Script Execution: From NoSQL / AWS RDS to Look-up Table

1. Loading card_transactions.csv into MongoDB

Here we are importing the card_transactions.csv file from amazon s3 to mongodb.

```
~
~
~
[hadoop@ip-172-31-74-8 ~]$ aws s3 cp s3://creditcardcapstone/card_transactions.csv - | mongoimport --db transaction_db --collection card_transactions --type csv --headerline
2024-02-08T07:06:19.429+0000    connected to: mongodb://localhost/
2024-02-08T07:06:20.574+0000    53292 document(s) imported successfully. 0 document(s) failed to import.
[hadoop@ip-172-31-74-8 ~]$
```

```
[hadoop@ip-172-31-74-8 ~]$ sudo vim /etc/yum.repos.d/mongodb-org-7.0.repo
[hadoop@ip-172-31-74-8 ~]$ sudo yum install -y mongodb-org
MongoDB Repository                                                                          170 kB/s |  24 kB     00:00
Dependencies resolved.
=========================================================================================================================
 Package                            Architecture     Version                        Repository                     Size
=========================================================================================================================
Installing:
 mongodb-org                        x86_64           7.0.5-1.el9                    mongodb-org-7.0                9.3 k
Installing dependencies:
 cyrus-sasl                         x86_64           2.1.27-18.amzn2023.0.3         amazonlinux                     73 k
 mongodb-database-tools             x86_64           100.9.4-1                      mongodb-org-7.0                 28 M
 mongodb-mongosh                    x86_64           2.1.3-1.el8                    mongodb-org-7.0                 51 M
 mongodb-org-database               x86_64           7.0.5-1.el9                    mongodb-org-7.0                9.4 k
 mongodb-org-database-tools-extra   x86_64           7.0.5-1.el9                    mongodb-org-7.0                 14 k
 mongodb-org-mongos                 x86_64           7.0.5-1.el9                    mongodb-org-7.0                 24 M
 mongodb-org-server                 x86_64           7.0.5-1.el9                    mongodb-org-7.0                 35 M
 mongodb-org-tools                  x86_64           7.0.5-1.el9                    mongodb-org-7.0                9.3 k

Transaction Summary
=========================================================================================================================
Install  9 Packages

Total download size: 137 M
Installed size: 604 M
Downloading Packages:
(1/9): cyrus-sasl-2.1.27-18.amzn2023.0.3.x86_64.rpm                                         779 kB/s |  73 kB     00:00
(2/9): mongodb-org-7.0.5-1.el9.x86_64.rpm                                                   183 kB/s | 9.3 kB     00:00
(3/9): mongodb-org-database-7.0.5-1.el9.x86_64.rpm                                          198 kB/s | 9.4 kB     00:00
(4/9): mongodb-org-database-tools-extra-7.0.5-1.el9.x86_64.rpm                              286 kB/s |  14 kB     00:00
(5/9): mongodb-mongosh-2.1.3.x86_64.rpm                                                      40 MB/s |  51 MB     00:01
(6/9): mongodb-database-tools-100.9.4.x86_64.rpm                                             19 MB/s |  28 MB     00:01
(7/9): mongodb-org-mongos-7.0.5-1.el9.x86_64.rpm                                             18 MB/s |  24 MB     00:01
(8/9): mongodb-org-tools-7.0.5-1.el9.x86_64.rpm                                              55 kB/s | 9.3 kB     00:00
(9/9): mongodb-org-server-7.0.5-1.el9.x86_64.rpm                                             43 MB/s |  35 MB     00:00
-------------------------------------------------------------------------------------------------------------------------
Total                                                                                         64 MB/s | 137 MB     00:02
MongoDB Repository                                                                            21 kB/s | 1.6 kB     00:00
Importing GPG key 0x1785BA38:
 Userid     : "MongoDB 7.0 Release Signing Key <packaging@mongodb.com>"
 Fingerprint: E588 3020 1F7D D82C D808 AA84 160D 26BB 1785 BA38
 From       : https://pgp.mongodb.com/server-7.0.asc
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                1/1
  Installing       : mongodb-org-database-tools-extra-7.0.5-1.el9.x86_64                                            1/9
  Running scriptlet: mongodb-org-server-7.0.5-1.el9.x86_64                                                          2/9
  Installing       : mongodb-org-server-7.0.5-1.el9.x86_64                                                          2/9
```

2. Ingest RDBMS - Load card_member.csv and member_score.csvinto Spark data frame and table.

-> We have loaded the card_member.csv and member_score.csv from amazon s3

```
>>> crd=spark.read.csv("s3://creditcardcapstone/card_member.csv",header=True)
>>> crd.createOrReplaceTempView('card_mem')
>>> mem=spark.read.csv("s3://creditcardcapstone/member_score.csv",header=True)
>>> mem.createOrReplaceTempView('mem_score')
```

-> Schema for card_member.csv & member_score.csv

```
>>> mem.printSchema()
root
 |-- member_id: string (nullable = true)
 |-- score: double (nullable = true)

>>> crd.printSchema()
root
 |-- card_id: string (nullable = true)
 |-- member_id: string (nullable = true)
 |-- member_joining_dt: string (nullable = true)
 |-- card_purchase_dt: string (nullable = true)
 |-- country: string (nullable = true)
 |-- city: string (nullable = true)

>>>
```

3. Creation of Look-up table

```
>>> spark.sql("CREATE TABLE lookup_table (card_id STRING, ucl DOUBLE, postcode STRING, transaction_dt STRING, score INT)")
24/02/08 10:11:53 WARN ResolveSessionCatalog: A Hive serde table will be created as there is no table provider specified. You can set spark.sql.legacy.createHiveTableByDefault to false so t
hat native data source table will be created instead.
24/02/08 10:11:53 WARN HiveConf: HiveConf of name hive.server2.thrift.url does not exist
24/02/08 10:11:54 WARN SessionState: METASTORE_FILTER_HOOK will be ignored, since hive.security.authorization.manager is set to instance of HiveAuthorizerFactory.
DataFrame[]
>>> spark.sql('select * from lookup_table')
DataFrame[card_id: string, ucl: double, postcode: string, transaction_dt: string, score: int]
>>>
```

4. Load data into Look-up table

- Take last 10 transactions for each card

```
>>> df_10trans = spark.sql("\
... SELECT card_id, amount, postcode, transaction_dt, status, rn \
... from(\
... SELECT card_id, amount, postcode, transaction_dt, status, ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY \
... unix_timestamp(transaction_dt,'dd-MM-yyyy hh:mm:ss') DESC) AS rn \
... FROM transact_table \
... WHERE status = 'GENUINE') a \
... WHERE a.rn <= 10")
>>> df_10trans.createOrReplaceTempView('table_10transact')
```

```
>>> spark.sql('SELECT * FROM table_10transact LIMIT 20').show();
+----------------+------+--------+-------------------+------+---+
|         card_id|amount|postcode|     transaction_dt|status| rn|
+----------------+------+--------+-------------------+------+---+
|340028465709212|8696557|   24658|02-01-2018 03:25:35|GENUINE|  1|
|340028465709212| 430409|   58270|15-11-2017 01:59:54|GENUINE|  2|
|340028465709212|6503191|   84776|09-11-2017 07:18:21|GENUINE|  3|
|340028465709212|8884049|   25537|07-10-2017 09:17:12|GENUINE|  4|
|340028465709212|9291309|   31322|12-08-2017 08:29:54|GENUINE|  5|
|340028465709212|8370505|   84056|12-07-2017 02:51:29|GENUINE|  6|
|340028465709212|9687739|   51542|05-07-2017 11:05:55|GENUINE|  7|
|340028465709212|6500086|   25040|24-06-2017 01:13:31|GENUINE|  8|
|340028465709212| 581323|   46182|17-05-2017 12:36:12|GENUINE|  9|
|340028465709212|5118701|   12045|30-03-2017 04:09:10|GENUINE| 10|
|340054675199675|  29445|   50140|15-01-2018 10:56:43|GENUINE|  1|
|340054675199675|9728785|   77373|10-01-2018 02:47:11|GENUINE|  2|
|340054675199675|2223104|   35973|09-01-2018 10:59:10|GENUINE|  3|
|340054675199675|1201277|   84530|28-12-2017 05:48:04|GENUINE|  4|
|340054675199675|6140357|   40023|18-12-2017 10:33:04|GENUINE|  5|
|340054675199675|7914699|   41844|12-12-2017 07:04:51|GENUINE|  6|
|340054675199675|7573707|   12024|06-12-2017 08:52:38|GENUINE|  7|
|340054675199675|2797924|   54141|04-12-2017 12:59:15|GENUINE|  8|
|340054675199675|7876899|   71047|27-11-2017 01:54:59|GENUINE|  9|
|340054675199675|5418389|   21084|05-11-2017 12:00:53|GENUINE| 10|
+----------------+------+--------+-------------------+------+---+

>>>
```

- Calculate UCL

```
>>> df_ucl = spark.sql("\
... SELECT a.card_id, (a.avge + (3 * a.std)) as UCL \
... FROM (\
... SELECT t.card_id, AVG(t.amount) AS avge, STDDEV(t.amount) as std \
... FROM table_10transact t \
... GROUP BY t.card_id) a")
>>> df_ucl.createOrReplaceTempView('UCL_table')
>>> spark.sql('SELECT * FROM UCL_table LIMIT 3').show()
+---------------+--------------------+
|        card_id|                 UCL|
+---------------+--------------------+
|340028465709212|1.6685076623853374E7|
|340054675199675|1.5032693399975928E7|
|340082915339645|1.5323729774843596E7|
+---------------+--------------------+

>>>
```

- Insert data into Look-up table

```
>>> spark.sql("INSERT INTO TABLE lookup_table \
... SELECT trans.card_id, ucl.ucl, trans.postcode, trans.transaction_dt, CAST(cdsc.score as double)\
... FROM table_10transact trans \
... JOIN UCL_table ucl \
... ON ucl.card_id = trans.card_id \
... JOIN (\
... SELECT DISTINCT crd.card_id, scr.score \
... FROM card_mem crd \
... JOIN mem_score scr \
... ON crd.member_id = scr.member_id) AS cdsc \
... ON trans.card_id = cdsc.card_id \
... WHERE trans.rn = 1")
DataFrame[]
>>> spark.sql('SELECT * FROM lookup_table LIMIT 3').show()
+---------------+--------------------+--------+-------------------+-----+
|        card_id|                 ucl|postcode|     transaction_dt|score|
+---------------+--------------------+--------+-------------------+-----+
|340028465709212|1.6685076623853374E7|   24658|02-01-2018 03:25:35|  233|
|340054675199675|1.5032693399975928E7|   50140|15-01-2018 10:56:43|  631|
|340082915339645|1.5323729774843596E7|   41754|03-11-2017 09:06:10|  407|
+---------------+--------------------+--------+-------------------+-----+

>>>
```

- Save lookup table in S3 bucket

```
>>> lookup_df = sqlContext.sql("SELECT * FROM lookup_table")
>>> lookup_df.coalesce(1).write.format('csv').options(header='True',delimiter=',').mode('overwrite').save("s3://creditcardcapstone/lookup_table")
```