BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

SUBJECT:
Big Data Analytics and Visualization

***By**: Pranav Nair*
***UID:** 2022300065*
***BRANCH:** TE COMPS A*

INDEX

Exp 1 - To setup and install Hadoop in Pseudo-Distributed Mode and monitoring Hadoop
Exp 2 - Hadoop Commands
Exp 3 - File management tasks in Hadoop
Exp 4 - Install Apache PySpark(Apache Spark) Using Miniconda
Exp5 - Execute wordcount program in pyspark. Compare the execution time
Exp 6 - Analyze a large dataset using Apache Spark
Exp 7 - Pivot and Unpivot of DataFrame in Spark SQL
Exp 8 - Visual Analytics in Tableau: Connection with multiple tables, Create data Extracts, Aim: Create a Report based on passed parameter
Exp 9 - Visual Analytics in Tableau : Sorting, Grouping, Filtering, Formatting Pane, Trend lines, reference lines
Exp 10 - Explore and present interactive data insights from real world dataset (Dashboards) using POWER BI
EXTRA QUESTION

Experiment no. 1	
AIM :	To setup and install Hadoop in Pseudo-Distributed Mode and monitoring Hadoop

Theory	Apache Hadoop is an open-source framework that allows for the distributed storage and processing of large datasets across clusters of computers using simple programming models. It is designed to scale up from a single server to thousands of machines.
	
	Hadoop Architecture Components:
	
	1. **HDFS (Hadoop Distributed File System):** For data storage across nodes.
	   ○ **NameNode:** Manages metadata and file system namespace.
	   ○ **DataNode:** Stores actual data blocks.
	2. **MapReduce:** For processing large datasets in a parallel and distributed manner.
	   ○ **JobTracker (YARN ResourceManager):** Manages jobs and resources.
	   ○ **TaskTracker (YARN NodeManager):** Executes tasks on each node.

Code & Output:	**Install Java** ● **sudo apt-get update**

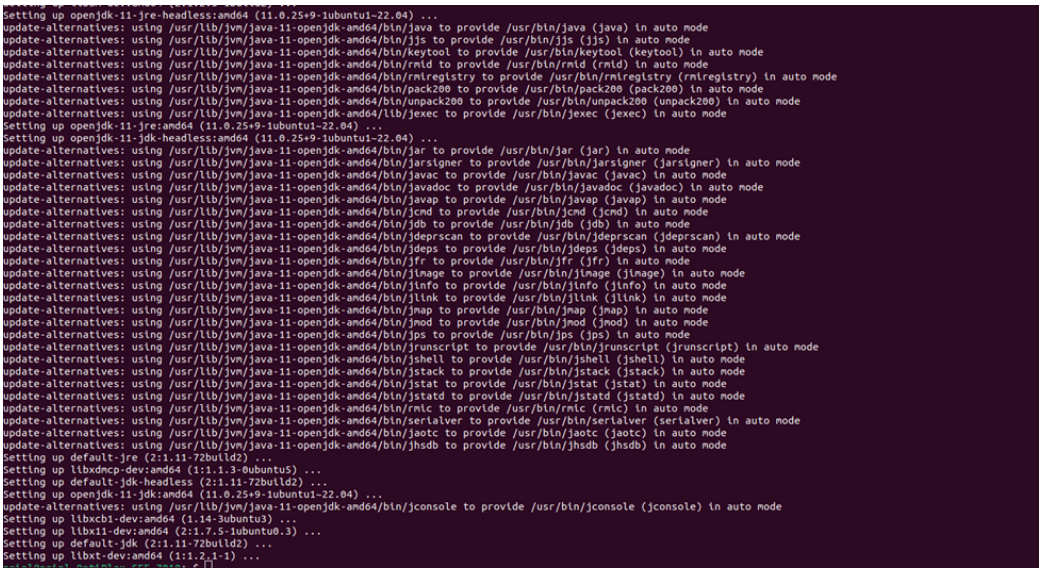
```
psipl@psipl-OptiPlex-SFF-7010:~$ sudo apt-get update
[sudo] password for psipl:
Hit:1 https://brave-browser-apt-release.s3.brave.com stable InRelease
Hit:2 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:3 https://packages.microsoft.com/repos/code stable InRelease
Hit:4 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:6 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/8.0 InRelease
Hit:7 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:8 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:9 http://dell.archive.canonical.com jammy InRelease
Hit:10 http://oem.archive.canonical.com jammy InRelease
Hit:11 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
N: Skipping acquire of configured file 'main/binary-i386/Packages' as repository
 'https://brave-browser-apt-release.s3.brave.com stable InRelease' doesn't suppo
rt architecture 'i386'
```

- **sudo apt-get install default-jdk**

```
psipl@psipl-OptiPlex-SFF-7010:~$ sudo apt-get install default-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  cifs-utils dctrl-tools dmraid gir1.2-timezonemap-1.0 gir1.2-xkl-1.0 keyutils kpartx kpartx-boot libdebian-installer4 libdmraid1.0.0.rc16 liblvm15
  libtimezonemap-data libtimezonemap1 python3-icu python3-pam rdate user-setup
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ca-certificates-java default-jdk-headless default-jre default-jre-headless fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni libice-dev
  libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-11-jdk openjdk-11-jdk-headless openjdk-11-jre
  openjdk-11-jre-headless x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-11-demo openjdk-11-source visualvm fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  | fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java default-jdk default-jdk-headless default-jre default-jre-headless fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni
  libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-11-jdk openjdk-11-jdk-headless openjdk-11-jre
  openjdk-11-jre-headless x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 24 newly installed, 0 to remove and 42 not upgraded.
Need to get 122 MB of archives.
After this operation, 276 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 java-common all 0.72build2 [6,782 B]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openjdk-11-jre-headless amd64 11.0.25+9-1ubuntu1~22.04 [42.6 MB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 default-jre-headless amd64 2:1.11-72build2 [3,042 B]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ca-certificates-java all 20190909ubuntu1.2 [12.1 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openjdk-11-jre amd64 11.0.25+9-1ubuntu1~22.04 [216 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 default-jre amd64 2:1.11-72build2 [896 B]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openjdk-11-jdk-headless amd64 11.0.25+9-1ubuntu1~22.04 [73.7 MB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 default-jdk-headless amd64 2:1.11-72build2 [942 B]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openjdk-11-jdk amd64 11.0.25+9-1ubuntu1~22.04 [1,334 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 default-jdk amd64 2:1.11-72build2 [908 B]
Get:11 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-extra all 2.37-2build1 [2,041 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java all 0.38.0-5build1 [53.1 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java-jni amd64 0.38.0-5build1 [49.0 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 xorg-sgml-doctools all 1:1.11-1.1 [10.9 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 x11proto-dev all 2021.5-1 [604 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libice-dev amd64 2:1.0.10-1build2 [51.4 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libpthread-stubs0-dev amd64 0.4-1build2 [5,516 B]
Get:18 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libsm-dev amd64 2:1.2.3-1build2 [18.1 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libxau-dev amd64 1:1.0.9-1build5 [9,724 B]
Get:20 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libxdmcp-dev amd64 1:1.1.3-0ubuntu5 [26.5 kB]
Get:21 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 xtrans-dev all 1.4.0-1 [68.9 kB]
Get:22 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libxcb1-dev amd64 1.14-3ubuntu3 [86.5 kB]
Get:23 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libx11-dev amd64 2:1.7.5-1ubuntu0.3 [744 kB]
```

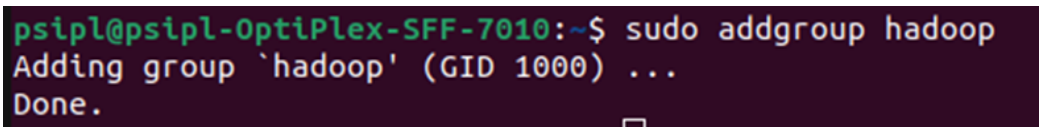
```
Get:23 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libx11-dev amd64 2:1.7.5-1ubuntu0.3 [744 kB]
Get:24 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libxt-dev amd64 1:1.2.1-1 [396 kB]
Fetched 122 MB in 1min 40s (1,221 kB/s)
Selecting previously unselected package java-common.
(Reading database ... 219362 files and directories currently installed.)
Preparing to unpack .../00-java-common_0.72build2_all.deb ...
Unpacking java-common (0.72build2) ...
Selecting previously unselected package openjdk-11-jre-headless:amd64.
Preparing to unpack .../01-openjdk-11-jre-headless_11.0.25+9-1ubuntu1~22.04_amd64.deb ...
Unpacking openjdk-11-jre-headless:amd64 (11.0.25+9-1ubuntu1~22.04) ...
Selecting previously unselected package default-jre-headless.
Preparing to unpack .../02-default-jre-headless_2%3a1.11-72build2_amd64.deb ...
Unpacking default-jre-headless (2:1.11-72build2) ...
Selecting previously unselected package ca-certificates-java.
Preparing to unpack .../03-ca-certificates-java_20190909ubuntu1.2_all.deb ...
Unpacking ca-certificates-java (20190909ubuntu1.2) ...
Selecting previously unselected package openjdk-11-jre:amd64.
Preparing to unpack .../04-openjdk-11-jre_11.0.25+9-1ubuntu1~22.04_amd64.deb ...
Unpacking openjdk-11-jre:amd64 (11.0.25+9-1ubuntu1~22.04) ...
Selecting previously unselected package default-jre.
Preparing to unpack .../05-default-jre_2%3a1.11-72build2_amd64.deb ...
Unpacking default-jre (2:1.11-72build2) ...
Selecting previously unselected package openjdk-11-jdk-headless:amd64.
Preparing to unpack .../06-openjdk-11-jdk-headless_11.0.25+9-1ubuntu1~22.04_amd64.deb ...
Unpacking openjdk-11-jdk-headless:amd64 (11.0.25+9-1ubuntu1~22.04) ...
Selecting previously unselected package default-jdk-headless.
Preparing to unpack .../07-default-jdk-headless_2%3a1.11-72build2_amd64.deb ...
Unpacking default-jdk-headless (2:1.11-72build2) ...
Selecting previously unselected package openjdk-11-jdk:amd64.
Preparing to unpack .../08-openjdk-11-jdk_11.0.25+9-1ubuntu1~22.04_amd64.deb ...
Unpacking openjdk-11-jdk:amd64 (11.0.25+9-1ubuntu1~22.04) ...
Selecting previously unselected package default-jdk.
Preparing to unpack .../09-default-jdk_2%3a1.11-72build2_amd64.deb ...
Unpacking default-jdk (2:1.11-72build2) ...
Selecting previously unselected package fonts-dejavu-extra.
Preparing to unpack .../10-fonts-dejavu-extra_2.37-2build1_all.deb ...
Unpacking fonts-dejavu-extra (2.37-2build1) ...
Selecting previously unselected package libatk-wrapper-java.
Preparing to unpack .../11-libatk-wrapper-java_0.38.0-5build1_all.deb ...
Unpacking libatk-wrapper-java (0.38.0-5build1) ...
Selecting previously unselected package libatk-wrapper-java-jni:amd64.
Preparing to unpack .../12-libatk-wrapper-java-jni_0.38.0-5build1_amd64.deb ...
Unpacking libatk-wrapper-java-jni:amd64 (0.38.0-5build1) ...
Selecting previously unselected package xorg-sgml-doctools.
Preparing to unpack .../13-xorg-sgml-doctools_1%3a1.11-1.1_all.deb ...
Unpacking xorg-sgml-doctools (1:1.11-1.1) ...
```

```
Setting up xtrans-dev (1.4.0-1) ...
Setting up fonts-dejavu-extra (2.37-2build1) ...
Setting up xorg-sgml-doctools (1:1.11-1.1) ...
Setting up libatk-wrapper-java (0.38.0-5build1) ...
Setting up libatk-wrapper-java-jni:amd64 (0.38.0-5build1) ...
Setting up ca-certificates-java (20190909ubuntu1.2) ...
head: cannot open '/etc/ssl/certs/java/cacerts' for reading: No such file or directory
Adding debian:DigiCert_High_Assurance_EV_Root_CA.pem
Adding debian:certSIGN_ROOT_CA.pem
Adding debian:CommScope_Public_Trust_RSA_Root-01.pem
Adding debian:AffirmTrust_Premium.pem
Adding debian:TUBITAK_Kamu_SM_SSL_Kok_Sertifikasi_-_Surum_1.pem
Adding debian:COMODO_ECC_Certification_Authority.pem
Adding debian:CommScope_Public_Trust_RSA_Root-02.pem
Adding debian:Security_Communication_RootCA3.pem
Adding debian:HARICA_TLS_RSA_Root_CA_2021.pem
Adding debian:SecureTrust_CA.pem
Adding debian:BJCA_Global_Root_CA1.pem
Adding debian:Go_Daddy_Class_2_CA.pem
Adding debian:DigiCert_Assured_ID_Root_CA.pem
Adding debian:GlobalSign_Root_E46.pem
Adding debian:SSL.com_EV_Root_Certification_Authority_RSA_R2.pem
Adding debian:T-TeleSec_GlobalRoot_Class_2.pem
Adding debian:Amazon_Root_CA_4.pem
Adding debian:vTrus_ECC_Root_CA.pem
Adding debian:emSign_ECC_Root_CA_-_C3.pem
Adding debian:Entrust_Root_Certification_Authority_-_G2.pem
Adding debian:Microsec_e-Szigno_Root_CA_2009.pem
Adding debian:Amazon_Root_CA_1.pem
Adding debian:Hellenic_Academic_and_Research_Institutions_ECC_RootCA_2015.pem
Adding debian:DigiCert_Assured_ID_Root_G2.pem
Adding debian:TrustAsia_Global_Root_CA_G3.pem
Adding debian:Telia_Root_CA_v2.pem
Adding debian:Starfield_Class_2_CA.pem
Adding debian:Microsoft_RSA_Root_Certificate_Authority_2017.pem
Adding debian:certSIGN_Root_CA_G2.pem
Adding debian:AffirmTrust_Networking.pem
Adding debian:Actalis_Authentication_Root_CA.pem
Adding debian:Izenpe.com.pem
Adding debian:Certum_EC-384_CA.pem
Adding debian:CommScope_Public_Trust_ECC_Root-01.pem
Adding debian:vTrus_Root_CA.pem
Adding debian:Buypass_Class_3_Root_CA.pem
Adding debian:NAVER_Global_Root_Certification_Authority.pem
Adding debian:Autoridad_de_Certificacion_Firmaprofesional_CIF_A62634068.pem
Adding debian:SSL.com_EV_Root_Certification_Authority_ECC.pem
```

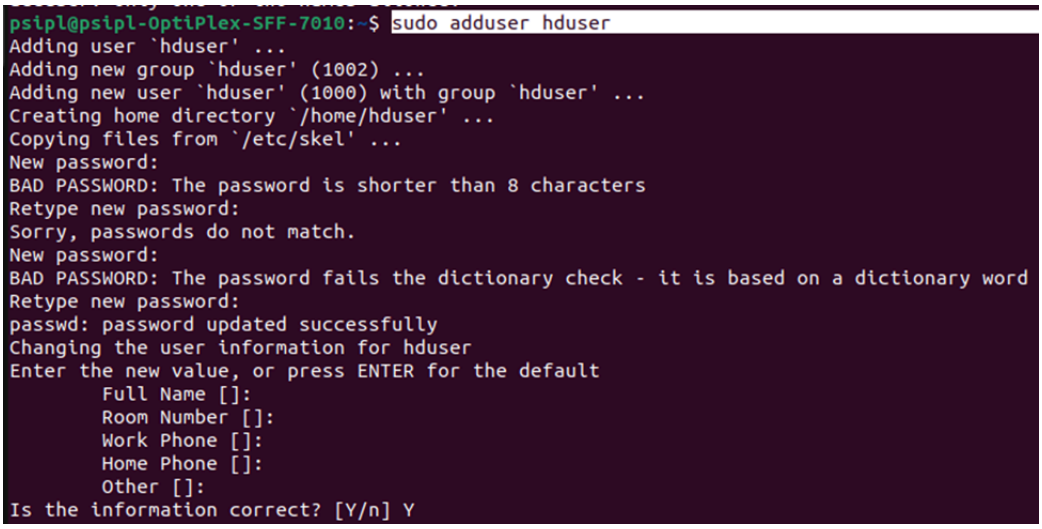


Add a dedicated hadoop user and add to groups

- **sudo add group hadoop**



- **sudo adduser hduser**



- **sudo usermod -aG hadoop hduser**

```
psipl@psipl-OptiPlex-SFF-7010:~$ sudo usermod -aG hadoop hduser
```

- **groups hduser**

```
psipl@psipl-OptiPlex-SFF-7010:~$ groups hduser
hduser : hduser hadoop
```

Install and configure SSH

- **sudo apt-get install ssh**

```
psipl@psipl-OptiPlex-SFF-7010:~$ sudo apt-get install ssh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  clfs-utils dctrl-tools dmraid gir1.2-timezonemap-1.0 gir1.2-xkl-1.0 keyutils kpartx kpartx-boot libdebian-installer4 libdmraid1.0.0.rc16 libllvm15
  libtimezonemap-data libtimezonemap1 python3-icu python3-pam rdate user-setup
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh ssh-import-id
0 upgraded, 5 newly installed, 0 to remove and 42 not upgraded.
Need to get 756 kB of archives.
After this operation, 6,180 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-server amd64 1:8.9p1-3ubuntu0.10 [38.9 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-server amd64 1:8.9p1-3ubuntu0.10 [435 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ssh all 1:8.9p1-3ubuntu0.10 [4,850 B]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term all 6.3-2ubuntu0.1 [267 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.11-0ubuntu1 [10.1 kB]
Fetched 756 kB in 2s (359 kB/s)
Preconfiguring packages ...
Selecting previously unselected package openssh-sftp-server.
(Reading database ... 220558 files and directories currently installed.)
Preparing to unpack .../openssh-sftp-server_1%3a8.9p1-3ubuntu0.10_amd64.deb ...
Unpacking openssh-sftp-server (1:8.9p1-3ubuntu0.10) ...
Selecting previously unselected package openssh-server.
Preparing to unpack .../openssh-server_1%3a8.9p1-3ubuntu0.10_amd64.deb ...
Unpacking openssh-server (1:8.9p1-3ubuntu0.10) ...
Selecting previously unselected package ssh.
Preparing to unpack .../ssh_1%3a8.9p1-3ubuntu0.10_all.deb ...
Unpacking ssh (1:8.9p1-3ubuntu0.10) ...
Selecting previously unselected package ncurses-term.
Preparing to unpack .../ncurses-term_6.3-2ubuntu0.1_all.deb ...
Unpacking ncurses-term (6.3-2ubuntu0.1) ...
Selecting previously unselected package ssh-import-id.
Preparing to unpack .../ssh-import-id_5.11-0ubuntu1_all.deb ...
Unpacking ssh-import-id (5.11-0ubuntu1) ...
Setting up openssh-sftp-server (1:8.9p1-3ubuntu0.10) ...
Setting up openssh-server (1:8.9p1-3ubuntu0.10) ...
```

```
Preconfiguring packages ...
Selecting previously unselected package openssh-sftp-server.
(Reading database ... 220558 files and directories currently installed.)
Preparing to unpack .../openssh-sftp-server_1%3a8.9p1-3ubuntu0.10_amd64.deb ...
Unpacking openssh-sftp-server (1:8.9p1-3ubuntu0.10) ...
Selecting previously unselected package openssh-server.
Preparing to unpack .../openssh-server_1%3a8.9p1-3ubuntu0.10_amd64.deb ...
Unpacking openssh-server (1:8.9p1-3ubuntu0.10) ...
Selecting previously unselected package ssh.
Preparing to unpack .../ssh_1%3a8.9p1-3ubuntu0.10_all.deb ...
Unpacking ssh (1:8.9p1-3ubuntu0.10) ...
Selecting previously unselected package ncurses-term.
Preparing to unpack .../ncurses-term_6.3-2ubuntu0.1_all.deb ...
Unpacking ncurses-term (6.3-2ubuntu0.1) ...
Selecting previously unselected package ssh-import-id.
Preparing to unpack .../ssh-import-id_5.11-0ubuntu1_all.deb ...
Unpacking ssh-import-id (5.11-0ubuntu1) ...
Setting up openssh-sftp-server (1:8.9p1-3ubuntu0.10) ...
Setting up openssh-server (1:8.9p1-3ubuntu0.10) ...

Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
3072 SHA256:zQgOYwsGmOSVPKlXCrXXUCTXC4juNhZXTJf/ANiwUE0 root@psipl-OptiPlex-SFF-7010 (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:eAKu7C1OSGTUTZq+8P/SSsN5KkBUp2pKeXWzTvtpmbg root@psipl-OptiPlex-SFF-7010 (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:oHIb9GgQNl4BwupiAJuvZtAMdPVbSpLj7F/9VydsEz8 root@psipl-OptiPlex-SFF-7010 (ED25519)
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
ssh.socket is a disabled or a static unit, not starting it.
Setting up ssh-import-id (5.11-0ubuntu1) ...
Setting up ncurses-term (6.3-2ubuntu0.1) ...
Setting up ssh (1:8.9p1-3ubuntu0.10) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
```

- **su - hduser -c "ssh-keygen -t rsa -P ""**

```
psipl@psipl-OptiPlex-SFF-7010:~$ su - hduser -c "ssh-keygen -t rsa -P ''"
Password:
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
Created directory '/home/hduser/.ssh'.
Your identification has been saved in /home/hduser/.ssh/id_rsa
Your public key has been saved in /home/hduser/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:AM2Ti1RHr6M+olkkopjdEu9cgNrQQIU4BrHHzfWcI18 hduser@psipl-OptiPlex-SFF-7010
The key's randomart image is:
+---[RSA 3072]----+
|=oo..+ooo        |
|+= o.o=+ o       |
|+.o.o.oo= E      |
| + .. .+ +       |
|o = o   S        |
|oB * . . .       |
|= + + o          |
|   *.o.          |
|  o.o...         |
+----[SHA256]-----+
```

- **cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys**

```
hduser@psipl-OptiPlex-SFF-7010:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

- **chmod 700 ~/.ssh**

```
hduser@psipl-OptiPlex-SFF-7010:~$ chmod 700 ~/.ssh
```

- **chmod 600 ~/.ssh/authorized_keys**

```
hduser@psipl-OptiPlex-SFF-7010:~$ chmod 600 ~/.ssh/authorized_keys
```

- **ssh localhost**

```
hduser@psipl-OptiPlex-SFF-7010:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:oHIb9GgQNi4BwupiAJuvZtAMdPVbSpLj7F/9VydsEz8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

1 device has a firmware upgrade available.
Run `fwupdmgr get-upgrades` for more information.


Expanded Security Maintenance for Applications is not enabled.

42 updates can be applied immediately.
5 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

2 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


1 device has a firmware upgrade available.
Run `fwupdmgr get-upgrades` for more information.
```

- **logout**

```
hduser@psipl-OptiPlex-SFF-7010:~$ exit
logout
Connection to localhost closed.
```

Install hadoop

- **wget**

https://archive.apache.org/dist/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.t ar.gz

```
hduser@psipl-OptiPlex-SFF-7010:~$ wget https://archive.apache.org/dist/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz
--2025-01-30 09:22:39--  https://archive.apache.org/dist/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 195257604 (186M) [application/x-gzip]
Saving to: 'hadoop-2.6.0.tar.gz'

hadoop-2.6.0.tar.gz                 100%[===================================================================================>] 186.21M  1.62MB/s    in 1m 55s

2025-01-30 09:24:35 (1.62 MB/s) - 'hadoop-2.6.0.tar.gz' saved [195257604/195257604]
```

- **tar xvzf hadoop-2.6.0.tar.gz**

```
hduser@psipl-OptiPlex-SFF-7010:~$ tar xvzf hadoop-2.6.0.tar.gz
hadoop-2.6.0/
hadoop-2.6.0/etc/
hadoop-2.6.0/etc/hadoop/
hadoop-2.6.0/etc/hadoop/hdfs-site.xml
hadoop-2.6.0/etc/hadoop/hadoop-metrics2.properties
hadoop-2.6.0/etc/hadoop/container-executor.cfg
hadoop-2.6.0/etc/hadoop/mapred-site.xml.template
hadoop-2.6.0/etc/hadoop/yarn-env.cmd
hadoop-2.6.0/etc/hadoop/hadoop-policy.xml
hadoop-2.6.0/etc/hadoop/log4j.properties
hadoop-2.6.0/etc/hadoop/httpfs-env.sh
hadoop-2.6.0/etc/hadoop/core-site.xml
hadoop-2.6.0/etc/hadoop/hadoop-env.cmd
hadoop-2.6.0/etc/hadoop/yarn-env.sh
hadoop-2.6.0/etc/hadoop/capacity-scheduler.xml
hadoop-2.6.0/etc/hadoop/kms-site.xml
hadoop-2.6.0/etc/hadoop/slaves
hadoop-2.6.0/etc/hadoop/ssl-server.xml.example
hadoop-2.6.0/etc/hadoop/mapred-env.sh
hadoop-2.6.0/etc/hadoop/mapred-queues.xml.template
hadoop-2.6.0/etc/hadoop/httpfs-site.xml
hadoop-2.6.0/etc/hadoop/configuration.xsl
hadoop-2.6.0/etc/hadoop/hadoop-env.sh
hadoop-2.6.0/etc/hadoop/hadoop-metrics.properties
hadoop-2.6.0/etc/hadoop/httpfs-signature.secret
hadoop-2.6.0/etc/hadoop/ssl-client.xml.example
hadoop-2.6.0/etc/hadoop/httpfs-log4j.properties
hadoop-2.6.0/etc/hadoop/kms-acls.xml
hadoop-2.6.0/etc/hadoop/kms-env.sh
hadoop-2.6.0/etc/hadoop/kms-log4j.properties
hadoop-2.6.0/etc/hadoop/yarn-site.xml
hadoop-2.6.0/etc/hadoop/mapred-env.cmd
hadoop-2.6.0/sbin/
hadoop-2.6.0/sbin/start-dfs.cmd
hadoop-2.6.0/sbin/stop-balancer.sh
hadoop-2.6.0/sbin/start-balancer.sh
hadoop-2.6.0/sbin/stop-all.sh
hadoop-2.6.0/sbin/yarn-daemon.sh
hadoop-2.6.0/sbin/yarn-daemons.sh
hadoop-2.6.0/sbin/stop-yarn.cmd
hadoop-2.6.0/sbin/stop-dfs.sh
hadoop-2.6.0/sbin/stop-dfs.cmd
hadoop-2.6.0/sbin/start-dfs.sh
```

```
hadoop-2.6.0/share/hadoop/mapreduce/lib/protobuf-java-2.5.0.jar
hadoop-2.6.0/share/hadoop/mapreduce/lib/xz-1.0.jar
hadoop-2.6.0/share/hadoop/mapreduce/lib/avro-1.7.4.jar
hadoop-2.6.0/share/hadoop/mapreduce/lib/log4j-1.2.17.jar
hadoop-2.6.0/share/hadoop/mapreduce/lib/guice-servlet-3.0.jar
hadoop-2.6.0/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar
hadoop-2.6.0/share/hadoop/mapreduce/hadoop-mapreduce-client-hs-plugins-2.6.0.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-core-2.6.0-test-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-examples-2.6.0-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-jobclient-2.6.0-test-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-core-2.6.0-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-hs-plugins-2.6.0-test-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-shuffle-2.6.0-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-app-2.6.0-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-app-2.6.0-test-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-common-2.6.0-test-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-shuffle-2.6.0-test-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-examples-2.6.0-test-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-hs-2.6.0-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-hs-plugins-2.6.0-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-jobclient-2.6.0-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-common-2.6.0-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-hs-2.6.0-test-sources.jar
hadoop-2.6.0/LICENSE.txt
hadoop-2.6.0/README.txt
hadoop-2.6.0/bin/
hadoop-2.6.0/bin/hdfs.cmd
hadoop-2.6.0/bin/test-container-executor
hadoop-2.6.0/bin/container-executor
hadoop-2.6.0/bin/hadoop.cmd
hadoop-2.6.0/bin/rcc
hadoop-2.6.0/bin/hdfs
hadoop-2.6.0/bin/mapred
hadoop-2.6.0/bin/hadoop
hadoop-2.6.0/bin/yarn.cmd
hadoop-2.6.0/bin/mapred.cmd
hadoop-2.6.0/bin/yarn
hadoop-2.6.0/include/
hadoop-2.6.0/include/TemplateFactory.hh
hadoop-2.6.0/include/StringUtils.hh
hadoop-2.6.0/include/hdfs.h
hadoop-2.6.0/include/Pipes.hh
hadoop-2.6.0/include/SerialUtils.hh
hduser@psipl-OptiPlex-SFF-7010:~$
```

- **sudo mkdir /usr/local/hadoop**

```
hduser@psipl-OptiPlex-SFF-7010:~$ sudo mkdir -p /usr/local/hadoop
```

- **sudo mv hadoop-2.6.0/* /usr/local/hadoop**

```
hduser@psipl-OptiPlex-SFF-7010:~$ sudo mv hadoop-2.6.0/* /usr/local/hadoop
```

- **sudo rm â€“r hadoop-2.6.0**

```
hduser@psipl-OptiPlex-SFF-7010:~$ sudo rm -r hadoop-2.6.0
```

- **sudo chown â€“R hduser:hadoop /usr/local/hadoop**

```
hduser@pslpl-OptiPlex-SFF-7010: $ sudo chown -R hduser:hadoop /usr/local/hadoop
```

sudo nano ~/.bashrc
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386 export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME export HADOOP_COMMOMN_HOME=$HADOOP_HOME export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME  # Fixed typo (was HADOOP_COMMOMN_HOME)
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
```

sudo nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386

```
# The java implementation to use.
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol.  Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}
```

sudo nano /usr/local/hadoop/etc/hadoop/core-site.xml
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>

```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

sudo nano /usr/local/hadoop/etc/hadoop/hdfs-site.xml

<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>

```
<configuration>
    <!-- Replication factor for HDFS -->
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>

    <!-- Directory where NameNode stores its metadata -->
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
    </property>

    <!-- Directory where DataNode stores data blocks -->
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
    </property>
</configuration>
```

sudo nano /usr/local/hadoop/etc/hadoop/yarn-site.xml

<property>
<name>yarn.nodemanager.aux-services</name>
<value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

```
<configuration>
    <!-- Auxiliary services for NodeManager -->
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
    </property>

    <!-- Class to handle MapReduce shuffle -->
    <property>
        <name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</name>
        <value>org.apache.hadoop.mapred.ShuffleHandler</value>
    </property>
</configuration>
```

cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml
sudo nano /usr/local/hadoop/etc/hadoop/mapred-site.xml
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>

```
<configuration>
    <!-- Set the framework name to YARN -->
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
```

sudo mkdir -p /usr/local/hadoop_tmp
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode sudo chown -R hduser /usr/local/hadoop_tmp

```
hduser@psipl-OptiPlex-SFF-7010:~$ sudo mkdir -p /usr/local/hadoop_tmp
hduser@psipl-OptiPlex-SFF-7010:~$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
hduser@psipl-OptiPlex-SFF-7010:~$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
hduser@psipl-OptiPlex-SFF-7010:~$ sudo chown -R hduser:hadoop /usr/local/hadoop_tmp
```

hdfs namenode -format

```
hduser@psipl-OptiPlex-SFF-7010:~$ hdfs namenode -format
25/01/30 10:03:06 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = psipl-OptiPlex-SFF-7010/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.6.0
STARTUP_MSG:   classpath = /usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/asm-3
ar:/usr/local/hadoop/share/hadoop/common/lib/jackson-core-asl-1.9.13.jar:/usr/local/hadoop/share/hadoop
adoop/common/lib/avro-1.7.4.jar:/usr/local/hadoop/share/hadoop/common/lib/apacheds-kerberos-codec-2.0.0
-asl-1.9.13.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-beanutils-1.7.0.jar:/usr/local/hadoop
share/hadoop/common/lib/commons-lang-2.6.jar:/usr/local/hadoop/share/hadoop/common/lib/guava-11.0.2.jar
```

source .bashrc

```
hduser@psipl-OptiPlex-SFF-7010:~$ source .bashrc
```

start-all.sh

```
hduser@psipl-OptiPlex-SFF-7010:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.6.0.jar)
 to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
25/01/30 10:05:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-psipl-OptiPlex-SFF-7010.out
localhost: WARNING: An illegal reflective access operation has occurred
localhost: WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth
-2.6.0.jar) to method sun.security.krb5.Config.getInstance()
localhost: WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
localhost: WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
localhost: WARNING: All illegal access operations will be denied in a future release
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-psipl-OptiPlex-SFF-7010.out
localhost: WARNING: An illegal reflective access operation has occurred
localhost: WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth
-2.6.0.jar) to method sun.security.krb5.Config.getInstance()
localhost: WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
localhost: WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
localhost: WARNING: All illegal access operations will be denied in a future release
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ED25519 key fingerprint is SHA256:oHIb9GgQNl4BwupiAJuvZtAMdPVbSpLj7F/9VydsEz8.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ED25519) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hduser-secondarynamenode-psipl-OptiPlex-SFF-7010.out
0.0.0.0: WARNING: An illegal reflective access operation has occurred
0.0.0.0: WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2
.6.0.jar) to method sun.security.krb5.Config.getInstance()
0.0.0.0: WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
0.0.0.0: WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
0.0.0.0: WARNING: All illegal access operations will be denied in a future release
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.6.0.jar)
 to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
25/01/30 10:06:12 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

jps

```
hduser@psipl-OptiPlex-SFF-7010:~$ jps
19972 DataNode
20344 ResourceManager
19737 NameNode
20171 SecondaryNameNode
20735 Jps
```

stop-all.sh

```
hduser@pslpl-OptiPlex-SFF-7010:~$ stop-all.sh
This script is Deprecated. Instead use stop-dfs.sh and stop-yarn.sh
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.
 to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
25/01/30 10:06:59 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Stopping namenodes on [localhost]
localhost: stopping namenode
localhost: stopping datanode
Stopping secondary namenodes [0.0.0.0]
0.0.0.0: stopping secondarynamenode
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.
 to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
25/01/30 10:07:15 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
stopping yarn daemons
stopping resourcemanager
localhost: no nodemanager to stop
no proxyserver to stop
```

Basic Hadoop Commands

1. Print the Hadoop version

hadoop version

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /home/hadoop/hadoop/share/hadoop/common/hadoop-common-3.3.6.jar
```

2. List all the hadoop file system shell commands

hadoop fs

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs
Usage: hadoop fs [generic options]
        [-appendToFile [-n] <localsrc> ... <dst>]
        [-cat [-ignoreCrc] <src> ...]
        [-checksum [-v] <src> ...]
        [-chgrp [-R] GROUP PATH...]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-concat <target path> <src path> <src path> ...]
        [-copyFromLocal [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <localsrc> ... <dst>]
        [-copyToLocal [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ... <localdst>]
        [-count [-q] [-h] [-v] [-t [<storage type>]] [-u] [-x] [-e] [-s] <path> ...]
        [-cp [-f] [-p | -p[topax]] [-d] [-t <thread count>] [-q <thread pool queue size>] <src> ... <dst>]
        [-createSnapshot <snapshotDir> [<snapshotName>]]
        [-deleteSnapshot <snapshotDir> <snapshotName>]
        [-df [-h] [<path> ...]]
        [-du [-s] [-h] [-v] [-x] <path> ...]
        [-expunge [-immediate] [-fs <path>]]
        [-find <path> ... <expression> ...]
        [-get [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ... <localdst>]
        [-getfacl [-R] <path>]
        [-getfattr [-R] {-n name | -d} [-e en] <path>]
        [-getmerge [-nl] [-skip-empty-file] <src> <localdst>]
        [-head <file>]
        [-help [cmd ...]]
        [-ls [-C] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [-e] [<path> ...]]
        [-mkdir [-p] <path> ...]
        [-moveFromLocal [-f] [-p] [-l] [-d] <localsrc> ... <dst>]
        [-moveToLocal <src> <localdst>]
        [-mv <src> ... <dst>]
        [-put [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <localsrc> ... <dst>]
        [-renameSnapshot <snapshotDir> <oldName> <newName>]
        [-rm [-f] [-r|-R] [-skipTrash] [-safely] <src> ...]
        [-rmdir [--ignore-fail-on-non-empty] <dir> ...]
        [-setfacl [-R] [{-b|-k} {-m|-x <acl_spec>} <path>]|[--set <acl_spec> <path>]]
        [-setfattr {-n name [-v value] | -x name} <path>]
        [-setrep [-R] [-w] <rep> <path> ...]
        [-stat [format] <path> ...]
        [-tail [-f] [-s <sleep interval>] <file>]
        [-test -[defswrz] <path>]
        [-text [-ignoreCrc] <src> ...]
```

```
Generic options supported are:
-conf <configuration file>        specify an application configuration file
-D <property=value>               define a value for a given property
-fs <file:///|hdfs://namenode:port> specify default filesystem URL to use, overrides 'fs.defaultFS' property from configurations.
-jt <local|resourcemanager:port>  specify a ResourceManager
-files <file1,...>                specify a comma-separated list of files to be copied to the map reduce cluster
-libjars <jar1,...>               specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...>          specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] [commandOptions]
```

3. Help hadoop
fs -help

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -help
Usage: hadoop fs [generic options]
        [-appendToFile [-n] <localsrc> ... <dst>]
        [-cat [-ignoreCrc] <src> ...]
        [-checksum [-v] <src> ...]
        [-chgrp [-R] GROUP PATH...]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-concat <target path> <src path> <src path> ...]
        [-copyFromLocal [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <localsrc> ... <dst>]
        [-copyToLocal [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ... <localdst>]
        [-count [-q] [-h] [-v] [-t [<storage type>]] [-u] [-x] [-e] [-s] <path> ...]
        [-cp [-f] [-p | -p[topax]] [-d] [-t <thread count>] [-q <thread pool queue size>] <src> ... <dst>]
        [-createSnapshot <snapshotDir> [<snapshotName>]]
        [-deleteSnapshot <snapshotDir> <snapshotName>]
        [-df [-h] [<path> ...]]
        [-du [-s] [-h] [-v] [-x] <path> ...]
        [-expunge [-immediate] [-fs <path>]]
        [-find <path> ... <expression> ...]
        [-get [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ... <localdst>]
        [-getfacl [-R] <path>]
        [-getfattr [-R] {-n name | -d} [-e en] <path>]
        [-getmerge [-nl] [-skip-empty-file] <src> <localdst>]
        [-head <file>]
        [-help [cmd ...]]
        [-ls [-C] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [-e] [<path> ...]]
        [-mkdir [-p] <path> ...]
        [-moveFromLocal [-f] [-p] [-l] [-d] <localsrc> ... <dst>]
        [-moveToLocal <src> <localdst>]
        [-mv <src> ... <dst>]
        [-put [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <localsrc> ... <dst>]
        [-renameSnapshot <snapshotDir> <oldName> <newName>]
        [-rm [-f] [-r|-R] [-skipTrash] [-safely] <src> ...]
        [-rmdir [--ignore-fail-on-non-empty] <dir> ...]
        [-setfacl [-R] [{-b|-k} {-m|-x <acl_spec>} <path>]|[--set <acl_spec> <path>]]
        [-setfattr {-n name [-v value] | -x name} <path>]
        [-setrep [-R] [-w] <rep> <path> ...]
        [-stat [format] <path> ...]
        [-tail [-f] [-s <sleep interval>] <file>]
        [-test -[defswrz] <path>]
```

```
        [-usage [cmd ...]]

-appendToFile [-n] <localsrc> ... <dst> :
  Appends the contents of all the given local files to the given dst file. The dst
  file will be created if it does not exist. If <localSrc> is -, then the input is
  read from stdin. Option -n represents that use NEW_BLOCK create flag to append
  file.

-cat [-ignoreCrc] <src> ... :
  Fetch all files that match the file pattern <src> and display their content on
  stdout.

-checksum [-v] <src> ... :
  Dump checksum information for files that match the file pattern <src> to stdout.
  Note that this requires a round-trip to a datanode storing each block of the
  file, and thus is not efficient to run on a large number of files. The checksum
  of a file depends on its content, block size and the checksum algorithm and
  parameters used for creating the file.

-chgrp [-R] GROUP PATH... :
  This is equivalent to -chown ... :GROUP ...

-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH... :
  Changes permissions of a file. This works similar to the shell's chmod command
  with a few exceptions.

  -R          modifies the files recursively. This is the only option currently
              supported.
  <MODE>      Mode is the same as mode used for the shell's command. The only
              letters recognized are 'rwxXt', e.g. +t,a+r,g-w,+rwx,o=r.
  <OCTALMODE> Mode specifed in 3 or 4 digits. If 4 digits, the first may be 1 or
              0 to turn the sticky bit on or off, respectively.  Unlike the
              shell command, it is not possible to specify only part of the
              mode, e.g. 754 is same as u=rwx,g=rx,o=r.

  If none of 'augo' is specified, 'a' is assumed and unlike the shell command, no
  umask is applied.

-chown [-R] [OWNER][:[GROUP]] PATH... :
  Changes owner and group of a file. This is similar to the shell's chown command
  with a few exceptions.
```

```
  -R  modifies the files recursively. This is the only option currently
      supported.

  If only the owner or group is specified, then only the owner or group is
  modified. The owner and group names may only consist of digits, alphabet, and
  any of [-_./@a-zA-Z0-9]. The names are case sensitive.

  WARNING: Avoid using '.' to separate user name and group though Linux allows it.
  If user names have dots in them and you are using local file system, you might
  see surprising results since the shell command 'chown' is used for local files.

-concat <target path> <src path> <src path> ... :
  Concatenate existing source files into the target file. Target file and source
  files should be in the same directory.

-copyFromLocal [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <localsrc> ... <dst> :
  Identical to the -put command.

-copyToLocal [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ... <localdst> :
  Identical to the -get command.

-count [-q] [-h] [-v] [-t [<storage type>]] [-u] [-x] [-e] [-s] <path> ... :
  Count the number of directories, files and bytes under the paths
  that match the specified file pattern.  The output columns are:
  DIR_COUNT FILE_COUNT CONTENT_SIZE PATHNAME
  or, with the -q option:
  QUOTA REM_QUOTA SPACE_QUOTA REM_SPACE_QUOTA
        DIR_COUNT FILE_COUNT CONTENT_SIZE PATHNAME
  The -h option shows file sizes in human readable format.
  The -v option displays a header line.
  The -x option excludes snapshots from being calculated.
  The -t option displays quota by storage types.
  It should be used with -q or -u option, otherwise it will be ignored.
  If a comma-separated list of storage types is given after the -t option,
  it displays the quota and usage for the specified types.
  Otherwise, it displays the quota and usage for all the storage
  types that support quota. The list of possible storage types(case insensitive):
  ram_disk, ssd, disk and archive.
  It can also pass the value '', 'all' or 'ALL' to specify all the storage types.
  The -u option shows the quota and
  the usage against the quota without the detailed content summary.The -e option
```

```
-cp [-f] [-p | -p[topax]] [-d] [-t <thread count>] [-q <thread pool queue size>] <src> ... <dst> :
  Copy files that match the file pattern <src> to a destination. When copying
  multiple files, the destination must be a directory.
  Flags :

  -p[topax]                    Preserve file attributes [topx] (timestamps,
                               ownership, permission, ACL, XAttr). If -p is
                               specified with no arg, then preserves timestamps,
                               ownership, permission. If -pa is specified, then
                               preserves permission also because ACL is a
                               super-set of permission. Determination of whether
                               raw namespace extended attributes are preserved is
                               independent of the -p flag.
  -f                           Overwrite the destination if it already exists.
  -d                           Skip creation of temporary file(<dst>._COPYING_).
  -t <thread count>            Number of threads to be used, default is 1.
  -q <thread pool queue size>  Thread pool queue size to be used, default is
                               1024.

-createSnapshot <snapshotDir> [<snapshotName>] :
  Create a snapshot on a directory

-deleteSnapshot <snapshotDir> <snapshotName> :
  Delete a snapshot from a directory

-df [-h] [<path> ...] :
  Shows the capacity, free and used space of the filesystem. If the filesystem has
  multiple partitions, and no path to a particular partition is specified, then
  the status of the root partitions will be shown.

  -h  Formats the sizes of files in a human-readable fashion rather than a number
      of bytes.

-du [-s] [-h] [-v] [-x] <path> ... :
  Show the amount of space, in bytes, used by the files that match the specified
  file pattern. The following flags are optional:

  -s  Rather than showing the size of each individual file that matches the
      pattern, shows the total (summary) size.
  -h  Formats the sizes of files in a human-readable fashion rather than a number
```

```
  -v  option displays a header line.
  -x  Excludes snapshots from being counted.

  Note that, even without the -s option, this only shows size summaries one level
  deep into a directory.

  The output is in the form
        size    disk space consumed     name(full path)

-expunge [-immediate] [-fs <path>] :
  Delete files from the trash that are older than the retention threshold

-find <path> ... <expression> ... :
  Finds all files that match the specified expression and
  applies selected actions to them. If no <path> is specified
  then defaults to the current working directory. If no
  expression is specified then defaults to -print.

  The following primary expressions are recognised:
    -name pattern
    -iname pattern
      Evaluates as true if the basename of the file matches the
      pattern using standard file system globbing.
      If -iname is used then the match is case insensitive.

    -print
    -print0
      Always evaluates to true. Causes the current pathname to be
      written to standard output followed by a newline. If the -print0
      expression is used then an ASCII NULL character is appended rather
      than a newline.

  The following operators are recognised:
    expression -a expression
    expression -and expression
    expression expression
      Logical AND operator for joining two expressions. Returns
      true if both child expressions return true. Implied by the
      juxtaposition of two expressions and so does not need to be
      explicitly specified. The second expression will not be
      applied if the first fails.
```

```
-get [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ... <localdst> :
  Copy files that match the file pattern <src> to the local name. <src> is kept.
  When copying multiple files, the destination must be a directory.
  Flags:

  -p                            Preserves timestamps, ownership and the mode.
  -f                            Overwrites the destination if it already exists.
  -crc                          write CRC checksums for the files downloaded.
  -ignoreCrc                    Skip CRC checks on the file(s) downloaded.
  -t <thread count>             Number of threads to be used, default is 1.
  -q <thread pool queue size>   Thread pool queue size to be used, default is
                                1024.

-getfacl [-R] <path> :
  Displays the Access Control Lists (ACLs) of files and directories. If a
  directory has a default ACL, then getfacl also displays the default ACL.

  -R      List the ACLs of all files and directories recursively.
  <path>  File or directory to list.

-getfattr [-R] {-n name | -d} [-e en] <path> :
  Displays the extended attribute names and values (if any) for a file or
  directory.

  -R             Recursively list the attributes for all files and directories.
  -n name        Dump the named extended attribute value.
  -d             Dump all extended attribute values associated with pathname.
  -e <encoding>  Encode values after retrieving them.Valid encodings are "text",
                 "hex", and "base64". Values encoded as text strings are enclosed
                 in double quotes ("), and values encoded as hexadecimal and
                 base64 are prefixed with 0x and 0s, respectively.
  <path>         The file or directory.

-getmerge [-nl] [-skip-empty-file] <src> <localdst> :
  Get all the files in the directories that match the source file pattern and
  merge and sort them to only one file on local fs. <src> is kept.

  -nl               Add a newline character at the end of each file.
  -skip-empty-file  Do not add new line character for empty file.

-head <file> :
```

```
-text [-ignoreCrc] <src> ... :
  Takes a source file and outputs the file in text format.
  The allowed formats are zip and TextRecordInputStream and Avro.

-touch [-a] [-m] [-t TIMESTAMP (yyyyMMdd:HHmmss) ] [-c] <path> ... :
  Updates the access and modification times of the file specified by the <path> to
  the current time. If the file does not exist, then a zero length file is created
  at <path> with current time as the timestamp of that <path>.
  -a Change only the access time
  -m Change only the modification time
  -t TIMESTAMP Use specified timestamp instead of current time
   TIMESTAMP format yyyyMMdd:HHmmss
  -c Do not create any files

-touchz <path> ... :
  Creates a file of zero length at <path> with current time as the timestamp of
  that <path>. An error is returned if the file exists with non-zero length

-truncate [-w] <length> <path> ... :
  Truncate all files that match the specified file pattern to the specified
  length.

  -w  Requests that the command wait for block recovery to complete, if
      necessary.

-usage [cmd ...] :
  Displays the usage for given command or all commands if none is specified.

Generic options supported are:
-conf <configuration file>        specify an application configuration file
-D <property=value>               define a value for a given property
-fs <file:///|hdfs://namenode:port> specify default filesystem URL to use, overrides 'fs.defaultFS' property from configurations.
-jt <local|resourcemanager:port>  specify a ResourceManager
-files <file1,...>                specify a comma-separated list of files to be copied to the map reduce cluster
-libjars <jar1,...>               specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...>          specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] [commandOptions]
```

4. List the contents of the root directory in HDFS
hadoop fs -ls /

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /
Found 3 items
drwxr-xr-x   - hadoop supergroup          0 2025-01-29 11:49 /logs
drwxr-xr-x   - hadoop supergroup          0 2025-01-29 11:48 /test1
drwxr-xr-x   - hadoop supergroup          0 2025-01-30 05:21 /user
```

 # 5. Report the amount of space used and available on currently mounted filesystem
hadoop fs -df hdfs:/

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -df hdfs:/
Filesystem                      Size     Used      Available  Use%
hdfs://localhost:9000  1081101176832  9457664  1022186688512    0%
```

 # 6. Count the number of directories,files and bytes under the paths that match the specified file pattern
hadoop fs -count hdfs:/

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -count hdfs:/
          14           14             9278479 hdfs:///
```

 # 7. Create a new directory named â€œhadoopâ€ below the /user/training directory in HDFS.
hadoop fs -mkdir /user/training/hadoop

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -mkdir /user/training/hadoop
mkdir: `/user/training/hadoop': File exists
```

 # 8. Add a sample text file from the local directory named â€œdataâ€ to the new directory you created in HDFS during the previous step.
hadoop fs -put data/sample.txt /user/training/hadoop

```
hadoop@DESKTOP-2LPHD1B:~$ echo "This is a sample file" > data/sample.txt
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -put data/sample.txt /user/training/hadoop
put: `/user/training/hadoop/sample.txt': File exists
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop
Found 1 items
-rw-r--r--   1 hadoop supergroup         12 2025-01-30 05:24 /user/training/hadoop/sample.txt
```

 # 9. List the contents of this new directory in HDFS. hadoop fs -ls /user/training/hadoop

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop
Found 1 items
-rw-r--r--   1 hadoop supergroup         12 2025-01-30 05:24 /user/training/hadoop/sample.txt
```

 # 10. Add the entire local directory called â€œretailâ€ to the /user/training directory in HDFS.
hadoop fs -put data/retail /user/training/hadoop

```
hadoop@DESKTOP-2LPHD1B:~$ mkdir -p data/retail
hadoop@DESKTOP-2LPHD1B:~$ echo "Customer data" > data/retail/customers
hadoop@DESKTOP-2LPHD1B:~$ echo "Product data" > data/retail/products
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -put data/retail /user/training/hadoop
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop
Found 2 items
drwxr-xr-x   - hadoop supergroup          0 2025-01-30 12:52 /user/training/hadoop/retail
-rw-r--r--   1 hadoop supergroup         12 2025-01-30 05:24 /user/training/hadoop/sample.txt
```

 # 11. Delete a file â€˜customersâ€™ from the â€œretailâ€ directory. hadoop fs -rm hadoop/retail/customers

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -rm /user/training/hadoop/retail/customers
Deleted /user/training/hadoop/retail/customers
```

 # 12. Delete all files from the â€œretailâ€ directory using a wildcard.
hadoop fs -rm hadoop/retail/*

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -rm /user/training/hadoop/retail/*
Deleted /user/training/hadoop/retail/products
Deleted /user/training/hadoop/retail/sample.txt
```

 # 13. Remove the entire retail directory and all of its contents in HDFS.
hadoop fs -rm -r hadoop/retail

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -rm -r /user/training/hadoop/retail
Deleted /user/training/hadoop/retail
```

 # 14. Add the purchases.txt file from the local directory named â€œ/home/training/â€ to the hadoop directory you created in HDFS hadoop fs -copyFromLocal /home/training/purchases.txt hadoop/

```
hadoop@DESKTOP-2LPHD1B:~$ mkdir -p ~/training
hadoop@DESKTOP-2LPHD1B:~$ echo "Purchase history data" > ~/training/purchases.txt
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -copyFromLocal /~/training/purchases.txt /user/training/hadoop/
copyFromLocal: `/~/training/purchases.txt': No such file or directory
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -copyFromLocal ~/training/purchases.txt /user/training/hadoop/
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop
Found 2 items
-rw-r--r--   1 hadoop supergroup         22 2025-01-30 13:00 /user/training/hadoop/purchases.txt
-rw-r--r--   1 hadoop supergroup         12 2025-01-30 05:24 /user/training/hadoop/sample.txt
```

 # 15. To view the contents of your text file purchases.txt which is present in your hadoop directory.
hadoop fs -cat hadoop/purchases.txt

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -cat /user/training/hadoop/purchases.txt
Purchase history data
```

 # 16. cp is used to copy files between directories present in HDFS hadoop fs -cp /user/training/*.txt /user/training/hadoop

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/
Found 2 items
drwxr-xr-x   - hadoop supergroup          0 2025-01-30 13:00 /user/training/hadoop
drwxr-xr-x   - hadoop supergroup          0 2025-01-30 05:26 /user/training/retail
hadoop@DESKTOP-2LPHD1B:~$ echo "File 1 content" > file1.txt
hadoop@DESKTOP-2LPHD1B:~$ echo "File 2 content" > file2.txt
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -put file1.txt /user/training/
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -put file2.txt /user/training/
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/
Found 4 items
-rw-r--r--   1 hadoop supergroup         15 2025-01-30 13:03 /user/training/file1.txt
-rw-r--r--   1 hadoop supergroup         15 2025-01-30 13:03 /user/training/file2.txt
drwxr-xr-x   - hadoop supergroup          0 2025-01-30 13:00 /user/training/hadoop
drwxr-xr-x   - hadoop supergroup          0 2025-01-30 05:26 /user/training/retail
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -cp /user/training/*.txt /user/training/hadoop/
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop
Found 4 items
-rw-r--r--   1 hadoop supergroup         15 2025-01-30 13:03 /user/training/hadoop/file1.txt
-rw-r--r--   1 hadoop supergroup         15 2025-01-30 13:03 /user/training/hadoop/file2.txt
-rw-r--r--   1 hadoop supergroup         22 2025-01-30 13:00 /user/training/hadoop/purchases.txt
-rw-r--r--   1 hadoop supergroup         12 2025-01-30 05:24 /user/training/hadoop/sample.txt
```

 # 17. â€˜-getâ€™ command can be used alternaively to â€˜-copyToLocalâ€™ command
hadoop fs -get hadoop/sample.txt /home/training/

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -get /user/training/hadoop/sample.txt ~/training/
```

 # 18. Display last kilobyte of the file â€œpurchases.txtâ€ to stdout. hadoop fs -tail hadoop/purchases.txt

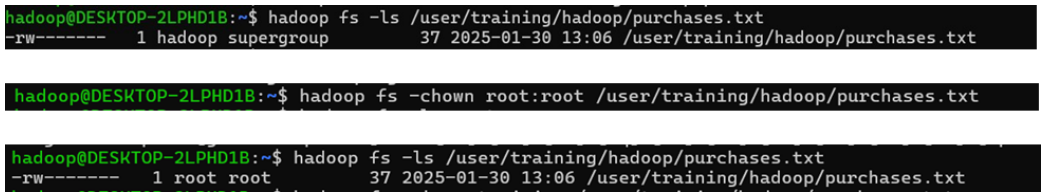
```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -tail /user/training/hadoop/purchases.txt
Purchase history data
```

 # 19. Default file permissions are 666 in HDFS. Use â€˜-chmodâ€™ command to change permissions of a file
hadoop fs -ls hadoop/purchases.txt
sudo -u hdfs hadoop fs -chmod 600 hadoop/purchases.txt

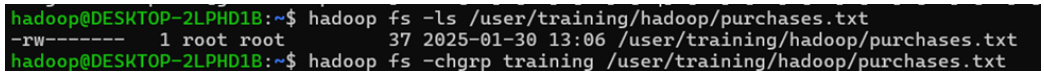
```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop/purchases.txt
-rw-r--r--   1 hadoop supergroup         37 2025-01-30 13:06 /user/training/hadoop/purchases.txt
```

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -chmod 600 /user/training/hadoop/purchases.txt
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop/purchases.txt
-rw-------   1 hadoop supergroup         37 2025-01-30 13:06 /user/training/hadoop/purchases.txt
```

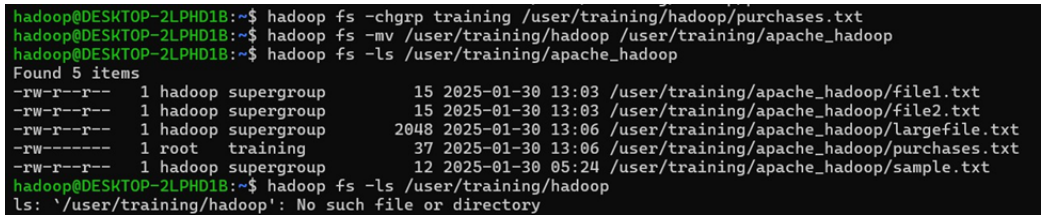
 # 20. Default names of owner and group are training,training. Use â€˜-chownâ€™ to change owner name and group name simultaneously
hadoop fs -ls hadoop/purchases.txt
sudo -u hdfs hadoop fs -chown root:root hadoop/purchases.txt



 # 21. Default name of group is training Use â€˜-chgrpâ€™ command to change group name
hadoop fs -ls hadoop/purchases.txt
sudo -u hdfs hadoop fs -chgrp training hadoop/purchases.txt



 # 22. Move a directory from one location to other hadoop fs -mv hadoop apache_hadoop



CONCLUSION:	Through this experiment I learned how to install Hadoop on Ubuntu and also learned how to write basic commands for Hadoop.

Experiment no. 2	
AIM :	To Perform Hadoop Commands
Theory	**What is HDFS?** Hadoop Distributed File System (HDFS) is the primary storage system of Hadoop. It is a distributed file system designed to store very large datasets reliably and to stream those data sets at high bandwidth to user applications. ● HDFS works on master-slave architecture ● Large files are split into blocks (default size: 128 MB) and distributed across multiple nodes.
Code & Output:	**<u>Basic Hadoop Commands</u>** 1. Print the Hadoop version hadoop version ``` hadoop@DESKTOP-2LPHD1B:~$ hadoop version Hadoop 3.3.6 Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c Compiled by ubuntu on 2023-06-18T08:22Z Compiled on platform linux-x86_64 Compiled with protoc 3.7.1 From source with checksum 5652179ad55f76cb287d9c633bb53bbd This command was run using /home/hadoop/hadoop/share/hadoop/common/hadoop-common-3.3.6.jar ``` # 2. List all the hadoop file system shell commands hadoop fs

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs
Usage: hadoop fs [generic options]
        [-appendToFile [-n] <localsrc> ... <dst>]
        [-cat [-ignoreCrc] <src> ...]
        [-checksum [-v] <src> ...]
        [-chgrp [-R] GROUP PATH...]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-concat <target path> <src path> <src path> ...]
        [-copyFromLocal [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <localsrc> ... <dst>]
        [-copyToLocal [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ... <localdst>]
        [-count [-q] [-h] [-v] [-t [<storage type>]] [-u] [-x] [-e] [-s] <path> ...]
        [-cp [-f] [-p | -p[topax]] [-d] [-t <thread count>] [-q <thread pool queue size>] <src> ... <dst>]
        [-createSnapshot <snapshotDir> [<snapshotName>]]
        [-deleteSnapshot <snapshotDir> <snapshotName>]
        [-df [-h] [<path> ...]]
        [-du [-s] [-h] [-v] [-x] <path> ...]
        [-expunge [-immediate] [-fs <path>]]
        [-find <path> ... <expression> ...]
        [-get [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ... <localdst>]
        [-getfacl [-R] <path>]
        [-getfattr [-R] {-n name | -d} [-e en] <path>]
        [-getmerge [-nl] [-skip-empty-file] <src> <localdst>]
        [-head <file>]
        [-help [cmd ...]]
        [-ls [-C] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [-e] [<path> ...]]
        [-mkdir [-p] <path> ...]
        [-moveFromLocal [-f] [-p] [-l] [-d] <localsrc> ... <dst>]
        [-moveToLocal <src> <localdst>]
        [-mv <src> ... <dst>]
        [-put [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <localsrc> ... <dst>]
        [-renameSnapshot <snapshotDir> <oldName> <newName>]
        [-rm [-f] [-r|-R] [-skipTrash] [-safely] <src> ...]
        [-rmdir [--ignore-fail-on-non-empty] <dir> ...]
        [-setfacl [-R] [{-b|-k} {-m|-x <acl_spec>} <path>]|[--set <acl_spec> <path>]]
        [-setfattr {-n name [-v value] | -x name} <path>]
        [-setrep [-R] [-w] <rep> <path> ...]
        [-stat [format] <path> ...]
        [-tail [-f] [-s <sleep interval>] <file>]
        [-test -[defswrz] <path>]
        [-text [-ignoreCrc] <src> ...]
```

```
Generic options supported are:
-conf <configuration file>        specify an application configuration file
-D <property=value>               define a value for a given property
-fs <file:///|hdfs://namenode:port> specify default filesystem URL to use, overrides 'fs.defaultFS' property from configurations.
-jt <local|resourcemanager:port>  specify a ResourceManager
-files <file1,...>                specify a comma-separated list of files to be copied to the map reduce cluster
-libjars <jar1,...>               specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...>          specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] [commandOptions]
```

3. Help

hadoop fs -help

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -help
Usage: hadoop fs [generic options]
        [-appendToFile [-n] <localsrc> ... <dst>]
        [-cat [-ignoreCrc] <src> ...]
        [-checksum [-v] <src> ...]
        [-chgrp [-R] GROUP PATH...]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-concat <target path> <src path> <src path> ...]
        [-copyFromLocal [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <localsrc> ... <dst>]
        [-copyToLocal [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ... <localdst>]
        [-count [-q] [-h] [-v] [-t [<storage type>]] [-u] [-x] [-e] [-s] <path> ...]
        [-cp [-f] [-p | -p[topax]] [-d] [-t <thread count>] [-q <thread pool queue size>] <src> ... <dst>]
        [-createSnapshot <snapshotDir> [<snapshotName>]]
        [-deleteSnapshot <snapshotDir> <snapshotName>]
        [-df [-h] [<path> ...]]
        [-du [-s] [-h] [-v] [-x] <path> ...]
        [-expunge [-immediate] [-fs <path>]]
        [-find <path> ... <expression> ...]
        [-get [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ... <localdst>]
        [-getfacl [-R] <path>]
        [-getfattr [-R] {-n name | -d} [-e en] <path>]
        [-getmerge [-nl] [-skip-empty-file] <src> <localdst>]
        [-head <file>]
        [-help [cmd ...]]
        [-ls [-C] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [-e] [<path> ...]]
        [-mkdir [-p] <path> ...]
        [-moveFromLocal [-f] [-p] [-l] [-d] <localsrc> ... <dst>]
        [-moveToLocal <src> <localdst>]
        [-mv <src> ... <dst>]
        [-put [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <localsrc> ... <dst>]
        [-renameSnapshot <snapshotDir> <oldName> <newName>]
        [-rm [-f] [-r|-R] [-skipTrash] [-safely] <src> ...]
        [-rmdir [--ignore-fail-on-non-empty] <dir> ...]
        [-setfacl [-R] [{-b|-k} {-m|-x <acl_spec>} <path>]|[--set <acl_spec> <path>]]
        [-setfattr {-n name [-v value] | -x name} <path>]
        [-setrep [-R] [-w] <rep> <path> ...]
        [-stat [format] <path> ...]
        [-tail [-f] [-s <sleep interval>] <file>]
        [-test -[defswrz] <path>]
```

```
        [-usage [cmd ...]]

-appendToFile [-n] <localsrc> ... <dst> :
  Appends the contents of all the given local files to the given dst file. The dst
  file will be created if it does not exist. If <localSrc> is -, then the input is
  read from stdin. Option -n represents that use NEW_BLOCK create flag to append
  file.

-cat [-ignoreCrc] <src> ... :
  Fetch all files that match the file pattern <src> and display their content on
  stdout.

-checksum [-v] <src> ... :
  Dump checksum information for files that match the file pattern <src> to stdout.
  Note that this requires a round-trip to a datanode storing each block of the
  file, and thus is not efficient to run on a large number of files. The checksum
  of a file depends on its content, block size and the checksum algorithm and
  parameters used for creating the file.

-chgrp [-R] GROUP PATH... :
  This is equivalent to -chown ... :GROUP ...

-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH... :
  Changes permissions of a file. This works similar to the shell's chmod command
  with a few exceptions.

  -R          modifies the files recursively. This is the only option currently
              supported.
  <MODE>      Mode is the same as mode used for the shell's command. The only
              letters recognized are 'rwxXt', e.g. +t,a+r,g-w,+rwx,o=r.
  <OCTALMODE> Mode specifed in 3 or 4 digits. If 4 digits, the first may be 1 or
              0 to turn the sticky bit on or off, respectively.  Unlike the
              shell command, it is not possible to specify only part of the
              mode, e.g. 754 is same as u=rwx,g=rx,o=r.

  If none of 'augo' is specified, 'a' is assumed and unlike the shell command, no
  umask is applied.

-chown [-R] [OWNER][:[GROUP]] PATH... :
  Changes owner and group of a file. This is similar to the shell's chown command
  with a few exceptions.
```

```
  -R  modifies the files recursively. This is the only option currently
      supported.

  If only the owner or group is specified, then only the owner or group is
  modified. The owner and group names may only consist of digits, alphabet, and
  any of [-_./@a-zA-Z0-9]. The names are case sensitive.

  WARNING: Avoid using '.' to separate user name and group though Linux allows it.
  If user names have dots in them and you are using local file system, you might
  see surprising results since the shell command 'chown' is used for local files.

-concat <target path> <src path> <src path> ... :
  Concatenate existing source files into the target file. Target file and source
  files should be in the same directory.

-copyFromLocal [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <localsrc> ... <dst> :
  Identical to the -put command.

-copyToLocal [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ... <localdst> :
  Identical to the -get command.

-count [-q] [-h] [-v] [-t [<storage type>]] [-u] [-x] [-e] [-s] <path> ... :
  Count the number of directories, files and bytes under the paths
  that match the specified file pattern.  The output columns are:
  DIR_COUNT FILE_COUNT CONTENT_SIZE PATHNAME
  or, with the -q option:
  QUOTA REM_QUOTA SPACE_QUOTA REM_SPACE_QUOTA
        DIR_COUNT FILE_COUNT CONTENT_SIZE PATHNAME
  The -h option shows file sizes in human readable format.
  The -v option displays a header line.
  The -x option excludes snapshots from being calculated.
  The -t option displays quota by storage types.
  It should be used with -q or -u option, otherwise it will be ignored.
  If a comma-separated list of storage types is given after the -t option,
  it displays the quota and usage for the specified types.
  Otherwise, it displays the quota and usage for all the storage
  types that support quota. The list of possible storage types(case insensitive):
  ram_disk, ssd, disk and archive.
  It can also pass the value '', 'all' or 'ALL' to specify all the storage types.
  The -u option shows the quota and
  the usage against the quota without the detailed content summary.The -e option
```

```
-cp [-f] [-p | -p[topax]] [-d] [-t <thread count>] [-q <thread pool queue size>] <src> ... <dst> :
  Copy files that match the file pattern <src> to a destination. When copying
  multiple files, the destination must be a directory.
  Flags :

  -p[topax]                  Preserve file attributes [topx] (timestamps,
                             ownership, permission, ACL, XAttr). If -p is
                             specified with no arg, then preserves timestamps,
                             ownership, permission. If -pa is specified, then
                             preserves permission also because ACL is a
                             super-set of permission. Determination of whether
                             raw namespace extended attributes are preserved is
                             independent of the -p flag.
  -f                         Overwrite the destination if it already exists.
  -d                         Skip creation of temporary file(<dst>._COPYING_).
  -t <thread count>          Number of threads to be used, default is 1.
  -q <thread pool queue size>  Thread pool queue size to be used, default is
                             1024.

-createSnapshot <snapshotDir> [<snapshotName>] :
  Create a snapshot on a directory

-deleteSnapshot <snapshotDir> <snapshotName> :
  Delete a snapshot from a directory

-df [-h] [<path> ...] :
  Shows the capacity, free and used space of the filesystem. If the filesystem has
  multiple partitions, and no path to a particular partition is specified, then
  the status of the root partitions will be shown.

  -h  Formats the sizes of files in a human-readable fashion rather than a number
      of bytes.

-du [-s] [-h] [-v] [-x] <path> ... :
  Show the amount of space, in bytes, used by the files that match the specified
  file pattern. The following flags are optional:

  -s  Rather than showing the size of each individual file that matches the
      pattern, shows the total (summary) size.
  -h  Formats the sizes of files in a human-readable fashion rather than a number
```

```
  -v  option displays a header line.
  -x  Excludes snapshots from being counted.

  Note that, even without the -s option, this only shows size summaries one level
  deep into a directory.

  The output is in the form
        size    disk space consumed     name(full path)

-expunge [-immediate] [-fs <path>] :
  Delete files from the trash that are older than the retention threshold

-find <path> ... <expression> ... :
  Finds all files that match the specified expression and
  applies selected actions to them. If no <path> is specified
  then defaults to the current working directory. If no
  expression is specified then defaults to -print.

  The following primary expressions are recognised:
    -name pattern
    -iname pattern
      Evaluates as true if the basename of the file matches the
      pattern using standard file system globbing.
      If -iname is used then the match is case insensitive.

    -print
    -print0
      Always evaluates to true. Causes the current pathname to be
      written to standard output followed by a newline. If the -print0
      expression is used then an ASCII NULL character is appended rather
      than a newline.

  The following operators are recognised:
    expression -a expression
    expression -and expression
    expression expression
      Logical AND operator for joining two expressions. Returns
      true if both child expressions return true. Implied by the
      juxtaposition of two expressions and so does not need to be
      explicitly specified. The second expression will not be
      applied if the first fails.
```

```
-get [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ... <localdst> :
  Copy files that match the file pattern <src> to the local name. <src> is kept.
  When copying multiple files, the destination must be a directory.
  Flags:

  -p                          Preserves timestamps, ownership and the mode.
  -f                          Overwrites the destination if it already exists.
  -crc                        write CRC checksums for the files downloaded.
  -ignoreCrc                  Skip CRC checks on the file(s) downloaded.
  -t <thread count>           Number of threads to be used, default is 1.
  -q <thread pool queue size>  Thread pool queue size to be used, default is
                              1024.

-getfacl [-R] <path> :
  Displays the Access Control Lists (ACLs) of files and directories. If a
  directory has a default ACL, then getfacl also displays the default ACL.

  -R      List the ACLs of all files and directories recursively.
  <path>  File or directory to list.

-getfattr [-R] {-n name | -d} [-e en] <path> :
  Displays the extended attribute names and values (if any) for a file or
  directory.

  -R             Recursively list the attributes for all files and directories.
  -n name        Dump the named extended attribute value.
  -d             Dump all extended attribute values associated with pathname.
  -e <encoding>  Encode values after retrieving them.Valid encodings are "text",
                 "hex", and "base64". Values encoded as text strings are enclosed
                 in double quotes ("), and values encoded as hexadecimal and
                 base64 are prefixed with 0x and 0s, respectively.
  <path>         The file or directory.

-getmerge [-nl] [-skip-empty-file] <src> <localdst> :
  Get all the files in the directories that match the source file pattern and
  merge and sort them to only one file on local fs. <src> is kept.

  -nl               Add a newline character at the end of each file.
  -skip-empty-file  Do not add new line character for empty file.

-head <file> :
```

```
-text [-ignoreCrc] <src> ... :
  Takes a source file and outputs the file in text format.
  The allowed formats are zip and TextRecordInputStream and Avro.

-touch [-a] [-m] [-t TIMESTAMP (yyyyMMdd:HHmmss) ] [-c] <path> ... :
  Updates the access and modification times of the file specified by the <path> to
  the current time. If the file does not exist, then a zero length file is created
  at <path> with current time as the timestamp of that <path>.
  -a Change only the access time
  -m Change only the modification time
  -t TIMESTAMP Use specified timestamp instead of current time
   TIMESTAMP format yyyyMMdd:HHmmss
  -c Do not create any files

-touchz <path> ... :
  Creates a file of zero length at <path> with current time as the timestamp of
  that <path>. An error is returned if the file exists with non-zero length

-truncate [-w] <length> <path> ... :
  Truncate all files that match the specified file pattern to the specified
  length.

  -w  Requests that the command wait for block recovery to complete, if
      necessary.

-usage [cmd ...] :
  Displays the usage for given command or all commands if none is specified.

Generic options supported are:
-conf <configuration file>        specify an application configuration file
-D <property=value>               define a value for a given property
-fs <file:///|hdfs://namenode:port> specify default filesystem URL to use, overrides 'fs.defaultFS' property from configurations.
-jt <local|resourcemanager:port>  specify a ResourceManager
-files <file1,...>                specify a comma-separated list of files to be copied to the map reduce cluster
-libjars <jar1,...>               specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...>          specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] [commandOptions]
```

4. List the contents of the root directory in HDFS
hadoop fs -ls /

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /
Found 3 items
drwxr-xr-x   - hadoop supergroup          0 2025-01-29 11:49 /logs
drwxr-xr-x   - hadoop supergroup          0 2025-01-29 11:48 /test1
drwxr-xr-x   - hadoop supergroup          0 2025-01-30 05:21 /user
```

5. Report the amount of space used and available on currently mounted filesystem
hadoop fs -df hdfs:/

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -df hdfs:/
Filesystem                      Size     Used      Available  Use%
hdfs://localhost:9000  1081101176832  9457664  1022186688512    0%
```

6. Count the number of directories,files and bytes under the paths that match the specified file pattern
hadoop fs -count hdfs:/

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -count hdfs:/
           14           14             9278479 hdfs:///
```

7. Create a new directory named â€œhadoopâ€ below the /user/training directory in HDFS.
hadoop fs -mkdir /user/training/hadoop

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -mkdir /user/training/hadoop
mkdir: `/user/training/hadoop': File exists
```

8. Add a sample text file from the local directory named â€œdataâ€ to the new directory you created in HDFS during the previous step.
hadoop fs -put data/sample.txt /user/training/hadoop

```
hadoop@DESKTOP-2LPHD1B:~$ echo "This is a sample file" > data/sample.txt
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -put data/sample.txt /user/training/hadoop
put: `/user/training/hadoop/sample.txt': File exists
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop
Found 1 items
-rw-r--r--   1 hadoop supergroup         12 2025-01-30 05:24 /user/training/hadoop/sample.txt
```

9. List the contents of this new directory in HDFS.
hadoop fs -ls /user/training/hadoop

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop
Found 1 items
-rw-r--r--   1 hadoop supergroup         12 2025-01-30 05:24 /user/training/hadoop/sample.txt
```

10. Add the entire local directory called â€œretailâ€ to the /user/training directory in HDFS.
hadoop fs -put data/retail /user/training/hadoop

```
hadoop@DESKTOP-2LPHD1B:~$ mkdir -p data/retail
hadoop@DESKTOP-2LPHD1B:~$ echo "Customer data" > data/retail/customers
hadoop@DESKTOP-2LPHD1B:~$ echo "Product data" > data/retail/products
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -put data/retail /user/training/hadoop
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop
Found 2 items
drwxr-xr-x   - hadoop supergroup          0 2025-01-30 12:52 /user/training/hadoop/retail
-rw-r--r--   1 hadoop supergroup         12 2025-01-30 05:24 /user/training/hadoop/sample.txt
```

11. Delete a file â€˜customersâ€™ from the â€œretailâ€ directory.
hadoop fs -rm hadoop/retail/customers

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -rm /user/training/hadoop/retail/customers
Deleted /user/training/hadoop/retail/customers
```

12. Delete all files from the â€œretailâ€ directory using a wildcard.
hadoop fs -rm hadoop/retail/*

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -rm /user/training/hadoop/retail/*
Deleted /user/training/hadoop/retail/products
Deleted /user/training/hadoop/retail/sample.txt
```

13. Remove the entire retail directory and all of its contents in HDFS.
hadoop fs -rm -r hadoop/retail

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -rm -r /user/training/hadoop/retail
Deleted /user/training/hadoop/retail
```

14. Add the purchases.txt file from the local directory named â€œ/home/training/â€ to the hadoop directory you created in HDFS
hadoop fs -copyFromLocal /home/training/purchases.txt hadoop/

```
hadoop@DESKTOP-2LPHD1B:~$ mkdir -p ~/training
hadoop@DESKTOP-2LPHD1B:~$ echo "Purchase history data" > ~/training/purchases.txt
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -copyFromLocal /~/training/purchases.txt /user/training/hadoop/
copyFromLocal: `/~/training/purchases.txt': No such file or directory
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -copyFromLocal ~/training/purchases.txt /user/training/hadoop/
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop
Found 2 items
-rw-r--r--   1 hadoop supergroup         22 2025-01-30 13:00 /user/training/hadoop/purchases.txt
-rw-r--r--   1 hadoop supergroup         12 2025-01-30 05:24 /user/training/hadoop/sample.txt
```

15. To view the contents of your text file purchases.txt which is present in your hadoop directory.
hadoop fs -cat hadoop/purchases.txt

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -cat /user/training/hadoop/purchases.txt
Purchase history data
```

16. cp is used to copy files between directories present in HDFS
hadoop fs -cp /user/training/*.txt /user/training/hadoop

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/
Found 2 items
drwxr-xr-x   - hadoop supergroup          0 2025-01-30 13:00 /user/training/hadoop
drwxr-xr-x   - hadoop supergroup          0 2025-01-30 05:26 /user/training/retail
hadoop@DESKTOP-2LPHD1B:~$ echo "File 1 content" > file1.txt
hadoop@DESKTOP-2LPHD1B:~$ echo "File 2 content" > file2.txt
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -put file1.txt /user/training/
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -put file2.txt /user/training/
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/
Found 4 items
-rw-r--r--   1 hadoop supergroup         15 2025-01-30 13:03 /user/training/file1.txt
-rw-r--r--   1 hadoop supergroup         15 2025-01-30 13:03 /user/training/file2.txt
drwxr-xr-x   - hadoop supergroup          0 2025-01-30 13:00 /user/training/hadoop
drwxr-xr-x   - hadoop supergroup          0 2025-01-30 05:26 /user/training/retail
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -cp /user/training/*.txt /user/training/hadoop/
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop
Found 4 items
-rw-r--r--   1 hadoop supergroup         15 2025-01-30 13:03 /user/training/hadoop/file1.txt
-rw-r--r--   1 hadoop supergroup         15 2025-01-30 13:03 /user/training/hadoop/file2.txt
-rw-r--r--   1 hadoop supergroup         22 2025-01-30 13:00 /user/training/hadoop/purchases.txt
-rw-r--r--   1 hadoop supergroup         12 2025-01-30 05:24 /user/training/hadoop/sample.txt
```

17. â€˜-getâ€™ command can be used alternaively to â€˜-copyToLocalâ€™ command
hadoop fs -get hadoop/sample.txt /home/training/

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -get /user/training/hadoop/sample.txt ~/training/
```

18. Display last kilobyte of the file â€œpurchases.txtâ€ to stdout.
hadoop fs -tail hadoop/purchases.txt

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -tail /user/training/hadoop/purchases.txt
Purchase history data
```

19. Default file permissions are 666 in HDFS. Use â€˜-chmodâ€™ command to change permissions of a file
hadoop fs -ls hadoop/purchases.txt
sudo -u hdfs hadoop fs -chmod 600 hadoop/purchases.txt

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop/purchases.txt
-rw-r--r--   1 hadoop supergroup         37 2025-01-30 13:06 /user/training/hadoop/purchases.txt
```

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -chmod 600 /user/training/hadoop/purchases.txt
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop/purchases.txt
-rw-------   1 hadoop supergroup         37 2025-01-30 13:06 /user/training/hadoop/purchases.txt
```

20. Default names of owner and group are training,training. Use â€˜-chownâ€™ to change owner name and group name simultaneously

hadoop fs -ls hadoop/purchases.txt
sudo -u hdfs hadoop fs -chown root:root hadoop/purchases.txt

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop/purchases.txt
-rw-------   1 hadoop supergroup         37 2025-01-30 13:06 /user/training/hadoop/purchases.txt

hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -chown root:root /user/training/hadoop/purchases.txt

hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop/purchases.txt
-rw-------   1 root root         37 2025-01-30 13:06 /user/training/hadoop/purchases.txt
```

21. Default name of group is training Use â€˜-chgrpâ€™ command to change group name
hadoop fs -ls hadoop/purchases.txt
sudo -u hdfs hadoop fs -chgrp training hadoop/purchases.txt

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop/purchases.txt
-rw-------   1 root root         37 2025-01-30 13:06 /user/training/hadoop/purchases.txt
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -chgrp training /user/training/hadoop/purchases.txt
```

22. Move a directory from one location to other
hadoop fs -mv hadoop apache_hadoop

```
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -chgrp training /user/training/hadoop/purchases.txt
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -mv /user/training/hadoop /user/training/apache_hadoop
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/apache_hadoop
Found 5 items
-rw-r--r--   1 hadoop supergroup         15 2025-01-30 13:03 /user/training/apache_hadoop/file1.txt
-rw-r--r--   1 hadoop supergroup         15 2025-01-30 13:03 /user/training/apache_hadoop/file2.txt
-rw-r--r--   1 hadoop supergroup       2048 2025-01-30 13:06 /user/training/apache_hadoop/largefile.txt
-rw-------   1 root   training          37 2025-01-30 13:06 /user/training/apache_hadoop/purchases.txt
-rw-r--r--   1 hadoop supergroup         12 2025-01-30 05:24 /user/training/apache_hadoop/sample.txt
hadoop@DESKTOP-2LPHD1B:~$ hadoop fs -ls /user/training/hadoop
ls: `/user/training/hadoop': No such file or directory
```

Shell Commands:

1. Move `data.csv` to HDFS:

   I. Create a directory & Upload File:

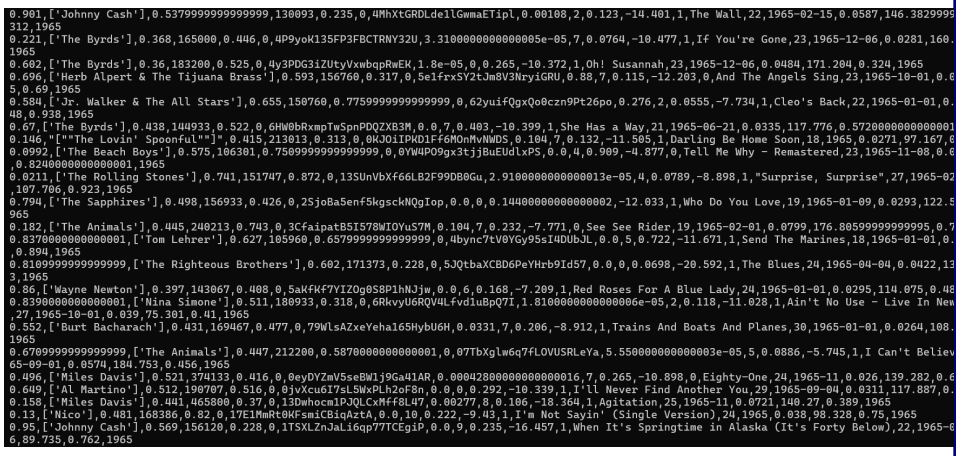
```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ cp /mnt/c/Users/Pranav/Downloads/data.csv ./
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -put data.csv /user/hadoop/
put: `/user/hadoop/': No such file or directory: `hdfs://localhost:9000/user/ha
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -mkdir -p /user/hadoop/
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -put data.csv /user/hadoop/
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -ls /user/hadoop/
Found 1 items
-rw-r--r--   1 hadoop supergroup   29673575 2025-02-05 04:12 /user/hadoop/data.
```

2. File Viewing & Manipulation Commands:

I. View File Contents:

 # View entire file

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -cat /user/hadoop/data.csv
```



 # View first 10 lines

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -head /user/hadoop/data.csv
acousticness,artists,danceability,duration_ms,energy,explicit,id,instrumentalness,key,liveness,loudness,mode,name,popularity,release_date,speech
valence,year
0.995,['Carl Woitschach'],0.708,158648,0.195,0,6KbQ3uYMLKb5jDxLF7wYDD,0.563,10,0.151,-12.428,1,Singende Bataillone 1. Teil,0,1928,0.0506,118.469
0.994,"['Robert Schumann', 'Vladimir Horowitz']",0.379,282133,0.0135,0,6KuQTIuiKoTTkLXKrwLLPV,0.901,8,0.0763,-28.454,1,"Fantasiestücke, Op. 111:
ento",0,1928,0.0462,83.97200000000002,0.0767,1928
0.604,['Seweryn Goszczyński'],0.7490000000000001,104300,0.22,0,6L63VW0PibdM1HDSBoqnoM,0.0,5,0.119,-19.924,0,Chapter 1.18 - Zamek kaniowski,0,192
177,0.88,1928
0.995,['Francisco Canaro'],0.7809999999999999,180760,0.13,0,6M94FkXd15sOAOQYRnWPN8,0.887,1,0.111,-14.734000000000002,0,Bebamos Juntos - Instrume
erizado),0,1928-09-25,0.0926,108.003,0.72,1928
0.99,"['Frédéric Chopin', 'Vladimir Horowitz']",0.21,687733,0.204,0,6N6tiFZ9vLTSOIxkj8qKrd,0.908,11,0.098,-16.hadoop@DESKTOP-2LPHD1B:~/hadoop$
```

 # View last few lines

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -tail /user/hadoop/data.csv
8501,0.585,0,52Cpyvd2dKb6XRn313nH87,1.2e-06,8,0.112,-4.4510000000000005,1,Ojos De Maniaco,68,2020-02-28,0.0374,97.479,0.934,2020
0.173,"['DripReport', 'Tyga']",0.875,163800,0.443,1,4KppkflX7I3vJQk7urOJaS,3.24e-05,1,0.0891,-7.461,1,Skechers (feat. Tyga) - Remix,75,2020-05-
000000002,100.012,0.306,2020
0.0167,"['Leon Bridges', 'Terrace Martin']",0.7190000000000001,167468,0.385,0,1ehhGlTvjtHo2e4xJFB0SZ,0.0313,8,0.111,-10.907,1,Sweeter (feat. Te
,64,2020-06-08,0.0403,128.0,0.27,2020
0.5379999999999999,"['Kygo', 'Oh Wonder']",0.514,180700,0.539,0,52eycxprLhK3lPcRLbQiVk,0.00233,7,0.108,-9.332,1,How Would I Know,70,2020-05-29,
.153,2020
0.0714,"['Cash Cash', 'Andy Grammer']",0.6459999999999999,167308,0.7609999999999999,0,3wYOGJYD31sLRmBgCvWxa4,0.0,1,0.222,-2.557,1,I Found You,7
0.0385,129.916,0.472,2020
0.109,['Ingrid Andress'],0.512,214787,0.428,0,60RFlt48hm0l4Fu0JoccOl,0.0,0,0.105,-7.387,1,More Hearts Than Mine,65,2020-03-27,0.0271,80.5880000
,2020
```

Ii. Append to File

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -appendToFile /mnt/data/data.csv /user/hadoop/da
appendToFile: /mnt/data/data.csv
```

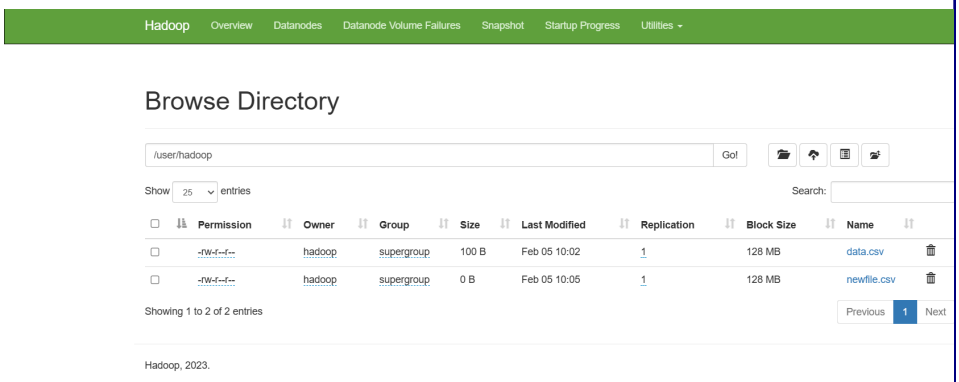
Iii. Truncate (Shrink) File:

 # Trim file to 100 bytes

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -truncate -w 100 /user/hadoop/data.csv
Waiting for /user/hadoop/data.csv ...
Truncated /user/hadoop/data.csv to length: 100
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -cat /user/hadoop/data.csv
acousticness,artists,danceability,duration_ms,energy,explicit,id,instrumentalness,key,livene
```

Iv. Create an Empty File:

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -touchz /user/hadoop/newfile.csv
```



3. File Management Commands:

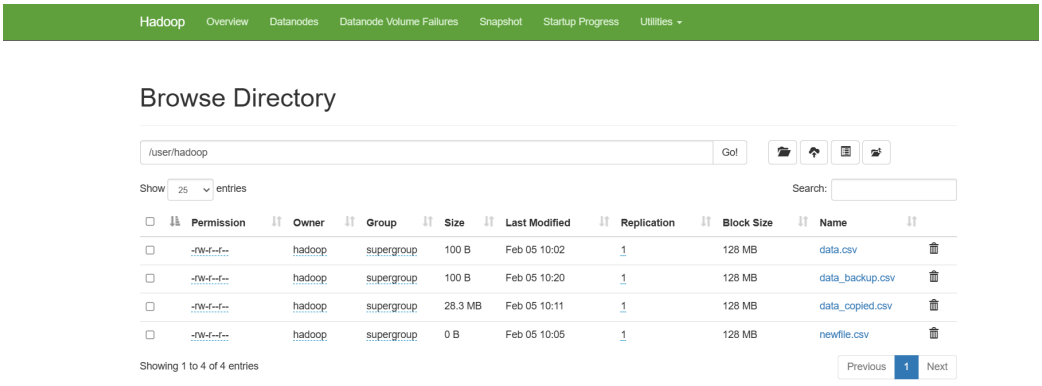
I. Copy Files:

 # From local to HDFS

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -copyFromLocal /mnt/data/data.csv /user/hadoop/data_copied.csv
copyFromLocal: `/user/hadoop/data_copied.csv': File exists
```

 # Copy within HDFS

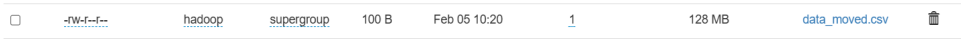
```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -cp /user/hadoop/data.csv /user/hadoop/data_backup.csv
```



Ii. Move & Rename Files:

 # Rename/move file

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -mv /user/hadoop/data_backup.csv /user/hadoop/data_moved.csv
hadoop@DESKTOP-2LPHD1B:~/hadoop$
```



4. File Permissions & Ownership:

I. Change File Permissions:

 # Set read/write/execute permissions

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -chmod 755 /user/hadoop/data.csv
```

Ii. Change File Ownership:

 # Change owner
 # Change group ownership

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -chown hadoop:hadoop /user/hadoop/data.csv
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -chgrp hadoop /user/hadoop/data.csv
hadoop@DESKTOP-2LPHD1B:~/hadoop$
```

5. File Metadata & Information:
I. Check File Status

 # Get metadata

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -stat /user/hadoop/data.csv
2025-02-05 04:32:37
```

 # Get file checksum

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -checksum /user/hadoop/data.csv
/user/hadoop/data.csv    MD5-of-0MD5-of-512CRC32C        000002000000000000000099e888218c03b09606135002f7f16415
```

Ii. Check Disk Usage

 # Check HDFS disk space
 # Show human-readable file sizes
 # Show directory size summary

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -df /
Filesystem                      Size      Used      Available  Use%
hdfs://localhost:9000  1081101176832  69341184  1022006296576    0%
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -du -h /user/hadoop/
100     100     /user/hadoop/data.csv
28.3 M  28.3 M  /user/hadoop/data_copied.csv
100     100     /user/hadoop/data_moved.csv
0       0       /user/hadoop/newfile.csv
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -dus -h /user/hadoop/
dus: DEPRECATED: Please use 'du -s' instead.
28.3 M  28.3 M  /user/hadoop
```

Iii. Count Files in a Directory

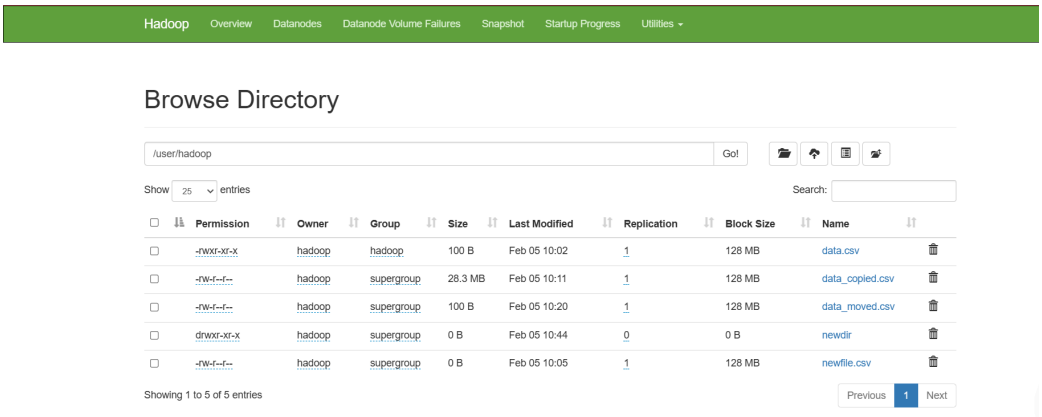
 # Count files, directories, and storage

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -count /user/hadoop/
           1            4           29673775 /user/hadoop
```

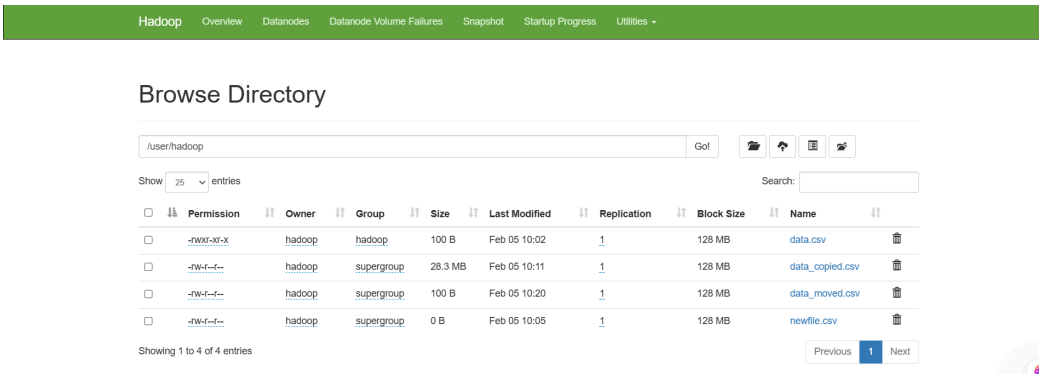
6. Directory Operations
   I. Create & Delete Directories

 # Create new directory
 # Remove empty directory

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -mkdir /user/hadoop/newdir
```




```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -rmdir /user/hadoop/newdir
```



Ii. List Files in a Directory

 # Show file list
 # Recursively list all files

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -ls /user/hadoop/
Found 4 items
-rwxr-xr-x   1 hadoop hadoop            100 2025-02-05 04:32 /user/hadoop/data.csv
-rw-r--r--   1 hadoop supergroup   29673575 2025-02-05 04:41 /user/hadoop/data_copied.csv
-rw-r--r--   1 hadoop supergroup        100 2025-02-05 04:50 /user/hadoop/data_moved.csv
-rw-r--r--   1 hadoop supergroup          0 2025-02-05 04:35 /user/hadoop/newfile.csv
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -lsr /user/hadoop/
lsr: DEPRECATED: Please use 'ls -R' instead.
-rwxr-xr-x   1 hadoop hadoop            100 2025-02-05 04:32 /user/hadoop/data.csv
-rw-r--r--   1 hadoop supergroup   29673575 2025-02-05 04:41 /user/hadoop/data_copied.csv
-rw-r--r--   1 hadoop supergroup        100 2025-02-05 04:50 /user/hadoop/data_moved.csv
-rw-r--r--   1 hadoop supergroup          0 2025-02-05 04:35 /user/hadoop/newfile.csv
```

7. Snapshots (Versioning)

 # Create snapshot

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfsadmin -allowSnapshot /user/hadoop/
Allowing snapshot on /user/hadoop/ succeeded
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -createSnapshot /user/hadoop/ snapshot1
Created snapshot /user/hadoop/.snapshot/snapshot1
```

 # Rename snapshot
 # Delete snapshot

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -renameSnapshot /user/hadoop/ snapshot1 snapshot_renamed
Renamed snapshot snapshot1 to snapshot_renamed under hdfs://localhost:9000/user/hadoop
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -deleteSnapshot /user/hadoop/ snapshot_renamed
Deleted snapshot snapshot_renamed under hdfs://localhost:9000/user/hadoop
hadoop@DESKTOP-2LPHD1B:~/hadoop$
```

8. Find, Search & Testing
   I. Find Files

 # Find all CSV files

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -find /user/hadoop/ -name "*.csv"
/user/hadoop/data.csv
/user/hadoop/data_copied.csv
/user/hadoop/data_moved.csv
/user/hadoop/newfile.csv
hadoop@DESKTOP-2LPHD1B:~/hadoop$
```

II. Test File/Directory Existence

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -test -e /user/hadoop/data.csv && echo "File exists" || echo "File not found"
File exists
hadoop@DESKTOP-2LPHD1B:~/hadoop$
```

9. Display File as Text

 # Convert file to readable text format

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -text /user/hadoop/data.csv
acousticness,artists,danceability,duration_ms,energy,explicit,id,instrumentalness,key,liveness,loudn
```

10. File Expunge & Cleanup

 # Permanently remove deleted files from trash

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -expunge
hadoop@DESKTOP-2LPHD1B:~/hadoop$
```

CONCLUSION: Through this experiment I learned how to write basic commands for Hadoop and How different shell commands work.

Experiment no. 3	
AIM :	To Perform File management tasks in Hadoop
Theory	Just like a traditional file system (e.g., Windows or Linux), Hadoop provides a command-line interface for file system operations in HDFS. These operations allow users to: ● Create directories ● Upload/download files ● Rename or move files ● Delete files/directories ● Check disk usage and file details
Q&A:	1. do not remove stopwords? Code:

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

public class WordCount {

    public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();
        private Set<String> stopwords = new HashSet<>();

        protected void setup(Context context) throws IOException {
            Scanner scanner = new Scanner(context.getConfiguration().get("stopword.path"));
            while (scanner.hasNext()) {
                stopwords.add(scanner.next().toLowerCase());
            }
            scanner.close();
        }
```

```
        public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
            String line = value.toString().replaceAll("[^a-zA-Z ]", "").toLowerCase();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {
                String token = tokenizer.nextToken();
                if (!stopwords.contains(token)) {
                    word.set(token);
                    context.write(word, one);
                }
            }
        }
    }

    public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();

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

```
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        conf.set("stopword.path", args[2]); // Stopword file path

        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

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

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Output:

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -cat /user/hadoop/output/part-r-00000 | sort -nrk2 | head -25
the     57
your    32
of      32
you     24
and     20
to      19
that    14
on      14
words   13
will    13
have    13
application     13
in      12
data    11
is      10
be      10
are     10
output  9
each    9
a       9
stopwords       8
common  8
this    7
most    7
as      7
```

2. What are the 25 most common words and the number of occurrences of each when you do remove stopwords?

Code:

```java
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.HashSet;
import java.util.Set;
import java.util.StringTokenizer;

public class WordCount {

    public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();
        private Set<String> stopwords = new HashSet<>();

        // Setup method to load stopwords from HDFS
        protected void setup(Context context) throws IOException {
            Path stopwordPath = new Path(context.getConfiguration().get("stopword.path"));
            FileSystem fs = FileSystem.get(context.getConfiguration());
            BufferedReader br = new BufferedReader(new InputStreamReader(fs.open(stopwordPath)));

            String line;
            while ((line = br.readLine()) != null) {
                stopwords.add(line.trim().toLowerCase()); // Add stopword to HashSet
            }
            br.close();
        }
```

```java
        public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
            String line = value.toString().replaceAll("[^a-zA-Z ]", "").toLowerCase();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {
                String token = tokenizer.nextToken();
                if (!stopwords.contains(token)) { // Ignore stopwords
                    word.set(token);
                    context.write(word, one);
                }
            }
        }
    }

    public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();

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

```java
    public static void main(String[] args) throws Exception {
        if (args.length < 3) {
            System.err.println("Usage: WordCount <input path> <output path> <stopwords path>");
            System.exit(-1);
        }

        Configuration conf = new Configuration();
        conf.set("stopword.path", args[2]); // Set stopwords file path in HDFS

        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

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

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Output:

3. Based on the output of your application, how does removing stop words affect the total amount of bytes output by your mappers? Name one concrete way that this would affect the performance of your application.

- Removing stopwords **reduces the total number of key-value pairs emitted by the mappers**. Since stopwords are common words (e.g., "the," "and," "is"), they would have been among the **most frequent** words in the dataset. By filtering them out, the amount of intermediate data (key-value pairs) is **significantly reduced**.

- **Before removing stopwords:** The mapper emits **all words** as key-value pairs (word, 1), even frequent but non-meaningful words like "the" and "is".
- **After removing stopwords:** The mapper **ignores** stopwords, reducing the total number of key-value pairs written to disk and sent over the network.

My output:

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -cat /user/hadoop/output/part-r-00000 | sort -nrk2 | head -25
words   13
will    13
application     13
data    11
output  9
stopwords       8
common  8
word    6
pairs   6
key     6
value   5
size    5
setup   5
mappers 5
list    5
hadoop  5
assume  5
stop    4
per     4
number  4
now     4
must    4
keyvalue        4
keyspace        4
input   4
```

Since **common stopwords were removed**, the word count distribution now contains **more meaningful words** like "data", "output", "application", etc.

Concrete Performance Improvement

1. **Reduced File I/O**
   - The mapper writes fewer key-value pairs to **disk**, leading to **faster processing** and **lower storage requirements**.
2. **Less Network Transfer**
   - Less intermediate data is shuffled between the **Mapper and Reducer**, reducing network congestion and improving overall **job execution time**.
3. **Faster Reducer Execution**
   - The reducer now **processes fewer unique keys**, making aggregation **faster**.

4. Based on the output of your application, what is the size of your keyspace with and without removing stopwords? How does this

correspond to the number of stopwords you have chosen to remove?

My output:

```
hadoop@DESKTOP-2LPHD1B:~/hadoop$ hdfs dfs -cat /user/hadoop/output/part-r-00000 | sort -nrk2 | head -25
the     57
your    32
of      32
you     24
and     20
to      19
that    14
on      14
words   13
will    13
have    13
application     13
in      12
data    11
is      10
be      10
are     10
output  9
each    9
a       9
stopwords       8
common  8
this    7
most    7
as      7
```

Understanding Keyspace

The **keyspace** refers to the **number of unique words** that appear in the output.

- **Before removing stopwords**: Every word, including common stopwords, is included in the keyspace.
- **After removing stopwords**: The keyspace decreases because stopwords are ignored by the Mapper.

Keyspace Calculation (Using Your Output)

- From your output **before removing stopwords**, we see that **common words like "the", "of", "you", "to", "and"** are still present.

Let's calculate the **keyspace size**:

1. **Keyspace WITH Stopwords**
   - Your output **contains at least 25 unique words**.

- Since **stopwords are present**, the **actual keyspace is larger** (likely much higher than 25 in the full dataset).

2. **Keyspace WITHOUT Stopwords**

   If you apply the **stopword filter** using stopwords.txt, **common words like "the", "of", "you", "to", "and"** would be removed.

   From your **stopwords.txt** (which contains 127 words):

   - Many words in your current output are **in the stopword list**.
   - **Removing those stopwords will reduce the keyspace.**

   Let's estimate the **difference in keyspace size**:

   - **With stopwords**: At least **25 words** (as seen in the output).
   - **Without stopwords**: Removing **all 127 stopwords** reduces the keyspace.
   - New keyspace = **(Original Keyspace - Number of Stopwords Removed)**.

   If we assume **all 127 stopwords were in the dataset**, the **difference in keyspace size** would be **127 words**.

Final Answer

- **Keyspace WITH stopwords** = **Much larger** (includes common words like "the", "of", "you", etc.).
- **Keyspace WITHOUT stopwords** = **Reduced by approximately the number of stopwords removed (up to 127 fewer words)**.
- **Correspondence**: The keyspace reduction **is equal to the number of stopwords that actually appear in the dataset**.

CONCLUSION:	Thus we have understood the working of File task managing using hadoop. I understood the working of MapReduce and how neglecting the common words increases the performance.

Experiment no. 4	
AIM :	Install Apache PySpark(Apache Spark) Using Miniconda
Theory	Apache Spark is a fast, open-source, distributed computing framework designed for processing large-scale datasets. It supports various programming languages such as Scala, Java, Python, and R. One of Spark's key advantages over traditional MapReduce is its ability to perform **in-memory computations**, which greatly enhances speed and performance, especially for iterative algorithms and interactive data analytics. **PySpark** is the Python API for Apache Spark, enabling Python programmers to harness the power of Spark for tasks like batch processing, interactive queries, streaming, and machine learning. PySpark provides high-level APIs for working with **Resilient Distributed Datasets (RDDs)**, **DataFrames**, and **SQL** operations. To ensure a clean and isolated development environment, **Miniconda**—a minimal version of Anaconda—is used for installation. Miniconda allows developers to manage Python versions and packages without interfering with system-level Python setups. Using Miniconda ensures reproducibility, avoids dependency conflicts, and simplifies the setup process through environment management. By installing PySpark using Miniconda, users can quickly set up a reliable development environment for big data applications, data processing, and machine learning pipelines.
Installation Process:	**To install Miniconda:** **1)** wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux - x86_64.sh

2) bash Miniconda3-latest-Linux-x86_64.sh

To create Pyspark environment and install pyspark:

```
root@DESKTOP-2LPHD1B:~# source ~/.bashrc
(base) root@DESKTOP-2LPHD1B:~# conda create -n pyspark_env python=3.8
Channels:
```

```
(base) root@DESKTOP-2LPHD1B:~# conda activate pyspark_env
(pyspark_env) root@DESKTOP-2LPHD1B:~# conda install -c conda-forge pyspark
```

To install SPARK:

```
(pyspark_env) root@DESKTOP-2LPHD1B:~# wget https://archive.apache.org/dist/spark/spark-3.5.0/spark-3.5.0-bin-hadoop3.tgz
--2025-03-05 03:33:17--  https://archive.apache.org/dist/spark/spark-3.5.0/spark-3.5.0-bin-hadoop3.tgz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 400395283 (382M) [application/x-gzip]
Saving to: 'spark-3.5.0-bin-hadoop3.tgz'

spark-3.5.0-bin-hadoop3.tgz   100%[=============================================>] 381.85M  1.04MB/s    in 6m 57s

2025-03-05 03:40:41 (937 KB/s) - 'spark-3.5.0-bin-hadoop3.tgz' saved [400395283/400395283]
```

CONCLUSION:	Apache Spark (PySpark) was successfully installed and configured on the Linux system. The setup was verified by launching PySpark, confirming that SparkContext and SparkSession were initialized without errors. This enables efficient big data processing and analytics using Python, leveraging Spark's powerful distributed computing capabilities.

Experiment no. 5	
AIM :	To Execute wordcount program in pyspark. Compare the execution time
Theory	The WordCount program is a classic example used to demonstrate the capabilities of distributed data processing frameworks like Hadoop and Spark. It counts the frequency of each word in a large text file and is ideal for performance benchmarking. 🔹 **Hadoop MapReduce:** **Hadoop uses a disk-based, two-stage processing model:** ● The Map phase extracts key-value pairs (e.g., `<word, 1>`), ● The Reduce phase aggregates the counts for each word. While robust and fault-tolerant, Hadoop MapReduce incurs higher disk I/O overhead, as intermediate data is written to disk between stages, making it slower for iterative and real-time tasks. 🔹 **Apache Spark (PySpark):** PySpark is the Python API for Apache Spark, which uses a directed acyclic graph (DAG) execution model and performs most operations in-memory, drastically reducing disk I/O. This results in faster execution times, especially for repeated or iterative operations. **When running the same WordCount logic in PySpark:** ● Text data is loaded using RDDs or DataFrames. ● Operations like `flatMap()`, `map()`, and `reduceByKey()` are applied to compute word frequencies. ● The execution is parallelized and optimized using Spark's DAG

	scheduler.

Code & Output:	**Wordcount Program in Pyspark:**
	
	`(base) root@DESKTOP-2LPHD1B:~# nano wordcount.py`
	`(base) root@DESKTOP-2LPHD1B:~# nano word.py`
	`(base) root@DESKTOP-2LPHD1B:~# time spark-submit word.py`
	
	`from pyspark.sql import SparkSession`
	`import time`
	
	`# Initialize Spark Session`
	`spark = SparkSession.builder.appName("WordCount").getOrCreate()`
	
	`# Start timing`
	`start_time = time.time()`
	
	`# Read input file`
	`input_path = "wordcount.txt"  # Change this to your text file path`
	`text_rdd = spark.sparkContext.textFile(input_path)`
	
	`# Process text: Split into words, map to (word, 1), and reduce by key`
	`word_counts = (`
	`    text_rdd`
	`    .flatMap(lambda line: line.split())  # Split lines into words`
	`    .map(lambda word: (word, 1))  # Map each word to (word, 1)`
	`    .reduceByKey(lambda a, b: a + b)  # Sum occurrences of each word`
	`)`
	
	`# Collect results (forces execution) and capture end time`
	`result = word_counts.collect()`
	
	`end_time = time.time()`
	
	Output:
	
	`Hadoop: 2`
	`fast: 1`
	`Java: 1`
	`powerful: 1`
	`is: 3`
	`great: 1`
	`Execution Time: 1652.76 ms`
	
	WordCount using MapReduce in Hadoop:
	
	`hadoop@DESKTOP-2LPHD1B:~$ hdfs dfs -mkdir -p /user/hadoop/input`
	`hadoop@DESKTOP-2LPHD1B:~$ nano sample.txt`
	`hadoop@DESKTOP-2LPHD1B:~$ hdfs dfs -put sample.txt /user/hadoop/input/`
	`put: '/user/hadoop/input/sample.txt': File exists`
	`hadoop@DESKTOP-2LPHD1B:~$ hdfs dfs -rm /user/hadoop/input/sample.txt`
	`Deleted /user/hadoop/input/sample.txt`
	`hadoop@DESKTOP-2LPHD1B:~$ hdfs dfs -put sample.txt /user/hadoop/input/`
	`hadoop@DESKTOP-2LPHD1B:~$ hadoop jar WordCountSorted.jar WordCountSorted /user/hadoop/input /user/hadoop/output`

GNU nano 7.2 WordCountSorted.java

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.WritableComparable;
import org.apache.hadoop.io.WritableComparator;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import java.io.IOException;
import java.util.StringTokenizer;

public class WordCountSorted {

    public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken().replaceAll("[^a-zA-Z]", "").toLowerCase());
                if (word.getLength() > 0) {
                    context.write(word, one);
                }
            }
        }
    }

    public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
            int sum = 0;
```

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

    public static class FrequencyMapper extends Mapper<Object, Text, IntWritable, Text> {
        public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
            String[] tokens = value.toString().split("\t");
            if (tokens.length == 2) {
                String word = tokens[0];
                int count = Integer.parseInt(tokens[1]);
                context.write(new IntWritable(count), new Text(word));
            }
        }
    }

    public static class FrequencyReducer extends Reducer<IntWritable, Text, Text, IntWritable> {
        public void reduce(IntWritable key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
            for (Text word : values) {
                context.write(word, key);
            }
        }
    }

    public static class DescendingIntComparator extends WritableComparator {
        protected DescendingIntComparator() {
            super(IntWritable.class, true);
        }

        @SuppressWarnings("rawtypes")
        @Override
        public int compare(WritableComparable a, WritableComparable b) {
```

```java
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();

    long startTime = System.currentTimeMillis();

    Job job1 = Job.getInstance(conf, "word count");
    job1.setJarByClass(WordCountSorted.class);
    job1.setMapperClass(TokenizerMapper.class);
    job1.setCombinerClass(IntSumReducer.class);
    job1.setReducerClass(IntSumReducer.class);
    job1.setOutputKeyClass(Text.class);
    job1.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job1, new Path(args[0]));
    FileOutputFormat.setOutputPath(job1, new Path(args[1] + "/temp"));

    if (!job1.waitForCompletion(true)) {
        System.exit(1);
    }

    Job job2 = Job.getInstance(conf, "sort by frequency");
    job2.setJarByClass(WordCountSorted.class);
    job2.setMapperClass(FrequencyMapper.class);
    job2.setReducerClass(FrequencyReducer.class);
    job2.setSortComparatorClass(DescendingIntComparator.class);
    job2.setMapOutputKeyClass(IntWritable.class);
    job2.setMapOutputValueClass(Text.class);
    job2.setOutputKeyClass(Text.class);
    job2.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job2, new Path(args[1] + "/temp"));
    FileOutputFormat.setOutputPath(job2, new Path(args[1] + "/sorted"));

    boolean success = job2.waitForCompletion(true);

    long endTime = System.currentTimeMillis();
    System.out.println("Execution Time: " + (endTime - startTime) + " milliseconds");
```

Output:

```
hadoop@DESKTOP-2LPHD1B:~$ hdfs dfs -cat /user/hadoop/output/sorted/part-r-00000
is      3
hadoop  2
powerful        1
java    1
great   1
fast    1
```

```
Execution Time: 2900 milliseconds
```

CONCLUSION:

On running a small file, Hadoop is faster than PySpark:

PySpark Execution Time (Small File) → 1.652 s

Hadoop Execution Time (Small File) → 2.900 s

PySpark is **faster** than Hadoop MapReduce (almost **2x faster** in this case).

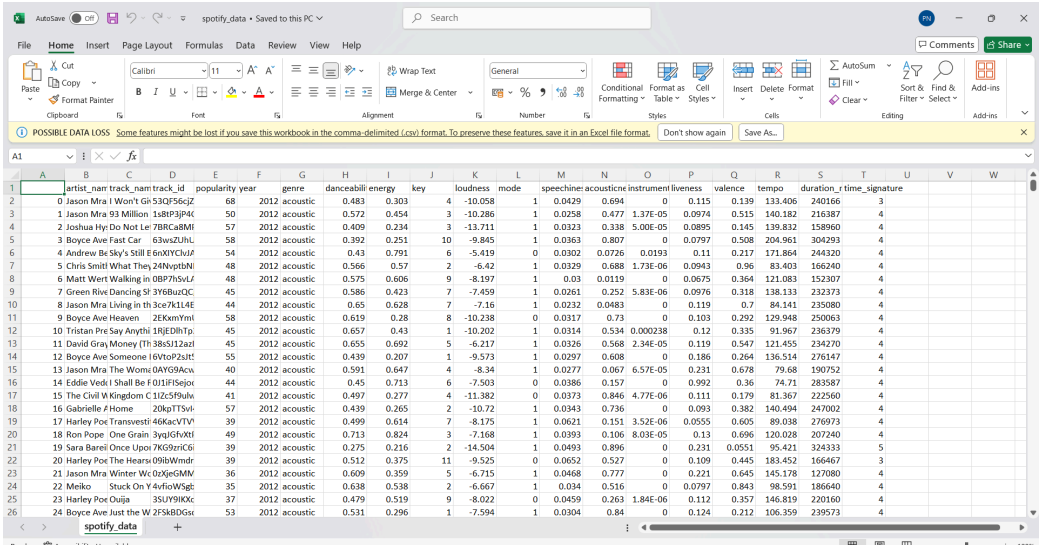
Experiment no. 6	
AIM :	To Analyze a large dataset using Apache Spark
Theory	Apache Spark is a powerful open-source distributed computing engine designed for fast and scalable data processing. Unlike traditional batch processing systems like Hadoop MapReduce, Spark performs computations in memory, significantly reducing the time required for large-scale data analysis. Spark's core abstraction is the Resilient Distributed Dataset (RDD), a fault-tolerant collection of elements that can be operated on in parallel. In addition, DataFrames and Spark SQL provide higher-level APIs that are optimized for structured data analysis, enabling operations similar to SQL queries on massive datasets. When analyzing large datasets with Spark, the process generally includes: • Loading the dataset into Spark (from CSV, JSON, Parquet, HDFS, etc.) • Performing transformations like `filter()`, `map()`, `groupBy()`, `join()`, and actions like `collect()`, `count()`, `show()` • Using Spark SQL or DataFrame API for advanced data exploration • Optionally visualizing or exporting the results for reporting Spark automatically distributes the computation across available nodes in the cluster (or cores in a local machine in pseudo mode), ensuring performance, scalability, and fault tolerance. This makes it ideal for analyzing real-world large datasets such as customer records, web logs, social media data, financial transactions, and more.

Step By Step process with Code & Output:

Loading a Large Dataset into a PySpark DataFrame

Overview

A DataFrame in PySpark is a distributed collection of data organized into named columns, similar to a relational database table. PySpark provides built-in methods for loading data from sources such as CSV.



Loading Data from a CSV File

CSV (Comma-Separated Values) is a common format for storing structured data. PySpark's read.csv() method allows for efficient data ingestion.

Data Cleaning

Overview :

Data cleaning ensures data integrity by handling missing values, removing duplicates, and filtering outliers. This process is crucial for maintaining the accuracy of analytical models.

Handling Missing Values:

Large datasets often contain missing or null values, which can impact analysis and modeling. PySpark provides the following methods to handle missing values:

Removing Duplicate Records:
Duplicate records can skew analysis and must be eliminated.

Handling Outliers:
Outliers are extreme values that may distort analysis results. The Interquartile Range (IQR) method is commonly used to detect and remove them

Python code:

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, when, isnan, count,
desc, avg, sum, stddev
from pyspark.sql.types import StructType, StructField,
StringType, DoubleType, IntegerType

spark = SparkSession.builder \
    .appName("Spotify Data Cleaning and Transformation") \
    .config("spark.executor.memory", "2g") \
    .config("spark.driver.memory", "4g") \
    .getOrCreate()

file_path = "/mnt/c/Users/Pranav/Downloads/spotify_data.csv"

spotify_schema = StructType([
    StructField("index", IntegerType(), True),
    StructField("artist_name", StringType(), True),
    StructField("track_name", StringType(), True),
    StructField("track_id", StringType(), True),
    StructField("popularity", DoubleType(), True),
    StructField("year", IntegerType(), True),
    StructField("genre", StringType(), True),
```

```
    StructField("danceability", DoubleType(), True),
    StructField("energy", DoubleType(), True),
    StructField("key", IntegerType(), True),
    StructField("loudness", DoubleType(), True),
    StructField("mode", IntegerType(), True),
    StructField("speechiness", DoubleType(), True),
    StructField("acousticness", DoubleType(), True),
    StructField("instrumentalness", DoubleType(), True),
    StructField("liveness", DoubleType(), True),
    StructField("valence", DoubleType(), True),
    StructField("tempo", DoubleType(), True),
    StructField("duration_ms", DoubleType(), True),
    StructField("time_signature", IntegerType(), True)
])

df = spark.read \
    .option("header", "true") \
    .schema(spotify_schema) \
    .csv(file_path)

print("DataFrame Schema:")
df.printSchema()
print(f"Total number of records: {df.count()}")
print("Sample data:")
df.show(5)

# ----------------------------------------
# PART 1: DATA CLEANING
# ----------------------------------------
print("\n===== DATA CLEANING =====")

print("\n1. Checking for missing values:")
missing_values = df.select([count(when(col(c).isNull() |
isnan(c), c)).alias(c) for c in df.columns])
missing_values.show()

print("\n2. Filtering out invalid years:")
```

```
initial_count = df.count()
df = df.filter(col("year") >= 1900)
filtered_count = df.count()
print(f"Records with invalid years removed: {initial_count -
filtered_count}")

print("\n3. Handling missing values in numerical columns:")
numeric_cols = [field.name for field in df.schema.fields
                if isinstance(field.dataType, DoubleType) or
isinstance(field.dataType, IntegerType)]

mean_values = {}
for column in numeric_cols:
    mean_val = df.select(avg(col(column))).collect()[0][0]
    mean_values[column] = mean_val
    print(f"Mean value for {column}: {mean_val}")

for column in numeric_cols:
    df = df.withColumn(column,
                       when(col(column).isNull() |
isnan(column), mean_values[column])
                       .otherwise(col(column)))

categorical_cols = [field.name for field in df.schema.fields
                    if isinstance(field.dataType, StringType)]
for column in categorical_cols:
    df = df.withColumn(column,
                       when(col(column).isNull(), "unknown")
                       .otherwise(col(column)))

print("\n4. Removing duplicate records:")
before_dedupe = df.count()
df = df.dropDuplicates(["track_id"])
after_dedupe = df.count()
print(f"Duplicate records removed: {before_dedupe -
after_dedupe}")
```

```
print("\n5. Handling outliers:")
outlier_columns = ["popularity", "danceability", "energy",
"tempo", "loudness"]

for column in outlier_columns:
    stats = df.select(
        avg(col(column)).alias("mean"),
        stddev(col(column)).alias("std_dev")
    ).collect()[0]

    mean = stats["mean"]
    std_dev = stats["std_dev"]

    print(f"Outlier treatment for {column}: mean={mean},
stddev={std_dev}")

    before_outlier = df.count()
    df = df.filter(
        (col(column) <= mean + 3 * std_dev) &
        (col(column) >= mean - 3 * std_dev)
    )
    after_outlier = df.count()
    print(f"Outliers removed from {column}: {before_outlier -
after_outlier}")
```

OUTPUT:

```
(pyspark_env) root@DESKTOP-2LPHD1B:~# nano spotify_analysis.py
(pyspark_env) root@DESKTOP-2LPHD1B:~# python3 spotify_analysis.py
25/03/21 09:02:20 WARN Utils: Your hostname, DESKTOP-2LPHD1B resolves to a loopback address: 127.0.1.1; using 10.255.255.254 instead (on interface lo)
25/03/21 09:02:20 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/03/21 09:02:20 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
DataFrame Schema:
root
 |-- index: integer (nullable = true)
 |-- artist_name: string (nullable = true)
 |-- track_name: string (nullable = true)
 |-- track_id: string (nullable = true)
 |-- popularity: double (nullable = true)
 |-- year: integer (nullable = true)
 |-- genre: string (nullable = true)
 |-- danceability: double (nullable = true)
 |-- energy: double (nullable = true)
 |-- key: integer (nullable = true)
 |-- loudness: double (nullable = true)
 |-- mode: integer (nullable = true)
 |-- speechiness: double (nullable = true)
 |-- acousticness: double (nullable = true)
 |-- instrumentalness: double (nullable = true)
 |-- liveness: double (nullable = true)
 |-- valence: double (nullable = true)
 |-- tempo: double (nullable = true)
 |-- duration_ms: double (nullable = true)
 |-- time_signature: integer (nullable = true)
```

```
Total number of records: 1159764
Sample data:
25/03/21 09:02:30 WARN CSVHeaderChecker: CSV header does not conform to the schema.
 Header: , artist_name, track_name, track_id, popularity, year, genre, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalnes
s, liveness, valence, tempo, duration_ms, time_signature
 Schema: index, artist_name, track_name, track_id, popularity, year, genre, danceability, energy, key, loudness, mode, speechiness, acousticness, instrument
alness, liveness, valence, tempo, duration_ms, time_signature
Expected: index but found:
CSV file: file:///mnt/c/Users/Pranav/Downloads/spotify_data.csv/spotify_data.csv
+-----+-------------+--------------+--------------------+----------+----+--------+-----------+------+---+--------+----+-----------+------------+---------
-------+--------+-------+-------+-----------+--------------+
|index|  artist_name|    track_name|            track_id|popularity|year|   genre|danceability|energy|key|loudness|mode|speechiness|acousticness|instrumen
talness|liveness|valence|  tempo|duration_ms|time_signature|
+-----+-------------+--------------+--------------------+----------+----+--------+-----------+------+---+--------+----+-----------+------------+---------
-------+--------+-------+-------+-----------+--------------+
|    0|   Jason Mraz| I Won't Give Up|53QF56cjZA9RTuuMZ...|      68.0|2012|acoustic|       0.483| 0.303|  4| -10.058|   1|     0.0429|       0.694|
    0.0|   0.115|  0.139|133.406|   240166.0|             3|
|    1|   Jason Mraz|93 Million Miles|1s8tP3jP4GZcyHDsj...|      50.0|2012|acoustic|       0.572| 0.454|  3| -10.286|   1|     0.0258|       0.477|
1.37E-5|  0.0974|  0.515|140.182|   216387.0|             4|
|    2|Joshua Hyslop|Do Not Let Me Go|7BRCa8MPiyuvr2VU3...|      57.0|2012|acoustic|       0.409| 0.234|  3| -13.711|   1|     0.0323|       0.338|
 5.0E-5|  0.0895|  0.145|139.832|   158960.0|             4|
|    3| Boyce Avenue|     Fast Car|63wsZUhUZLlh1Osyr...|      58.0|2012|acoustic|       0.392| 0.251| 10|  -9.845|   1|     0.0363|       0.807|
    0.0|  0.0797|  0.508|204.961|   304293.0|             4|
|    4| Andrew Belle|Sky's Still Blue|6nXIYClvJAfi6ujLi...|      54.0|2012|acoustic|        0.43| 0.791|  6|  -5.419|   0|     0.0302|      0.0726|
 0.0193|    0.11|  0.217|171.864|   244320.0|             4|
+-----+-------------+--------------+--------------------+----------+----+--------+-----------+------+---+--------+----+-----------+------------+---------
-------+--------+-------+-------+-----------+--------------+
only showing top 5 rows
```

```
===== DATA CLEANING =====

1. Checking for missing values:
25/03/21 09:02:32 WARN CSVHeaderChecker: CSV header does not conform to the schema.
 Header: , artist_name, track_name, track_id, popularity, year, genre, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalnes
s, liveness, valence, tempo, duration_ms, time_signature
 Schema: index, artist_name, track_name, track_id, popularity, year, genre, danceability, energy, key, loudness, mode, speechiness, acousticness, instrument
alness, liveness, valence, tempo, duration_ms, time_signature
Expected: index but found:
CSV file: file:///mnt/c/Users/Pranav/Downloads/spotify_data.csv/spotify_data.csv
+-----+-----------+----------+--------+----------+----+-----+------------+------+---+--------+----+-----------+------------+----------------+--------+-----
--+-----+-----------+--------------+
|index|artist_name|track_name|track_id|popularity|year|genre|danceability|energy|key|loudness|mode|speechiness|acousticness|instrumentalness|liveness|valen
ce|tempo|duration_ms|time_signature|
+-----+-----------+----------+--------+----------+----+-----+------------+------+---+--------+----+-----------+------------+----------------+--------+-----
--+-----+-----------+--------------+
|    0|          0|         1|       0|      1673| 624|    0|        1184|   408|1564|      94|1298|         16|           8|               3|       2|
 0|    0|          0|       621|
+-----+-----------+----------+--------+----------+----+-----+------------+------+---+--------+----+-----------+------------+----------------+--------+-----
--+-----+-----------+--------------+
```

```
2. Filtering out invalid years:
Records with invalid years removed: 1673

3. Handling missing values in numerical columns:
25/03/21 09:02:38 WARN CSVHeaderChecker: CSV header does not conform to the schema.
 Header: , year
 Schema: index, year
Expected: index but found:
CSV file: file:///mnt/c/Users/Pranav/Downloads/spotify_data.csv/spotify_data.csv
Mean value for index: 658991.4676316455
Mean value for popularity: 18.393719491818864
Mean value for year: 2011.9550780247216
Mean value for danceability: 0.5377111884126548
Mean value for energy: 0.6403028873364046
Mean value for key: 5.288042131404182
Mean value for loudness: -8.966817867507823
Mean value for mode: 0.63450972332916 84
Mean value for speechiness: 0.09282859403967032
Mean value for acousticness: 0.32069253589465524
Mean value for instrumentalness: 0.25242039050744586
Mean value for liveness: 0.22299929568574 47
Mean value for valence: 0.455858327216134
Mean value for tempo: 121.4010958361649
Mean value for duration_ms: 249583.44692256482
Mean value for time_signature: 3.8861894272557165
```

```
4. Removing duplicate records:
Duplicate records removed: 0

5. Handling outliers:
Outlier treatment for popularity: mean=18.393719491818864, stddev=15.888212092912072
Outliers removed from popularity: 5565
Outlier treatment for danceability: mean=0.5371719646237909, stddev=0.18439627162797917
Outliers removed from danceability: 0
Outlier treatment for energy: mean=0.64030605174044, stddev=0.27043373713988444
Outliers removed from energy: 0
Outlier treatment for tempo: mean=121.39862583403752, stddev=29.77563998312893
Outliers removed from tempo: 1753
Outlier treatment for loudness: mean=-8.968764034262193, stddev=5.659655996920152
Outliers removed from loudness: 24443
```

Data Transformations Overview

Once the data is cleaned, transformations help in extracting meaningful insights. Common transformations include:

- Filtering specific records
- Grouping and aggregating data
- Sorting records
- Creating new derived columns

CODE:

```
# ----------------------------------------
# PART 2: DATA TRANSFORMATIONS
# ----------------------------------------
print("\n===== DATA TRANSFORMATIONS =====")

print("\n1. Dropping unnecessary columns:")
df = df.drop("index")
print("Dropped column: index")

print("\n2. Creating decade column:")
df = df.withColumn("decade", (col("year") / 10).cast("int") *
10)
print("Added column: decade")

print("\n3. Creating duration in minutes column:")
df = df.withColumn("duration_minutes", col("duration_ms") /
60000)
print("Added column: duration_minutes")

print("\n4. Filtering to recent songs:")
df_recent = df.filter(col("year") >= 2010)
print(f"Number of songs from 2010 onwards:
{df_recent.count()}")

print("\n5. Grouping by genre and calculating statistics:")
genre_stats = df.groupBy("genre") \
    .agg(
```

```
        count("*").alias("song_count"),
        avg("popularity").alias("avg_popularity"),
        avg("energy").alias("avg_energy"),
        avg("danceability").alias("avg_danceability"),
        sum("popularity").alias("total_popularity")
    ) \
    .orderBy(desc("song_count"))

print("Top 10 genres by song count:")
genre_stats.show(10)

print("\n6. Grouping by decade and genre:")
decade_genre = df.groupBy("decade", "genre") \
    .agg(count("*").alias("song_count")) \
    .orderBy("decade", desc("song_count"))

print("Top genres by decade (showing first 10 rows):")
decade_genre.show(10)

print("\n7. Calculating popularity statistics by year:")
year_stats = df.groupBy("year") \
    .agg(
        count("*").alias("song_count"),
        avg("popularity").alias("avg_popularity"),
        avg("energy").alias("avg_energy"),
        avg("danceability").alias("avg_danceability")
    ) \
    .orderBy("year")

print("Year statistics (showing first 10 rows):")
year_stats.show(10)

print("\n8. Grouping by artist and calculating statistics:")
artist_stats = df.groupBy("artist_name") \
    .agg(
        count("*").alias("song_count"),
        avg("popularity").alias("avg_popularity")
```

```
    ) \
    .filter(col("song_count") > 5) \
    .orderBy(desc("avg_popularity"))

print("Most popular artists (with more than 5 songs):")
artist_stats.show(10)

print("\n9. Creating categories based on audio features:")
df = df.withColumn(
    "energy_level",
    when(col("energy") > 0.8, "high")
    .when(col("energy") > 0.5, "medium")
    .otherwise("low")
)

df = df.withColumn(
    "mood",
    when(col("valence") > 0.7, "positive")
    .when(col("valence") > 0.3, "neutral")
    .otherwise("negative")
)

print("Added columns: energy_level, mood")
print("Sample with new categories:")
df.select("artist_name", "track_name", "energy",
"energy_level", "valence", "mood").show(5)

output_path =
"/mnt/c/Users/Pranav/Downloads/spotify_processed.parquet"
df.write.mode("overwrite").parquet(output_path)
print(f"\nProcessed data saved to {output_path}")

spark.stop()
```

OUTPUT:

```
===== DATA TRANSFORMATIONS =====

1. Dropping unnecessary columns:
Dropped column: index

2. Creating decade column:
Added column: decade

3. Creating duration in minutes column:
Added column: duration_minutes

4. Filtering to recent songs:
Number of songs from 2010 onwards: 693064

5. Grouping by genre and calculating statistics:
Top 10 genres by song count:
+-----------+----------+------------------+------------------+-------------------+----------------+
|      genre|song_count|    avg_popularity|        avg_energy|   avg_danceability|total_popularity|
+-----------+----------+------------------+------------------+-------------------+----------------+
|black-metal|     21802|11.088432253921658|0.83532745060544 97|0.26519238143289603|        241750.0|
|     gospel|     21598|18.908880451893694|0.5813364838873969| 0.5148130382442822|        408394.0|
|   acoustic|     20969| 17.83342076398493|0.4297145467261195| 0.5359004148981829|        373949.0|
|   alt-rock|     20739| 38.33183856502242|0.7399936419788804| 0.5019760644196923|        794964.0|
|        emo|     20731|23.437026675027735|0.7639660154985289|0.464100868264917 14|        485873.0|
|     indian|     20467| 8.691943127962086|0.5691582151756487| 0.5367975619289589|        177898.0|
|      k-pop|     19734| 27.17274754231276|0.6844805498125066| 0.6212595013681969|        536227.0|
|      blues|     19652|21.588744148178304|0.6456904996946875| 0.5222444382251168|        424262.0|
|      forro|     19358| 11.32405207149499|0.7929255088335572| 0.6538142886661849|        219211.0|
|     comedy|     19151| 10.03984126155292|0.6161011449532662| 0.5876291055297372|        192273.0|
+-----------+----------+------------------+------------------+-------------------+----------------+
only showing top 10 rows
```

```
6. Grouping by decade and genre:
Top genres by decade (showing first 10 rows):
+------+-----------+----------+
|decade|      genre|song_count|
+------+-----------+----------+
|  2000|     gospel|      8922|
|  2000|black-metal|      8735|
|  2000|        emo|      8558|
|  2000|   alt-rock|      8520|
|  2000|     indian|      8421|
|  2000|      k-pop|      8303|
|  2000|   acoustic|      8209|
|  2000|      forro|      8076|
|  2000|    spanish|      8036|
|  2000|      samba|      7902|
+------+-----------+----------+
only showing top 10 rows


7. Calculating popularity statistics by year:
Year statistics (showing first 10 rows):
+------+----------+-----------------+------------------+------------------+
|  year|song_count|    avg_popularity|        avg_energy|  avg_danceability|
+------+----------+-----------------+------------------+------------------+
|2000.0|     42760|10.770159027128157|0.6195634866206736|0.5351253999064548|
|2001.0|     40939|11.315762475878746|0.6178344907618651|0.5337011236229509|
|2002.0|     41039|11.712736665123419|0.6362818789444186|0.5384093179658371|
|2003.0|     41317|12.418641237263111|0.6391802547159763|0.5324414647723696|
|2004.0|     42414|12.267623897769605|0.6487885949521389|0.5353464681473099|
|2005.0|     42808|13.364931788450757|0.6424645062628482|0.5307661488506821|
|2006.0|     44614|13.092482180481463|0.6436103804343927|0.5370995292957366|
|2007.0|     44932| 13.46808510638298|0.6563583955510547|0.5391526373186145|
|2008.0|     46553|13.406504414323459|0.6599229460185169| 0.539185057891006|
|2009.0|     45890|14.359185007626934|0.6583417756221404|0.5407615558945305|
+------+----------+-----------------+------------------+------------------+
only showing top 10 rows
```

```
8. Grouping by artist and calculating statistics:
Most popular artists (with more than 5 songs):
+--------------+----------+------------------+
|   artist_name|song_count|    avg_popularity|
+--------------+----------+------------------+
|          LANY|         7| 64.42857142857143|
|        j-hope|        12|             64.25|
| Lewis Capaldi|         6|              63.0|
|         Ozuna|         9| 62.444444444444444|
|    Tate McRae|        10|              62.4|
|    Nio Garcia|         6|62.333333333333336|
|Racionais MC's|         8|             62.25|
|       Lovejoy|        11| 62.09090909090909|
|          Joji|        18| 61.944444444444444|
|         Lunay|         6|61.833333333333336|
+--------------+----------+------------------+
only showing top 10 rows


9. Creating categories based on audio features:
Added columns: energy_level, mood
Sample with new categories:
+--------------------+----------------+------+------------+-------+--------+
|         artist_name|      track_name|energy|energy_level|valence|    mood|
+--------------------+----------------+------+------------+-------+--------+
|The Righteous Bro...|         Justine| 0.887|        high|  0.698| neutral|
| Bombay Bicycle Club| It's Alright Now| 0.793|      medium|  0.597| neutral|
|          Uwe Kröger|You saved my life| 0.442|         low|  0.176|negative|
|               Pilot|             Рок| 0.952|        high|  0.675| neutral|
| Gyedu-Blay Ambolley|Little Small Girl| 0.914|        high|  0.937|positive|
+--------------------+----------------+------+------------+-------+--------+
only showing top 5 rows
```

CONCLUSION:

I successfully processed a large spotify dataset using PySpark by loading, cleaning, and transforming the data. Key steps included removing missing values, eliminating duplicates, handling outliers, filtering records, and grouping data by song and artist type. This workflow ensured efficient data handling and meaningful insights, demonstrating PySpark's scalability for large datasets.

Experiment no. 7	
AIM :	Pivot and Unpivot of DataFrame in Spark SQL
Theory	Apache Spark SQL is a module of Apache Spark that allows users to run SQL queries on structured data. One of its powerful features includes support for pivoting and unpivoting DataFrames — operations commonly used in data transformation and analytics to reshape datasets. **Pivot Operation:** Pivoting refers to rotating data from rows into columns. It is used to summarize data by grouping it and then spreading specific values across multiple columns based on unique categories. **Unpivot Operation:** Unpivoting (also called melting) is the reverse of pivoting. It transforms columns into rows, which is often needed to normalize wide data formats into long formats for easier analysis. **Why Use Pivot/Unpivot?** ● To reshape data for better visualization and reporting ● To perform aggregations across multiple dimensions ● To prepare data for machine learning or analytics workflows

Code:

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, isnan, when, count, mean, stddev, lit, expr, abs as sql_abs
from pyspark.sql.types import DoubleType, IntegerType
import os

# Create Spark session
spark = SparkSession.builder \
    .appName("Spotify Data Analysis") \
    .config("spark.driver.memory", "4g") \
    .getOrCreate()

# Define the path to your Windows file from WSL
# In WSL, Windows drives are mounted under /mnt/
windows_path = r"C:\Users\Pranav\Downloads\spotify_data.csv"
wsl_path = "/mnt/c/Users/Pranav/Downloads/spotify_data.csv"

print(f"Loading data from: {wsl_path}")
print("Checking if file exists:", os.path.exists(wsl_path))

# Load the dataset - specify that the first row is a header
df = spark.read.option("header", "true").csv(wsl_path, inferSchema=True)

# Print the schema and count to verify data loading
print("Original dataset schema:")
df.printSchema()
print(f"Total records: {df.count()}")

# Display first few rows
print("Sample data:")
df.show(5)
```

```
# Improved Pivot Operation
print("\n=== Improved Pivot Operation ===")

# First, check the distribution of mode values
print("Mode value distribution:")
df.groupBy("mode").count().orderBy("mode").show(20)

# Create a more manageable version of the mode column for pivoting
df = df.withColumn("mode_category",
    when(col("mode") == 0, "minor")
    .when(col("mode") == 1, "major")
    .otherwise("other")
)

# Now pivot using this categorized column
print("\nPivot: Average popularity by genre and mode category:")
pivoted_df = df.groupBy("genre").pivot("mode_category").agg(
    mean("popularity").alias("avg_popularity")
).orderBy("genre")
pivoted_df.show(10)

# Alternative approach: use SQL crosstab
print("\n=== CrossTab Alternative ===")

# Register the DataFrame as a temp view for SQL
df.createOrReplaceTempView("spotify_data")

# CrossTab query
crosstab_result = spark.sql("""
    SELECT genre,
           AVG(CASE WHEN mode_category = 'minor' THEN popularity END) as minor_popularity,
           AVG(CASE WHEN mode_category = 'major' THEN popularity END) as major_popularity,
           AVG(CASE WHEN mode_category = 'other' THEN popularity END) as other_popularity
    FROM spotify_data
    GROUP BY genre
    ORDER BY genre
```

```
""")

print("CrossTab result (alternative to pivot):")
crosstab_result.show(10)
```

```python
# Improved Unpivot Operation
print("\n=== Improved Unpivot Operation ===")

# Create a simplified DataFrame with just a few columns for clarity
simple_df = df.select("track_name", "artist_name", "genre", "popularity", "danceability", "energy")
simple_df.show(5)

# Unpivot using stack with explicit casting
unpivoted_df = simple_df.selectExpr(
    "track_name", 
    "artist_name",
    "genre",
    "stack(3, 'popularity', cast(popularity as double), 'danceability', cast(danceability as double), 'energy', cast(energy as double)) as (metric, value)"
)

print("Unpivoted metrics (popularity, danceability, energy):")
unpivoted_df.show(15)

# Save the processed data to parquet for future use
output_path = "/tmp/spotify_processed"
df.write.mode("overwrite").parquet(output_path)
print(f"\nProcessed data saved to {output_path}")

# Also save the pivot results
pivoted_path = "/tmp/spotify_pivoted"
pivoted_df.write.mode("overwrite").parquet(pivoted_path)
print(f"Pivoted data saved to {pivoted_path}")

# Save the unpivoted data
unpivoted_path = "/tmp/spotify_unpivoted"
unpivoted_df.write.mode("overwrite").parquet(unpivoted_path)
print(f"Unpivoted data saved to {unpivoted_path}")

# Clean up
spark.stop()
```

OUTPUT:

```
Loading data from: /mnt/c/Users/Pranav/Downloads/spotify_data.csv
Checking if file exists: True
Original dataset schema:
root
 |-- _c0: integer (nullable = true)
 |-- artist_name: string (nullable = true)
 |-- track_name: string (nullable = true)
 |-- track_id: string (nullable = true)
 |-- popularity: string (nullable = true)
 |-- year: string (nullable = true)
 |-- genre: string (nullable = true)
 |-- danceability: string (nullable = true)
 |-- energy: string (nullable = true)
 |-- key: string (nullable = true)
 |-- loudness: string (nullable = true)
 |-- mode: string (nullable = true)
 |-- speechiness: string (nullable = true)
 |-- acousticness: string (nullable = true)
 |-- instrumentalness: string (nullable = true)
 |-- liveness: string (nullable = true)
 |-- valence: double (nullable = true)
 |-- tempo: double (nullable = true)
 |-- duration_ms: double (nullable = true)
 |-- time_signature: double (nullable = true)

Total records: 1159764
Sample data:
25/04/10 17:58:46 WARN CSVHeaderChecker: CSV header does not conform to the schema.
 Header: , artist_name, track_name, track_id, popularity, year, genre, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration_ms, time_signature
 Schema: _c0, artist_name, track_name, track_id, popularity, year, genre, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration_ms, time_signature
Expected: _c0 but found: 
CSV file: file:///mnt/c/Users/Pranav/Downloads/spotify_data.csv/spotify_data.csv
+---+------------+---------------+--------------------+----------+----+--------+------------+------+---+--------+----+-----------+------------+----------------+--------+-------+-------+-----------+--------------+
|_c0| artist_name|     track_name|            track_id|popularity|year|   genre|danceability|energy|key|loudness|mode|speechiness|acousticness|instrumentalness|liveness|valence|  tempo|duration_ms|time_signature|
+---+------------+---------------+--------------------+----------+----+--------+------------+------+---+--------+----+-----------+------------+----------------+--------+-------+-------+-----------+--------------+
|  0|  Jason Mraz| I Won't Give Up|53QF56cjZA9RTuuMZ...|        68|2012|acoustic|       0.483| 0.303|  4| -10.058|   1|     0.0429|       0.694|             0.0|   0.115|  0.139|133.406|   240166.0|           3.0|
|  1|  Jason Mraz|93 Million Miles|1s8tP3jP4GZcyHDsj...|        50|2012|acoustic|       0.572| 0.454|  3| -10.286|   1|     0.0258|       0.477|        1.37e-05|  0.0974|  0.515|140.182|   216387.0|           4.0|
|  2|Joshua Hyslop|Do Not Let Me Go|7BRCa8MPiyuvr2VU3...|        57|2012|acoustic|       0.409| 0.234|  3| -13.711|   1|     0.0323|       0.338|           5e-05|  0.0895|  0.145|139.832|   158960.0|           4.0|
|  3| Boyce Avenue|        Fast Car|63wsZUhUZLlh1OsyrZ...|        58|2012|acoustic|       0.392| 0.251| 10|  -9.845|   1|     0.0363|       0.807|             0.0|  0.0797|  0.508|204.961|   304293.0|           4.0|
|  4|Andrew Belle|Sky's Still Blue|6nXIYClvJAfi6ujLi...|        54|2012|acoustic|        0.43| 0.791|  6|  -5.419|   0|     0.0302|      0.0726|          0.0193|    0.11|  0.217|171.864|   244320.0|           4.0|
+---+------------+---------------+--------------------+----------+----+--------+------------+------+---+--------+----+-----------+------------+----------------+--------+-------+-------+-----------+--------------+
only showing top 5 rows
```

```
=== Improved Pivot Operation ===
Mode value distribution:
+-------------------+-----+
|               mode|count|
+-------------------+-----+
|-20.439290360443675|  397|
|            -20.439|    1|
|            -20.428|    1|
|            -20.406|    1|
|            -20.397|    1|
|            -20.357|    1|
|            -20.356|    1|
|            -20.335|    1|
|            -20.331|    1|
|             -20.33|    1|
|            -20.329|    1|
|            -20.322|    1|
|            -20.314|    1|
|            -20.306|    1|
|            -20.301|    1|
|            -20.285|    1|
|            -20.271|    1|
|            -20.263|    1|
|            -20.233|    1|
|            -20.225|    1|
+-------------------+-----+
only showing top 20 rows


Pivot: Average popularity by genre and mode category:
+--------------------+-----+-----+------------------+
|               genre|major|minor|             other|
+--------------------+-----+-----+------------------+
|           ""Aida"""| NULL| NULL|18.393719491818864|
| ""Hymne an die J...| NULL| NULL|18.393719491818864|
| ""Spring"": I. A...| NULL| NULL|18.393719491818864|
|                9/12| NULL| NULL|18.393719491818864|
|             Adriano| NULL| NULL|18.393719491818864|
|           Adriano)"| NULL| NULL|18.393719491818864|
|            Alamar)"| NULL| NULL|18.393719491818864|
|                 Ali| NULL| NULL|18.393719491818864|
|              Ali)"| NULL| NULL|18.393719491818864|
|              Alisa| NULL| NULL|18.393719491818864|
+--------------------+-----+-----+------------------+
only showing top 10 rows
```

```
=== CrossTab Alternative ===
CrossTab result (alternative to pivot):
+--------------------+----------------+----------------+------------------+
|               genre|minor_popularity|major_popularity|  other_popularity|
+--------------------+----------------+----------------+------------------+
|           ""Aida"""|            NULL|            NULL|18.393719491818864|
| ""Hymne an die J...|            NULL|            NULL|18.393719491818864|
| ""Spring"": I. A...|            NULL|            NULL|18.393719491818864|
|                9/12|            NULL|            NULL|18.393719491818864|
|             Adriano|            NULL|            NULL|18.393719491818864|
|          Adriano)"|            NULL|            NULL|18.393719491818864|
|           Alamar)"|            NULL|            NULL|18.393719491818864|
|                 Ali|            NULL|            NULL|18.393719491818864|
|              Ali)"|            NULL|            NULL|18.393719491818864|
|              Alisa|            NULL|            NULL|18.393719491818864|
+--------------------+----------------+----------------+------------------+
only showing top 10 rows
```

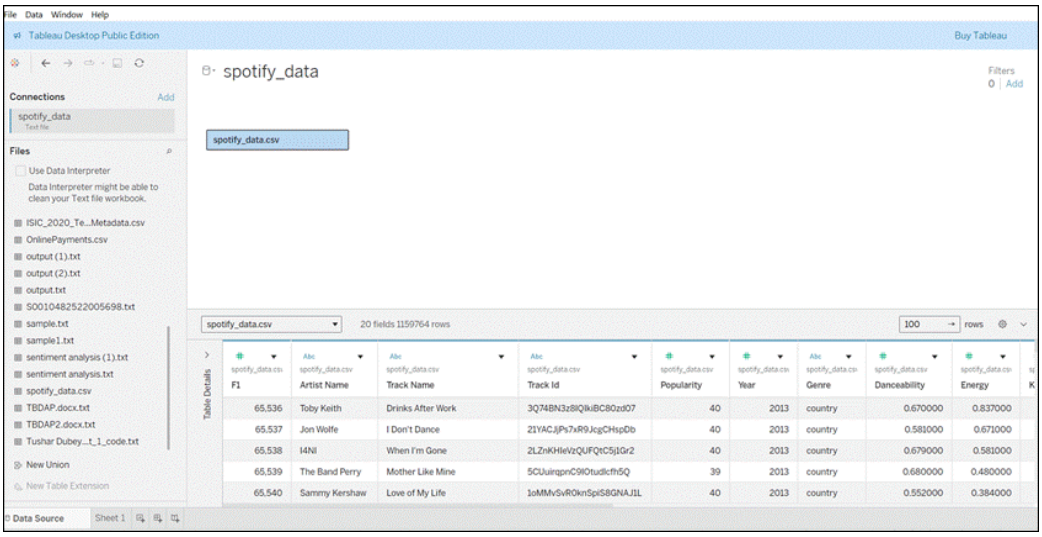
```
=== Improved Unpivot Operation ===
+----------------+-------------+-------+----------------+------------+------+
|      track_name|  artist_name|  genre|      popularity|danceability|energy|
+----------------+-------------+-------+----------------+------------+------+
| I Won't Give Up|   Jason Mraz|acoustic|66.0239643840082|       0.483| 0.303|
|93 Million Miles|   Jason Mraz|acoustic|            50.0|       0.572| 0.454|
|Do Not Let Me Go|Joshua Hyslop|acoustic|            57.0|       0.409| 0.234|
|        Fast Car| Boyce Avenue|acoustic|            58.0|       0.392| 0.251|
|Sky's Still Blue| Andrew Belle|acoustic|            54.0|        0.43| 0.791|
+----------------+-------------+-------+----------------+------------+------+
only showing top 5 rows

Unpivoted metrics (popularity, danceability, energy):
+----------------+-------------+-------+----------+----------------+
|      track_name|  artist_name|  genre|    metric|           value|
+----------------+-------------+-------+----------+----------------+
| I Won't Give Up|   Jason Mraz|acoustic|popularity|66.0239643840082|
| I Won't Give Up|   Jason Mraz|acoustic|danceability|           0.483|
| I Won't Give Up|   Jason Mraz|acoustic|    energy|           0.303|
|93 Million Miles|   Jason Mraz|acoustic|popularity|            50.0|
|93 Million Miles|   Jason Mraz|acoustic|danceability|           0.572|
|93 Million Miles|   Jason Mraz|acoustic|    energy|           0.454|
|Do Not Let Me Go|Joshua Hyslop|acoustic|popularity|            57.0|
|Do Not Let Me Go|Joshua Hyslop|acoustic|danceability|           0.409|
|Do Not Let Me Go|Joshua Hyslop|acoustic|    energy|           0.234|
|        Fast Car| Boyce Avenue|acoustic|popularity|            58.0|
|        Fast Car| Boyce Avenue|acoustic|danceability|           0.392|
|        Fast Car| Boyce Avenue|acoustic|    energy|           0.251|
|Sky's Still Blue| Andrew Belle|acoustic|popularity|            54.0|
|Sky's Still Blue| Andrew Belle|acoustic|danceability|            0.43|
|Sky's Still Blue| Andrew Belle|acoustic|    energy|           0.791|
+----------------+-------------+-------+----------+----------------+
only showing top 15 rows
```

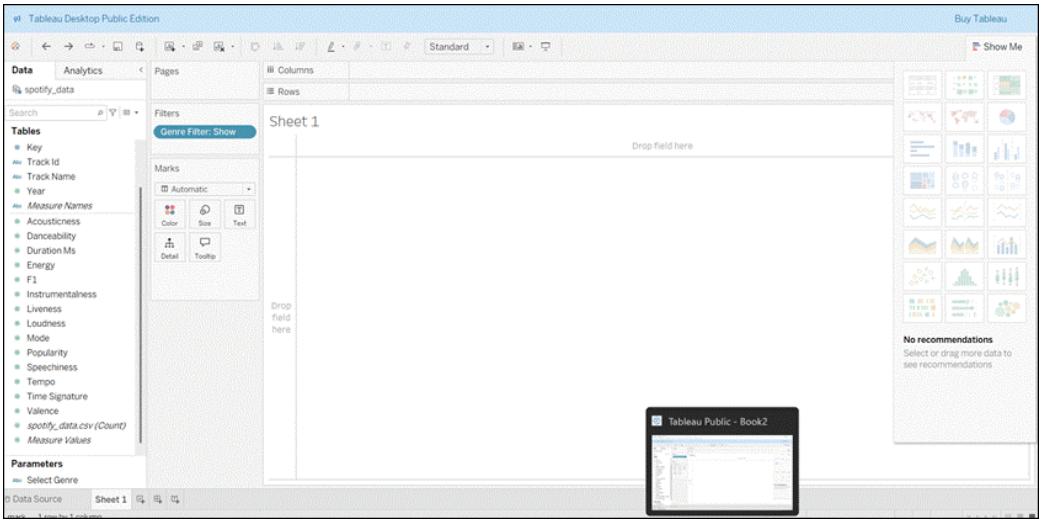
CONCLUSION: Pivoting and unpivoting are crucial data transformation techniques for reshaping data in Apache Spark. Pivoting helps convert row-based data into columnar format for easier analysis, while unpivoting reverses this process, making data suitable for aggregation and reporting. These operations enhance data processing efficiency in large-scale distributed environments.

Experiment no. 8	
AIM :	Visual Analytics in Tableau: Connection with multiple tables, Create data Extracts. Create a Report based on passed parameter
Theory	Tableau is a powerful data visualization and business intelligence tool that allows users to connect, analyze, and visualize data interactively. It enables users to build dashboards and reports with real-time updates and interactive features, making data-driven decision-making easier and faster. --- **1. Connection with Multiple Tables:** **In real-world analytics, data often exists in multiple related tables. Tableau allows users to:** ● Connect to multiple tables from databases, Excel sheets, or other sources ● Use Joins (Inner, Left, Right, Outer) or Relationships to combine data ● Automatically detect key fields for joining or allow manual linking This feature helps in integrating complex data schemas to create a unified dataset for analysis. --- **2. Creating Data Extracts:** **A Data Extract in Tableau is a snapshot of data optimized for performance. It allows:** ● Faster querying compared to live connections

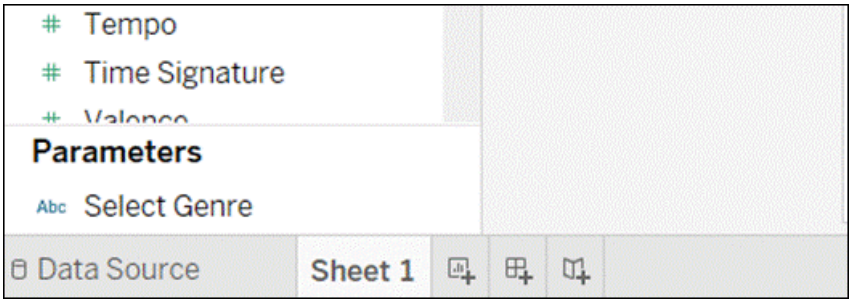
	• Offline access to data • Data filtering and aggregation at extract creation time Extracts improve performance, especially when working with large datasets or slow data sources. --- **3. Parameter-Based Reporting:** **Parameters in Tableau are dynamic values that users can control to change visualizations. They are used for:** • Filtering data • Changing measures or dimensions dynamically • Controlling calculations or reference lines Using parameters, users can generate interactive reports where the displayed data updates based on a selected or entered value.
Tableau Process:	**1.] Load Dataset on canvas**



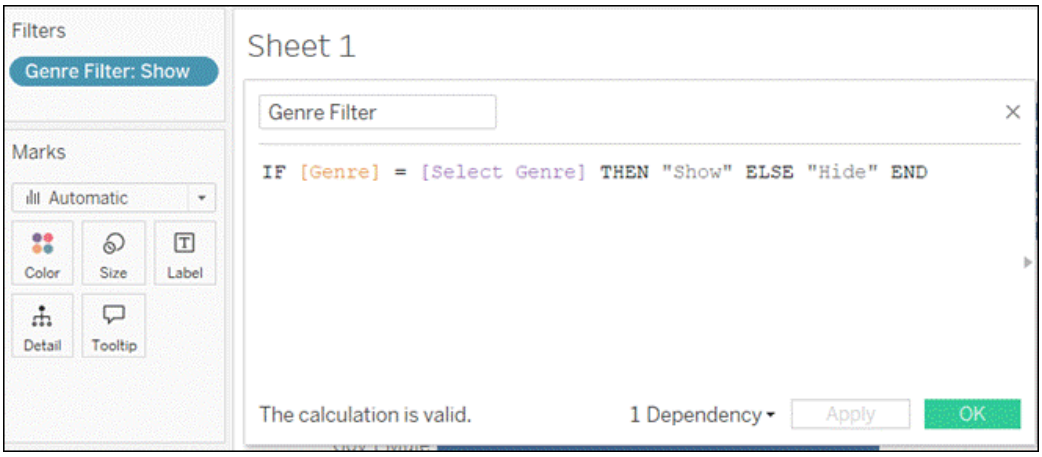
2.] Creating sheets -



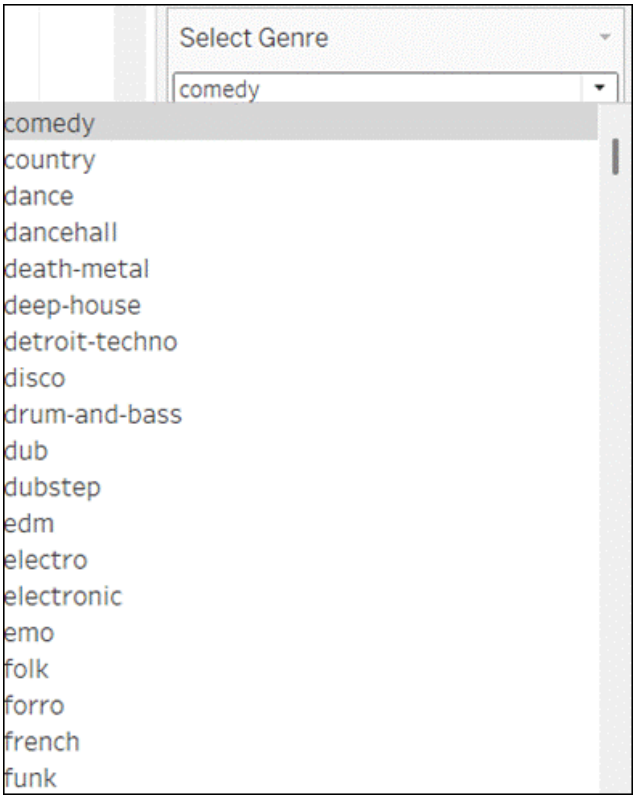
3.] Create a Parameter -

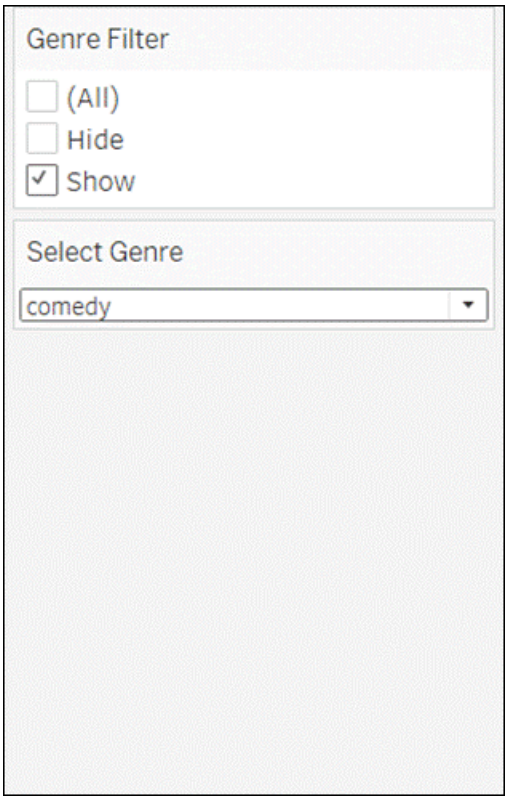


4.] Create a filter using the parameter -

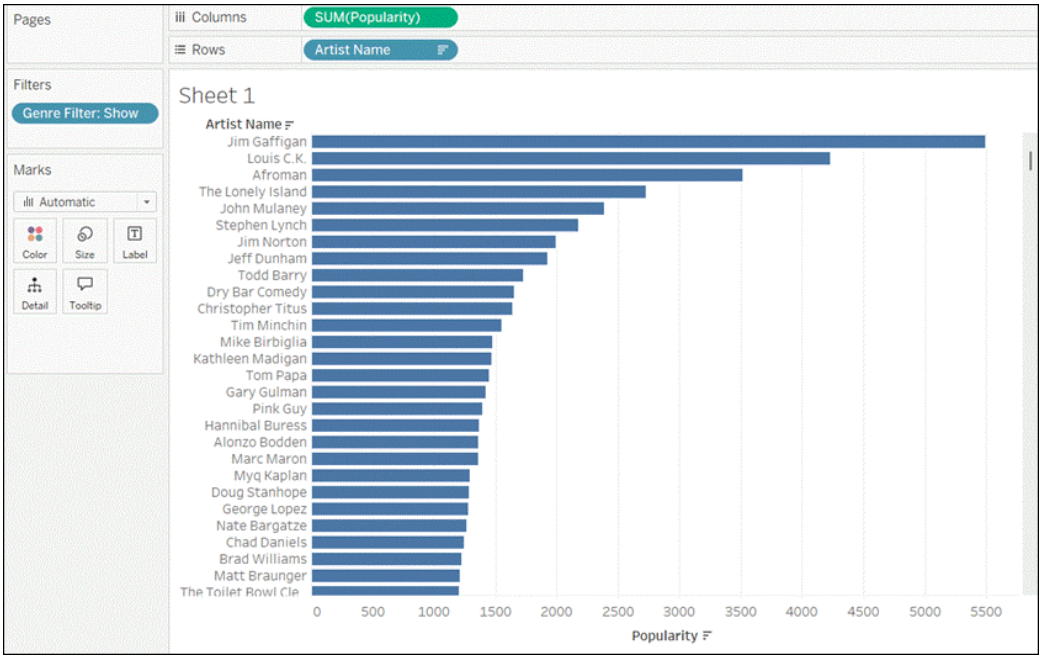


5.] Selecting genre from the dropdown -




Genre Filter
(All)
Hide
Show
Select Genre
comedy

6.] Report on Artist and Popularity (without parameter) -



- **Bar Chart of Artist Popularity**

1. **Overall Structure: The horizontal bars represent the popularity of different artists.**

2. **Horizontal Axis: The horizontal axis represents the SUM(Popularity), indicating a measure of each artist's overall popularity.**

3. **Vertical Axis: The vertical axis lists different Artist Names. Each bar corresponds to a specific artist.**

4. **Horizontal Bars: Each bar extends horizontally from zero to a value corresponding to the total popularity of that artist. Longer bars indicate higher popularity.**

5. **Ordering: The artists are ordered vertically based on their popularity, likely in descending order, with the most popular artists at the top.**

6. **Filter: A "Genre Filter" is present but currently set to "Show" with no specific genre selected, indicating that the popularity is shown across all genres.**

7.] Report on Artist and Popularity (genre filter: comedy) -

- **Area Chart of Artist Popularity**

1. **Overall Structure: The filled area under the line represents the cumulative popularity of artists.**

2. **Horizontal Axis: The horizontal axis represents the SUM(Popularity), indicating a measure of each artist's popularity.**

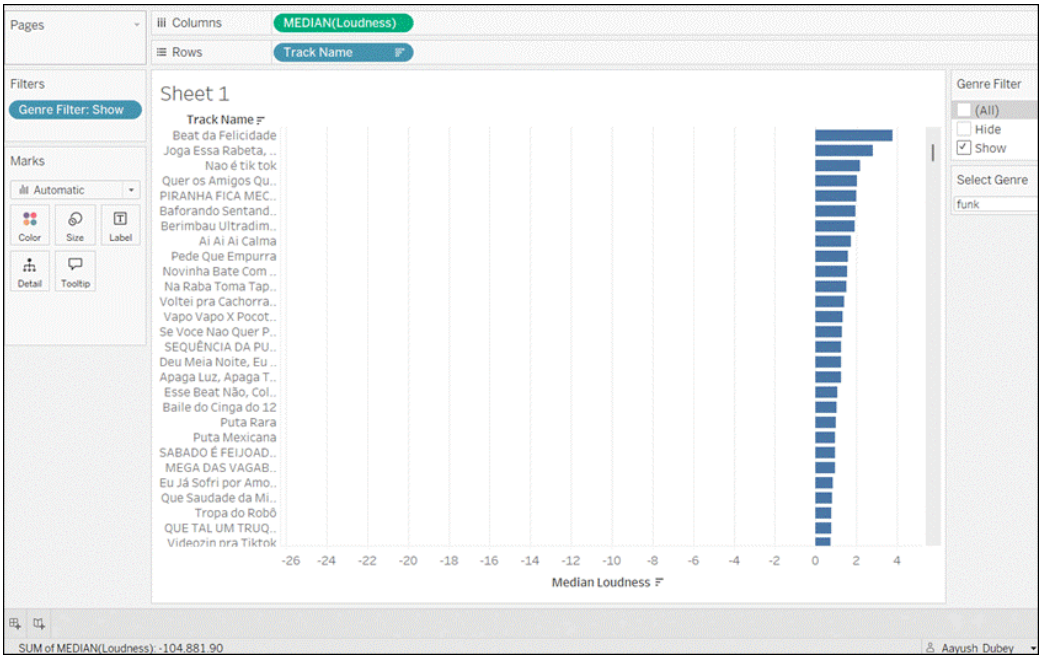
3. **Vertical Axis: The vertical axis lists different Artist Names. The chart displays the popularity extending from the left for each artist.**

4. **Filled Area: The colored area extends to the right for each artist, with the width of this area corresponding to their SUM(Popularity) value. Artists with a wider filled area have higher popularity.**

5. **Ordering: The artists are ordered vertically based on their popularity, likely in ascending or descending order of their SUM(Popularity).**

6. **Filter: A "Genre Filter" is applied, currently set to "comedy," so only comedy artists are displayed.**

8.] Report on Track Name (genre filter: funk) and median loudness in dB -



- **Bar Chart of Track Median Loudness (dB) for Funk Genre**

1. **Overall Structure: The horizontal bars represent the median loudness of different music tracks within the "funk" genre.**

2. **Horizontal Axis: The horizontal axis represents the MEDIAN(Loudness) measured in dB (decibels). The values extend from negative to positive, indicating the loudness level.**

3. **Vertical Axis: The vertical axis lists different Track Names. Each bar corresponds to a specific track.**

4. **Horizontal Bars: Each bar extends horizontally from zero to a**

value corresponding to the median loudness of that track. Longer bars indicate a higher (louder) median loudness.

5. **Ordering: The tracks are ordered vertically based on their median loudness, likely in descending order, with the loudest tracks at the top.**

6. **Filter: A "Genre Filter" is applied, currently set to "funk," so only tracks categorized under the funk genre are displayed.**

9.] Report on popularity (based on the number of streams) of different music tracks.

- **Treemap of Track Popularity by Streams**

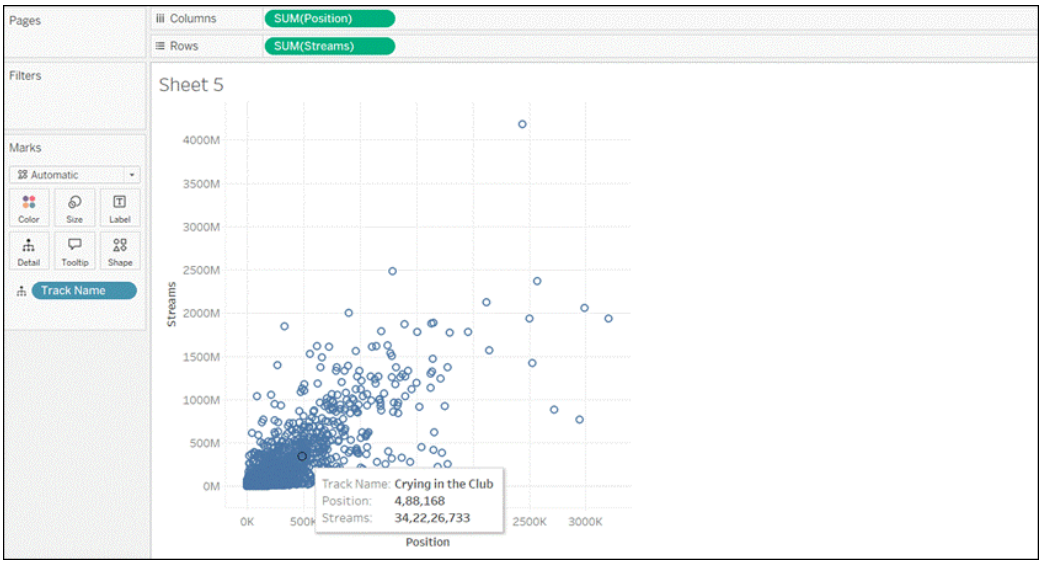
1. **The Overall Structure: The entire rectangular area represents a total value, likely the total number of streams.**

2. **Individual Rectangles (Tiles): Each smaller rectangle within the larger one represents a specific track name. This is indicated by the color legend on the right, where each color corresponds to a different song title.**

3. **Size of the Rectangles: The size of each rectangle is proportional to the SUM(Streams) for that particular track. Larger rectangles indicate tracks with a higher number of streams, while smaller rectangles represent tracks with fewer streams.**

4. **Color: The color of each rectangle visually distinguishes the different track names, making it easier to identify individual songs within the treemap.**

10.] Report on music track's position (ranking) and its streaming popularity



- **Scatter Plot of Position vs. Streams**

1. **Overall Structure: The entire chart area displays the relationship between two continuous variables: chart position and streams.**

2. **Individual Circles (Points): Each circle represents a unique music track.**

3. **Position on the Horizontal Axis: The horizontal placement of each circle corresponds to the SUM(Position) of that track. Tracks further to the right generally have a lower (worse) chart position.**

4. **Position on the Vertical Axis: The vertical placement of each circle corresponds to the SUM(Streams) of that track. Tracks higher up have a greater number of streams.**

11.] Report on total streams (popularity) of different artists across various countries

- **Heatmap of Artist Streams by Country**

1. **Overall Structure: The entire grid represents the total streams of various artists across different countries.**

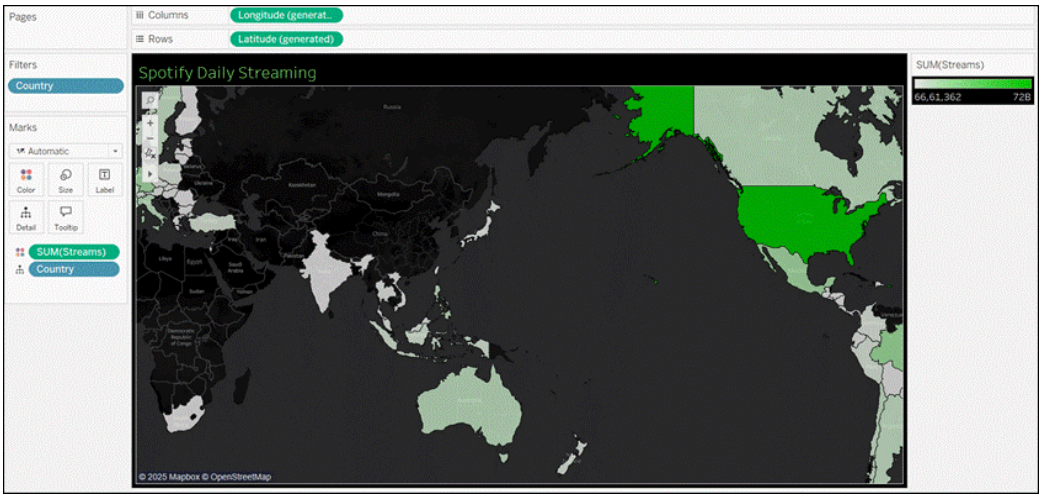
2. **Individual Cells (Tiles): Each cell in the grid represents the total streams for a specific artist within a particular country.**

3. **Rows: Each row corresponds to a different artist, as listed on the left.**

4. **Columns: Each column corresponds to a different country, as listed at the top.**

5. **Color Intensity: The color intensity of each cell is proportional to the SUM(Streams) for that artist in that country. Darker shades indicate a higher number of streams, while lighter shades indicate fewer streams.**

12.] Report on Spotify Daily Streaming by Country



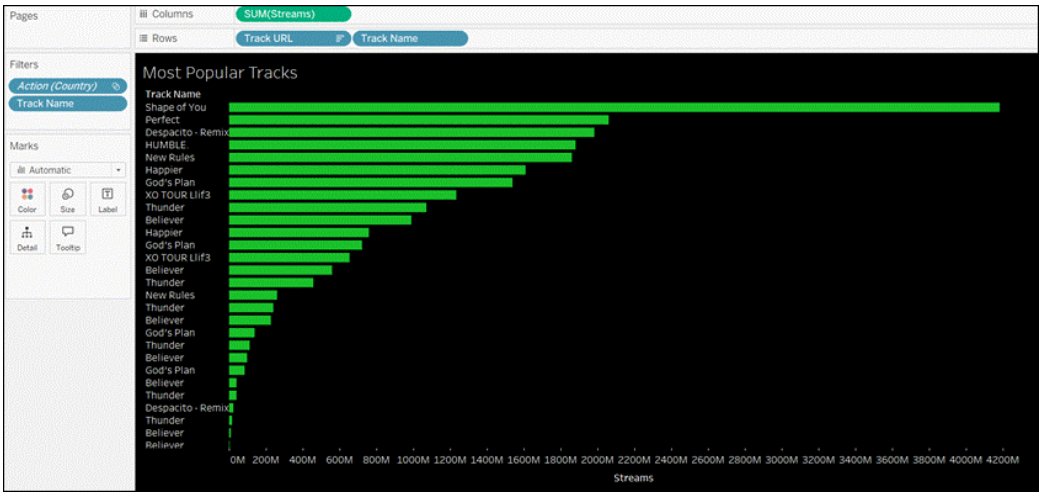
- **Map Visualization of Spotify Daily Streaming by Country**

1. **Overall Structure: The world map displays the geographical distribution of Spotify daily streaming.**

2. **Map Areas: Each country on the map is colored based on its SUM(Streams) value.**

3. **Color Intensity: The intensity of the green color corresponds to the total number of streams within that country. Brighter green indicates a higher volume of daily streams, while darker shades indicate lower streaming activity. Countries with no color likely**

	have no data or very low streaming numbers. **4. Horizontal Axis (Implicit): Longitude (generated) determines the horizontal placement of countries on the map.** **5. Vertical Axis (Implicit): Latitude (generated) determines the vertical placement of countries on the map.** **6. Title: The title "Spotify Daily Streaming" clearly indicates what the map represents.** **7. Filters: A "Country" filter is present, allowing the user to focus on specific countries if needed.** **8. Color Legend: The legend on the top right shows the range of SUM(Streams) values and their corresponding color intensities.** **13.] Report on Most Popular Tracks by Streams**



- **Bar Chart of Most Popular Tracks by Streams**

1. **Overall Structure: The horizontal bars represent the total number of streams for different music tracks.**

2. **Horizontal Axis: The horizontal axis represents the SUM(Streams), indicating the total stream count for each track. The scale is in millions.**

3. **Vertical Axis: The vertical axis lists different Track Names. Each bar corresponds to a specific track.**

4. **Horizontal Bars: Each bar extends horizontally from zero to a value corresponding to the total streams of that track. Longer bars indicate a higher number of streams and thus greater popularity.**

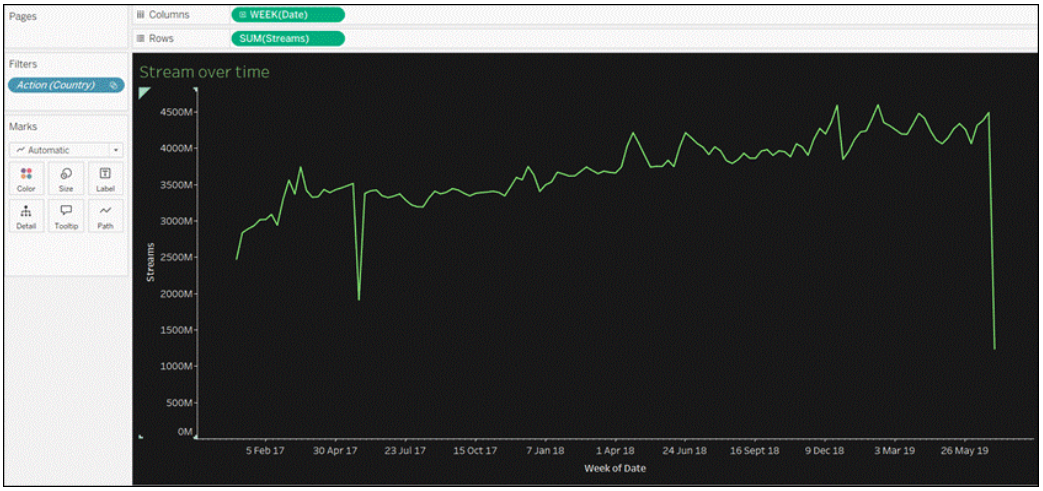
5. **Ordering: The tracks are ordered vertically based on their total**

streams, likely in descending order, with the most streamed tracks at the top.

6. **Title: The title "Most Popular Tracks" clearly indicates the focus of the visualization.**

7. **Filters: "Action (Country)" and "Track Name" filters are present, suggesting the data can be filtered by specific countries or individual track names.**

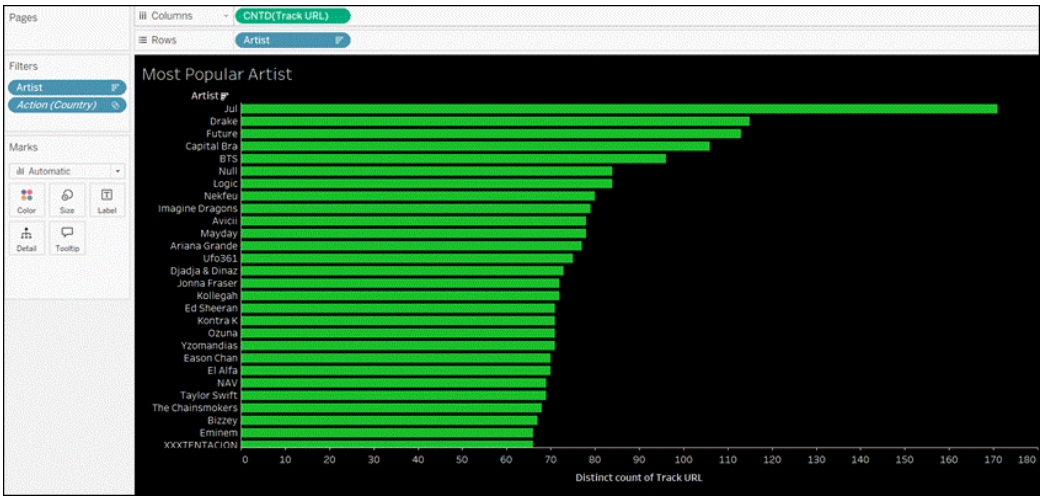
14.] Report on Streams Over Time



- **Line Chart of Streams Over Time**

1. **Overall Structure: The line connects data points showing the trend of total streams over a period of time.**

	2. **Horizontal Axis: The horizontal axis represents the WEEK(Date), showing the progression of weeks over the displayed timeframe.** 3. **Vertical Axis: The vertical axis represents the SUM(Streams), indicating the total number of streams in each corresponding week.** 4. **Line: The green line connects the data points, illustrating how the total number of streams has changed from week to week. Peaks indicate periods with higher stream counts, while troughs indicate periods with lower stream counts.** 5. **Title: The chart title "Stream over time" clearly describes the visualization's purpose.** 6. **Filter: An "Action (Country)" filter is present, suggesting that the stream data can be filtered by specific countries.** **15.] Report on Most Popular Artist by Track URL**



- **Bar Chart of Most Popular Artists by Distinct Track URL Count**

1. **Overall Structure: The horizontal bars represent the popularity of different artists based on the distinct count of their track URLs.**

2. **Horizontal Axis: The horizontal axis represents the CNTD(Track URL), indicating the distinct number of unique track URLs associated with each artist. A higher count suggests a larger and more diverse catalog of tracks.**

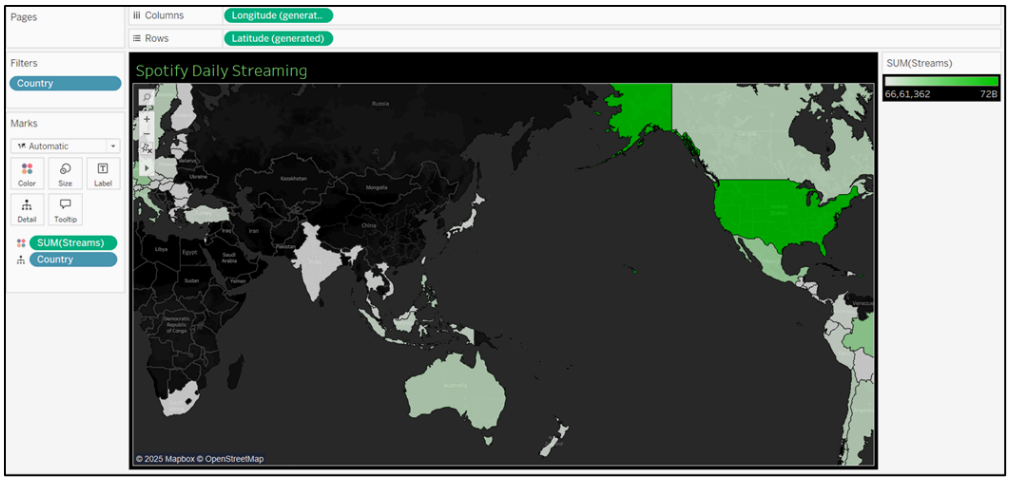
3. **Vertical Axis: The vertical axis lists different Artist names. Each bar corresponds to a specific artist.**

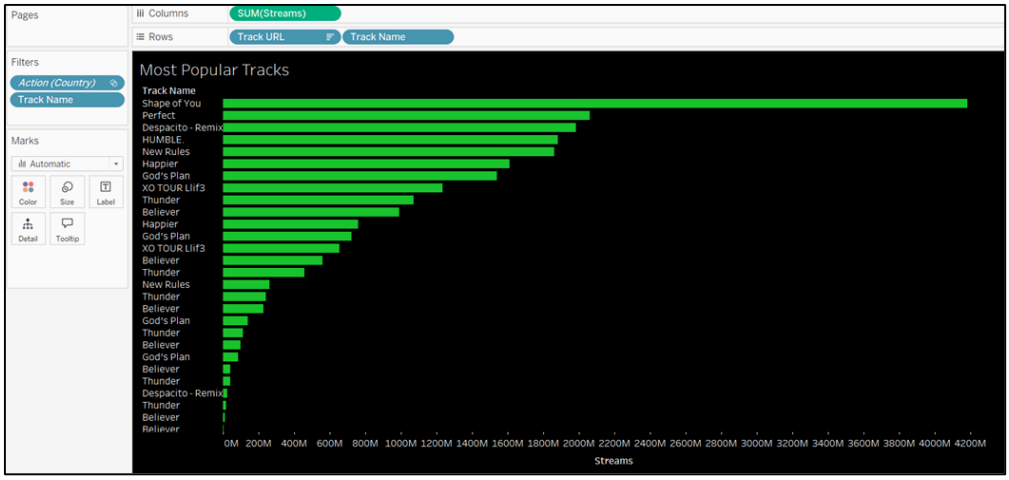
4. **Horizontal Bars: Each bar extends horizontally from zero to a value corresponding to the distinct count of track URLs for that artist. Longer bars indicate a larger number of unique tracks.**

5. **Ordering: The artists are ordered vertically based on their distinct track URL count, likely in descending order, with artists having**

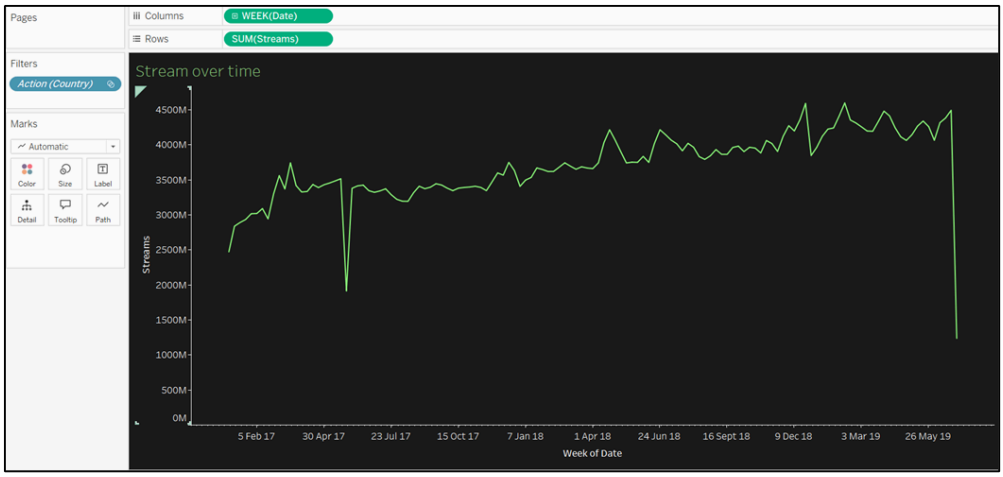
	the most unique tracks at the top. **6. Title: The title "Most Popular Artist" suggests that the number of unique tracks is being used as a measure of popularity or catalog size.** **7. Filters: "Artist" and "Action (Country)" filters are present, allowing the data to be filtered by specific artists or countries.**
CONCLUSION:	Successfully connected to and imported data from the Spotify dataset. Created data extracts optimized for publishing to Tableau Public. Implemented an interactive parameter enabling dynamic filtering of the report based on the selected genre. Developed a calculated field to apply parameter logic, ensuring visualizations display data relevant to the chosen genre, thus creating an interactive and user-driven report.

Experiment no. 9	
AIM :	Visual Analytics in Tableau : Sorting, Grouping, Filtering, Formatting Pane, Trend lines, reference lines. To Create a Dashboard for 1) Overall Revenue 2)State-wise Geo Map 3) Top 5 states 4) Bottom 5 states
Theory	Tableau is a leading data visualization and business intelligence tool that provides powerful visual analytics capabilities. This experiment explores the advanced visualization techniques in Tableau that help refine, structure, and enrich the data presentation for better decision-making. --- **Key Concepts:** **1. Sorting:** Sorting helps organize data in ascending or descending order. In dashboards, sorting is crucial for ranking metrics like revenue, sales, or profit, especially when highlighting top or bottom performers. **2. Grouping:** Grouping allows users to combine related categories into a single group, simplifying analysis. For example, multiple states can be grouped into regions for regional-level comparison. **3. Filtering:** Filters are used to include or exclude specific data based on conditions. Users can create dynamic dashboards where data views change based on user-selected filters (e.g., selecting a particular year or region). **4. Formatting Pane:** The formatting pane in Tableau enables customization of visual elements such as fonts, borders, shading, tooltips, and grid lines. Good formatting enhances readability and visual appeal of the dashboard.

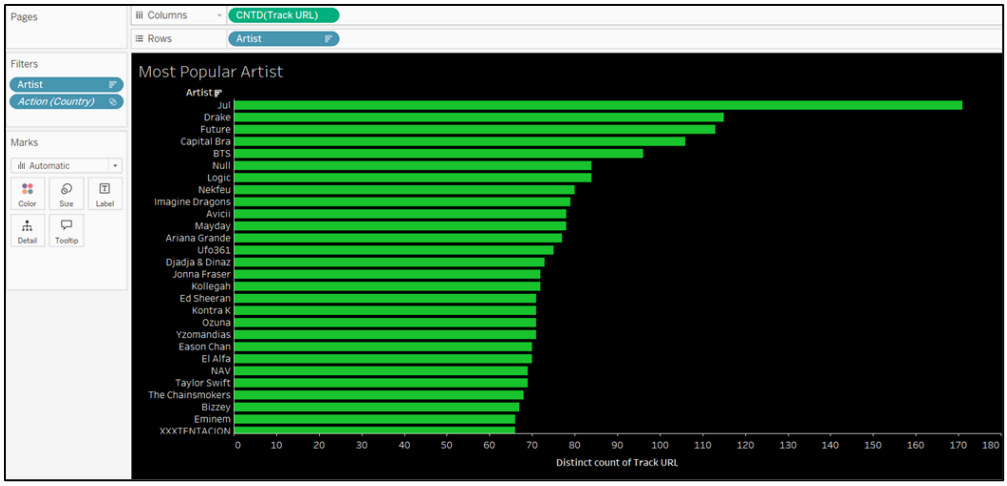
	5. Trend Lines: Trend lines reveal patterns or direction in data over time. Tableau supports linear, exponential, polynomial, and logarithmic trend lines to help forecast or understand movement in revenue or sales data. **6. Reference Lines:** Reference lines are static or dynamic indicators placed across charts to show averages, medians, targets, or benchmarks. They help compare actual performance against set goals.
Graph Used:	**1.] Report on Spotify Daily Streaming by Country**  **2.] Report on Most Popular Tracks by Streams**



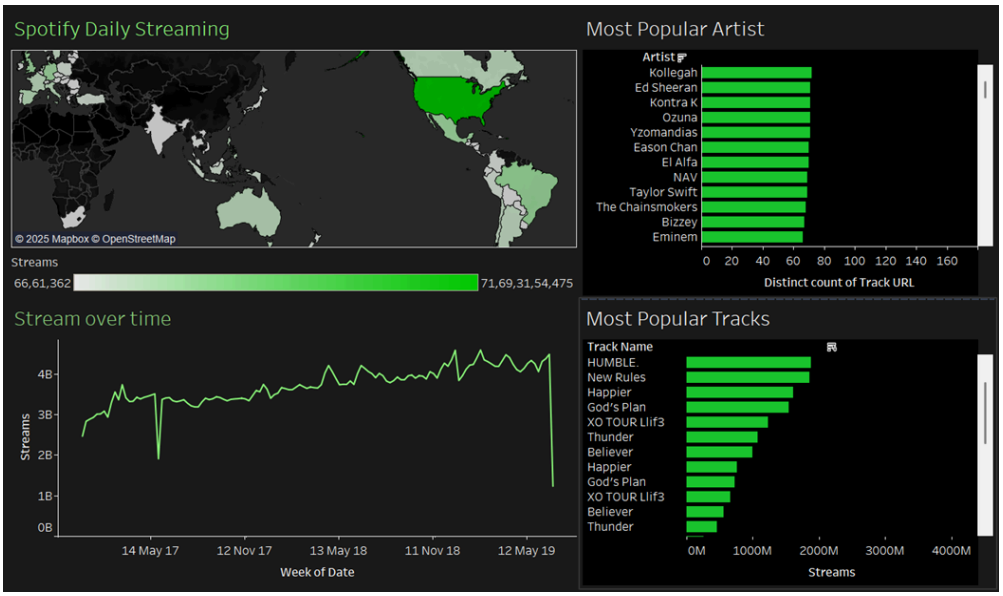
3.] Report on Streams Over Time



4.] Report on Most Popular Artists by Distinct Track URL Count



DASHBOARD:



Rationale for selecting these four specific visualizations:

1. **Spotify Daily Streaming (Map):** This provides a geographical context to music consumption. It allows you to see which countries have higher streaming activity overall. When a genre filter is applied, this map could highlight regions where that specific genre is particularly popular. This adds a spatial dimension to the analysis of popularity.

	2. **Most Popular Artist (Bar Chart):** This visualization identifies the top artists based on the distinct count of their track URLs (indicating a large and diverse catalog within the dataset). Filtering by genre would reveal which artists have the most extensive catalog within that specific genre present in the data. This focuses on the supply side of music within different genres**.** 3. **Stream over time (Line Chart):** This shows the temporal trend of overall streams. When a genre filter is applied, this chart would illustrate how the total streams for tracks within that genre have evolved over the selected time period. This helps in understanding the popularity trends of specific genres over time. 4. **Most Popular Tracks (Bar Chart):** This visualization directly addresses track-level popularity based on the total number of streams. Applying a genre filter would narrow this down to show the most streamed tracks belonging to the selected genre. This is a direct measure of which songs within a genre are most consumed.
CONCLUSION:	The interactive dashboard provides a multifaceted analysis of music popularity on Spotify by showcasing geographical streaming activity, prominent artist presence, temporal trends in stream volume, and the most popular tracks. This integrated view allows for a comprehensive understanding of music popularity beyond simple top charts.

Experiment no. 10	
AIM :	Explore and present interactive data insights from real world dataset (Dashboards) using POWER BI
Theory	Power BI is a business analytics and data visualization platform developed by Microsoft. It enables users to connect to various data sources, perform data cleaning and transformation, and create interactive visual reports and dashboards. Power BI is widely used in industry for its powerful data modeling, visualization capabilities, and ease of use. Using Power BI, data from real-world scenarios such as retail sales, customer feedback, finance, or web analytics can be explored to: ● Identify patterns and trends ● Monitor key performance indicators (KPIs) ● Make data-driven decisions --- **Key Features of Power BI for Dashboard Creation:** **Data Connection:** Power BI can connect to multiple data sources like Excel, SQL databases, web APIs, SharePoint, and cloud platforms. This flexibility supports comprehensive data integration. **Data Transformation (Power Query):** Using Power Query Editor, users can clean, filter, merge, and shape raw data into a meaningful structure suitable for analysis. **Visualizations:**

Power BI provides a variety of visualization tools such as:

- Bar and column charts
- Line and area charts
- Pie and donut charts
- Cards and KPIs
- Maps and tree maps
  These visuals help uncover trends and outliers in the data.

Slicers and Filters:

Slicers and filters enable users to interact with dashboards by selecting time periods, categories, or specific data points. This makes the report interactive and user-driven.

DAX (Data Analysis Expressions):

Power BI supports DAX functions for creating calculated fields and measures like total sales, average revenue, or year-over-year growth.

Dashboards:

A dashboard in Power BI is a single-page, real-time view of key data visualizations. It combines multiple reports and visuals to provide a high-level summary of business performance.

DASHBOARD:

Date & Filter Controls (Slicers)

- **Used for:** Time range selection, specific track and artist filtering, and year-wise toggles.
- **Insights Enabled:**
  - Users can drill down into performance across custom periods.
  - Track or artist-specific analysis becomes easy.
  - Example Use: Narrowing down streaming trends to songs released in 2022.

Card Visuals (KPI Tiles)

- **Used for:**
  - **Displaying summary stats like average yearly streams (e.g., 289.56M),**
  - **Comparative metrics (e.g., 767.9% above average),**
  - **Key track attributes (e.g., Acousticness, Danceability).**
- **Insights Enabled:**
  - **Offers at-a-glance performance indicators.**
  - **Highlights exceptional cases (e.g., viral songs).**

Bar Chart – Tracks by Streams

- **Used for:** Ranking top songs by number of streams.
- **Insights Enabled:**
  - Immediate visibility of which tracks dominate in terms of listener count.
  - Highlights how sharply stream volume drops after top performers.

Clustered Column Chart – Platform Comparison

- **Used for:** Comparing song performance across Spotify, Apple Music, and Deezer.
- **Insights Enabled:**
  - Shows which platform contributes most to a song's success.
  - Can reveal platform-specific preferences.
- **Example Insight:** A particular song may be heavily favored on Apple while underperforming on Deezer.

Gauge Chart – Energy %

- **Used for:** Displaying the energy level of the selected track as a percentage (e.g., 64%).
- **Insights Enabled:**
  - Gives a feel for a song's dynamic intensity.
  - Great for evaluating music mood or potential for workouts/dance.
- **Strength:** Visually impactful and intuitive.

Line Chart – Tracks by Release Date (BPM Trend)

- **Used for:** Showing track BPMs over time.
- **Insights Enabled:**

	○ Helps detect trends in song tempo across release months. ○ Peaks may correspond to energetic hit releases. **Image & Track Info Panel** ● **Used for:** Enhancing interactivity with dynamic updates based on filters. ● **Example:** Displaying the track "As It Was" with associated metadata and album art. **Music Feature Tiles** ● **Used for:** Acousticness, Speechiness, Valence, Liveness, and Danceability. ● **Insights Enabled:** ○ Supports deeper analysis of audio traits across different songs. ○ Helps correlate stream success with musical characteristics.
CONCLUSION:	This Spotify dashboard effectively combines slicers, KPI cards, bar charts, line graphs, and gauge visuals to deliver both high-level performance insights and detailed audio analysis. Its layout supports interactivity and scalability, making it well-suited for analyzing trends across multiple tracks, artists, and platforms in a visually engaging way.

Extra Question	
AIM :	To Create A Dashboard on Excel Sheet.
Theory	A dashboard in Excel is a visual representation of key data and metrics that allows users to monitor performance, track trends, and make informed decisions. Excel dashboards consolidate data into charts, tables, and summaries that are dynamically linked to underlying datasets. They provide a clear and concise overview of important information, often using interactive features like slicers, drop-down lists, and pivot tables. Excel dashboards are widely used in business environments for financial analysis, sales reporting, inventory tracking, and operational monitoring, thanks to Excel's powerful calculation, formatting, and visualization tools. --- **Key Components of an Excel Dashboard:** **1. Data Preparation:** Raw data is first cleaned and organized into structured tables. This may involve removing duplicates, handling missing values, and categorizing data. **2. Pivot Tables and Charts:** Pivot tables summarize data by categories and metrics, allowing dynamic grouping and aggregation. Pivot charts visually represent these summaries, providing insights into trends and comparisons. **3. Interactive Controls:** ● Slicers and Timeline filters allow users to filter data instantly. ● Drop-down lists (Data Validation) enable user-driven selections. ● Form controls like scroll bars and buttons may also be used to

enhance interactivity.

4. KPI Indicators:

Dashboards often include Key Performance Indicators (KPIs) such as total revenue, profit margin, or customer growth, highlighted using data bars, conditional formatting, or icons.

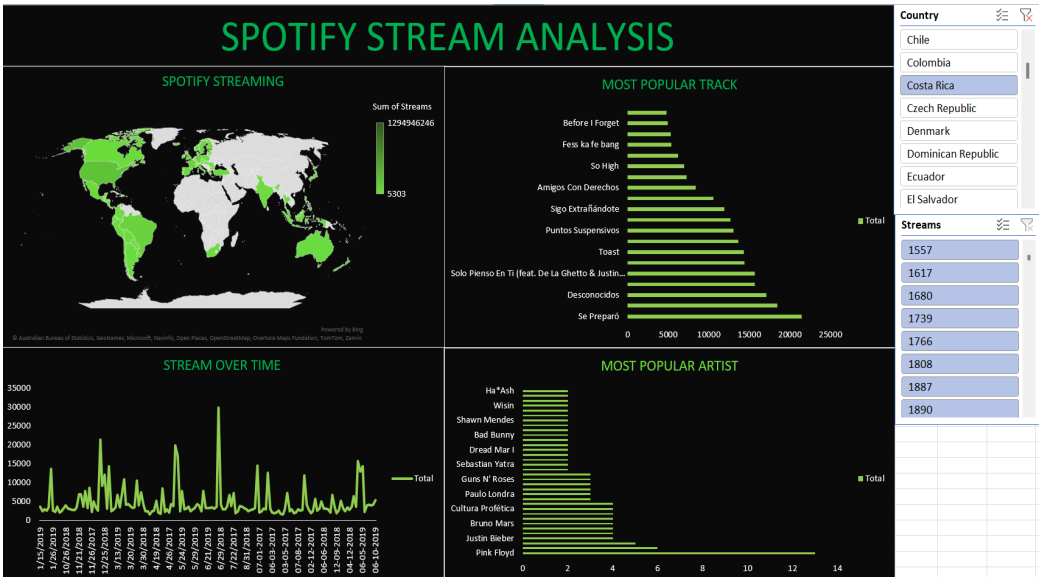
5. Formatting and Layout:

Proper use of cell formatting, colors, and grid layouts enhances readability and usability. Grouping visuals logically and minimizing clutter are essential for effective dashboards.

Advantages of Excel Dashboards:

- No additional software required—widely available and easy to use
- Highly customizable and flexible
- Ideal for small to medium datasets
- Interactive and real-time updates based on user inputs

Code:



Map Chart – Spotify Streaming by Country

- **Used for:** Visualizing total number of streams across different countries.
- **Insights Enabled:**
  - Highlights geographic distribution of Spotify usage.
  - Identifies high-engagement regions.

Example Insight: Latin American countries and parts of Europe show significantly higher streaming volumes, indicating strong Spotify penetration in these regions.

Bar Chart – Most Popular Track

- **Used for:** Ranking songs based on total number of streams.
- **Insights Enabled:**
  - Identifies top-performing songs globally or per country filter.

- Detects music preferences by comparing track popularity.

Example Insight: *"Se Preparó"* and *"Desconocidos"* are leading in total streams, suggesting their widespread popularity among listeners.

Line Chart – Stream Over Time

- **Used for:** Analyzing trends in streaming activity over time.
- **Insights Enabled:**
  - Reveals seasonal or campaign-driven peaks in listening behavior.
  - Helps in identifying dates of song or artist releases based on sudden spikes.

Example Insight: Several significant peaks in streaming occurred between early 2019 and mid-2019, possibly aligning with song releases or promotional events.

Bar Chart – Most Popular Artist

- **Used for:** Comparing artist popularity by stream count.
- **Insights Enabled:**
  - Identifies dominant music artists on the platform.
  - Highlights artist fanbase strength across selected regions.

Example Insight: *Pink Floyd* and *Justin Bieber* lead the chart, showcasing their sustained global popularity across diverse listener bases.

	--- **Slicer – Country Filter** ● **Used for:** Filtering all visualizations by selected country. ● **Insights Enabled:** ○ Enables localized insight generation. ○ Assists in comparative analysis between countries. **Example Insight:** Selecting "Costa Rica" refines the dashboard to reveal top songs and artists preferred by Costa Rican listeners specifically. --- **Slicer – Stream Volume Filter** ● **Used for:** Filtering data by specific stream count thresholds. ● **Insights Enabled:** ○ Focuses analysis on low or high stream values. ○ Helps exclude noise or outliers in data. **Example Insight:** Filtering to streams above 10,000 reveals only top-tier tracks and artists, useful for executive reporting.
CONCLUSION:	**Comparison Between Excel vs Tableau vs Power BI**

Feature / Tool	Excel	Tableau	Power BI
Ease of Use	Very easy for beginners	Moderate – requires some learning	Easy – especially for Microsoft Office users
Data Handling Capacity	Limited (small to medium datasets)	High (handles large datasets efficiently)	High (suitable for enterprise datasets)
Supported Data Sources	Excel, CSV, limited databases	Wide range – databases, cloud, APIs	Wide range – databases, Excel, cloud services
Visualization Options	Basic charts, pivot tables, conditional formatting	Advanced, interactive, beautiful visuals	Rich visuals with interactive and custom options
Interactivity	Basic (slicers, dropdowns)	High (filters, parameters, dashboard actions)	High (drill-downs, slicers, filters)
Real-Time Data Support	Manual refresh or via VBA	Yes – with live data connections	Yes – supports real-time dashboards
Data Modeling Features	Limited (Power Pivot, formulas)	Advanced (joins, relationships, calculated fields)	Very advanced (DAX, Power Query)
Learning Curve	Low	Medium to High	Low to Medium
Collaboration & Sharing	File sharing, OneDrive	Tableau Server/Public/Online	Power BI Service, Teams, SharePoint
Cost	Included with MS Office	Expensive – requires license	Free + Pro plans (affordable)
Best For	Simple reporting and small dashboards	Data exploration, storytelling, advanced visuals	Business dashboards, real-time BI, Microsoft ecosystem