Large Scale Data Ingestion Using Sqoop

Importing Data From MySQL To Hive & HBase Using Sqoop

edureka!

Importing Data From MySQL To Hive And HBase Using Sqoop

In this demo, we will learn how to transfer the data from MySQL (RDBMS) table to Hive and HBase tables using Sqoop.

Following are the steps involved in this:

1. Create a sample CSV file with some records.
2. Create a sample MySQL table to which the data has to be imported from CSV file.
3. Load data from CSV file into that table.
4. Write Sqoop command to import the data from the MySQL table created in Step2 to Hive.
5. Verify if the data is imported to Hive table or not
6. Write Sqoop command to import the data from the MySQL table created in Step2 to HBase
7. Verify if the data is imported to HBase table or not

Now let's implement these steps:

1. Create a sample CSV file with some records.
   (Create a file - ***custs.csv*** that has sample data as shown in the following image.)

```
[edureka_396201@ip-20-0-41-190 ~]$ cat custs.csv
4000001,Kristina,Chung,55,Pilot
4000002,Paige,Chen,74,Teacher
4000003,Sherri,Melton,34,Firefighter
4000004,Gretchen,Hill,66, Hardware Engineer
4000005,Karen,Puckett,74,Lawyer
```

2. Login to MySQL, create a table – ***custdetails*** and schema of this table must be defined as per the following image.

```
MySQL [labuser_database]> create table custdetails(
    -> id int(7),
    -> firstName varchar(50),
    -> lastName varchar(50),
    -> age int(2),
    -> profession varchar(50));
Query OK, 0 rows affected (0.01 sec)
```

3. Load the data from ***custs.csv*** file into ***custdetails*** table.

```
MySQL [labuser_database]> load data local infile 'custs.csv' into
    -> table custdetails fields terminated by ',';
Query OK, 5 rows affected (0.00 sec)
Records: 5  Deleted: 0  Skipped: 0  Warnings: 0
```

```
MySQL [labuser_database]> select * from custdetails;
+---------+-----------+----------+------+--------------------+
| id      | firstName | lastName | age  | profession         |
+---------+-----------+----------+------+--------------------+
| 4000001 | Kristina  | Chung    |   55 | Pilot              |
| 4000002 | Paige     | Chen     |   74 | Teacher            |
| 4000003 | Sherri    | Melton   |   34 | Firefighter        |
| 4000004 | Gretchen  | Hill     |   66 |  Hardware Engineer |
| 4000005 | Karen     | Puckett  |   74 | Lawyer             |
+---------+-----------+----------+------+--------------------+
5 rows in set (0.00 sec)
```

4. Write Sqoop command to import the data from the MySQL table created in Step2 to Hive.

   Following Sqoop command imports the data from MySQL table – ***custdetails*** to Hive table – ***cust_details***.
   The argument ***--create-hive-table*** will create a hive table based on the schema of the MySQL table

   Command: sqoop import --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser_database \
   --username edu_labuser \
   --password edureka \
   --table custdetails \
   --hive-import \
   --create-hive-table \
   --hive-table custs_details \
   --fields-terminated-by '\t' \
   -m 1

```
[edureka_396201@ip-20-0-41-190 ~]$ sqoop import  --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser_database \
> --username edu_labuser \
> --password edureka \
> --table custdetails \
> --hive-import \
> --create-hive-table \
> --hive-table custs_details \
> --fields-terminated-by '\t' \
> -m 1
```

```
19/12/27 07:06:40 INFO mapreduce.ImportJobBase: Transferred 175 bytes in 73.3558 seconds (2.3856 bytes/sec)
19/12/27 07:06:40 INFO mapreduce.ImportJobBase: Retrieved 5 records.
19/12/27 07:06:40 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `custdetails` AS t LIMIT 1
19/12/27 07:06:40 INFO hive.HiveImport: Loading uploaded data into Hive

Logging initialized using configuration in jar:file:/opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1.p0.4/jars/hive-common-1.1.0-cdh5.11.1.jar!/hive-log
4j.properties
OK
Time taken: 2.972 seconds
Loading data to table default.custs_details
Table default.custs_details stats: [numFiles=1, totalSize=175]
OK
Time taken: 0.615 seconds
```

Highlighted part indicates that the data has been successfully imported.

5. Verify if the data is imported to Hive table or not

   Login to Hive and verify if the data is imported by running a select query

```
[edureka_396201@ip-20-0-41-190 ~]$ hive
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=512M; support
 was removed in 8.0
Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated and w
ill likely be removed in a future release
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=512M; support
 was removed in 8.0

Logging initialized using configuration in jar:file:/opt/cloudera/parcels/CDH-5.11.1
-1.cdh5.11.1.p0.4/jars/hive-common-1.1.0-cdh5.11.1.jar!/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> select * from custs_details;
OK
4000001 Kristina        Chung   55      Pilot
4000002 Paige   Chen    74      Teacher
4000003 Sherri  Melton  34      Firefighter
4000004 Gretchen        Hill    66      Hardware Engineer
4000005 Karen   Puckett 74      Lawyer
Time taken: 1.892 seconds, Fetched: 5 row(s)
```

6. Write Sqoop command to import the data from the MySQL table created in Step1 to HBase.

   Similarly let's now import the MySQL data into HBase using Sqoop.
   Following Sqoop command imports the data from MySQL table – ***custdetails*** to HBase table – ***customer_hbase***.
   The argument ***--hbase-create-table*** will create an HBase table based on the schema of the MySQL table.

   Command: sqoop import --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser_database \
   --username edu_labuser \
   --password edureka \
   --table custdetails \
   --hbase-table customer_hbase \
   --column-family info \

```
--hbase-row-key id \
--hbase-create-table \
-m 1
```

```
[edureka_396201@ip-20-0-32-225 ~]$ sqoop import --connect  jdbc:mysql://dbserver.edu.cloudlab.com/labuser_database \
> --username edu_labuser \
> --password edureka \
> --table custdetails \
> --hbase-table customer_hbase \
> --column-family info \
> --hbase-row-key id \
> --hbase-create-table \
> -m 1
```

```
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
20/01/17 06:56:35 INFO mapreduce.ImportJobBase: Transferred 0 bytes in 44.1161 seconds (0 bytes/sec)
20/01/17 06:56:35 INFO mapreduce.ImportJobBase: Retrieved 6 records.
```

Highlighted part indicates that the data has been successfully imported into HBase.

7. Verify if the data is imported to HBase table or not

Login to HBase, verify if the data is imported by running scan command

```
[edureka_396201@ip-20-0-32-225 ~]$ hbase shell
Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated and will likely be removed in a future release
20/01/17 07:07:44 INFO Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.11.1, rUnknown, Thu Jun  1 10:19:43 PDT 2017

hbase(main):001:0> scan 'customer_hbase'
ROW                                    COLUMN+CELL
 4000001                               column=info:age, timestamp=1579244784915, value=55
 4000001                               column=info:firstName, timestamp=1579244784915, value=Kristina
 4000001                               column=info:lastName, timestamp=1579244784915, value=Chung
 4000001                               column=info:profession, timestamp=1579244784915, value=Pilot
 4000002                               column=info:age, timestamp=1579244784915, value=74
 4000002                               column=info:firstName, timestamp=1579244784915, value=Paige
 4000002                               column=info:lastName, timestamp=1579244784915, value=Chen
 4000002                               column=info:profession, timestamp=1579244784915, value=Teacher
 4000003                               column=info:age, timestamp=1579244784915, value=34
 4000003                               column=info:firstName, timestamp=1579244784915, value=Sherri
 4000003                               column=info:lastName, timestamp=1579244784915, value=Melton
 4000003                               column=info:profession, timestamp=1579244784915, value=Firefighter
 4000004                               column=info:age, timestamp=1579244784915, value=66
 4000004                               column=info:firstName, timestamp=1579244784915, value=Gretchen
 4000004                               column=info:lastName, timestamp=1579244784915, value=Hill
 4000004                               column=info:profession, timestamp=1579244784915, value= Hardware Engineer
 4000005                               column=info:age, timestamp=1579244784915, value=74
 4000005                               column=info:firstName, timestamp=1579244784915, value=Karen
 4000005                               column=info:lastName, timestamp=1579244784915, value=Puckett
 4000005                               column=info:profession, timestamp=1579244784915, value=Lawyer
 4000006                               column=info:age, timestamp=1579244784915, value=42
 4000006                               column=info:firstName, timestamp=1579244784915, value=Patrick
 4000006                               column=info:lastName, timestamp=1579244784915, value=Song
 4000006                               column=info:profession, timestamp=1579244784915, value=Teacher
6 row(s) in 0.2240 seconds
```

We have successfully imported the data from MySQL table to Hive and HBase tables using Sqoop 😊