FILL DATA FOR LOOK-UP TABLE

Take last 10 transactions for each card

Firstly, we will take last 10 transactions for each card from table tb_tran

```
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 tb_tran \
WHERE status = 'GENUINE') a \
WHERE a.rn <= 10”)
```

Then convert it to table tb_10trans

```
>>> df_10trans.createOrReplaceTempView('tb_10trans')
>>> spark.sql('SELECT * FROM tb_10trans LIMIT 20’).show()
```

```
>>> spark.sql('SELECT * FROM tb_10trans 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

Next, we will calculate UCL for each card

```
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 tb_10trans t \
GROUP BY t.card_id) a")
```

Then convert it to table tb_ucl

```
>>> df_ucl.createOrReplaceTempView('tb_ucl')
>>> spark.sql('SELECT * FROM tb_ucl LIMIT 3').show()
```

```
>>> df_ucl.createOrReplaceTempView('tb_ucl')
>>> spark.sql('SELECT * FROM tb_ucl LIMIT 3').show()
+---------------+--------------------+
|        card_id|                 UCL|
+---------------+--------------------+
|340028465709212|1.6685076623853374E7|
|340054675199675|1.5032693399975928E7|
|340082915339645|1.5323729774843596E7|
+---------------+--------------------+
```

Insert data for look-up table

Finally, join those 2 tables with tb_card and tb_score to insert data into look-up table

```
spark.sql("INSERT INTO TABLE tb_lookup \
SELECT trans.card_id, ucl.ucl, trans.postcode,
trans.transaction_dt, cdsc.score \
FROM  tb_10trans trans \
JOIN tb_ucl ucl \
ON ucl.card_id = trans.card_id \
JOIN (\
SELECT DISTINCT crd.card_id, scr.score \
FROM tb_card crd \
JOIN tb_score scr \
ON crd.member_id = scr.member_id) AS cdsc \
ON trans.card_id = cdsc.card_id \
WHERE trans.rn = 1")
```

```
>>> spark.sql('SELECT * FROM tb_lookup LIMIT 3’).show()
```

```
>>> spark.sql('SELECT * FROM tb_lookup LIMIT 3').show()
+---------------+-------------------+--------+-------------------+-----+
|        card_id|                ucl|postcode|     transaction_dt|score|
+---------------+-------------------+--------+-------------------+-----+
|5411141346922507|1.5997512196104523E7|   25156|20-09-2017 09:59:16|  225|
|4502008970767222| 1.684959540133459E7|   93010|31-01-2018 09:39:57|  657|
|5341963603599990| 1.516390152866365E7|   18419|09-01-2018 07:28:30|  554|
+---------------+-------------------+--------+-------------------+-----+
```

Saving

Save the lookup table into MongoDB

```
df_lookup.write.format("mongodb").mode("append").save()
```

```
switched to db transaction_db
transaction_db> db.card_transactions.findOne()
{
  _id: ObjectId("646a94c87760a83f9571a002"),
  card_id: Long("348702330256514"),
  member_id: Long("37495066290"),
  amount: 4310362,
  postcode: 33946,
  pos_id: Long("614677375609919"),
  transaction_dt: '11-02-2018 00:00:00',
  status: 'GENUINE'
}
transaction_db> db.tb_lookup.findOne()
{
  _id: ObjectId("646a9b313ec71934249ddea4"),
  card_id: Long("5411141346922507"),
  ucl: 15997512.196104523,
  postcode: 25156,
  transaction_dt: '20-09-2017 09:59:16',
  score: 225
}
```

We can also store the lookup table in S3 for further use

```
df_lookup = sqlContext.sql("SELECT * FROM tb_lookup")
df_lookup.coalesce(1).write.format('csv').options(header='Tru
e', delimiter=',').mode('overwrite').save("s3://history-
transactions/lookup")
```