SWAMI KESHVANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT AND GRAMOTHAN, JAIPUR

Hands on Lab Guide(Lab Manual)

Big Data Analytics Lab

IV Year B.Tech. VII SEM

(Course Code: 7IT4-21)

Session 2022-2023

Department of Information Technology

SKIT, JAIPUR

Swami Keshvanand Institute of Technology, Management& Gramothan, Ramnagaria, Jagatpura, Jaipur-302017

Big Data Analytics Lab (7IT4-21)

VERSION 1.0

	AUTHOR/ OWNER	REVIEWED BY	APPROVED BY
NAME	Mr. Praveen Kumar Yadav		Prof. (Dr.) Anil Choudhary
DESIGNATION	Assistant Professor		HOD IT
SIGNATURE			

LAB RULES

Responsibility of Users:

Users are expected to follow some obvious rules of conduct:

Always:

- Enter the lab on time and leave at proper time.
- Wait for the previous class to leave before the next class enters.
- Keep the bag outside in the respective racks.
- Utilize lab hours in the corresponding.
- Turn off the machine before leaving the lab unless a member of lab staff has specifically told you not to do so.
- Leave the labs at least as nice as you found them.
- If you notice a problem with a piece of equipment (e.g. a computer doesn't respond) or the room in general (e.g. cooling, heating, lighting) please report it to lab staff immediately. Do not attempt to fix the problem yourself.

- E-Mail: hodit@skit.ac.in, URL: www.skit.ac.in

Never:

- Don't abuse the equipment.
- Do not adjust the heat or air conditioners. If you feel the temperature is not properly set, inform lab staff; wewill attempt to maintain a balance that is healthy for people and machines.
- Do not attempt to reboot a computer. Report problems to lab staff.
- Do not remove or modify any software or file without permission.
- Do not remove printers and machines from the network without being explicitly told to do so by lab staff.
- Don't monopolize equipment. If you're going to be away from your machine for more than 10 or 15 minutes, log out before leaving. This is both for the security of your account, and to ensure that others are able to use thelab resources while you are not.
- Don't use internet, internet chat of any kind in your regular lab schedule.
- Do not download or upload of MP3, JPG or MPEG files.
- No games are allowed in the lab sessions.
- No hardware including USB drives can be connected or disconnected in the labs without prior permission ofthe lab in-charge.
- No food or drink is allowed in the lab or near any of the equipment. Aside from the fact that it leaves a mess and attracts pests, spilling anything on a keyboard or other piece of computer equipment could cause permanent, irreparable, and costly damage. (and in fact *has*) If you need to eat or drink, take a break and do so in the canteen.
- Don't bring any external material in the lab, except your lab record, copy and books.
- Don't bring the mobile phones in the lab. If necessary then keep them in silence mode.
- Please be considerate of those around you, especially in terms of noise level. While labs are a natural place for conversations of all types, kindly keep the volume turned down.
- **Note:** If you are having problems or questions, please go to either the faculty, lab in-charge or the lab supporting staff. They will help you. We need your full support and cooperation for smooth functioning of the lab.

INSTRUCTIONS BEFORE ENTERING IN THE LAB

- All the students are supposed to prepare the theory regarding the next experiment/ Program.

- Students are supposed to bring their lab records as per their lab schedule.
- Previous experiment/program should be written in the lab record.
- If applicable trace paper/graph paper must be pasted in lab record with proper labeling.
- All the students must follow the instructions, failing which he/she may not be allowed in the lab.

WHILE WORKING IN THE LAB

- Adhere to experimental schedule as instructed by the lab in-charge/faculty.
- Get the previously performed experiment/ program signed by the faculty/ lab in charge.
- Get the output of current experiment/program checked by the faculty/ lab in charge in the lab copy.
- Each student should work on his/her assigned computer at each turn of the lab.
- Take responsibility of valuable accessories.
- Concentrate on the assigned practical and do not play games.
- If anyone is caught red-handed carrying any equipment of the lab, then he/she will have to face seriousconsequences.

Marking Assessment System

Total Marks: 100

Distribution of Marks - 60 (Sessional)

Attendance	File Work	Performance	Viva	Total
10	10	20	20	60

Distribution of Marks - 40 (End Term) Depends on Examiner

File Work	Performance	Viva	Total
10	20	10	40

INDEX

Sr. No.	Topic	Page Number
1	Lab Plan	7
2	Lab Objective & Requirements	8
3	List of Experiments	9
4	Experiment No 1	10-15
5	Experiment No 2	16-20
6	Experiment No 3	21-22
7	Experiment No 4	23-26
8	Experiment No 5	27-33
9	Experiment No 6	34-35
10	Experiment No 7	36-39
11	Experiment No 8	40-65
12	Viva Questions	66

LAB PLAN

Total number of experiment 8

Total number of turns required 10

Number of turns required for

Experiment Number	Turns	Scheduled Day
Experiment 1	1	Monday, Tuesday
Experiment 2	1	Monday, Tuesday
Experiment3	1	Monday, Tuesday
Experiment 4	1	Monday, Tuesday
Experiment 5	1	Monday, Tuesday
Experiment 6	1	Monday, Tuesday
Experiment 7	1	Monday, Tuesday
Experiment 8	1	Monday, Tuesday
Experiment 9	1	Monday, Tuesday

Distribution of Lab Hours:

Attendance	10 minutes
Explanation of features of language	30 minutesExplanation of experiment 60 minutes
Performance of experiment	80 minutes
File Checking	20 minutes
Viva / Quiz / Queries	40 minutes
Total	**240 minutes**

LAB Objectives and Requirements

Objective: Upon successful completion of this course, students should be able to:

- Understand and implement the basics of data structures like Linked list, stack, queue, set and map inJava.
- Demonstrate the knowledge of big data analytics and implement different file management task inHadoop
- Understand Map Reduce Paradigm and develop data applications using variety of systems.
- Analyze and perform different operations on data using Pig Latin scripts.
- Illustrate and apply different operations on relations and databases using Hive.

Software / Hardware Required

a. *Software Requirement: Linux*

b. Hardware Requirement:

- Intel Pentium IV MHz Processor or Higher
- Intel Chipset 810 Motherboard or Higher
- 14" or Higher Color Monitor/LED/LCD/TFT
- Optical Mouse
- Keyboard
- 40 GB HDD or Higher
- 4 GB RAM or Higher

7IT4-21: Big Data Analytics Lab

Credit: 2 **Max. Marks: 100(IA: 60, ETE: 40)**

0L+0T+4P **End Term Exam: 2 Hours**

SN	List of Experiments
1	Implement the following Data structures in Java i) Linked Lists ii) Stacks iii) Queues iv) Set v) Map
2	Perform setting up and Installing Hadoop in its three operating modes: Standalone,Pseudo distributed, Fully distributed.
3	Implement the following file management tasks in Hadoop: • Adding files and directories • Retrieving files • Deleting files Hint: A typical Hadoop workflow creates data files (such as log files) elsewhere and copiesthem into HDFS using one of the above command line utilities.
4	Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.
5	Write a Map Reduce program that mines weather data. Weather sensors collecting data every hour at many locations across the globe gather a large volume of log data, which is a good candidate for analysis with MapReduce, since it is semi structured and record-oriented.
6	Implement Matrix Multiplication with Hadoop Map Reduce
7	Install and Run Pig then write Pig Latin scripts to sort, group, join, project, and filter yourdata.
8	Install and Run Hive then use Hive to create, alter, and drop databases, tables, views,functions, and indexes.
9	Solve some real life big data problems.

Exp. 1 Implement the following Data structures in Java
i) Linked Lists ii) Stacks iii) Queues iv) Set v) Map

Linked Lists:

```
import java.util.*;
public class LinkedListDemo
{
    public static void main(String args[])
    {
        // create a linked list
        LinkedList ll = new LinkedList();
        // add elements to the linked list
        ll.add("F");
        ll.add("B");
        ll.add("D");
        ll.add("E");
        ll.add("C");
        ll.addLast("Z");
        ll.addFirst("A");
        ll.add(1,"A2");
        System.out.println("Original contents of ll: " + ll);
        // remove elements from the linked list ll.remove("F");
        ll.remove(2);
        System.out.println("Contents of ll after deletion: "+
ll);
        // remove first and last elements ll.removeFirst();
        ll.removeLast();
        System.out.println("ll after deleting first and last: "+
ll);
        // get and set a value Object val = ll.get(2);
        ll.set(2,(String) val + " Changed");
System.out.println("ll after change: " + ll);
    }
}
```

Output:

Original contents of ll: [A, A2, F, B, D, E,C, Z]

Contents of ll after deletion: [A, A2, D, E, C, Z]

ll after deleting first and last: [A2, D, E, C]

ll after change: [A2, D, E Changed, C]

Stacks Program:

```
import java.util.*;
public class StackDemo
{
    static void showpush(Stack st, int a)
    {
        st.push(new Integer(a));
        System.out.println("push(" + a + ")");
        System.out.println("stack: " + st);
    }
    static void showpop(Stack st)
    {
        System.out.print("pop -> ");
        Integer a = (Integer) st.pop();
        System.out.println(a);
        System.out.println("stack: " + st);
    }

    public static void main(String args[])
    {
        Stack st = new Stack();
        System.out.println("stack: " + st);
        showpush(st, 42);
        showpush(st, 66);
        showpush(st, 99);

        showpop(st);
        showpop(st);
        showpop(st);
        try
        {
            showpop(st);
        }
        catch (EmptyStackException e)
        {
            System.out.println("empty stack");
        }
    }
}
```

output:

stack: [ ]

push(42)

stack: [42]

push(66)

stack: [42, 66]

push(99)

stack: [42, 66, 99]

pop -> 99

stack: [42, 66]

pop -> 66

stack: [42]

pop -42

stack: [ ]

pop -> empty stack

Queues

// Java program to demonstrate working of Queue interface in Java

```
import java.util.LinkedList;
import java.util.Queue;
public class QueueExample
{
    public static void main(String[] args)
    {
        Queue<Integer> q = new LinkedList<>();
        // Adds elements {0, 1, 2, 3, 4} to queue
        for (int i=0; i<5; i++)
        q.add(i);
        // Display contents of the queue.
```

```
        System.out.println("Elements of queue-"+q);
        // To remove the head of queue.
        int removedele = q.remove();
        System.out.println("removed element-" + removedele);
        System.out.println(q);
        // To view the head of queue int head = q.peek();
        System.out.println("head of queue-" + head);
        // Rest all methods of collection interface,
        // Like size and contains can be used with this
        // implementation.
        int size = q.size();
        System.out.println("Size of queue-" + size);
    }
}
```

Output:

Elements of queue-[0, 1, 2, 3, 4]

removed element-0

[1, 2, 3, 4]

head of queue-1

Size of queue-4

Set

```
import java.util.*;
public class SetDemo
{
    public static void main(String args[])
    {
        int count[] = {34,22,10,60,30,22};
        Set<Integer> set = new HashSet<Integer>();
        try
        {
            for (int i = 0; i < 5; i++)
            {
                set.add(count[i]);
            }
            System.out.println(set);
            TreeSet sortedSet = new TreeSet<Integer>(set);
            System.out.println("The sorted list is:");
            System.out.println(sortedSet);
            System.out.println("The First element of the set is:
" + (Integer)sortedSet.first());
            System.out.println("The last element of the set is:
" + (Integer)sortedSet.last());
        }
        catch (Exception e)
        {
        }
    }
}
```

Output:

[34, 22, 10, 60, 30]

The sorted list is:

[10, 22, 30, 34, 60]

The First element of the set is: 10

The last element of the set is: 60

Map Program:

```
import java.awt.Color;
import java.util.HashMap;
import java.util.Map;
import java.util.Set;
public class MapDemo
{
    public static void main(String[] args)
    {
        Map<String, Color> favoriteColors = new HashMap<String,
Color>();
        favoriteColors.put("sai", Color.BLUE);
        favoriteColors.put("Ram", Color.GREEN);
        favoriteColors.put("krishna", Color.RED);
        favoriteColors.put("narayana", Color.BLUE);
        // Print all keys and values in the map
        Set<String> keySet = favoriteColors.keySet();
        for (String key : keySet)
        {
            Color value = favoriteColors.get(key);
            System.out.println(key + " : " + value);
        }
    }
}
```

Output:

narayana : java.awt.Color[r=0,g=0,b=255]

sai : java.awt.Color[r=0,g=0,b=255]

krishna: java.awt.Color[r=255,g=0,b=0]

Ram : java.awt.Color[r=0,g=255,b=0]

Exp. 2 Perform setting up and Installing Hadoop in its three operating modes: Standalone,Pseudo distributed, Fully distributed.

Hadoop is written in Java, so you will need to have Java installed on your machine, version 6 or later. Sun's JDK is the one most widely used with Hadoop, although others have been reported to work.

Hadoop runs on Unix and on Windows. Linux is the only supported production platform, but other flavors of Unix (including Mac OS X) can be used to run Hadoop for development. Windows is only supported as a development platform, and additionally requires Cygwin to run. During the Cygwin installation process, you should include the open ssh package if you plan to run Hadoop in pseudo-distributed mode

ALGORITHM

STEPS INVOLVED IN INSTALLING HADOOP IN STANDALONE MODE:-

1. Command for installing ssh is **"sudo apt-get install ssh"**.
2. Command for key generation is **ssh-keygen –t rsa –P " ".**
3. Store the key into rsa.pub by using the command
4. **cat $HOME/.ssh/id_rsa.pub** >>$HOME/.ssh/authorized_keys
5. Extract the java by using the command **tar xvfz jdk-8u60-linux-i586.tar.gz.**
6. Extract the eclipse by using the command **tar xvfz eclipse-jee-mars-R-linux-gtk.tar.gz**
7. Extract the hadoop by using the command **tar xvfz hadoop-2.7.1.tar.gz**

8. Move the java to **/usr/lib/jvm/** and eclipse to **/opt/** paths. Configure the java path in the eclipse.ini file
9. Export java path and hadoop path in ./bashrc
10. Check the installation successful or not by checking the java version and hadoop version
11. Check the hadoop instance in standalone mode working correctly or not by using an implicit hadoop jar file named as word count.
12. If the word count is displayed correctly in part-r-00000 file it means that standalone modeis installed successfully.

ALGORITHM

STEPS INVOLVED IN INSTALLING HADOOP IN PSEUDO DISTRIBUTED MODE:-

1. In order install pseudo distributed mode we need to configure the hadoop configuration files resides in the directory
   /home/lendi/hadoop-2.7.1/etc/hadoop.
2. First configure the hadoop-env.sh file by changing the java path.
3. Configure the core-site.xml which contains a property tag, it contains name andvalue. Name as fs.defaultFS and value as hdfs://localhost:9000
4. Configure hdfs-site.xml
5. Configure yarn-site.xml.
6. Configure mapred-site.xml before configure the copy mapred-site.xml. template to mapred-site.xml.
7. Now format the name node by using command
   hdfs namenode –format.
8. Type the command start-dfs.sh,start-yarn.sh means that starts the daemons like NameNode,DataNode,SecondaryNameNode ,ResourceManager,NodeManager.
9. Run JPS, which views all daemons. Create a directory in the Hadoop by using command hdfs

dfs –mkdr /csedir and enter some data into lendi.txt using command nano lendi.txt and copy from local directory to hadoop using command hdfs dfs – copyFromLocal lendi.txt /csedir/and run sample jar file word count to check whetherpseudo distributed mode is working or not. Display the contents of file by using command hdfs dfs –cat /newdir/part-r-00000.

FULLY DISTRIBUTED MODE INSTALLATION: ALGORITHM

1. Stop all single node clusters

   $stop-all.sh

2. Decide one as NameNode (Master) and remaining as DataNodes(Slaves).
3. Copy public key to all three hosts to get a password less SSH access

   $ssh-copy-id –I $HOME/.ssh/id_rsa.pub lendi@l5sys24
4. Configure all Configuration files, to name Master and Slave Nodes.

   $cd $HADOOP_HOME/etc/hadoop

   $nano core-site.xml

   $ nano hdfs-site.xml

5. Add hostnames to file slaves and save it.

   $ nano slaves

6. Configure $ nano yarn-site.xml

7. Do in Master Node

   $ hdfs namenode –format

   $ start-dfs.sh

   $start-yarn.sh

   Format NameNode

Daemons Starting in Master and Slave Nodes

END

INPUT

ubuntu @localhost> jps

OUTPUT:

Data node, name nodem Secondary name node,

NodeManager, Resource Manager

VIVA VOCE QUESTIONS:-

1) What does _jps‘ command do?
2) How to restart Namenode?
3) Differentiate between Structured and Unstructured data?
4) What are the main components of a Hadoop Application?
5) Explain the difference between NameNode, Backup Node and Checkpoint NameNode.

Exp. 3. Implement the following file management tasks in Hadoop:Adding files and directories
Retrieving files
Deleting files

DESCRIPTION:-

- Adding files and directories to HDFS
- Retrieving files from HDFS to local file system
- Deleting files from HDFS

ALGORITHM:-

SYNTAX AND COMMANDS TO ADD, RETRIEVE AND DELETE DATA FROM HDFS

Step-1

Adding Files and Directories to HDFS

Before you can run Hadoop programs on data stored in HDFS, you'll need to put the data into HDFS first. Let's create a directory and put a file in it. HDFS has a default working directory of /user/$USER, where $USER is your login user name. This directory isn't automatically created for you, though, so let's create it with the mkdir command. For the purpose of illustration, we use chuck. You should substitute your user name in the example commands.

```
hadoop fs -mkdir /user/chuck
 hadoop fs -put example.txt
 hadoop fs -put example.txt /user/chuck
```

Step-2

Retrieving Files from HDFS

The Hadoop command get copies files from HDFS back to the local filesystem. To retrieve example.txt, we can run the following command:

hadoop fs -cat example.txt

Step-3

Deleting Files from HDFS

hadoop fs -rm example.txt

- Command for creating a directory in hdfs is “hdfs dfs –mkdir /lendicse”.
- Adding directory is done through the command “hdfs dfs –put lendi_english /”.

Step-4

Copying Data from NFS to HDFS

Copying from directory command is “hdfs dfs –copyFromLocal /home/lendi/Desktop/shakes/glossary /lendicse/”

- View the file by using the command “hdfs dfs –cat /lendi_english/glossary”
- Command for listing of items in Hadoop is “hdfs dfs –ls hdfs://localhost:9000/”.
- Command for Deleting files is “hdfs dfs –rm r /kartheek”.

SAMPLE INPUT:

Input as any data format of type structured, Unstructured or Semi Structured

EXPECTED OUTPUT:

VIVA-VOCE Questions

1) What is the command used to copy the data from local to hdfs

2) What is the command used to run the hadoop jar file

3) What command is used to remove directory from Hadoop recursively?

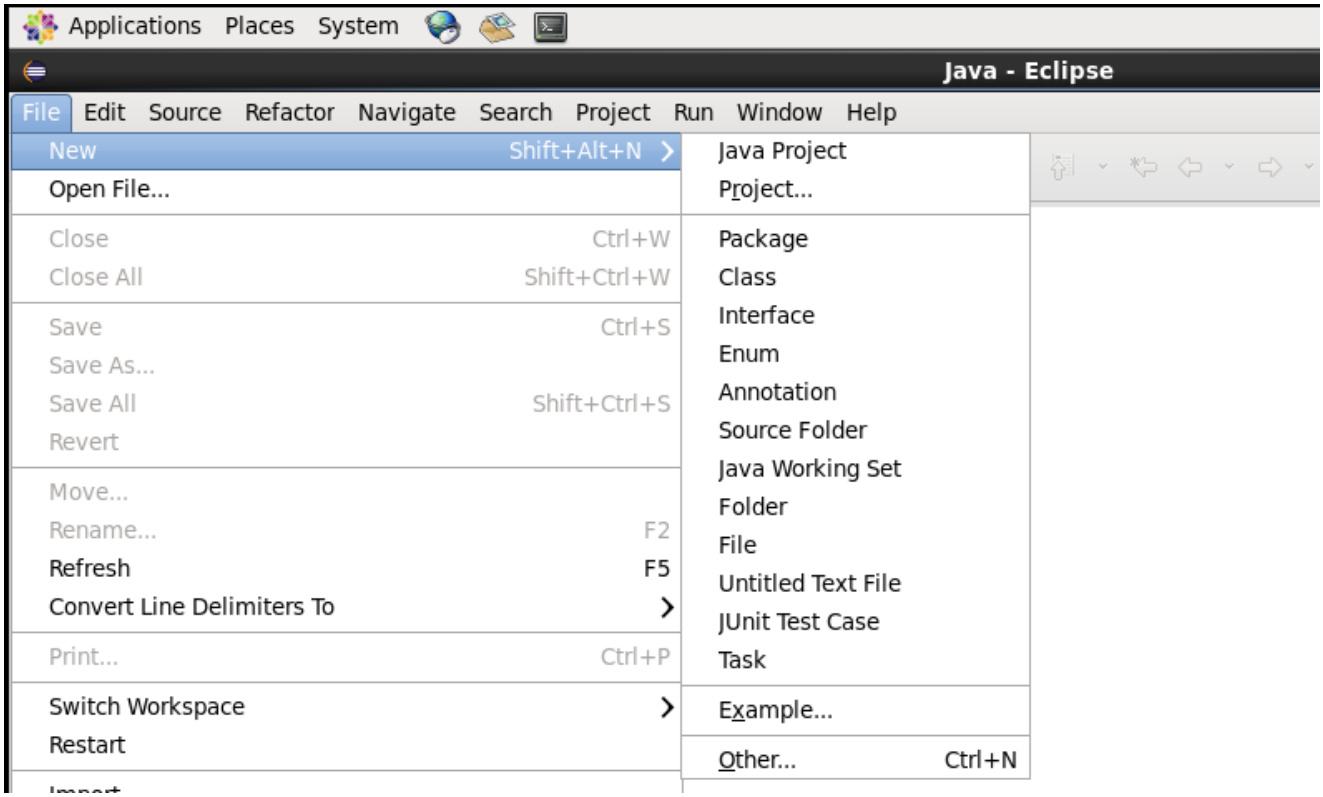
4) What is command used to list out directories of Data Node through web tool

Exp. 4. Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.

Word Count Map Reduce program to understand Map Reduce Paradigm

Source code:

First Open **Eclipse** -> then select **File** -> **New** -> **Java Project** ->Name it **WordCount** -> then **Finish**.



- Create Three Java Classes into the project. Name them **WCDriver**(having the main function), **WCMapper**, **WCReducer**.
- You have to include two Reference Libraries for that: Right Click on **Project** -> then select **Build Path**-> Click on **Configure Build Path**

WCMapper.java:

```
// Importing libraries
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;

public class WCMapper extends MapReduceBase implements
Mapper<LongWritable,Text, Text, IntWritable>
{
    // Map function
    public void map(LongWritable key, Text value, OutputCollector<Text,
                IntWritable> output, Reporter rep) throws IOException
    {
        String line = value.toString();
        // Splitting the line on spaces
        for (String word : line.split(" "))
        {
            if (word.length() > 0)
            {
                output.collect(new Text(word), new IntWritable(1));
            }
        }
    }
}
```

WordReducer.java:

```
// Importing libraries
import java.io.IOException;
import java.util.Iterator;
```

```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;

public class WCReducer extends MapReduceBase implements
Reducer<Text,IntWritable, Text, IntWritable>
{

    // Reduce function
    public void reduce(Text key, Iterator<IntWritable> value,
                OutputCollector<Text, IntWritable> output,
                            Reporter rep) throws IOException
    {

        int count = 0;

        // Counting the frequency of each words
        while (value.hasNext())
        {
            IntWritable i = value.next();
            count += i.get();
        }

        output.collect(key, new IntWritable(count));
    }
}
```

WCDriver.java:

```
// Importing libraries
import java.io.IOException;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
```

```
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class WCDriver extends Configured implements Tool {

    public int run(String args[]) throws IOException
    {
        if (args.length < 2)
        {
            System.out.println("Please give valid inputs");
            return -1;
        }

        JobConf conf = new JobConf(WCDriver.class);
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        conf.setMapperClass(WCMapper.class);
        conf.setReducerClass(WCReducer.class);
        conf.setMapOutputKeyClass(Text.class);
        conf.setMapOutputValueClass(IntWritable.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);
        JobClient.runJob(conf);
        return 0;
    }

    // Main Method
    public static void main(String args[]) throws Exception
    {
        int exitCode = ToolRunner.run(new WCDriver(), args);
        System.out.println(exitCode);
    }
}
```

Commands to execute the MAP-Reduce Program:

```
hadoop fs -mkdir /data_wordcount

touch wc.txt

hadoop fs -put /home/hdoop/wc.txt /data_wordcount

hadoop fs -ls /data_wordcount

hadoop dfs -cat /data_wordcount/wc.txt

hadoop jar /home/hdoop/WordCount.jar WCDriver /data_wordcount/wc.txt
/output_dir_wc

hadoop dfs -cat /output_dir_wc/*
```

Exp. 5 Write a Map Reduce program that mines weather data. Weather sensors collecting data every hour at many locations across the globe gather a large volume of log data, which is a good candidate for analysis with Map Reduce, since it is semi structured and record-oriented.

Weather Report POC-Map Reduce Program to analyze time-temperature statistics and generatereport with max/min temperature.

Problem Statement:

1. The system receives temperatures of various cities (Austin, Boston, etc) of USA captured at regular intervals of time on each day in an input file.
2. System will process the input data file and generates a report with Maximum and Minimum temperatures of each day along with time. 3. Generates a separate output report for each city.

Climate change has been seeking a lot of attention since long time. The antagonistic effect of this climate is being felt in every part of the earth. There are many examples for these, such as sea levels are rising, less rainfall, increase in humidity. The propose system overcomes the some issues that occurred by using other techniques. In this project, we use the concept of Big data Hadoop. In the proposed architecture, we are able to process offline data, which is stored in the National Climatic Data Centre (NCDC). Through this, we are able to find out the maximum temperature and minimum temperature of year, and able to predict the future weather forecast. Finally, we plot the graph for the obtained MAX and MIN temperature for each moth of the particular year to visualize the temperature. Based on the previous year data weather data of coming year is predicted.

ALGORITHM:-

MAPREDUCE PROGRAM

WordCount is a simple program which counts the number of occurrences of each word in a given text input data set. WordCount fits very well with the MapReduce programming model making it a great example to understand the Hadoop Map/Reduce programming style. Our implementation consists of three main parts:

1. Mapper
2. Reducer
3. Main program

Step-1. Write a Mapper

A Mapper overrides the ―map‖ function from the Class "org.apache.hadoop.mapreduce.Mapper" which provides <key, value> pairs as the input. A Mapper implementation may output <key,value> pairs using the provided Context .

Input value of the WordCount Map task will be a line of text from the input data file and the key would be the line number <line_number, line_of_text> . Map task outputs <word, one> for each word in the line of text.

Pseudo-code

```
void Map (key, value){
    for each max_temp x in value:
        output.collect(x, 1);
}
void Map (key, value){
    for each min_temp x in value:
```

```
        output.collect(x, 1);
}
```

Step-2 Write a Reducer

A Reducer collects the intermediate <key,value> output from multiple map tasks and assemble a single result. Here, the WordCount program will sum up the occurrence of each word to pairs as <word, occurrence>.

Pseudo-code

```
void Reduce (max_temp, <list of value>){
    for each x in <list of value>:
        sum+=x;
    final_output.collect(max_temp, sum);
}
void Reduce (min_temp, <list of value>){
    for each x in <list of value>:
        sum+=x;
    final_output.collect(min_temp, sum);
}
```

3. Write Driver

The Driver program configures and run the MapReduce job. We use the main program to perform basic configurations such as:

Job Name : name of this Job

Executable (Jar) Class: the main executable class. For here, WordCount.

Mapper Class: class which overrides the "map" function. For here, Map.

Reducer: class which override the "reduce" function. For here , Reduce.

Output Key: type of output key. For here, Text.

Output Value: type of output value. For here, IntWritable.

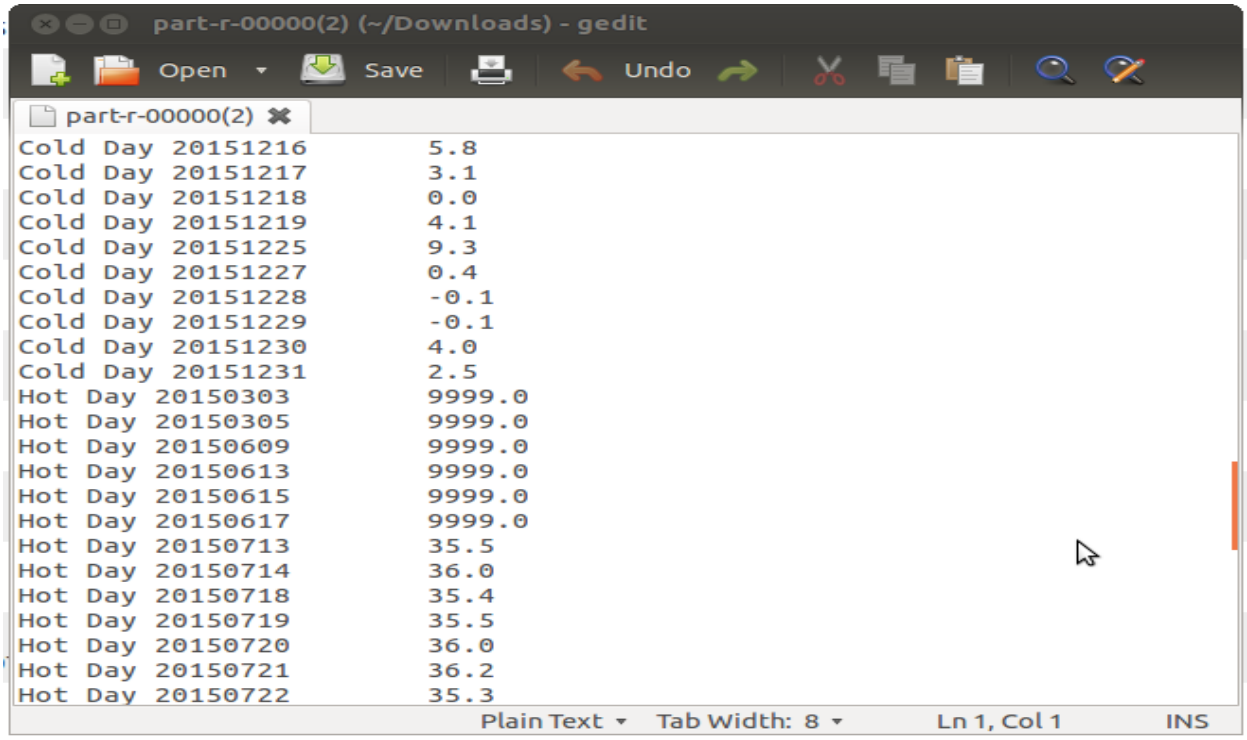
File Input Path

File Output Path

INPUT:-

Set of Weather Data over the years

OUTPUT:-


part-r-00000(2) (~/Downloads) - gedit
Open
Save
Undo
part-r-00000(2)
Cold Day 20151216 5.8
Cold Day 20151217 3.1
Cold Day 20151218 0.0
Cold Day 20151219 4.1
Cold Day 20151225 9.3
Cold Day 20151227 0.4
Cold Day 20151228 -0.1
Cold Day 20151229 -0.1
Cold Day 20151230 4.0
Cold Day 20151231 2.5
Hot Day 20150303 9999.0
Hot Day 20150305 9999.0
Hot Day 20150609 9999.0
Hot Day 20150613 9999.0
Hot Day 20150615 9999.0
Hot Day 20150617 9999.0
Hot Day 20150713 35.5
Hot Day 20150714 36.0
Hot Day 20150718 35.4
Hot Day 20150719 35.5
Hot Day 20150720 36.0
Hot Day 20150721 36.2
Hot Day 20150722 35.3
Plain Text
Tab Width: 8
Ln 1, Col 1
INS

VIVA VOCE QUESTIONS:

1) Explain what is the function of MapReducer partitioner?

2) Explain what is difference between an Input Split and HDFS Block?

3) Explain what is Sequencefileinputformat?

4) In Hadoop what is InputSplit?

Exp. 6 Implement Matrix Multiplication with Hadoop Map Reduce.

Implementing Matrix Multiplication with Hadoop Map Reduce.

Map.java:

```
package com.lendap.hadoop;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import java.io.IOException;

public class Map extends org.apache.hadoop.mapreduce.Mapper<LongWritable,
Text, Text, Text>
{
    @Override
    public void map(LongWritable key, Text value, Context context)
            throws IOException, InterruptedException
    {
        Configuration conf = context.getConfiguration();
        int m = Integer.parseInt(conf.get("m"));
        int p = Integer.parseInt(conf.get("p"));
        String line = value.toString();
        // (M, i, j, Mij);
        String[] indicesAndValue = line.split(",");
        Text outputKey = new Text();
        Text outputValue = new Text();
        if (indicesAndValue[0].equals("M")) {
            for (int k = 0; k < p; k++) {
                outputKey.set(indicesAndValue[1] + "," + k);
                // outputKey.set(i,k);
                outputValue.set(indicesAndValue[0] + "," +
indicesAndValue[2]
                        + "," + indicesAndValue[3]);
                // outputValue.set(M,j,Mij);
                context.write(outputKey, outputValue);
            }
        }
        else
        {
            // (N, j, k, Njk);
            for (int i = 0; i < m; i++)
            {
                outputKey.set(i + "," + indicesAndValue[2]);
```

```
                outputValue.set("N," + indicesAndValue[1] + ","
                        + indicesAndValue[3]);
                context.write(outputKey, outputValue);
            }
        }
    }
}
```

Reduce.java:

```
package com.lendap.hadoop;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import java.io.IOException;
import java.util.HashMap;

public class Reduce
  extends org.apache.hadoop.mapreduce.Reducer<Text, Text, Text, Text> {
    @Override
    public void reduce(Text key, Iterable<Text> values, Context context)
            throws IOException, InterruptedException {
        String[] value;
        //key=(i,k),
        //Values = [(M/N,j,V/W),..]
        HashMap<Integer, Float> hashA = new HashMap<Integer, Float>();
        HashMap<Integer, Float> hashB = new HashMap<Integer, Float>();
        for (Text val : values) {
            value = val.toString().split(",");
            if (value[0].equals("M")) {
                hashA.put(Integer.parseInt(value[1]),
Float.parseFloat(value[2]));
            } else {
                hashB.put(Integer.parseInt(value[1]),
Float.parseFloat(value[2]));
            }
        }
        int n = Integer.parseInt(context.getConfiguration().get("n"));
        float result = 0.0f;
        float m_ij;
        float n_jk;
        for (int j = 0; j < n; j++) {
            m_ij = hashA.containsKey(j) ? hashA.get(j) : 0.0f;
            n_jk = hashB.containsKey(j) ? hashB.get(j) : 0.0f;
            result += m_ij * n_jk;
```

```
        }
        if (result != 0.0f) {
            context.write(null,
                    new Text(key.toString() + "," +
Float.toString(result)));
        }
    }
}
```

MatrixMultiply.java:

```
package com.lendap.hadoop;

import org.apache.hadoop.conf.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class MatrixMultiply {

    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Usage: MatrixMultiply <in_dir>
<out_dir>");
            System.exit(2);
        }
        Configuration conf = new Configuration();
        // M is an m-by-n matrix; N is an n-by-p matrix.
        conf.set("m", "1000");
        conf.set("n", "100");
        conf.set("p", "1000");
        @SuppressWarnings("deprecation")
        Job job = new Job(conf, "MatrixMultiply");
        job.setJarByClass(MatrixMultiply.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);

        job.setMapperClass(Map.class);
        job.setReducerClass(Reduce.class);
```

```
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        job.waitForCompletion(true);
    }
}
```

EXERCISE-7:-

AIM:-Install and Run Pig then write Pig Latin scripts to sort, group, join, project and filter the data.

DESCRIPTION

Pig Latin is procedural and fits very naturally in the pipeline paradigm while SQL is instead declarative. In SQL users can specify that data from two tables must be joined, but not what join implementation to use (You can specify the implementation of JOIN in SQL, thus "... for many SQL applications the query writer may not have enough knowledge of the data or enough expertise to specify an appropriate join algorithm."). Pig Latin allows users to specify an implementation or aspects of an implementation to be used in executing a script in several ways. In effect, Pig Latin programming is similar to specifying a query execution plan, making it easier for programmers to explicitly control the flow of their data processing task.
SQL is oriented around queries that produce a single result. SQL handles trees naturally, but has no built in mechanism for splitting a data processing stream and applying different operators to each sub-stream. Pig Latin script describes a directed acyclic graph (DAG) rather than a pipeline.
Pig Latin's ability to include user code at any point in the pipeline is useful for pipeline development. If SQL is used, data must first be imported into the database, and then the cleansing and transformation process can begin.

ALGORITHM

STEPS FOR INSTALLING APACHE PIG

1) Extract the pig-0.15.0.tar.gz and move to home directory

2) Set the environment of PIG in bashrc file.

3) Pig can run in two modes
Local Mode and Hadoop Mode Pig –x local and pig
4) Grunt Shell
Grunt >
5) LOADING Data into Grunt Shell
DATA = LOAD <CLASSPATH> USING PigStorage(DELIMITER) as (ATTRIBUTE
:

DataType1, ATTRIBUTE : DataType2…..)

6) Describe Data

Describe DATA;

7) DUMP Data

Dump DATA;

8) FILTER Data

FDATA = FILTER DATA by ATTRIBUTE = VALUE;

9) GROUP Data

GDATA = GROUP DATA by ATTRIBUTE;

10) Iterating Data

FOR_DATA = FOREACH DATA GENERATE GROUP AS GROUP_FUN, ATTRIBUTE = <VALUE>

11) Sorting Data

SORT_DATA = ORDER DATA BY ATTRIBUTE WITH CONDITION;

12) LIMIT Data

LIMIT_DATA = LIMIT DATA COUNT;

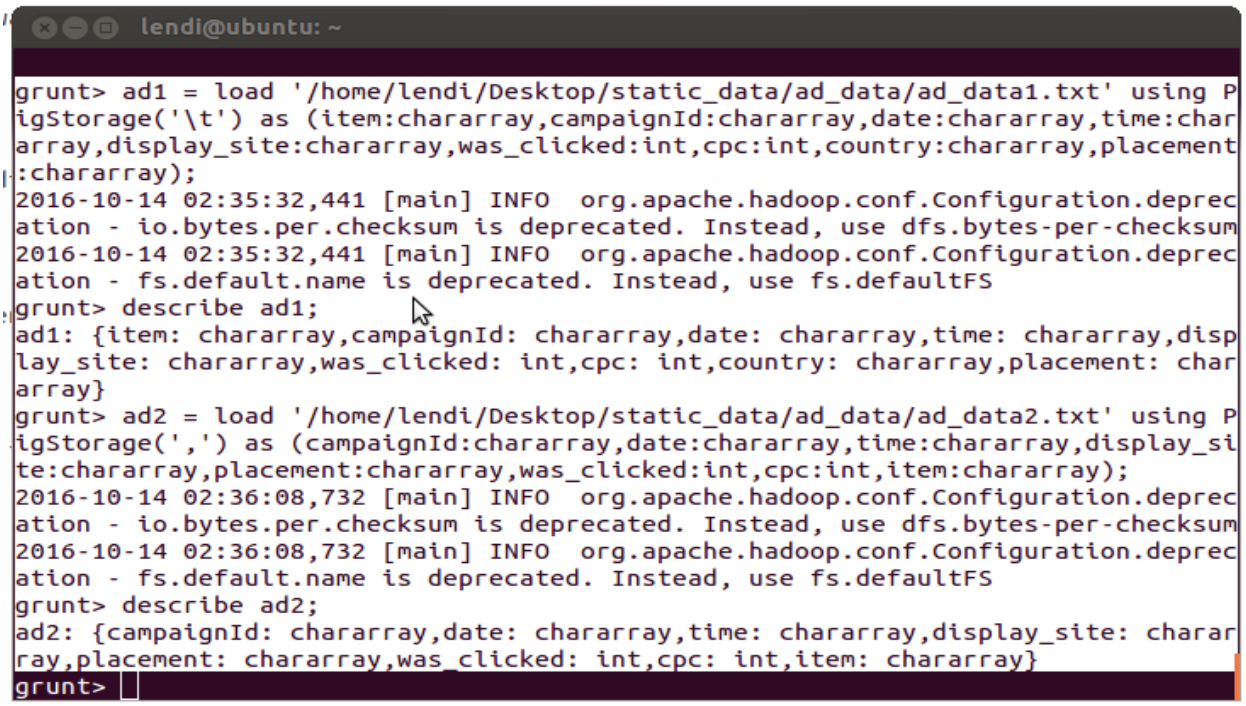
13) JOIN Data

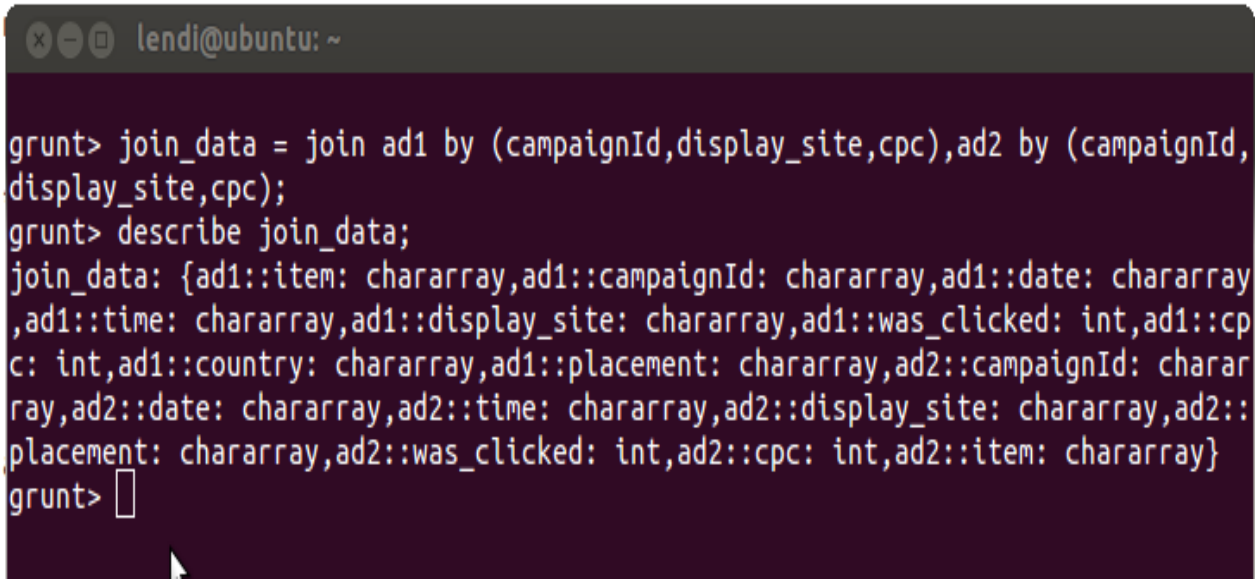
JOIN DATA1 BY (ATTRIBUTE1,ATTRIBUTE2….) , DATA2 BY (ATTRIBUTE3,ATTRIBUTE….N)

INPUT:

Input as Website Click Count Data

OUTPUT:





VIVA-VOCE Questions

1) What do you mean by a bag in Pig?
2) Differentiate between PigLatin and HiveQL
3) How will you merge the contents of two or more relations and divide a single relation into two or more relations?
4) What is the usage of foreach operation in Pig scripts?
5) What does Flatten do in Pig?

EXERCISE-8:-

AIM:-Install and Run Hive then use Hive to Create, alter and drop databases, tables, views, functions and Indexes.

DESCRIPTION

Hive, allows SQL developers to write Hive Query Language (HQL) statements that are

Hive service into MapReduce

ALGORITHM:

Apache HIVE INSTALLATION STEPS

1) Install MySQL-Server
   Sudo apt-get install mysql-server
2) Configuring MySQL UserName and Password
3) Creating User and granting all Privileges Mysql –uroot –proot
   Create user <USER_NAME> identified by <PASSWORD>
4) Extract and Configure Apache Hive
   tar xvfz apache-hive-1.0.1.bin.tar.gz
5) Move Apache Hive from Local directory to Home directory
6) Set CLASSPATH in bashrc

Export HIVE_HOME = /home/apache-hive Export PATH = $PATH:$HIVE_HOME/bin

7) Configuring hive-default.xml by adding My SQL Server Credentials

```
<property>
<name>javax.jdo.option.ConnectionURL</name>
<value> jdbc:mysql://localhost:3306/hive?createDatabaseIfNotExist=true
</value>
</property>
<property>
<name>javax.jdo.option.ConnectionDriverName</name>
<value>com.mysql.jdbc.Driver</value>
</property>
<property>
<name>javax.jdo.option.ConnectionUserName</name>
```

<value>hadoop</value>
</property>
<property>
<name>javax.jdo.option.ConnectionPassword</name>
<value>hadoop</value>
</property>

8) Copying mysql-java-connector.jar to hive/lib directory.

SYNTAX for HIVE Database Operations DATABASE Creation
CREATE DATABASE|SCHEMA [IF NOT EXISTS] <database name>

Drop Database Statement
DROP DATABASE StatementDROP (DATABASE|SCHEMA) [IF EXISTS] database_name [RESTRICT|CASCADE];

Creating and Dropping Table in HIVE

CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.] table_name
[(col_name data_type [COMMENT col_comment], ...)]

[COMMENT table_comment] [ROW FORMAT row_format] [STORED AS file_format]
Loading Data into table log_data Syntax:
LOAD DATA LOCAL INPATH '<path>/u.data' OVERWRITE INTO TABLE u_data;

Alter Table in HIVE

Syntax

ALTER TABLE name RENAME TO new_name
ALTER TABLE name ADD COLUMNS (col_spec[, col_spec ...]) ALTER TABLE name DROP [COLUMN] column_name

ALTER TABLE name CHANGE column_name new_name new_type ALTER TABLE name REPLACE COLUMNS (col_spec[, col_spec ...])

Creating and Dropping View

CREATE VIEW [IF NOT EXISTS] view_name [(column_name [COMMENT column_comment], ...)] [COMMENT table_comment] AS SELECT ...

Dropping View Syntax:
DROP VIEW view_name

Functions in HIVE

String Functions:- round(), ceil(), substr(), upper(), reg_exp() etc Date and Time Functions:- year(), month(), day(), to_date() etc Aggregate Functions :- sum(), min(), max(), count(), avg() etc
INDEXES

CREATE INDEX index_name ON TABLE base_table_name (col_name, ...) AS 'index.handler.class.name'
[WITH DEFERRED REBUILD]
[IDXPROPERTIES (property_name=property_value, ...)] [IN TABLE index_table_name]
[PARTITIONED BY (col_name, ...)] [
[ROW FORMAT ...] STORED AS ...
| STORED BY ...
]
[LOCATION hdfs_path] [TBLPROPERTIES (...)]

Creating Index

CREATE INDEX index_ip ON TABLE log_data(ip_address) AS 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler' WITH DEFERRED REBUILD;
Altering and Inserting Index

ALTER INDEX index_ip_address ON log_data REBUILD;
Storing Index Data in Metastore

SET hive.index.compact.file=/home/administrator/Desktop/big/metastore_db/tmp/index_ipaddress_re sult;

SET hive.input.format=org.apache.hadoop.hive.ql.index.compact.HiveCompactIndexInputFormat;

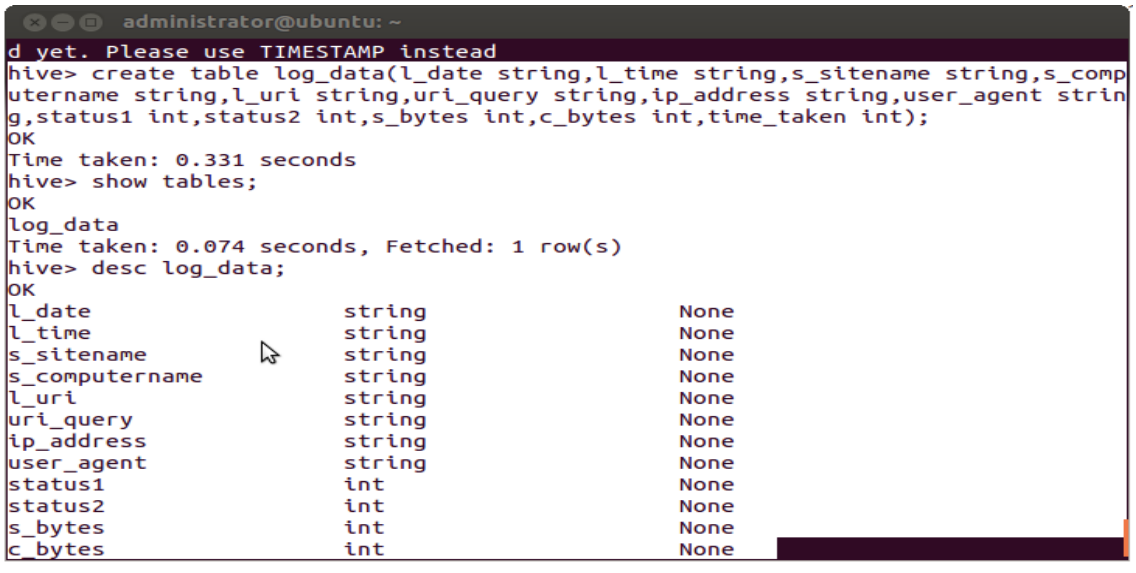
Dropping Index

DROP INDEX INDEX_NAME on TABLE_NAME;

INPUT

Input as Web Server Log Data

OUTPUT:



VIVA-VOCE Questions

1) I do not need the index created in the first question anymore. How can I delete the above index named index_bonuspay?
2) What is the use of Hcatalog?
3) Write a query to rename a table Student to Student_New.
4) Is it possible to overwrite Hadoop MapReduce configuration in Hive?
5) What is the use of explode in Hive?

Viva Questions

1. What do you know about the term “Big Data”?
2. What are the five V’s of Big Data?
3. Tell us how big data and Hadoop are related to each other.
4. How is big data analysis helpful in increasing business revenue?
5. Explain the steps to be followed to deploy a Big Data solution.
6. Define respective components of HDFS and YARN
7. Why is Hadoop used for Big Data Analytics?
8. What are the main differences between NAS (Network-attached storage) andHDFS?
9. Do you have any Big Data experience? If so, please share it with us.
10. Do you prefer good data or good models? Why?
11. Will you optimize algorithms or code to make them run faster?
12. How would you transform unstructured data into structured data?
13. How to recover a NameNode when it is down?
14. Explain the difference between Hadoop and RDBMS.
15. What happens when two users try to access the same file in the HDFS?

Vision and Mission of Institute

Vision

To promote higher learning in technology and industrial research to make our country a global player."

Mission

To promote quality education, training and research in the field of engineering by establishing effective interface with industry and to encourage the faculty to undertake industry sponsored projects for the students."

Quality Policy

We are committed to 'achievement of quality' as an integral part of our institutional policy by continuous self-evaluation and striving to improve ourselves.

Institute would pursue quality in

- All its endeavours like admissions, teaching- learning processes, examinations, extra and co-curricular activities, industry institution interaction, research & development, continuing education, and consultancy.

- Functional areas like teaching departments, Training & Placement Cell, library, administrative office, accounts office, hostels, canteen, security services, transport, maintenance section and all other services."

Vision of IT Department

To design and deliver intelligent IT industry oriented education.

Mission of IT Department

To prepare students to meet the need of users within an organizational and societal context through:

- Selection, creation, application, integration and administration of computing technologies.

- Delivering student resource in the IT enabled domain.

Program Outcome/Program Specific Outcome	Indicator	Competency
PO 1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialisation for the solution of complex engineering problems.	1.1.1	Apply mathematical techniques such as calculus, linear algebra, and statistics to solve problems
	1.1.2	Apply advanced mathematical techniques to model and solve computer science & engineering problems
	1.2.1	Apply laws of natural science to an engineering problem
	1.3.1	Apply fundamental engineering concepts to solve engineering problems
	1.4.1	Apply computer science & engineering concepts to solve engineering problems.
PO 2: Problem analysis: Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	2.1.1	Articulate problem statements and identify objectives
	2.1.2	Identify engineering systems, variables, and parameters to solve the problems
	2.1.3	Identify the mathematical, engineering and other relevant knowledge that applies to a given problem
	2.2.1	Reframe complex problems into interconnected sub-problems
	2.2.2	problems Identify, assemble and evaluate information
	2.2.3	Identify existing processes/solution methods for solving the problem, including forming justified approximations and assumptions
	2.2.4	Compare and contrast alternative solution processes to select the best process.
	2.3.1	Combine scientific principles and engineering concepts to formulate model/s (mathematical or otherwise) of a system or process that is appropriate in terms of applicability and required accuracy.
	2.3.2	Identify assumptions (mathematical and physical) necessary to allow modeling of a system at the level of accuracy required.
	2.4.1	Apply engineering mathematics and computations to solve mathematical models
	2.4.2	Produce and validate results through skilful use of contemporary engineering tools and models
	2.4.3	Identify sources of error in the solution process, and limitations of the solution.
	2.4.4	Extract desired understanding and conclusions consistent with objectives and limitations of the analysis
PO 3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.	3.1.1	Recognize that need analysis is key to good problem definition
	3.1.2	Elicit and document, engineering requirements from stakeholders
	3.1.3	Synthesize engineering requirements from a review of the state-of-the-art
	3.1.4	Extract engineering requirements from relevant engineering Codes and Standards such as IEEE, ACM, ISO etc.
	3.1.5	Explore and synthesize engineering requirements considering health, safety risks, environmental, cultural and societal issues

	3.1.6	Determine design objectives, functional requirements and arrive at specifications
	3.2.1	Apply formal idea generation tools to develop multiple engineering design solutions
	3.2.2	Build models/prototypes to develop diverse set of design solutions
	3.2.3	Identify suitable criteria for evaluation of alternate design solutions
	3.3.1	Apply formal decision making tools to select optimal engineering design solutions for further development
	3.3.2	Consult with domain experts and stakeholders to select candidate engineering design solution for further development
	3.4.1	Refine a conceptual design into a detailed design within the existing constraints (of the resources)
	3.4.2	Generate information through appropriate tests to improve or revise design
PO 4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	4.1.1	Define a problem, its scope and importance for purposes of investigation
	4.1.2	Examine the relevant methods, tools and techniques of experiment design, system calibration, data acquisition, analysis and presentation
	4.1.3	Apply appropriate instrumentation and/or software tools to make measurements of physical quantities
	4.1.4	Establish a relationship between measured data and underlying physical principles.
	4.2.1	Design and develop experimental approach, specify appropriate equipment and procedures
	4.2.2	Understand the importance of statistical designof experiments and choose an appropriate experimental design plan based on the study objectives
	4.3.1	Use appropriate procedures, tools and techniques to conduct experiments and collect data
	4.3.2	Analyze data for trends and correlations, stating possible errors and limitations
	4.3.3	Represent data (in tabular and/or graphical forms) so as to facilitate analysis andexplanation of the data, and drawing of conclusions
	4.3.4	Synthesize information and knowledge aboutthe problem from the raw data to reach appropriate conclusions
PO 5: Modern tool usage: Create, select, andapply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.	5.1.1	Identify modern engineering tools, techniques and resources for engineering activities
	5.1.2	Create/adapt/modify/extend tools and techniques to solve engineering problems
	5.2.1	Identify the strengths and limitations of tools for (i) acquiring information, (ii) modelling and simulating, (iii) monitoring system performance, and (iv) creating engineering designs.
	5.2.2	Demonstrate proficiency in using discipline specific tools
	5.3.1	Discuss limitations and validate tools,

		techniques and resources
	5.3.2	Verify the credibility of results from tool use with reference to the accuracy and limitations, and the assumptions inherent in their use.
PO 6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	6.1.1	Identify and describe various engineering roles; particularly as pertains to protection of the public and public interest at global, regional and local level
	6.2.1	Interpret legislation, regulations, codes, and standards relevant to your discipline and explain its contribution to the protection of the public
PO 7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	7.1.1	Identify risks/impacts in the life-cycle of an engineering product or activity
	7.1.2	Understand the relationship between the technical, socio economic and environmental dimensions of sustainability
	7.2.1	Describe management techniques for sustainable development
	7.2.2	Apply principles of preventive engineering and sustainable development to an engineeringactivity or product relevant to the discipline
PO 8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	8.1.1	Identify situations of unethical professional conduct and propose ethical alternatives
	8.2.1	Identify tenets of the ASME professional code of ethics
	8.2.2	Examine and apply moral & ethical principles to known case studies
PO 9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	9.1.1	Recognize a variety of working and learning preferences; appreciate the value of diversity on a team
	9.1.2	Implement the norms of practice (e.g. rules, roles, charters, agendas, etc.) of effective team work, to accomplish a goal.
	9.2.1	Demonstrate effective communication, problem solving, conflict resolution and leadership skills
	9.2.2	Treat other team members respectfully
	9.2.3	Listen to other members
	9.2.4	Maintain composure in difficult situations
	9.3.1	Present results as a team, with smooth integration of contributions from all individual efforts
PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	10.1.1	Read, understand and interpret technical and non-technical information
	10.1.2	Produce clear, well-constructed, and well-supported written engineering documents
	10.1.3	Create flow in a document or presentation
	10.2.1	Listen to and comprehend information, instructions, and viewpoints of others
	10.2.2	Deliver effective oral presentations to technical and non- technical audiences
	10.3.1	Create engineering-standard figures, reports and drawings to complement writing and presentations
	10.3.2	Use a variety of media effectively to convey a message in a document or a presentation
PO 11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader	11.1.1	Describe various economic and financial costs/benefits of an engineering activity
	11.1.2	Analyze different forms of financial statements to evaluate the financial status of an engineering

in a team, to manage projects and in multidisciplinary environments.		project
	11.2.1	Analyze and select the most appropriateproposal based on economic and financial considerations.
	11.3.1	Identify the tasks required to complete an engineering activity, and the resources required to complete the tasks.
	11.3.2	Use project management tools to schedule an engineering project so it is completed on time and on budget.
PO 12: Life-long learning: Recognise the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	12.1.1	Describe the rationale for requirement for continuing professional development
	12.1.2	Identify deficiencies or gaps in knowledge and demonstrate an ability to source information to close this gap
	12.2.1	Identify historic points of technological advance in engineering that required practitioners to seek education in order to stay current
	12.2.2	Recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in your field
	12.3.1	Source and comprehend technical literature and other credible sources of information
	12.3.2	Analyze sourced technical and popular information for feasibility, viability, sustainability, etc.
PSO1: Core Engineering Skills: Acquire basic concepts of Data Structures, Databases, Operating Systems, Computer Network, Theory of Computation, Advanced Programming and Software Engineering..	PSO1.1.1	Possess the concepts of Data Structure and Database Management System
	PSO1.1.2	Possess the concepts of core engineering subjects including Operating System, Computer Networks and Software Engineering.
	PSO1.1.3	Apply basic programming skills to solve real world problems
PSO2: Standard Software Engineeringpractices: Demonstrate an ability to design, develop, test, debug, deploy, analyse, troubleshoot, maintain,and secure mobile applications and software solutions for automation applications.	PSO2.1.1	Apply fundamental software engineering concepts to solve real world problem
	PSO2.1.2	Possess conceptual knowledge for designing, analysing and testing a software
	PSO2.1.3	Estimate and evaluate the cost related to a Software
PSO3: Project Endeavours: Provide platform for students to develop new and innovative projects as per current industry needs.	PSO3.1.1	Recognise the need and feasibility of project and apply standard practices for software project development
	PSO3.1.2	Identify the functional and non-functional requirement of current industry trends.
	PSO3.1.3	Recognise the challenges of changing trends and career opportunities as per current industry needs.

Programme:

B.Tech. (Information Technology)
Semester: VII
Course Name (Course Code): Big Data Analytics Lab (7IT4-21)

Course Outcomes

After completion of this course, students will be able to –

7IT4-21.1	**Understand** and implement the basics of data structures like Linked list, stack, queue, set and map in Java.
7IT4-21.2	**Demonstrate** the knowledge of big data analytics and implement different file management task in Hadoop.
7IT4-21.3	**Understand** Map Reduce Paradigm and develop data applications using variety of systems.
7IT4-21.4	**Analyze** and perform different operations on data using Pig Latin scripts.
7IT4-21.5	**Illustrate** and apply different operations on relations and databases using Hive.

Name of Faculty: - Praveen Kumar Yadav.

(Signature)

COURSE: Big Data Analytics Lab (7IT4-21)

Course Outcomes		Bloom's Level	PO Indicators	PSO Indicators
Upon successful completion of this course, students should be able to:				
7IT4-21.1	**Understand** and implement the basics of data structures like Linked list, stack, queue, set and map in Java.	2	1.3.1,1.4.1,2.1.3,2.2.1,2.2.4,4.1.2,4.3.3, 4.3.4,5.1.2,5.2.1,5.3.1.	PSO1.1.1, PSO1.1.2
7IT4-21.2	**Demonstrate** the knowledge of big data analytics and implement different file management task in Hadoop.	3	1.1.2,1.3.1,1.4.1,2.1.3,2.4.1,2.4.2,3.1.5, 3.2.1,3.3.2,3.4.2,4.1.2,4.3.2,4.3.3,4.3.4, 5.1.1,5.1.2,5.2.1,5.2.2,5.3.2	
7IT4-21.3	**Understand** Map Reduce Paradigm and develop data applications using variety of systems.	2,5	1.3.1,1.4.1,2.1.2,2.1.3,2.2.1,2.1.2,2.2.4, 2.4.2,3.2.1,3.2.2,3.3.1,4.2.1,4.3.1,4.3.3, 5.1.1,5.1.2,5.2.1,5.3.1	PSO2.1.2
7IT4-21.4	**Analyze** and perform different operations on data using Pig Latin scripts.	3,4	1.3.1,1.4.1,3.1.5,3.2.1,3.2.2,3.3.1,4.1.2, 4.1.3,4.3.2,4.3.3,5.1.1,5.1.2,5.2.2	PSO1.1.1, PSO1.1.3
7IT4-21.5	**Illustrate** and apply different operations on relations and databases using Hive.	3,4	1.3.1,1.4.1,3.1.5,3.2.1,3.2.2,3.3.1,4.1.2, 4.1.3,4.3.2,4.3.3,5.1.1,5.1.2,5.2.2,5.3.2	PSO1.1.1, PSO1.1.3

CO-PO/PSO Mapping
Programme: B.Tech.
(Information Technology)
Semester: VII
Course Name (Course Code): Big Data Analytics Lab (7IT4-21)

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO 1	2	1		2	2								3		
CO 2	2	1	1	2	3										
CO 3	2	2	1	1	3									2	
CO 4	2		1	2	2								3		
CO 5	2		1	2	3								3		