Lab # : 2. Week #: 3;
Lab Name : Hive 2 - a few features; Subject Name : Information Storage and Retrieval; Lab Duration : 20 to 30 mins

Intro

In this lab, we will go over few more Hive features and commands, which are useful for managing data in Hive. We will go over the following features:

- Partitioning a Table
- Bucketing a Table
- Check Storage format of a Hive table
- Hive on MR vs Tez
- Hive Views
- User Defined Functions

Here are a few points to get to know Hive :

- Hive Partitioning allows
- Hive Bucketing allows
- Hive lets checking metadata of a Hive table
- Hive on Tez is a new feature in Hive, which runs faster than Hive alone on MR.
- Hive views could be created to filter data. Even UDF could be applied to views.
- For creating analytical operators, you can create custom User Defined Functions in java, python and other languages.

Let's go!

Step-1.PARTITIONED Table

Partitions are horizontal slices of data which allow large sets of data to be segmented into more manageable blocks. Partitioning creates folder at HDFS level.

```
CREATE TABLE Web_Session_Log_Partitioned(

DATETIME varchar(500), USERID varchar(500), SESSIONID varchar(500),
PRODUCTID varchar(500), REFERERURL varchar(500))

COMMENT 'This is the Twitter streaming data'

PARTITIONED BY(DATETIME STRING)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE;

FROM Web_Session_Log

INSERT OVERWRITE TABLE Web_Session_Log_Partitioned PARTITION
(DATETIME="2014-01-02 00:00:06 GMT") SELECT * where
PRODUCTID='/product/MT65XF2YA' limit 100;
```

Step-2. Bucketing a table

Bucketing is a technique that allows you to cluster or segment large sets of data to optimize query performance

```
CREATE TABLE Web_Session_Log_Bucketing
(DATETIME varchar(500),
USERID varchar(500),
SESSIONID varchar(500),
PRODUCTID varchar(500),
REFERERURL varchar(500))
COMMENT 'This is the Web Session Log data' PARTITIONED BY( PRODUCTID
STRING)
CLUSTERED BY(USERID) INTO 2 BUCKETS ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE;

set hive.enforce.bucketing = true;

FROM Web_Session_Log
INSERT OVERWRITE TABLE Web_Session_Log_Bucketing PARTITION
(PRODUCTID="/product/MT65XF2YA")
SELECT * where PRODUCTID='/product/MT65XF2YA' limit 100;
```

Step-3. Let’s check an existing table

```
describe Web_Session_Log

datetime varchar(500)
userid varchar(500)
sessionid varchar(500)
productid varchar(500)
refererurl varchar(500)
Time taken: 0.111 seconds, Fetched: 5 row(s)

describe formatted Web_Session_Log;

col_name data_type comment
datetime varchar(500)
userid varchar(500)
sessionid varchar(500)
productid varchar(500)
refererurl varchar(500)
Detailed Table Information
Database: default
Owner: ubuntu
CreateTime: Thu May 28 06:11:32 UTC 2015
LastAccessTime: UNKNOWN
Protect Mode: None
Retention: 0
Location: hdfs://ip-10-85-31-243.eu-west-1.compute.internal:8020/user/hive/warehouse/web_session_log
Table Type: MANAGED_TABLE
Table Parameters:
COLUMN_STATS_ACCURATE true
numFiles 1
numRows 0
rawDataSize 0
totalSize 4513792
transient_lastDdlTime 1432793495
Storage Information
SerDe Library: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat: org.apache.hadoop.mapred.TextInputFormat
```

```
OutputFormat: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed: No
Num Buckets: -1
Bucket Columns: []
Sort Columns: []
Storage Desc Params:
field.delim \t
serialization.format \t
Time taken: 0.1 seconds, Fetched: 36 row(s)
```

Step-4. Let's join two tables.

In Hive, you can do various kinds of joins like, inner join, left outer join, right outer join, etc.

Now let's join on useridid.

```
SELECT
Web_Session_Log.DATETIME,Web_Session_Log.USERID,User_Data.FIRSTNAME,User_Data.LASTNAME,User_Data.LOCATION,Web_Session_Log.PRODUCTID,Web_Session_Log.REFERERURL from Web_Session_Log JOIN User_Data ON (User_Data.USERID=Web_Session_Log.USERID);
```

Step-5. Hive on Tez.

Tez is a new application framework built on Hadoop Yarn that can execute complex directed acyclic graphs of general data processing tasks. In many ways it can be thought of as a more flexible and powerful successor of the map-reduce framework.

```
Set Tez Environment Variable on hive
set hive.execution.engine=tez;
you can change back to MR
set hive.execution.engine=mr;
```

Step-6. UDF ??

Let's write a simple udf function in python.

Streaming.py code:

```
import sys
from datetime import datetime
```

```
for line in sys.stdin.readlines():
boolVal = “false”
line = line.strip()
DATETIME = datetime.strptime(line, “%m/%d/%Y”)
print DATETIME
```

UDF (User Defined Function)

Add Pyhton file:

```
add file streaming.py;
```

Register Python function :

```
create table dev_schema.rpt_asset_extract as
select TRANSFORM(DATETIME) USING ‘streaming.py’ AS DATETIME from Web_Log_Data;
```