Sinhgad Institutes

Name of the Student:_____**Roll no:**_____

CLASS:- T.E.[I.T] **Division: A**

Course: - 2019

Subject: 314457: Data Science and Big Data Analytics Laboratory

PART_ A _Assignment No. 03

Marks: _____/10

Date of Performance: ___/____/_____ **Sign with Date: _____**

Part- A

ASSIGNMENT NO: 03

TITLE:

Application using HiveQL for flight information system

AIM:

To write an application using HiveQL for

1) Creating, Dropping, and altering Database tables
2) Creat an external Hive table to connect to the HBase for Customer Information Table
3) Load table with data, insert new values and field in the table, and Join tables with Hive
4) Create index on Flight information Table
5) Find the average departure delay per day in 2008.

OBJECTIVE:

1) To apply Big data primitives and fundamentals for application development.
2) To design algorithms and techniques for Big data analytics.

SOFTWARE USED: Hadoop 3.2.2, hive-3.2.1, jdk11.

THEORY:

HiveQL: 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 is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.

Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive. It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.

Hive is not

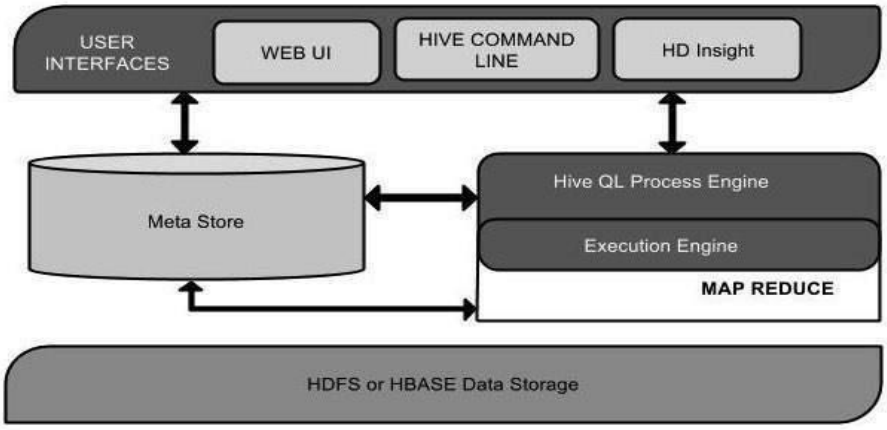
- A relational database
- A design for OnLine Transaction Processing (OLTP)
- A language for real-time queries and row-level updates

Features of Hive

- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP.
- It provides SQL type language for querying called HiveQL or HQL.
- It is familiar, fast, scalable, and extensible.

Architecture of Hive

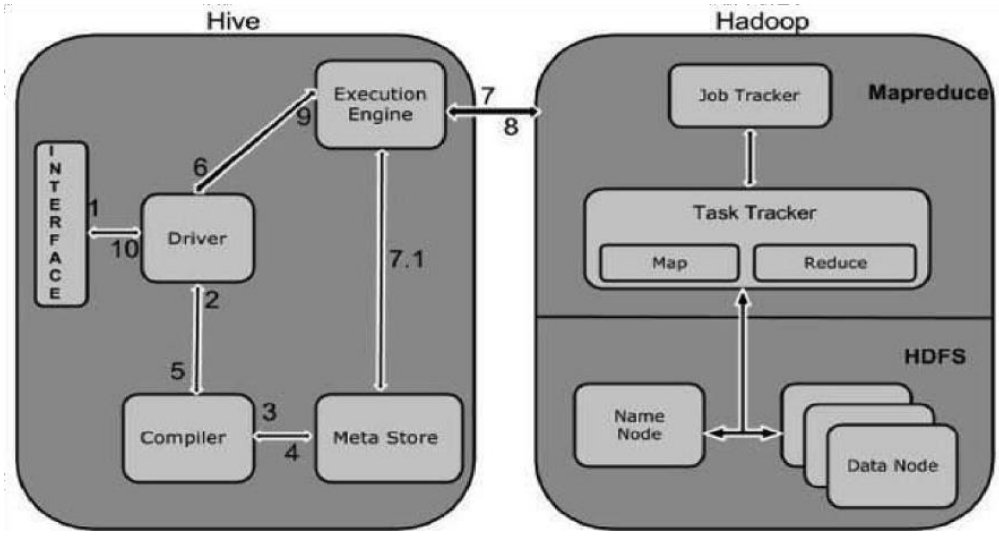
The following component diagram depicts the architecture of Hive:



Working of Hive

The following diagram depicts the workflow between Hive and Hadoop.

Hive



provides SQL-like declarative language, called HiveQL, which is used for expressing queries. Using Hive-QL users associated with SQL are able to perform data analysis very easily. Hive comes with a command-line shell interface which can be used to create tables and execute queries.

Hive query language is similar to SQL wherein it supports subqueries. With Hive query language, it is possible to take a MapReduce joins across Hive tables. It has a support for simple SQL like functions- CONCAT, SUBSTR, ROUND etc., and aggregation functions- SUM, COUNT, MAX etc. It also supports GROUP BY and SORT BY clauses. It is also possible to write user defined functions in Hive query language.

Download Hive from: https://hive.apache.org/downloads.html

Installation steps:

HIVE INSTALLATION

```
cd /home/student/Downloads
sudo tar xzf apache-hive-1.2.1-bin.tar.gz
sudo mkdir -p /usr/local/hive
cd apache-hive-1.2.1-bin
sudo mv * /usr/local/hive
sudo chown -R hduser:hadoop /usr/local/hive
cd
sudo nano .bashrc
##############hive ############################
Export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin
##########################################
cd /usr/local/hadoop/bin
hadoop fs -mkdir /tmp
hadoop fs -mkdir -p /user/hive/warehouse
hadoop fs -chmod g+w /tmp
hadoop fs -chmod g+w /user/hive/warehouse
cd /usr/local/hive/bin
start-all.sh
hive
```

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-1.2.1.jar!/hive-log4j.properties

hive>exit;

###############

stop-dfs.sh

stop-yarn.sh

Data Types in Hive

All the data types in Hive are classified into four types, given as follows:

- Column Types
- Literals
- Null Values
- Complex Types

Column Types

Column type are used as column data types of Hive. They are as follows:

Integral Types

Integer type data can be specified using integral data types, INT. When the data range exceeds the range of INT, you need to use BIGINT and if the data range is smaller than the INT, you use SMALLINT. TINYINT is smaller than SMALLINT.

The following table depicts various INT data types:

Type	Postfix	Example
TINYINT	Y	10Y
SMALLINT	S	10S
INT	-	10
BIGINT	L	10L

String Types

String type data types can be specified using single quotes (' ') or double quotes (" "). It contains two data types: VARCHAR and CHAR. Hive follows C-types escape characters.

The following table depicts various CHAR data types:

Data Type	Length
VARCHAR	1 to 65355
CHAR	255

Timestamp

It supports traditional UNIX timestamp with optional nanosecond precision. It supports java.sql.Timestamp format "YYYY-MM-DD HH:MM:SS.fffffffff" and format "yyyy-mm-dd hh:mm:ss.fffffffff".

Dates

DATE values are described in year/month/day format in the form {{YYYY-MM-DD}}.

Decimals

The DECIMAL type in Hive is as same as Big Decimal format of Java. It is used for representing immutable arbitrary precision. The syntax and example is as follows:

DECIMAL(precision, scale)
decimal(10,0)

Union Types

Union is a collection of heterogeneous data types. You can create an instance using **create union**. The syntax and example is as follows:

UNIONTYPE<int, double, array<string>, struct<a:int,b:string>>
{0:1}
{1:2.0}
{2:["three","four"]}
{3:{"a":5,"b":"five"}}
{2:["six","seven"]}
{3:{"a":8,"b":"eight"}}
{0:9}
{1:10.0}

Literals

The following literals are used in Hive:

Floating Point Types

Floating point types are nothing but numbers with decimal points. Generally, this type of data is composed of DOUBLE data type.

Decimal Type

Decimal type data is nothing but floating point value with higher range than DOUBLE data type. The range of decimal type is approximately -10^{-308} to 10^{308}.

Null Value

Missing values are represented by the special value NULL.

Complex Types

The Hive complex data types are as follows:

Arrays

Arrays in Hive are used the same way they are used in Java.

Syntax: ARRAY<data_type>

Maps

Maps in Hive are similar to Java Maps.

Syntax: MAP<primitive_type, data_type>

Structs

Structs in Hive is similar to using complex data with comment.

Syntax: STRUCT<col_name : data_type [COMMENT col_comment], ...>

Few Hive Commands

1. Create Database Statement :

Syntax:

CREATE DATABASE|SCHEMA [IF NOT EXISTS] <database name>

e.g create database sampledatabase;

2. Display databases

Show databases;

3.Create Table

Syntax:

CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.] table_name
[(col_name data_type [COMMENT col_comment], ...)]
[COMMENT table_comment]
[ROW FORMAT row_format]
[STORED AS file_format]

e.g. hive> CREATE TABLE IF NOT EXISTS employee (eid int, name String,
salary String, destination String)
COMMENT 'Employee details'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;

4. Load Data Statement : It is used LOAD DATA to store bulk records.

Syntax :

LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE tablename
[PARTITION (partcol1=val1, partcol2=val2 ...)]

e.g. hive> LOAD DATA LOCAL INPATH '/home/user/sample.txt' OVERWRITE INTO TABLE employee;

5. Alter Table Statement

ALTER TABLE name RENAME TO new_name
ALTER TABLE name ADD COLUMNS (col_spec[, col_spec ...])
ALTER TABLE name DROP [COLUMN] column_name

ALTER TABLE name CHANGE column_name new_name new_type
ALTER TABLE name REPLACE COLUMNS (col_spec[, col_spec ...])

e.g. 1. hive> ALTER TABLE employee RENAME TO emp;
2. hive> ALTER TABLE employee CHANGE name ename String;
3. hive> ALTER TABLE employee CHANGE salary salary Double
4. hive> ALTER TABLE employee ADD COLUMNS (
dept STRING COMMENT 'Department name');

5. Drop Table

Syntax:
DROP TABLE [IF EXISTS] table_name;

6. Partitioning the Table

Hive organizes tables into partitions. It is a way of dividing a table into related parts based on the values of partitioned columns such as date, city, and department. Using partition, it is easy to query a portion of the data.

Tables or partitions are sub-divided into **buckets,** to provide extra structure to the data that may be used for more efficient querying. Bucketing works based on the value of hash function of some column of a table.

Adding a Partition : We can add partitions using alter table

General Syntax : ALTER TABLE table_name ADD [IF NOT EXISTS] PARTITION partition_spec [LOCATION 'location1'] partition_spec [LOCATION 'location2'] ...;
partition_spec:
: (p_column = p_col_value, p_column = p_col_value, ...)

e.g. hive> ALTER TABLE employee ADD PARTITION (year=’2012’) location /
2012/part2012';

Renaming a partition

ALTER TABLE table_name PARTITION partition_spec RENAME TO PARTITION partition_spec;
e.g. hive> ALTER TABLE employee PARTITION (year=’1203’)
RENAME TO PARTITION (Yoj=’1203’);

7. ***Creating an Index:***

An index means pointer on a particular column in a table.

Syntax:

CREATE INDEX index_name ON TABLE base_table_name (col_name, ...)
AS 'index.handler.class.name' [WITH DEFERRED REBUILD] [IDXPROPERTIES (property_name=property_value, ...)] [IN TABLE index_table_name]
[PARTITIONED BY (col_name, ...)] [[ROW FORMAT ...] STORED AS ...
| STORED BY ...] [LOCATION hdfs_path]

e.g. hive> CREATE INDEX inedx_salary ON TABLE employee(salary) AS 'COMPACT WITH DEFERRED REBUILD'

CONCLUSION:

After the study of this assignment, we write an application using HiveQL

Write Short Answers for Following Questions:

1) What is Hive and HiveQL?
2) What are the different components of Hive architecture?
3) What is the use of Partitions and Bucketing in Hive?
4) What kind of Joins supported by Hive.
5) Write Syntax for creating table, loading data in table using Hive.
6) Where is table data stored in Apache Hive by default?
7) When executing Hive queries in different directories, why is metastore_db created in all places from where Hive is launched?
8) How will you read and write HDFS files in Hive?
9) Differentiate between describe and describe extended.

Viva Questions:

1) Explain about SORT BY, ORDER BY, DISTRIBUTE BY and CLUSTER BY in Hive.
2) What is the difference between HBase and Hive?
3) What is the default database provided by Hive for Metastore?
4) What is Apache Hbase and its use?
5) Give the name of the key components of Hbase.
6) What is the use of get () method?
7) What is the reason of using HBase?
8) Define column families in Hbase?
9) Define standalone mode in HBase?
10) What is regionserver?
11) What is the use of MasterServer and HMaster?
12) What are the operational commands of HBase?
13) Which command is used to run HBase Shell?
14) What is the use of ZooKeeper?