RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

DEPARTMENTOFARTIFICALINTELLIGENCEAND MACHINE LEARNING

AI19741-BIGDATATECHNOLOGYLABORATORY

SUBHASH P

FINAL YEAR

SEVENTHSEMESTER

2024- 2025

ODDSEMESTER

ListofExperiments

1. InstallationofHadoop(3)
2. FileManagementtasksinHadoop.(3)
    - UploadanddownloadafileinHDFS
    - Copyafilefromsourcetodestination
    - Copytofilefrom/tolocalfilesystemtoHDFS
    - Movefilefromsourcetodestination
    - Removeafile/directoryinHDFS
3. ImplementwordcountprogramusingMapReduce.(3)
4. WeatherReportPOC-MapReduceProgramtoanalyzetime-temperaturestatisticsand generate report with max/min temperature.(3)
5. PigLatinscriptstosort,group,join,project,andfilteryourdata.(6)
6. HiveDatabases,Tables,Views,Functionsand Indexes.(6)
7. ProgramsinSqoop:ExportdatafromHadoopusingSqooptoimportdatatoHive.(6)

Ex. No:1	**InstallationOfHadoopFramework**
Date:	

AIM:

InstallationofHadoopFramework,it‘scomponentsandstudytheHADOOP ecosystem

Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. It is designed to scaleupfromsingleserverstothousandsofmachines,eachofferinglocalcomputationandstorage.

HadoopArchitecture:

TheApacheHadoopframeworkincludesfollowingfourmodules:

HadoopCommon:

Contains Java libraries and utilities needed by other Hadoop modules. These libraries give file system and OS level abstraction and comprise of the essential Java files and scripts that are required to start Hadoop.

Hadoop Distributed File System (HDFS): A distributed file-system that provides highthroughput access to application data on the community machines thus providing very high aggregate bandwidth across the cluster.

Hadoop YARN: A resource-management framework responsible for job scheduling and cluster resource management.

HadoopMapReduce:ThisisaYARN-basedprogrammingmodelforparallelprocessingoflarge data sets.

HadoopInstallationprocedure:

Step 1: Download and install Java

https://www.oracle.com/java/technologies/javase-downloads.html

Step 2: Download Hadoop

https://hadoop.apache.org/releases.html

Step3: SetEnvironmentVariables

Step4: SetupHadoop

oumustconfigureHadoopinthisphasebymodifyingseveralconfigurationfiles.Navigatetothe “etc/hadoop” folder in the Hadoop folder. You must make changes to three files:

core-site.xml

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

hdfs-site.xml

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/hadoop-3.3.1/data/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/hadoop-3.3.1/data/datanode</value>
</property>
</configuration>
```

mapred-site.xml

```
<configuration>
<property>
<name>mapred.job.tracker</name>
<value>localhost:54311</value>
</property>
</configuration>
```

Step5:FormatHadoopNameNode

```
hadoopnamenode–format
```

Step6:StartHadoop

```
start-all.cmd
```

Step7:VerifyHadoopInstallation

```
http://localhost:50070/
```
.

Ex. No:2	**FileManagementtasksinHadoop**
Date:	

AIM:

ToperformvariousfileoperationinHDFS

Step1:AddingFilesandDirectoriestoHDFS

BeforerunningHadoopprogramsondatastoredinHDFS,thedataneedstobeaddedtoHDFS. Let's start by creating a directory and adding a file to it.

1. **Createadirectoryin HDFS:**

   ```
   hadoopfs-mkdir/user/myfile
   ```

   Thiscommand createsanewdirectorynamedmyfileinthe/userdirectoryin HDFS.

2. **AddafiletoHDFS:**

   ```
   hadoopfs-puta.txt
   ```

   Thiscommanduploadsthefilea.txtfromthelocalfilesystemtotherootdirectoryof HDFS.

3. **Addthefiletothenewlycreated directory:**

   ```
   hadoopfs-puta.txt/user/myfile
   ```

   Thiscommanduploadsthefilea.txtfromthelocalfilesystemdirectlyinto the /user/myfile directory in HDFS.

Step2:RetrievingFilesfromHDFS

TocopyfilesfromHDFS backtothelocalfilesystem,usethegetcommand.Here'showto retrieve a.txt:

```
hadoopfs-cata.txt
```

Thiscommanddisplaysthecontentsofthefilea.txtdirectlytotheconsole.Toactuallycopythe file to the local filesystem, you would use:

```
hadoopfs-geta.txt/local/path
```

Replace/local/pathwiththedesiredpathonyourlocalfilesystem.

Step3:DeletingFilesfromHDFS

TodeleteafilefromHDFS,usetherm command.Here'showtodeletea.txt:

```
hadoopfs-rma.txt
```

Thiscommandremovesthefilea.txtfromHDFS.

Output

Thesuccessfulexecution oftheabovecommandswillresultinthefollowing:

- Creationofthe/ user/myfiledirectory in HDFS.
- Addition of a.txt toHDFSandthento /user/myfile
- Retrievalof a.txt fromHDFStothelocalfilesystem.
- Deletionofa.txt fromHDFS.

Ex. No:3	**ImplementwordcountprogramusingMapReduce**
Date:	

AIM:

ToimplementingdistinctwordcountproblemusingMap-Reduce

Thefunctionofthemapperisasfollows:

- Createa IntWritablevariable'one'withvalueas1
- ConverttheinputlineinTexttypeto aString
- Useatokenizertosplitthelineintowords
- IteratethrougheachwordandaformkeyvaluepairsasAssigneachworkfromthetokenizer (of String type) to a Text 'word'
- Formkeyvaluepairsforeachwordas<word,one>andpushittotheoutput collector

ThefunctionofSortand Group:

Afterthis,"aggregation"and"ShufflingandSorting"donebyframework. Then Reducers task these final pair to produce output.

Thefunctionofthereducerisasfollows

- Initializeavariable'sum'as0
- Iteratethrough all thevalueswithrespecttoakeyandsumupallof them
- PushtotheoutputcollectortheKeyandtheobtainedsumasvalue For

Example:

Forthegivensampleinput1 datafile(input1.txt:Hello WorldByeWorld) mapperemits:

<Hello,1>

<World,1>

<Bye,1>

<World,1>

Thesecondinput2datafile(input2.txt:HelloHadoopGoodbyeHadoop)mapperemits:

<Hello,1>

<Hadoop,1>

<Goodbye,1>

<Hadoop,1>

WordCountalsospecifiesacombiner.Hence,theoutputofeachmapispassedthroughthelocal combiner(whichissameastheReducerasperthejobconfiguration)forlocalaggregation,after being sorted on the keys.

Theoutputofthefirst map:

<Hello,1>

<Bye,1>

<World,2>

Theoutputofthesecondmap:

<Hello,1>

<Hadoop,2>

<Goodbye,1>

TheReducerimplementationviathereducemethodjustsumsupthevalues,whicharethe occurence counts for each key (i.e. words in this example).

Thustheoutputofthejobis:

<Goodbye,1>

<Bye,1>

<Hello,2>

<Hadoop,2>

<World,2>

Ex. No:4	**MapReduceProgramforWeatherReport**
Date:	

AIM:

TowriteaMapReduceProgramtoanalyzetime-temperaturestatisticsandgeneratereportwith max/min temperature Weather Report POC.

PROGRAM:

```
// importing Libraries
importjava.io.IOException;
import java.util.Iterator;
importorg.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
importorg.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.Job;
import  org.apache.hadoop.mapreduce.Mapper;
importorg.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;

publicclass MyMaxMin {


        //Mapper
```

```
/*MaxTemperatureMapperclassisstatic
* andextendsMapperabstractclass
* havingfourHadoopgenerics type
* LongWritable,Text,Text,Text.
*/


public static class MaxTemperatureMapper extends
        Mapper<LongWritable,Text,Text,Text>{

    /**
    * @methodmap
    * Thismethod takestheinputasatextdatatype.
    * Nowleavingthefirstfivetokens,it takes
    * 6thtokenistakenastemp_max and
    * 7thtokenistakenastemp_min.Now
    * temp_max>30 andtemp_min <15 are
    * passedtothereducer.
    */

//thedatainourdatasetwith
//thisvalueisinconsistentdata
publicstaticfinalintMISSING=9999;

@Override
    publicvoidmap(LongWritablearg0,TextValue,Contextcontext) throws
            IOException, InterruptedException {

    //Convertthesinglerow(Record)to
    //StringandstoreitinString
    //variablenameline
```

```
            Stringline=Value.toString();

                    //Checkfortheemptyline
                    if(!(line.length()==0)) {

                            //fromcharacter6to14wehave
                            //thedateinourdataset
                            Stringdate=line.substring(6,14);

                            //similarlywehavetakenthemaximum
                            //temperaturefrom39to 45characters
                            floattemp_Max=Float.parseFloat(line.substring(39,45).trim());

                            //similarlywehavetakentheminimum
                            //temperaturefrom47to 53characters

                            floattemp_Min=Float.parseFloat(line.substring(47,53).trim());

                            //ifmaximumtemperatureis
                            //greaterthan30,itisahotday if
                            (temp_Max > 30.0) {

                                    //Hot day
                                    context.write(newText("TheDayisHotDay:"+ date),
                                                                        new
Text(String.valueOf(temp_Max)));
                            }


                            //iftheminimumtemperatureis
                            //less than15,it isacoldday
```

```
                if(temp_Min <15){

                    //Coldday
                    context.write(newText("TheDayisColdDay:"+date), new
                            Text(String.valueOf(temp_Min)));
                }
            }
        }

    }

//Reducer

    /*MaxTemperatureReducerclassisstatic
    and extends Reducer abstract class
    having four Hadoop generics type
    Text,Text,Text,Text.
    */

    publicstatic class MaxTemperatureReducer extends
            Reducer<Text, Text, Text, Text> {

        /**
        * @methodreduce
        * Thismethodtakestheinputaskeyand
        * listofvaluespairfromthemapper,
        * itdoesaggregationbasedonkeys and
        * producesthefinalcontext.
        */

        publicvoidreduce(TextKey,Iterator<Text>Values,Contextcontext)
```

```
                throwsIOException,InterruptedException{
            //puttingallthevaluesin
            //temperaturevariableoftypeString
            Stringtemperature= Values.next().toString();
            context.write(Key, new Text(temperature));
        }

    }

    /**
    * @methodmain
    * Thismethodisusedforsetting
    * alltheconfigurationproperties.
    * Itactsasadriverformap-reduce
    * code.
    */

    publicstaticvoid main(String[]args)throwsException {

        //readsthedefault configurationof the
        //clusterfromtheconfigurationXMLfiles
        Configurationconf=newConfiguration();

        //Initializingthejobwith the
        //defaultconfigurationofthecluster
        Jobjob=newJob(conf,"weatherexample");

        // Assigning the driver class name
        job.setJarByClass(MyMaxMin.class);

        //Keytypecomingoutofmapper
```

```
job.setMapOutputKeyClass(Text.class);

// value type coming out of mapper
job.setMapOutputValueClass(Text.class);

// Defining the mapper class name
job.setMapperClass(MaxTemperatureMapper.class);

// Defining the reducer class name
job.setReducerClass(MaxTemperatureReducer.class);

//DefininginputFormat classwhichis
//responsibletoparsethedataset
// into a key value pair
job.setInputFormatClass(TextInputFormat.class);

//DefiningoutputFormatclasswhichis
//responsibletoparsethedataset
// into a key value pair
job.setOutputFormatClass(TextOutputFormat.class);

//settingthesecondargument
//asapathinapath variable
PathOutputPath=newPath(args[1]);

//Configuringtheinputpath
// from the filesystem into the job
FileInputFormat.addInputPath(job, new Path(args[0]));

//Configuringtheoutputpathfrom
//thefilesystemintothejob
```

```
        FileOutputFormat.setOutputPath(job,newPath(args[1]));

        //deletingthecontextpathautomatically
        //fromhdfssothatwedon'thave
        // to delete it explicitly
        OutputPath.getFileSystem(conf).delete(OutputPath);

        //exitingthejob onlyifthe
        // flag value becomes false
        System.exit(job.waitForCompletion(true)?0:1);

    }
}
```

Ex.No:5.a	**PigLatinscriptstosort,group**
Date:	

AIM:

Towriteascript forsortingandgroupingofdata.

Studentdata:

Assumewehaveafile**student_data.txt**inHDFSwiththefollowingcontent.

001,Rajiv,Reddy,21,9848022337,Hyderabad

002,siddarth,Battacharya,22,9848022338,Kolkata

003,Rajesh,Khanna,22,9848022339,Delhi

004,Preethi,Agarwal,21,9848022330,Pune

005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar

006,Archana,Mishra,23,9848022335,Chennai

007,Komal,Nayak,24,9848022334,trivendram

008,Bharathi,Nambiayar,24,9848022333,Chennai

Step1:

Loadand storethe studentdatainHDFS .

```
grunt>student=LOAD'hdfs://localhost:9000/pig_data/student_data.txt'USING
  PigStorage(',')
  as ( id:int, firstname:chararray, lastname:chararray, phone:chararray,
  city:chararray );
```

The**ORDERBY**operatorisusedtodisplaythecontentsofarelationinasortedorderbasedononeormorefields. grunt>

Relation_name2 = ORDER Relatin_name1 BY (ASC|DESC);

Verifytherelation**order_by_data**usingthe**DUMP**operatorasshownbelow.

```
grunt>Dumporder_by_data;
```

Output

Itwillproducethefollowingoutput,displayingthecontentsoftherelation**order_by_data**.

```
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
```

```
(7,Komal,Nayak,24,9848022334,trivendram)
(6,Archana,Mishra,23,9848022335,Chennai)
(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar)


(3,Rajesh,Khanna,22,9848022339,Delhi)
(2,siddarth,Battacharya,22,9848022338,Kolkata)
(4,Preethi,Agarwal,21,9848022330,Pune)
(1,Rajiv,Reddy,21,9848022337,Hyderabad)
```

The**GROUP**operatorisusedtogroupthedatainoneormorerelations.Itcollectsthedatahavingthesamekey. Given below is the syntax of the **group** operator.

Now,letusgrouptherecords/tuplesintherelationbyageasshownbelow. grunt>
group_data = GROUP student_details by age;

Verifytherelation**group_data**usingthe**DUMP**operatorasshownbelow. grunt>

Dump group_data;

Output:

```
(21,{(4,Preethi,Agarwal,21,9848022330,Pune),(1,Rajiv,Reddy,21,9848022337,Hydera bad)})
(22,{(3,Rajesh,Khanna,22,9848022339,Delhi),(2,siddarth,Battacharya,22,984802233 8,Kolkata)})
(23,{(6,Archana,Mishra,23,9848022335,Chennai),(5,Trupthi,Mohanthy,23,9848022336 ,Bhuwaneshwar)})
(24,{(8,Bharathi,Nambiayar,24,9848022333,Chennai),(7,Komal,Nayak,24,9848022334, trivendram)})
```

Ex.No:5.b	**PigLatinscriptstoproject,andfilteryourdata**
Date:	

AIM:

Towriteascripttoperformingprojectand filtering.

The**FILTER**operatorisusedtoselecttherequiredtuplesfromarelationbasedonacondition.

Givenbelowisthesyntaxofthe**FILTER**operator.

grunt> Relation2_name= FILTERRelation1_nameBY(condition);

student_details.txt

```
001,Rajiv,Reddy,21,9848022337,Hyderabad
002,siddarth,Battacharya,22,9848022338,Kolkata
003,Rajesh,Khanna,22,9848022339,Delhi
004,Preethi,Agarwal,21,9848022330,Pune
005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar
006,Archana,Mishra,23,9848022335,Chennai
007,Komal,Nayak,24,9848022334,trivendram
008,Bharathi,Nambiayar,24,9848022333,Chennai
```

Andwehaveloadedthisfileinto Pigwiththerelationname**student_details**asshownbelow.

```
grunt>student_details=LOAD'hdfs://localhost:9000/pig_data/student_details.txt'USINGPigStorage(',') as
   (id:int, firstname:chararray, lastname:chararray, age:int, phone:chararray, city:chararray);
```

LetusnowusetheFilteroperatortogetthedetailsofthestudentswhobelongtothecityChennai.

```
filter_data=FILTERstudent_detailsBYcity=='Chennai';
```

Verification

Verifytherelation**filter_data**usingthe**DUMP**operatorasshownbelow.

```
grunt>Dumpfilter_data;
```

Output

Itwillproducethefollowingoutput,displayingthecontentsoftherelation**filter_data**asfollows.

```
(6,Archana,Mishra,23,9848022335,Chennai)
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
```

Ex.No:6.a	**HiveDatabases->Tables,Views**
Date:	

AIM:

TowriteascripttoHiveDatabases->Tables,Views,

CreateDatabaseStatement

CreateDatabaseisastatementusedtocreateadatabaseinHive.AdatabaseinHiveisa**namespace**oracollection of tables. The **syntax** for this statement is as follows:

```
CREATEDATABASE|SCHEMA[IFNOTEXISTS] <databasename>
```

Here, IF NOT EXISTS is an optional clause, which notifies the user that a database with the same name already exists.WecanuseSCHEMAinplaceofDATABASEinthiscommand.Thefollowingqueryisexecutedtocreatea database named **userdb**:

```
hive>CREATEDATABASE[IFNOT EXISTS]userdb;
```

or

```
hive>CREATESCHEMAuserdb;
```

Thefollowingqueryisusedtoverifyadatabaseslist:

```
hive>SHOWDATABASES;
default
userdb
```

JDBCProgram

TheJDBCprogramtocreateadatabaseisgivenbelow.

```
importjava.sql.SQLException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
importjava.sql.DriverManager;

publicclassHiveCreateDb{
  privatestaticStringdriverName="org.apache.hadoop.hive.jdbc.HiveDriver";

  publicstaticvoidmain(String[]args)throwsSQLException{
    //Registerdriverandcreatedriverinstance

    Class.forName(driverName);
    //getconnection

    Connectioncon=DriverManager.getConnection("jdbc:hive://localhost:10000/default","","");
    Statement stmt = con.createStatement();

    stmt.executeQuery("CREATE DATABASE userdb");
    System.out.println("Database userdb created successfully.");

    con.close();
  }
}
```

SavetheprograminafilenamedHiveCreateDb.java.Thefollowingcommandsareusedtocompileandexecutethis program.

```
$javacHiveCreateDb.java
$javaHiveCreateDb
```

Output:

Databaseuserdbcreatedsuccessfully.

CreatingaView

YoucancreateaviewatthetimeofexecutingaSELECTstatement.Thesyntaxisasfollows:

```
CREATEVIEW[IFNOTEXISTS]view_name[(column_name[COMMENTcolumn_comment],...)] [COMMENT
table_comment]
ASSELECT...
```

Example

Let us take an example for view. Assume employee table as given below, with the fields Id, Name, Salary, Designation,andDept.GenerateaquerytoretrievetheemployeedetailswhoearnasalaryofmorethanRs30000. We store the result in a view named **emp_30000.**

```
+------+--------------+------------+------------------+--------+
|ID|Name            |Salary      |Designation       |Dept|
+------+--------------+------------+------------------+--------+
|1201|Gopal          |45000       |Technicalmanager|TP       |
|1202|Manisha        |45000       |Proofreader       |PR     |
|1203|Masthanvali|40000           |Technicalwriter|TP        |
|1204|Krian          |40000       |Hr Admin          |HR     |
|1205|Kranthi        |30000       |Op Admin          |Admin|
+------+--------------+------------+------------------+--------+
```

Thefollowingqueryretrievestheemployeedetailsusingtheabovescenario:

```
hive>CREATEVIEWemp_30000AS SELECT
* FROM employee
WHEREsalary>30000;
```

DroppingaView

Usethefollowingsyntaxtodropaview:

```
DROPVIEWview_name
```

Thefollowingquerydropsaviewnamedasemp_30000: hive>

```
DROP VIEW emp_30000;
```

Ex.No:6.b	**HiveDatabases->FunctionsandIndexes**
Date:	

AIM:

TowriteascripttoHiveDatabases->**FunctionsandIndexes**

Thefollowingqueriesdemonstratesomebuilt-infunctions:

round()function

```
hive>SELECTround(2.6) fromtemp;
```

Onsuccessfulexecutionofquery,yougettoseethefollowingresponse:

```
3.0
```

floor() function

```
hive>SELECTfloor(2.6)fromtemp;
```

Onsuccessfulexecutionofthequery,yougettoseethefollowingresponse:

```
2.0
```

ceil()function

```
hive>SELECTceil(2.6)fromtemp;
```

Onsuccessfulexecutionofthequery,yougettoseethefollowingresponse:

```
3.0
```

AggregateFunctions

Hivesupportsthefollowingbuilt-in**aggregatefunctions**.TheusageofthesefunctionsisassameastheSQL aggregate functions.

ReturnType	**Signature**	**Description**
BIGINT	count(*), count(expr),	count(*)-Returnsthetotalnumberofretrieved rows.

DOUBLE	sum(col), sum(DISTINCTcol)	Itreturnsthesumoftheelementsinthegroupor the sum of the distinct values of the column in the group.
DOUBLE	avg(col), avg(DISTINCTcol)	It returns the average of the elements in the grouportheaverageofthedistinctvaluesofthe column in the group.
DOUBLE	min(col)	Itreturnstheminimumvalueofthecolumnin the group.
DOUBLE	max(col)	Itreturnsthemaximumvalueofthecolumnin the group.

CreatinganIndex

AnIndexisnothingbutapointeronaparticularcolumnofatable.Creatinganindexmeanscreatingapointerona particular column of a table. Its syntax is as follows:

```
CREATEINDEXindex_name
ONTABLEbase_table_name (col_name,...) AS
'index.handler.class.name'
[WITHDEFERREDREBUILD]
[IDXPROPERTIES (property_name=property_value, ...)]
[IN TABLE index_table_name]
[PARTITIONEDBY(col_name,...)] [
  [ROWFORMAT...]STOREDAS...
  |STOREDBY...
]
[LOCATIONhdfs_path]
[TBLPROPERTIES(...)]
```

Example

Letustakeanexampleforindex.UsethesameemployeetablethatwehaveusedearlierwiththefieldsId,Name, Salary, Designation, and Dept. Create an index named index_salary on the salarycolumn of the employee table.

Thefollowingquerycreatesanindex:

```
hive>CREATEINDEXinedx_salaryONTABLEemployee(salary)
AS'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler';
```

Itisapointertothesalarycolumn.Ifthecolumnismodified,thechangesarestoredusinganindexvalue.

DroppinganIndex

Thefollowingsyntaxisusedtodropanindex:

```
DROPINDEX<index_name>ON<table_name>
```

Thefollowingquerydropsanindexnamedindex_salary:

```
hive> DROP INDEX index_salary ON employee;
```

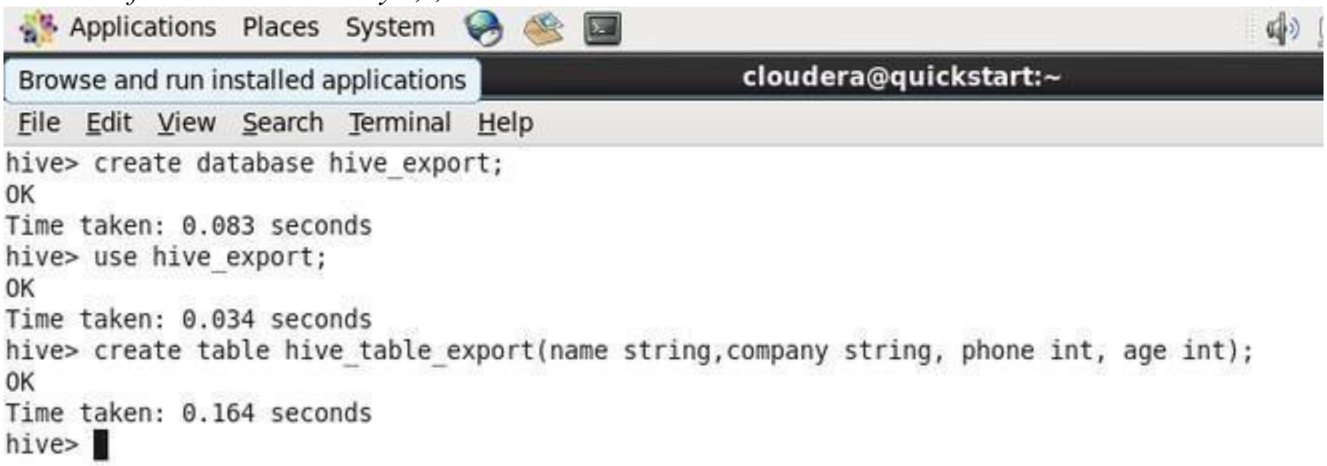
Ex. No:7	ExportdatafromHadoopusingSqoop
Date:	

AIM:

ToexportdatafromHadoopusingSqooptoimportdatatoHive**.**

ToexportdataintoMySQLfromHDFS,performthefollowingsteps:

Step1: Createadatabaseand tableinthe hive.
createtablehive_table_export(namestring,companystring,phoneint,ageint)rowformat delimited fields terminated by ‘,’;

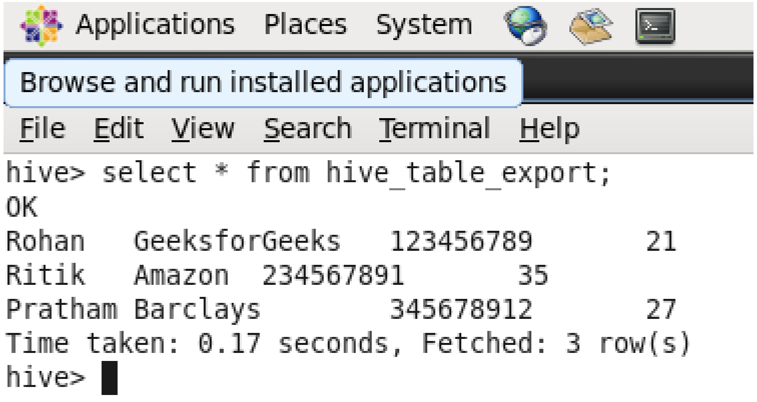

Applications Places System
Browse and run installed applications
cloudera@quickstart:~
File Edit View Search Terminal Help

```
hive> create database hive_export;
OK
Time taken: 0.083 seconds
hive> use hive_export;
OK
Time taken: 0.034 seconds
hive> create table hive_table_export(name string,company string, phone int, age int);
OK
Time taken: 0.164 seconds
hive> 
```

HiveDatabase:hive_exportandHiveTable:hive_table_export

Step2: Insertdataintothehivetable.

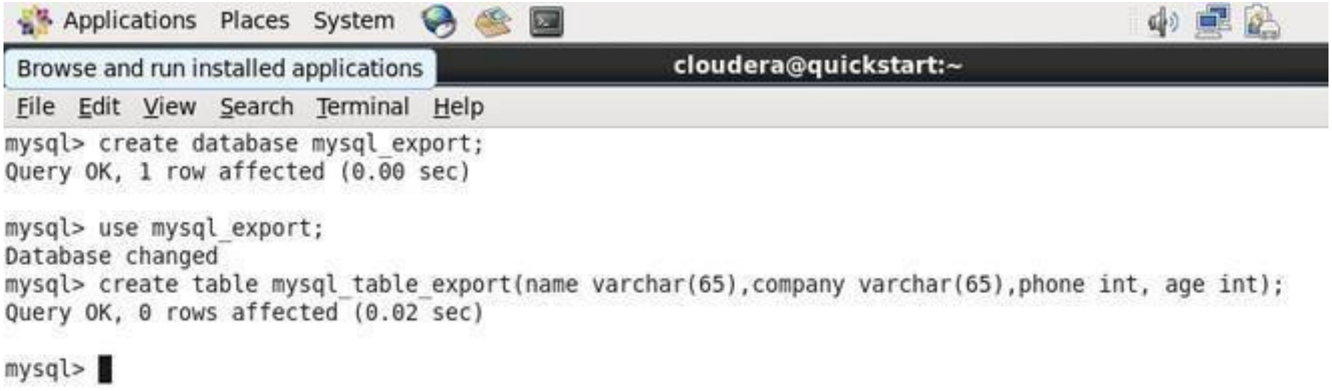
insertintohive_table_exportvalues("Ritik","Amazon",234567891,35);


Applications Places System
Browse and run installed applications
File Edit View Search Terminal Help

```
hive> select * from hive_table_export;
OK
Rohan   GeeksforGeeks   123456789       21
Ritik   Amazon  234567891       35
Pratham Barclays        345678912       27
Time taken: 0.17 seconds, Fetched: 3 row(s)
hive> 
```

Data inHivetable

Step3: Createadatabaseand tableinMySQLinwhichdatashouldbeexported.



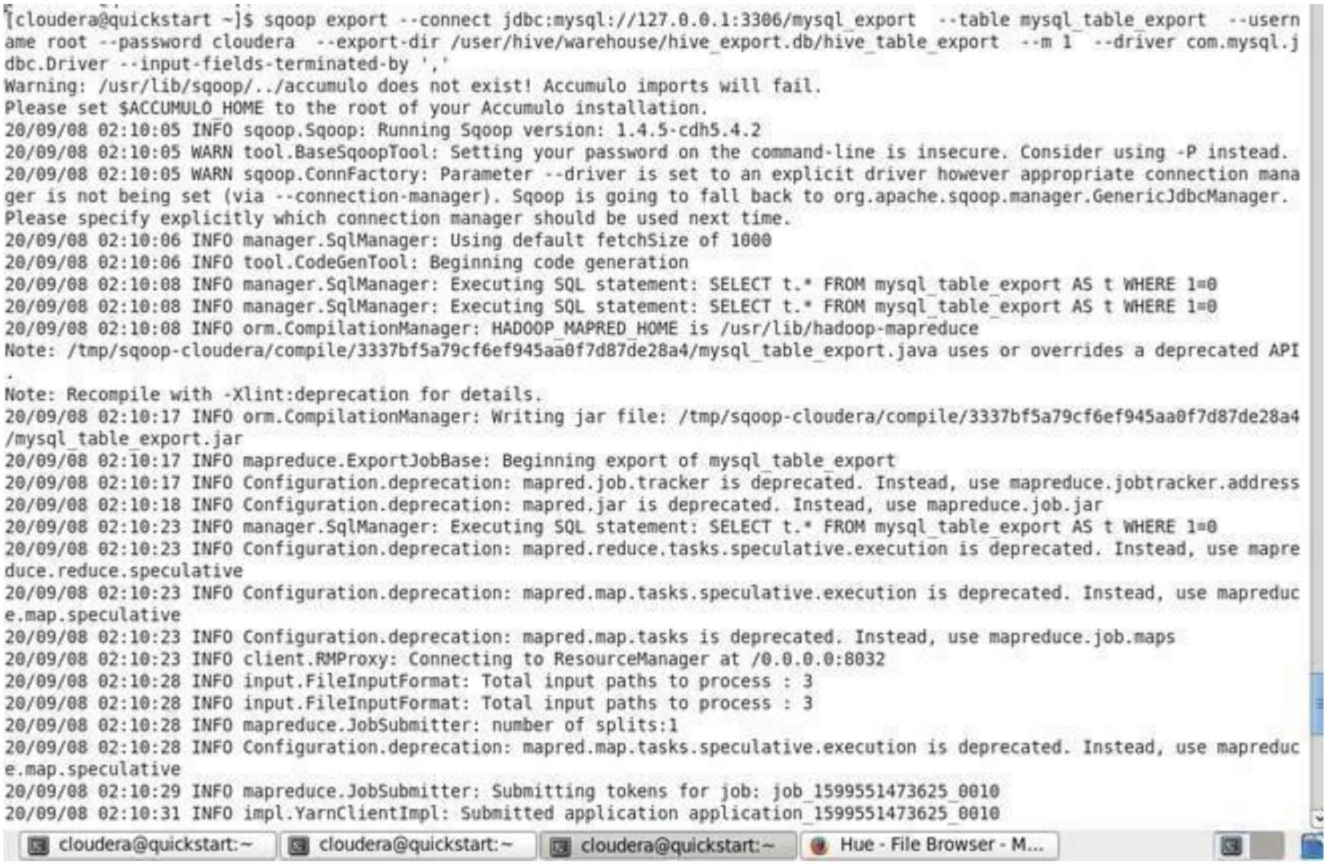
Applications Places System
Browse and run installed applications
cloudera@quickstart:~
File Edit View Search Terminal Help

```
mysql> create database mysql_export;
Query OK, 1 row affected (0.00 sec)

mysql> use mysql_export;
Database changed
mysql> create table mysql_table_export(name varchar(65),company varchar(65),phone int, age int);
Query OK, 0 rows affected (0.02 sec)

mysql> 
```

MySQLDatabase:mysql_exportandMySQLTable:mysql_table_export

Step4: RunthefollowingcommandonHadoop.

```
sqoop export --connect \
jdbc:mysql://127.0.0.1:3306/database_name_in_mysql \
 --tabletable_name_in_mysql\
 --usernameroot--passwordcloudera\
 --export-dir/user/hive/warehouse/hive_database_name.db/table_name_in_hive\
 --m1\
 --drivercom.mysql.jdbc.Driver
 --input-fields-terminated-by','
```



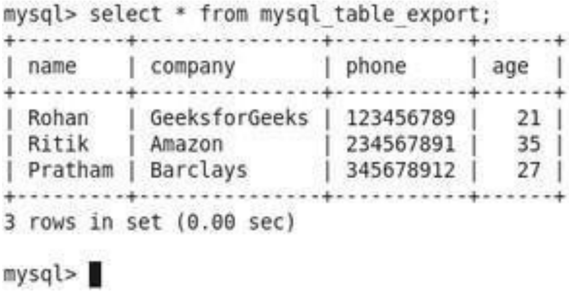
```
[cloudera@quickstart ~]$ sqoop export --connect jdbc:mysql://127.0.0.1:3306/mysql_export  --table mysql_table_export  --usern
ame root --password cloudera  --export-dir /user/hive/warehouse/hive_export.db/hive_table_export  --m 1  --driver com.mysql.j
dbc.Driver --input-fields-terminated-by ','
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/09/08 02:10:05 INFO sqoop.Sqoop: Running Sqoop version: 1.4.5-cdh5.4.2
20/09/08 02:10:05 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
20/09/08 02:10:05 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection mana
ger is not being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager.
Please specify explicitly which connection manager should be used next time.
20/09/08 02:10:06 INFO manager.SqlManager: Using default fetchSize of 1000
20/09/08 02:10:06 INFO tool.CodeGenTool: Beginning code generation
20/09/08 02:10:08 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM mysql_table_export AS t WHERE 1=0
20/09/08 02:10:08 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM mysql_table_export AS t WHERE 1=0
20/09/08 02:10:08 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/3337bf5a79cf6ef945aa0f7d87de28a4/mysql_table_export.java uses or overrides a deprecated API
.
Note: Recompile with -Xlint:deprecation for details.
20/09/08 02:10:17 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/3337bf5a79cf6ef945aa0f7d87de28a4
/mysql_table_export.jar
20/09/08 02:10:17 INFO mapreduce.ExportJobBase: Beginning export of mysql_table_export
20/09/08 02:10:17 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
20/09/08 02:10:18 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
20/09/08 02:10:23 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM mysql_table_export AS t WHERE 1=0
20/09/08 02:10:23 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated. Instead, use mapre
duce.reduce.speculative
20/09/08 02:10:23 INFO Configuration.deprecation: mapred.map.tasks.speculative.execution is deprecated. Instead, use mapreduc
e.map.speculative
20/09/08 02:10:23 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
20/09/08 02:10:23 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/09/08 02:10:28 INFO input.FileInputFormat: Total input paths to process : 3
20/09/08 02:10:28 INFO input.FileInputFormat: Total input paths to process : 3
20/09/08 02:10:28 INFO mapreduce.JobSubmitter: number of splits:1
20/09/08 02:10:28 INFO Configuration.deprecation: mapred.map.tasks.speculative.execution is deprecated. Instead, use mapreduc
e.map.speculative
20/09/08 02:10:29 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1599551473625_0010
20/09/08 02:10:31 INFO impl.YarnClientImpl: Submitted application application_1599551473625_0010
```

SQOOPcommandtoexportdata

Intheabovecodefollowingthingsshouldbenoted.

- **127.0.0.1**isthelocalhostIPaddress.
- **3306**istheportnumberforMySQL.
- Inthecaseofexportingdata,theentirepath tothe tableshouldbespecified
- **m**isthenumberofmappers

Step5: Check-inMySQLifdataisexportedsuccessfullyornot.



```
mysql> select * from mysql_table_export;
+---------+---------------+-----------+------+
| name    | company       | phone     | age  |
+---------+---------------+-----------+------+
| Rohan   | GeeksforGeeks | 123456789 |   21 |
| Ritik   | Amazon        | 234567891 |   35 |
| Pratham | Barclays      | 345678912 |   27 |
+---------+---------------+-----------+------+
3 rows in set (0.00 sec)

mysql> █
```