EXPERIMENT 3

Using SqoopTool To Transfer Data Between Hadoop And MySQL

THEORY

The traditional application management system, that is, the interaction of applications with relational databases using RDBMS, is one of the sources that generate Big Data. Such Big Data, generated by RDBMS, is stored in Relational Database Servers in the relational database structure.

When Big Data storage and analyzers such as MapReduce, Hive, HBase, Cassandra, Pig, etc. of the Hadoop ecosystem came into the picture, they required a tool to interact with the relational database servers for importing and exporting the Big Data residing in them. Here, Sqoop occupies a place in the Hadoop ecosystem to provide feasible interaction between the relational database server and Hadoop's HDFS.

Sqoop − "SQL to Hadoop and Hadoop to SQL"

Sqoop is a tool designed to transfer data between Hadoop and relational database servers. It is used to import data from relational databases such as MySQL, and Oracle to Hadoop HDFS, and export from the Hadoop file system to relational databases. It is provided by the Apache Software Foundation.

Sqoop Import

The import tool imports individual tables from RDBMS to HDFS. Each row in a table is treated as a record in HDFS. All records are stored as text data in text files or as binary data in Avro and Sequence files.

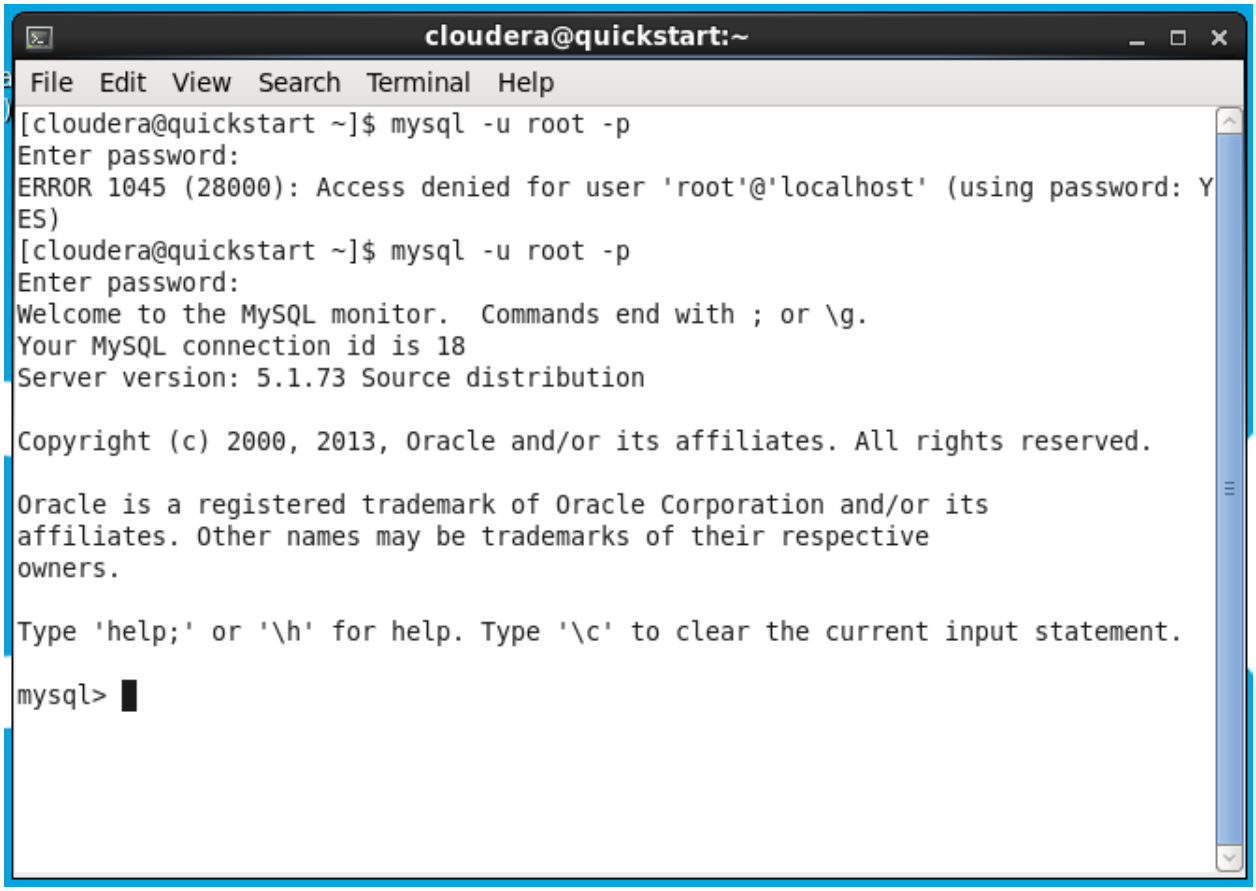
Sqoop Export

The export tool exports a set of files from HDFS back to an RDBMS. The files given as input to Sqoop contain records, which are called rows in the table. Those are read and parsed into a set of records and delimited with a user-specified delimiter.

STEP 1

Open the Cloudera Terminal and execute the following command in order to start the MySQL server. (Note: The default password is cloudera for the root user)

mysql -u root -p



STEP 2

Creating a database
create database bank;

```
mysql> create database bank;
Query OK, 1 row affected (0.00 sec)

mysql>
```

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| bank               |
| cm                 |
| firehose           |
| hue                |
| metastore          |
| mysql              |
| nav                |
| navms              |
| oozie              |
| retail_db          |
| rman               |
| sentry             |
+--------------------+
13 rows in set (0.02 sec)
```

STEP 3

Creating a Table
(Note: The database must be in use before you create a table.)

```
mysql> use bank;
Database changed
mysql>
```

```
mysql> create table register(accno varchar(20), name varchar(20), number varchar
(10), email varchar(30), password varchar(10), age varchar(3));
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> describe register;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| accno    | varchar(20) | YES  |     | NULL    |       |
| name     | varchar(20) | YES  |     | NULL    |       |
| number   | varchar(10) | YES  |     | NULL    |       |
| email    | varchar(30) | YES  |     | NULL    |       |
| password | varchar(10) | YES  |     | NULL    |       |
| age      | varchar(3)  | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
6 rows in set (0.03 sec)
```

STEP 4

Insert values

```
mysql> insert into register values ("1","raj","11","raj@gmail.com","raj123","11"
);
Query OK, 1 row affected (0.08 sec)

mysql> insert into register values ("2","tanay","12","tanay@gmail.com","tanay123
","12");
Query OK, 1 row affected (0.05 sec)

mysql> insert into register values ("3","sid","13","sid@gmail.com","sid123","13"
);
Query OK, 1 row affected (0.10 sec)

mysql> insert into register values ("4","raunak","14","raunak@gmail.com","raunak
123","14");
Query OK, 1 row affected (0.05 sec)

mysql>
```

```
mysql> select * from register;
+-------+--------+--------+------------------+-----------+------+
| accno | name   | number | email            | password  | age  |
+-------+--------+--------+------------------+-----------+------+
| 1     | raj    | 11     | raj@gmail.com    | raj123    | 11   |
| 2     | tanay  | 12     | tanay@gmail.com  | tanay123  | 12   |
| 3     | sid    | 13     | sid@gmail.com    | sid123    | 13   |
| 4     | raunak | 14     | raunak@gmail.com | raunak123 | 14   |
+-------+--------+--------+------------------+-----------+------+
4 rows in set (0.01 sec)

mysql> exit;
Bye
[cloudera@quickstart ~]$
```

CLOUDERA

After you exit MySQL, create a folder in the Cloudera file system to import the above MySQL table which was created.
(In the following steps,'myfirstdata' folder is created in /home/cloudera)

STEP 5

Importing the table using Sqoop
sqoop import connect jdbc:mysql://youripaddress:3306/<database_name> --username root --password cloudera —table <table_name> --targetdir=<target_directory> -m 1

Here,
-m specifies the number of mappers
3306 is the default port for MySQL

In our case:

[cloudera@quickstart ~]$ sqoop import --connect jdbc:mysql://localhost:3306/bank--username root --password cloudera --table register--target-dir=/home/cloudera/myfirstdata -m 1

```
22/08/26 23:37:02 INFO mapreduce.Job: The url to track the job: http://quickstar
t.cloudera:8088/proxy/application_1661580441152_0001/
22/08/26 23:37:02 INFO mapreduce.Job: Running job: job_1661580441152_0001
22/08/26 23:37:21 INFO mapreduce.Job: Job job_1661580441152_0001 running in uber
 mode : false
22/08/26 23:37:21 INFO mapreduce.Job:  map 0% reduce 0%
22/08/26 23:37:33 INFO mapreduce.Job:  map 100% reduce 0%
22/08/26 23:37:34 INFO mapreduce.Job: Job job_1661580441152_0001 completed succe
ssfully
22/08/26 23:37:34 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=151445
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=87
                HDFS: Number of bytes written=147
                HDFS: Number of read operations=4
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Other local map tasks=1
                Total time spent by all maps in occupied slots (ms)=7999
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=7999
                Total vcore-milliseconds taken by all map tasks=7999
                Total megabyte-milliseconds taken by all map tasks=8190976
        Map-Reduce Framework
                Map input records=4
                Map output records=4
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=31
                CPU time spent (ms)=1660
                Physical memory (bytes) snapshot=180097024
                Virtual memory (bytes) snapshot=1569824768
                Total committed heap usage (bytes)=174587904
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=147
22/08/26 23:37:34 INFO mapreduce.ImportJobBase: Transferred 147 bytes in 39.6203
 seconds (3.7102 bytes/sec)
22/08/26 23:37:34 INFO mapreduce.ImportJobBase: Retrieved 4 records.
[cloudera@quickstart ~]$
```

STEP 6

Displaying the contents in HDFS

hadoop fs -ls /home/cloudera/myfirstdata

hadoop fs -cat /home/cloudera/myfirstdata/part-m-00000

```
[cloudera@quickstart ~]$ hadoop fs -ls /home/cloudera/myfirstdata
Found 2 items
-rw-r--r--   1 cloudera supergroup          0 2022-08-26 23:37 /home/cloudera/my
firstdata/_SUCCESS
-rw-r--r--   1 cloudera supergroup        147 2022-08-26 23:37 /home/cloudera/my
firstdata/part-m-00000
```

```
[cloudera@quickstart ~]$ hadoop fs -cat /home/cloudera/myfirstdata/part-m-00000
1,raj,11,raj@gmail.com,raj123,11
2,tanay,12,tanay@gmail.com,tanay123,12
3,sid,13,sid@gmail.com,sid123,13
4,raunak,14,raunak@gmail.com,raunak123,14
[cloudera@quickstart ~]$
```

Export Data from HDFS to MySQL

In order to export data from HDFS to MySQL, an appropriate table has to be created in MySQL as we export data into a particular table. In our case, we will be exporting the contents in the **'myfirstdata'** folder by creating a table **'registercopy'** in the **'bank'** database. The table which we will be creating needs to have the same structure as the **'register'** table which we created earlier.

STEP 7

Creating the table

```
[cloudera@quickstart ~]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 23
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use bank;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create table registercopy(accno varchar(20), name varchar(20), number var
char(10), email varchar(30), password varchar(10), age varchar(3));
Query OK, 0 rows affected (0.01 sec)

mysql> describe registercopy;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| accno    | varchar(20) | YES  |     | NULL    |       |
| name     | varchar(20) | YES  |     | NULL    |       |
| number   | varchar(10) | YES  |     | NULL    |       |
| email    | varchar(30) | YES  |     | NULL    |       |
| password | varchar(10) | YES  |     | NULL    |       |
| age      | varchar(3)  | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> exit;
Bye
[cloudera@quickstart ~]$ 
```

STEP 8

Exporting data from HDFS to MySQL

Syntax:
sqoop export --connect jdbc:mysql://localhost/db --username root --table <table_name> --export-dir <directory>

In our case,

[cloudera@quickstart ~]$ sqoop export --connect jdbc:mysql://localhost/bank--username root --password cloudera --table registercopy --export-dir /home/cloudera/myfirstdata

```
22/08/26 23:47:04 INFO Configuration.deprecation: mapred.map.tasks.speculative.execution is deprecated. Instead, use mapreduce.map.speculative
22/08/26 23:47:04 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1661580441152_0002
22/08/26 23:47:05 INFO impl.YarnClientImpl: Submitted application application_1661580441152_0002
22/08/26 23:47:05 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1661580441152_0002/
22/08/26 23:47:05 INFO mapreduce.Job: Running job: job_1661580441152_0002
22/08/26 23:47:17 INFO mapreduce.Job: Job job_1661580441152_0002 running in uber mode : false
22/08/26 23:47:17 INFO mapreduce.Job:  map 0% reduce 0%
22/08/26 23:47:31 INFO mapreduce.Job:  map 25% reduce 0%
22/08/26 23:47:36 INFO mapreduce.Job:  map 100% reduce 0%
22/08/26 23:47:37 INFO mapreduce.Job: Job job_1661580441152_0002 completed successfully
22/08/26 23:47:37 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=604992
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1098
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=19
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
        Job Counters
                Launched map tasks=4
                Data-local map tasks=4
                Total time spent by all maps in occupied slots (ms)=56748
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=56748
                Total vcore-milliseconds taken by all map tasks=56748
                Total megabyte-milliseconds taken by all map tasks=58109952
        Map-Reduce Framework
                Map input records=4
                Map output records=4
                Input split bytes=691
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=343
                CPU time spent (ms)=4470
                Physical memory (bytes) snapshot=692604928
                Virtual memory (bytes) snapshot=6284853248
                Total committed heap usage (bytes)=699924480
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
22/08/26 23:47:38 INFO mapreduce.ExportJobBase: Transferred 1.0723 KB in 36.4458 seconds (30.1269 bytes/sec)
22/08/26 23:47:38 INFO mapreduce.ExportJobBase: Exported 4 records.
```

STEP 9

Verifying in MySQL

We can see that the data is exported successfully into ‘registercopy’

```
[cloudera@quickstart ~]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use bank;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from registercopy;
+-------+--------+--------+------------------+-----------+------+
| accno | name   | number | email            | password  | age  |
+-------+--------+--------+------------------+-----------+------+
| 3     | sid    | 13     | sid@gmail.com    | sid123    | 13   |
| 4     | raunak | 14     | raunak@gmail.com | raunak123 | 14   |
| 1     | raj    | 11     | raj@gmail.com    | raj123    | 11   |
| 2     | tanay  | 12     | tanay@gmail.com  | tanay123  | 12   |
+-------+--------+--------+------------------+-----------+------+
4 rows in set (0.00 sec)

mysql> 
```