Q. store and retreive data in pig.

Aim :- To implement store and retreive data in pig.

student-data.txt in the HDFS directory named /data/ with the following content.

001, Rajiv, Reddy, 98480023 7, Hyderabad.
002, siddarth, Battacharya, 9848022338, kolkata
003, Rajesh, khanna, 9848022359, Delhi
004, preeti, Agarwal, 9848022330, pune
005, Thrupthi, mohanthy, 9848 0022336, Bhuwaneshwar

we can load data using the pig storage function as shown below.

```
student = LOAD 'hdfs://local host:9000/pig-data/student
-data.txt' USING pigstorage (',') as ( id: int,
first name : chararray, last name : chararray.
STORE student INTO 'hdfs:// local host : 9000/ pig
output /' USING pig storage (',') ;
```

This will store the data into the given directory.

output

you can verify the stored data as shown below. first of all, list out the files in the directory named pig-output using ls command as shown below.

```
$ hdfs dfs-ls 'hdfs://localhost:9000/pig-output/'
found 2 items
rw-r--r-1 Hadoop supergroup 0 2020-10-05-13:03
hdfs://localhost:9000/pig-output/-success.
$ hdfs dfs-cat 'hdfs://localhost: 9000 /pig-output
1, Rajiv, Reddy, 9848022337, Hyderabad.
2, siddarth, battacharya, 9848022338, kolkata
3, Rajesh, khanna, 9848022339, Delhi
4, preeti, Agarwal, 9848022336, Bhuwaneshwar
5, Trupthi, mohanthy, 9848022336, pune.
```

Q. perform social media analysis using cassandra.

Aim: To implement social media analysis using cassandra

create a table for storing user posts.

```
CREATE TABLE social-media.posts (
post-id uuid PRIMARY KEY,
user-id uuid,
post_text text,
post_time timestamp,
likes int,
shares int
);
```

insert a post into the table

```
INSERT INTO social-media.posts (post-id, user-id,
post-text,
post-time,
likes,
shares)
VALUES (uuid(), uuid(), 'excited to be learning
about cassandra!', to Timestamp(now()), 0, 0);
```

To find posts with more than 100 likes

```
SELECT * FROM social-media-posts WHERE likes > 100;
```

output

```
post id       likes        post text
sddu3fa87     101          excited to be learning
                                 about cassandra!

shares        userid
  0           dffbaffb-ae
```

MGIT

MGIT

Q. Buyer event analytics using cassandra on suitable product sales data.

**Aim:-** implement Buyer event analytics using cassandra on suitable product sales data.

Create a table for storing product sales data events.

```
CREATE TABLE sales. product_events (
event_id uuid PRIMARY KEY,
product_ id uuid,
buyer_id uuid,
event_time timestamp,
event_type text,
quantity int,
price decimal
);
```

Insert a sales event into the table.

```
INSERT INTO sales. product_events (event_id,
product-id,
buyer_id,
event_time,
event_type,
quantity, price)
VALUES (uuid (), uuid(), to Timestamp ()), 'purchase'
1, 19.99)
```

**output**

```
SELECT*FROM sales. product_events
WHERE product_id = <specific_product_id>
AND event_type = "purchase';
```

```
event-id    buyer-id    event-time     event-type
781a003d    64b96de1    2024-06-11      purchase
                        09:23:13051000

price       product-id    quantity
19.99       0b95658e-df5a    1
```

8. use R-project to carry out statistical analysis of big data.

sample data for big-data.csv

```
product-ID, product-category, sales-Amount, Date
1. Electronics, 150, 2022-03-15
2. clothing, 80, 2022-07-22
3. BOOKS, 120, 2022-05-10
4. Homedecor, 90, 2022-08-05
5. Electronics, 200, 2022-01-28
6. clothing, 50, 2022-11-14
7. BOOKS, 110, 2022,-09-19
8. HomeDecor, 70, 2022-04-03
9. Electronics, 180, 2022-06-30
10. clothing, 70, 2022-10-17
```

```
install.packages("dplyr")
install.packages("ggplot2")

library (dplyr)
library (ggplot2)
```

```
big-data<-read.csv("big-data.csv")
str (big-data)
total-sales<-big-data %>%
group-by(product-category) %>%
summarise (Total-sales=sum (sales))
total-sales
ggplot(total-sales, aes(x=product-category,
y=Total-sales))+
geom-bar(stat ="identity", fill ="skyblue")+
labs(title="Total sales by product category",
        x ="product category",
        y = "Total sales")+ theme-minimal()
```

output

```
product-category Total-sales
1 BOOKS          230
2 clothing       200
3 Electronics    530
4 HomeDecor      160
```

9. use R-project for data visualization of social media data.

social_media_data.csv

Text, Hashtags

" Excited to announce the launch of our new product!

" check out our latest blog post about sustainability!

" we're hosting a webinar next week on digital marketing startegies join us!

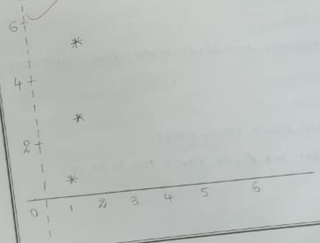
" Happy friday everyone! Have a great weekend!

" Throwback to our team outing last summer.

" Excited to attend the conference next month!

```
library (ggplot 2)
social_media_data<-read.csv("social_media_data.csv")
particular_hashtag <- "your_hashtag"
hashtag_data<- subset (social_media_data, grepl
(paste0(" \\b", particular_hashtag, "\\b"),
hashtags, ignore.case = TRUE))
hashtag_frequency <- nrow(hashtag_data)
ggplot( )+
  geom_bar (data = NULL, aes(x = "", y = hashtag_
frequency), fill = "skyblue", stat = "identity")+
  theme_minimal( )
```

output



6. using power point (excel) perform the following on any dataset.

a) Big data analytics

b) Big data charting.

a) Aim: using power point to perform big data analytics.

procedure.

1) consider sample data

→ we use two datasets one is customer into table and the other is the order into table both have customerID as the common field.

2) Getting excel power pivot Add In

→ open excel

→ select files > options

→ select add-ins

→ select the manage drop down menu, then select com Add-ins

→ select Go

→ select microsoft power pivot Excel

→ select OK. It adds the power pivot tab to excel.

3) Adding data into the data model

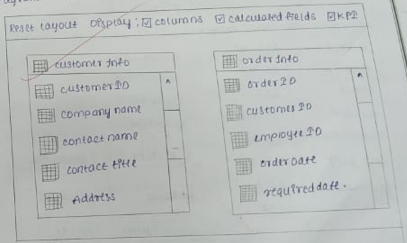
→ select the range of customer into table. Then, Add to data model from power pivot tab.

→ you will notice that new pop up window appears. This is poaer pivot window.

4) creating realationship between tables.

→ select the power pivot window Go to home then diagram view.

→ The imported table will appear as separate boxes in diagram view.

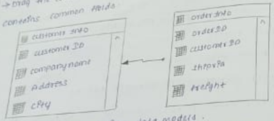


Reset layout Display: ☑ columns ☑ calculated fields ☑ KPI

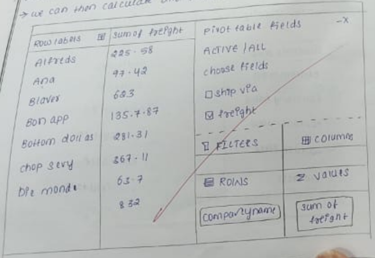
customer info
- customerID
- company name
- contact name
- contact title
- Address

order info
- orderID
- customerID
- employeeID
- orderDate
- requireddate.

→ Drag the column heading from one table to another that conteins common fields



5) create pivot tables using data models.
→ In power pivot window go to home > pivot table.
→ create pivot table using dialog box will appear. select new worksheet and then select ok.
→ we can then calculate and various things.



| Row labels ⊞ | sum of freight |
|---|---|
| Alfreds | 225.58 |
| Ana | 97.42 |
| Blaves | 623 |
| Bon app | 135.7.87 |
| Bottom dollas | 281.31 |
| chop sery | 367.11 |
| Die monde | 63.7 |
|  | 832 |

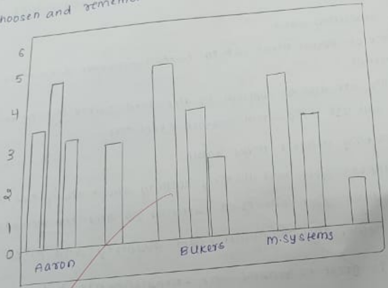
Pivot table fields –X
ACTIVE | ALL
choose fields
□ ship via
☑ freight

| FILTERS | columns |
|---|---|
|  |  |
| ROWS | Σ values |
| companyname | sum of freight |

6) Aim:- using power point to do big data charting.
procedure.
1. Accessing power pivot
file → menu → options → Add. Ins → microsoft power pivot for excel.
2. Importing data
click on power pivot tab in button → manage data → get external.
→ This are alot of options in the data source list in this we will use data from another Excel file.
3 making a basic pivot table
→ suppose you have looking keeping about that has a item deals. each line is a customer ID and the sections are name, invoice numer, cate, quantity and price.
4. It is great to rename the friendly name. hedec to a title that the informational index for this situation the title has been changed to invoices click finish.
5. making a pivot check a power pivot tables.
5. making a pivot click the pivot table
→ To make this from power pivot click the pivot chart in the excel cheap and click pivot chart

→ Another exercise manual will open utilize the fields on the option to choosen fields.

→ so customer name, Date, and quantity have been choosen and remembered for pivot chart.



**Aim**: To implement a simple word count using mapReduce

**program** : // BDA Word Count Sample Driver. java

```
import java.io.IOException;
import org.apache.hadoop.fs.path;
import org.apache.hadoop.io.Intwritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.Jobconf;
import org.apache.hadoop.mapred.mapper;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.inputFile
                                        InputFormat;
import org.apache.hadoop.mapreduce.lib.input.Text
                                        InputFormat;
import org.apache.hadoop.mapreduce.lib..output.File
                                        outputFormat;
import org.apache.hadoop.mapreduce.lib..output:Text
                                        outputFormat;
public class BDAWordCountSampleDriver {
     public static void main (String []args) throws
                                        IOException,
     Interrupted Exception. ClassNotfound Exception
Job. job =new Job();
job. setJobName (" word counter');
Job.setJar By class (BDA word count sample Driver.
                                         class);
Job. setmapper class (BDA word count sample mapper.class);
Job. set Reducer class (BDA wordcount sample Reducer.class);
Job. set output key class ( Text.class);
Job. set output value class ( Intwritable.class);
File Input Format . add Input path (Job, new path (" /sample
                                        /word txt"));

File output Format . set output path (Job, new path (" /sample
                              /word count"));
system. exit (Job. wait For compiletion (true) ? 0 : 1);
}
}
// BDA Word Count sample mapper. java
import java. io . IOException;
import org. apache. hadoop. io . Intwritable;
import org. apache. hadoop. io. Long writable;
import org. apache. hadoop. io. Text;
import org. apache. hadoop. mapreduce. mapper;
public class BPAword Count sample mapper extends
mapper
< Long writable , Text, Text, Int writable > {
@ override
```

```
protected void map ( Longwritable key, Text value , context
                           context)
throws IO Exception , Interrupted Exception {
string[] words = value . toString () split (" ,");
for (string word : word) {
context . write (new Text (word), new Intwritable(1));
}
}

// BDA Word Count sample Reducer java

import java. io. IOException;
import org. apache. hadoop. io. Intwritable;
import org. apache. hadoop. io. Text;
import org. apache. hadoop. mapreduce. Reducer;
public class BDA word count SampleDriver Reducer extends
               Reducer
          < Text , Intwritable , Text, Int writable > {
           @ override
protected void Reduce (Text word, Iterable < Intwritable>
values, context context)
throws IO Exception , Interrupted Exception {
 Integer count = 0;
 for (Int writable val : values) {
```

```
count += value.get();
 }
context.write(word, new IntWritable(count));
 }
}
```

**Aim:** To implement a simple map-reduce job that builds and inverted index on the set of input docs (Hadoop)

**program**

```
// mapper class
import java.io.IOException;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
public class InvertedIndexMapper extends Mapper<Long
        Writable, Text, Text, Text> {
    private Text word = new Text();
    private Text docid = new Text();
  @override
protected void map (LongWritable key, Text value, Context
                    context)
throws IOException, InterruptedException {
string line = value.toString();
string[] parts = line.split("\t");
 if (parts.length >= 2) {
   string DocId = parts[0];
   docid.set(DocId);
   string[] words = parts[1].split[" "];
```

```
for (string w:words) {
word.set (w);
context.write (word, docid);
  }
 }
 }
}

// Reducer class
import java.io.IOException;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
public class Inverted Index Reducer extends Reducer
<Text, Text, Text, Text> {
@override
protected void reduce (Text Key, Iterable <Text> values,
context context)
throws IOException. Interrupted Exception {
string Builder doclist =new string Builder ();
for ( Text docid :values ) {
if (doclist.length ()>0){
 doclist.append (',');
}
```

```
doclist.append (docid. to string ());
}
context.write (key.new Text (doclist to string ());
}
}
// main class
import org.apache.hadoop.fs.path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.file
                                    Input Format;
import org.apache.hadoop.mapreduce.lib.output.file
                                    output format;
public class Inverted Indexer {
  public static void main (string []args) throws Exception{
if (args.length !=2) {
system.err.println ("Usage: Inverted Index <input path>
                                  <output path>");
system.exit ();
}
Job job= new Job ();
job.set Jar By class (Inverted Index.class)
job.set Job Name ("Inverted Index");
```

```
File Input Format . addInput path (job, new path (args(0]));
File output Format . set outputpath ( job, new path (args(1]));
job. set mapper class (Inverted Index mapper. class);
job. set Reducer class (Inverted Index Reducer. class);
job. set output key class (Text .class);
job. set output value class ( Text . class);
system. exit (job. wait For completion (true) ? 0 : 1 );
}
}
```

compile your code and create a jar file . Then you can run your map-reduce job using hadoop with the following command.

```
hadoop jar Inverted Index. jar inverted Index < input path>
<output path>
```

DOC 1. txt

hello world

hello hadoop

DOC 2. txt

hadoop. is a framework

hello map reduce .

output

```
framework     DOC 2 . txt
hello         DOC1 . txt , DOC 2. txt
hadoop        DOC1. txt, DOC 2. txt
is              DOC 2 . txt
mapreduce       DOC 2 . txt
world             DOC 1 . txt .
```