

## Loading the Lookup Table

### Commands to load the relevant data in the Lookup Table

- Calculate the Upper Control Limit (UCL),  $UCL = \text{Moving Average} + 3 * (\text{Standard Deviation})$ , We shall first calculate the moving average of card amount for last 10 transactions.
- Create a window over existing dataframe and aggregate the same card\_id, the dataframe is grouped by card\_id and then order by transaction\_date.

```
from pyspark.sql import SparkSession
from pyspark.sql import functions as f
from pyspark.sql import Window

# Create a window specification
window = Window.partitionBy('card_id').orderBy('transaction_dt').desc() # Assuming 'transaction_dt'
is the correct column name

# Rank transactions and filter for top 10 per card_id
history_df = history.withColumn('rank', f.rank().over(window)) \
    .filter(f.col('rank') <= 10)

# Show the resulting DataFrame
history_df.show()
```

```
window = Window.partitionBy(history['card_id']).orderBy(history['transaction_date'].desc())
```

```
history_df = history.select('*', f.rank().over(window).alias('rank')).filter(f.col('rank') <= 10)
```

```
history_df.show()
```

card_id	amount	postcode	pos_id	status	score	transaction_date	rank
340379737226464	1784098	26656	000383013889790	GENUINE	229	2018-01-27 00:19:47	1
340379737226464	3759577	61334	016312401940277	GENUINE	229	2018-01-18 14:26:09	2
340379737226464	4080612	51338	562082278231631	GENUINE	229	2018-01-14 20:54:02	3
340379737226464	4242710	96105	285501971776349	GENUINE	229	2018-01-11 19:09:55	4
340379737226464	9061517	40932	232455833079472	GENUINE	229	2018-01-10 20:20:33	5
340379737226464	102248	40932	232455833079472	GENUINE	229	2018-01-10 15:04:33	6
340379737226464	7445128	50455	915439934619047	GENUINE	229	2018-01-07 23:52:27	7
340379737226464	5706163	50455	915439934619047	GENUINE	229	2018-01-07 22:07:07	8
340379737226464	8090127	18626	359283931604637	GENUINE	229	2017-12-29 13:24:07	9
340379737226464	9282351	41859	808326141065551	GENUINE	229	2017-12-28 19:50:46	10

```
history_df = history_df.groupBy("card_id").agg(f.round(f.avg('amount'),2).alias('moving_avg'), \
    f.round(f.stddev('amount'),2).alias('Std_Dev'))
```

card_id	moving_avg	Std_Dev
340379737226464	5355453.1	3107063.55
345406224887566	5488456.5	3252527.52
348962542187595	5735629.0	3089916.54
377201318164757	5742377.7	2768545.84
379321864695232	4713319.1	3203114.94
4389973676463558	4923904.7	2306771.9
4407230633003235	4348891.3	3274883.95

- Calculate UCL from the computed standard deviation and moving average:

```
history_df = history_df.withColumn('UCL',history_df.moving_avg+3*(history_df.Std_Dev))
```

card_id	moving_avg	Std_Dev	UCL
340379737226464	5355453.1	3107063.55	1.46766437499999998E7
345406224887566	5488456.5	3252527.52	1.524603906E7
348962542187595	5735629.0	3089916.54	1.5005378620000001E7
377201318164757	5742377.7	2768545.84	1.4048015219999999E7
379321864695232	4713319.1	3203114.94	1.432266392E7
4389973676463558	4923904.7	2306771.9	1.1844220399999999E7
4407230633003235	4348891.3	3274883.95	1.4173543150000002E7

### <Command to see the table created and it's content>

- `list` command is used to view the created table
- `scan 'look_up_table'` command to view data inserted into table.

### Screenshot of the created table

```
5232083808576685 column=info:card_id, timestamp=1607880086427, value=5232083808576685
5232083808576685 column=info:postCode, timestamp=1607880086427, value=17965
5232083808576685 column=info:score, timestamp=1607880086427, value=566
5232083808576685 column=info:transaction_date, timestamp=1607880086427, value=2018-01-09 12:44:31
5232271306465150 column=info:UCL, timestamp=1607880087122, value=10951781.35
5232271306465150 column=info:card_id, timestamp=1607880087122, value=5232271306465150
5232271306465150 column=info:postCode, timestamp=1607880087122, value=12920
5232271306465150 column=info:score, timestamp=1607880087122, value=638
5232271306465150 column=info:transaction_date, timestamp=1607880087122, value=2018-01-22 16:44:59
5232695950818720 column=info:UCL, timestamp=1607880087849, value=15220850.52
5232695950818720 column=info:card_id, timestamp=1607880087849, value=5232695950818720
5232695950818720 column=info:postCode, timestamp=1607880087849, value=79080
5232695950818720 column=info:score, timestamp=1607880087849, value=207
5232695950818720 column=info:transaction_date, timestamp=1607880087849, value=2018-01-29 08:30:32
5239380866598772 column=info:UCL, timestamp=1607880086358, value=12835247.22
5239380866598772 column=info:card_id, timestamp=1607880086358, value=5239380866598772
5239380866598772 column=info:postCode, timestamp=1607880086358, value=72471
5239380866598772 column=info:score, timestamp=1607880086358, value=440
5239380866598772 column=info:transaction_date, timestamp=1607880086358, value=2017-12-07 21:44:43
5242841712000086 column=info:UCL, timestamp=1607880088013, value=15646358.41
5242841712000086 column=info:card_id, timestamp=1607880088013, value=5242841712000086
5242841712000086 column=info:postCode, timestamp=1607880088013, value=48821
5242841712000086 column=info:score, timestamp=1607880088013, value=236
5242841712000086 column=info:transaction_date, timestamp=1607880088013, value=2018-01-27 10:51:48
5249623960609831 column=info:UCL, timestamp=1607880087191, value=12497504.76
5249623960609831 column=info:card_id, timestamp=1607880087191, value=5249623960609831
5249623960609831 column=info:postCode, timestamp=1607880087191, value=16858
5249623960609831 column=info:score, timestamp=1607880087191, value=265
5249623960609831 column=info:transaction_date, timestamp=1607880087191, value=2018-01-28 00:54:29
5252551880815473 column=info:UCL, timestamp=1607880086480, value=11540779.75
5252551880815473 column=info:card_id, timestamp=1607880086480, value=5252551880815473
5252551880815473 column=info:postCode, timestamp=1607880086480, value=39352
5252551880815473 column=info:score, timestamp=1607880086480, value=449
5252551880815473 column=info:transaction_date, timestamp=1607880086480, value=2018-02-01 10:14:39
5253084214148600 column=info:UCL, timestamp=1607880087349, value=13198338.6
5253084214148600 column=info:card_id, timestamp=1607880087349, value=5253084214148600
5253084214148600 column=info:postCode, timestamp=1607880087349, value=78054
5253084214148600 column=info:score, timestamp=1607880087349, value=512
5253084214148600 column=info:transaction_date, timestamp=1607880087349, value=2018-01-27 10:51:49
5254025009868430 column=info:UCL, timestamp=1607880087698, value=14556419.87
```