NAME – SHUBHAM KUMAR
COURSE – MCA (REGULAR)
REG. NO – 2019202053

GUIDED BY:-
DR. H KHANNA NEHEMIAH

PIG AND HIVE PROJECT

BIG DATA:

Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not the amount of data that's important. It's what organizations do with the data that matters. Big data can be analysed for insights that lead to better decisions and strategic business moves.

Why Is BIG DATA Important?

The importance of big data doesn't revolve around how much data you have, but what we do with it. We can take data from any source and analyze it to find answers that enable

1. Cost reductions
2. Time reductions
3. New product development and optimized offerings
4. Smart decision making. When we combine big data with high-powered analytics.

HADOOP:

Hadoop is an open-source framework to store and process Big Data in a distributed environment. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs. It contains two modules, one is MapReduce and another is Hadoop Distributed File System (HDFS).

- **MapReduce:**

  It is a parallel programming model for processing large amounts of structured, semi-structured, and unstructured data on large clusters of commodity hardware.

- **HDFS:**

  Hadoop Distributed File System is a part of Hadoop framework, used to store and process the datasets. It provides a fault-tolerant file system to run on commodity hardware.

Importance of HADOOP

Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.

The Hadoop ecosystem contains different sub-projects (tools) such as Sqoop, Pig, and Hive that are used to help Hadoop modules.

- **Sqoop:**
  It is used to import and export data to and from between HDFS and RDBMS.
- **Pig:**
  It is a procedural language platform used to develop a script for MapReduce operations.
- **Hive:**
  It is a platform used to develop SQL type scripts to do MapReduce operations.

HIVE:

Apache Hive is a data warehouse system for data summarization, analysis and querying of large data systems in open source Hadoop platform. It converts SQL-like queries into MapReduce jobs for easy execution and processing of extremely large volumes of data.Hive is a data warehousing infrastructure built on top of apache Hadoop.

Hive enables easy data summarization, ad-hoc querying and analysis of large volumes of data.

It provides a simple query language called Hive QL, which is based on SQL and which enables users familiar with SQL to easily perform ad-hoc querying, summarization and data analysis.

Working of Hive:

The following diagram depicts the workflow between Hive and Hadoop.

The following table defines how Hive interacts with Hadoop framework:

STEP	OPERTAIONS
1	Execute Query The Hive interface such as Command Line or Web UI sends query to Driver (any database driver such as JDBC, ODBC, etc.) to execute.
2	Get Plan The driver takes the help of query compiler that parses the query to check the syntax and query plan or the requirement of query.
3	Get Metadata The compiler sends metadata request to Metastore (any database)
4	Send Metadata Metastore sends metadata as a response to the compiler.
5	Send Plan The compiler checks the requirement and resends the plan to the driver. Up to here, the parsing and compiling of a query is complete.
6	Execute Plan The driver sends the execute plan to the execution engine.
7	Execute Job Internally, the process of execution job is a MapReduce job. The execution engine sends the job to JobTracker, which is in Name node and it assigns this job to TaskTracker, which is in Data node. Here, the query executes MapReduce job.

8	Metadata Ops Meanwhile in execution, the execution engine can execute metadata operations with Metastore.
9	Send Results The execution engine sends those resultant values to the driver. The driver sends the results to Hive Interfaces.

Apache Pig

Apache Pig is an abstraction over MapReduce. It is a tool/platform which is used to analyze larger sets of data representing them as data flows. Pig is generally used with Hadoop; we can perform all the data manipulation operations in Hadoop using Apache Pig.

Apache Pig Vs Hive

Both Apache Pig and Hive are used to create MapReduce jobs. And in some cases, Hive operates on HDFS in a similar way Apache Pig does. In the following table, we have listed a few significant points that set Apache Pig apart from Hive.

Apache Pig	Hive
1. Apache Pig uses a language called Pig Latin. It was originally created at Yahoo.	Hive uses a language called HiveQL. It was originally created at Facebook.
2. Pig Latin is a data flow language. language.	HiveQL is a query processing
3. Pig Latin is a procedural language and it fits in pipeline paradigm.	HiveQL is a declarative language.
4. Apache Pig can handle structured, unstructured, and semi-structured data.	Hive is mostly for structured data.

INSTALLATION

---

1. Download **Java jdk-8u231-windows-x64.exe** from by visiting the following link **http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-1880260.html/**

   Then jdk-8u231-windows-x64.exe will be downloaded onto our system.

   We use **jdk version 8**

2. Install the **jdk-8u231-windows-x64.exe** in the **C:\Java** directrory
3. Set Path Of Java
4. Open Environment Variable
5. Click On New Button in User Variable Set Varaible_Name is **JAVA_HOME** and set Variable_value is **C:\Java**
6. Select Path Click on Edit Button
7. Click on New
8. Set **C:\Java\jdk1.8.0_231\bin** and Click on **Ok** Button
9. And then Click on OK Button
10. Verifying JAVA Installation
    i. Open Command Prompt
    ii. Type **java –version**
    java version "1.8.0_231"
    Java(TM) SE Runtime Environment (build 1.8.0_231-b11)
    Java HotSpot(TM) 64-Bit Server VM (build 25.231-b11, mixed mode)

HADOOP:

1. Download Hadoop2.7.7.zip file by visiting link https://hadoop.apache.org/releases.html
2. Extract **Hadoop 2.7.7.zip** and Copy into **C:/ drive**
3. Create **datanode** & **namenode** folder inside the **data** folder
4. Navigate the location **C:\hadoop-2.7.7\etc\hadoop**
5. Open **core-site.xml** in notepad and Add

```
<property>
    <name>fs.defaultFS</name>
```

<value>hdfs://localhost:9000</value>

</property> between

<configuration>

</configuration>

After Editing the Contents of core-site.xml

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

6. Open **hdfs-site.xml** in notepad and Add

```
<property>
    <name>dfs.replication</name>
    <value>1</value>
</property>
<property>
    <name>dfs.namenode.name.dir</name>
    <value>C:\hadoop-2.7.7\data\namenode</value>
</property>
<property>
    <name>dfs.datanode.data.dir</name>
    <value>C:\hadoop-2.7.7\data\datanode</value>
</property> between
 <configuration>
</configuration>
```

After Editing the Contents of hdfs-site.xml

```
<configuration>
```

```
        <property>
            <name>dfs.replication</name>
            <value>1</value>
        </property>
        <property>
            <name>dfs.namenode.name.dir</name>
            <value>C:\hadoop-2.7.7\data\namenode</value>
        </property>
        <property>
            <name>dfs.datanode.data.dir</name>
            <value>C:\hadoop-2.7.7\data\datanode</value>
        </property>
    </configuration>
```

7. Open **mapred-site.xml** in notepad and Add

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property> between

<configuration>

</configuration>

<u>After Editing the Contents of mapred -site.xml</u>

```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
```

8. Open **yarn-site.xml** in notepad and Add

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce_shuffle</value>
</property>

<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property> between

<configuration>

</configuration>

After Editing the Contents of mapred -site.xml

```
<configuration>

<!-- Site specific YARN configuration properties -->
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>

    <property>
        <name>yarn.nodemanager.aux-
        services.mapreduce.shuffle.class</name>
        <value>org.apache.hadoop.mapred.ShuffleHandler</v
        alue>
    </property>
</configuration>
```

9. Open Environment Variable
10. Click On New Button in User Variable Set Varaible_Name is **HADOOP_HOME** and set Variable_value is **C:\hadoop-2.7.7**
11. Click On New Button in User Variable Set Varaible_Name is **HADOOP_USER_CLASSPATH_FIRST** and set Variable_value is **true**
12. Select Path Click on Edit Button
13. Click on New
14. Set **C:\hadoop-2.7.7\bin** and Click on **Ok** Button
15. Set **C:\hadoop-2.7.7\sbin** and Click on **Ok** Button
16. And then Click on **OK** Button

17. Verifying Hadoop Installation
    I. Open Command Prompt (With Admin Privilege)
    II. Type **start-all** and press Enter
    III. Open another Command Prompt(With Admin Privilege)
    IV. Type **startNetworkServer –h 0.0.0.0** and press Enter
    V. Open any browser type URL **localhost:8088**, **localhost:50070** and **localhost:50075**

DERBY:

1. Open Environment Variable
2. Select Path Click on Edit Button
3. Click on New
4. Set **C:\derby\bin** and Click on **OK** Button
5. And then Click on **OK** Button

HIVE:

1. Downloading Hive, We use hive here. We can download it by visiting the following link **http://mirrors.wuchna.com/apachemirror/hive/** Let us assume it gets downloaded onto the Downloads directory. Here, we download Hive archive named “apache-hive-3.1.1-bin.tar.gz” for this installation. The following command is used to verify the download:
2. Extract the File and Copy on the C:\ drive
3. Copy files of Derby Library (C:\derby\lib) to Hive library (C:\hive\lib)
4. Open Environment Variable
5. Click On New Button in User Variable Set Varaible_Name is **HIVE_HOME** and set Variable_value is **C:\hive**
6. Select Path Click on Edit Button
7. Click on New
8. Set **C:\hive\bin** and Click on **Ok** Button
9. And then Click on **OK** Button
10. Execute the Hive
    i. Open Command Prompt (With Admin Privilege)
    ii. Type **hive** and press Enter

PIG:

1. Download Apache Pig, First of all, download the latest version of Apache Pig from the following website - **http://mirrors.wuchna.com/apachemirror/pig/**
2. Extract the File and Copy on the C:\ drive
3. Navigate C:\pig-0.17.0\bin, Open pig.cmd on notepad and edit the contents as
   set HADOOP_BIN_PATH=C:\hadoop-2.7.7\bin
   set hadoop-config-script=C:\hadoop-2.7.7\libexec\hadoop-config.cmd
4. Open Environment Variable
5. Click On New Button in User Variable Set Varaible_Name is **PIG_HOME** and set Variable_value is **C:\pig-0.17.0**
6. Select Path Click on Edit Button
7. Click on New
8. Set **C:\pig-0.17.0\bin** and Click on **Ok** Button
9. And then Click on OK Button
10. Execute the Pig
    i. Open Command Prompt (With Admin Privilege)
    ii. Type **pig –x local** and press Enter

COMMANDS

HIVE:

CREATING DATABASE

CREATE DATABASE CEG;

DISPLAYING DATABASE

SHOW DATABASES;

CHANGING DATABASE

USE CEG;

CREATING TABLE

```
CREATE TABLE IF NOT EXISTS AUTHOR
(
    AUTHOR_ID String,
    AUTHOR_NAME String
)
COMMENT 'AUTHOR DETAILS'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;

CREATE TABLE IF NOT EXISTS BOOK
(
    BOOK_ID String,
    BOOK_TITLE String,
    PUBLISHER String,
    NO_OF_BOOK int,
    AUTHOR_ID String
)
COMMENT 'BOOK DETAILS'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

LOADING THE DATA (TEXT FILE) INTO THE TABLE

```
LOAD DATA LOCAL INPATH
'/C:/Users/gadge/Desktop/PIG_AND_HIVE/AUTHOR_DATA.txt'
OVERWRITE INTO TABLE AUTHOR;
```

```
LOAD DATA LOCAL INPATH
'/C:/Users/gadge/Desktop/PIG_AND_HIVE/BOOK_DATA.txt'
OVERWRITE INTO TABLE BOOK;
```

SELECTING THE DATA FROM THE TABLE

```
SELECT * FROM AUTHOR;
```

```
SELECT * FROM BOOK;
```

SELECTING THE DATA FROM THE TABLE BOOK WHOSE NO_OF_BOKOK IS GREATER THAN 15

```
SELECT * FROM BOOK WHERE NO_OF_BOOK>15;
```

SELECTING THE BOOK NAME AND PUBLISHER OF THE BOOK FROM THE TABLE BOOK

```
SELECT BOOK_TITLE, PUBLISHER FROM BOOK;
```

SELECTING THE DATA FROM THE TABLE BOOK WHOSE PUBLICATION IS ARIHANT

```
SELECT * FROM BOOK WHERE PUBLISHER='ARIHANT';
```

SELCTING THE DATA FROM THE TABLE WRITER AND ORDERING THE DATA IN ASCENDING ORDER (USING ORDER BY)

SELECT BOOK_TITLE, PUBLISHER FROM BOOK ORDER BY BOOK_TITLE;

USING GROUP BY

SELECT PUBLISHER, COUNT(*) FROM BOOK GROUP BY PUBLISHER;

JOIN

SELECT * FROM

(SELECT * FROM BOOK)B

JOIN

(SELECT * FROM AUTHOR)A

ON A.AUTHOR_ID=B.AUTHOR_ID;

SELECT B.BOOK_TITLE, B.PUBLISHER, A.AUTHOR_NAME FROM BOOK B JOIN AUTHOR A ON (B.AUTHOR_ID=A.AUTHOR_ID);

LEFT OUTER JOIN

SELECT * FROM

(SELECT BOOK_TITLE FROM BOOK)B

LEFT OUTER JOIN

(SELECT AUTHOR_NAME FROM AUTHOR)A

ON A.AUTHOR_ID=B.AUTHOR_ID;

RIGHT OUTER JOIN

SELECT * FROM

(SELECT BOOK_TITLE FROM BOOK)B

```
RIGHT OUTER JOIN
(SELECT AUTHOR_NAME FROM AUTHOR)A
ON A.AUTHOR_ID=B.AUTHOR_ID;
```

FULL OUTER JOIN

```
SELECT * FROM
(SELECT BOOK_TITLE FROM BOOK)B
FULL OUTER JOIN
(SELECT AUTHOR_NAME FROM AUTHOR)A
ON A.WRITER_ID=B.WRITER_ID;
```

PIG:

LOADING DATA FROM FILE

```
author = load
'C:/Users/gadge/Desktop/PIG_AND_HIVE/PIG/AUTHOR_DATA.t
xt' using PigStorage(',') as
(author_id:chararray,author_name:chararray);


book = load
'C:/Users/gadge/Desktop/PIG_AND_HIVE/PIG/BOOK_DATA.txt'
using PigStorage(',') as
(book_id:chararray,author_name:chararray,publication:chararray
,no_of_book:int,author_id:chararray);
```

PROJECTING DATA

```
dump author;
dump book;
```

DESCRIBING

describe author;

describe book;

GROUPING

groupby_publtype= group book by publication;

dump groupby_publtype;

grouped = group book by author_id;

sum = foreach grouped generate group , SUM(book.no_of_book);

DISPLAY RESULT IMMEDIATELY

dump sum;

STORING THE IMMEDIATE RESULT

store sum into 'C:/Users/gadge/Desktop/PIG_AND_HIVE/sum';

d = load 'C:/Users/gadge/Desktop/PIG_AND_HIVE/sum';

dump d

JOIN OPERATION

book_author1 = join book by author_id,author by author_id;

dump book_author1

book_author2 = JOIN book BY author_id LEFT OUTER, author BY author_id;

dump book_author2;