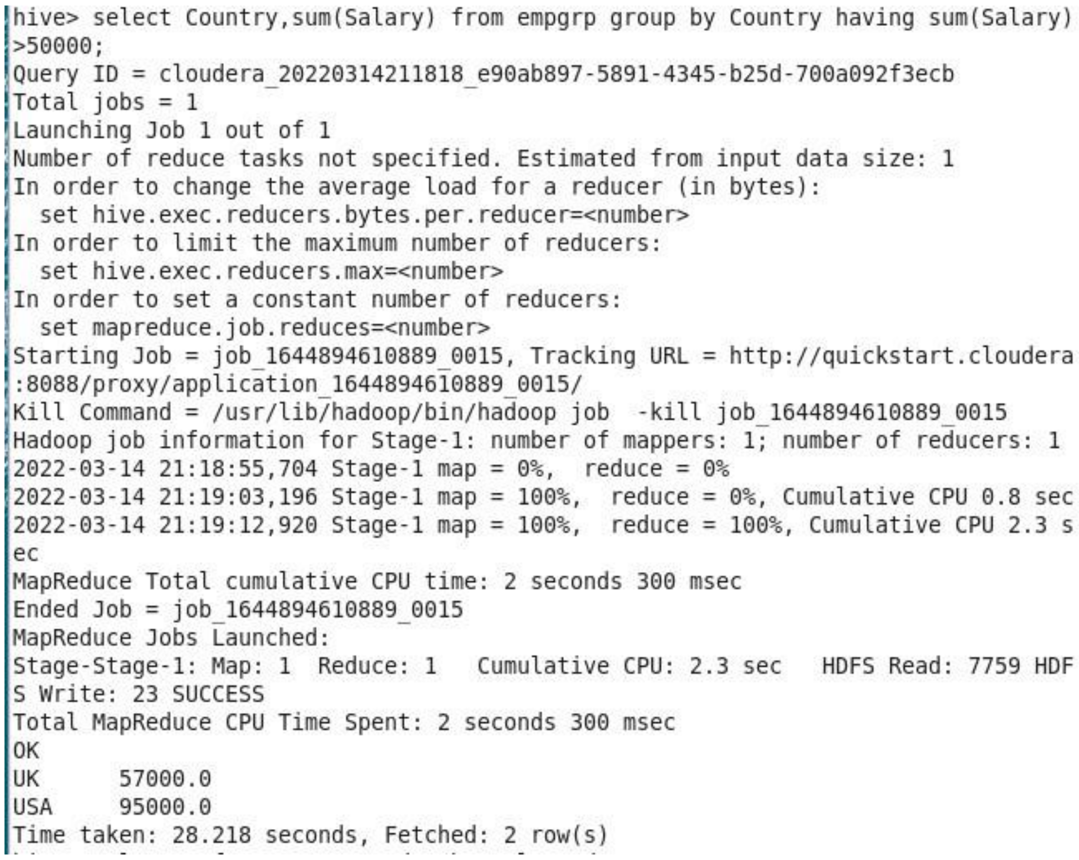
Now we display the total sum of salary of employees country wise using below command,



```
hive> select Country,sum(Salary) from empgrp group by Country;
Query ID = cloudera_20220314211717_43a54ee2-d4d8-4df0-9fea-1cbc1a8e97bb
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_1644894610889_0014, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1644894610889_0014/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1644894610889_0014
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-03-14 21:17:56,866 Stage-1 map = 0%,  reduce = 0%
2022-03-14 21:18:04,436 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.81 se
c
2022-03-14 21:18:12,942 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 1.88
sec
MapReduce Total cumulative CPU time: 1 seconds 880 msec
Ended Job = job_1644894610889_0014
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.88 sec   HDFS Read: 7321 HD
FS Write: 23 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 880 msec
OK
UK      57000.0
USA     95000.0
Time taken: 28.81 seconds, Fetched: 2 row(s)
```

51. Groupby clause along with the having clause

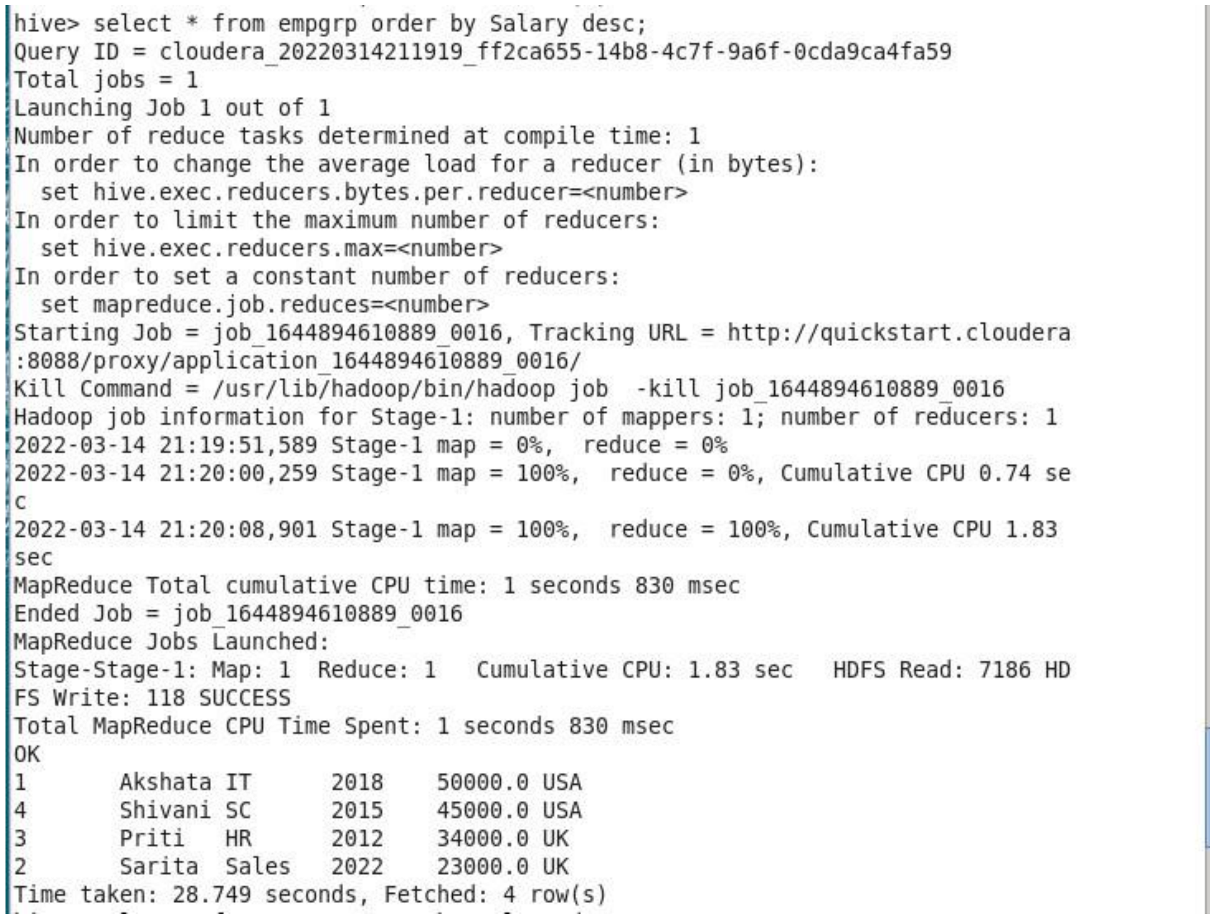
Taking the total sum of salary countrywise using groupby clause and from that selecting or displaying those country whose total sum of salary>50000 using having clause.



```
hive> select Country,sum(Salary) from empgrp group by Country having sum(Salary)
>50000;
Query ID = cloudera_20220314211818_e90ab897-5891-4345-b25d-700a092f3ecb
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_1644894610889_0015, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1644894610889_0015/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1644894610889_0015
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-03-14 21:18:55,704 Stage-1 map = 0%,  reduce = 0%
2022-03-14 21:19:03,196 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.8 sec
2022-03-14 21:19:12,920 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.3 s
ec
MapReduce Total cumulative CPU time: 2 seconds 300 msec
Ended Job = job_1644894610889_0015
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.3 sec   HDFS Read: 7759 HDF
S Write: 23 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 300 msec
OK
UK      57000.0
USA     95000.0
Time taken: 28.218 seconds, Fetched: 2 row(s)
```

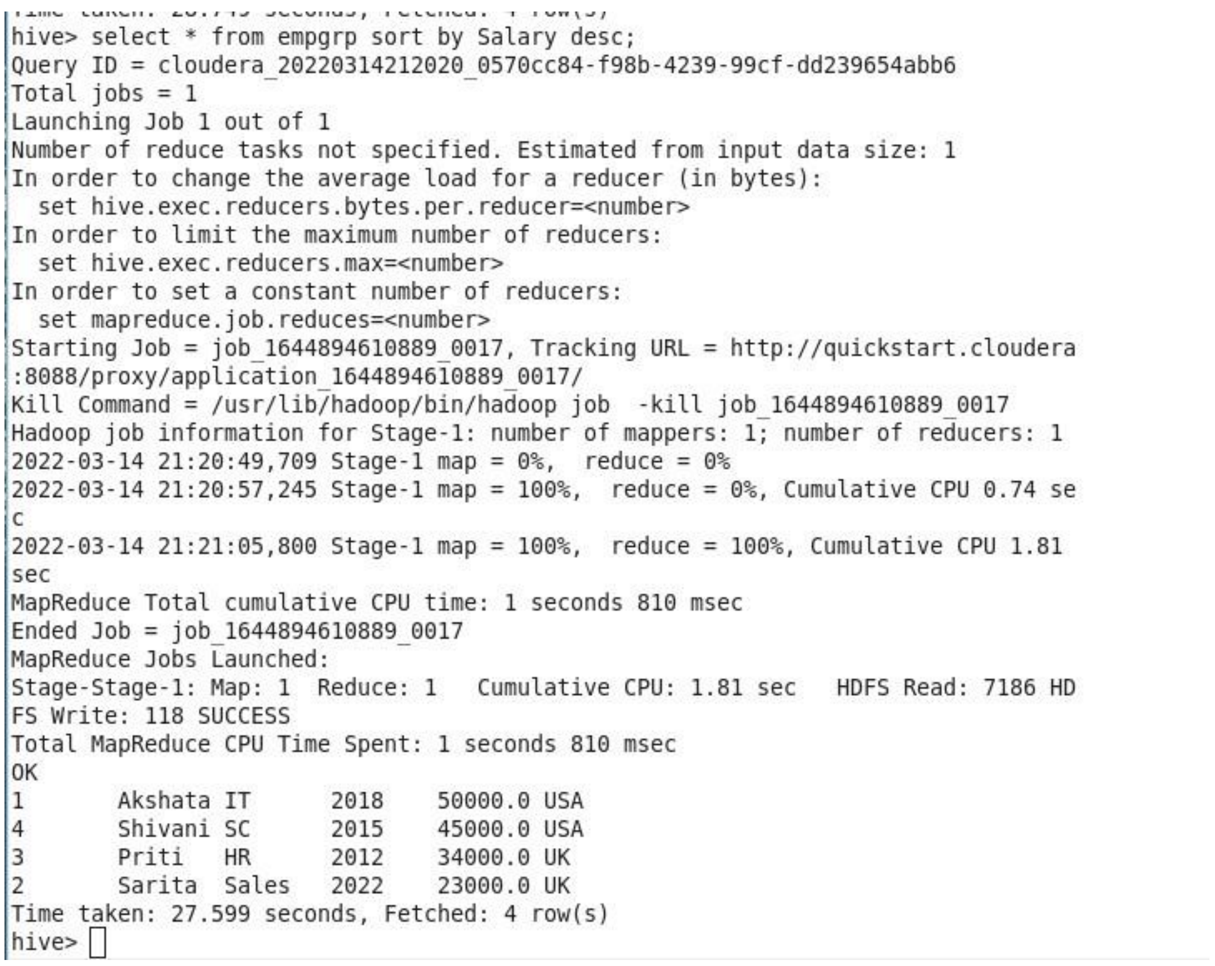
Sorting : Order by

52. Now we are displaying the table by sorting the salary in descending order.



```
hive> select * from empgrp order by Salary desc;
Query ID = cloudera_20220314211919_ff2ca655-14b8-4c7f-9a6f-0cda9ca4fa59
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_1644894610889_0016, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1644894610889_0016/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1644894610889_0016
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-03-14 21:19:51,589 Stage-1 map = 0%,  reduce = 0%
2022-03-14 21:20:00,259 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.74 se
c
2022-03-14 21:20:08,901 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 1.83
sec
MapReduce Total cumulative CPU time: 1 seconds 830 msec
Ended Job = job_1644894610889_0016
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.83 sec   HDFS Read: 7186 HD
FS Write: 118 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 830 msec
OK
1       Akshata IT      2018    50000.0 USA
4       Shivani SC      2015    45000.0 USA
3       Priti   HR      2012    34000.0 UK
2       Sarita  Sales   2022    23000.0 UK
Time taken: 28.749 seconds, Fetched: 4 row(s)
```

53. Instead of order by if we have sort by,



```
hive> select * from empgrp sort by Salary desc;
Query ID = cloudera_20220314212020_0570cc84-f98b-4239-99cf-dd239654abb6
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_1644894610889_0017, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1644894610889_0017/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1644894610889_0017
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-03-14 21:20:49,709 Stage-1 map = 0%,  reduce = 0%
2022-03-14 21:20:57,245 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.74 se
c
2022-03-14 21:21:05,800 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 1.81
sec
MapReduce Total cumulative CPU time: 1 seconds 810 msec
Ended Job = job_1644894610889_0017
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.81 sec   HDFS Read: 7186 HD
FS Write: 118 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 810 msec
OK
1       Akshata IT      2018    50000.0 USA
4       Shivani SC      2015    45000.0 USA
3       Priti   HR      2012    34000.0 UK
2       Sarita  Sales   2022    23000.0 UK
Time taken: 27.599 seconds, Fetched: 4 row(s)
hive> 
```