

Data Ingestion from the RDS to HDFS using Sqoop

Before running Sqoop import command we have installed & Configure MySQL Connector.

Sqoop Import command used for importing table from RDS to HDFS:

```
sqoop import \
--connect jdbc:mysql://upgraddetest.cyaieic9bmnf.us-east-1.rds.amazonaws.com/testdatabase \
--table SRC_ATM_TRANS \
--username student --password STUDENT123 \
--target-dir /user/root/ETL_Spar_ATM \
-m 1
```

Screenshot of Sqoop import command Execution:

```
[hadoop@ip-172-31-81-136 ~]$ sudo -i

EEEEEEEEEEEEEEEEEEEE MMMMMMM          MMMMMMM RRRRRRRRRRRRRR
E::::::::::::::::::::E M::::::::M          M::::::::M R::::::::::::R
EE::::::::EEEEEEEE::::E M::::::::M          M::::::::M R::::RRRRRR::::R
  E::::E          EEEEE M::::::::M          M::::::::M RR::::R          R::::R
  E::::E          M::::M:M::M          M::M::::M          R::R          R::::R
  E::::EEEEEEEEEE M::::M M::M M::M M::::M          R::RRRRRR::::R
  E::::::::::E M::::M M::M:M::M M::::M          R::::::::::::RR
  E::::EEEEEEEEEE M::::M M::::M M::::M          R::RRRRRR::::R
  E::::E          M::::M M::M M::::M          R::R          R::::R
  E::::E          EEEEE M::::M          M M::::M          R::R          R::::R
EE::::::::EEEEEEEE::::E M::::M          M::::M          R::R          R::::R
E::::::::::::::::::::E M::::M          M::::M RR::::R          R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM          MMMMMMM RRRRRRR          RRRRRR

[root@ip-172-31-81-136 ~]# sqoop import \
> --connect jdbc:mysql://upgraddetest.cyaieic9bmnf.us-east-1.rds.amazonaws.com/testdatabase \
> --table SRC_ATM_TRANS \
> --username student --password STUDENT123 \
> --target-dir /user/root/ETL_Spar_ATM \
> -m 1
```

Screenshot of success of above import command:

```
22/12/12 12:02:30 INFO mapreduce.Job: Job job_1670845557400_0001 completed successfully
22/12/12 12:02:30 INFO mapreduce.Job: Counters: 30
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=189411
    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=531214815
    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)=1199808
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=24996
    Total vcore-milliseconds taken by all map tasks=24996
    Total megabyte-milliseconds taken by all map tasks=38393856
  Map-Reduce Framework
    Map input records=2468572
    Map output records=2468572
    Input split bytes=87
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=230
    CPU time spent (ms)=28160
    Physical memory (bytes) snapshot=616722432
    Virtual memory (bytes) snapshot=3291242496
    Total committed heap usage (bytes)=537395200
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=531214815
22/12/12 12:02:30 INFO mapreduce.ImportJobBase: Transferred 506.6059 MB in 45.0473 seconds (11.2461 MB/sec)
22/12/12 12:02:30 INFO mapreduce.ImportJobBase: Retrieved 2468572 records.
[root@ip-172-31-81-136 ~]#
```

In above screenshot we can see that total 246572 records have been imported.

Command used to see the list of imported data in HDFS:

```
hadoop fs -ls /user/root/ETL_Spar_ATM
```

```
[root@ip-172-31-81-136 ~]# hadoop fs -ls /user/root/ETL_Spar_ATM
Found 2 items
-rw-r--r--  1 root hadoop          0 2022-12-12 12:02 /user/root/ETL_Spar_ATM/_SUCCESS
-rw-r--r--  1 root hadoop 531214815 2022-12-12 12:02 /user/root/ETL_Spar_ATM/part-m-00000
[root@ip-172-31-81-136 ~]#
```

In the screenshot above we can see two items:

- The first file is the success file, indicating that the MapReduce job was successful.
- The second file 'part-m-00000' is the one that we imported. Since we used only one mapper in my import command, thus the data is in a single file.

Screenshot of the imported data:

hadoop fs -cat /user/root/ETL_Spar_ATM/part-m-00000 |head -10

```
[root@ip-172-31-81-136 ~]# hadoop fs -cat /user/root/ETL_Spar_ATM/part-m-00000 |head -10
2017,January,1,Sunday,0,Active,1,NCR,NÅfÅ|stved,Farimagssvej,8,4700,55.233,11.763,DKK,MasterCard,5643,Withdrawal,,,55.
2017,January,1,Sunday,0,Inactive,2,NCR,Vejgaard,Hadsundvej,20,9000,57.043,9.950,DKK,MasterCard,1764,Withdrawal,,,57.0
2017,January,1,Sunday,0,Inactive,2,NCR,Vejgaard,Hadsundvej,20,9000,57.043,9.950,DKK,VISA,1891,Withdrawal,,,57.048,9.9
2017,January,1,Sunday,0,Inactive,3,NCR,Ikast,RÅfÅVdhusstrÅfÅ|det,12,7430,56.139,9.154,DKK,VISA,4166,Withdrawal,,,56.1
2017,January,1,Sunday,0,Active,4,NCR,Svogerslev,BrÅfÅnsager,1,4000,55.634,12.018,DKK,MasterCard,5153,Withdrawal,,,58
2017,January,1,Sunday,0,Active,5,NCR,Nibe,Torvet,1,9240,56.983,9.639,DKK,MasterCard,3269,Withdrawal,,,56.981,9.639,26
2017,January,1,Sunday,0,Active,6,NCR,Fredericia,SjÅfÅ|llandsgade,33,7000,55.564,9.757,DKK,MasterCard,887,Withdrawal,,
2017,January,1,Sunday,0,Active,7,Diebold Nixdorf,Hjallerup,Hjallerup Centret,18,9320,57.168,10.148,DKK,Mastercard - c
2017,January,1,Sunday,0,Active,8,NCR,GlyngÅfÅre,FÅfÅrgevej,1,7870,56.762,8.867,DKK,MasterCard,470,Withdrawal,,,56.7
2017,January,1,Sunday,0,Active,9,Diebold Nixdorf,Hadsund,Storegade,12,9560,56.716,10.114,DKK,VISA,8473,Withdrawal,,,5
cat: Unable to write to output stream.
[root@ip-172-31-81-136 ~]#
```

With above command we are checking 10 rows of imported data.

Now below location and file we will use in PySpark for data transformation:

/user/root/ETL_Spar_ATM/part-m-00000