

Data Analysis

Analyzes updated data and writes output to Blob storage

In []:

```
from pyspark.sql.functions import *
from pyspark.sql.types import *
from pyspark.sql.window import Window
```

In []:

```
# Azure credentials
storageAccountName = 'exchangedata1'
storageAccountAccessKey = '<your-access-key>'
ContainerName = 'source-container'
spark.conf.set(f'fs.azure.account.key.{storageAccountName}.blob.core.windows.net', storageAccountAccessKey)
```

In []:

```
import os
os.listdir('/dbfs/output/')
```

Out[5]: ['parsed_data']

In []:

```
def EOD_parquet_path(date,type):
    blob_path='wasbs://{}.blob.core.windows.net'.format(ContainerName,storageAccountName)
    file_path='/output/EOD_corrected/{}/partition={}/'.format(date,type)
    df = spark.read.parquet(blob_path+file_path)
    return df
```

In []:

```
#read parquet files from EOD Corrected data
curr_quote=EOD_parquet_path('2020-08-06','Q')
curr_trade=EOD_parquet_path('2020-08-06','T')
prev_trade=EOD_parquet_path('2020-08-05','T')
```

In []:

```
#creating temp view
curr_quote.createOrReplaceTempView('tmp_curr_quote')
curr_trade.createOrReplaceTempView('tmp_curr_trade')
prev_trade.createOrReplaceTempView('tmp_prev_trade')
```

Calculate Current Day Trade Analytics

30 minute moving average and trade price

In []:

```
# uses tmp_curr_trade table
curr_trade_analytics=spark.sql('SELECT symbol,exchange,event_tm,event_seq_nb, trade_pr,AVG(trade_pr) OVER (PARTITION BY exchange,symbol ORDER BY event_tm RANGE BETWEEN INTERVAL 3
curr_trade_analytics.show(5,truncate=False)
# create tmp_trade_analytics temporary table
# curr_trade_analytics.createOrReplaceTempView('tmp_curr_trade_analytics_test')
```

+-----+-----+-----+-----+-----+-----+

symbol	exchange	event_tm	event_seq_nb	trade_pr	mov_avg_pr
SYMA	NASDAQ	2020-08-06 10:42:21.079	10	78.93246	78.93245697021484
SYMA	NASDAQ	2020-08-06 12:00:29.595	20	77.0967	77.0967025756836
SYMA	NASDAQ	2020-08-06 13:09:29.883	30	78.31462	78.31462097167969
SYMA	NASDAQ	2020-08-06 14:27:08.62	40	75.84401	75.84400939941406
SYMA	NASDAQ	2020-08-06 15:39:00.929	50	77.62613	77.62612915039062

only showing top 5 rows

In []:

```
# save temp view of current trade analytics as hive table
# this will create a table in folder the structure /dbfs/user/hive/warehouse/
curr_trade_analytics.write.saveAsTable('tmp_curr_trade_analytics')
```

Calculate Previous Day Trade Analytics

30 minute moving average and trade price

In []:

```
# 30MA for previous days trade information
prev_trade_analytics=spark.sql('SELECT symbol,exchange,event_tm,event_seq_nb, trade_pr,AVG(trade_pr) OVER (PARTITION BY exchange,symbol ORDER BY event_tm RANGE BETWEEN INTERVAL 3
prev_trade_analytics.orderBy('exchange','symbol','event_tm').show(5,truncate=False)
prev_trade_analytics.createOrReplaceTempView('tmp_prev_trade_analytics')
```

symbol	exchange	event_tm	event_seq_nb	trade_pr	mov_avg_pr
SYMA	NASDAQ	2020-08-05 10:38:50.046	10	77.7757	77.77570343017578
SYMA	NASDAQ	2020-08-05 11:58:33.106	20	75.715225	75.71522521972656
SYMA	NASDAQ	2020-08-05 13:09:24.38	30	75.87926	75.87925720214844
SYMA	NASDAQ	2020-08-05 14:22:41.39	40	78.324715	78.32471466064453
SYMA	NASDAQ	2020-08-05 15:33:58.825	50	75.72602	75.72602081298828

only showing top 5 rows

In []:

```
# filtering previous days trade analytics for only closing trade price and closing MA
prev_close_trade_analytics=spark.sql('SELECT symbol,exchange,event_tm,event_seq_nb,trade_pr, mov_avg_pr FROM (SELECT *, ROW_NUMBER() OVER (PARTITION BY exchange, symbol ORDER BY
prev_close_trade_analytics.show()
# creating temp view
# prev_trade_analytics.createOrReplaceTempView('tmp_prev_close_trade_analytics')
```

symbol	exchange	event_tm	event_seq_nb	trade_pr	mov_avg_pr
SYMA	NASDAQ	2020-08-05 21:40:...	100	77.24676	77.24675750732422
SYMB	NASDAQ	2020-08-05 21:03:...	100	35.537262	35.537261962890625
SYMC	NASDAQ	2020-08-05 21:49:...	100	158.02032	158.02032470703125
SYMA	NYSE	2020-08-05 21:30:...	100	77.78611	77.7861099243164
SYMB	NYSE	2020-08-05 21:27:...	100	33.956287	33.9562873840332
SYMC	NYSE	2020-08-05 21:52:...	100	160.61949	160.61949157714844

```
In [ ]: # save temp view of previous close trade analytics as table
prev_close_trade_analytics.write.saveAsTable('tmp_prev_close_trade_analytics')
```

```
In [ ]: os.listdir('/dbfs/user/hive/warehouse/')
```

Out[54]: ['tmp_curr_trade_analytics', 'tmp_prev_close_trade_analytics']

Join Quote with Current Trade Analytics

```
In [ ]: Updated_quote_df= spark.sql('SELECT q.symbol, q.exchange, q.event_tm, q.event_seq_nb, q.bid_pr, q.bid_size, q.ask_pr, q.ask_size, ma.trade_pr,ma.mov_avg_pr, ROW_NUMBER() OVER (PARTITION BY q.symbol, q.exchange, q.event_tm, q.event_seq_nb ORDER BY q.bid_pr) AS row_num FROM tmp_prev_close_trade_analytics q JOIN tmp_curr_trade_analytics ma ON q.symbol=ma.symbol AND q.event_tm=ma.event_tm AND q.event_seq_nb=ma.event_seq_nb')
Updated_quote_df.show()
Updated_quote_df.createOrReplaceTempView('tmp_updated_quote')
```

symbol	exchange	event_tm	event_seq_nb	bid_pr	bid_size	ask_pr	ask_size	trade_pr	mov_avg_pr	row_num
SYMA	NASDAQ	2020-08-06 09:38:...	1	78.133705	100	79.825165	100	null	null	1
SYMA	NASDAQ	2020-08-06 09:46:...	2	76.52305	100	76.57241	100	null	null	1
SYMA	NASDAQ	2020-08-06 09:52:...	3	78.74535	100	79.0928	100	null	null	1
SYMA	NASDAQ	2020-08-06 09:58:...	4	75.613625	100	76.949776	100	null	null	1
SYMA	NASDAQ	2020-08-06 10:07:...	5	77.45084	100	78.725334	100	null	null	1
SYMA	NASDAQ	2020-08-06 10:15:...	6	79.29842	100	81.071915	100	null	null	1
SYMA	NASDAQ	2020-08-06 10:22:...	7	77.76145	100	79.24523	100	null	null	1
SYMA	NASDAQ	2020-08-06 10:29:...	8	75.601135	100	76.95535	100	null	null	1
SYMA	NASDAQ	2020-08-06 10:35:...	9	76.30005	100	77.618706	100	null	null	1
SYMA	NASDAQ	2020-08-06 10:50:...	11	77.96207	100	79.18113	100	78.93246	78.93245697021484	1
SYMA	NASDAQ	2020-08-06 10:59:...	12	76.279816	100	77.32987	100	78.93246	78.93245697021484	1
SYMA	NASDAQ	2020-08-06 11:06:...	13	78.76234	100	79.92418	100	78.93246	78.93245697021484	1
SYMA	NASDAQ	2020-08-06 11:15:...	14	79.196526	100	80.117584	100	78.93246	78.93245697021484	1
SYMA	NASDAQ	2020-08-06 11:23:...	15	74.83831	100	76.313354	100	78.93246	78.93245697021484	1
SYMA	NASDAQ	2020-08-06 11:32:...	16	76.65085	100	77.56526	100	78.93246	78.93245697021484	1
SYMA	NASDAQ	2020-08-06 11:40:...	17	75.345146	100	76.79113	100	78.93246	78.93245697021484	1
SYMA	NASDAQ	2020-08-06 11:49:...	18	75.37165	100	76.572685	100	78.93246	78.93245697021484	1
SYMA	NASDAQ	2020-08-06 11:55:...	19	75.16338	100	75.755356	100	78.93246	78.93245697021484	1
SYMA	NASDAQ	2020-08-06 12:07:...	21	75.25071	100	76.894264	100	77.0967	77.0967025756836	1
SYMA	NASDAQ	2020-08-06 12:07:...	21	75.25071	100	76.894264	100	78.93246	78.93245697021484	2

only showing top 20 rows

Join Updated Quote with previous day close trade analytics and calculate spread

```
In [ ]: #uses coalesce to select non-null trade analytics value in updated quote or previous day trade table
Final_quote=spark.sql('SELECT q.symbol, q.exchange, q.event_tm, q.event_seq_nb, q.bid_pr, q.bid_size, q.ask_pr, q.ask_size, q.bid_pr-c.trade_pr as bid_spread, q.ask_pr-c.trade_pr as ask_spread, COALESCE(q.last_trade_pr, c.last_trade_pr) as last_trade_pr, COALESCE(q.last_mov_avg_pr, c.last_mov_avg_pr) as last_mov_avg_pr FROM tmp_updated_quote q JOIN tmp_prev_close_trade_analytics c ON q.symbol=c.symbol AND q.event_tm=c.event_tm AND q.event_seq_nb=c.event_seq_nb')
Final_quote.show()
```

symbol	exchange	event_tm	event_seq_nb	bid_pr	bid_size	ask_pr	ask_size	bid_spread	ask_spread	last_trade_pr	last_mov_avg_pr
SYMA	NASDAQ	2020-08-06 09:38:...	1	78.133705	100	79.825165	100	0.88694763	2.5784073	77.24676	77.24675750732422
SYMA	NASDAQ	2020-08-06 09:46:...	2	76.52305	100	76.57241	100	-0.7237091	-0.6743469	77.24676	77.24675750732422
SYMA	NASDAQ	2020-08-06 09:52:...	3	78.74535	100	79.0928	100	1.4985962	1.8460388	77.24676	77.24675750732422

				Data_Analytic_prod			
SYMA	NASDAQ 2020-08-06 09:58:...	4	75.613625	100	76.949776	100	-1.6331329 -0.2969818 77.24676 77.24675750732422
SYMA	NASDAQ 2020-08-06 10:07:...	5	77.45084	100	78.725334	100	0.2040863 1.4785767 77.24676 77.24675750732422
SYMA	NASDAQ 2020-08-06 10:15:...	6	79.29842	100	81.071915	100	2.0516663 3.8251572 77.24676 77.24675750732422
SYMA	NASDAQ 2020-08-06 10:22:...	7	77.76145	100	79.24523	100	0.5146942 1.9984741 77.24676 77.24675750732422
SYMA	NASDAQ 2020-08-06 10:29:...	8	75.601135	100	76.95535	100	-1.6456223 -0.29140472 77.24676 77.24675750732422
SYMA	NASDAQ 2020-08-06 10:35:...	9	76.30005	100	77.618706	100	-0.9467087 0.37194824 77.24676 77.24675750732422
SYMA	NASDAQ 2020-08-06 10:50:...	11	77.96207	100	79.18113	100	0.71530914 1.934372 78.93246 78.93245697021484
SYMA	NASDAQ 2020-08-06 10:59:...	12	76.279816	100	77.32987	100	-0.96694183 0.083114624 78.93246 78.93245697021484
SYMA	NASDAQ 2020-08-06 11:06:...	13	78.76234	100	79.92418	100	1.5155792 2.6774216 78.93246 78.93245697021484
SYMA	NASDAQ 2020-08-06 11:15:...	14	79.196526	100	80.117584	100	1.9497681 2.8708267 78.93246 78.93245697021484
SYMA	NASDAQ 2020-08-06 11:23:...	15	74.83831	100	76.313354	100	-2.4084473 -0.933403 78.93246 78.93245697021484
SYMA	NASDAQ 2020-08-06 11:32:...	16	76.65085	100	77.56526	100	-0.5959091 0.31850433 78.93246 78.93245697021484
SYMA	NASDAQ 2020-08-06 11:40:...	17	75.345146	100	76.79113	100	-1.9016113 -0.45562744 78.93246 78.93245697021484
SYMA	NASDAQ 2020-08-06 11:49:...	18	75.37165	100	76.572685	100	-1.8751068 -0.67407227 78.93246 78.93245697021484
SYMA	NASDAQ 2020-08-06 11:55:...	19	75.16338	100	75.755356	100	-2.083374 -1.4914017 78.93246 78.93245697021484
SYMA	NASDAQ 2020-08-06 12:07:...	21	75.25071	100	76.894264	100	-1.996048 -0.3524933 77.0967 77.0967025756836
SYMA	NASDAQ 2020-08-06 12:15:...	22	77.74453	100	78.667305	100	0.49777222 1.4205475 77.0967 77.0967025756836

+-----+-----+-----+-----+-----+-----+-----+-----+
only showing top 20 rows

Load analytical data to Blob storage

```
In [ ]: def load_analytical_data(df,date):
        blob_path='wasbs://{ }@{ }.blob.core.windows.net'.format(ContainerName,storageAccountName)
        dir_path='/output/Analytical_data/{ }/'.format(date)
        df.write.parquet(blob_path+dir_path)
        return

In [ ]: load_analytical_data(Final_quote,'2020-08-06')
```