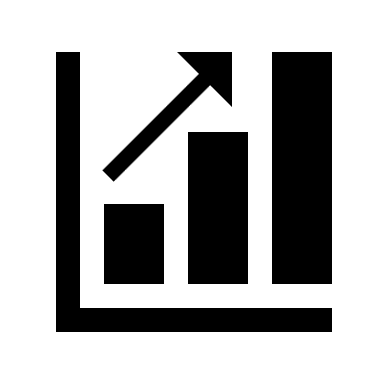
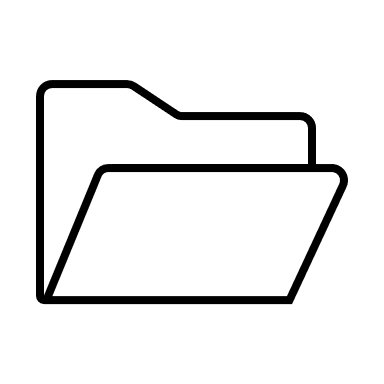
** Apple and Google Stocks Analysis in SQL**

A SQL project exploring Apple and Google stock data to uncover trends, compare performance, and analyze returns. Key insights include volume trends, price comparisons, daily returns, and rolling averages using advanced SQL techniques.



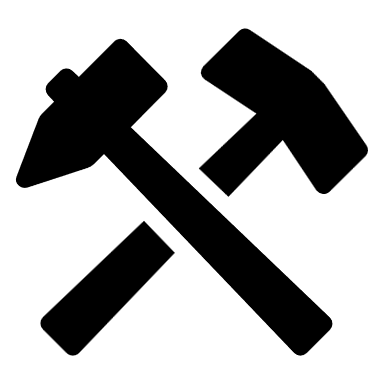
Dataset Columns

apple.csv:

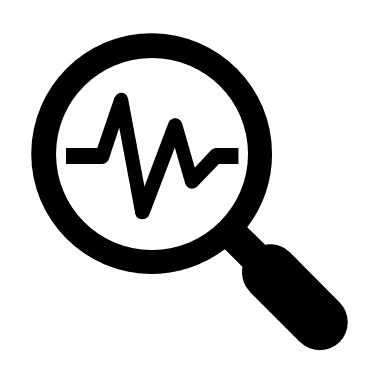
Date, Open, High, Low, Close, Adj\_close, Volume

google.csv:

Date, Open, High, Low, Close, Adj\_close, Volume

 Tools Used

* **MySQL**: for sql-based data querying
* **Git & Github**: For version control and project showcase

 Analysis Questions and Answers

1. **What is the total trading volume for Apple stock?**

**SELECT** SUM(volume) **as** total\_volume **FROM** apple;

1. **What is the average closing price for Google stock?**

**SELECT** **ROUND**(**AVG**(**close**),2) **FROM** google;

1. **Identify the highest and lowest daily High price for Apple.**

**SELECT**   
 **MAX**(**High**) **as** Highest\_High\_Price,  
 **Min**(**High**) **as** Lowest\_High\_Price  
**FROM**   
 apple

1. **What is the number of trading days in the Google dataset?**

**SELECT**   
 **COUNT**(**DISTINCT** date) **AS** trading\_days   
**FROM**   
 google;

1. **List all dates on which Apple’s closing price was greater than Google’s average closing price.**

**SELECT** date  
**FROM** apple   
**WHERE** **close** > (**SELECT**   
 **AVG**(**close**)   
 **FROM** google);

1. **Find the Google trade date(s) where the trading volume exceeded the average Apple volume.**

**SELECT** g.Date, g.Volume **AS** Google\_Volume  
**FROM**   
 google g  
**WHERE**   
 g.Volume > (  
 **SELECT** AVG(a.Volume)  
 **FROM** apple a);

1. For Apple, retrieve the date and Close when Close equals the maximum Close across all Google data

**SELECT** a.Date, a.**Close**  
**FROM** apple a  
**WHERE**   
 a.**Close** = (  
 **SELECT** MAX(g.**Close**)  
 **FROM** google g);

1. Perform an inner join on the two tables by Date and return Date, apple\_close, google\_close.

**SELECT** a.Date, a.**Close** **AS** apple\_close,  
 g.**Close** **AS** google\_close  
**FROM** apple a  
**INNER** **JOIN**   
 google g  
**ON** a.Date = g.Date;

1. **Using a left join, list all Apple trade dates and corresponding Google Close values**

**SELECT** a.Date,  
 a.**Close** **AS** apple\_close,  
 g.**Close** **AS** google\_close  
**FROM** apple a  
**LEFT JOIN** google g  
**ON** a.Date = g.Date;

1. **Which company had the higher closing price on each date? Return Date, Close\_Higher, and Winner**

**SELECT** a.Date,  
 **CASE**   
 **WHEN** a.**Close** > g.**Close** **THEN** a.**Close**  
 **ELSE** g.**Close**  
 **END** **AS** Close\_Higher,  
 **CASE**   
 **WHEN** a.**Close** > g.**Close** **THEN** 'Apple'  
 **WHEN** a.**Close** < g.**Close** **THEN** 'Google'  
 **ELSE** 'Tie'  
 **END** **AS** Winner  
**FROM** apple a  
**INNER** **JOIN** google g  
**ON** a.Date = g.Date;

1. **Use a CTE to calculate the daily return for Apple and select the top 5 dates with the highest returns**

**WITH** apple\_returns **AS** (  
 **SELECT** Date,  
 **Close**,  
 LAG(**Close**) **OVER** (**ORDER** **BY** Date) **AS** prev\_close,  
 (**Close** - LAG(**Close**) **OVER** (**ORDER** **BY** Date)) / LAG(**Close**) **OVER** (**ORDER** **BY** Date) **AS** daily\_return  
 **FROM** apple\_stock\_data  
)  
**SELECT** Date,   
 **Close**,  
 daily\_return  
**FROM** apple\_returns  
**WHERE** daily\_return **IS** **NOT** **NULL**  
**ORDER** **BY** daily\_return **DESC**  
**LIMIT** 5;

1. **Create a CTE that labels each Google trading day as HighVol or LowVol and select counts of each label**

**WITH** google\_labeled **AS** (  
 **SELECT**   
 Date,  
 Volume,  
 **CASE**   
 **WHEN** Volume > (**SELECT** **AVG**(Volume) **FROM** google) **THEN** 'HighVol'  
 **ELSE** 'LowVol'  
 **END** **AS** Vol\_Label  
 **FROM**   
 google   
)  
**SELECT** Vol\_Label, **COUNT**(\*) **AS** **Count**  
**FROM** google\_labeled  
**GROUP** **BY** Vol\_Label;

1. **With a CTE, compute monthly average close for Apple and order descending by average close**

**WITH** monthly\_avg\_close **AS** (  
 **SELECT**   
 **DATE\_FORMAT**(Date, '%Y-%m') **AS** **Month**,  
 **AVG**(**Close**) **AS** Avg\_Close  
 **FROM**   
 apple   
 **GROUP** **BY**   
 **DATE\_FORMAT**(Date, '%Y-%m')  
)  
**SELECT** **Month**, Avg\_Close  
**FROM** monthly\_avg\_close  
**ORDER** **BY** Avg\_Close **DESC**;

1. **For Google, calculate a 7-day rolling average of Close using a window function**

**SELECT**   
 Date,  
 **Close**,  
 **AVG**(**Close**) **OVER** (  
 **ORDER** **BY** Date  
 **ROWS** **BETWEEN** 6 **PRECEDING** **AND** **CURRENT** **ROW**  
 ) **AS** Rolling\_7Day\_Avg  
**FROM** google\_stock\_data;

1. **Partition Apple data by year and rank daily volumes within each year**

**SELECT** Date, **Open**, **High**, **Low**, **Close**, `Adj Close`, Volume,  
 **YEAR**(Date) **AS** **Year**,  
 **RANK**() **OVER** (**PARTITION** **BY** **YEAR**(Date) **ORDER** **BY** Volume **DESC**) **AS** Volume\_Rank  
**FROM** apple;

1. **Compute the cumulative sum of Apple’s Volume ordered by Date.**

**SELECT**   
 Date,  
 **Open**,  
 **High**,  
 **Low**,  
 **Close**,  
 [Adj **Close**],  
 Volume,  
 **SUM**(Volume) **OVER** (**ORDER** **BY** Date) **AS** Cumulative\_Volume  
**FROM**   
 apple;

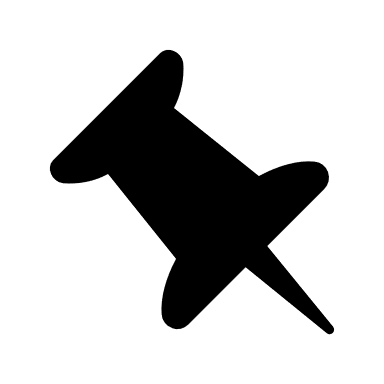
1. **Compute and compare the average daily return for Apple and Google over the entire period; indicate which performed better**

**WITH** apple\_returns **AS** (  
 **SELECT**   
 Date,  
 **Close**,  
 LAG(**Close**) **OVER** (**ORDER** **BY** Date) **AS** Prev\_Close,  
 (**Close** - LAG(**Close**) **OVER** (**ORDER** **BY** Date)) / LAG(**Close**) **OVER** (**ORDER** **BY** Date) **AS** Daily\_Return  
 **FROM** apple  
)  
**SELECT** AVG(Daily\_Return) **AS** Avg\_Daily\_Return\_Apple **FROM** apple\_returns;

**WITH** google\_returns **AS** (  
 **SELECT**   
 Date,  
 **Close**,  
 LAG(**Close**) **OVER** (**ORDER** **BY** Date) **AS** Prev\_Close,  
 (**Close** - LAG(**Close**) **OVER** (**ORDER** **BY** Date)) / LAG(**Close**) **OVER** (**ORDER** **BY** Date) **AS** Daily\_Return  
 **FROM** google  
)  
**SELECT** AVG(Daily\_Return) **AS** Avg\_Daily\_Return\_Google **FROM** google\_returns;

1. **Identify the single trading date where the difference between Google and Apple closing prices was maximum; return date, both closes, and the difference**

**SELECT**   
 a.Date,  
 a.**Close** **AS** Apple\_Close,  
 g.**Close** **AS** Google\_Close,  
 ABS(a.**Close** - g.**Close**) **AS** Price\_Difference  
**FROM**   
 apple a  
**JOIN**   
 google g **ON** a.Date = g.Date  
**ORDER** **BY**   
 Price\_Difference **DESC**  
**LIMIT** 1;

**** Key Findings

* Cumulative Volume Trend (Apple): Apple’s trading volume showed a steady cumulative increase over time, indicating strong and consistent investor activity.
* Volume Peaks by Year: By partitioning Apple’s data by year and ranking daily volumes, we identified the highest-volume trading days annually—potentially linked to earnings calls or major events.
* Daily Return Comparison: A cross-company comparison revealed that Apple had a slightly higher average daily return than Google, suggesting marginally better short-term performance.
* Max Price Divergence: The single trading date with the maximum difference in Apple and Google closing prices was pinpointed, highlighting key market divergence.



Reports & Insights

* Annual Trading Volume Analysis: Ranking daily volumes by year helped identify the most active trading days, which can guide future market timing or news-based trading strategies.
* Performance Benchmarking: Comparing Apple and Google’s average daily returns provided a clear benchmark of stock performance for short-term investment decisions.
* Market Divergence Insight: The day with the largest closing price difference offered insight into how market sentiment and valuation can sharply differ between two tech giants on the same day.
* Data-Driven Investment Signals: These combined analyses serve as a foundation for detecting signals in volume surges, return trends, and price gaps—critical for technical traders and analysts**.**



Conclusion

This analysis used SQL and Python to extract meaningful insights from Apple and Google’s historical stock data. By computing cumulative volumes, ranking peak trading days, comparing average daily returns, and identifying key price gaps, we created a well-rounded financial snapshot of both companies. These insights can support data-backed investment decisions, help time the market, and enable better comparisons across tech stocks. The approach can be scaled for broader market studies or real-time portfolio strategy development.