ASSIGNMENT # 1

ETL PIPELINE

DWM (CT-463)

ETL PROCESS REPORT

Executive Summary:

This report provides a comprehensive overview of the Extract, Transform, Load (ETL) process conducted for historical stock data of Apple (AAPL), Tesla (TSLA), and Microsoft (MSFT). The aim was to seamlessly prepare the raw data for thorough analysis within the database.

Background:

In order to gain meaningful insights into the historical performance of Apple, Tesla, and Microsoft stocks, we embarked on a systematic ETL journey. This process involved extracting raw data from diverse sources, transforming it into a unified format, and loading it into SSMS.

Data Sources:

Apple (AAPL): Leveraged Yahoo Finance for a comprehensive dataset capturing historical stock prices, trading volumes, and related metrics.

Tesla (TSLA): Acquired stock data from a CSV file, encompassing daily opening and closing prices, high and low points, and trade volumes.

Microsoft (MSFT): Utilized the Alpha Vantage's API to fetch historical stock data.

ETL Process:

1. Data Extraction:

Raw stock data was extracted from diverse sources, laying the foundation for subsequent processing.

2. Data Transformation:

Data Merging: Combined disparate datasets for a holistic view of the stock data.

Data Conversion: Standardized data types and date formats for compatibility.

Summarization: Extracted key metrics, such as average closing prices and the summary statistics of the data.

Enrichment: Introduced additional information, AAPL_MSFT_Ratio to enrich the dataset for a more comprehensive analysis.
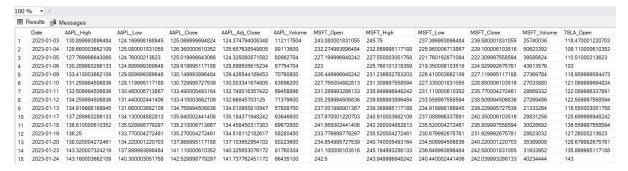
3. Loading into SSMS:

Transformed, summarized, and enriched data was loaded into SQL Server Management Studio (SSMS), providing a centralized repository for efficient storage and robust querying.

Loading Process:

```sql
-- Create a new database
CREATE DATABASE STOCK_DATA;
GO

-- Use the new database
USE STOCK_DATA;
GO

-- Create a new table in the new database
CREATE TABLE AAPL_MSFT_TSLA (
    Date DATE,
    AAPL_High FLOAT,
    AAPL_Low FLOAT,
    AAPL_Close FLOAT,
    AAPL_Adj_Close FLOAT,
    AAPL_Volume INT,
    MSFT_Open FLOAT,
    MSFT_High FLOAT,
    MSFT_Low FLOAT,
    MSFT_Close FLOAT,
    MSFT_Volume INT,
    TSLA_Open FLOAT,
    TSLA_High FLOAT,
    TSLA_Low FLOAT,
    TSLA_Close FLOAT,
    TSLA_Adj_Close FLOAT,
TSLA_Volume INT,
    AAPL_MSFT_Ratio FLOAT
);

-- Use BULK INSERT to load data from CSV into the new table
BULK INSERT AAPL_MSFT_TSLA
FROM 'C:\Users\USER\Downloads\Stock_Data.csv'
WITH (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2
);

-- Verify that data has been inserted into the new table SELECT
* FROM AAPL_MSFT_TSLA;
```

All the data present in the csv file will be shown in this way on SSMS.

Colab Link:

Data Warehouse Queries

ROLLUP Query: Calculate Total Volume for Each Company (AAPL, MSFT, TSLA) by Date

```sql
SELECT Date, SUM(AAPL_Volume) AS AAPL_Total_Volume, SUM(MSFT_Volume) AS
MSFT_Total_Volume, SUM(TSLA_Volume) AS TSLA_Total_Volume
FROM Stock_Data
GROUP BY ROLLUP (Date);
```

CUBE Query: Calculate Total Volume for Each Company (AAPL, MSFT, TSLA) by Date, and Overall Total

```sql
SELECT Date, SUM(AAPL_Volume) AS AAPL_Total_Volume, SUM(MSFT_Volume) AS
MSFT_Total_Volume, SUM(TSLA_Volume) AS TSLA_Total_Volume FROM
Stock_Data
GROUP BY CUBE (Date);
```

PIVOT Query: Display Total Volume for Each Company (AAPL, MSFT, TSLA) Across Different Dates:

```sql
SELECT *
FROM (
  SELECT Date, 'AAPL' AS Company, AAPL_Volume AS Volume
  FROM Stock_Data
  UNION ALL
  SELECT Date, 'MSFT' AS Company, MSFT_Volume AS Volume
  FROM Stock_Data
  UNION ALL
  SELECT Date, 'TSLA' AS Company, TSLA_Volume AS Volume
  FROM Stock_Data
) AS SourceTable
PIVOT (
  SUM(Volume)
  FOR Company IN ([AAPL], [MSFT], [TSLA])
) AS PivotTable;
```

Create a View for Daily Closing Prices of AAPL, MSFT, and TSLA:

```sql
CREATE VIEW DailyClosingPrices AS
SELECT Date, AAPL_Close, MSFT_Close, TSLA_Close FROM
Stock_Data;
```
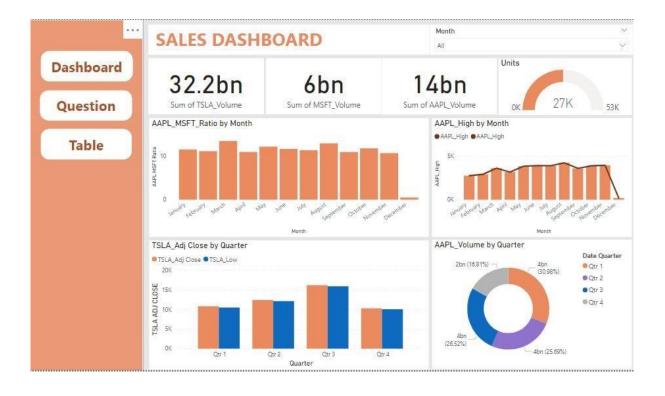
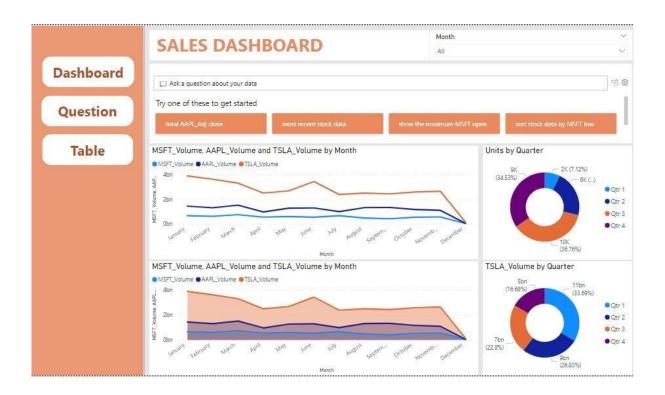### Create a Materialized View for Monthly Average Closing Prices of AAPL:

```sql
CREATE VIEW MonthlyAvgClosingPrices AS
SELECT CONVERT(VARCHAR(7), Date, 120) AS YearMonth, AVG(AAPL_Close) AS Avg_AAPL_Close
FROM Stock_Data
GROUP BY CONVERT(VARCHAR(7), Date, 120);
```

Retrieve the Top 5 Dates with the Highest Combined Volume of AAPL, MSFT, and TSLA:

```sql
SELECT TOP 5
    Date,
    (AAPL_Volume + MSFT_Volume + TSLA_Volume) AS Combined_Volume
FROM
    Stock_Data
ORDER BY
    Combined_Volume DESC;
```

Dashboard

SALES DASHBOARD

Month: All

AAPL_Volume	Units	MSFT_Volume	TSLA_Volume	MSFT_High	MSFT_Low	MSFT_Open	AAPL_Low	AAPL_High	AAPL_Close	AAPL_MSFT_Ratio	TSLA_Adj Close
24048300	225	10176649	65125200	377.97	375.14	377.33	189.25	190.90	189.97	0.50	235.45
31458200	124	12508692	119685900	340.90	336.57	339.19	191.76	193.88	192.46	0.57	279.82
35175100	144	18381253	83166000	338.54	333.70	335.19	195.28	196.73	195.61	0.58	261.07
37266700	91	16336547	105592500	309.91	307.59	309.10	171.47	173.21	172.07	0.56	166.35
37283200	139	41637739	112757300	351.89	345.07	347.11	192.92	194.44	193.62	0.55	265.28
38134500	223	28423145	122288000	376.22	371.12	375.67	189.74	191.52	190.64	0.51	241.20
38415400	227	20453112	148549900	383.00	378.16	378.35	189.40	191.08	190.40	0.50	246.72
38824100	143	25446022	84582200	337.70	333.36	336.92	195.26	196.49	196.45	0.58	267.43
39617700	224	23361184	117950600	379.79	374.97	378.00	190.83	192.93	191.31	0.51	234.21
40552600	226	22179228	112031800	380.64	376.20	376.78	188.90	190.67	189.79	0.50	236.08
41342300	131	20567159	112681500	343.74	339.02	339.56	189.78	191.19	190.54	0.56	277.90
41516200	71	23791151	116662200	291.60	286.16	289.93	164.03	165.39	165.23	0.57	187.04
41573900	132	28352729	119771100	351.43	344.31	347.59	189.63	191.18	190.69	0.55	281.38
41949600	76	26611014	140006600	284.95	278.72	282.09	163.89	165.60	165.33	0.59	162.55
42084200	159	16102024	130597900	326.08	321.46	325.50	176.25	177.68	177.23	0.55	233.19
42110300	92	26730347	98288800	313.71	309.83	309.83	171.80	173.14	172.07	0.55	166.52
42390800	192	19891180	101377900	330.30	323.18	324.75	175.80	179.05	178.99	0.54	259.67
43014200	228	28963399	135401300	384.30	377.44	383.76	188.97	192.09	189.37	0.50	244.14
43570900	96	24115664	132001400	322.59	318.01	318.60	173.45	174.71	174.20	0.54	188.87
43622600	154	16966285	88197600	325.09	320.90	323.00	177.05	179.48	177.45	0.55	232.96
43627500	217	19986506	140447600	368.47	365.90	368.22	184.21	186.03	184.80	0.50	223.71
43675600	153	18836139	98595300	324.06	320.08	321.39	177.31	179.69	179.46	0.55	239.76
43698000	193	20557094	122656000	331.10	327.67	330.96	177.95	179.72	178.39	0.54	263.62
43816600	203	31153571	118231100	331.84	327.60	331.30	171.45	173.67	173.44	0.52	216.52
43820700	163	14808482	107673700	326.15	321.72	325.66	178.55	180.59	180.19	0.56	238.82
44846000	208	20265282	118068300	339.00	334.69	338.85	167.90	170.90	170.77	0.51	200.84
44998500	37	21190042	161028300	252.82	249.39	252.46	147.45	149.17	147.92	0.59	207.63
45094300	126	28195534	120332100	342.99	335.50	337.30	189.20	192.02	191.81	0.56	276.54
45143500	98	23384887	137605100	316.50	312.61	314.73	170.52	172.42	171.84	0.55	182.90