Project 1: data architecture and modeling

Group 19: Philip Spaier, Danni Zhang, Philipp Weissensteiner, Merlin Kägu

Business Brief

This project analyzes the stock prices of S&P 500 companies from historical data to the present. By monitoring sector composition changes within the S&P 500, we can observe long-term investment trends. Focusing on an individual company's stock allows us to track its growth or decline over time.

Stakeholders:

Finance and Risk managers – Identify stocks that need closer monitoring or rebalancing.
Investors – evaluate high-volatility or high-growth companies to inform trading and investment strategies.
Policymakers – detect early warning signals of financial stress and gather evidence for potential reforms or regulatory adjustments.

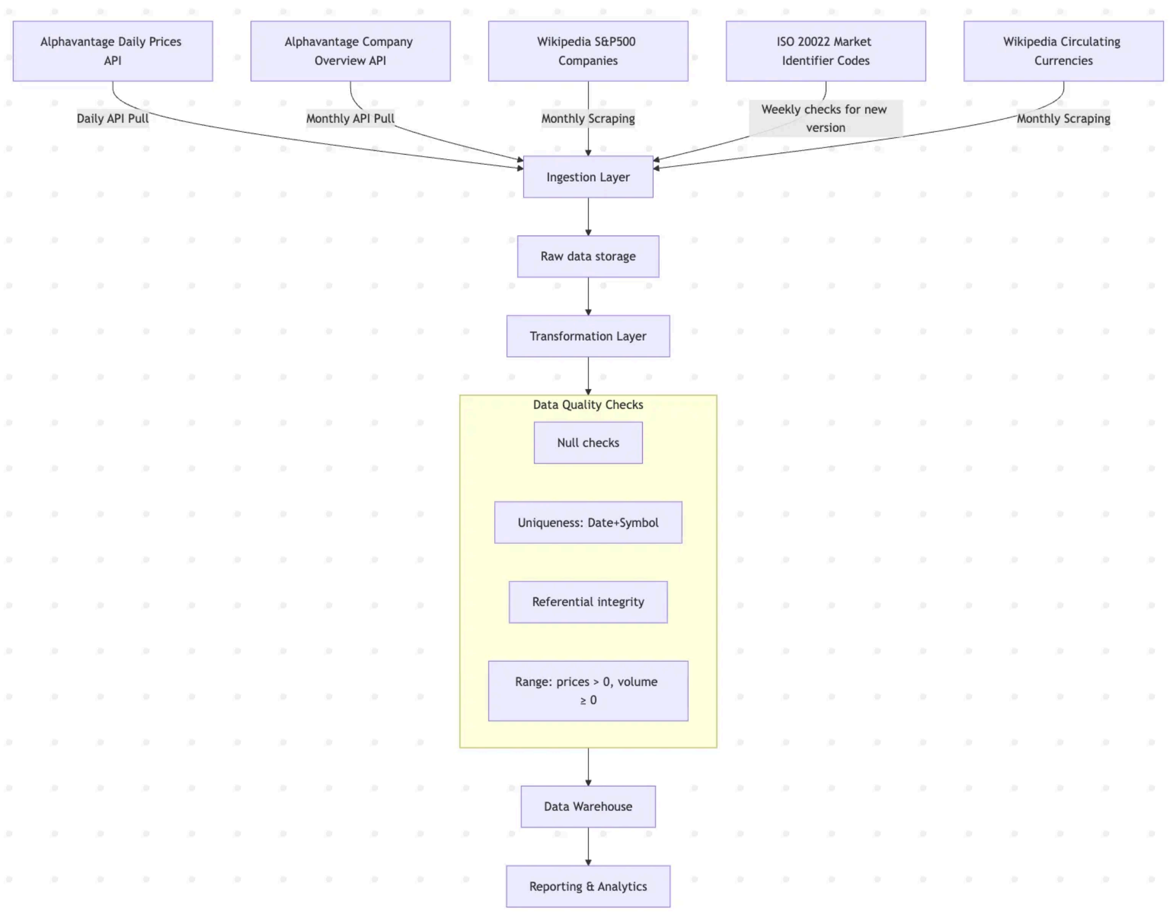
Key Metrics:

1. Price Growth (%) = ((Close_end - Close_start) / Close_start) * 100
   Close_end → Closing price at the end of the period
   Close_start → Closing price at the start of the period
2. Average Daily Return = (Close_today - Close_yesterday) / Close_yesterday
   Close_today → Closing price for the current day
   Close_yesterday → Closing price for the previous day
3. Relative Volatility = (High - Low) / Close
   High → Highest price of the day
   Low → Lowest price of the day
   Close → Closing price of the day

Business Questions:

1. Which are the top 3 most-traded companies in each sector during Q1 2025?
2. How has the average price by sectors changed over the past 10 years?
3. Which 10 companies experienced the highest price growth in Q1 2025 from highest to lowest , and what were their percentage gains?
4. Which are the top 5 companies with the highest average relative volatility in Q1 2025?
5. How has the sector composition of S&P 500 companies changed from 2000 to 2025?

Data Architecture & Data Flow



The project integrates five data sources:

1. **Alphavantage Daily Prices** (API, refreshed daily) – TIME_SERIES_DAILY example

   Comment: The derived example .csv in the Github repository only has a handful of rows. This is due to the API request limits for free users. A fully implemented production version would have hundreds of rows loaded every day, easily reaching the expected 1000 row minimum after just a couple of days.

2. **Alphavantage Company Overview** (API, refreshed monthly) – OVERVIEW example
3. **Wikipedia — List of S&P 500 Companies** (scraped into CSV, refreshed monthly) – Wikipedia S&P 500 component stocks
4. **ISO 20022 Market Identifier Codes (MIC)** (CSV file, refreshed weekly) – ISO 20022 MIC
5. **Wikipedia — List of circulating currencies** (scraped into CSV, refreshed monthly) - https://en.wikipedia.org/wiki/List_of_circulating_currencies

Ingestion Layer

- **Alphavantage data** (prices and overviews) is collected with scheduled API pulls.
- The **Wikipedia S&P 500 & currencies list** is scraped monthly and saved as CSV.
- **ISO 20022 MIC** is downloaded as CSV and checked weekly for updates.
- All ingestion tasks are orchestrated with **Apache Airflow**.
- Refresh frequencies:
  - Daily → stock prices
  - Monthly → company overviews, currencies & S&P 500 list
  - Weekly → MIC data

Storage Layer

- All ingested files (API pulls, CSVs) are stored in a **S3 object storage** in their original form for traceability.
- Data is read from raw storage directly into transformation processes.

Processing & Transformation

- Data is transformed into a **star schema** using, for example, **Spark**.
- Transformations include:
  - Parsing API/CSV data into structured tables
  - Key generation (surrogate keys for fact and dimension tables)
  - Type casting and schema alignment across sources
  - Applying data quality checks
- The **Wikipedia S&P CSV** enriches DimCompany with ticker symbols and company details
- The **currencies CSV** provides the data for the DimCurrency table
- **MIC data** is integrated into DimExchange to standardize exchange identifiers and allow consistent joins with company listings.

Data Quality Checks

- **Null checks** on numeric fields (Open, High, Low, Close, Volume)
- **Uniqueness** of (Date, Symbol) in fact table
- **Referential integrity** between fact and dimension tables (no “orphan” fact records)
- **Range checks** (prices > 0, volume ≥ 0)

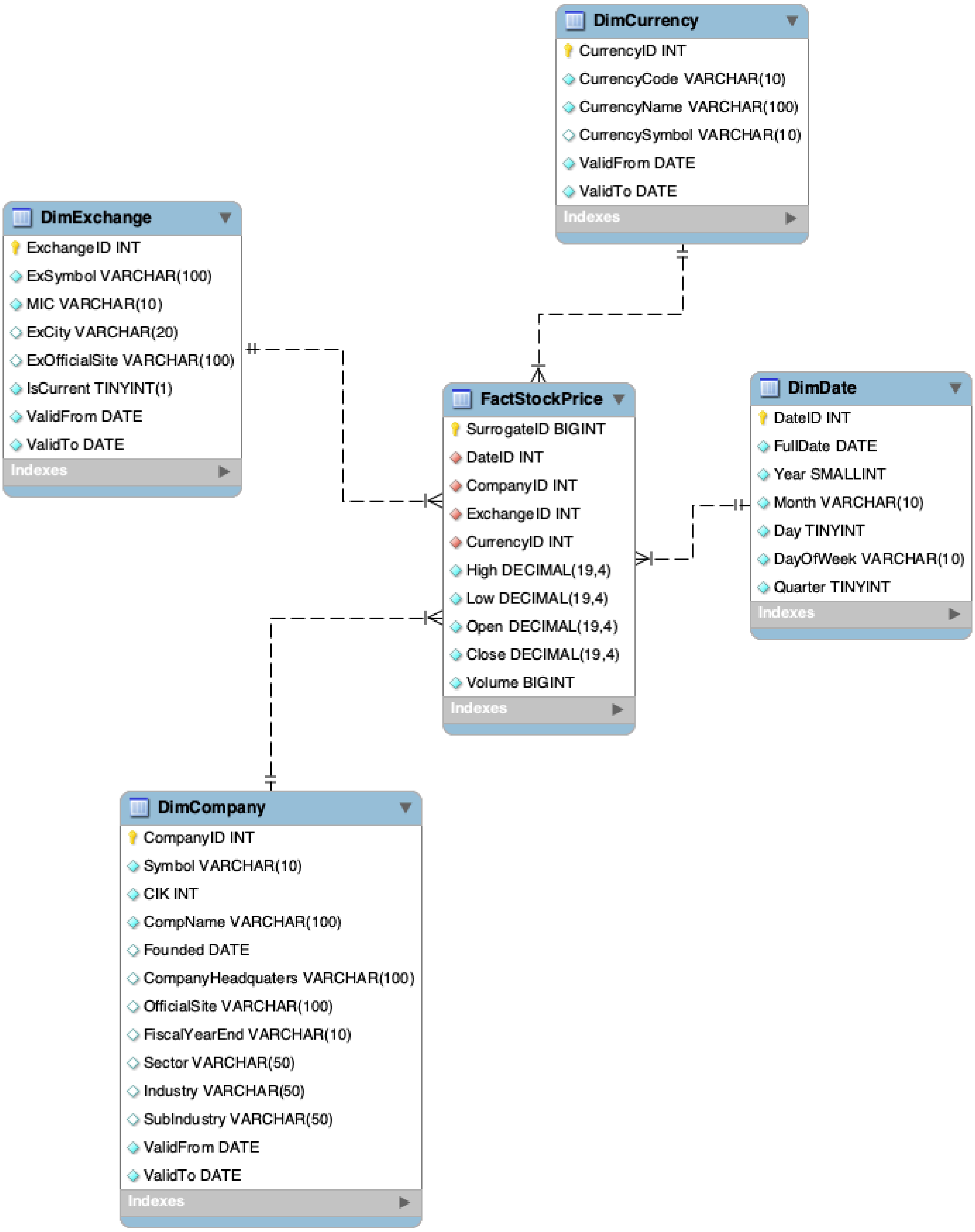
Warehouse Layer

- The dimensional model is stored in a **cloud data warehouse.**
- **Fact Table**: one row per company per trading day
- **Dimensions**:
  - DimDate (calendar breakdown: year, month, day, quarter, etc.)

    - DimCompany (SCD Type 2, with attributes from Alphavantage and Wikipedia)
    - DimExchange (with MIC integration, SCD Type 2)
    - Dim Currency (SCD Type2)
    - Sector/Industry attributes embedded in DimCompany

Reporting & Analytics

- BI tools (Power BI, Tableau, Metabase, etc.) connect to the warehouse.
- Enable dashboards and reports for:
    - Daily stock movements
    - Sector or industry comparisons
    - Exchange-level analysis using standardized MIC codes
- MIC integration ensures globally consistent exchange reporting.


DimCurrency
CurrencyID INT
CurrencyCode VARCHAR(10)
CurrencyName VARCHAR(100)
CurrencySymbol VARCHAR(10)
ValidFrom DATE
ValidTo DATE
Indexes
DimExchange
ExchangeID INT
ExSymbol VARCHAR(100)
MIC VARCHAR(10)
ExCity VARCHAR(20)
ExOfficialSite VARCHAR(100)
IsCurrent TINYINT(1)
ValidFrom DATE
ValidTo DATE
Indexes
FactStockPrice
SurrogateID BIGINT
DateID INT
CompanyID INT
ExchangeID INT
CurrencyID INT
High DECIMAL(19,4)
Low DECIMAL(19,4)
Open DECIMAL(19,4)
Close DECIMAL(19,4)
Volume BIGINT
Indexes
DimDate
DateID INT
FullDate DATE
Year SMALLINT
Month VARCHAR(10)
Day TINYINT
DayOfWeek VARCHAR(10)
Quarter TINYINT
Indexes
DimCompany
CompanyID INT
Symbol VARCHAR(10)
CIK INT
CompName VARCHAR(100)
Founded DATE
CompanyHeadquaters VARCHAR(100)
OfficialSite VARCHAR(100)
FiscalYearEnd VARCHAR(10)
Sector VARCHAR(50)
Industry VARCHAR(50)
SubIndustry VARCHAR(50)
ValidFrom DATE
ValidTo DATE
Indexes

Star Schema Summary

This segment outlines the structure of the stock price data warehouse, which employs a star schema design. The schema consists of one central fact table, FactStockPrice, surrounded by four dimension tables: DimDate, DimCompany, DimExchange, and DimCurrency.

Fact Table

The FactStockPrice table contains the measured daily stock metrics (High, Low, Open, Close, Volume) linked to the relevant dimensions via foreign keys. It is the central table used for analysis and reporting.

Dimension Tables

Dimension tables provide the contextual information for analyzing the facts.

- The DimDate table stores calendar-based attributes like year, month, day, and quarter for time-based analysis.
- The DimCompany table stores descriptive information about the companies, including their symbol, CIK, and industry classification, managing changes over time using ValidFrom and ValidTo dates.
- The DimExchange table stores details about the stock exchanges where trading occurs, identified by their Market Identifier Code (MIC).
- The DimCurrency table stores the codes and names for the currencies in which the stock prices are denominated.

The full schema can be found in the Github repository. SQL queries adapted for MySQL and Postgres, as well as a diagram exist there.

Data Dictionary

(The specific datatypes can be found in the SQL files and visualized schema in the Github repository. Constraints are also present in the SQL files.)

FactStockPrice

Granularity: Daily, on individual stock-level

- SurrogateID: Unique identifier for each record in the fact table.
- DateID: Foreign key linking to the DimDate table. This is a surrogate key and does not map to a specific column in the source data.
- CompanyID: Foreign key linking to the DimCompany table. This is a surrogate key and does not map to a specific column in the source data.
- ExchangeID: Foreign key linking to the DimExchange table. This is a surrogate key and does not map to a specific column in the source data.
- CurrencyID: Foreign key linking to the DimCurrency table. This is a surrogate key and does not map to a specific column in the source data.
- High: The highest price of the stock for the day.
    - Dataset: DataSet1(AlphaVantage)_dailyData.csv
    - Column: High
    - Description: Highest price of the stock for the day.
- Low: The lowest price of the stock for the day.
    - Dataset: DataSet1(AlphaVantage)_dailyData.csv
    - Column: Low
    - Description: Lowest price of the stock for the day.
- Open: The opening price of the stock for the day.
    - Dataset: DataSet1(AlphaVantage)_dailyData.csv
    - Column: Open
    - Description: Opening price of the stock for the day.
- Close: The closing price of the stock for the day.
    - Dataset: DataSet1(AlphaVantage)_dailyData.csv
    - Column: Close
    - Description: Closing price of the stock for the day.
- Volume: The volume of shares traded for the day.
    - Dataset: DataSet1(AlphaVantage)_dailyData.csv
    - Column: Volume
    - Description: Number of shares traded during the day.

DimDate

SCD Type: 0
Justification: The data is static and existing entries do not change.

- DateID: Unique identifier for each date. This is a surrogate key and does not map to a specific column in the source data.
- FullDate: The full date.
  - Dataset: DataSet1(AlphaVantage)_dailyData.csv
  - Column: Date
  - Description: The full date.
- Year: The year of the date.
  - Dataset: DataSet1(AlphaVantage)_dailyData.csv
  - Column: Date (derived)
  - Description: The year component of the date.
- Month: The month of the date.
  - Dataset: DataSet1(AlphaVantage)_dailyData.csv
  - Column: Date (derived)
  - Description: The month component of the date.
- Day: The day of the month.
  - Dataset: DataSet1(AlphaVantage)_dailyData.csv
  - Column: Date (derived)
  - Description: The day component of the date.
- DayOfWeek: The day of the week.
  - Dataset: DataSet1(AlphaVantage)_dailyData.csv
  - Column: Date (derived)
  - Description: The day of the week.
- Quarter: The quarter of the year.
  - Dataset: DataSet1(AlphaVantage)_dailyData.csv
  - Column: Date (derived)
  - Description: The quarter of the year.

DimCompany

SCD Type: 2
Justification: Company information can occasionally change. For financial data, historic accuracy is important. All information needs to be preserved.

- CompanyID: Unique identifier for each company. This is a surrogate key and does not map to a specific column in the source data.
- Symbol: The stock symbol of the company.
  - Dataset: DataSet2(AlphaVantage)_company_overviews.csv
  - Column: Symbol
  - Description: The stock ticker symbol.
- CIK: Central Index Key.
  - Dataset: DataSet2(AlphaVantage)_company_overviews.csv
  - Column: CIK

    - Description: Central Index Key (CIK) is a unique identifier assigned by the SEC.
- CompName: The name of the company.
    - Dataset: DataSet2(AlphaVantage)_company_overviews.csv
    - Column: Name
    - Description: The legal name of the company.
- Founded: The date the company was founded.
    - Dataset: DataSet3(Wikipedia)_sp500_components.csv
    - Column: Founded
    - Description: The year the company was founded.
- CompanyHeadquaters: The location of the company's headquarters.
    - Dataset: DataSet3(Wikipedia)_sp500_components.csv
    - Column: Headquarters Location
    - Description: The city and state of the company's headquarters.
- OfficialSite: The official website of the company.
    - Dataset: DataSet2(AlphaVantage)_company_overviews.csv
    - Column: OfficialSite
    - Description: The official website URL of the company.
- FiscalYearEnd: The end of the company's fiscal year.
    - Dataset: DataSet2(AlphaVantage)_company_overviews.csv
    - Column: FiscalYearEnd
    - Description: The month in which the company's fiscal year ends.
- Sector: The sector the company belongs to.
    - Dataset: DataSet2(AlphaVantage)_company_overviews.csv
    - Column: Sector
    - Description: The economic sector the company operates in.
- Industry: The industry the company belongs to.
    - Dataset: DataSet2(AlphaVantage)_company_overviews.csv
    - Column: Industry
    - Description: The specific industry the company operates in.
- SubIndustry: The sub-industry the company belongs to.
    - Dataset: DataSet3(Wikipedia)_sp500_components.csv
    - Column: GICS Sub-Industry
    - Description: The specific sub-industry classification.
- ValidFrom: The start date of the validity of the record.
- ValidTo: The end date of the validity of the record.

DimExchange

SCD Type: 2
Justification: Exchange information can occasionally (though rarely) change. For financial data, historic accuracy is important. All information needs to be preserved.

- ExchangeID: Unique identifier for each exchange. This is a surrogate key and does not map to a specific column in the source data.
- ExSymbol: The symbol of the exchange.
    - Dataset: DataSet2(AlphaVantage)_company_overviews.csv
    - Column: Exchange
    - Description: The symbol or name of the stock exchange.
- MIC: Market Identifier Code.
    - Dataset: DataSet4(ISO)_ISO10383_MIC.csv
    - Column: MIC
    - Description: Market Identifier Code (MIC) is a unique identification code for exchanges.
- ExCity: The city where the exchange is located.
    - Dataset: DataSet4(ISO)_ISO10383_MIC.csv
    - Column: CITY
    - Description: The city where the exchange is located.
- ExOfficialSite: The official website of the exchange.
    - Dataset: DataSet4(ISO)_ISO10383_MIC.csv
    - Column: WEBSITE
    - Description: The official website URL of the exchange.
- IsCurrent: A boolean indicating if the exchange is current.
- ValidFrom: The start date of the validity of the record.
- ValidTo: The end date of the validity of the record.

DimCurrency

SCD Type: 2
Justification: Currency information can occasionally (though rarely) change. For financial data, historic accuracy is important. All information needs to be preserved.

- CurrencyID: Unique identifier for each currency. This is a surrogate key and does not map to a specific column in the source data.
- CurrencyCode: The code of the currency.
    - Dataset: DataSet2(AlphaVantage)_company_overviews.csv
    - Column: Currency
    - Description: The ISO currency code. Relates to the currency the stock price is listed in.
- CurrencyName: The name of the currency.
    - Dataset: DataSet5(Wikipedia)_circulating_currencies.csv
    - Column: Currency
    - Description: The name of the currency.
- CurrencySymbol: The symbol of the currency.

    - Dataset: DataSet5(Wikipedia)_circulating_currencies.csv
    - Column: Symbol or abbrev.
    - Description: The symbol of the currency.
- ValidFrom: The start date of the validity of the record.
- ValidTo: The end date of the validity of the record.

SQL demo queries to the business questions:

```
-- 1. Which are the top 3 most-traded companies in each sector during Q1 2025?
WITH Q1_2025 AS (
   SELECT f.CompanyID, SUM(f.Volume) AS total_volume, d.Sector
   FROM FactStockPrice f
   JOIN DimDate dd ON f.DateID = dd.DateID
   JOIN DimCompany d ON f.CompanyID = d.CompanyID
   WHERE dd.Year = 2025 AND dd.Quarter = 1
   GROUP BY f.CompanyID, d.Sector
)
SELECT Sector, CompanyID, total_volume
FROM (
   SELECT Sector, CompanyID, total_volume,
        ROW_NUMBER() OVER (PARTITION BY Sector ORDER BY total_volume DESC)
AS rn
   FROM Q1_2025
) ranked
WHERE rn <= 3
ORDER BY Sector, total_volume DESC;
```

```
-- 2. How has the average price by sectors changed over the past 10 years?
-- using (Open+Close)/2 as “average price”
SELECT
   d.Sector,
   dd.Year,
   AVG((f.Open + f.Close) / 2) AS avg_price
FROM FactStockPrice f
JOIN DimDate dd
   ON f.DateID = dd.DateID
JOIN DimCompany d
   ON f.CompanyID = d.CompanyID
WHERE dd.Year BETWEEN EXTRACT(YEAR FROM CURRENT_DATE) - 10
              AND EXTRACT(YEAR FROM CURRENT_DATE)
GROUP BY d.Sector, dd.Year
ORDER BY d.Sector, dd.Year;
```

```
-- 3. Which 10 companies experienced the highest price growth in Q1 2025 from highest to
lowest , and what were their percentage gains?
-- (Growth = % change from first close in Q1 to last close in Q1)
```

```
WITH q1_prices AS (
    SELECT
        f.CompanyID,
        dd.FullDate,
        f.Close
    FROM FactStockPrice f
    JOIN DimDate dd
        ON f.DateID = dd.DateID
    WHERE dd.Year = 2025
      AND dd.Quarter = 1
),
first_last AS (
    SELECT
        CompanyID,
        FIRST_VALUE(Close) OVER (
            PARTITION BY CompanyID
            ORDER BY FullDate ASC
        ) AS start_price,
        LAST_VALUE(Close) OVER (
            PARTITION BY CompanyID
            ORDER BY FullDate ASC
            ROWS BETWEEN UNBOUNDED PRECEDING
                     AND UNBOUNDED FOLLOWING
        ) AS end_price
    FROM q1_prices
)
SELECT
    CompanyID,
    ROUND(((end_price - start_price) / start_price) * 100, 2) AS pct_growth
FROM first_last
GROUP BY CompanyID, start_price, end_price
ORDER BY pct_growth DESC
LIMIT 10;


-- 4. Which are the top 5 companies with the highest average relative volatility in Q1 2025?
-- (Relative volatility = (High - Low)/((High+Low)/2), averaged across Q1)
SELECT
    f.CompanyID,
    ROUND(AVG((f.High - f.Low) / NULLIF(((f.High + f.Low)/2), 0)), 4) AS
avg_rel_volatility
FROM FactStockPrice f
JOIN DimDate dd
    ON f.DateID = dd.DateID
```

```
WHERE dd.Year = 2025
  AND dd.Quarter = 1
GROUP BY f.CompanyID
ORDER BY avg_rel_volatility DESC
LIMIT 5;


-- 5. How has the sector composition of S&P 500 companies changed from 2000 to 2025?
SELECT
    d.Sector,
    dd.Year,
    COUNT(DISTINCT f.CompanyID) AS company_count
FROM FactStockPrice f
JOIN DimDate dd
    ON f.DateID = dd.DateID
JOIN DimCompany d
    ON f.CompanyID = d.CompanyID
WHERE dd.Year BETWEEN 2000 AND 2025
GROUP BY d.Sector, dd.Year
ORDER BY dd.Year, d.Sector;
```

AI Assistance

https://chatgpt.com/share/68e14744-0b70-800e-944c-913a09b39734
https://chatgpt.com/share/68e15c2f-d444-8011-b75b-b992ed1730a6
https://chatgpt.com/share/68e15c69-1d70-8011-aa69-30d70a7e8698

https://g.co/gemini/share/a4ecf31f6b94
https://g.co/gemini/share/f002630c83d0
https://g.co/gemini/share/e532071b5901
https://chatgpt.com/share/68de4bca-c778-8006-84ed-fa28d4174f4e

Github Repository

https://github.com/Penguinbeanie/DataEngineering_Project_1

Contribution

Philip Spaier (25%): Finding suitable datasets, extracting sample data from the AlphaVantage API, extracting sample data by scraping Wikipedia, implementing the star schema in SQL (MySQL and Postgres), creating a diagram of the star schema based on the implementation, setting up the Github Repository
Danni Zhang (25%): Based on the star schema, I wrote a short introduction to our project, analyzed the possible stakeholders based on their unique needs, summarized 5 business questions, and identified 3 possible KPIs.
Philipp Weissensteiner (25%): Finding suitable datasets, discussing & modelling star schema, creating data flow diagram & accompanying report
Merlin Kägu (25%): Helped come up with the business questions, proposed possible data sets, created SQL queries based on the business questions.