Hive-HBase Integration

IMPORTANT

Copyright Infringement and Illegal Content Sharing Notice

What are we trying to do?

We will see how to create a table which we can access from hive as well as hbase.

Remember: we want to access this table through hive if we want to do any processing or analysis.

We want to access this table through hbase for quick searching or transactional activities.

Hive-hbase integration

Step 1: Make sure file kv1.txt file is placed inside Downloads folder in cloudera VM (the delimiter in this file is CTRL-A character which is default in delimiter in hive)

```
[cloudera@quickstart Downloads]$ head kv1.txt
238val_238
86val_86
311val_311
27val_27
165val_165
409val_409
255val_255
278val_278
98val_98
484val_484
[cloudera@quickstart Downloads]$
```

Hive-hbase integration

Step 2: Create a table in hive

CREATE TABLE pokes (foo INT, bar STRING);

```
hive> CREATE TABLE pokes (foo INT, bar STRING);
OK
Time taken: 0.929 seconds
hive>
```

Hive-hbase integration

Step 3: Load the data in hive table

LOAD DATA LOCAL INPATH '/home/cloudera/Downloads/kv1.txt'
OVERWRITE INTO TABLE pokes;

```
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive>
hive> CREATE TABLE pokes (foo INT, bar STRING);
OK
Time taken: 0.929 seconds
hive> LOAD DATA LOCAL INPATH '/home/cloudera/Downloads/kv1.txt' OVERWRITE INTO TABLE pokes;
Loading data to table default.pokes
Table default.pokes stats: [numFiles=1, numRows=0, totalSize=5812, rawDataSize=0]
OK
Time taken: 0.648 seconds
```

Hive-hbase integration

Step 4: verify the data in hive table

SELECT * FROM pokes WHERE foo = 98;

```
hive> SELECT * FROM pokes WHERE foo = 98;
OK
98      val_98
98      val_98
Time taken: 0.51 seconds, Fetched: 2 row(s)
hive>
```

The output of the SELECT command displays two identical rows because there are two identical rows in the Hive pokes table with a key of 98.

Note: This is a good illustration of the concept that Hive tables can have multiple identical keys. As we will see shortly, HBase tables cannot have multiple identical keys, only unique keys.

Hive-hbase integration

step 5: create a Hive-HBase table

```
CREATE TABLE hbase_table_1(key int, value string) STORED BY
'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH
SERDEPROPERTIES ("hbase.columns.mapping" = ":key,cf1:val")
TBLPROPERTIES ("hbase.table.name" = "xyz");
```

Note: The TBLPROPERTIES command is not required, but those new to Hive-HBase integration may find it easier to understand what's going on if Hive and HBase use different names for the same table.

In this example, Hive will recognize this table as "hbase_table_1" and HBase will recognize this table as "xyz".

Hive-hbase integration

```
hive> CREATE TABLE hbase_table_1(key int, value string) STORED BY 'org.apache.hadoop.hive.hbase.HBase
StorageHandler' WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,cf1:val") TBLPROPERTIES ("hbase
.table.name" = "xyz");
OK
Time taken: 2.685 seconds
```

Check the tables in hive using show tables command

```
hive> show tables;
OK
hbase_table_1
pokes
Time taken: 0.068 seconds, Fetched: 2 row(s)
hive>
```

We can see **hbase_table_1**

Hive-hbase integration

Step 6: in hbase check the list of table

list

```
hbase(main):003:0* list
TABLE
census
xyz
2 row(s) in 0.0090 seconds

=> ["census", "xyz"]
```

Note: HBase recognizes the Hive-HBase table named xyz. This is the same table known to Hive as hbase_table_1.

Hive-hbase integration

step 7: From the Hive prompt, insert data from the Hive table pokes into the Hive-HBase table hbase_table_1

INSERT OVERWRITE TABLE hbase_table_1 SELECT * FROM pokes WHERE foo=98;

```
hive> INSERT OVERWRITE TABLE hbase_table_1 SELECT * FROM pokes WHERE foo=98;
Query ID = cloudera_20200526125454_72ec09ab-59c7-4bcc-8582-fce8ee50b71f
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1590135721705_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/applicati
on_1590135721705_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1590135721705_0001
Hadoop job information for Stage-0: number of mappers: 1; number of reducers: 0
2020-05-26 12:55:06,509 Stage-0 map = 0%,  reduce = 0%
2020-05-26 12:55:18,920 Stage-0 map = 100%,  reduce = 0%, Cumulative CPU 2.68 sec
MapReduce Total cumulative CPU time: 2 seconds 680 msec
Ended Job = job_1590135721705_0001
MapReduce Jobs Launched:
Stage-Stage-0: Map: 1   Cumulative CPU: 2.68 sec   HDFS Read: 17508 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 680 msec
OK
Time taken: 31.064 seconds
hive>
```

Hive-hbase integration

step 8: In hive Query hbase_table_1 to see the data we have inserted into the Hive-HBase table:

SELECT * FROM hbase_table_1;

```
hive> SELECT * FROM hbase_table_1;
OK
98      val_98
Time taken: 0.149 seconds, Fetched: 1 row(s)
hive>
```

Note: Note: Even though we loaded two rows from the Hive pokes table that had the same key of 98, only one row was actually inserted into hbase_table_1. This is because hbase_table_1 is an HBASE table, and although Hive tables support duplicate keys, HBase tables only support unique keys. HBase tables arbitrarily retain only one key, and will silently discard all the data associated with duplicate keys.

Hive-hbase integration

Step 9: let's try to see the same data using hbase shell

scan 'xyz'

```
hbase(main):005:0* scan 'xyz'
ROW                                   COLUMN+CELL
 98                                   column=cf1:val, timestamp=1590522917727, value=val_98
1 row(s) in 0.2510 seconds

hbase(main):006:0>
```

We have learnt Hive-Hbase Integration

Happy Learning!!!

Google REVIEWS

5 Star Google Rated
Big Data Course

LEARN FROM THE EXPERT

9108179578

Call for more details

Follow US

Trainer	Mr. Sumit Mittal
Phone	9108179578
Email	trendytech.sumit@gmail.com
Website	https://trendytech.in/courses/big-data-online-training/
LinkedIn	https://www.linkedin.com/in/bigdatabysumit/
Twitter	@BigdataBySumit
Instagram	bigdatabysumit
Facebook	https://www.facebook.com/trendytech.in/
Youtube	TrendyTech