**Table of Contents**

1. **INTRODUCTION**

1.1. Introduction about the Project

1.2. Business Requirements

1. **Data Sources**

2.1. Wikipedia List of S&P 500 companies.

2.2. Datahub’s S&P 500 Companies with Financial Information.

2.3. Kaggle S&P 500 Stocks.

1. **Data Warehouse Data Model**

3.1. Why? “Why you choose specific schema (star schema,… etc.) ”

3.2. Dimensional Model

1. **Logical Data Mapping**
2. **Queries**
3. **Visualization**
4. **Conclusion**

**1. Introduction**

* 1. **Introduction about the Project**

**The Standard and Poor's 500 or S&P 500 is the most famous financial benchmark in the world.**

**This stock market index tracks the performance of 500 large companies listed on stock exchanges in the United States. As of December 31, 2020, more than $5.4 trillion was invested in assets tied to the performance of this index.**

**1.2. Business Requirements**

**This project seeks to analyse the data of the S&P 500 stock market and its firms using a variety of KPIs in order to derive insights that could aid investors in better understanding the market and identifying profitable investment opportunities.**

**Analysis of stocks will be useful for new investors to invest in stock market based on the various KPIs like Market capitalization, earning per share, and stock price etc. considered by dashboards.**

**2. Data Sources**

**2.1. Wikipedia List of S&P 500 companies**

[**Link**](https://en.wikipedia.org/wiki/List_of_S%26P_500_companies)

**Date of creation: Unknown.**

**Descriptions:**

**The data source comprises 503 common stocks which are issued by 500 large-cap companies traded on American stock exchanges (including the 30 companies that compose the Dow Jones Industrial Average).**

**The data shows the symbol, sector, sub-industry, Date first added**

**ETL:**

**The data was extracted using** [**https://wikitable2csv.ggor.de/**](https://wikitable2csv.ggor.de/) **website and saved as a CSV file.**

**We dropped the unnecessary columns like (SEC fillings).**

**We had 2 columns having multi-values rows like (Date first added, Founded) we separated each of them to two columns based on the space delimiter and then we dropped the two second columns.**

**2.2. Datahub’s S&P 500 Companies with Financial Information**

[**Link**](https://datahub.io/core/s-and-p-500-companies-financials#readme)

**Date of creation: 4 years ago.**

**Description:**

**List of companies in the S&P 500 (Standard and Poor’s 500). The S&P 500 is a free-float, capitalization-weighted index of the top 500 publicly listed stocks in the US (top 500 by market cap). The dataset includes a list of all the stocks contained therein and associated key financials such as price, market capitalization, earnings, price/earnings ratio, price to book etc.**

**Notes: Market Capitalization and EBIDTA are in Billions.**

**ETL:**

**The data was downloaded as a CSV file. No transformation was required.**

**2.3. Kaggle S&P 500 Stocks**

[**Link**](https://www.kaggle.com/datasets/andrewmvd/sp-500-stocks)

**Date of creation: Unknown.**

**Description:**

## The data consists of 3 tables (sp500\_companies.csv, sp500\_index.csv, sp500\_stocks.csv).

## The data is daily updated from 2009 till now.

**ETL:**

**The data was downloaded as a CSV file. The data from 2009 till 2017 was dropped.**

**3. Data Warehouse Data Model**

3.1. Why?

We chose “Galaxy schema” because we have two fact tables linked to four dimensions tables with different granularity levels.

Advantages:

1. Its multidimensional nature helps in structuring complex Database systems efficiently.
2. Minimum or no redundancy, because of Normalization.
3. This is a flexible Schema, considering the complexity of the system.
4. Data Quality will be fine, as Normalization provides the advantage for well-defined tables/ data formats.
5. When queried with Joins, clear & accurate data can be extracted.
6. High Data quality & accuracy helps in creating exceptional Reporting & Analytical results.

Disadvantages:

1. Galaxy schema can be Complex in structure.
2. Working on this schema is tedious, as the complexity in both Schema and database system makes it more intricate all together.
3. Data retrieval is done with multi-level joins combined with conditional expressions.
4. The number of levels of normalization is expected, depending on the depth of the given database.
5. Maintenance and support tasks get difficult as Galaxy schema is applied for larger database systems with complex structures.
6. Large storage space is required for its larger design arrangement and detailed querying process.
7. The analysis gets difficult, as it has no limitation on how many fact and dimension tables it can have.

3.2. Dimensional Model

1. Calendar dimension table: It contains: (Date\_ID, Full date, Month, Quarter, Year) columns.
2. Sector dimension table: It contains: (Symbol, Sector\_ID, Sector, Sub-sector) columns.
3. Location dimension table: It contains: (Symbol, Location\_ID, Country, State, City) columns.
4. Company dimension table: It contains: (Stock\_ID, Symbol, Name, Founded, Date\_First\_Added, Cik).
5. Stock\_Daily\_Fact fact table: It contains: (Open, Low, High, Close, Adj\_Close, Stock\_ID, Sector\_ID, Location\_ID, Date\_ID, Symbol\_Location, Symbol\_sector, Volume) columns.
6. constituents\_Fact fact table: It contains: (Symbol\_Location, Company\_Name, Sector, Price, Price\_Per\_Earnings, Dividend\_Yield, Earning\_Per\_Share, Week\_high\_52, Week\_low\_52, Market\_cap, Ebitda, Price\_to\_sales, Price\_per\_booking, Year, Stock\_ID, Sector\_ID, Location\_ID, Symbol\_Sector) columns.
7. Sp\_500\_index\_dimension table: It contains: (Full\_Date, Price) columns.

Diagram

Description automatically generated

**4. Logical Data Mapping**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Source Table Name | Column | Data Type | PK | Table Type | Data Source | Transformation | Target Table Name |
| Generated | Date\_ID | int | (Y) | Dimension | Kaggle | None | Calendar dimension |
| Sp\_500\_index | Full date | Date | (N) | Dimension | Kaggle | None | Calendar dimension |
| Sp\_500\_index | Month | int | (N) | Dimension | Kaggle | Extracted  From  Full  Date | Calendar dimension |
| Sp\_500\_index | Quarter | int | (N) | Dimension | Kaggle | Extracted  From  Full  Date | Calendar dimension |
| Sp\_500\_index | Year | int | (N) | Dimension | Kaggle | Extracted  From  Full  Date | Calendar dimension |
| Generated | Sector\_ID | int | (Y) | Dimension | Wiki | None | Sector dimension |
| Sp\_500\_wiki | Symbol | string | (Y) | Dimension | Wiki | None | Sector dimension |
| Sp\_500\_wiki | Sector | string | (N) | Dimension | Wiki | None | Sector dimension |
| Sp\_500\_wiki | Sub-sector | string | (N) | Dimension | Wiki | None | Sector dimension |
| Sp\_500\_wiki & Sp\_500\_Comp | Symbol | string | (Y) | Dimension | Wiki & Kaggle | None | Location dimension |
| Sp\_500\_wiki & Sp\_500\_Comp | Location\_ID | int | (Y) | Dimension | Wiki & Kaggle | None | Location dimension |
| Sp\_500\_wiki & Sp\_500\_Comp | Country | string | (N) | Dimension | Wiki & Kaggle | None | Location dimension |
| Sp\_500\_wiki & Sp\_500\_Comp | State | string | (N) | Dimension | Wiki & Kaggle | Extracted from  head  quarter | Location dimension |
| Sp\_500\_wiki & Sp\_500\_Comp | City | String | (N) | Dimension | Wiki & Kaggle | Extracted from  head  quarter | Location dimension |
| Sp\_500\_index | Full\_Date | Date | (Y) | Dimension | Kaggle | None | Sp\_500\_index |
| Sp\_500\_index | Price | float | (N) | Dimension | Kaggle | None | Sp\_500\_index |
| Sp\_500\_wiki | Stock\_ID | int | (Y) | Dimension | Wiki | None | Company dimension |
| Sp\_500\_wiki | Symbol | string | (Y) | Dimension | Wiki | None | Company dimension |
| Sp\_500\_wiki | Name | string | (N) | Dimension | Wiki | None | Company dimension |
| Sp\_500\_wiki | Founded | int | (N) | Dimension | Wiki | None | Company dimension |
| Sp\_500\_wiki | Date\_First\_Added | Date | (N) | Dimension | Wiki | None | Company dimension |
| Sp\_500\_wiki | Cik | int | (N) | Dimension | Wiki | None | Company dimension |
| Sp\_500\_stocks | Open | float | (N) | Fact | Kaggle | None | Stock\_Daily\_Fact |
| Sp\_500\_stocks | Low | float | (N) | Fact | Kaggle | None | Stock\_Daily\_Fact |
| Sp\_500\_stocks | High | float | (N) | Fact | Kaggle | None | Stock\_Daily\_Fact |
| Sp\_500\_stocks | Close | float | (N) | Fact | Kaggle | None | Stock\_Daily\_Fact |
| Sp\_500\_stocks | Adj\_Close | float | (N) | Fact | Kaggle | None | Stock\_Daily\_Fact |
| Sp\_500\_stocks | Stock\_ID | int | (N) | Fact | Kaggle | None | Stock\_Daily\_Fact |
| Sp\_500\_stocks | Sector\_ID | int | (N) | Fact | Kaggle | None | Stock\_Daily\_Fact |
| Sp\_500\_stocks | Location\_ID | int | (N) | Fact | Kaggle | None | Stock\_Daily\_Fact |
| Sp\_500\_stocks | Date\_ID | Date | (N) | Fact | Kaggle | None | Stock\_Daily\_Fact |
| Sp\_500\_stocks | Symbol\_Location | string | (N) | Fact | Kaggle | None | Stock\_Daily\_Fact |
| Sp\_500\_stocks | Symbol\_sector | string | (N) | Fact | Kaggle | None | Stock\_Daily\_Fact |
| Sp\_500\_stocks | Volume | int | (N) | Fact | Kaggle | None | Stock\_Daily\_Fact |
| constituents | Symbol\_Location | string | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Company\_Name | string | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Sector | string | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Price | float | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Price\_Per\_Earnings | float | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Dividend\_Yield | float | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Earning\_Per\_Share | float | (N) |  | Datahub | None |  |
| constituents | Week\_high\_52 | float | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Week\_low\_52 | float | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Market\_cap | float | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Ebitda | float | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Price\_to\_sales | float | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Price\_per\_booking | float | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Year | int | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Stock\_ID | int | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Sector\_ID | int | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Location\_ID | int | (N) | Fact | Datahub | None | constituents\_Fact |
| constituents | Symbol\_Sector | string | (N) | Fact | Datahub | None | constituents\_Fact |

**5. Queries**

Graphical user interface, table

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Text, table

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, table

Description automatically generated

**6. Visualization**

Graphical user interface

Description automatically generated with medium confidence

**This report page shows the S&P 500 Index's different stocks' daily behaviour in the past 5 years with options to choose the date or the desired companies (stocks). It also shows the most recent price of each stock and the year-to-date change percentage per stock.**

Timeline

Description automatically generated with low confidence

**This report page shows the different sectors/ sub-industries/ companies of the stocks and their market capitalization which is visualized through a tree map that highlights the top performing stocks in the S&P 500 index per Sub-Industry and per companies. The investor can select the sector he is interested investing in it and the tree map is changed interactively with his choice showing the top performing stocks. The tree map also contains a very informative tooltip that shows the stock price's behaviour in 2022 in terms of change percent from the beginning of the year till now, and a sparkline showing the trend of the stock price during the year which gives the investor insights about whether it is worth investing in this stock or not.**

Graphical user interface, application

Description automatically generated

**This report page has more advanced KPI's targeting more expert investors who understands the stock market terms (Ex. Dividend Yield, Book Per Price,) helping them making the right choice of investment. The dashboard allows the investor to choose or search for a specific company to see it's KPI's and helping him make the right decision. It also shows information about the company (Location - Headquarters - Date Founded...).**

Table

Description automatically generated

**7. Conclusion**

As a summary for our work, we defined the business needs, then we collected the datasets from different sources to help us find meaningful solutions to these needs.

We loaded the data into Postgresql RDMS, cleaned it and integrated it to build our model.

We executed non-trivial SQL queries to get the data of interest.

We connected Microsoft power Bi to our database to visualize our data through representative dashboards.

Future work: Index price prediction using RNN in machine learning.