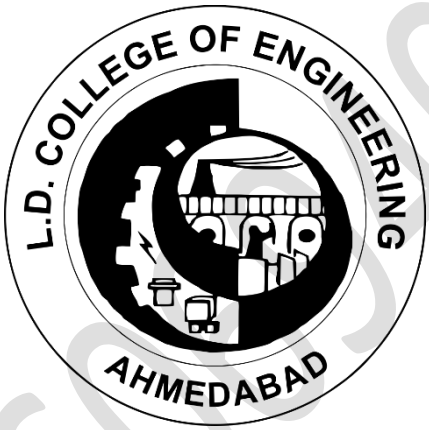
L. D. College of Engineering

Ahmedabad – 380015



Lab Manual

Subject Name: Big Data Tools

(629407)

MCA Semester – 2

Academic year: 2020-21

Certificate

This is to certify that **Mr. Parth Kukadiya** having enrolment no. **205160694013** of **MCA Semester – 2** has satisfactorily completed course in **Big Data Tools** at L. D. College of Engineering, Ahmedabad – 380015.

Date of Submission: 30/07/2021

Staff in-charge: Prof. Vidisha Thakkar

Head of Department:

Index

Tasks	Topics (Programs) to be Completed	Date	Faculty Sign
1.	Case study on Rolls Royce.		
2.	Hadoop installation and basics of HDFS.		
3.	Managing the file system in HDFS.		
4.	Map Reduce exercise		
5.	Hive Practice HQL		
6.	Working with NoSQL Databases. (MongoDB)		
7.	Working with NoSQL Databases. (MongoDB) - 2		

Case Study: Rolls-Royce

Rolls-Royce: Using Big Data to drive manufacturing success

Rolls-Royce is a British luxury car and later an aero-engine manufacturing business established in 1904 in Manchester, United Kingdom.

It is an extremely high-tech industry where failures and mistakes can cost billions – and human lives. It's no surprise then that the company – which split from its automobile manufacturer parent company following insolvency in 1971 – has wholeheartedly embraced Big Data.

The company has spoken about a time when it believes ships may pilot themselves, making logistical decisions such as whether to alter course due to weather or ocean conditions. It is anticipating that there will be a time when computers will simply be able to make these decisions more efficiently than humans, and it will be a wise financial move – as well as safer – to listen to them.

Vision and how they collect the data

The engine and propulsion systems are all fitted with hundreds of sensors which record every tiny detail about their operation and report any changes in data in real-time to engineers who will decide the best course of action such as scheduling maintenance or dispatching engineering teams should the problem require it. And that is just a tiny part of what kind of data they collect.

Paul Stein, the company's chief scientific officer, explained that Rolls Royce puts Big Data processes to use in three key areas of its operations.

He says, "we have huge clusters of high-power computing which are used in the design process. We generate tens of terabytes of data on each simulation of one of our jet engines. We then have to use some pretty sophisticated computer techniques to look into that massive data set and visualize whether that particular product we've designed is good or bad. Visualizing Big Data is just as important as the techniques we use for manipulating it. It decreases development time and improves the quality and performance."

For example, the company is able to generate half a terabyte of manufacturing data on each individual fan blade they produce. As they manufacture 6,000 of those fan-blades a year, that's an incredible three petabytes of data just from the manufacture of one component.

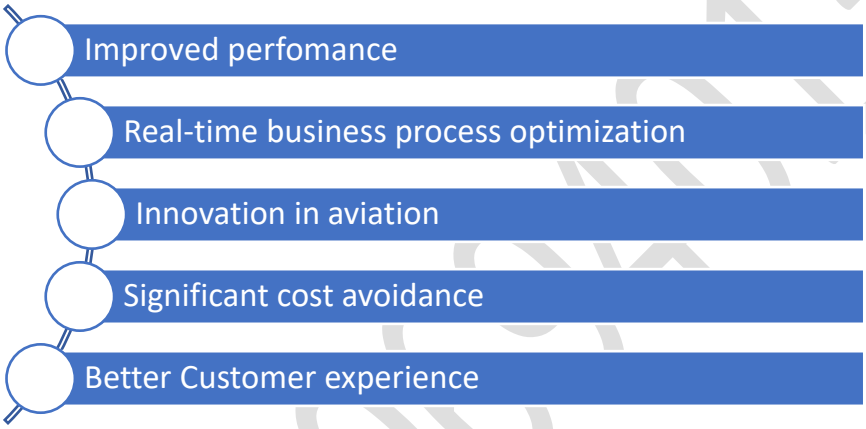
Later Rolls-Royce launched R^2 Data Labs – the company's in-house data innovation catalyst in 2017. R^2 Data Labs has a significant presence in India: the digital platform capability has been built in partnership with Tata Consultancy Services (TCS). The platform enables data to be captured, shared and analysed more easily across all areas of Rolls-Royce, so that new products and services can be developed at pace.

Big Data Processing

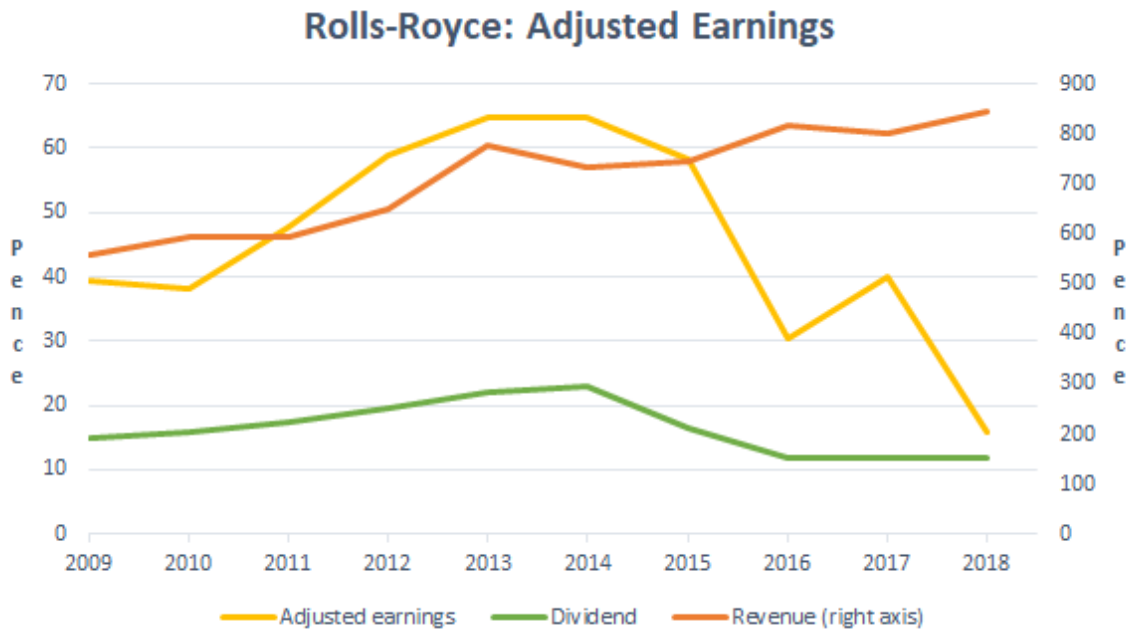
Rolls-Royce maintains a robust and secure private cloud facility with a proprietary storage approach, as well as a data lake for offline investigations.

Smart Discovery a suite of technologies enabling SMEs (Subject Matter Expert) to perform data analytics. Instead of using automated machine learning to help data scientists expedite analysis, they flipped the approach and gave SMEs an intuitive tool that automates the heavy lifting of data science. With the collaborative effort of the Technology team and the Rolls-Royce@NTU Corporate Laboratory at Singapore's Nanyang Technological University, they incorporated AI methodologies and best practices to drive data analytics in Smart Discovery.

Achieved goals using Big Data Tools



Effect on Revenue



We can clearly see how Rolls-Royce as a company serving in different market segments generated revenue in a decade, from 2009 to 2018.

Ideas and insights you can steal

Rolls-Royce serves as an inspirational example of an industrial giant transitioning to the new age of data-enabled efficiency – which shows how any company, regardless of its industry, can and should adapt to the data age. And their commitment to Big Data right across the company, from product design to aftercare support, is something that every company should look to emulate. As Stein says of Big Data, "It forms a big part of our present but is going to form an even bigger part of our future."

Practical - 2

Task 1: Identify the local file system in your Linux OS system. (Command followed by the output screenshot is expected).

Solution: **df -Th**

```
hirwuser150430@ip-172-31-45-217:~$ df -Th
Filesystem     Type      Size  Used Avail Use% Mounted on
udev           devtmpfs  2.0G  8.0K  2.0G   1% /dev
tmpfs          tmpfs     396M  476K  395M   1% /run
/dev/xvda1     ext4       99G   28G   67G  30% /
none           tmpfs     4.0K     0  4.0K   0% /sys/fs/cgroup
none           tmpfs     5.0M     0  5.0M   0% /run/lock
none           tmpfs     2.0G     0  2.0G   0% /run/shm
none           tmpfs     100M     0  100M   0% /run/user
hirwuser150430@ip-172-31-45-217:~$
```

Task 2: Study of Hadoop installation. Study of Hadoop configurations files. Enlist the contents of the files core-site.xml and hdfs-site.xml

Solution:

1) Content of core-site.xml

```
<property>
        <name>hadoop.tmp.dir</name>
        <value>/data/hdfs/tmp</value>
        <description>Where Hadoop will place all of its working files</description>
</property>
<property>
        <name>fs.defaultFS</name>
        <value>hdfs://master:9000</value>
        <description>Where HDFS NameNode can be found on the network</description>
</property>
<property>
        <name>hadoop.proxyuser.hduser.groups</name>
        <value>*</value>
        <description>
```

```
                What user groups are allow to connect to the HDFS proxy.
                * for all.</description>
        </property>
        <property>
                <name>hadoop.proxyuser.hduser.hosts</name>
                <value>*</value>
                <description>
                        What user hosts are allow to connect to the HDFS proxy.
                        * for all.
                </description>
        </property>
```

2) Content of hdfs-site.xml

```
        <property>
                <name>dfs.replication</name>
                <value>2</value>
                <description>The default replication factor of files on HDFS</description>
        </property>
        <property>
                <name>dfs.block.size</name>
                <value>16777216</value>
                <description>The default block size in bytes of data saved to HDFS</description>
        </property>
        <property>
                <name>dfs.namenode.rpc-bind-host</name>
                <value>0.0.0.0</value>
                <description>
                        controls what IP address the NameNode binds to.
                        0.0.0.0 means all available.
                </description>
        </property>
```

```
        <property>
                <name>dfs.namenode.servicerpc-bind-host</name>
                <value>0.0.0.0</value>
                <description>
                        controls what IP address the NameNode binds to.
                        0.0.0.0 means all available.
                </description>
        </property>
        <property>
                <name>dfs.namenode.http-bind-host</name>
                <value>0.0.0.0</value>
                <description>
                        controls what IP address the NameNode binds to.
                        0.0.0.0 means all available.
                </description>
        </property>
        <property>
                <name>dfs.namenode.https-bind-host</name>
                <value>0.0.0.0</value>
                <description>
                        controls what IP address the NameNode binds to.
                        0.0.0.0 means all available.
                </description>
        </property>
        <property>
                <name>nfs.dump.dir</name>
                <value>/tmp/.hdfs-nfs</value>
                <description>A temporary working directory for files coming into the HDFS
proxy.</description>
        </property>
        <property>
```

```
        <name>nfs.metrics.percentiles.intervals</name>
        <value>100</value>
        <description>
                Enable the latency histograms for read, write and commit requests.
                The time unit is 100 seconds in this example.
        </description>
</property>
<property>
        <name>nfs.exports.allowed.hosts</name>
        <value>* rw</value>
        <description>Host permissions for connecting to the proxy.</description>
</property>
<property>
        <name>dfs.permissions</name>
        <value>true</value>
        <description>Enforce permissions</description>
</property>
<property>
        <name>dfs.permissions.supergroup</name>
        <value>hadoop</value>
        <description>The name of the group of Hadoop super-users.</description>
</property>
```

Task 3: Study and run following commands on the hadoop cluster and show the output.

JPS: to check out all the **Hadoop** daemons like DataNode, NodeManager, NameNode, and ResourceManager that are currently running on the machine.

```
hirwuser150430@ip-172-31-45-217:~$ jps
27606 Jps
```

Fsck: to check health of the HDFS.

```
hirwuser150430@ip-172-31-45-217:~$ hdfs fsck /user/hirwuser150430/MYNEWDIRECTORY -files -blocks -locations
Connecting to namenode via http://ec2-54-92-244-237.compute-1.amazonaws.com:50070
FSCK started by hirwuser150430 (auth:SIMPLE) from /172.31.45.217 for path /user/hirwuser150430/MYNEWDIRECTORY at
Sat Jun 12 07:23:13 UTC 2021
/user/hirwuser150430/MYNEWDIRECTORY <dir>
/user/hirwuser150430/MYNEWDIRECTORY/dwp-payments-april10.csv 3326129 bytes, 1 block(s):  OK
0. BP-2125152513-172.31.45.216-1410037307133:blk_1075416868_1676131 len=3326129 Live_repl=2 [DatanodeInfoWithStor
age[172.31.46.124:50010,DS-fe24aecb-f56f-4c9c-8cf9-a3b1259bc0d0,DISK], DatanodeInfoWithStorage[172.31.45.216:5001
0,DS-d0d9eb5c-f35f-4a12-bfdf-544085d693a3,DISK]]

Status: HEALTHY
 Total size:    3326129 B
 Total dirs:    1
 Total files:   1
 Total symlinks:                0
 Total blocks (validated):      1 (avg. block size 3326129 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    3
 Average block replication:     2.0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          3
 Number of racks:               1
FSCK ended at Sat Jun 12 07:23:13 UTC 2021 in 1 milliseconds


The filesystem under path '/user/hirwuser150430/MYNEWDIRECTORY' is HEALTHY
```

Touchz: to create an empty file.

```
hirwuser150430@ip-172-31-45-217:~$ hdfs dfs -touchz emptyexample.txt
hirwuser150430@ip-172-31-45-217:~$
```

copyFromlocal: to copy a file from local file system to HDFS.

```
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -copyFromLocal /hirw-starterkit/hdfs/commands/dwp-payments-april10.csv ILOVEBIGDATA
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -ls ILOVEBIGDATA
Found 1 items
-rw-r--r--   3 hirwuser150430 hirwuser150430    3326129 2021-06-12 06:45 ILOVEBIGDATA/dwp-payments-april10.csv
```

copyToLocal: to copy a file from HDFS to local file system.

```
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -copyToLocal ILOVEBIGDATA/dwp-payments-april10.csv .
copyToLocal: /home/hirwuser150430/dwp-payments-april10.csv._COPYING_ (Permission denied)
```

cat: to print the content of a specific file.

```
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -cat ILOVEBIGDATA/dwp-payments-april10.csv
```

```
Department for Work and Pensions ,Jobcentre Plus,30/04/2010,PRINTING STATIONERY IT & CONSUMABLES,JOBCENTRE PLUS,XEROX UK LIMITED,2015
125301,91.38,,
Department for Work and Pensions ,Jobcentre Plus,30/04/2010,PRINTING STATIONERY IT & CONSUMABLES,JOBCENTRE PLUS,XEROX UK LIMITED,2015
125301,14.17,,
Department for Work and Pensions ,Jobcentre Plus,30/04/2010,PRINTING STATIONERY IT & CONSUMABLES,JOBCENTRE PLUS,XEROX UK LIMITED,2015
125301,91.66,,
Department for Work and Pensions ,Jobcentre Plus,30/04/2010,PRINTING STATIONERY IT & CONSUMABLES,JOBCENTRE PLUS,XEROX UK LIMITED,2015
125301,15.07,,
Department for Work and Pensions ,Jobcentre Plus,30/04/2010,PRINTING STATIONERY IT & CONSUMABLES,JOBCENTRE PLUS,XEROX UK LIMITED,2015
125301,10.18,,
```

moveFromLocal: to move a file from local file system to HDFS.

Hadoop fs -moveFromLocal <local source> <destination>

Put:

Similar to **copyFromLocal.**

Get:

Similar to **copyToLocal.**

Rmr: to remove a file or directory recursively.

```
hirwuser150430@ip-172-31-45-217:~$ hdfs dfs -rm -r test-antony-derectory
Deleted test-antony-derectory
```

Setrep: to change the replication factor of a file/directory in HDFS. By default the value is 3.

```
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -setrep 2 MYNEWDIRECTORY/dwp-payments-april10.csv
Replication 2 set: MYNEWDIRECTORY/dwp-payments-april10.csv
```

Practical – 3

1) List all the files and directories under /user/hirw/input in HDFS.

   Command - **hadoop fs -ls /user/hirw/input**

```
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -ls /user/hirw/input
Found 12 items
drwxr-xr-x   - hirw hirw          0 2017-01-07 19:59 /user/hirw/input/companies
drwxr-xr-x   - hirw hirw          0 2017-01-07 19:59 /user/hirw/input/dividends
drwxr-xr-x   - hirw hirw          0 2017-02-15 02:16 /user/hirw/input/employee-pig
drwxr-xr-x   - hirw hirw          0 2017-01-07 19:58 /user/hirw/input/facebook
drwxr-xr-x   - hirw hirw          0 2017-03-25 12:38 /user/hirw/input/hive
drwxr-xr-x   - hirw hirw          0 2020-07-12 16:00 /user/hirw/input/kickstarter
drwxr-xr-x   - hirw hirw          0 2020-07-12 16:02 /user/hirw/input/nyse
drwxr-xr-x   - hirw hirw          0 2017-01-07 19:58 /user/hirw/input/pagerank
drwxr-xr-x   - hirw hirw          0 2017-01-07 19:58 /user/hirw/input/songs
drwxr-xr-x   - hirw hirw          0 2017-04-09 13:46 /user/hirw/input/stocks
drwxr-xr-x   - hirw hirw          0 2017-01-07 19:58 /user/hirw/input/text-to-pdf
drwxr-xr-x   - hirw hirw          0 2020-05-25 17:05 /user/hirw/input/twitter
hirwuser150430@ip-172-31-45-217:~$ ^C
hirwuser150430@ip-172-31-45-217:~$
```

2) Create a directory in HDFS named ilovebigdata on HDFS.

   Command - **hadoop fs -mkdir ILOVEBIGDATA**

```
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -ls
Found 97 items
drwx------   - hirwuser150430 hirwuser150430          0 2021-06-09 09:31 .staging
drwxr-xr-x   - hirwuser150430 hirwuser150430          0 2021-05-25 09:29 25052021training
-rw-r--r--   3 hirwuser150430 hirwuser150430    3326129 2021-05-19 02:35 AlfaroA
drwxr-xr-x   - hirwuser150430 hirwuser150430          0 2021-06-11 12:45 BIGDATA
drwxr-xr-x   - hirwuser150430 hirwuser150430          0 2021-06-11 12:58 Big
drwxr-xr-x   - hirwuser150430 hirwuser150430          0 2021-05-19 05:57 FuentesJ
drwxr-xr-x   - hirwuser150430 hirwuser150430          0 2021-06-12 06:45 ILOVEBIGDATA
```

3) Copy the file dwp-payments-april10.csv from /hirw-workshop/input/hdfs to ilovebigdata directory in HDFS.

   Command - **hadoop fs -copyFromLocal /hirw-starterkit/hdfs/commands/dwp-payments-april10.csv ILOVEBIGDATA**

```
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -mkdir ILOVEBIGDATA
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -copyFromLocal /hirw-starterkit/hdfs/commands/dwp-payments-april10.csv ILOVEBIGDATA
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -ls ILOVEBIGDATA
Found 1 items
-rw-r--r--   3 hirwuser150430 hirwuser150430    3326129 2021-06-12 06:45 ILOVEBIGDATA/dwp-payments-april10.csv
```

4) Display the content of file dwp-payments-april10.csv under ilovebigdata in hdfs.

   Command - **hadoop fs -cat /ILOVEBIGDATA/dwp-payments-april10.csv**

```
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -cat ILOVEBIGDATA/dwp-payments-april10.csv
Department for Work and Pensions ,Jobcentre Plus,30/04/2010,PRINTING STATIONERY IT & CONSUMABLES,JOBCENTRE PLUS,XEROX UK LIMITED,2015
125301,91.38,,
Department for Work and Pensions ,Jobcentre Plus,30/04/2010,PRINTING STATIONERY IT & CONSUMABLES,JOBCENTRE PLUS,XEROX UK LIMITED,2015
125301,14.17,,
Department for Work and Pensions ,Jobcentre Plus,30/04/2010,PRINTING STATIONERY IT & CONSUMABLES,JOBCENTRE PLUS,XEROX UK LIMITED,2015
125301,91.66,,
Department for Work and Pensions ,Jobcentre Plus,30/04/2010,PRINTING STATIONERY IT & CONSUMABLES,JOBCENTRE PLUS,XEROX UK LIMITED,2015
125301,15.07,,
Department for Work and Pensions ,Jobcentre Plus,30/04/2010,PRINTING STATIONERY IT & CONSUMABLES,JOBCENTRE PLUS,XEROX UK LIMITED,2015
125301,10.18,,
```

5) Create directory with name mynewdirectory in HDFS.

   Command - **hadoop fs -mkdir /mynewdirectory**

```
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -mkdir MYNEWDIRECTORY
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -ls
Found 98 items
drwxr-xr-x   - hirwuser150430 hirwuser150430          0 2021-06-12 06:53 MYNEWDIRECTORY
```

6) Copy dwp-payments-april10.csv from ilovebigdata to mynewdirectory in HDFS.

   Command - **hadoop fs -cp ILOVEBIGDATA/dwp-payments-april10.csv MYNEWDIRECTORY**

```
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -cp ILOVEBIGDATA/dwp-payments-april10.csv MYNEWDIRECTORY
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -ls MYNEWDIRECTORY
Found 1 items
-rw-r--r--   3 hirwuser150430 hirwuser150430    3326129 2021-06-12 06:55 MYNEWDIRECTORY/dwp-payments-april10.csv
```

7) Set the replication factor of dwp-payments-april10.csv under mynewdirectory directory in HDFS to 2.

   Command **- hadoop fs -setrep 2 MYNEWDIRECTORY/dwp-payments-april10.csv**

```
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -setrep 2 MYNEWDIRECTORY/dwp-payments-april10.csv
Replication 2 set: MYNEWDIRECTORY/dwp-payments-april10.csv
```

8) check health of mynewdirectory in HDFS (chmod)(fsck) - **hdfs fsck /user/hirwuser150430/MYNEWDIRECTORY -files -blocks -locations**

```
hirwuser150430@ip-172-31-45-217:~$ hdfs fsck /user/hirwuser150430/MYNEWDIRECTORY -files -blocks -locations
Connecting to namenode via http://ec2-54-92-244-237.compute-1.amazonaws.com:50070
FSCK started by hirwuser150430 (auth:SIMPLE) from /172.31.45.217 for path /user/hirwuser150430/MYNEWDIRECTORY at
Sat Jun 12 07:23:13 UTC 2021
/user/hirwuser150430/MYNEWDIRECTORY <dir>
/user/hirwuser150430/MYNEWDIRECTORY/dwp-payments-april10.csv 3326129 bytes, 1 block(s):  OK
0. BP-2125152513-172.31.45.216-1410037307133:blk_1075416868_1676131 len=3326129 Live_repl=2 [DatanodeInfoWithStor
age[172.31.46.124:50010,DS-fe24aecb-f56f-4c9c-8cf9-a3b1259bc0d0,DISK], DatanodeInfoWithStorage[172.31.45.216:5001
0,DS-d0d9eb5c-f35f-4a12-bfdf-544085d693a3,DISK]]

Status: HEALTHY
 Total size:    3326129 B
 Total dirs:    1
 Total files:   1
 Total symlinks:                0
 Total blocks (validated):      1 (avg. block size 3326129 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    3
 Average block replication:     2.0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          3
 Number of racks:               1
FSCK ended at Sat Jun 12 07:23:13 UTC 2021 in 1 milliseconds


The filesystem under path '/user/hirwuser150430/MYNEWDIRECTORY' is HEALTHY
```

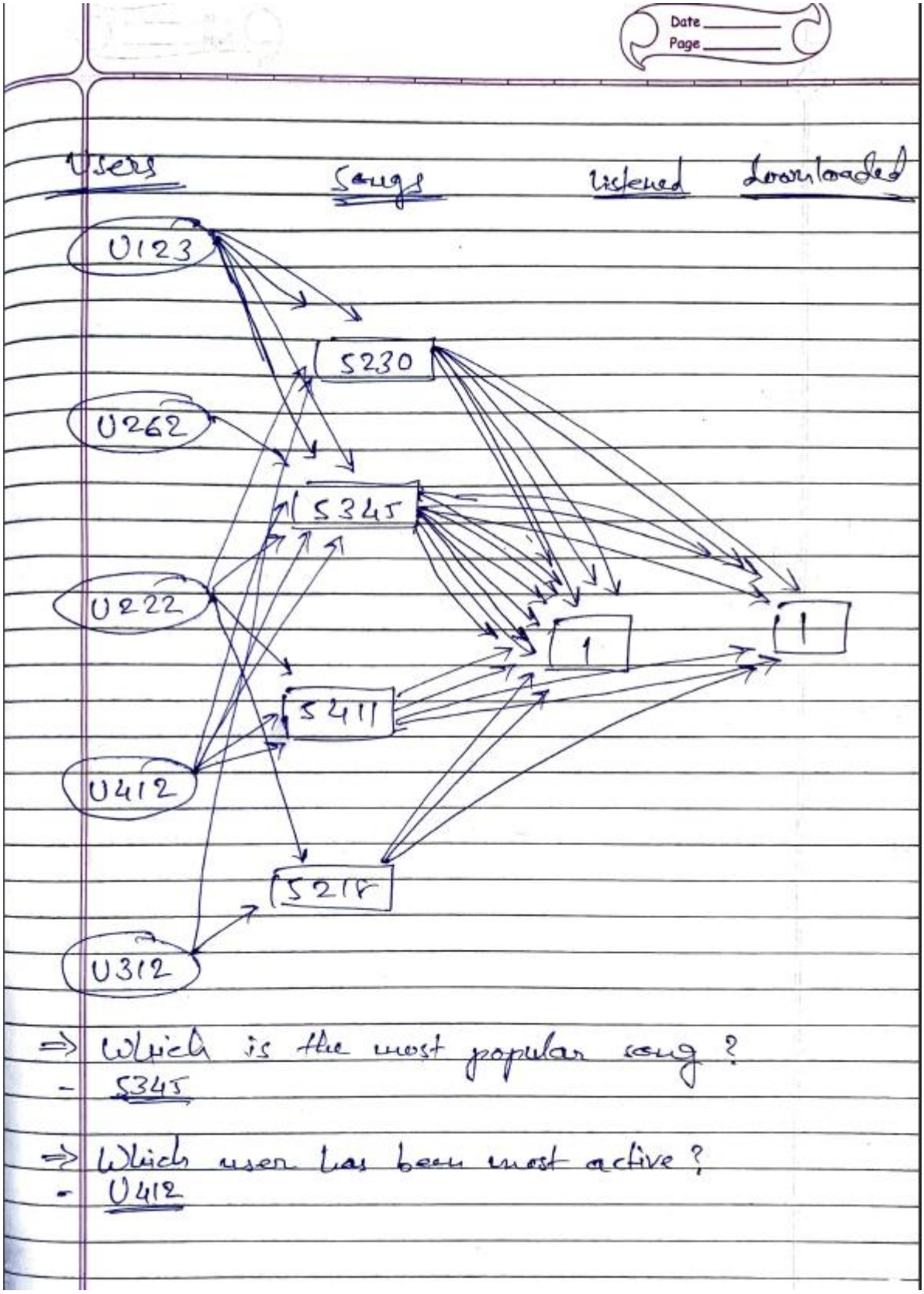
9) Bring the file dwp-payments-april10.csv under ilovebigdata directory to the local file system.

   Command - **hadoop fs -copyToLocal ILOVEBIGDATA/dwp-payments-april10.csv**

```
hirwuser150430@ip-172-31-45-217:~$ hadoop fs -copyToLocal ILOVEBIGDATA/dwp-payments-april10.csv
copyToLocal: /home/hirwuser150430/dwp-payments-april10.csv._COPYING_ (Permission denied)
```

<u>Note</u> - Since I don't have rights to make changes in the local system, here is the command and its output.

Practical – 4



⇒ Which is the most popular song ?

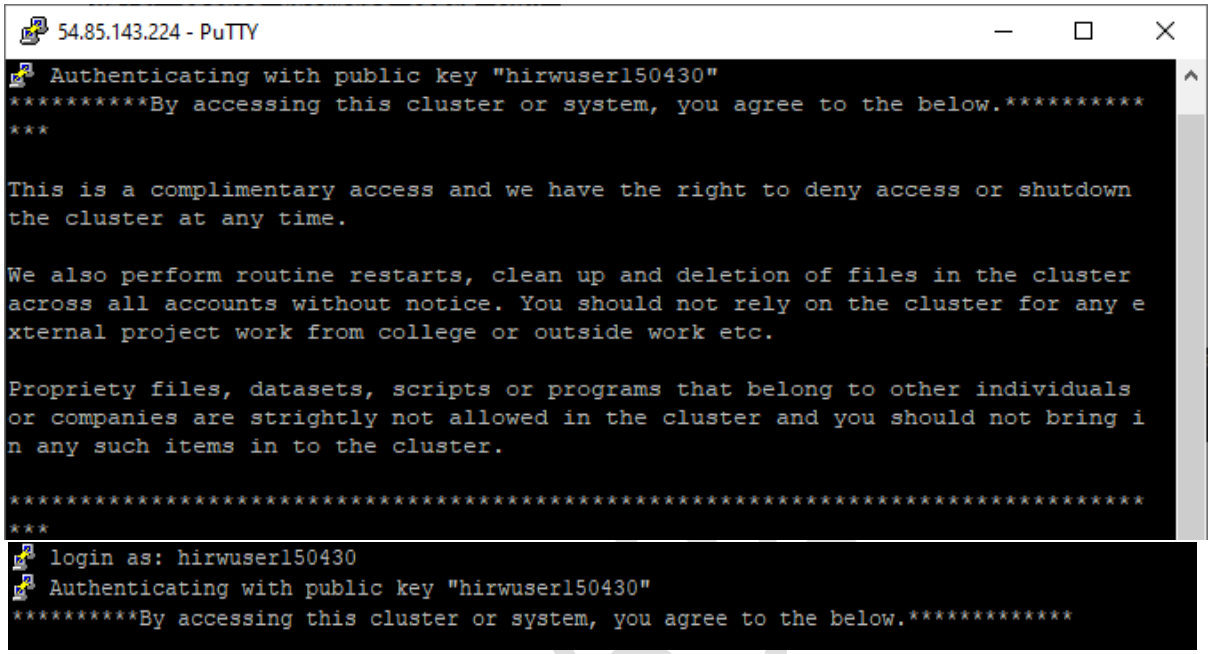
- S345

⇒ Which user has been most active ?

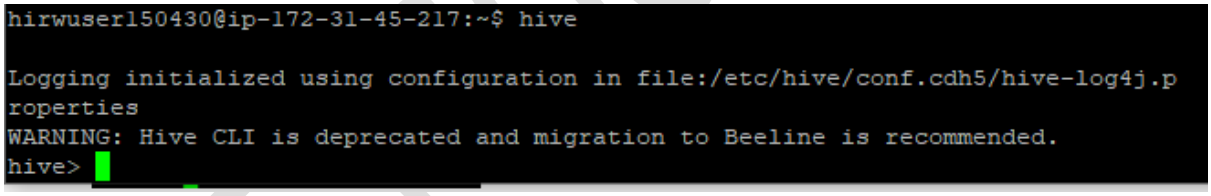
- U412

Practical – 5

1. **Login: hirwuser150430**



2. **Enter into hive shell**



3. **Display existing databases**

```
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> show databases;
OK
abhi
ansh
aparna
ayushi
bends
default
demo_hive
devansh
devansh_demo
hirwuser2074
hirwuser2093
kickstarter
mitali
parth
sagar
sagar_78019
stockmarket
stocks_db
stocks_db1
stocks_db2
stocks_db3
stocks_db_vbt
stocks_db_vr
student_zeel
trial_123
trial_jhr
trial_vbt
user2055
user_cluster
Time taken: 0.682 seconds, Fetched: 29 row(s)
hive>
```

4. **Create a database**

```
hive> create database trial_pk;
OK
Time taken: 0.664 seconds
hive>
```

5. **use that database**

```
hive> use trial_pk;
OK
Time taken: 0.016 seconds
hive>
```

6. **Create a table stocks_db**

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS stocks_db (exchng STRING, SYMBOL STRING, YMD STRING, PRICE_
OPN FLOAT, PRICE_HIGH FLOAT, PRICE_LOW FLOAT, PRICE_CLOSE FLOAT, VOLUME INT, OTHER_PRICE FLOAT)ROW FO
RMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION '/user/hirw/input/stocks';
OK
Time taken: 0.029 seconds
hive>
```

7. **Display existing tables in that database**

```
hive> show tables;
OK
stocks_db
Time taken: 0.092 seconds, Fetched: 1 row(s)
hive>
```

8. **Display records of that table (up to 100 records)**

```
hive> SELECT * FROM STOCKS_DB LIMIT 10;
OK
ABCSE   B7J     2010-02-08      8.63    8.7     8.57    8.64    78900   8.64
ABCSE   B7J     2010-02-05      8.63    8.71    8.31    8.58    218700  8.58
ABCSE   B7J     2010-02-04      8.88    8.88    8.59    8.66    89900   8.66
ABCSE   B7J     2010-02-03      8.83    8.92    8.8     8.89    119000  8.89
ABCSE   B7J     2010-02-02      8.77    8.9     8.73    8.87    51900   8.87
ABCSE   B7J     2010-02-01      8.69    8.77    8.66    8.75    38600   8.75
ABCSE   B7J     2010-01-29      8.81    8.81    8.56    8.57    91700   8.57
ABCSE   B7J     2010-01-28      8.9     8.9     8.6     8.69    92100   8.69
ABCSE   B7J     2010-01-27      8.87    8.87    8.68    8.79    82400   8.79
ABCSE   B7J     2010-01-26      8.83    8.92    8.71    8.82    106000  8.82
Time taken: 0.28 seconds, Fetched: 10 row(s)
hive>
```

9. **Describe the structure of that table**

```
hive> DESC STOCKS_DB;
OK
exchng                  string
symbol                  string
ymd                     string
price_opn               float
price_high              float
price_low               float
price_close             float
volume                  int
other_price             float
Time taken: 0.028 seconds, Fetched: 9 row(s)
hive>
```

Practical – 6

1. Install MongoDB on your operating system. (Windows/Linux/Mac)

```
C:\Users\parth>mongo
MongoDB shell version v5.0.1
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("0492765d-3ff3-42e6-a8d6-62fdb771196b") }
MongoDB server version: 5.0.1
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2021-07-24T17:13:38.330+05:30: Access control is not enabled for the database. Read and write access t
on is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>
```

2. Create a database of your name on MongoDB. verify whether the databaseshows in the list or not.

```
> use ParthKukadiya
switched to db ParthKukadiya
>
```

3. Create a collection named “stories” and verify whether the database showsin the list or not.

```
> db.createCollection("stories");
{ "ok" : 1 }
> show collections
stories
> db
ParthKukadiya
>
```

4. Create documents under stories, where various documents contains: time ofputting the story, name of the person who viewed the story, comment on the story, number of shares of the story and so on.

```
> db.stories.insert({name : 'fairy tale' , time : '11.11 AM' , viewedBy : 'person 1' , comment  : 'charming' , shares : '22'})
WriteResult({ "nInserted" : 1 })
> db.stories.insert({name : 'horror' , time : '11.11 PM' , viewedBy : 'person 2' , comment  : 'thrilling' , shares : '121'})
WriteResult({ "nInserted" : 1 })
```

5. Enlist all the documents under the collection stories. – Just like the sql queryof “select * from the stories”.

```
> db.stories.find()
{ "_id" : ObjectId("60fc23f48df50300ebc985bf"), "name" : "fairy tale", "time" : "11.11 AM", "vie
wedBy" : "person 1", "comment" : "charming", "shares" : "22" }
{ "_id" : ObjectId("60fc241b8df50300ebc985c0"), "name" : "horror", "time" : "11.11 PM", "viewedB
y" : "person 2", "comment" : "thrilling", "shares" : "121" }
>
```

Practical – 7

1. Consider the domain of an ecommerce website, opting for mongoDB as their database solution.

```
> db
ParthKukadiya
> db.createCollection("customer");
{ "ok" : 1 }
> show collections
customer
stories
```

2. Run the basic CRUD (Create, Read, Update, Delete) operations for the domain asked using the queries of your choice.

```
> db.customer.insertMany([ {name : 'parth' , surname : 'kukadiya', address : 'ahmedabad', email : '
parthkukadiya@gmail.com'} , {name : 'hina' , surname : 'joshi', address : 'ahmedabad', email : 'hin
ajoshi@gmail.com'} ,  {name : 'bhoomi' , surname : 'sheth', address : 'ahmedabad', email : 'bhoomis
heth@gmail.com'} , {name : 'shivangi' , surname : 'nayak', address : 'ahmedabad', email : 'shivangi
nayak@gmail.com'} , {name : 'abc' , surname : 'xyz' , address : 'ahmedabad', email : 'abcxyz@gmail.
com'} , {name : 'ab' , surname : 'cd' , address : 'ahmedabad', email : 'abcd@gmail.com'}])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("60fc27968df50300ebc985c1"),
                ObjectId("60fc27968df50300ebc985c2"),
                ObjectId("60fc27968df50300ebc985c3"),
                ObjectId("60fc27968df50300ebc985c4"),
                ObjectId("60fc27968df50300ebc985c5"),
                ObjectId("60fc27968df50300ebc985c6")
        ]
}
```

```
> db.customer.find().pretty()
{
        "_id" : ObjectId("60fc27968df50300ebc985c1"),
        "name" : "parth",
        "surname" : "kukadiya",
        "address" : "ahmedabad",
        "email" : "parthkukadiya@gmail.com"
}
{
        "_id" : ObjectId("60fc27968df50300ebc985c2"),
        "name" : "hina",
        "surname" : "joshi",
        "address" : "ahmedabad",
        "email" : "hinajoshi@gmail.com"
}
{
        "_id" : ObjectId("60fc27968df50300ebc985c3"),
        "name" : "bhoomi",
        "surname" : "sheth",
        "address" : "ahmedabad",
        "email" : "bhoomisheth@gmail.com"
}
{
        "_id" : ObjectId("60fc27968df50300ebc985c4"),
        "name" : "shivangi",
        "surname" : "nayak",
        "address" : "ahmedabad",
        "email" : "shivanginayak@gmail.com"
}
{
        "_id" : ObjectId("60fc27968df50300ebc985c5"),
        "name" : "abc",
        "surname" : "xyz",
        "address" : "ahmedabad",
        "email" : "abcxyz@gmail.com"
}
{
        "_id" : ObjectId("60fc27968df50300ebc985c6"),
        "name" : "ab",
        "surname" : "cd",
        "address" : "ahmedabad",
        "email" : "abcd@gmail.com"
}
>
```

Update: -

```
> db.customer.update({'name'  : 'hina'} , {$set: {'address' : 'Delhi'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customer.find()
{ "_id" : ObjectId("60fc27968df50300ebc985c1"), "name" : "parth", "surname" : "kukadiya", "address"
 : "ahmedabad", "email" : "parthkukadiya@gmail.com" }
{ "_id" : ObjectId("60fc27968df50300ebc985c2"), "name" : "hina", "surname" : "joshi", "address" : "
Delhi", "email" : "hinajoshi@gmail.com" }
{ "_id" : ObjectId("60fc27968df50300ebc985c3"), "name" : "bhoomi", "surname" : "sheth", "address" :
 "ahmedabad", "email" : "bhoomisheth@gmail.com" }
{ "_id" : ObjectId("60fc27968df50300ebc985c4"), "name" : "shivangi", "surname" : "nayak", "address"
 : "ahmedabad", "email" : "shivanginayak@gmail.com" }
{ "_id" : ObjectId("60fc27968df50300ebc985c5"), "name" : "abc", "surname" : "xyz", "address" : "ahm
edabad", "email" : "abcxyz@gmail.com" }
{ "_id" : ObjectId("60fc27968df50300ebc985c6"), "name" : "ab", "surname" : "cd", "address" : "ahmed
abad", "email" : "abcd@gmail.com" }
>
```

Remove: -

```
> db.customer.remove({'name'  : 'ab'})
WriteResult({ "nRemoved" : 1 })
> db.customer.find()
{ "_id" : ObjectId("60fc27968df50300ebc985c1"), "name" : "parth", "surname" : "kukadiya", "address"
 : "ahmedabad", "email" : "parthkukadiya@gmail.com" }
{ "_id" : ObjectId("60fc27968df50300ebc985c2"), "name" : "hina", "surname" : "joshi", "address" : "
Delhi", "email" : "hinajoshi@gmail.com" }
{ "_id" : ObjectId("60fc27968df50300ebc985c3"), "name" : "bhoomi", "surname" : "sheth", "address" :
 "ahmedabad", "email" : "bhoomisheth@gmail.com" }
{ "_id" : ObjectId("60fc27968df50300ebc985c4"), "name" : "shivangi", "surname" : "nayak", "address"
 : "ahmedabad", "email" : "shivanginayak@gmail.com" }
{ "_id" : ObjectId("60fc27968df50300ebc985c5"), "name" : "abc", "surname" : "xyz", "address" : "ahm
edabad", "email" : "abcxyz@gmail.com" }
>
```

3. Run the queries of your choice which are Where Clause Equivalentsof RDBMS in MongoDB.

```
> db.customer.find({'name'   : 'parth'})
{ "_id" : ObjectId("60fc27968df50300ebc985c1"), "name" : "parth", "surname" : "kukadiya", "address"
 : "ahmedabad", "email" : "parthkukadiya@gmail.com" }
>
```