

TASK 2

Task 2. Use Sqoop command to ingest the data from RDS into the HBase Table.

The following steps were followed for data ingestion from RDS into HBase table

- 1) Load the data to RDS instance

Following screen short for reference from previous step (i.e. Task 1)

```
MySQL [yellow_taxi]> select count(*) from taxi;
+-----+
| count(*) |
+-----+
| 18880595 |
+-----+
1 row in set (1 min 2.43 sec)
```

[illegible]

- 2) Exit form RDS and load the table data in to the hbase.

Created the hbase table : hbase taxi, set the column-family as cf. Copied the data from the RDS taxi table

Sqoop import command:

```
sqoop import \
```

```
--connect "jdbc:mysql://casestudy.ci2jpset7w3y.us-east-1.rds.amazonaws.com:3306/yellow_taxi" \
```

```
--username root \
```

```
--password 123456789 \
```

```
--table taxi \
```

--columns

"vendorID,tpep_pickup_datetime,tpep_dropoff_datetime,passenger_count,trip_distance,puLocationID,doLoc

TASK 2

```

ationID,rateCodeID,store_and_fwd_flag,payment_type,fare_amount,extra,mta_tax,improvement_surcharge,t
ip_amount,tolls_amount,total_amount,congestion_Surcharge,airport_fee" \
--hbase-create-table \
--hbase-table hbasetaxi \
--column-family trip_details \
--hbase-row-key "vendorID,tpep_pickup_datetime,tpep_dropoff_datetime" \
--split-by tpep_dropoff_datetime \
-m 8

```

Screenshots for reference:

```

at org.apache.hadoop.mapreduce.db.DatabaseReader.nextKeyValue(DatabaseReader.java:277)
at org.apache.hadoop.mapred.MapTask$MapReduceTrackerRecordReader.nextKeyValue(MapTask.java:565)
at org.apache.hadoop.mapreduce.task.MapContextImpl.nextKeyValue(MapContextImpl.java:80)
at org.apache.hadoop.mapreduce.lib.map.MapMapper.map(ContextMapper.java:91)
at org.apache.hadoop.mapreduce.Mapper.run(Mapper.java:145)
at org.apache.hadoop.mapreduce.AutoProgressMapper.run(AutoProgressMapper.java:64)
at org.apache.hadoop.mapred.MapTask.run(MapTaskRunner$MapTask.java:795)
at org.apache.hadoop.mapred.MapTask.run(MapTask.java:342)
at org.apache.hadoop.mapred.YarnChild$1.run(YarnChild.java:175)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:422)
at org.apache.hadoop.mapreduce.SecurityGroupInformation.doAs(GroupInformation.java:1844)
at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:169)
used by: java.sql.SQLException: ConnectionException: (conn=169) Server has closed the connection.
at org.mariadb.jdbc.internal.util.exceptions.ConnectionMapper.get(ConnectionMapper.java:241)
at org.mariadb.jdbc.internal.util.exceptions.ConnectionMapper.getConnection(ExceptionMapper.java:164)
at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSetHandler.handleConnection(SelectResultSetHandler.java:315)
at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSetSet.next(SelectResultSetSet.java:595)
at org.apache.hadoop.mapreduce.db.DatabaseReader.nextKeyValue(DatabaseReader.java:237)
... 12 more
used by: java.sql.SQLException: Server has closed the connection.
cause check read_timeout/set_write_timeout/wait_timeout server variables. If result set contain huge amount of data, Server expects client to read off the result set relatively fast
this error can occur if you are using a large result set without setting fetch_size / processing your result set faster (check Streaming result sets documentation for more information)
at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSetHandler.handleConnection(SelectResultSetHandler.java:323)
... 10 more
used by: java.io.IOException: unexpected end of stream, read 48 bytes from 93 (socket was closed by server)
at org.mariadb.jdbc.internal.io.input.StandardPacketInputStream.getBytesFromStream(GetPacketArray(StandardPacketInputStream.java:271)
at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSetHandler.getNextValue(SelectResultSetHandler.java:370)
at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSetHandler.getNextResultValue(SelectResultSetHandler.java:384)
at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSetHandler.nextStreamingValue(SelectResultSetHandler.java:344)
at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSetSet.next(SelectResultSetSet.java:592)
... 13 more
10/02 18:04:37 INFO mapreduce.Job: map 0% reduce 0%
10/02 18:05:34 INFO mapreduce.Job: map 25% reduce 0%
10/02 18:14:39 INFO mapreduce.Job: map 0% reduce 0%
10/02 18:15:56 INFO mapreduce.Job: map 25% reduce 0%
10/02 18:24:39 INFO mapreduce.Job: map 0% reduce 0%
10/02 18:25:34 INFO mapreduce.Job: map 25% reduce 0%
10/02 18:34:40 INFO mapreduce.Job: map 0% reduce 0%
10/02 18:35:58 INFO mapreduce.Job: map 25% reduce 0%
10/02 18:44:37 INFO mapreduce.Job: map 0% reduce 0%
10/02 18:45:53 INFO mapreduce.Job: map 25% reduce 0%
```

TASK 2

After the mapreduce

Lets check for the hbase taxi by running

Scan 'hbasetaxi'

In hbase shell

```
ROW                                COLUMN+CELL
1                                  column=Trip_details:Airport fee, timestamp=1678600360710, value=0.0
1                                  column=Trip_details:DOLocationID, timestamp=1678600360710, value=234
1                                  column=Trip_details:FULocationID, timestamp=1678600360710, value=211
1                                  column=Trip_details:RatecodeID, timestamp=1678600360710, value=1
1                                  column=Trip_details:extra, timestamp=1678600360710, value=0.0
1                                  column=Trip_details:fare_amount, timestamp=1678600360710, value=10.0
1                                  column=Trip_details:improvement surcharge, timestamp=1678600360710, value=0.3
1                                  column=Trip_details:mta_tax, timestamp=1678600360710, value=0.5
1                                  column=Trip_details:passenger count, timestamp=1678600360710, value=1
1                                  column=Trip_details:payment type, timestamp=1678600360710, value=2
1                                  column=Trip_details:store_and_fwd flag, timestamp=1678600360710, value=N
1                                  column=Trip_details:tip amount, timestamp=1678600360710, value=1.15
1                                  column=Trip_details:tolls amount, timestamp=1678600360710, value=0.0
1                                  column=Trip_details:total amount, timestamp=1678600360710, value=10.8
1                                  column=Trip_details:tpep_dropoff_datetime, timestamp=1678600360710, value=2017-01-10 15:11:08.0
1                                  column=Trip_details:tpep_pickup_datetime, timestamp=1678600360710, value=2017-01-10 14:34:55.0
1                                  column=Trip_details:trip_distance, timestamp=1678600360710, value=2.1
2                                  column=Trip_details:Airport fee, timestamp=1678600360716, value=0.0
2                                  column=Trip_details:DOLocationID, timestamp=1678600360716, value=230
2                                  column=Trip_details:FULocationID, timestamp=1678600360716, value=42
2                                  column=Trip_details:RatecodeID, timestamp=1678600360716, value=1
2                                  column=Trip_details:extra, timestamp=1678600360716, value=0.0
2                                  column=Trip_details:fare_amount, timestamp=1678600360716, value=10.5
2                                  column=Trip_details:improvement surcharge, timestamp=1678600360716, value=0.3
2                                  column=Trip_details:mta_tax, timestamp=1678600360716, value=0.5
2                                  column=Trip_details:passenger count, timestamp=1678600360716, value=1
2                                  column=Trip_details:payment type, timestamp=1678600360716, value=1
2                                  column=Trip_details:store_and_fwd flag, timestamp=1678600360716, value=N
2                                  column=Trip_details:tip amount, timestamp=1678600360716, value=0.0
2                                  column=Trip_details:tolls amount, timestamp=1678600360716, value=0.0
2                                  column=Trip_details:total amount, timestamp=1678600360716, value=11.3
2                                  column=Trip_details:tpep_dropoff_datetime, timestamp=1678600360716, value=2017-01-08 16:03:27.0
2                                  column=Trip_details:tpep_pickup_datetime, timestamp=1678600360716, value=2017-01-08 15:51:18.0
2                                  column=Trip_details:trip_distance, timestamp=1678600360716, value=2.26
2 row(s) in 0.1510 seconds
hbase(main):004:0>
```