V N R V J I E T	Sheet No……
Name of the Laboratory: _____	**Name of the Experiment:**_____ _____ **Experiment No.**____**Date**_____

HDFS COMMANDS

Aim: To understand the concept of HDFS and to execute HDFS commands

Objectives:

1. Create a file in local file system and copy it in HDFS
2. Copy a file from HDFS to local file system
3. Create a directory in HDFS and copy a file into it from local file system 4. Execute some other HDFS shell commands like: get, put, ls, cat, cp, mv, mkdir, rmdir, rm,appendToFile. tail, touchz, expunge etc.

Key concept

HDFS commands are used to access the Hadoop File System. HDFS stands for '**H**adoop **D**istributed **F**ile **S**ystem'.

The HDFS is a sub-project of the Apache Hadoop project. This Apache Software Foundation project is designed to provide a fault-tolerant file system designed to run on commodity hardware. HDFS is accessed through a set of shell commands.

For executing the HDFS commands , open the terminal in cloudera vm.

Q1) How to check the version of Hadoop framework?

Command: hadoop version

[cloudera@quickstart Desktop]$ hadoop version

Hadoop 2.6.0-cdh5.10.0

Q2) List the files and subdirectories present in HDFS

Command: **hadoop fs –ls**

This command will list all the available files and subdirectories under default directory. For instance, in our example the default directory for Cloudera VM is /user/cloudera

[cloudera@quickstart Desktop]$ hadoop fs -ls

Q3) Return all the directories under root directory

Variations of Hadoop ls Shell Command

[cloudera@quickstart Desktop]$ hadoop fs -ls /

Q4) Copy a file from local to HDFS?

copyFromLocal

HDFS Command to copy the file from a Local file system to HDFS.

Usage: **hadoop fs -copyFromLocal <localsrc> <hdfs destination>**

First I have created a file in local named student.txt

[cloudera@quickstart Desktop]$ gedit students.txt

[cloudera@quickstart Desktop]$ hadoop fs -copyFromLocal students.txt /user/cloudera

Q5: Check if the file is copied to HDFS?

[cloudera@quickstart Desktop]$ hadoop fs -ls

Q6: Check the contents of file that you copied in HDFS?

cat: HDFS Command that reads a file on HDFS and prints the content of that file to the standard output.

Usage: **hadoop fs –cat /path_to_file_in_hdfs**

[cloudera@quickstart Desktop]$ hadoop fs -cat students.txt

1,kriti,cse,45

2,neha,ece,56

3,jyothi,ce,78

4.priya,mechanical,89

Q7: Achieve the same operation as above with put command?

put: HDFS Command to copy single source or multiple sources from local file system to the destination file system.

Usage: **hadoop fs -put <localsrc> <destination>**

[cloudera@quickstart Desktop]$ hadoop fs -put students.txt /user/cloudera/studentscopied.txt

[cloudera@quickstart Desktop]$ hadoop fs -ls

Q8) Copy any file from HDFS to Local File System

· copyToLocal

HDFS Command to copy the file from HDFS to Local File System.

Usage: **hadoop fs -copyToLocal <hdfs source> <localdst>**

[cloudera@quickstart Desktop]$ hadoop fs -copyToLocal students.txt studentscopiedtolocal.txt

[cloudera@quickstart Desktop]$ ls

Q9) Check the health of the Hadoop file system.

· fsck

HDFS Command to check the health of the Hadoop file system.

Command: **hdfs fsck /** hdfs fsck /

The filesystem under path '/' is HEALTHY

Q10) Create a directory in HDFS

HDFS Command to create the directory in HDFS.

Usage: **hadoop fs –mkdir /directory_name**

[cloudera@quickstart Desktop]$ hadoop fs –mkdir kriti2018

[cloudera@quickstart Desktop]$ hadoop fs -ls

Q11) Copy any file present in HDFS to a directory which is also present in HDFS

[cloudera@quickstart Desktop]$ hadoop fs –mkdir kriti2018

[cloudera@quickstart Desktop]$ hadoop fs -ls

[cloudera@quickstart Desktop]$ hadoop fs –cp kriti.txt kriti2018

Q12): Create an empty file in HDFS?

Touchz: HDFS Command to create a file in HDFS with file size 0 bytes. Usage:

hadoop fs –touchz /directory/filename

[cloudera@quickstart Desktop]$ hadoop fs –touchz empty.txt

[cloudera@quickstart Desktop]$ hadoop fs -ls

Q13) Check the file size of any file in HDFS?

Du: HDFS Command to check the file size.

Usage: hadoop fs –du –s /directory/filename

[cloudera@quickstart Desktop]$ hadoop fs –du students.txt

67 67 /user/cloudera/students.txt

Q14) Print the contents of a file stored in HDFS?

· cat

HDFS Command that reads a file on HDFS and prints the content of that file to the standard output.

Usage: **hadoop fs –cat /path/to/file_in_hdfs**

[cloudera@quickstart Desktop]$ hadoop fs -cat students.txt

1,kriti,cse,45

2,neha,ece,56

3,jyothi,ce,78

4.priya,mechanical,89

Q15) Count the number of directories and files inside a directory in HDFS?

count: **HDFS Command to count the number of directories, files, and bytes under the paths that match the specified file pattern.**

Usage: hadoop fs -count <path>

[cloudera@quickstart Desktop]$ hadoop fs –count kriti2018

1 1 13 /user/cloudera/kriti2018

Q16) Remove a file from HDFS?

rm: HDFS Command to remove the file from HDFS.

Usage: hadoop fs dfs –rm <path>

[cloudera@quickstart Desktop]$ hadoop fs -rm empty.txt

Deleted empty.txt

Q17) Delete a directory completely in HDFS?

rm -r

HDFS Command to remove the entire directory and all of its content from HDFS. Usage:

hadoop fs -rm -r <path>

[cloudera@quickstart Desktop]$ hadoop fs -rm –r kriti2018

Q18) Copy a file or multiple files in a directory in HDFS

Cp: HDFS Command to copy files from source to destination. This command allows multiple sources as well, in which case the destination must be a directory.

Usage: **hadoop fs -cp <src> <dest>**

Command: **hadoop fs -cp /user/hadoop/file1 /user/hadoop/file2**

[cloudera@quickstart Desktop]$ hadoop fs -mkdir /user/cloudera/kriti2018

[cloudera@quickstart Desktop]$ hadoop fs –cp dummy3.txt kriti2018

Q19)Move a file into a directory in HDFS?

mv: HDFS Command to move files from source to destination. This command allows multiple sources as well, in which case the destination needs to be a directory.

Usage: hadoop fs -mv <src> <dest>

[cloudera@quickstart Desktop]$ hadoop fs –mv emptyfile.txt kriti2018

Q20) Find a help for an individual command?

Usage command gives all the options that can be used with a particular hdfs command. HDFS

Command that returns the help for an individual command.

Usage: hadoop fs -usage <command>

Command: hdfs dfs -usage mkdir

[cloudera@quickstart Desktop]$ hdfs dfs -usage mkdir

Usage: hadoop fs [generic options] -mkdir [-p] <path> ...

Q21) Find the help for a given or all commands?

help

HDFS Command that displays help for given command or all commands if none is specified. Command:

hadoop fs -help

Q22)Check the memory status?

Check memory status:

Usage: hadoop fs -df hdfs :/

[cloudera@quickstart Desktop]$ hadoop fs -df

Filesystem Size Used Available Use%

hdfs://quickstart.cloudera:8020 58531520512 1245229056 46116413440 2%

[cloudera@quickstart Desktop]$

Q23) Check for cluster balancing in HDFS?

Cluster Balancing

Usage: hadoop balancer

Type command

hadoop balancer

Q24) Change permission for a file to 777?

chmod: Changes the permissions of files.

[cloudera@quickstart Desktop]$ hadoop fs -ls -r

[cloudera@quickstart Desktop]$ hadoop fs -chmod 777 /user/cloudera/students.txt

[cloudera@quickstart Desktop]$ hadoop fs -ls -r

Q25) Empty the trash in HDFS?

expunge: Empties the trash. When you delete a file, it isn't removed immediately from HDFS, but is renamed to a file in the /trash directory. As long as the file remains there, you can undelete it if you change your mind, though only the latest copy of the deleted file can be restored.

[cloudera@quickstart Desktop]$ hadoop fs -expunge

26) Display the last kilobyte of the particular file?

tail

This hadoop command will show the last kilobyte of the file to stdout.

[cloudera@quickstart Desktop]$ hadoop fs -tail /user/cloudera/n.txt

27) Append the contents of a file present in local to a file present in HDFS

[cloudera@quickstart Desktop]$ gedit first.txt

[cloudera@quickstart Desktop]$ gedit second.txt

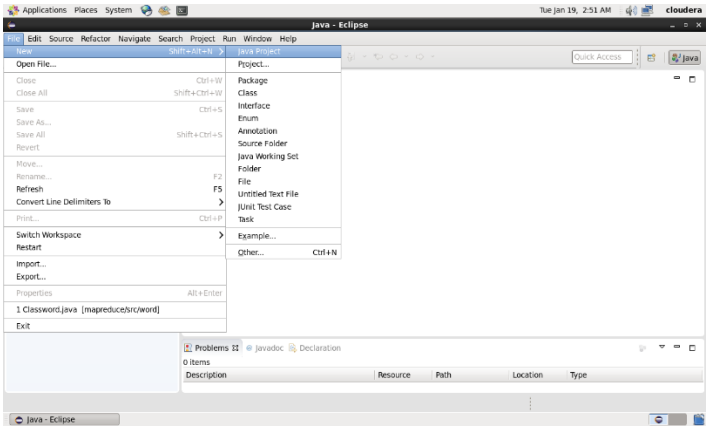
[cloudera@quickstart Desktop**]$ hadoop fs -copyFromLocal second.txt /user/cloudera/**

[cloudera@quickstart Desktop**]$ hadoop fs -appendToFile** /home/cloudera/Desktop/first.txt /user/cloudera/second.txt

[cloudera@quickstart Desktop]$ hadoop fs -cat /user/cloudera/second.txt

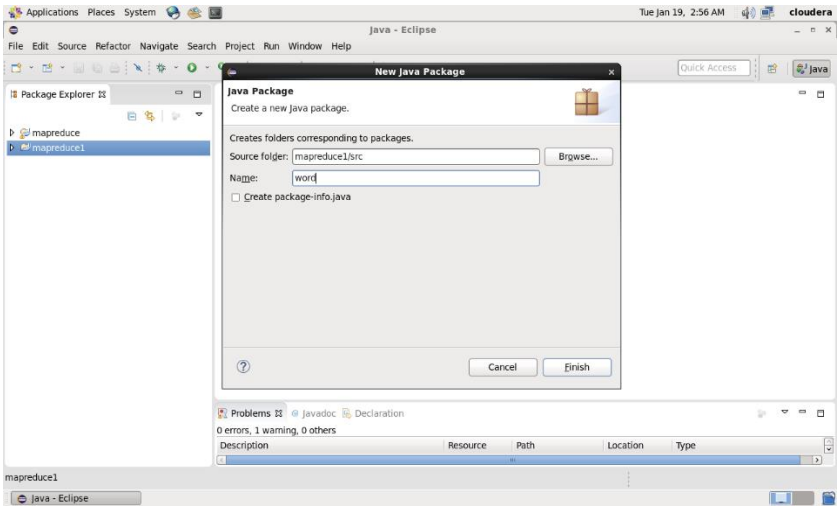
MAP REDUCE

1. Download jar files
2. Open eclipse from the desktop of cloudera
3. Create project



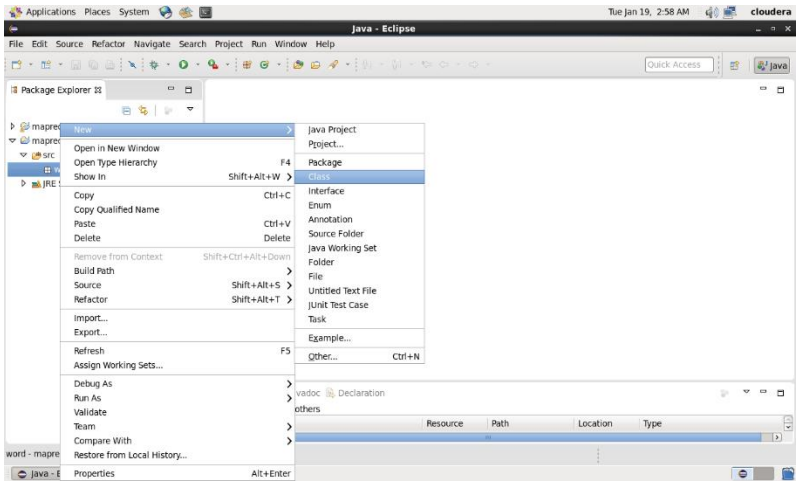
Give project name-- > mapreduce press on finish button

4. Create package
   Right click on the project select new package



Give package name as word then press finish

5. Create class Classword



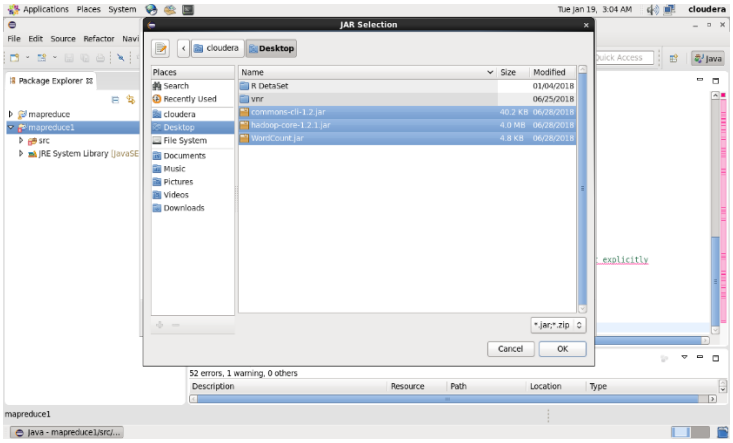
6. Write the given mapreduce program in the created class and save it package word;

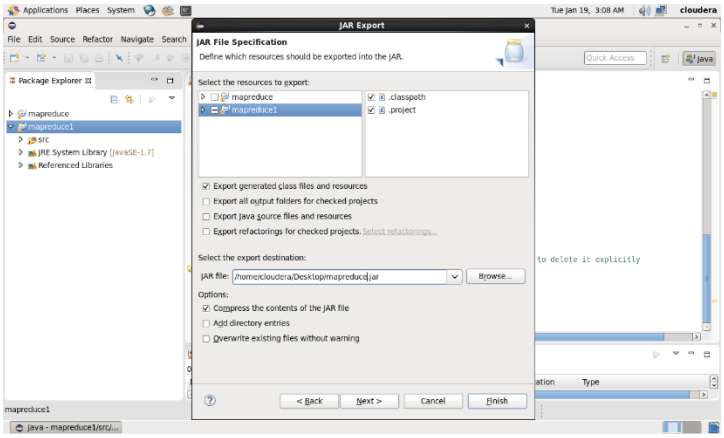
```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
    import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.fs.Path;

public class Classword
{
        public static class Map extends Mapper<LongWritable,Text,Text,IntWritable>
        {
                public void map(LongWritable key, Text value,Context context) throws
IOException,InterruptedException
                {
                        String line = value.toString();
                        StringTokenizer tokenizer = new StringTokenizer(line);
                        while (tokenizer.hasMoreTokens())
                        {
                                value.set(tokenizer.nextToken());
                                context.write(value, new IntWritable(1));
                  }
          }
  }

        public static class Reduce extends Reducer<Text,IntWritable,Text,IntWritable>
```

```
        {
                public void reduce(Text key, Iterable<IntWritable> values,Context context) throws
IOException,InterruptedException
                {
                        int sum=0;
                        for(IntWritable x: values)
                        {
                                sum+=x.get();
                        }
                        context.write(key, new IntWritable(sum));
                        }
        }

        public static void main(String[] args) throws Exception
        {
                Configuration conf= new Configuration();
                Job job = new Job(conf,"My Word Count Program");
                job.setJarByClass(Classword.class); //here put your class name
                job.setMapperClass(Map.class);
                job.setReducerClass(Reduce.class);
                job.setOutputKeyClass(Text.class);
                job.setOutputValueClass(IntWritable.class);
                job.setInputFormatClass(TextInputFormat.class);
                job.setOutputFormatClass(TextOutputFormat.class);
                Path outputPath = new Path(args[1]);
                //Configuring the input/output path from the filesystem into the job
                FileInputFormat.addInputPath(job, new Path(args[0]));
                FileOutputFormat.setOutputPath(job, new Path(args[1]));
                //deleting the output path automatically from hdfs so that we don't have to delete it explicitly
                outputPath.getFileSystem(conf).delete(outputPath);
                //exiting the job only if the flag value becomes false
                System.exit(job.waitForCompletion(true) ? 0 : 1);
        }
}
```

7. Add jar files

8. Create a jar file for a given program





9. Jar file will be created on the desktop

10. Open the terminal, Create a txt file (count.txt) on the desktop and move it to HDFS
    [cloudera@quickstart ~]$ cd Desktop

    [cloudera@quickstart Desktop]$ hadoop fs -copyFromLocal count.txt /user/cloudera/

    [cloudera@quickstart Desktop]$ hadoop fs -cat count.txt

    [cloudera@quickstart Desktop]$ hadoop jar mapreduce.jar word.classword /user/cloudera/count23.txt /user/cloudera/result

open another terminal

[cloudera@quickstart ~]$ hadoop fs -ls result

Found 2 items

-rw-r--r-- 1 cloudera cloudera 0 2021-12-28 22:58 result/_SUCCESS

-rw-r--r-- 1 cloudera cloudera 89 2021-12-28 22:58 result/part-r-00000

[cloudera@quickstart ~]$ hadoop fs -cat result/part-r-00000

all 1

are 3

fine 1

good 1

V N R V J I E T	Sheet No……
Name of the Laboratory: _____	**Name of the Experiment:**_____ _____ **Experiment No.**_____**Date**_____

HIVE

Aim: To understand Data Processing Tool – Hive and HQL (Hive query language)

Objectives:

1. Create Managed and External tables in HIVE
2. Load data in HIVE table from Local File System
3. Load data in HIVE table from HDFS
4. Query data sets using Hive QL
5. Create partitions and buckets

Key concept:

- Hive is a Data warehousing tool in Hadoop ecosystem. · HIVE Facilitates reading, writing, and managing large datasets residing in distributed storage and queried using SQL syntax.
- It is used for analyzing structured and semi-structured data. Hive abstracts the complexity of Hadoop MapReduce. Basically, it provides a mechanism to project structure onto the data and perform queries written in HQL (Hive Query Language) that are similar to SQL statements.
- Internally, these queries or HQL gets converted to map reduce jobs by the Hive compiler. Therefore, you don't need to worry about writing complex MapReduce programs to process your data using Hadoop.
- Tools to enable easy access to data via SQL, thus enabling data warehousing tasks such as extract/transform/load (ETL), reporting, and data analysis.

Q1: How to enter the HIVE Shell?

Go to the Terminal and type hive, you will see the hive on the prompt.

[cloudera@quickstart Desktop]$ hive

Q2: Create a database

```
create database emp_details;

use emp_details;
```

Q3: How to create Managed Table in HIVE?

create table emp(empno int, ename string, job string, sal int, deptno int)

row format delimited fields terminated by ',';

Q4: How to load the data from LOCAL to HIVE TABLE

Suppose you created a comma separated file in local system named empdetails.txt

1,A,clerk,4000,10

2,A,clerk,4000,30

3,B,mgr,8000,20

4,C,peon,2000,40

5,D,clerk,4000,10

6,E,mgr,8000,50

hive> LOAD DATA LOCAL INPATH

'/home/cloudera/Desktop/empdetails.txt' OVERWRITE INTO TABLE emp;

 # Note: If 'LOCAL' is omitted then it looks for the file in HDFS.

The keyword 'OVERWRITE' signifies that existing data in the table is deleted. If the 'OVERWRITE' keyword is omitted, data files are appended to existing data sets.

Q5: How to check where the managed table is created in hive db

[cloudera@quickstart Desktop]$ hadoop fs -ls /user/hive/warehouse/emp_details.db

Found 2 items

drwxrwxrwx - cloudera supergroup 0 2018-07-24 02:40 /user/hive/warehouse/emp_details.db/emp

drwxrwxrwx - cloudera supergroup 0 2018-07-24 02:28 /user/hive/warehouse/emp_details.db/emp1

Also check the contents inside emp:

[cloudera@quickstart Desktop]$ hadoop fs -ls

/user/hive/warehouse/emp_details.db/emp

Found 1 items

-rwxrwxrwx 1 cloudera supergroup 104 2018-07-24 02:40
/user/hive/warehouse/emp_details.db/emp/empdetails.txt

Now see the contents inside empdetails.txt

[cloudera@quickstart Desktop]$ hadoop fs -cat /user/hive/warehouse/emp_details.db/emp/empdetails.txt

1,A,clerk,4000,10

2,A,clerk,4000,30

3,B,mgr,8000,20

4,C,peon,2000,40

5,D,clerk,4000,10

6,E,mgr,8000,50

Q6:Check the schema of the created table emp?

describe emp;

For a detailed schema use:

describe extended emp;

Q7: How to see all the tables present in database

show tables;

Q8: Select all the enames from emp table

select ename from emp;

Q9: Get the records where name is 'A'

select * from emp where ename='A';

Q10: Count the total number of records in the created table

Count aggregate function is used count the total number of the records in a table.

select count(1) from emp;

OR

Select count(*) from emp;

Q11: Group the sum of salaries as per the deptno select deptno, sum(sal) from emp group by deptno;

Q12: Get the salary of people between 1000 and 2000

select * from emp where sal between 1000 and 2000;

Q13: Select the name of employees where job has exactly 5 characters

hive> select ename from emp where job LIKE '_____';

Q14: List the employee names where job has l as the second character

hive> select ename from emp where job LIKE '_l%';

Q15: Retrieve the total salary for each department select deptno, sum(sal) from emp group by deptno;

Q16: Add a column to the table

alter table emp add COLUMNS(lastname string);

Q17: How to Rename a table

alter table emp rename to emp1;

18: How to drop table

drop table emp; **Q19: How to create External Table:**

Syntax:

CREATE EXTERNAL TABLE <table_name> (column1 data_type, column2 data_type)

row format delimited fields terminated by ','

LOCATION '<table_hive_location>';

Eg. I have created a comma separated file in local machine called extdata.txt

1,2,3

4,5,6

Then I have copied this file in HDFS

[cloudera@quickstart Desktop]$ hadoop fs -put extdata.txt /user/cloudera/

Then I copied this file in a directory named hivedata

[cloudera@quickstart Desktop]$ hadoop fs -cp extdata.txt hivedata

Then I created a table ext1 and loaded it with the data

hive> create external table ext1(a int, b int, c int) > row format delimited fields terminated by ',' > LOCATION '/user/cloudera/hivedata'

> ;

Now check if table is populated with data

hive> select * from ext1;

/user/hive/warehouse/emp_details.db

Found 2 items

drwxrwxrwx - cloudera supergroup 0 2018-07-24 02:40 /user/hive/warehouse/emp_details.db/emp

drwxrwxrwx - cloudera supergroup 0 2018-07-24 02:28 /user/hive/warehouse/emp_details.db/emp1

In above output we saw that two managed tables only being seen and not ext1 which is an external table.

If you drop the external table we will not lose the data in hdfs as shown below.

[cloudera@quickstart Desktop]$ hadoop fs -ls hivedata Found 1 items

-rw-r--r-- 1 cloudera cloudera 12 2018-09-25 20:51 hivedata/extdata.txt

[cloudera@quickstart Desktop]$ hadoop fs -cat

hivedata/extdata.txt

1,2,3

4,5,6

.....WORKING WITH MOVIES DATA SET ……

Q1: Create a database called movies

create database movies;

Q2: Work with database movies

use movies;

Q3: create a table movies_details inside movies database hive> create table movie_details(no int,

> name string,

> year int,

> rating decimal,

> views int)

> row format delimited fields terminated by ',';

Q4: Load the data set of movies from local to hive table hive> LOAD DATA LOCAL INPATH

'/home/cloudera/Desktop/hive_demo/movies_new' INTO table movie_details;

Q5: Check the table created inside database.

hive> show tables;

Q6: Retrieve all the records in movies_details?

hive> select * from movie_details;

Q7: Print all movies between year 1920 and 1990

hive> select * from movie_details where year between 1920 and 1990;

Q8: Select all records where movie name starts from letter c or C

hive> select * from movie_details where name LIKE 'C%' or name LIKE 'c%';

Q9: select all records where movie name starts with The hive> select * from movie_details

where name LIKE 'The%';

Q10: What is the maximum rating of the movie

hive> select max(rating) from movie_details;

Q11: count the number of records

hive> select count(*) from movie_details;

Q12: select rating of the movie School Ties

hive> SELECT name,rating FROM movie_details WHERE name = 'School Ties';

Q13: List all the years with total number of views in each year (hint group by year), restrict the records to 5

hive> select year, sum(views) from movie_details group by year LIMIT 5;

OK

PARTITIONING AND BUCKETING

Q1: Explain the concept of Partitioning and bucketing?

Assume that you are storing information of people in entire world spread across 196+ countries spanning around 500 crores of entries. If you want to query people from a particular country

(Vatican city), in absence of partitioning, you have to scan all 500 crores of entries even to fetch thousand entries of a country. If you partition the table based on country, you can fine tune querying process by just checking the data for only one country partition. Hive partition creates a separate directory for a column(s) value.

Bucketing decomposes data into more manageable or equal parts.

With partitioning, there is a possibility that you can create multiple small partitions based on column values. If you go for bucketing, you are restricting number of buckets to store the data. This number is defined during table creation scripts

Q2: create a database shopping

hive>create database shopping;

use shopping;

Q3: create table (shopping1) inside the database shopping create table shopping1(code INT, item_name string, category string, place string)

> row format delimited fields terminated by ',';

create table shopping1(code INT, item_name string, category string, place string) row format delimited fields terminated by ',';

Q4: Load the data in HIVE table from local

Suppose you have a file shop.txt (see the contents below) in your local machine, you can create a CSV file using gedit command

1,purse,bag,shimla

2,lipstick,cosmetic,delhi

3,bowl,utensils,jammu

4,mobile,electronic gadget,hyderabad

5,skirt,apparel,chennai

6,bed cover,furnishing,chandigarh

7,car,toys,karnal

8,hand purse,bag,solan

9,cream,cosmetic,jhodpur

10,plate,utensils,mohali

11,head phones,electronic gadget,calicut

12,top,apparel,mumbai

13,table cover,furnishing,agra

17,truck,toys,jaipur

18,wallet,bag,solan

19,foundation,cosmetic,jhodpur

20,spoon,utensils,mohali

21,speaker,electronic gadget,calicut

22,suit,apparel,mumbai

33,table sheet,furnishing,agra

24,auto,toys,jaipur

hive> LOAD DATA LOCAL INPATH '/home/cloudera/Desktop/shop.txt' OVERWRITE INTO TABLE shopping1;

Q5: create a partition (shopping3) for table shopping1 and also create 3 buckets inside each partition

hive> create table shopping3(code INT, item_name STRING, place string)

> partitioned by (category string)

> clustered by (place) into 3 buckets

> row format delimited fields terminated by ',' > stored as texfile;

Q4: Populate the partition with data

hive> from shopping1 txn INSERT OVERWRITE TABLE shopping3 PARTITION(category)

select txn.code, txn.item_name, txn.place,

txn.category DISTRIBUTE by category;

Q5: Check your partition

[cloudera@quickstart Desktop]$ hadoop fs -ls

/user/hive/warehouse/shopping.db/

[cloudera@quickstart Desktop]$ hadoop fs -ls

/user/hive/warehouse/shopping.db/shopping3

Found 8 items

drwxrwxrwx - cloudera supergroup 0 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=__HIVE_DEFAU LT_PARTITION__

drwxrwxrwx - cloudera supergroup 0 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=apparel drwxrwxrwx - cloudera supergroup 0 2018-07-24 10:48 /user/hive/warehouse/shopping.db/shopping3/category=bag drwxrwxrwx - cloudera supergroup 0 2018-07-24 10:48 /user/hive/warehouse/shopping.db/shopping3/category=cosmetic

Q6: Check out the buckets for the partition "utensils"?

[cloudera@quickstart Desktop]$ hadoop fs -ls

/user/hive/warehouse/shopping.db/shopping3/category=utensils

Found 3 items

-rwxrwxrwx 1 cloudera supergroup 0 2018-07-24 10:48

/user/hive/warehouse/shopping.db/shopping3/category=utensils/000 000_0

-rwxrwxrwx 1 cloudera supergroup 13 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=utensils/000 001_0

-rwxrwxrwx 1 cloudera supergroup 32 2018-07-24 10:48
/user/hive/warehouse/shopping.db/shopping3/category=utensils/000 002_0

Q7: Check out the contents of a particular bucket suppose 000001_0

[cloudera@quickstart Desktop]$ hadoop fs -cat

/user/hive/warehouse/shopping.db/shopping3/category=utensils/000 001_0

3,bowl,jammu

PIG

Aim: To understand Data Processing Tool – Pig and to execute pig latin commands.

Objectives:

- To read a file from HDFS into Pig environment
- To execute pig commands in grunt shell
- To be familiar with tuples and bags in Pig

Key concept:

Pig is an open-source high level data flow system. It provides a simple language called Pig Latin, for queries and data manipulation, which are then compiled in to MapReduce jobs that run on Hadoop.

Q1: How to enter in grunt shell?

[cloudera@quickstart Desktop]$ pig

Q2: Create two data sets using gedit command in local?

[cloudera@quickstart Desktop]$ cat pigfile.txt

1,2,3

4,5,6

5,7,0

[cloudera@quickstart Desktop]$ gedit pigfile1.txt

2,4,5

2,4,5

4,7,9

Q3: Copy the above files in HDFS?

[cloudera@quickstart Desktop]$ hadoop fs -put pigfile.txt /user/cloudera/

[cloudera@quickstart Desktop]$ hadoop fs -put pigfile1.txt /user/cloudera/

Q4: How to read your (pigfile.txt and pigfile1.txt) data in PIG

```
grunt> a = LOAD '/user/cloudera/pigfile.txt' using PigStorage(',');
grunt> dump a;
```

OUTPUT

```
(1,2,3)
(4,5,6)
(5,7,0)
```

```
grunt> b = LOAD '/user/cloudera/pigfile1.txt' using PigStorage(',');
grunt> dump b;
```

OUTPUT

```
2,4,5
2,4,5
4,7,9
```

NOTE: If we want to specify schema, we can, but pig is flexible in that. The columns can be referred as $0 , $1 and so on. But even if you want to specify schema we can.

Q4: Specify the schema for above two tables?

```
grunt> a = LOAD '/user/cloudera/pigfile.txt' using PigStorage(',') as (a1:int, a2:int, a3:int);
grunt> dump a;
grunt> b = LOAD '/user/cloudera/pigfile1.txt' using PigStorage(',') as (b1:int, b2:int, b3:int);
grunt> dump b;
```

Q5: Check the schema of the two tables?

```
grunt> describe a;
```

OUTPUT

```
a: {a1: int,a2: int,a3: int}
```

```
grunt> describe b;
```

OUTPUT

```
b: {b1: int,b2: int,b3: int}
```

Q6: Combine the two tables

```
grunt> c= union a,b;
grunt> dump c
```

OUTPUT

(2,4,5)

(2,4,5)

(4,7,9)

(1,2,3)

(4,5,6)

(5,7,0)

Q7: Split the c data set into two different relations eg. d and e? E.g. I want one data set where $0 is having value 1 and other data set where value of $0 is 4

grunt> SPLIT c INTO d IF $0 == 1 , e IF $0 == 4;

dump d;

OUTPUT

(1,2,3) :- here this is the row starting with 1.

dump e;

(4,7,9) :- here this is the row starting with 4.

(4,5,6) :- here this is the row starting with 4.

Q8: Do filtering on data set c where $1 is greater than 6?

grunt> f = FILTER c BY $1 > 6;

grunt> dump f

OUTPUT

(5,7,0) :: here $1 means column 1, so column 1>6

(4,7,9) :: here $1 means column 1, so column 1>6

Q9: Group data set c by $2?

grunt> g = GROUP c by $2;

grunt> dump g;

OUTPUT

(0,{(5,7,0)})

(3,{(1,2,3)})

(5,{(2,4,5),(2,4,5)})

(6,{(4,5,6)})

Q: 10: Select column 1 and 2 from dataset a ?

grunt> s1 = foreach a generate $1,$2;

grunt> dump s1;

OUTPUT

(2,3)

(5,6)

(7,0)

Q11: Store the above result in HDFS?

grunt> store s1 into '/user/cloudera/pigresult';

Q12: check the file written in HDFS

[cloudera@quickstart Desktop]$ hadoop fs -ls /user/cloudera/pigresult/

OUTPUT

Found 2 items

-rw-r--r-- 1 clouderacloudera 0 2018-09-19 03:50 /user/cloudera/pigresult/_SUCCESS

-rw-r--r-- 1 clouderacloudera 12 2018-09-19 03:50 /user/cloudera/pigresult/part-m-00000

Now see what's inside part-m-00000

[cloudera@quickstart Desktop]$ hadoop fs -cat /user/cloudera/pigresult/part-m-00000

2 3

5 6

7 0

Sqoop

Aim: To understand the concept of data ingestion tool "Sqoop"

Objective:

- To import structured data from MYSQL to HDFS
- To export text file (structured data) to MYSQL

Key concept:

- Apache Sqoop is a tool in *Hadoop ecosystem* which is designed to transfer data between *HDFS* (Hadoop storage) and relational database servers like mysql, Oracle RDB, SQLite, Teradata, Netezza, Postgres etc.
- Apache Sqoop imports data from relational databases to HDFS, and exports data from HDFS to relational databases. It efficiently transfers bulk data between Hadoop and external datastores such as enterprise data warehouses, relational databases, etc.
- *This is how Sqoop got its name – "SQL to Hadoop & Hadoop to SQL".*

Q1: How to enter in mysql CLI in cloudera

[cloudera@quickstart Desktop]$ mysql -uroot -pcloudera

Q2: Create a database

mysql> create database sqoop_db;

Query OK, 1 row affected (0.27 sec)

Q3: Select the database created

mysql> use sqoop_db;

Q4: Create a table inside created database

mysql> create table emp(empno int primary key, ename varchar(10),job varchar(9),mgr int,hiredate date,sal int,deptno int);

Query OK, 0 rows affected (0.08 sec)

Q5: Insert records in the table created

mysql> INSERT INTO emp VALUES (1,'ABC','teacher',100,2013-1-12, 50000,10);

mysql> INSERT INTO emp VALUES (2,'XYZ','teacher',200,2014-10-2,650000,20);

mysql> INSERT INTO emp VALUES (3,'PQRS','teacher',200,2010-10-2,640000,30);

Q6: Check the contents in the table?

mysql> select * from emp;

```
+-------+--------+---------+------+------------+--------+--------+
| empno | ename  | job     | mgr  | hiredate   | sal    | deptno |
+-------+--------+---------+------+------------+--------+--------+
|     1 | ABC| teacher |  100 | 2013-1-12|  50000 |     10 |
|     2 | XYZ | teacher |  200 | 2014-10-2| 650000 |     20 |
|     3 |PQRS  | teacher |  200 | 2010-10-2 | 640000 |     30 |
+-------+--------+---------+------+------------+--------+--------+
```

3 rows in set (0.00 sec)

Sqoop – Import

Q7: Import this table in HDFS

[cloudera@quickstart Desktop]$ sqoop import --connect jdbc:mysql://localhost/sqoop_db --username root --password cloudera --table emp --target-dir /user/cloudera/scoop_data

Q8: Check if the data is imported in HDFS

[cloudera@quickstart Desktop]$ hadoop fs -ls /user/cloudera/scoop_data

Found 4 items

```
-rw-r--r--   1 cloudera cloudera          0 2018-10-02 10:26 /user/cloudera/scoop_data/_SUCCESS
-rw-r--r--   1 cloudera cloudera         34 2018-10-02 10:26 /user/cloudera/scoop_data/part-m-00000
-rw-r--r--   1 cloudera cloudera         36 2018-10-02 10:26 /user/cloudera/scoop_data/part-m-00001
-rw-r--r--   1 cloudera cloudera         35 2018-10-02 10:26 /user/cloudera/scoop_data/part-m-00002
```

Q9: Open the partitions and check for the records

[cloudera@quickstart Desktop]$ hadoop fs -cat /user/cloudera/scoop_data/part-m-00000

1, Veena ,teacher,100, 2013-1-12,50000,10

[cloudera@quickstart Desktop]$ hadoop fs -cat /user/cloudera/scoop_data/part-m-00001

2,sheena,teacher,200, 2014-10-2,650000,20

[cloudera@quickstart Desktop]$ hadoop fs -cat /user/cloudera/scoop_data/part-m-00002

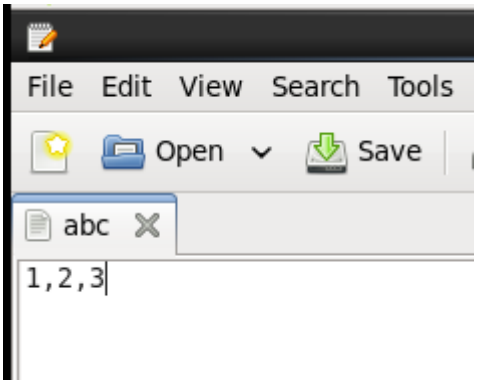
3,meena,teacher,200, 2010-10-2,640000,30

Sqoop – Export

Q10: Export data(like csv) in MYSQL in a table?

Create a file in local in the Desktop, suppose abc.txt

gedit abc.txt



Next put this abc.txt in HDFS

[cloudera@quickstart Desktop]$ hadoop fs -put abc.txt /user/cloudera/

Come to MYSQL prompt

Create a table suppose abc_table

mysql> use sqoop_db;

sql> create table abc_table(a int, b int, c int);

Query OK, 0 rows affected (0.13 sec)

In terminal type the following command to export abc.txt in abc_table table

[cloudera@quickstart Desktop]$ sqoop export --connect jdbc:mysql://localhost/sqoop_db --username root --password cloudera --table abc_table --export-dir /user/cloudera/abc.txt

Q11: Check in MYSQL if data is exported from HDFS into abc_table

mysql> select * from abc_table;

```
+------+------+------+
| a    | b    | c    |
+------+------+------+
|    1 |    2 |    3 |
+------+------+------+
```

1 row in set (0.01 sec)

V N R V J I E T	Sheet No……
Name of the Laboratory: _____	**Name of the Experiment:**_____ _____ **Experiment No.**_____**Date**_____

Data Visualization Using Tableau

Aim: To understand the concept and working of data visualisation tool "Tableau"

Objective:

- To visualise data using various available charts.
- To compare sheets/charts using 'dashboard' feature.
- To implement the 'story' feature for narrative insights.

Key Concept:

Tableau is an excellent data visualization and business intelligence tool used for reporting and analyzing vast vo lumes of data. It helps users create different charts, graphs, maps, dashboards, and stories for visualizing and an alysing data, to help in making business decisions. Tableau has a lot of unique, exciting features that make it on e of the most popular tools in business intelligence (BI).

Key Features of tableau-

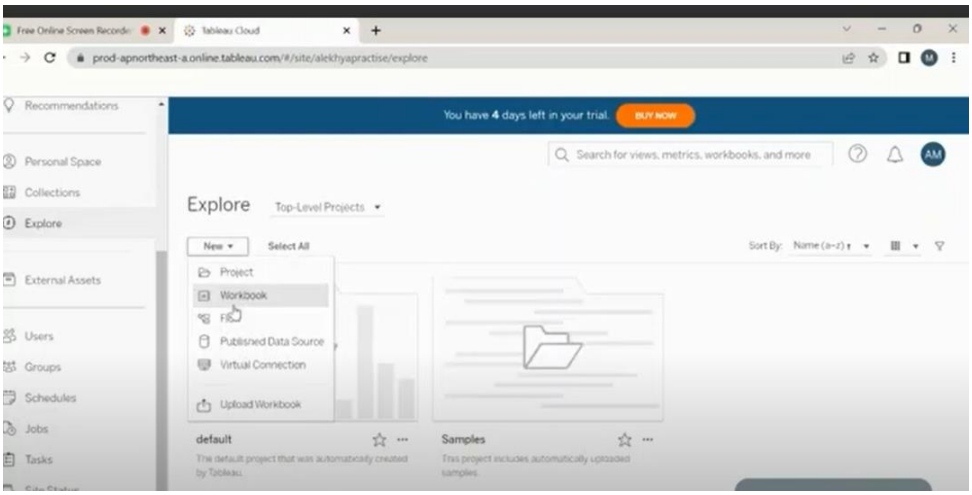
- Tableau supports powerful data discovery and exploration that enables users to answer important questio ns in seconds.
- No prior programming knowledge is needed; users without relevant experience can start immediately wi th creating visualizations using Tableau.
- It can connect to several data sources that other BI tools do not support. Tableau enables users to create r eports by joining and blending different datasets.
- Tableau Server supports a centralized location to manage all published data sources within an organizati on.

Procedure:

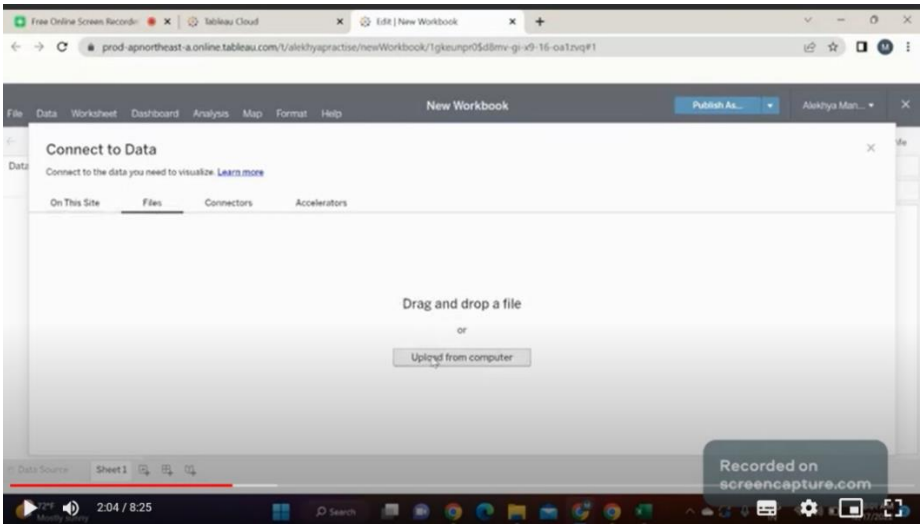
Step 1: Go to www.tableau.com and under products navigate to "Tableau Cloud' and login to your tableau cloud account. If you are a new user, please click on start free trail and fill in your details to get

access.

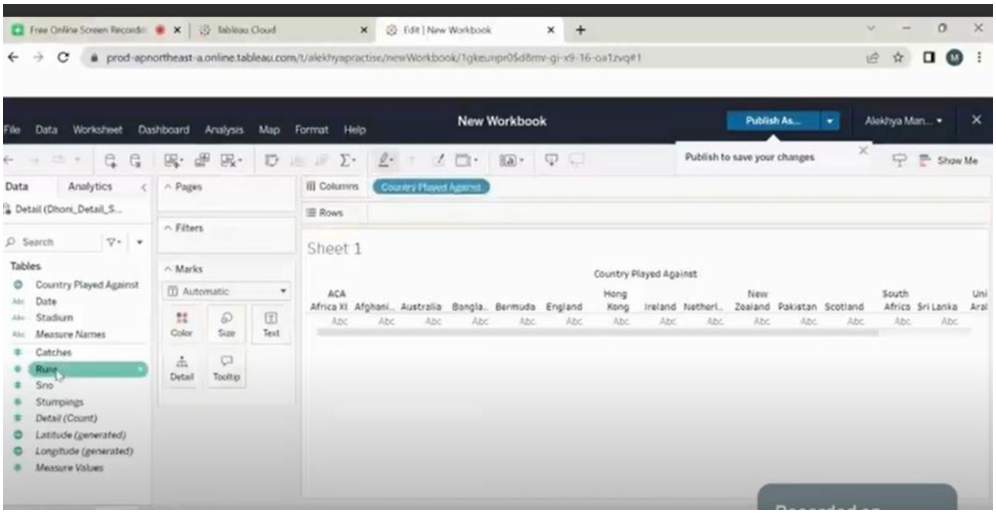
Step 2: Click on "Explore" and then from the drop-down, select new workbook.



Step 3: A pop-up window is displayed, navigate to 'Files' and upload Dhoni_details_statistics.csv from your device.

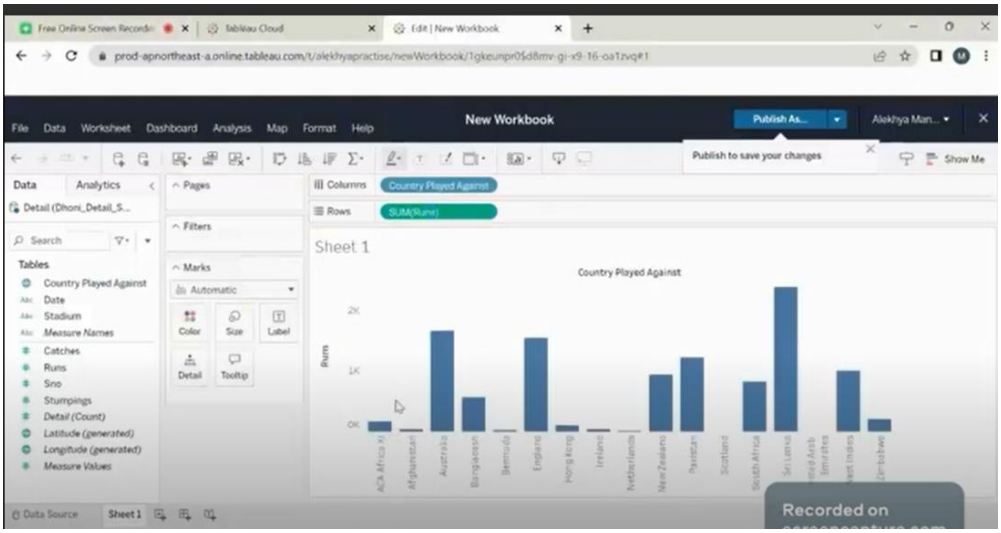


Step-4: Drag and drop 'Country Played Against to Columns field

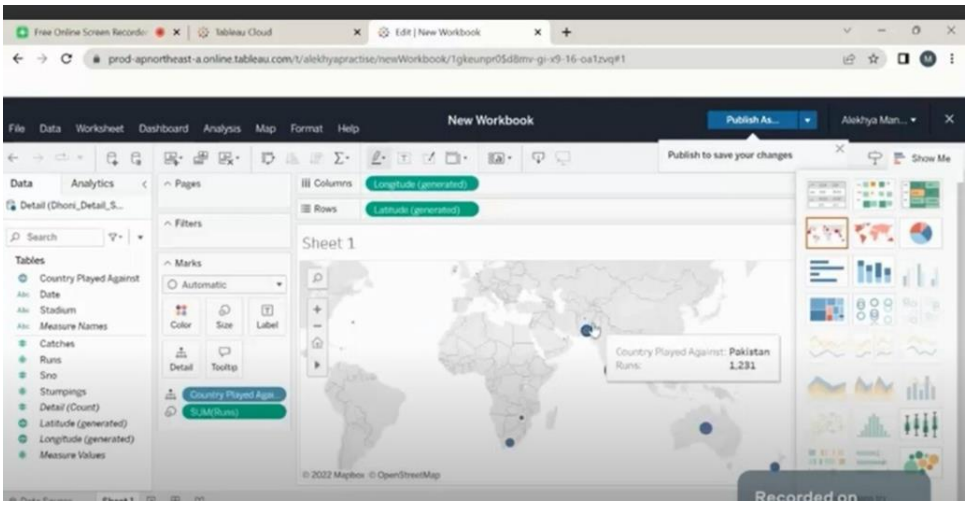


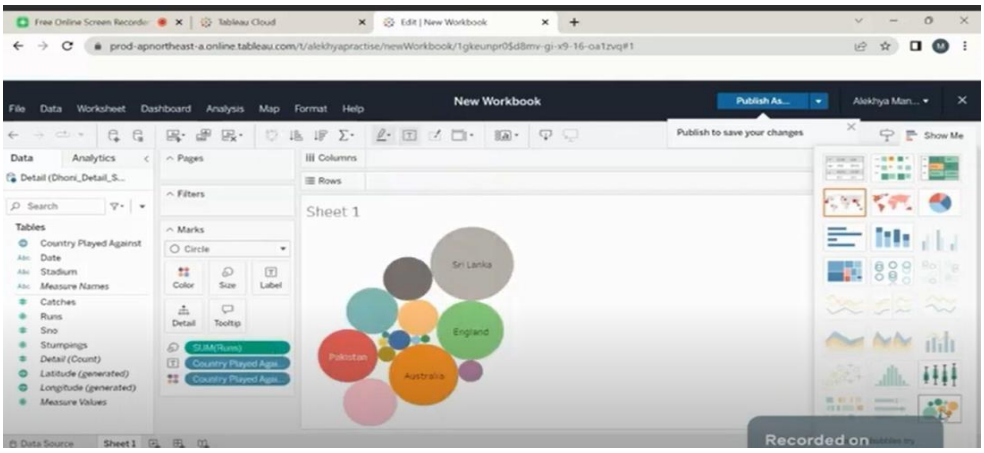
You can already see that a bar graph on the data selected above is plotted.

Step 5: To get more such visualization plots, click on "show me" in the top right corner. Based on the dimension ality of the data the plots need to be selected.



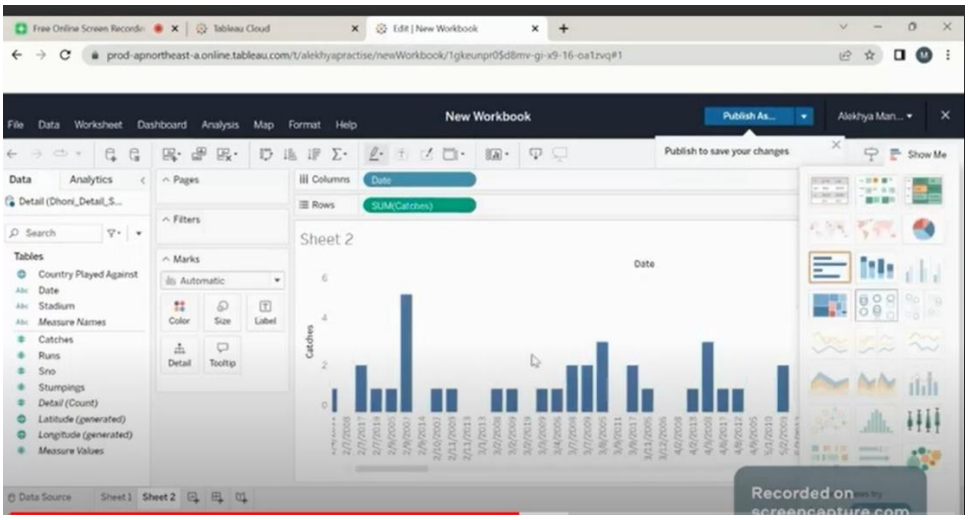
When the same data is visualized using a world map, the chart is as follows which highlights the countries and runs with variable internsity.





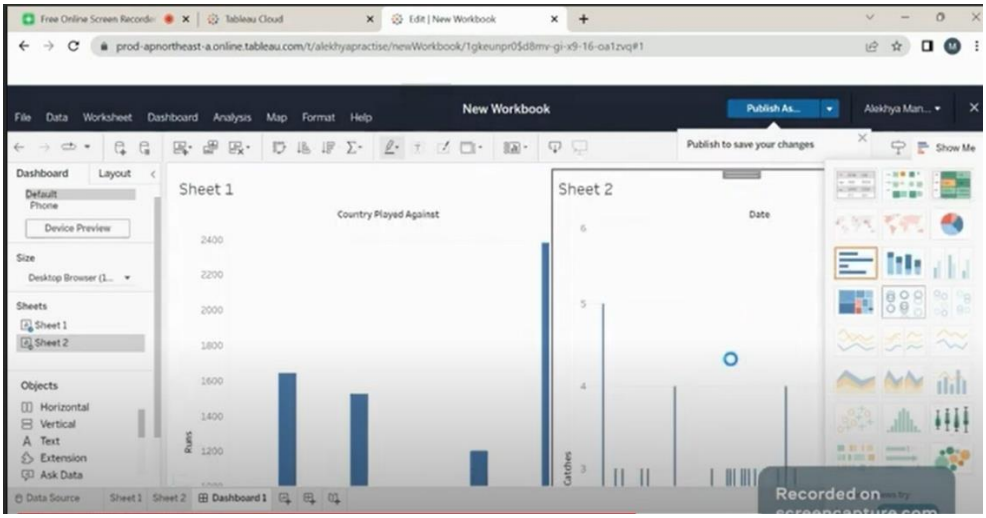
Step 6: Similarly to create a new worksheet, click on 'new sheet' option in the bottom left.

Step-7: Similar to step-3, drag and drop "Date" and "Catches" onto the column and row value of workspace resp ectively.

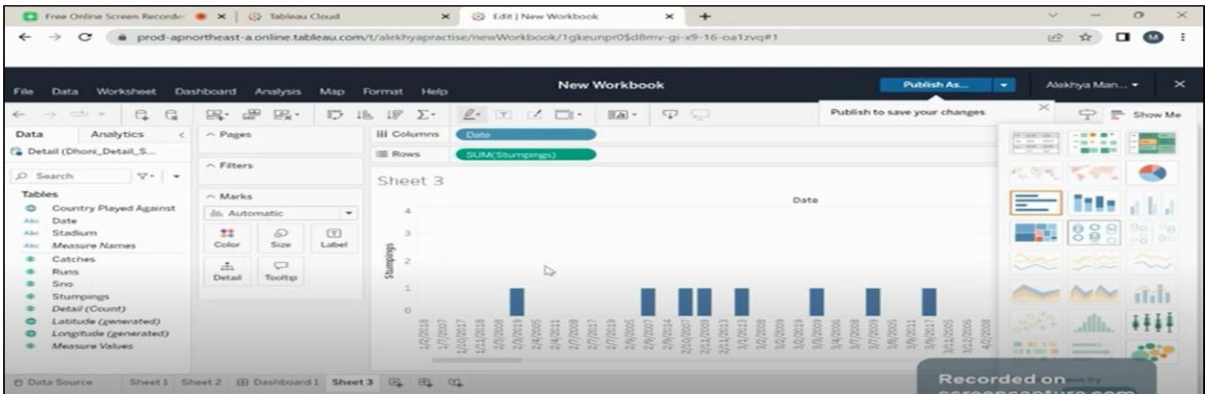


Step 8: To view both the sheets at a time, we will create a dashboard. Just beside 'new sheet' option, click on 'ne w dashboard' button.

Then drag sheet-1 and then sheet-2 right next to it onto the workspace.



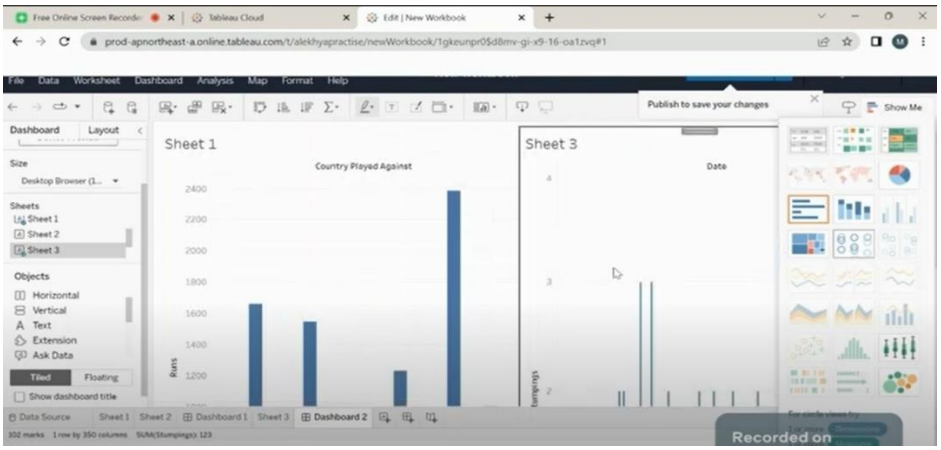
Step-9: Now let us create another sheet in the similar process as above with fields "Date" and "Stumpings".



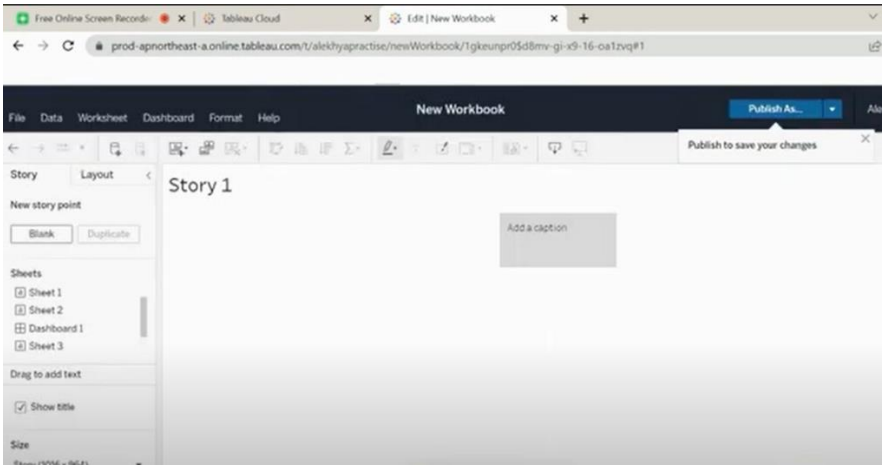
The dashboard feature of tableau is a way of displaying various types of visual data in one place. Usually, a das hboard is intended to convey different, but related information in an easy-to-digest form.

Step 11: Let us explore another feature called "Story" in tableau.

Story is a sequence of visualizations that work together to convey information. You can create stories to tell a d ata narrative, provide context, demonstrate how decisions relate to outcomes, or to simply make a compelling ca se.



On the bottom left corner of the window just beside 'new dashboard' option we also have 'new story' option. Cli ck on the 'new story' button.



You are now able to see all the available worksheets and dashboards in our workbook. You can click on anyone of them to view.