EXP.1: Downloading and installing Hadoop, Understanding different Hadoop modes, Startup scripts, Configuration files.

- Install java 8, set path on both user and system variables.
- Download Hadoop-3.3.6 and modify the xml file configurations.
- Set path for Hadoop-bin and sbin folders in system variables.
- Open Command Prompt and run as Administrator
- To check version of java: ***java –version***
- To check version of Hadoop: ***hadoop version***
- Format the namenode using the command:

  hdfs namenode -format

- After formatting, open Hadoop sbin folder using the command:

  ***cd ***

  cd <hadoop sbin path>

- Start hadoop services using the command:

  start-all.cmd (Starts both yarn and hdfs services)

  Or

  Start hadoop services separately using:

  start-dfs.cmd

  start-yarn.cmd

- To check if all the services are running properly, use the following command:

  jps

- Go to your web browser and type **"localhost:9870"** or **"localhost:50070"** to check the hadoop services are running properly.
- To check resource manager, type **"localhost:8088".**

EXP 2: Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.

- Start hadoop services using the command:

  start-all.cmd (Starts both yarn and hdfs services)

  Or

  Start hadoop services separately using:

  start-dfs.cmd

  start-yarn.cmd

- To check if all the services are running properly, use the following command:

  jps

- To create a new directory "WordCount" in localhost:

  hdfs dfs -mkdir /WordCount

- To upload the input text file inside the "WordCount" directory:

  hdfs dfs -put <path to input.txt file> /WordCount

- To run the mapreduce program using mapper.py and reducer.py files:

  hadoop jar C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.3.6.jar ^
  -input /WordCount/input.txt ^
  -output /WordCount/output ^
  -mapper "python <path to mapper.py file>" ^
  -reducer "python <path to reducer.py file>"

- To check output in cmd, run the command:

  hdfs dfs -cat /WordCount/output/part-00000

- To check output in localhost, browse for **"WordCount"** directory and go to **output-> part-00000.**

EXP 3: Map Reduce program to process a weather dataset.

- Start hadoop services using the command:

  start-all.cmd (Starts both yarn and hdfs services)

  Or

  Start hadoop services separately using:

  start-dfs.cmd

  start-yarn.cmd

- To check if all the services are running properly, use the following command:

  jps

- To create a new directory "WeatherData" in localhost:

  hdfs dfs -mkdir /WeatherData

- To upload the input text file inside the "WeatherData" directory:

  hdfs dfs -put <path to sample_weather.txt file> /WeatherData

- To run the mapreduce program using mapper.py and reducer.py files:

  ```
  hadoop jar C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.3.6.jar ^
  -input /WeatherData/input.txt ^
  -output /WeatherData/output ^
  -mapper "python <path to mapper.py file>" ^
  -reducer "python <path to reducer.py file>"
  ```

- To check output in cmd, run the command:

  hdfs dfs -cat /WeatherData/output/part-00000

- To check output in localhost, browse for **"WeatherData"** directory and go to **output-> part-00000.**

EXP 4: Create UDF (User Defined Functions) in Apache Pig and execute it in MapReduce / HDFS mode

- Download and install pig-0.17.0.
- Set path of pig and pig-bin folders in system variables.
- Open Command Prompt and run as Administrator.
- Start hadoop services using the command:

  start-all.cmd (Starts both yarn and hdfs services)

  Or

  Start hadoop services separately using:

  start-dfs.cmd

  start-yarn.cmd

- To check if all the services are running properly, use the following command:

  jps

- Open pig bin folder using the command:

  ***cd ***

  cd <pig bin path>

- Start Apache pig by typing:

  p***ig***

- A grunt shell will open, indicating that pig is installed properly.
- To quit pig, type:

  quit;

- Create a new directory "Pig_UDF" in localhost:

  hdfs dfs –mkdir /Pig_UDF

- Upload the input sample text file inside the "Pig_UDF" directory:

  hdfs dfs -put <path to sample.txt file> /Pig_UDF

- Create another new directory "udfs" inside "Pig_UDF" in localhost:

  hdfs dfs –mkdir /UDF/udfs

- Upload the python file which has the user-defined function inside the “Pig_UDF/udfs” directory:

  hdfs dfs -put <path to udf.py file> /Pig_UDF/udfs

- Execute the pig file using the command:

  pig -x mapreduce <path to UDF.pig>

- To check output in cmd, run the command:

  hdfs dfs -cat /Pig_UDF/output/part-m-00000

- To check output in localhost, browse for **“Pig_UDF”** directory and go to **output-> part-m-00000.**

EXP: 5 Create tables in Hive and write queries to access the data in the table

- Download and install Apache Derby 10.15.2.0, and set it's environment variables.
- Download and install Apache Hive 3.1.3 and set it's environment variables.
- Start hadoop services using the command:

  start-all.cmd (Starts both yarn and hdfs services)

  Or

  Start hadoop services separately using:

  start-dfs.cmd

  start-yarn.cmd

- Open Windows Powershell and run as Administrator.
- To open derby, run the following command:

  StartNetworkServer -h 0.0.0.0

- To check if all the services are running properly, open a new Command Prompt and use the following command:

  jps

- To open Apache Hive, run the following command:

  hive --service schematool -dbType derby –initSchema

- Open hive bin folder using the command:

  ***cd ***

  cd <hive bin path>

- Start Apache hive by typing:

  hive

Queries:

1. To create a new database:

   CREATE DATABASE <database_name>;

2. To verify if the database is present:

SHOW DATABASES;

3. To switch to the new database:

USE <database_name>;

4. To create a table in Hive:

CREATE TABLE <table_name>(<variable_name>, <data_type>);

5. To insert values into the table:

INSERT INTO <table_name> VALUES (<value1>, <value2>, …..,<value_i>);

6. To query your data:

CREATE VIEW <view_name> AS SELECT <variable_name> FROM <table_name>;

7. To show all tables in a selected database, use the following statement:

SHOW TABLES;

8. To show table column names and data types, run:

DESCRIBE <table_name>;

9. To display table data, use a **SELECT** statement. For example, to select everything in a table, run:

SELECT*FROM <view_name>;

10. To alter a table, use the following command:

ALTER TABLE <table_name> ADD COLUMNS (<variable_name> <data_type>);

11. To quit Hive, type:

quit;

EXP NO:6 Import a JSON file from the command line. Apply the following actions with the data present in the JSON file where, projection, aggregation, remove, count, limit, skip and sort

- Open Windows Powershell and run as Administrator.
- Run the following command to install choco:

  Set-ExecutionPolicy Bypass -Scope Process -Force;
  [System.Net.ServicePointManager]::SecurityProtocol =
  [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object
  System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1')

- To install jq, run the following command:

  choco install jq

Running jq queries:

I. **Projection:** Extracts variables from each element.

jq ".[] | {variable1: .variable1, variable2: .variable2}" <json file path>

II. **Aggregation:** Calculates the total sum of variable1 across all elements.

jq "[.[] | .variable1] | add" <json file path>

III. **Remove:** Removes the variable1 field from every element.

jq "del(.[] | .variable1)" <json file path>

IV. **Count:** Returns the number of elements in the array.

jq ". | length" <json file path>

V. **Limit:** Selects the first ‘n’ elements in the array.

jq ".[0:<n>]" <json file path>

VI. **Skip:** Skips the first ‘n’ elements in the array and returns the other values.

jq ".[<n>:]" <json file path>

VII. **Sort:** Sorts the elements by a variable in Ascending order.

jq "sort_by(.variable1)" <json file path>