Apache Hadoop: Hive

Core Concepts, Architecture, and Optimizations

Lab Exercise-1 Workbook

HIVE Lab exercises

Exercise 1: Create Movies table (Managed)

Task 1: Use Sqoop to import movies table in to hive

```
$ sqoop import --connect jdbc:mysql://localhost/moviesdb --table movies --username training --password training --hive-import --delete-target-dir
```

Task 2: Check the imported table in HDFS. Execute the following HDFS commands

```
hdfs dfs  -ls /user/hive/warehouse/movies
hdfs dfs  -cat /user/hive/warehouse/movies/part-m-00000
```

Task 3: Fire an SQL query to find out the count of movies released in 1950,genre contains Drama

```
$hive
Select count(*) FROM movies WHERE year = 1950;
Select * from movies Where genre like '%Drama%';
Select * from movies Where genre like 'Drama';
```

Lab Exercise 02 – Managed and External Tables

Task 1: Creating a hive table manually (Managed Table)--beeline

```
$hive
hive> CREATE TABLE users(userid int , gender string,age int,occupation int, zipcode int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ':';
```

HIVE Lab exercises

Task 2: Load data in to a hive table

```
LOAD DATA local INPATH
"/home/ubuntu/training_materials/developer/data/movies/users.dat"
OVERWRITE INTO TABLE users;
```

Task 3: Creating EXTERNAL TABLES

```
CREATE EXTERNAL TABLE userratings (userid int,movieid int,rating
int,createtimestamp int) ROW FORMAT DELIMITED FIELDS
TERMINATED BY ':' LOCATION '/user/ubuntu/userratings'
```

Task 4: Load data in to a hive table

```
LOAD DATA local INPATH
'/home/ubuntu/training_materials/developer/data/movies/ratings.dat'
OVERWRITE INTO TABLE userratings;
```