NAME: VINEESHA PEDDI

COURSE : BIGDATA CS 5620 TAKE HOME FINAL EXAM

STUDENT ID: 700701688

CS 5620 – Final Exam

First Name: Vineesha
Last Name: Peddi
Student 700 Number: 700701688

Question 01:

Three modes for hive metastore are

1. Embedded Metastore
2. Local Metastore
3. Remote Metastore

Embedded Metastore	Local Metastore	Remote Metastore
In embedded Metastore both metastore service and hive service runs in the same JVM by using embedded Derby database. This mode has a limitation that it allows only a single user session to connect to metastore. If we try to open second session it results in error. It is not suggested for protection	To overcome the limitation of embedded metastore, this mode allows to have many hive sessions. This can be achieved by using any JDBC databases like MySQL which runs in different machine that that of hive ~~session~~ service and metastore service running in the same JVM.	In remote mode, metastore runs on its own separate JVM, not in the hive service JVM. Thrift Network API's can be used to communicate with the metastore servers & Other Process

Question-2

False,

Because Hive is a hadoop client and it runs on top of hadoop it's not required to install hive on every node in the cluster to run the hive queries.

Question-3

```
Create Table test(line string);

Load data inpath 'test'.
Overwrite into table test;

Select sum(length(line)) from test;
```

Question-4:-

a) Show databases;

b) Show tables;

c) Show tables in companydb;

d) Select * from companydb.employees;

e) Select * from companydb.products limit 5;

f) Set hive.execution.engine;

g) Set hive.metastore.warehouse.dir;

h) To display the content under users home directory

dfs -cat /*;

To list the files

dfs -ls /;

i) Pwd

j) Load data local inpath './foo.txt' into table mytable;

k) describe extended mytable; (or)

describe formatted mytable;

Question-5:

Explode function is used to split output in individual tokens rather than a list

Example: welcome
to
programming
Hive!

Question-6:

Schema on Write: In traditional database, database has Control over storage. Here data is being checked against the schema when written into the db. during load.

Schema on Read: Hive has no control over storage and the data schema is not verified during load-time, rather it is verified while processing the query.

Hive uses this process called Schema on Read.

Question-7:

To share the data between tools we create external tables where external tables doesn't take ownership of data. When we delete the (drop) external table it does not delete the data but metadata for that table will be deleted.

```
Create external Table If not exists mytable (
        Student_id int,
        Student_name string)
Row format Delimited fields Terminated by ','
Location '/data/dataset-2020;
```

Question-8:

8.a:

```
Create Table Customers (
 Cust_id int,
 Cust_name string,
 Street string,
 City string,
 zip int,
 region string)
Partitioned by (country string);
```

8.b:

To load the data into customers table without moving files in HDFS, we will create one staging table.

```
Create table staging.table (
    cust_id int,
    cust_name string, street string,
    city string,
    zip string,
    region string)
Partitioned by(country string);

Load data inpath '/data/customers/usa'
into table staging-table
Partition (country = 'usa');

Load data inpath '/data/customers/canada'
into table staging_table
Partition(country = 'canada');

Load data inpath '/data/customers/mexico'
into table staging_table
Partition(country = 'mexico')
```

Now In order to allow dynamic partition first set the below properties in hive.

```
Set hive.exec.dynamic.partition=true;
Set hive.exec.dynamic.partition.mode=nonstrict;
```

Now insert the data to customers from staging_table

```
Insert into table customers partition(country)
Select cust_id, cust_name, street, city, zip, region,
       country from staging_table
```

8.c:-

Yes, we can use dynamic partition to avoid creating Id of partitions as we have many countries. In the question 8.b, we have loaded data using single SQL query using dynamic partition.

8.d:

Based on filter condition, where clause on partition table, we no need to scan thousands of records it is only necessary to scan the contents of one directory by this way for larger datasets, Partition can dramatically improve query performance.

Question-9:

```
lines = sc.textFile('./data/logfiles/*')
lines.count()
```

Question 10:

```
error_rdd = lines.filter(lambda line: 'error' in
                                    line.lower())

error_rdd.count()
```

Question 11:

```
string_count = lines.map(lambda line : len(line))
string_count.sum()
```

Question 12:

```
Import numpy as np. (Requires only if we use it)
numbers = sc.parallelize(range(0,100))
numbers.sum
```

Question 13:

Lazy Evaluation:

Lazy Evaluation in spark means execution of RDD will not start until an action is triggered.

This occurs when spark transformation comes in picture. Spark maintains the record of operations called through DAG (Directed Acyclic Graph) Since transformations are lazy in nature we can execute operation by calling an action only.

Example :- ~~Filter~~
Filter

Does it apply to every Spark operation?

No, It does not apply to every operation in Spark, lazy evaluations comes for transformations only