# Stephen Reagin

## March 19, 2024

### Take-Home Challenge for Align Technology: Task 3

The final query is:

```
SELECT sub.* FROM
                (SELECT t1.customer_name, t2.order_date, AVG(t2.amount) as mean_amt,
                    ROW_NUMBER() OVER
                            (PARTITION BY t1.customer_name
                              ORDER BY AVG(t2.amount) DESC
                            ) as rank
                        FROM dbo.customer t1
                        JOIN dbo.purchase_order t2 on t1.customer_id = t2.customer_id
                        GROUP BY t1.customer_name, t2.order_date
                        ORDER BY t1.customer_name, mean_amt DESC) sub
        WHERE rank <= 5
```

Run through all the cells below to reproduce the query (there is no need to pull in any additional files).

In [1]:
```
import pandas as pd
import numpy as np

from datetime import date, timedelta
from random import choices
```

In [2]:
```
customer_id = [1,2,3,4,5,6,7,8,9,10]
customer_name = ['Bond, James','McCormick, Kenny', 'Newton, Isaac',
                 'Potter, Harry','Dumbledore, A.P.W.B.','Baggins, Frodo','Gandalf, Gray',
                 'Einstein, Albert', 'Altuve, Jose','Sagan, Carl']

dbo_customer = pd.DataFrame([customer_id, customer_name]).T
dbo_customer = dbo_customer.rename(columns={0:'customer_id', 1: "customer_name"})

dbo_customer
```

Out[2]:

| | customer_id | customer_name |
|---|---|---|
| 0 | 1 | Bond, James |
| 1 | 2 | McCormick, Kenny |
| 2 | 3 | Newton, Isaac |
| 3 | 4 | Potter, Harry |
| 4 | 5 | Dumbledore, A.P.W.B. |
| 5 | 6 | Baggins, Frodo |
| 6 | 7 | Gandalf, Gray |
| 7 | 8 | Einstein, Albert |
| 8 | 9 | Altuve, Jose |
| 9 | 10 | Sagan, Carl |

In [3]:
```
test_date1, test_date2 = date(2015, 6, 3), date(2015, 7, 1)
K = 10000

res_dates = [test_date1]
while test_date1 != test_date2:
    test_date1 += timedelta(days=1)
    res_dates.append(test_date1)

res = choices(res_dates, k=K)

purchase_order_id = np.arange(1,10001,1)
customer_id = np.random.randint(1,10,10000)
amount = np.random.randint(1,50000, 10000)
order_date = res
```

In [4]:
```
dbo_purchase_order = pd.DataFrame([purchase_order_id, customer_id, amount, order_date]).T
dbo_purchase_order = dbo_purchase_order.rename(columns={0: "purchase_order_id", 1: 'customer_id', 2: 'amount', 3: 'order_date'})

dbo_purchase_order.head()
```

Out[4]:

| | purchase_order_id | customer_id | amount | order_date |
|---|---|---|---|---|
| 0 | 1 | 4 | 44219 | 2015-06-05 |
| 1 | 2 | 3 | 46054 | 2015-06-24 |
| 2 | 3 | 5 | 23141 | 2015-06-13 |
| 3 | 4 | 9 | 29903 | 2015-06-24 |
| 4 | 5 | 9 | 21027 | 2015-07-01 |

In [5]:
```
dbo_customer.to_csv('dbo_customer.csv', index=False)
dbo_purchase_order.to_csv('dbo_purchase_order.csv', index=False)
```

Create SQLITE database

In [6]:
```
from sqlalchemy.engine import create_engine
import sqlite3
```

In [7]:
```
def create_connection(db_file):
    """ create a database connection to a SQLite database """
    conn = None
    try:
        conn = sqlite3.connect(db_file)
        print(sqlite3.version)
    except Error as e:
        print(e)
    finally:
        if conn:
            conn.close()


if __name__ == '__main__':
    create_connection('main_table.db')
```

```
2.6.0
```

In [8]:
```
import sqlite3

conn = sqlite3.connect('main_table.db')
c = conn.cursor()

c.execute('''CREATE TABLE customer (customer_id int, customer_name text)''')
c.execute('''CREATE TABLE purchase_order (purchase_order_id int, customer_id int, amount int, order_date date)''')

#load CSV info to database
df1 = pd.read_csv('dbo_customer.csv')
df2 = pd.read_csv('dbo_purchase_order.csv')
df1.to_sql('customer', conn, if_exists='append', index = False)
df2.to_sql('purchase_order', conn, if_exists='append', index = False)
```

Out[8]:
```
10000
```

## THIS IS THE EXAMPLE QUERY

In [10]:
```
query_df = pd.DataFrame(c.execute(
    '''SELECT sub.* FROM
                (SELECT t1.customer_name, t2.order_date, ROUND(AVG(t2.amount)) as mean_amt,
                    ROW_NUMBER() OVER
                            (PARTITION BY t1.customer_name
                              ORDER BY AVG(t2.amount) DESC
                            ) as rank
                        FROM customer t1
                        JOIN purchase_order t2 on t1.customer_id = t2.customer_id
                        GROUP BY t1.customer_name, t2.order_date
                        ORDER BY t1.customer_name, mean_amt DESC) sub
        WHERE rank <= 5''').fetchall())

query_df
```

Out[10]:

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | Altuve, Jose | 2015-06-04 | 31361.0 | 1 |
| 1 | Altuve, Jose | 2015-06-26 | 30608.0 | 2 |
| 2 | Altuve, Jose | 2015-06-24 | 29975.0 | 3 |
| 3 | Altuve, Jose | 2015-06-17 | 29875.0 | 4 |
| 4 | Altuve, Jose | 2015-06-11 | 29197.0 | 5 |
| 5 | Baggins, Frodo | 2015-06-03 | 29359.0 | 1 |
| 6 | Baggins, Frodo | 2015-06-27 | 28630.0 | 2 |
| 7 | Baggins, Frodo | 2015-06-13 | 27412.0 | 3 |
| 8 | Baggins, Frodo | 2015-06-07 | 27383.0 | 4 |
| 9 | Baggins, Frodo | 2015-06-12 | 27300.0 | 5 |
| 10 | Bond, James | 2015-06-19 | 28941.0 | 1 |
| 11 | Bond, James | 2015-06-12 | 28495.0 | 2 |
| 12 | Bond, James | 2015-06-29 | 28357.0 | 3 |
| 13 | Bond, James | 2015-06-21 | 28206.0 | 4 |
| 14 | Bond, James | 2015-06-14 | 27245.0 | 5 |
| 15 | Dumbledore, A.P.W.B. | 2015-06-17 | 30018.0 | 1 |
| 16 | Dumbledore, A.P.W.B. | 2015-06-08 | 29167.0 | 2 |
| 17 | Dumbledore, A.P.W.B. | 2015-06-15 | 28410.0 | 3 |
| 18 | Dumbledore, A.P.W.B. | 2015-06-18 | 28268.0 | 4 |
| 19 | Dumbledore, A.P.W.B. | 2015-07-01 | 27118.0 | 5 |
| 20 | Einstein, Albert | 2015-06-21 | 30675.0 | 1 |
| 21 | Einstein, Albert | 2015-06-10 | 29907.0 | 2 |
| 22 | Einstein, Albert | 2015-06-12 | 28164.0 | 3 |
| 23 | Einstein, Albert | 2015-06-17 | 27487.0 | 4 |
| 24 | Einstein, Albert | 2015-06-11 | 27089.0 | 5 |
| 25 | Gandalf, Gray | 2015-06-21 | 30761.0 | 1 |
| 26 | Gandalf, Gray | 2015-06-09 | 29828.0 | 2 |
| 27 | Gandalf, Gray | 2015-06-10 | 28740.0 | 3 |
| 28 | Gandalf, Gray | 2015-06-29 | 28687.0 | 4 |
| 29 | Gandalf, Gray | 2015-06-13 | 27609.0 | 5 |
| 30 | McCormick, Kenny | 2015-06-22 | 29616.0 | 1 |
| 31 | McCormick, Kenny | 2015-06-25 | 29593.0 | 2 |
| 32 | McCormick, Kenny | 2015-06-23 | 28487.0 | 3 |
| 33 | McCormick, Kenny | 2015-06-19 | 28383.0 | 4 |
| 34 | McCormick, Kenny | 2015-06-03 | 28157.0 | 5 |
| 35 | Newton, Isaac | 2015-06-10 | 28872.0 | 1 |
| 36 | Newton, Isaac | 2015-06-19 | 28259.0 | 2 |
| 37 | Newton, Isaac | 2015-06-27 | 28050.0 | 3 |
| 38 | Newton, Isaac | 2015-06-06 | 26840.0 | 4 |
| 39 | Newton, Isaac | 2015-06-05 | 26638.0 | 5 |
| 40 | Potter, Harry | 2015-06-28 | 29634.0 | 1 |
| 41 | Potter, Harry | 2015-06-27 | 27809.0 | 2 |
| 42 | Potter, Harry | 2015-06-16 | 27554.0 | 3 |
| 43 | Potter, Harry | 2015-06-20 | 26924.0 | 4 |
| 44 | Potter, Harry | 2015-06-13 | 26537.0 | 5 |

# Thank you for reading!