CUSTOMER SENTIMENT ANALYSIS

(END CAPSTONE)

SUBMITTED BY: T.SASHIDHAR

COHORT: 20

INTEGRATED DATA SCIENCE

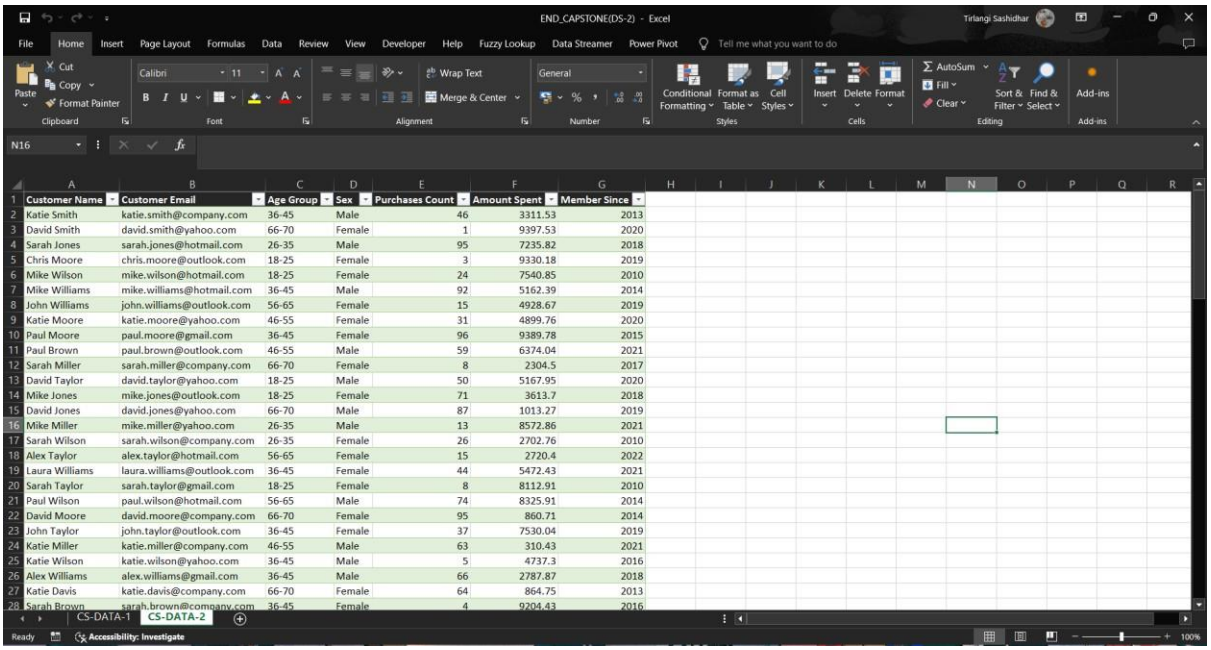
1. Introduction

1.1 Project Overview

This project aims to build an automated system to analyze customer data, calculate Customer Lifetime Value (CLV), and segment customers based on their purchasing behavior. The system also includes interactive visualizations for exploring relationships between variables such as customer segments, purchase history, and revenue.

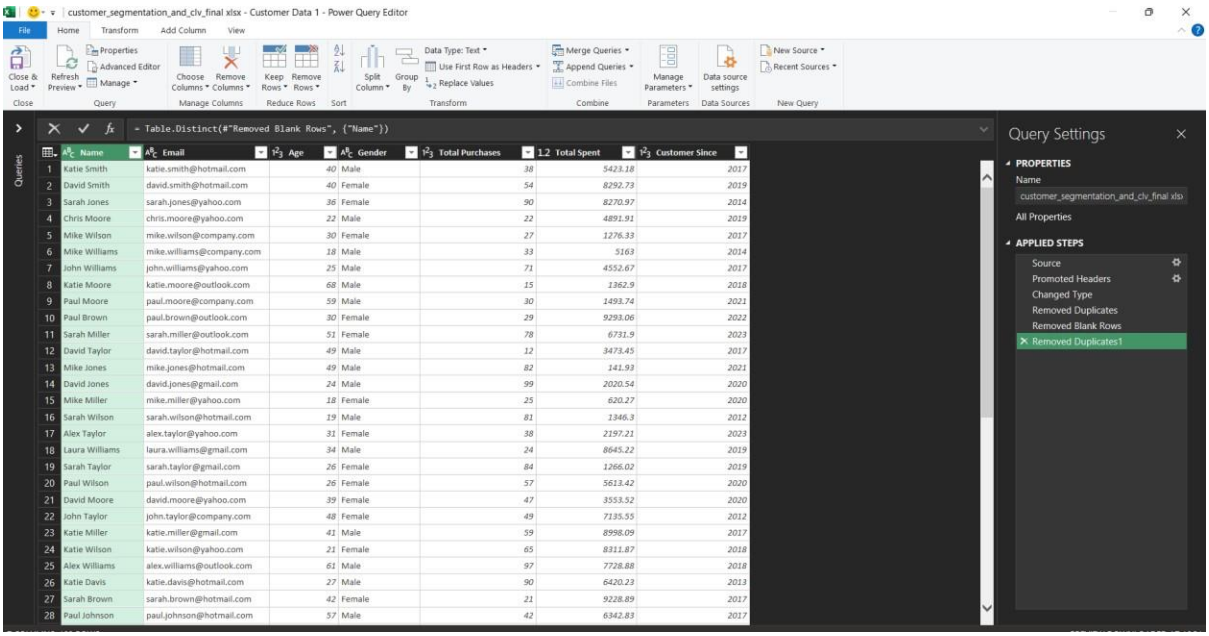
The system is designed to handle large datasets efficiently, using Python-based tools and libraries such as Pandas, Matplotlib, Plotly, and IPyWidgets. Key features include:

- Customer Lifetime Value (CLV) Calculation: Calculate CLV to identify high-value customers.
- Customer Segmentation: Customers are segmented based on CLV into High, Medium, and Low-value segments.
- Memory-Efficient Data Processing: A generator function is used to handle large datasets without consuming excessive memory.
- Interactive Visualizations: Visual analysis with dynamic, interactive charts that allow users to filter and explore customer segments and trends over time.



2.1. Data Preprocessing

- **Loading Data**: The dataset was loaded from an Excel file.
- **Date Conversion**: The 'Customer Since' column was converted to a datetime format for easier manipulation and filtering.
- **Handling Missing Data**: Any invalid date entries were identified and handled appropriately.



	Name	Email	Age	Gender	Total Purchases	Total Spent	Customer Since
1	Katie Smith	katie.smith@hotmail.com	40	Male	38	5423.18	2017
2	David Smith	david.smith@hotmail.com	40	Female	54	8292.73	2019
3	Sarah Jones	sarah.jones@yahoo.com	36	Female	90	8270.97	2014
4	Chris Moore	chris.moore@yahoo.com	22	Male	22	4891.91	2019
5	Mike Wilson	mike.wilson@company.com	30	Female	27	1276.33	2017
6	Mike Williams	mike.williams@company.com	18	Male	33	5163	2014
7	John Williams	john.williams@yahoo.com	25	Male	71	4552.67	2017
8	Katie Moore	katie.moore@outlook.com	68	Male	15	1362.9	2018
9	Paul Moore	paul.moore@company.com	59	Male	30	1493.74	2021
10	Paul Brown	paul.brown@outlook.com	30	Female	29	9293.06	2022
11	Sarah Miller	sarah.miller@outlook.com	51	Female	78	6731.9	2023
12	David Taylor	david.taylor@hotmail.com	49	Male	12	3473.45	2017
13	Mike Jones	mike.jones@hotmail.com	49	Male	82	141.93	2021
14	David Jones	david.jones@gmail.com	24	Male	99	2020.54	2020
15	Mike Miller	mike.miller@yahoo.com	18	Female	25	620.27	2020
16	Sarah Wilson	sarah.wilson@hotmail.com	19	Male	81	1346.3	2012
17	Alex Taylor	alex.taylor@yahoo.com	31	Female	38	2197.21	2023
18	Laura Williams	laura.williams@gmail.com	34	Male	24	8645.22	2019
19	Sarah Taylor	sarah.taylor@gmail.com	26	Female	84	1266.02	2019
20	Paul Wilson	paul.wilson@hotmail.com	26	Female	57	5613.42	2020
21	David Moore	david.moore@yahoo.com	39	Female	47	3553.52	2020
22	John Taylor	john.taylor@company.com	48	Female	49	7135.55	2012
23	Katie Miller	katie.miller@gmail.com	41	Male	59	8998.09	2017
24	Katie Wilson	katie.wilson@yahoo.com	21	Female	65	8311.87	2018
25	Alex Williams	alex.williams@outlook.com	61	Male	97	7728.88	2018
26	Katie Davis	katie.davis@hotmail.com	27	Male	90	6420.23	2013
27	Sarah Brown	sarah.brown@hotmail.com	42	Female	21	9228.89	2017
28	Paul Johnson	paul.johnson@hotmail.com	57	Male	42	6342.83	2017

2. System Architecture

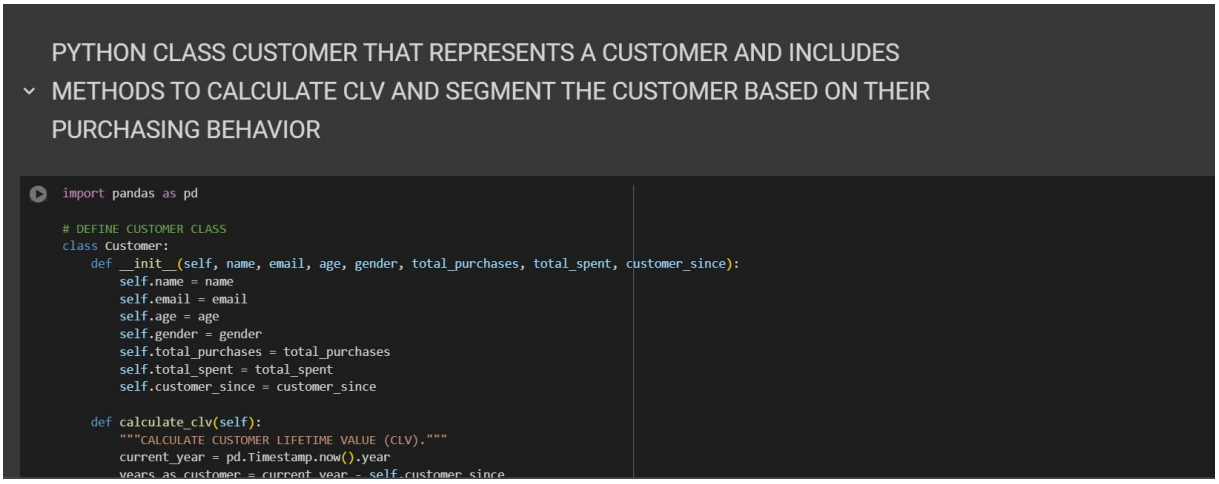
The system architecture revolves around a single Python class called Customer, which encapsulates the main functionalities for processing customer data. The key methods and components are described below.

2.1 Class and Methods

- Customer Class: The Customer class is responsible for managing customer data and performing the core analysis.
    - Attributes:
        - customers_df: A Pandas DataFrame containing customer data such as Customer ID, Total Spent, Total Purchases, and CLV.

Methods:

- calculate_clv: Calculates CLV using the formula CLV = Total Spent / Total Purchases. This method is wrapped with a timing decorator to log execution time.
- segment_customers: Segments customers based on their calculated CLV into three categories: High Value, Medium Value, and Low Value.
- customer_generator: A generator function that yields one customer record at a time, useful for memory-efficient data processing.
- process_large_dataset: Uses the customer_generator to process customer records in a memory-efficient manner.
- plot_summary_statistics: Generates summary statistics and visualizations for customer data, including histograms for total spending and bar charts for revenue.
- interactive_charts: Creates interactive charts using IPyWidgets and Plotly to dynamically explore customer segments.



IMPLEMENT A TIMING_DECORATOR AND APPLY IT TO THE CALCULATE_CLVMETHOD OF THE CUSTOMER CLASS TO LOG EXECUTION TIME.

```python
import time

# DEFINE THE TIMING DECORATOR
def timing_decorator(func):
    def wrapper(*args, **kwargs):
        start_time = time.time()
        result = func(*args, **kwargs)
        end_time = time.time()
        execution_time = end_time - start_time
        print(f"Execution time for {func.__name__}: {execution_time:.6f} seconds")
        return result
    return wrapper


# SETTING UP THE FILE PATH
file_path = "/content/drive/MyDrive/END_CAPSTONE(DS-2).xlsx"
data_1 = pd.read_excel(file_path, sheet_name='CS-DATA-1')

# CREATING A CUSTOMER INSTANCE FROM THE FIRST ROW OF DATA IN CS-DATA-1
customer_example = Customer(
    name=data_1.loc[0, 'Name'],
    email=data_1.loc[0, 'Email'],
    age=data_1.loc[0, 'Age'],
    gender=data_1.loc[0, 'Gender'],
    total_purchases=data_1.loc[0, 'Total Purchases'],
```

```
Execution time for calculate_clv: 0.000082 seconds
Name: Katie Smith
Email: katie.smith@hotmail.com
Age: 40
Gender: Male
Total Purchases: 38
Total Spent: $5423.18
Customer Since: 2017
CLV: $774.74
Segment: Loyal
```

WRITE A GENERATOR FUNCTION TO ITERATE OVER LARGE DATASETS AND YIELD CUSTOMER RECORDS ONE AT A TIME. INTEGRATE THIS GENERATOR INTO THE CUSTOMER CLASS TO PROCESS DATA MEMORY-EFFICIENTLY.

```python
import pandas as pd
import time

# DEFINE THE TIMING DECORATOR
def timing_decorator(func):
    def wrapper(*args, **kwargs):
        start_time = time.time()
        result = func(*args, **kwargs)
        end_time = time.time()
        execution_time = end_time - start_time
        print(f"Execution time for {func.__name__}: {execution_time:.6f} seconds")
        return result
    return wrapper

# DEFINE A GENERATOR FUNCTION TO YIELD CUSTOMER RECORDS FROM A LARGE EXCEL DATASET
def customer_record_generator(file_path, sheet_name, chunk_size=1000):
    """Generator function to read and yield customer records from a large dataset."""
    start_row = 0
    while True:
        chunk = pd.read_excel(file_path, sheet_name=sheet_name, skiprows=start_row, nrows=chunk_size)
```

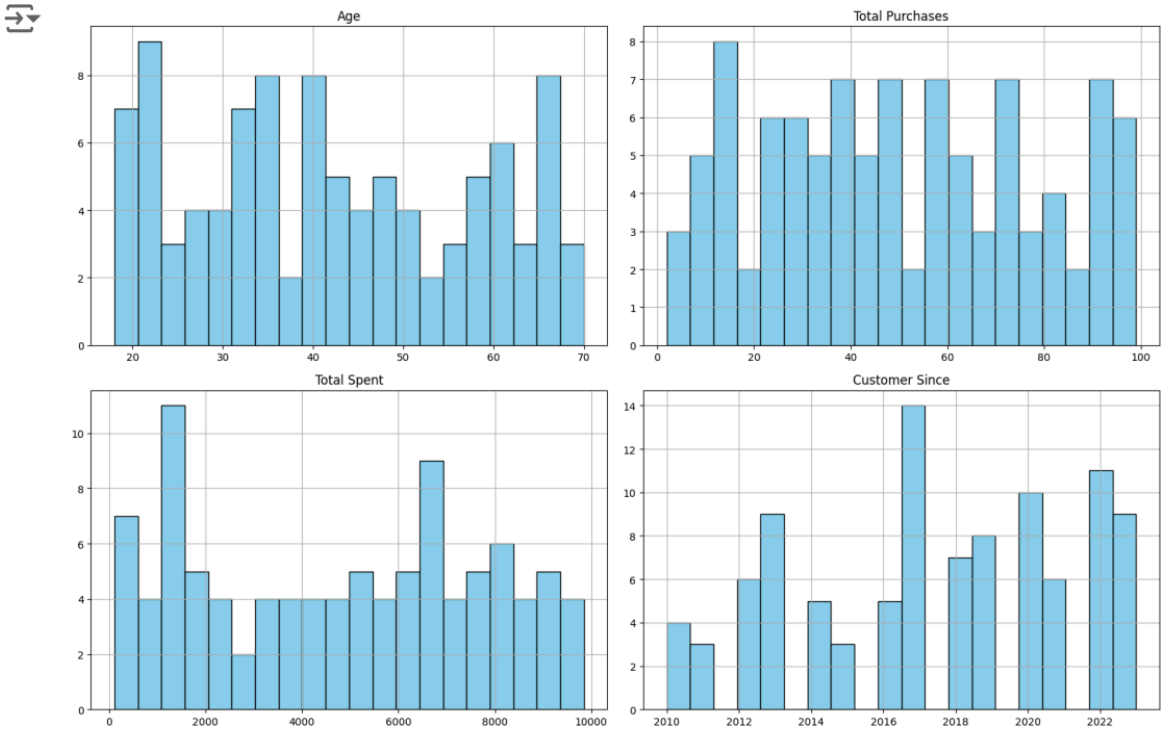
```
CLV: $277.92
Segment: Regular
--------------------------------------------------
Execution time for calculate_clv: 0.000020 seconds
Name: Alex Davis
Email: alex.davis@hotmail.com
Age: 21
Gender: Female
Total Purchases: 78
Total Spent: $6271.78
Customer Since: 2010
CLV: $447.98
Segment: Loyal
--------------------------------------------------
Execution time for calculate_clv: 0.000021 seconds
Name: Jane Johnson
Email: jane.johnson@gmail.com
Age: 35
Gender: Female
Total Purchases: 98
Total Spent: $8400.11
Customer Since: 2012
CLV: $700.01
Segment: Loyal
--------------------------------------------------
```

SUMMARY STATISTICS, VISUALIZATIONS, AND CORRELATION MATRICES TO EXPLORE RELATIONSHIPS BETWEEN VARIABLES

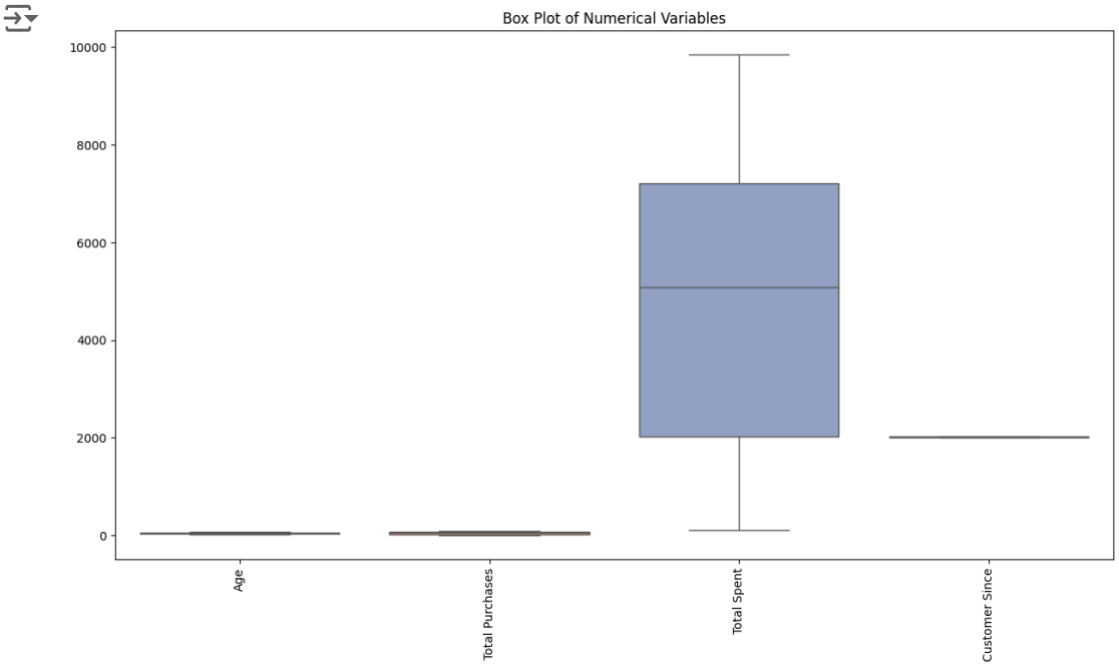
```
[ ] import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    import numpy as np
```

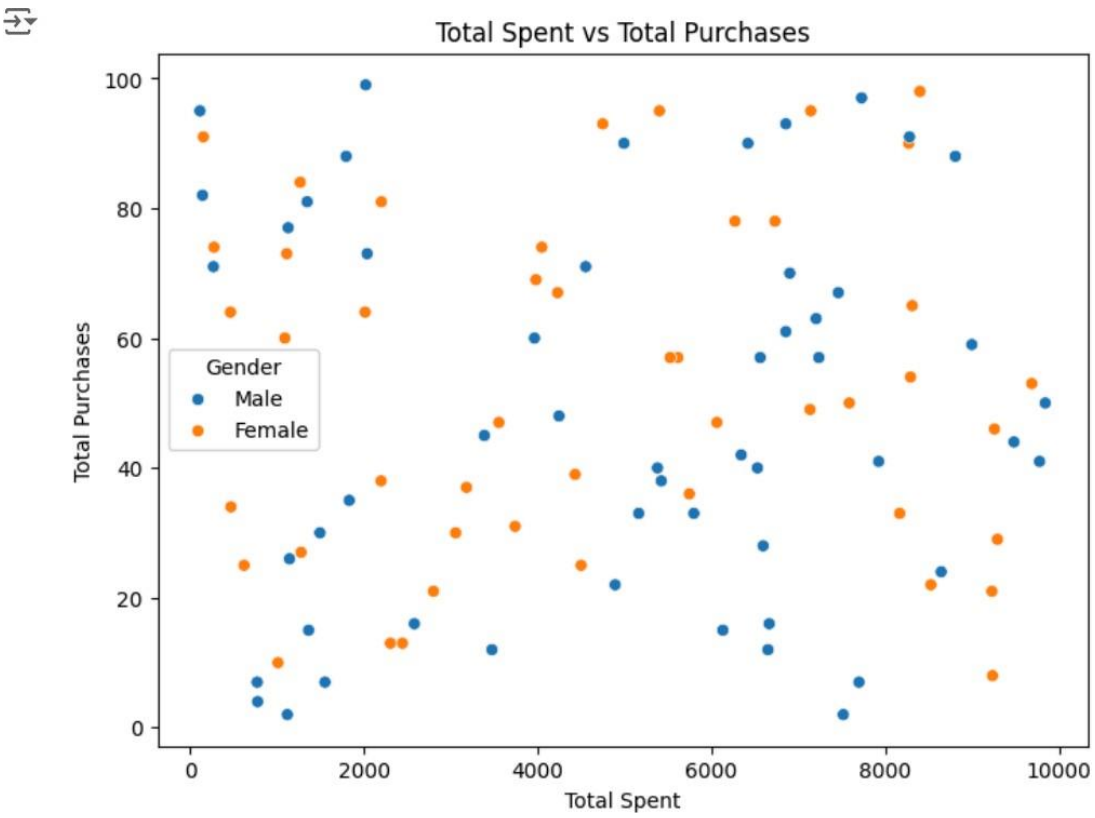
```
# LOAD THE DATASET FROM EXCEL
file_path = "/content/drive/MyDrive/END_CAPSTONE(DS-2).xlsx"
df = pd.read_excel(file_path, sheet_name='CS-DATA-1')

# GENERATE SUMMARY STATISTICS
summary_statistics = df.describe()
print(summary_statistics)
```

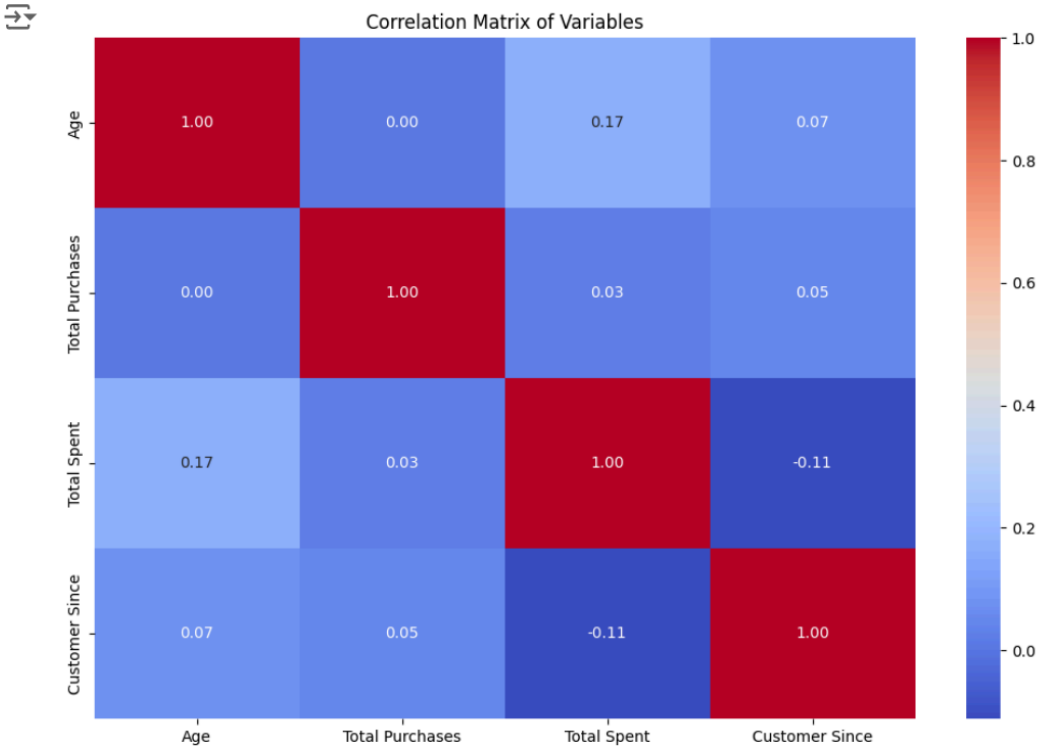
```
                Age  Total Purchases  Total Spent  Customer Since
count  100.000000       100.000000   100.000000      100.000000
mean    42.180000        50.000000  4829.193000     2017.480000
std     15.582547        27.891203  2958.068726        3.841506
min     18.000000         2.000000   112.120000     2010.000000
25%     30.000000        27.750000  2018.805000     2014.000000
50%     41.000000        47.500000  5078.085000     2018.000000
75%     56.250000        73.000000  7213.442500     2021.000000
max     70.000000        99.000000  9843.330000     2023.000000
```



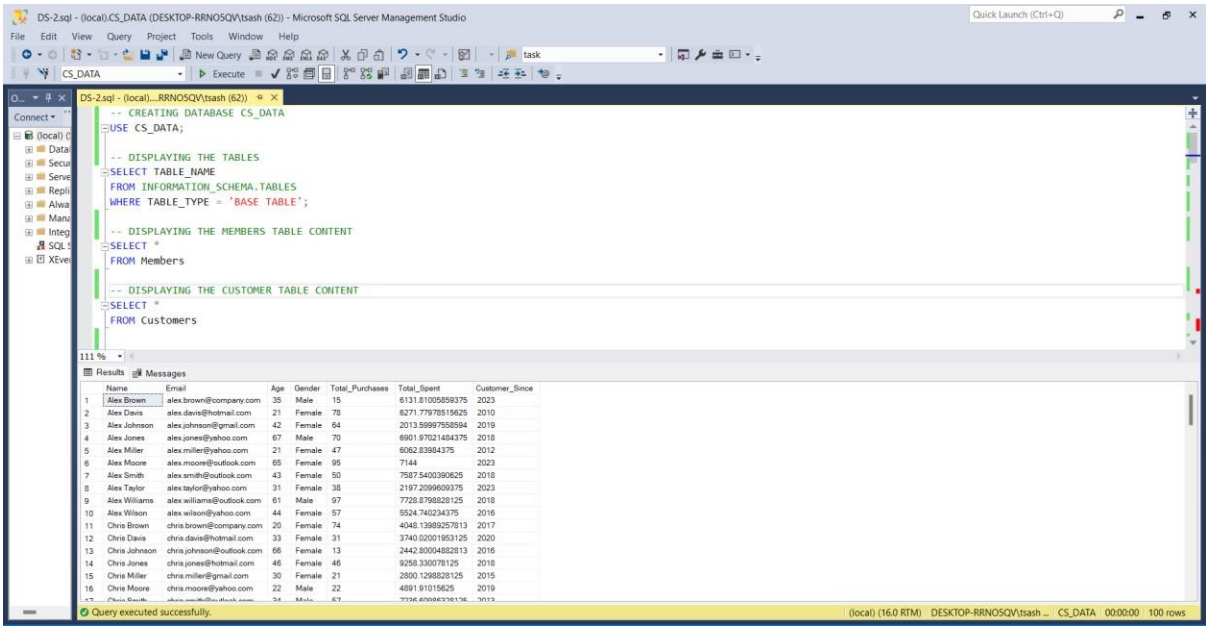

```
# BOX PLOT FOR ALL NUMERICAL VALUES
plt.figure(figsize=(15, 8))
sns.boxplot(data=df.select_dtypes(include=np.number), palette='Set2')
plt.xticks(rotation=90)
plt.title('Box Plot of Numerical Variables')
plt.show()
```


Box Plot of Numerical Variables
10000
8000
6000
4000
2000
0
Age
Total Purchases
Total Spent
Customer Since

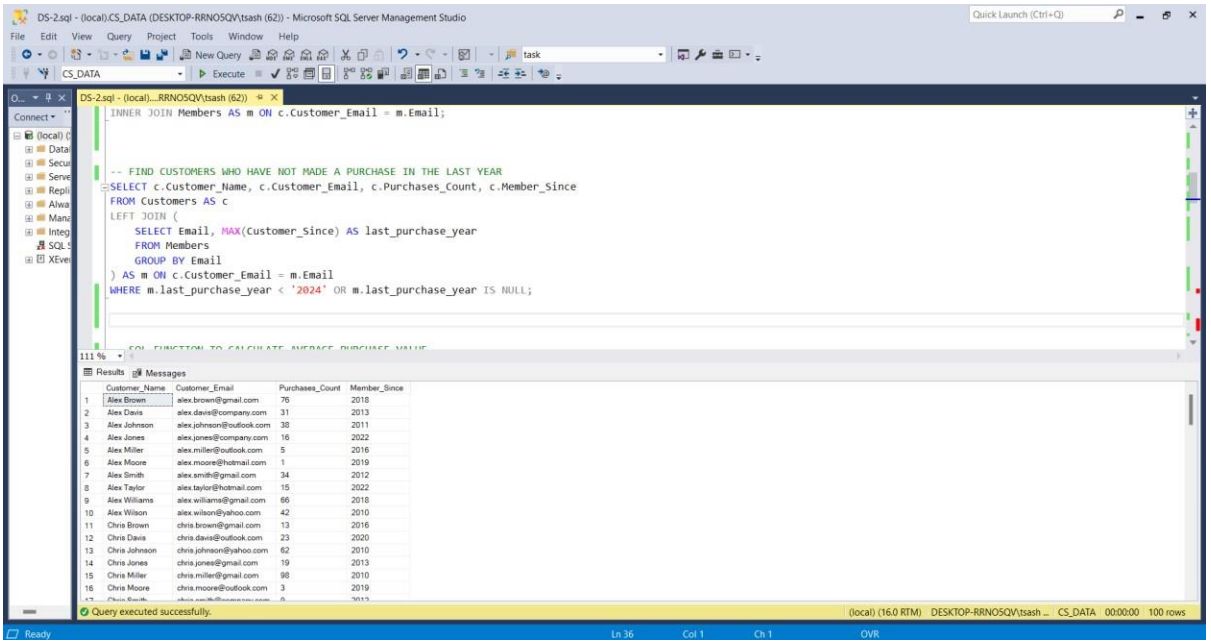

Total Spent vs Total Purchases
Total Purchases
Total Spent
Gender
Male
Female
0
20
40
60
80
100
0
2000
4000
6000
8000
10000



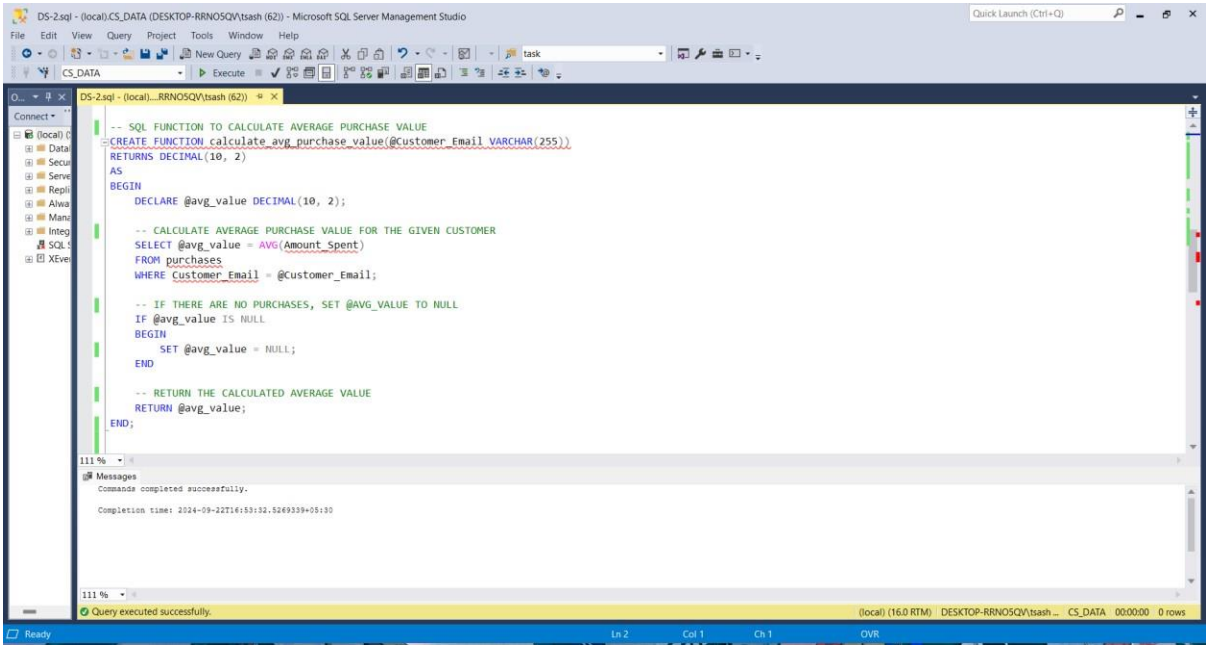
After that, we load the table in SQL SMS to perform further analysis



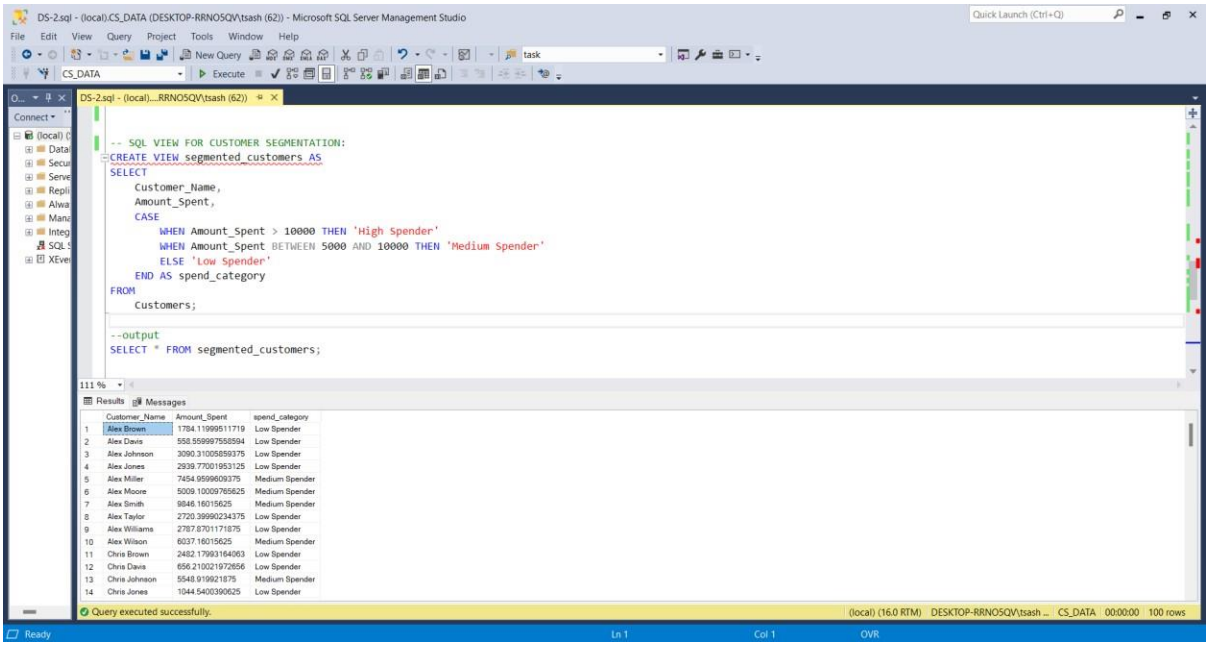
FIND CUSTOMERS WHO HAVE NOT MADE A PURCHASE IN THE LAST YEAR



SQL FUNCTION TO CALCULATE AVERAGE PURCHASE VALUE



SQL VIEW FOR CUSTOMER SEGMENTATION:



POWER BI (DATA VISUALIZATION AND DASHBOARD ANALYSIS)

1. Data Model Design:

In Power BI, the customer dataset should be modeled to reflect key relationships and attributes:

- **Customer Table**: Contains details like Customer NAME, Total Spent, Total Purchases, and Segment (High, Medium, Low Value).
- **CLV Calculation**: Use calculated columns or measures to compute Customer Lifetime Value (CLV) within Power BI, similar to the Python logic.
- **Segmentation**: Apply DAX formulas to segment customers based on CLV thresholds.

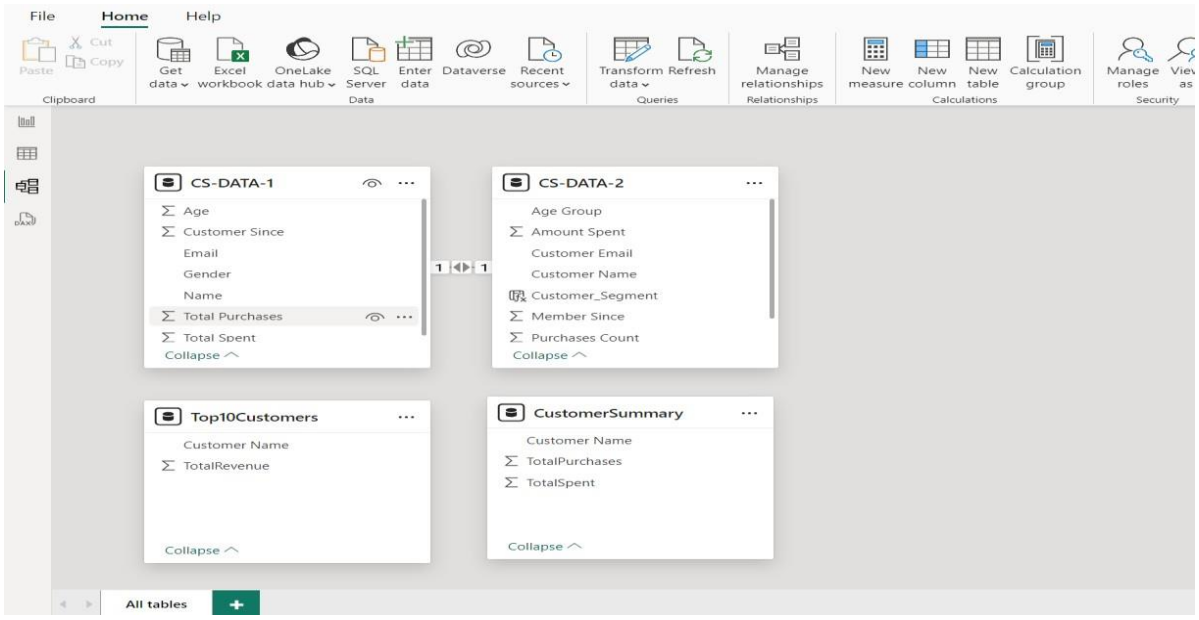
2. Key Measures:

- **Total CLV**: Aggregates CLV across the customer base.
- **Revenue by Segment**: A measure to Total Spent by each segment.
- **Average CLV**: Calculated as total CLV divided by the number of customers.
- **Customer Count by Segment**: Counts customers in each segment (High, Medium, Low).

3. Visualizations:

Power BI will create dynamic visualizations to explore the relationships between customer segments and behavior:

- **Segmented Bar Charts**: Display revenue or CLV by customer segment.
- **Interactive Slicers**: Enable users to filter data by customer segment or date ranges.
- **Line Charts**: Show CLV trends or Total Spent over time, helping analyze the evolution of customer value.
- **Distribution Histograms**: Visualize the distribution of total spending or CLV across customers.



This project has successfully developed an automated system for calculating Customer Lifetime Value (CLV) and segmenting customers. By leveraging Python for data processing, memory-efficient techniques, and interactive visualizations, the system provides scalable solutions for businesses to analyze and categorize customers based on purchasing behavior. The inclusion of a generator function ensures efficient processing of large datasets, while the interactive charts enable real-time exploration of customer segments and trends. Furthermore, the solution's modular and expandable design lays a solid foundation for future enhancements, such as advanced machine learning models and integration with real-time data sources, offering valuable insights for informed business decisions.