BIG DATA HADOOP & SPARK TRAINING

ACADGILD

ASSIGNMENT 7.3

BY :-

SAHIL KHURANA

PROBLEM STATEMENT

Explain the below concepts with an example in brief.

- Hive Data Definitions
- Hive Data Manipulations
- HiveQL Manipulations

Hive Data Definitions

The Driver for Apache Hive supports a broad set of DDL, including (but not limited to) the following:

(a) CREATE Database and DROP Database
(b) CREATE Table and DROP Table
(c) ALTER Table and Alter Partition statements
(d) CREATE View and Drop View
(e) CREATE Function and Drop Function

CREATE Database and DROP Database:

```
CREATE (DATABASE|SCHEMA) [IF NOT EXISTS] database_name
  [COMMENT database_comment]
  [LOCATION hdfs_path]
  [WITH DBPROPERTIES (property_name=property_value, ...)];
```

The uses of SCHEMA and DATABASE are interchangeable – they mean the same thing.

```
DROP (DATABASE|SCHEMA) [IF EXISTS] database_name [RESTRICT|CASCADE];
```

CREATE Table and DROP Table:

e.g.

```
CREATE TABLE IF NOT EXISTS
custom.olympix_data (athlete_name string,athlete_age int,athlete_country string,
year int,closing_date string,sport string,gold_medals int,
silver_medals int,bronze_medals int,total_medals int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
```

This query will create a table in the database custom with table name olympix_data.

```
DROP TABLE custom.olympix_data;
```

It will drop the table olympix_data from the database "custom".

Alter Table

```
ALTER TABLE table_name RENAME TO new_table_name;
```

This statement will change the name of a table to a different name.

Alter Partition

Partitions can be added, renamed, exchanged (moved), dropped, or (un)archived by using the PARTITION clause in an ALTER TABLE statement

ALTER TABLE table_name ADD [IF NOT EXISTS] PARTITION partition_spec
[LOCATION 'location'][, PARTITION partition_spec [LOCATION 'location'], ...];

partition_spec:
  : (partition_column = partition_col_value, partition_column = partition_col_value, ...)

Create View

CREATE VIEW [IF NOT EXISTS] [db_name.]view_name [(column_name [COMMENT column_co
  [COMMENT view_comment]
  [TBLPROPERTIES (property_name = property_value, ...)]
 AS SELECT ...;

CREATE VIEW creates a view with the given name. An error is thrown if a table or view with the same name already exists. we can use IF NOT EXISTS to skip the error.

Drop View

DROP VIEW removes metadata for the specified view.

DROP VIEW [IF EXISTS] [db_name.]view_name;

Create Function

This statement lets you create a function that is implemented by the class_name. Jars, files, or archives which need to be added to the environment can be specified with the USING clause; when the function is referenced for the first time by a Hive session, these resources will be added to the environment as if add jar file had been issued.

CREATE FUNCTION [db_name.]function_name AS class_name
  [USING JAR|FILE|ARCHIVE 'file_uri' [, JAR|FILE|ARCHIVE 'file_uri']];

Hive Data Manipulations

Loading files into tables

Hive does not do any transformation while loading data into tables. Load operations are currently pure copy/move operations that move datafiles into locations corresponding to Hive tables.

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

Update

Updates can only be performed on tables that support ACID.

UPDATE tablename SET column = value [, column = value ...] [WHERE expression]

Delete

Deletes can only be performed on tables that support ACID

DELETE FROM tablename [WHERE expression]

Merge

Merge can only be performed on tables that support ACID

MERGE INTO <target table> AS T USING <source expression/table> AS S
ON <boolean expression1>
WHEN MATCHED [AND <boolean expression2>] THEN UPDATE SET <set clause list>
WHEN MATCHED [AND <boolean expression3>] THEN DELETE
WHEN NOT MATCHED [AND <boolean expression4>] THEN INSERT VALUES<value list>

HiveQL Manipulations

1. Loading Data into Managed Tables.

2. Inserting Data into Tables from Queries.

3. Creating Tables and Loading Them in One Query.

4. Exporting Data.

Loading Data into Managed Tables:

```
LOAD DATA LOCAL INPATH
'/home/acadgild/Desktop/assignments_work/Data_from_acadgild/6.1/dataset_Sessi
on.txt' into table CUSTOM.TEMPERATURE_DATA;
```

It will load data in the table TEMPERATURE_DATA of the custom database. Data will be loaded from the text file dataset_Session.txt.

Inserting Data into Tables from Queries:

```
INSERT OVERWRITE TABLE employees
PARTITION (country, state)
SELECT ..., se.cnty, se.st
FROM staged_employees se;
```

Creating Tables and Loading Them in One Query

```
CREATE TABLE ca_employees
AS SELECT name, salary, address
FROM employees
WHERE se.state = 'CA';
```

Exporting Data

e.g.

```
hive -e 'select * from temperature_data_vw;' | sed 's/[[:space:]]\+/\|/g'  >
/home/sahil/Desktop/temperature_data_vw.txt;
```

It will store the data fetched from the query in a text file named temperature_data_vw.txt on the desktop of the local machine.