

Big Data Analytics – Advanced (Industry Assignment – 2)

**Vaishnavi Bhambure**

**vaishanvisbhambure@gmail.com**

**Title: Master data management**

# Create a directory on HDFS with the name 'PublishingHouse'.

step 1: To configure docker

/home/user34/Desktop/BigData$ sudo docker-compose up

step 2: Now we need to enter into hadoop cluster using command

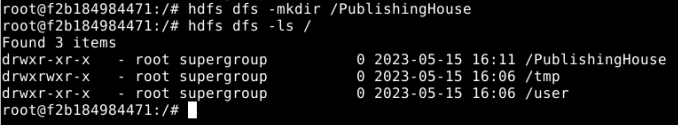
/home/user34/Desktop/BigData$ sudo docker exec -it namenode bash

step 3: Now we can create directory 'PublishingHouse' in HDFS

root@573d8339fa78:/# **hdfs dfs -mkdir/PublishingHouse**

step 4: Display list of directories and check whether 'PublishingHouse' created or not

root@573d8339fa78:/# **hdfs dfs -ls /**

****

# Load the dataset to the directory created using HDFS commands.

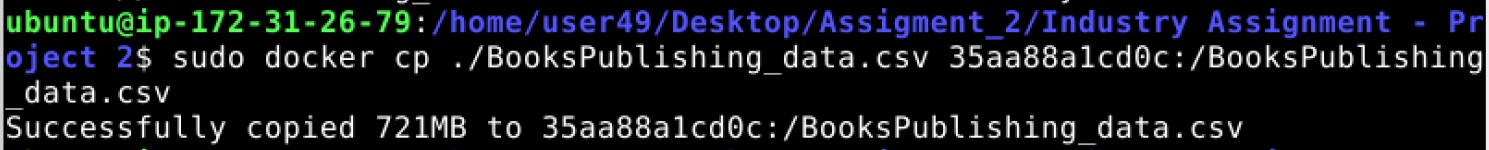
step 1: Now go to the path where dataset is and perform command to display it in local system

/home/user34/Desktop/Assignment2$ docker container ls

/home/user34/Desktop/Assignment2$ **sudo docker cp**

# ./BooksPublishing\_data.csv a0a9d611810b:/BooksPublishing\_data.csv

# 



step 2: Now go to root node and check whether dataset is present or not

root@a0a9d611810b:/# ls

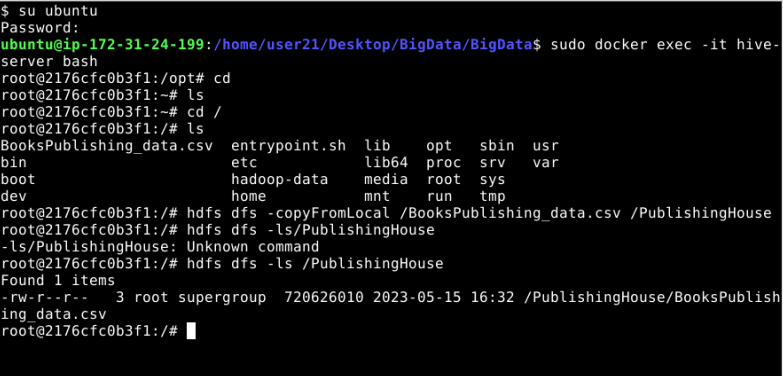
step 4: Copy dataset to 'PublishingHouse' directory

root@a0a9d611810b:/# **hdfs dfs -copyFromLocal /BooksPublishing\_data.csv**

# /PublishingHouse/

step 5: To check whether dataset copied to directory

root@a0a9d611810b:/# **hdfs dfs -ls /PublishingHouse**



# Use PySpark programming to load data from HDFS and create a dataframe to analyze the fields/columns for analysis by displaying all fields.

step 1: To start with spark

ubuntu@ip-172-31-25-38:/home/user34/Desktop/BigData$ sudo docker exec

-it spark-master bash

step 2: Now in bash type code to load data from hdfs and create dataframe

bash-4.4# **vi first1.py**

# Program:

from pyspark import SparkContext,SparkConf from pyspark.conf import SparkConf

from pyspark.sql import SparkSession

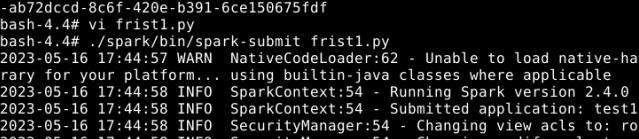
ss=SparkSession.builder.appName("test1").getOrCreate() from pyspark.sql.functions import col

ss.sparkContext.setLogLevel("ERROR")

df=ss.read.csv("hdfs://namenode:8020/PublishingHouse/BooksPublishing\_dat a.csv",inferSchema=True,sep='\t',header=True)

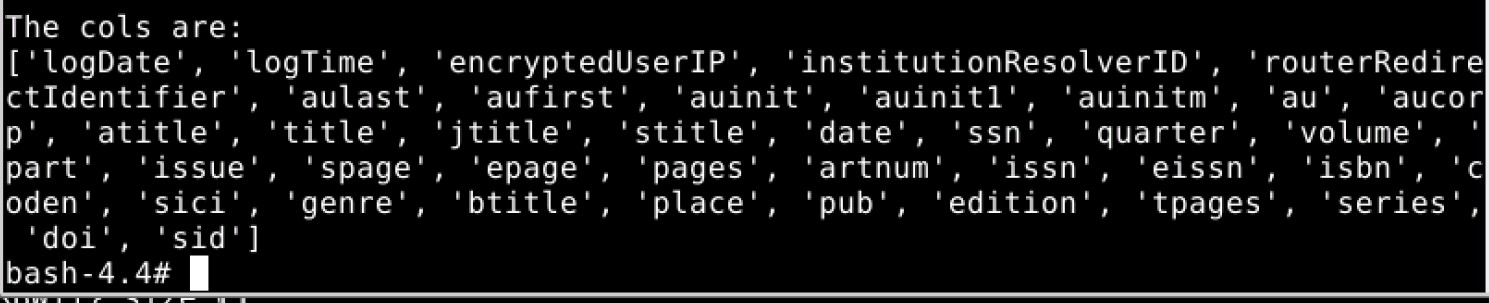
df.show(10) df.printSchema() print("\nThe cols are:") print(df.columns)

Here dataframe df is created and fields in dataframe are displayed along with top 10 rows of dataframe and schema of dataframe



step 3: To display output use command

bash-4.4# **./spark/bin/spark-submit first1.py**



# Use Select query to extract relevant information related for analysis:

* 1. Count and display publisher name wise contents developed.
  2. Count and display Author wise contents developed.
  3. Find Top 5 Books as per its volumes (No. Of Pages).
  4. Find highest no. of contents published by an author. e)Partition by contents developed and save the table.

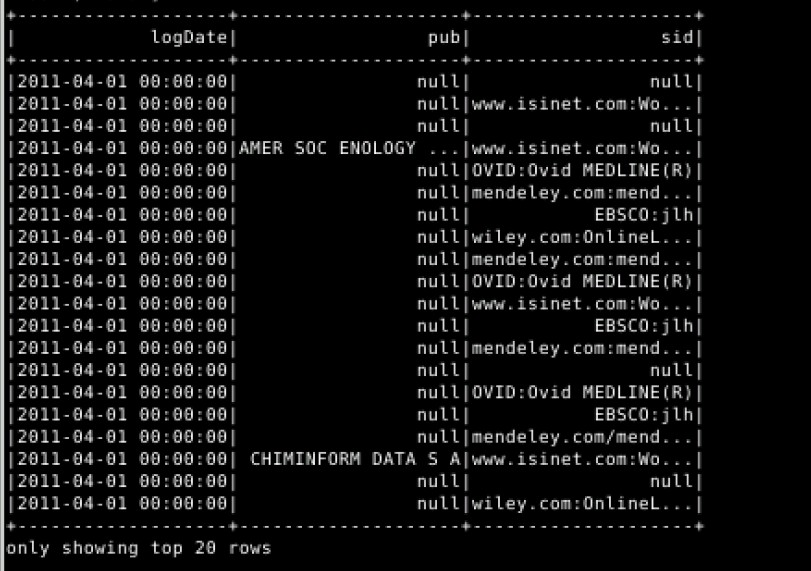
1. Count and display publisher name wise contents developed.

Import necessary packages to perform sql select queries,

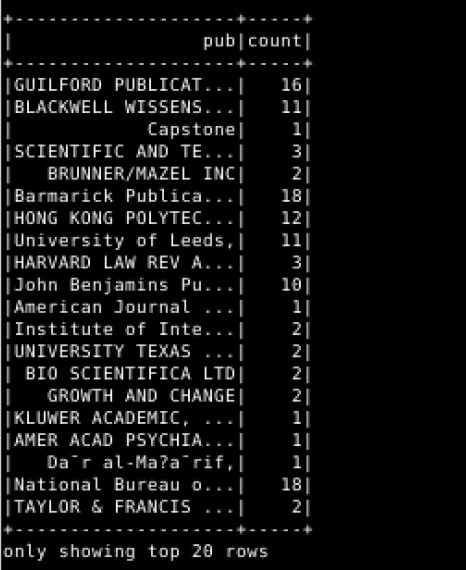
# from pyspark.sql.functions import col,asc,desc,sum

Query 1:

# df.select(col("logDate"),col("pub"),col("sid")).show()



**df.groupby("pub").count().show()**



1. Count and display Author wise contents developed.

Query 2:

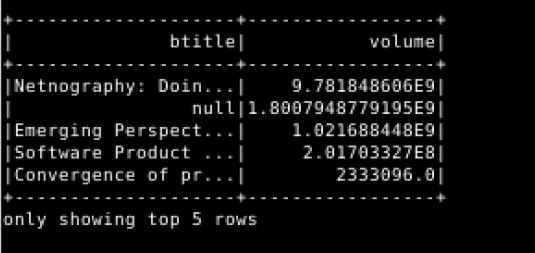
# df.groupBy("aufirst").count().show()

1. Find Top 5 Books as per its volumes (No. Of Pages).

Query 3:

# df.groupby("btitle").agg(sum("volume").alias("volume")).sort(desc("volume"

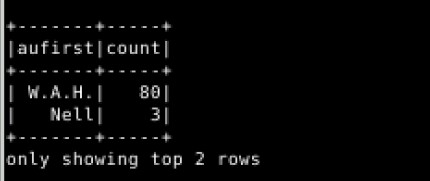
**)).show(5)**



1. Find highest no. of contents published by an author.

Query 4:

# df.groupBy("aufirst").count().show(2)



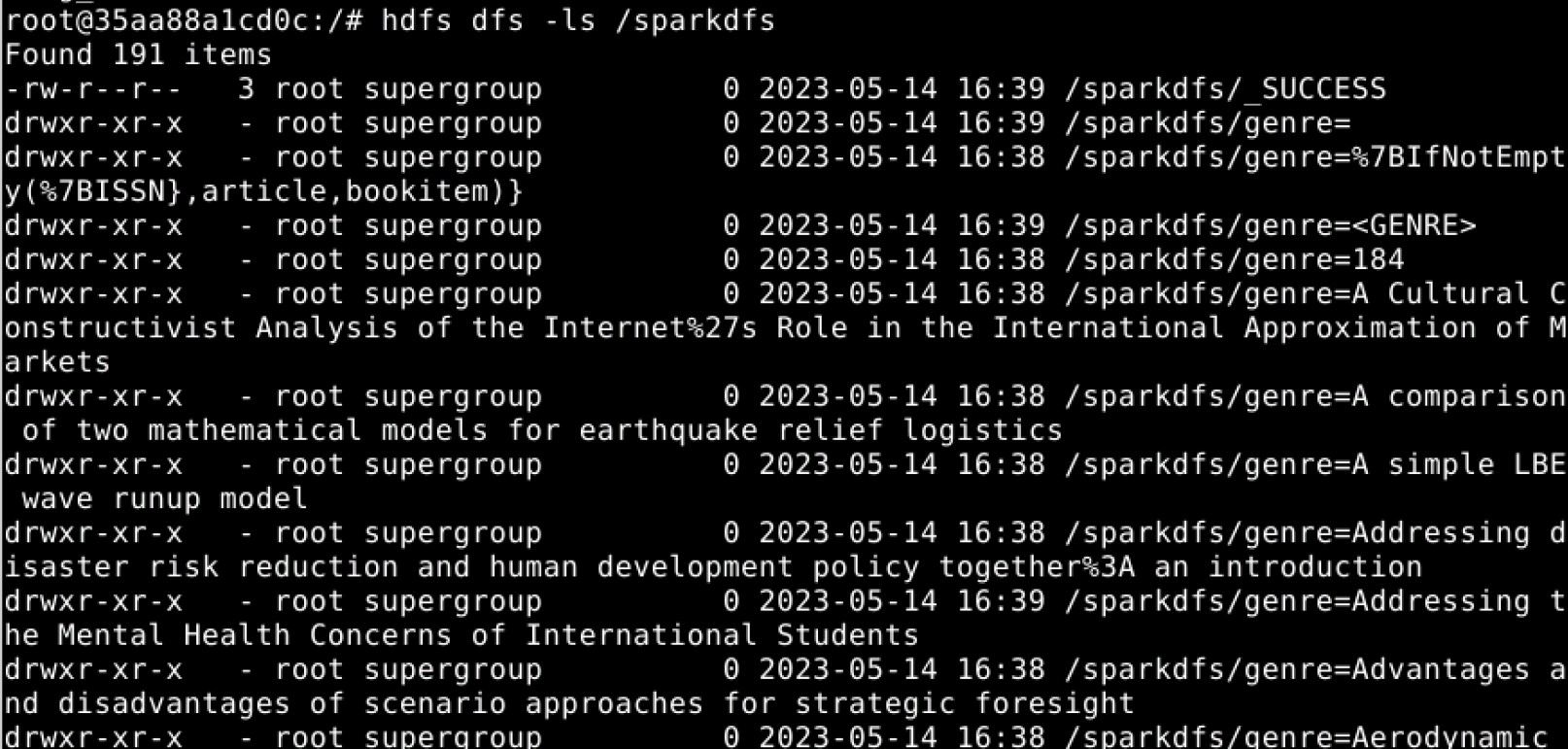
1. Partition by contents developed and save the table.

Query 5:

Write the below command to save

# Part\_df = df.write.partitionBy(“genre”)

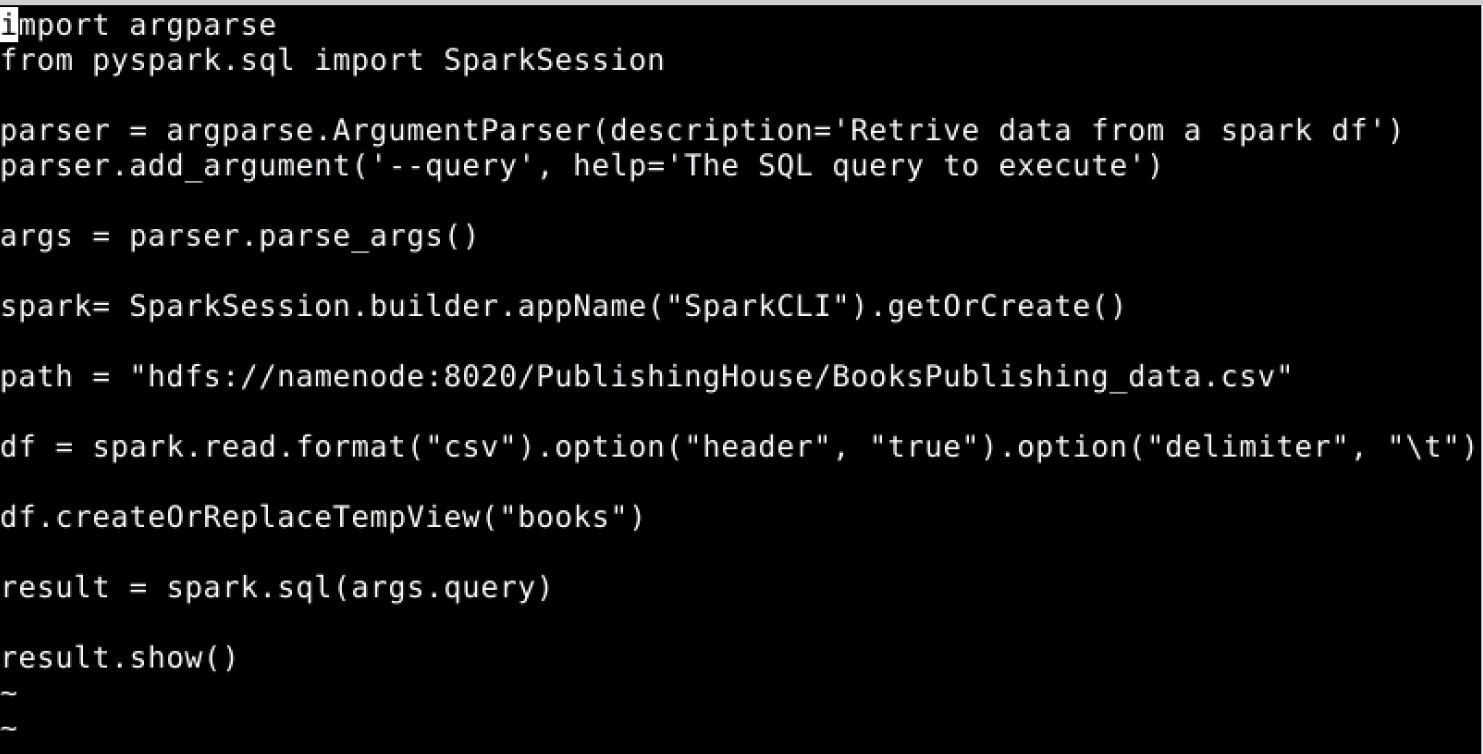
**Part\_df.format(“CSV”).save(“hdfs://namenode:8020/sparkdfs”)**



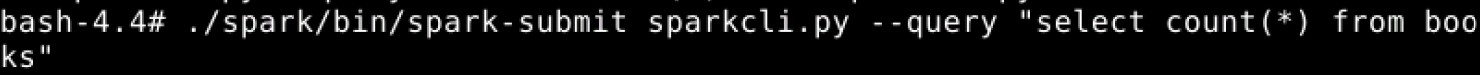
# Q.5) Develop a CLI based interface for any ad-hoc query resolution, where users can type Spark-SQL queries and it would display the output.

We will create a file “sparkcli.py” to write the code using the command

# vi sparkcli.py



We execute the file using the command



The output generated should be count as shown below

