First Name - Palak
Last Name - Gupta
Student 700 Number - 700706770

Big data Final Exam

Answer 1:- The three configuration modes for running Hive cli service with respect to the metastore service and the metastore database are:-

- Embedded Mode
- Local Mode
- Remote Mode

- Embedded Mode is the default Mode for CDH. In this mode, the metastore uses a Derby Database, and both the database and the metastore service are embedded in the main Hive server process.

- In local mode, the Hive service runs in the same process as the main Hive service process, but the metastore database runs in a seperate process, and can be on a seperate host.

- In Remote Mode, the Hive metastore service runs in its own JVM process. HiveServer2, HCatalog, Impala, and other processes communicate with it using the Thrift Network API.

Answer 2:- False. Hive runs on top of hadoop.

Answer 4:-

a) To list existing database

(default) > show databases;

b) To list existing Tables

(default)> show tables;

c) To list all tables in a database called companydb

(default) > Show Tables in companydb;

d) To list all records for employees table that exists in companydb.

(default)> Select * from Companydb.employees;

e) To list only five records from products table that exist in companydb.

(default) > Select * from companydb.products limit 5;

f) To show the current execution engine

(default) > SET hive.execution.engine;

g) To show the default file system in Hadoop

(default) > SET fs.defaultFS;

h) To display the content of the user's home directory in HDFS

(default)> hdfs dfs -ls home/user/file

i) To print the current working directory

(default)> !pwd

j) To append data from a file stored locally in the current working directory into an existing table called mytable. let the filename be foo.txt

(default) > Load Data local INPATH 'foo.txt'
INTO TABLE mytable ;

k) To find whether the table mytable is an external table or not.

(default) > Describe Embedded mytable ;

table type : Managed-Table if managed
External-Table if external

Answer 5 :-

Welcome
To
Programming
Hive!

Answer 6:- • Schema-On-Read helps in very fast initial data load, since data does not have to follow any internal schema to read, as it is just a copy/move of a file.

• Schema-On-Write helps in faster performance of the query, as the data is already loaded in a particular format and it is easy to locate the column index.

→ So in scenarios of large data load or where the schema is not known at load time, Schema-On-Read is more efficient than Schema-On-Write

Hive uses Schema-On-Read.

Answer-13:- Lazy evaluation in spark means that the execution will not start until an action is triggered in. In Spark, the picture of lazy evaluation comes in when Spark Transformations occur.

Transformations are lazy in nature means when we call some operations in RDD, it does not execute immediately. Spark maintains the record of which operation is being called.

No, it does not apply to all Spark operations. It only apply's on transformation operations and not on action operations.

Answer 12:- Commands in spark python shell to find the sum of all the values in the range 0 to 99 are:-

```
Val rdd1 = sc.parallelize (0 to 99)
rdd1.sum
```

Answer7:- When a table is created, by default Hive will manage the data, which means that Hive moves the data into its warehouse directory.
Hive does not manage the data of the external Table. External table is used for external use as when we want to use data outside the Hive. External tables are stored outside the warehouse directory.

→ To create external Table:-

```
Create External Table mytable (wban INT, date STRING)
Row Format Delimited
Fields terminated by ','
Location '/hive/data';
Load Data Inpath '/data/dataset-2020';
```

Answer 9:- `sc.textfile ("hdfs://data/logfiles").count()`

Answer 10:- `sc.textfile ("hdfs://data/logfiles").filter (lambda x: "error" in x).count()`

Answer 11:- sc.textfiles("hdfs://data/logfiles").map(lambda x: len(x)).sum()

Answer 3:-
```
create table lines (line string);
load data INPATH 'transactions.txt'
Into table lines;
select (sum (length (line)))' from lines;
```

Answer 8:- a)
```
Create Table if Not Exists
Customer (Customer id INT, cust_Name String,
street string, city String, zip int, region
string)
Partitioned By (country string)
Comment 'Customer details'
Row format Delimited
Fields Terminated by '\t'
lines Terminated by '\n'
stored as Textfile;
```

Answer 8:- b) Set hive.exec.dynamic.partition=true

Set hive.exec.dynamic.partition.mode=nonstrict

Create External Table IF Not Exists

Customer (customer_id int, cust_name string,

street string, city string, zip int, region string)

Partitioned by (country string)

Comment 'Customer details'

Row Format Delimited

Fields Terminated by '\t'

lines Terminated by '\n'

Stored as Textfile

location '/data/customers/';

Answer 8:- c) As we don't need any intermediate stages for transforming data, in the mentioned scenario it is not required to have hive dynamic partition.

Answer 8:- d) When a query is fired on data stored in hive partitioned tables, hive execution engine searches for data based on partitions created on desired columns rather than entire table scan.

In full table scan (without partitions), the execution engine scans each and every files stored in hadoop directory where as in partitioned scan, the engine scans chunks of divided data

created as a result of Partions. Indeed partition scan is much more efficient compared to full table scan.