**Important Instructions:**

* Please read the document thoroughly before you code.
* Please do not change the Business Requirements.

**Coverage:**

1. HDFS
2. Map Reduce
3. HIVE
4. PIG Script

For all the hands-on, please consider the following,

**UNIX User name: training**

**Home Directory: /home/training**

**HDFS**

**Requirement 1:**

You are supposed to create a directory name “samples” in the home directory and create two files named “emp” and “dept” as follows,

**Emp:**

EMPNO,ENAME,JOB,SAL,DEPTNO

7369,SMITH,CLERK,800,20

7499,ALLEN,SALESMAN,1600,30

7521,WARD,SALESMAN,1250,30

7566,JONES,MANAGER,2975,20

7654,MARTIN,SALESMAN,1400,30

7698,BLAKE,MANAGER,2850,30

7782,CLARK,MANAGER,2450,10

7788,SCOTT,ANALYST,3000,20

7839,KING,PRESIDENT,5000,10

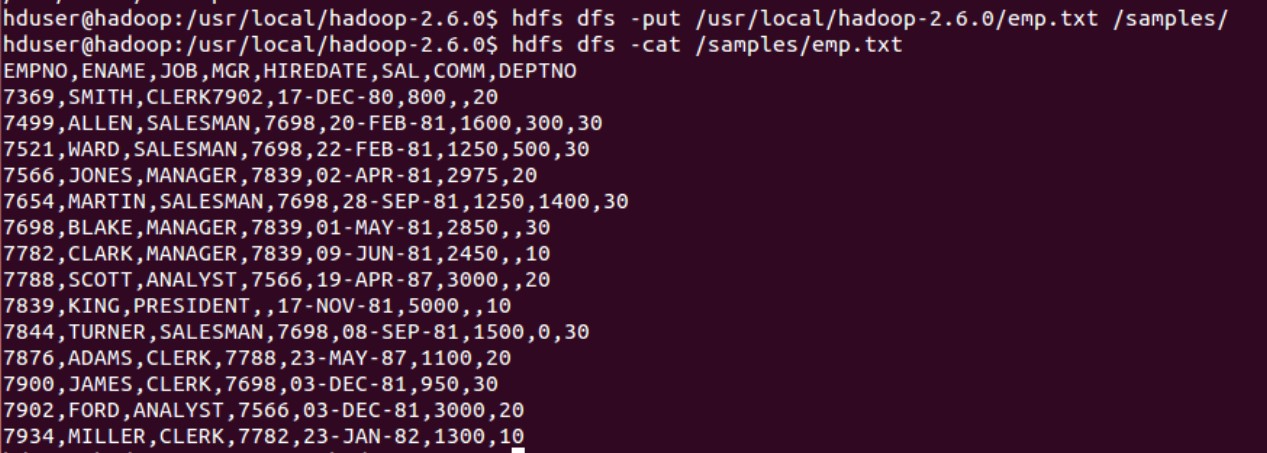
7844,TURNER,SALESMAN,1500,30

7876,ADAMS,CLERK,1100,20

7900,JAMES,CLERK,950,30

7902,FORD,ANALYST,3000,20

7934,MILLER,CLERK,1300,10



**Dept :**

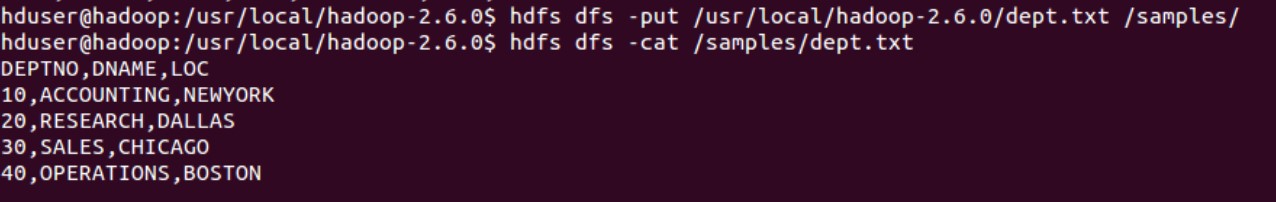
DEPTNO,DNAME,LOC

10,ACCOUNTING,NEWYORK

20,RESEARCH,DALLAS

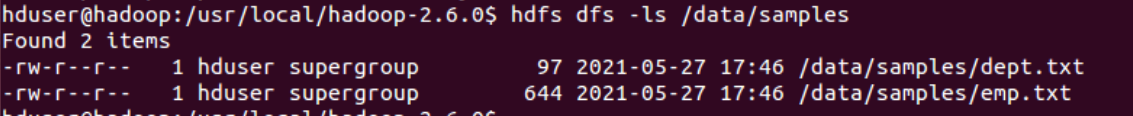
30,SALES,CHICAGO

40,OPERATIONS,BOSTON

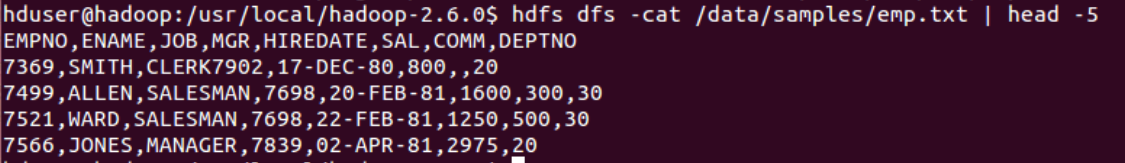


Now create a directory named “data” in HDFS. Copy the “samples” directory into HDFS. Now write commands

1. to see the content of the samples directory in HDFS (hdfs dfs -ls /samples)



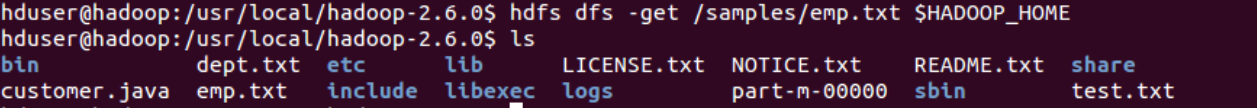
1. to see the first 5 employees from the “emp” file located in HDFS (hdfs dfs -ls /samples/emp | head -5)



1. to remove the “dept “file located in HDFS (hdfs dfs -rm /samples/dept)



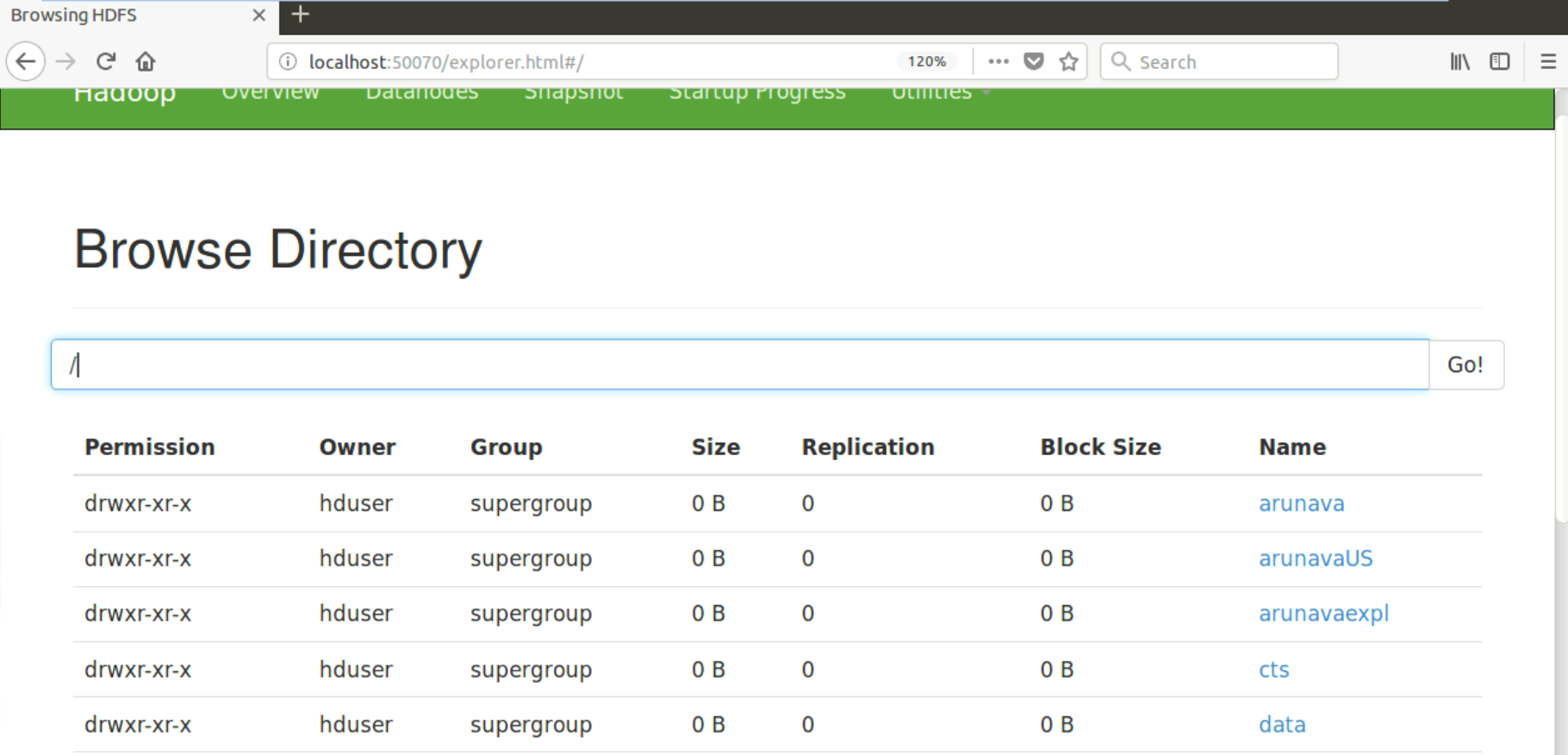
1. to copy the “emp” file from HDFS to the home directory in Unix (hdfs dfs -copyToLocal /samples/emp /home/hduser/



1. Open the browser, type in the following URL

<http://localhost:50070>

And browse the file system



**Map Reduce**

**Requirement 2:**

We asked few Questions and the response of the Batch Representative is

Q1 Name the Creators in the class

Ans : Sasi, Sreejith, Vijay

Q2 Name few who solves all activities

Ans : Dharan, Varsha, Gowtham

Q3 Who are quick learners in the class

Ans: Sasi, Sreejith, Vijay

Q4: Who takes more effort to solve all activities?

Ans: Pavan, Rajesh, Surya, Aasik

Q5. Who come by cab to office?

Ans: Varsha, Aasik, Surya

Q6: Who all have 100% Attendance in the class?

Ans: Sreejith, Varsha, Pavan

Problem Statement: The above Q & A should be used as the input and Need to count the number of occurrence of each name

Stage 1: Mapper

Task 1: Splitting the input to get the Key-Values

Key Values

---------------- ------------------------------

Creators : Sasi, Sreejith, Vijay

Developers Dharan, Varsha, Gowtham

Quick Learners : Sasi, Sreejith, Vijay

Warriors : Pavan, Rajesh, Surya, Aasik

Travellers : Varsha, Aasik, Surya

MrPunctual : Sreejith, Varsha, Pavan

Task 2: Mapping - Each key would be mapped to a Value and count would be 1

Creators : Sasi 1

Creators : Sreejith 1

Creators : Vijay 1

Developers : Dharan 1

Developers : Varsha 1

Developers : Gowtham 1

Quick Learners : Sasi 1

Quick Learners : Sreejith 1

Quick Learners : Vijay 1

Warriors : Pavan 1

Warriors : Rajesh 1

Warriors : Surya 1

Warriors : Aasik 1

Travellers : Varsha 1

Travellers : Aasik 1

Travellers : Surya 1

MrPunctual : Sreejith 1

MrPunctual : Varsha 1

MrPunctual : Pavan 1

Stage 2: Reducer

Task 1: Shuffling so that values occur as a group

Creators: Sasi 1

Quick Learners: Sasi 1

Creators: Sreejith 1

Quick Learners: Sreejith 1

MrPunctual: Sreejith 1

Creators: Vijay 1

Quick Learners: Vijay 1

Developers: Dharan 1

Developers: Varsha 1

Travellers: Varsha 1

MrPunctual: Varsha 1

Developers : Gowtham 1

Warriors : Pavan 1

MrPunctual : Pavan 1

Warriors : Rajesh 1

Warriors : Surya 1

Travellers : Surya 1

Warriors : Aasik 1

Travellers : Aasik 1

Now we can start with the following code to perform the Stage 1 - Task 1, which is to construct a Map with Key-Value pairs.

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.util.\*;

public class FileRead {

public static void main ( String[] args ) throws FileNotFoundException

{

Scanner scanner = new Scanner(new FileReader("D:/neuron.txt"));

Map<String, ArrayList> myMap = new HashMap<String, ArrayList>();

String line;

while (scanner.hasNextLine())

{

line = scanner.nextLine();

if (!line.isEmpty())

{

String[] columns = line.split("=");

String k = columns[0];

String[] v = columns[1].split(" ");

System.out.println("Key : " + k);

ArrayList al = new ArrayList<String>();

for ( int i=0; i<v.length; i++)

{

System.out.println("Value : " + v[i]);

al.add(v[i]);

}

myMap.put(k, al);

}

}

System.out.println ( myMap );

}

}

**Complete the Code to implement Map-Reduce by writing a Driver class and configure your Mapper and Reducer. Specify the input and output directories. Please make sure you are using a unique non-existent output directory.**

**HIVE**

**Requirement 3:**

Create a Database named MyDB in HIVE and create a table named "students" with the following columns

regno NUMERIC 4

sname VARCHAR 20

mark1 NUMERIC 3

mark2 NUMERIC 3

mark3 NUMERIC 3

We have a file named STUDENT.TXT in the samples directory of the Home Directory ( /home/training/samples) with following data.

1000,'RAJ',56,76,91

1001,'AMIT',96,86,91

1002,'DINESH',82,81,84

1003,'MAHESH',77,78,79

1004,'DAVID',86,86,81

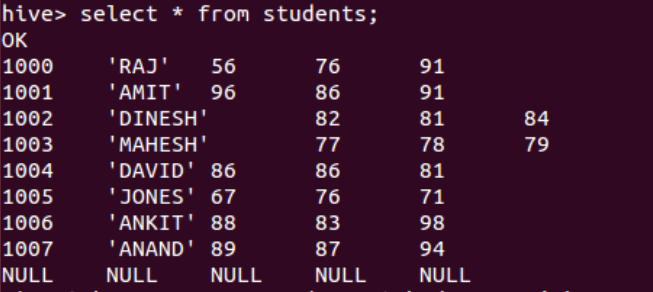
1005,'JONES',67,76,71

1006,'ANKIT',88,83,98

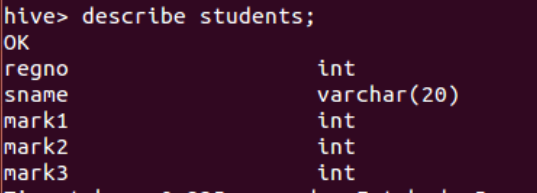
1007,'ANAND',89,87,94

You are supposed to write a command to load the data from student.txt which is in the local file system.

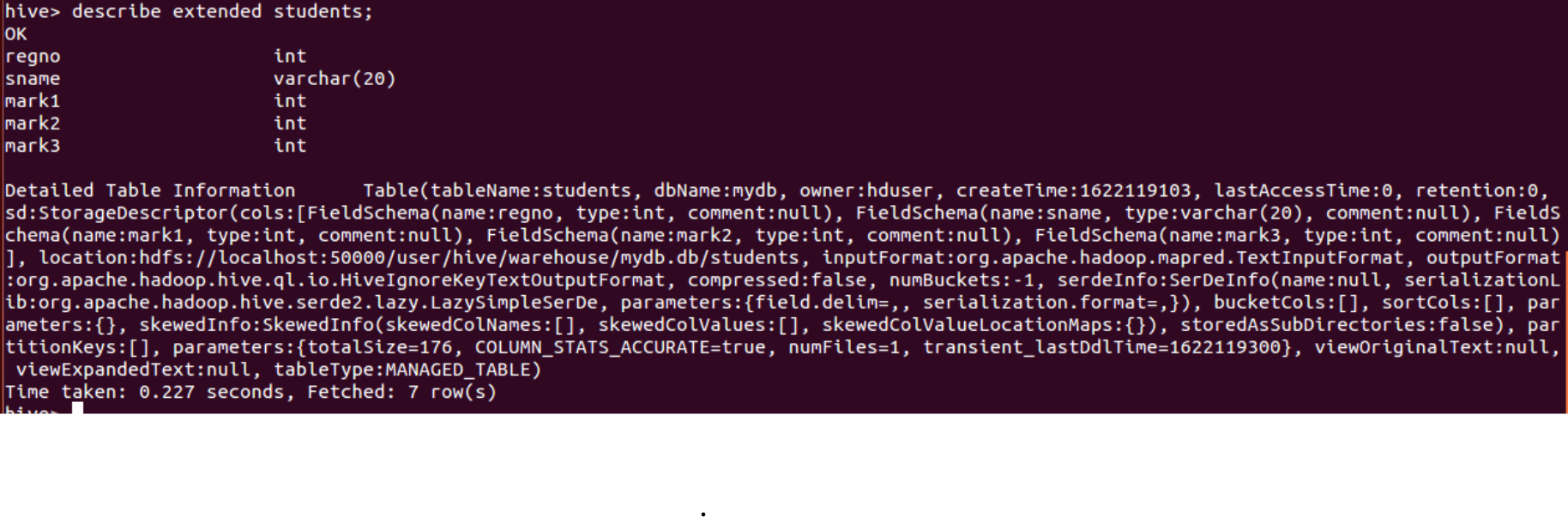
1. Write a command to view the contents of the table students on the console.



1. Write a command to view the schema of the Table



1. Write a command to view the size of the table and other information.



**Requirement 4:**

Create a HIVE table named "EMP" with the following columns

EMPNO NUMERIC 4

ENAME VARCHAR 20

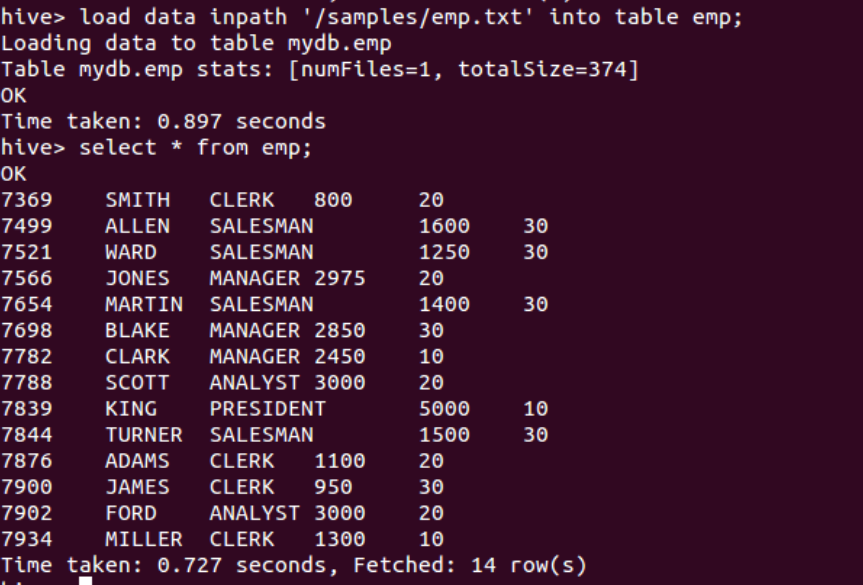
JOB VARCHAR 20

SAL NUMERIC 6

DEPTNO NUMERIC 2

We have a “emp” file available in HDFS (We have created this file in Hands-on 1).

Write a command to load the data from “emp” file in HDFS and load into EMP table in Hive.



**Requirement 5:**

We have already created and loaded a table named “emp” in the previous Activity. Now create a Java JDBC program to read the data from the emp table and display all the employees.

The template of the Java JDBC program is given below,

import java.sql.SQLException;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.sql.DriverManager;

public class readEmpHive

{ private static String driverName = "org.apache.hadoop.hive.jdbc.HiveDriver";

public static void main(String[] args) throws SQLException

{

// Register driver and create driver instance

Class.forName(driverName);

// get connection

Connection con = DriverManager.getConnection("jdbc:hive://localhost:10000/MyDB", "", "");

// create statement

Statement stmt = con.createStatement();

// execute statement

Resultset res = stmt.executeQuery( << Place your Query >> );

System.out.println(" Empno \t EName \t Job \t Salary \t Deptno ");

<< Place your code block to display all employee details >>

con.close();

}

}

Save the program in a file named readEmpHive.java. Use the following commands to compile and execute this program.

javac readEmpHive.java

java readEmpHive

**Requirement 6:**

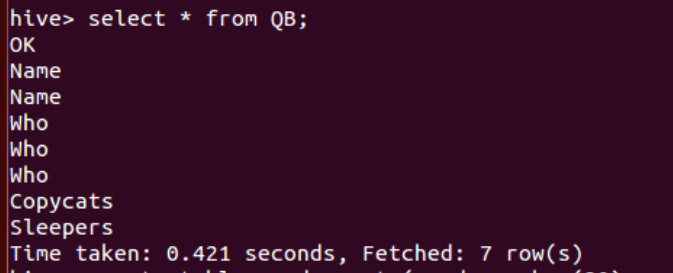
We have a file named “QBank” located in the samples directory of the Home directory and following is the content of the file.



Create a Hive table as follows,

create table QB (words string );

Write a command to load the contents of the QBank file into the table name QB.



Write a Hive command to count the frequency of each word from the QB and load them into a new Hive table named word\_count which has the following two columns

word VARCHAR 20

wcount NUMERIC 2

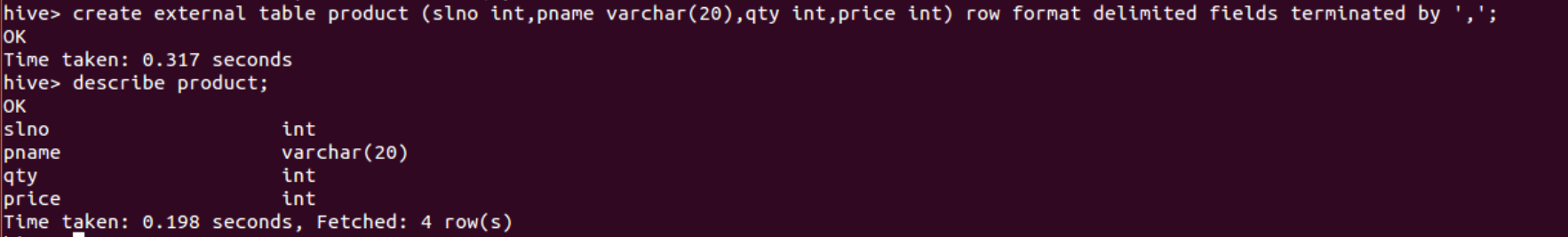


**Requirement 7:**

We have a file named “product” located in the samples directory of the Home directory and following is the content of the file.



Write a Hive command to create an External table named “product”.



**Requirement 8:**

We have the following 2 Activity log files which contains the browsing history of Wikipedia under the samples directory of the home directory

Create a directory named ‘wiki’ on HDFS. Place both the above files into the directory ‘wiki’ on HDFS. Create an External table named ‘wiki\_data’ in the directory named ‘myData’ in the home directory with the following structure.

projectname STRING

pagename STRING

pageview INT

pagesize INT

Now create an internal table named ‘wiki\_en\_views’ that has the total views for every page with the following structure,

pagename STRING

total\_views INT

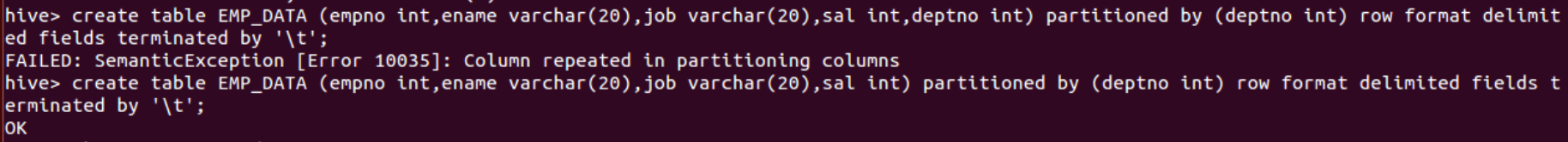
Now load the ‘wiki\_en\_view’ table only for the projectname ‘en’ from the external table ‘wiki’.

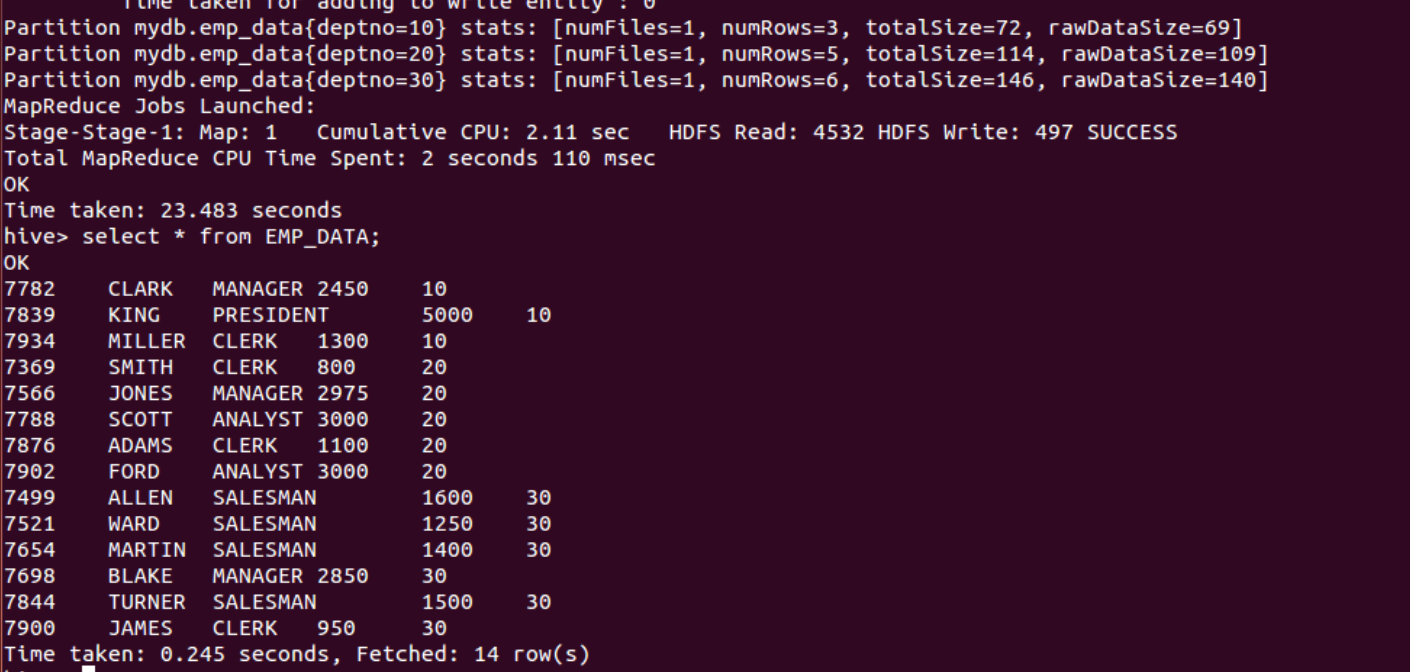
**Requirement 9:**

We have a file named “employee” in the samples directory of the home directory.

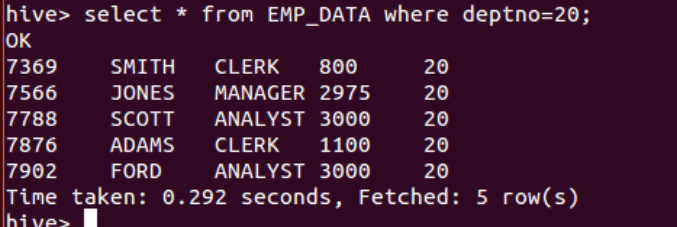


1. Create a Hive table named “EMP\_DATA”
2. Alter the table to add the following 3 partitions
   1. Partition that has the employees of deptno 10
   2. Partition that has the employees of deptno 20
   3. Partition that has the employees of deptno 30





1. Write a query to display the employees from the 2nd partition that contains the employees of deptno 20.



**Requirement 10:**

We have a HIVE table created with the name "EMP" with the following columns

EMPNO NUMERIC (4)

ENAME VARCHAR( 20)

MGRID NUMERIC (4 )

DEPTNO NUMERIC( 4)

SAL NUMERIC( 6)

DEPTNO NUMERIC( 2)

We have another HIVE table created with the name "DEPT" with the following columns

DEPTNO NUMERIC( 2)

DNAME VARCHAR( 20)

LOC VARCHAR( 20)

Requirement is to create

1. A view that has only MANAGERs and a count of employees reporting to him with the following fields ( DEPTNO Field is a common column available in both the tables, that should be used for JOIN )

EMPNO, ENAME, JOB, SAL, DEPTNO, COUNT\_OF\_REPORTING\_EMP

1. A view that has the Employee details along with Managers details with the following structure ( We don’t have MGR\_NAME Column. ENAME should be used in 2 different contexts – 1. EmpName and 2.MgrName )

EMPNO, ENAME, JOB, MGRID, MGR\_NAME, DEPTNO

**PIG Script**

**Requirement 11:**

We have a file named “empDetails” in the samples directory of the home directory which is given below.



Write a PIG Script to count the number of each work in the file empDetails.

**Requirement 12:**

We have the following student data available in Samples directory of the Home directory with the following data,



Write a PIG Script which is

1. to display the total marks of each student
2. to display the total marks scored in every subject.

**Requirement 13:**

We have the EMP data which has empno, ename, job, sal and deptno fields available in HDFS( Emp file is embedded here).



Write a PigScript to display the department wise employee details s for every deptno available in the emp file ( Hint : Use the GROUP Operator )

**Requirement 14:**

We have the EMP data and DEPT data available in HDFS as follows,

Write a PigScript to display the department details and the list of employees working in the department for every deptno ( Hint : Use the COGROUP Operator )

**Requirement 15:**

We have the CUSTOMER data and ORDERS data available in HDFS as follows,

Write a Pig Script to

1. display the customer details along with the order details of that customer.
2. display all the customer details along with the order details of that customer, though the customer has not ordered anything.

**Requirement 16:**

We have the employee details available in 2 different files named EMP1 and EMP2 in HDFS. The requirement is combine all the employee details from EMP1 and EMP2.

Write a Pig Script to combine the employee details from EMP1 and EMP2.

**Requirement 17:**

We have the EMP data available in HDFS as follows,



Write a PigScript to split the data into two, one has the list of employees whose salary is less than 2800 and the second one the list of employees whose salary is greater than or equal to 2800 ( Hint : Use the SPLIT Operator )

**Requirement 18:**

We have the EMP data available in HDFS as follows,



Write a Pig Script to display the TOP FIVE SALARIED employees. (Hint : Use ORDER BY and LIMIT operators)