DWBI Project

Architect, Populate and Explore

Data Warehouse For Stock Market Analysis

(Graduation Project Documentation)

Submitted By:

Abdelrahman Gamal

Mazen Magdy

Ahmed Osama Maher

Nov. 2022

Table of Contents

1. Introduction

1.1 Introduction

Stock analysis refers to the method that an investor or trader uses to evaluate and investigate a particular trading instrument, investment sector, or the stock market as a whole. Stock analysis is also called equity analysis or market analysis. Investors or traders make buying or selling decisions based on stock analysis information.

Also, the Stock analysis helps traders to gain an insight into the economy, stock market, or securities. It involves studying the past and present market data and creating a methodology to choose appropriate stocks for trading. Stock analysis also includes the identification of ways of entry into and exit from investments.

The S&P 500 tracks the stock performance and market capitalization of the roughly 500 companies included in the index which reflects almost 80% of the whole USA economy, measuring the value of the stock of those companies.

The index is one of the factors in the computation of the conference board leading economic index, used to forecast the direction of the economy.

1.2 Business process

- Reporting Stock Market Daily Statistics
- Reporting Top And Lowest Performers
- Tracking Stock Market Changes

2. Data Sources

2.1 S&P 500 Component Stocks

Link Source: List of S&P 500 companies - Wikipedia

Description:

The **S&P500** stock market index is maintained by S&P Dow Jones Indices. It 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 index includes about 80 percent of the American equity market by capitalization. It is weighted by free-float market capitalization, so more valuable companies account for relatively more weight in the index. The index constituents and the constituent weights are updated regularly using rules published by S&P Dow Jones Indices. Although called the S&P 500, the index contains 503 stocks because it includes two share classes of stock from 3 of its component companies.

Attributes:

- Symbol: Stock symbol is an abbreviation used to uniquely identify publicly traded shares of a particular on a particular stock market.

- Security: The Term "Security" Refers to a Financial Instrument that holds some type of monetary value. However, In this context it is sufficient to say it is just the name of the corporation.

- Sec Fillings : The SEC filing is a financial statement or other formal document submitted to the U.S. Securities and Exchange Commission (SEC). Public companies, certain insiders, and broker-dealers are required to make regular SEC filings. Investors and financial professionals rely on these filings for information about companies they are evaluating for investment purposes.

- GICS Sector : The Global Industry Classification Standard (GICS) is an industry taxonomy developed in 1999 by MSCI and Standard & Poor's (S&P) for use by the global financial community. The GICS structure consists of 11 sectors, 24 industry groups, 69 industries and 158 sub-industries into which S&P has categorized all major public companies.

- <u>GICS Sub-Industry</u>: It's the same as GICS Sector attributes, Except that the sub-industry has a higher level of detailed attribute.

- <u>Headquarters Location:</u> The main physical location of the corporation.

- <u>Date First Added:</u> The date that this stock was firstly into duce d to be traded in the public stock market.

- <u>CIK</u> : A Central Index Key or CIK number is a number given to an individual, company, or foreign government by the United States Securities and Exchange Commission. The number is used to identify its filings in several online databases, including EDGAR.

- <u>Founded:</u> Contains the year the corporation was first founded

Preparation and Cleansing:

Using Power Query:

Checked null values

Columns like SEC Fillings, CIK, Founded and Date Added have been removed

Separated Headquarters column to City and State.

2.2 S&P 500 Stock Data

Link Source: https://www.kaggle.com/camnugent/sandp500

Description: This data set includes daily changes for the stock for the past 5 years for all companies currently found on the S&P 500 index. This data set was first published in 2017 and was lately updated on Feb.2018.

It consists of 619,400 rows; Each row represents the daily numbers of a single stock.
There are a total of 505 corporations that were collected over 1825 days between 08/02/2013 and 07/02/2018 represented with USD currency.

Attributes:

- **Date**: The date on which a transaction was conducted.

- **Open**: The price at which a stock started on a specific day.

- **High**: The highest price reached by a certain stock on a specific day.

- **Low**: The lowest price reached by a certain stock on a specific day.

- **Volume**: The number of shares traded by a certain stock on a specific day.

- **Name**: The symbol name of a stock.

Preparation and Cleansing:

Using Power Query:

No Cleaning process needed, Data was ready to use.

2.3 S&P 500 Historical Data

Link Source: [S&P 500 Historical Data | Kaggle](#)

Description: This data set includes the historical data of S&P 500 index from 30/12/1927 to 04/11/2020. This data set was both published and updated in Nov.2020. This data set included the same attributes mentioned in the source above having the attributes aggregated on whole 505 corporate totals.

Attributes:

- **Date**: The date on which a transaction was conducted.

- **Open**: The price at which a stock started on a specific day overall the market

- **High**: The highest price reached by a certain stock on a specific day overall the market

- **Low**: The lowest price reached by a certain stock on a specific day overall the market

- **Volume**: The number of shares traded by a certain stock on a specific day overall the market

Preparation and Cleansing:

Using Power Query:

Removing the unwanted data range, and set the range from 2013 to 2018.

2.4 Constituents financials

Link source: https://datahub.io/core/s-and-p-500-companies-financials#resource-constituents-financials

Description: It contains statistics about each stock at the end of one year. Although it was not clear which year these statistics were reported, the data set was uploaded 4 years ago. It was assumed that the indicators belonged to 2018.

Attributes:

- **Symbol:** It is the of stock name abbreviation.

- **Name:** The symbol name of a stock.

- **Sector:** The Global Industry Classification Standard is an industry.

- **Price:** The price at which a stock started on a specific day.

- **Price/ Earnings:** It is the ratio for valuing a company that measures its current share price relative to its Earnings/Share. A high P/E ratio could mean that a company's stock is over-valued, or else that investors are expecting high growth rates in the future.

- **Dividend Yield:** It is the amount of money a company pays shareholders for owning a share of its stock divided by its current stock price.

- **Earnings/Share:** It is a company's net profit divided by the number of common shares it has outstanding. The resulting number serves as an indicator of a company's profitability.

- **52 Week Low:** It is the highest price at which a stock has traded during the year.

- **52 Week High:** It is the lowest price at which a stock, has traded during the year.

- **Market Cap:** It is the total value of all a company's shares of stock. It is calculated by multiplying the price of a stock by its total number of outstanding shares.

$$\textit{Market Capitalization} \; = \; \frac{\textit{Latest Closing}}{\textit{Share Price}} \; x \; \frac{\textit{Total Diluted Shares}}{\textit{Outstanding}}$$

- **EBITDA:** It is the company's earnings before Interest, Taxes, Depreciation, and Amortization.

- **Price/Sales:** It shows how much investors are willing to pay per dollar of sales for a stock. It is typically calculated by dividing the stock price by the underlying company's sales per share.

$$\text{Price-to-Sales Ratio} \; = \; \frac{\text{Market Value per Share}}{\text{Sales per Share}}$$

- **Price/Book**: It measures the market's valuation of a company relative to its book value.

$$\text{Price to Book Value Ratio} \; = \; \frac{\text{Market Price Per Share}}{\text{Book Value Per Share}}$$

Preparation and Cleansing:

Using Power Query:

Removed the unwanted Columns like SEC Fillings

2.5 Daily Gold Price Historical Data

Link Source: [Gold Price Prediction Dataset | Kaggle](#)

Description: Historically gold coinage was widely used as currency; when paper money was introduced, it typically was a receipt redeemable for gold coin or bullion. In a monetary system known as the gold standard, a certain weight of gold was given the name of a unit of currency. For a long period, the United States government set the value of the US dollar so that one troy ounce was equal to $20.67 ($0.665 per gram), but in 1934 the dollar was devalued to $35.00 per troy ounce ($0.889/g). By 1961, it was becoming hard to maintain this price, and a pool of US and European banks agreed to manipulate the market to prevent further currency devaluation against increased gold demand.

This data was first added on May 2022 and was last updated on Sep. 2022

Attributes:

- **Date**: The date on which a transaction was conducted.

- **Open**: The price at which gold price started on a specific day**.**

- **High**: The highest price reached by gold price on a specific day.

- **Low**: The highest price reached by a certain stock on a specific day.

- **Close**: The price at which a gold price ended on a specific day.

- **Volume**: The number of traded gold on a specific day.

Preparation and Cleansing:

Using Power Query:

Removing the unwanted data range, and set the range from 2013 to 2018.

3. Data Warehouse Data Model

3.1 Logical Model Structure

We used Galaxy Schema to implement DWH model

- **Business Process:** Stock market analysis and portfolio recommendations.
- **Granularity:** Per day.
- **Dimensions:**

    - **Date**: Contains full date, Year, Month, Quarter, Week, Day
    - **Company**: keeps information about corporations that are components of the S&P500 index.

- **Facts:**

    - **Stock Market:** Has information as price, price/earnings, dividend yield, earning/share, 52-week high/low, market cap, Ebitda and price/sale.
    - **Daily Monitoring**: Contains information about the daily change for every stock like: open price, low, high, close, and volume
    - **SP500 Index:** Contains information about the daily change for every overall companies open price, low, high, close, and volume
    - **Gold:** Contains the change in gold price for every day

3.2 Bus Matrix and Model

Facts\Dimensions	Date Dimension	Company Dimension
Stock Market Fact		✓
SP500 Index Fact	✓	
Daily Monitoring Fact	✓	✓
Gold Fact	✓	

STOCK.STOCK_MARKET_FACT

F	SYMBOL	VARCHAR2 (100 BYTE)
	PRICE	NUMBER (20,2)
	PRICE_EARNING	NUMBER (20,2)
	DIVIDEND_YIELD	NUMBER (20,2)
	EARNING_SHARE	NUMBER (20,2)
	YEAR_HIGH	NUMBER (20,2)
	YEAR_LOW	NUMBER (20,2)
	MARKET_CAP	NUMBER (30,2)
	EBITDA	NUMBER (20,2)
	PRICE_PER_SALE	NUMBER (20,2)
	PRICE_PER_BOOK	NUMBER (20,2)

SYM_FK (SYMBOL)

STOCK.COMPANY_DIMENSION

P *	SYMBOL	VARCHAR2 (100 BYTE)
	COMPANY_NAME	VARCHAR2 (100 BYTE)
	SECTOR	VARCHAR2 (100 BYTE)
	INDUSTRY	VARCHAR2 (100 BYTE)
	CITY	VARCHAR2 (100 BYTE)
	STATE	VARCHAR2 (100 BYTE)

COMPANY_DIMENSION_PK (SYMBOL)

STOCK.DAILY_MONITORING_FACT

F	FULL_DATE	DATE
F	SYMBOL	VARCHAR2 (100 BYTE)
	OPEN	NUMBER (30,20)
	HIGH	NUMBER (30,20)
	LOW	NUMBER (30,20)
	CLOSE	NUMBER (30,20)
	VOLUME	NUMBER (30,2)

DATE_FK (FULL_DATE)
SYMB_FK (SYMBOL)

STOCK.SP_IDNDEX_FACT

F	FULL_DATE	DATE
	OPEN	NUMBER (20,7)
	HIGH	NUMBER (20,7)
	LOW	NUMBER (20,7)
	CLOSE	NUMBER (20,7)
	VOLUME	NUMBER (20,2)

SP_IDNDEX_FACT_FK1 (FULL_DATE)

STOCK.DATE_DIMENSION

P *	FULL_DATE	DATE
	YEAR	VARCHAR2 (100 BYTE)
	MONTH	VARCHAR2 (100 BYTE)
	MONTH_NAME	VARCHAR2 (100 BYTE)
	WEEK	VARCHAR2 (100 BYTE)
	DAY	VARCHAR2 (100 BYTE)
	DAY_NAME	VARCHAR2 (100 BYTE)
	QUARTER	VARCHAR2 (100 BYTE)

DATE_DIMENSION_PK (FULL_DATE)

STOCK.GOLD_FACT

F	FULL_DATE	DATE
	OPEN	NUMBER (20,2)
	HIGH	NUMBER (20,2)
	LOW	NUMBER (20,2)
	CLOSE	NUMBER (20,2)
	VOLUME	NUMBER (20,2)

DAT_FK (FULL_DATE)

4. Physical model & Importing data

We created tables on SQL developer, we have Six tables in our data model.

First, we create a new Database called Stock, then we created the tables.

First table: Company dimension table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SYMBOL	VARCHAR2(100 BYTE)	No	(null)	1	(null)
2	COMPANY_NAME	VARCHAR2(100 BYTE)	Yes	(null)	2	(null)
3	SECTOR	VARCHAR2(100 BYTE)	Yes	(null)	3	(null)
4	INDUSTRY	VARCHAR2(100 BYTE)	Yes	(null)	4	(null)
5	CITY	VARCHAR2(100 BYTE)	Yes	(null)	5	(null)
6	STATE	VARCHAR2(100 BYTE)	Yes	(null)	6	(null)

Second table: Date dimension table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	FULL_DATE	DATE	No	(null)	1	(null)
2	YEAR	VARCHAR2(100 BYTE)	Yes	(null)	2	(null)
3	MONTH	VARCHAR2(100 BYTE)	Yes	(null)	3	(null)
4	MONTH_NAME	VARCHAR2(100 BYTE)	Yes	(null)	4	(null)
5	WEEK	VARCHAR2(100 BYTE)	Yes	(null)	5	(null)
6	DAY	VARCHAR2(100 BYTE)	Yes	(null)	6	(null)
7	DAY_NAME	VARCHAR2(100 BYTE)	Yes	(null)	7	(null)
8	QUARTER	VARCHAR2(100 BYTE)	Yes	(null)	8	(null)

Third table: Daily monitoring fact table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	FULL_DATE	DATE	Yes	(null)	1	(null)
2	SYMBOL	VARCHAR2(100 BYTE)	Yes	(null)	2	(null)
3	OPEN	NUMBER(30,20)	Yes	(null)	3	(null)
4	HIGH	NUMBER(30,20)	Yes	(null)	4	(null)
5	LOW	NUMBER(30,20)	Yes	(null)	5	(null)
6	CLOSE	NUMBER(30,20)	Yes	(null)	6	(null)
7	VOLUME	NUMBER(30,2)	Yes	(null)	7	(null)
8	SP500_INDEX	NUMBER(30,2)	Yes	(null)	8	(null)

Forth table: Gold fact table

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
FULL_DATE	DATE	Yes	(null)	1	(null)
OPEN	NUMBER(20,2)	Yes	(null)	2	(null)
HIGH	NUMBER(20,2)	Yes	(null)	3	(null)
LOW	NUMBER(20,2)	Yes	(null)	4	(null)
CLOSE	NUMBER(20,2)	Yes	(null)	5	(null)
VOLUME	NUMBER(20,2)	Yes	(null)	6	(null)

Fifth table: Stock market fact

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
SYMBOL	VARCHAR2(100 BYTE)	Yes	(null)	1	(null)
PRICE	NUMBER(20,2)	Yes	(null)	2	(null)
PRICE_EARNING	NUMBER(20,2)	Yes	(null)	3	(null)
DIVIDEND_YIELD	NUMBER(20,2)	Yes	(null)	4	(null)
EARNING_SHARE	NUMBER(20,2)	Yes	(null)	5	(null)
YEAR_HIGH	NUMBER(20,2)	Yes	(null)	6	(null)
YEAR_LOW	NUMBER(20,2)	Yes	(null)	7	(null)
MARKET_CAP	NUMBER(30,2)	Yes	(null)	8	(null)
EBITDA	NUMBER(20,2)	Yes	(null)	9	(null)
PRICE_PER_SALE	NUMBER(20,2)	Yes	(null)	10	(null)
PRICE_PER_BOOK	NUMBER(20,2)	Yes	(null)	11	(null)

Sixth table: S&P index fact

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
FULL_DATE	DATE	Yes	(null)	1	(null)
OPEN	NUMBER(20,7)	Yes	(null)	2	(null)
HIGH	NUMBER(20,7)	Yes	(null)	3	(null)
LOW	NUMBER(20,7)	Yes	(null)	4	(null)
CLOSE	NUMBER(20,7)	Yes	(null)	5	(null)
VOLUME	NUMBER(20,2)	Yes	(null)	6	(null)

After creating tables on SQL developer we data was imported from Excel CSV files to SQL developer program.

5. BI Queries

1: Top and Bottom 10 stocks of market cap

```sql
select *from(

select stock_market_fact.symbol,company_dimension.company_name,market_cap,
rank()over(order by market_cap desc) as rnk_comp
from stock_market_fact,company_dimension
where stock_market_fact.symbol=company_dimension.symbol
)
where rnk_comp<=10
union
select *from(

select stock_market_fact.symbol,company_dimension.company_name,market_cap,
rank()over(order by market_cap ASC) as rnk_comp
from stock_market_fact,company_dimension
where stock_market_fact.symbol=company_dimension.symbol
)
where rnk_comp<=10;
```

2: Relation between the price of stock and gold market

```sql
--Q2 : the ratio of the price of stock and gold market close price doc
select d.full_date ,g.close as gol_close,sp.close as sp500_close
from date_dimension d, gold_fact g,sp_idndex_fact sp
where g.full_date=d.full_date
and sp.full_date=sp.full_date
order by full_date ;
```

3: S&P To Gold Correlation Ratio

```sql
--Q 3:claculate  sp gold ratio
select full_date,max(sp_gold_ratio)over(partition by full_date order by full_date asc)from(
select distinct d.full_date , round((sp.close/g.close),2) as sp_gold_ratio
from date_dimension d,sp_idndex_fact sp,gold_fact g
where sp.full_date=d.full_date and
g.full_date=d.full_date
);
```

4: Daily Performance Change Per Company

```sql
--Q 4: calculate  daily perforamnce change per company
SELECT symbol,full_Date,ROUND(high,2)as"high",ROUND(Low,2)as"Low",ROUND(open,2) as "Open",ROUND(Close,2) "CLose",volume,
ROUND(COALESCE(LAG(Close) over (order by full_DATE ASC),Close),2) as Previous_close,
COALESCE(LAG(volume) over (order by full_DATE ASC),volume) as Previous_Volume,
round((close-open)/open *100,2) as daily_CHANGE
from daily_monitoring_fact;
```

5: S&P500 Index Daily Performance

```sql
--Q 5:  calculte S&P  index  performance  daily
select * from(
SELECT full_Date,ROUND(high,2)as"high",ROUND(Low,2)as"Low",ROUND(open,2) as "Open",ROUND(Close,2) "CLose",volume,
round((close-open)/open *100,2) as daily_CHANGE, rank()over(order by round((close-open)/open *100,2) desc)as top
from sp_idndex_fact
)
where top<=10
union

select * from(
SELECT full_Date,ROUND(high,2)as"high",ROUND(Low,2)as"Low",ROUND(open,2) as "Open",ROUND(Close,2) "CLose",volume,
round((close-open)/open *100,2) as daily_CHANGE, rank()over(order by round((close-open)/open *100,2) asc)as bottom
from sp_idndex_fact
)
where bottom<=10;
```

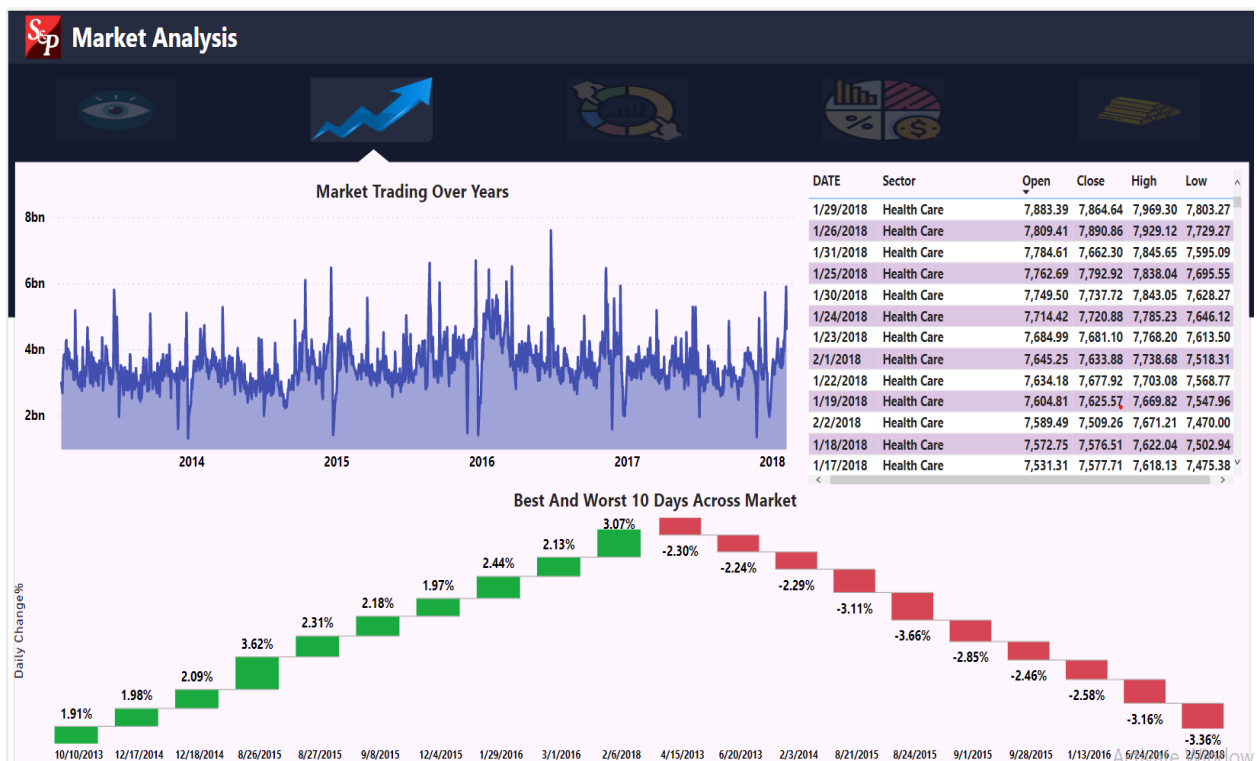## 6: Top And Bottom Prices Compared to 52 Week High, 52 Week Low

```sql
--Q8 : top 10 price compare 52 week high ,52 week low, price _earning per market cap  doc
select * from(
select c.company_name,market_cap,round(avg(price),2),round(avg(price_earning),2), round(avg(year_high),2),
round(avg(year_low),2),rank()over(order by s.market_cap desc)as rnk
from company_dimension c, stock_market_fact s
where s.symbol=c.symbol
group by c.company_name,market_cap)
where rnk<=10;
--Q worest 10 price rice compare 52 week high ,52 week low, price _earning per market cap
select * from(
select c.company_name,market_cap,round(avg(price),2),round(avg(price_earning),2), round(avg(year_high),2),
round(avg(year_low),2),rank()over(order by s.market_cap asc)as rnk
from company_dimension c, stock_market_fact s
where s.symbol=c.symbol
group by c.company_name,market_cap)
where rnk<=10;
```

7: Top And Bottom Companies Paying Dividends

```sql
select *from(
select c.company_name,s.dividend_yield, round(s.dividend_yield*price,2 )as Annual_dividend ,
round((s.dividend_yield*s.price)/s.earning_share,2) as payout_ratio,s.market_cap,
rank()over(order by market_cap desc)as top_worst10_devidend_group
from company_dimension c,stock_market_fact s
where s.symbol=c.symbol
)
where top_worst10_devidend_group<=10

union

select *from(
select c.company_name,s.dividend_yield, round(s.dividend_yield*price,2 )as Annual_dividend ,
round((s.dividend_yield*s.price)/s.earning_share,2) as payout_ratio,s.market_cap,
rank()over(order by market_cap )as top_worst10_devidend_group
from company_dimension c,stock_market_fact s
where s.symbol=c.symbol
)
where top_worst10_devidend_group<=10;
```

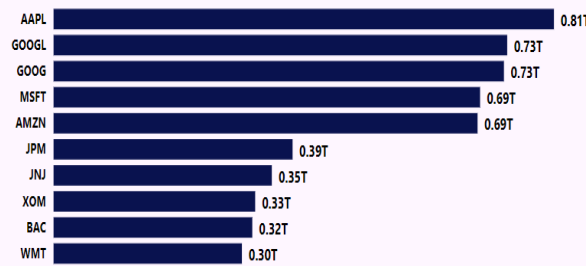# 6. Visualization Using PowerBI:
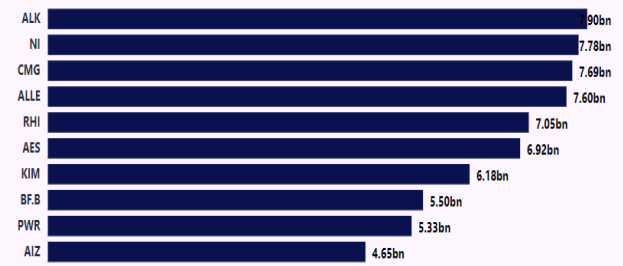
Market Cap Analysis

Company Name	Sector	Industry
3M	Communication Services	Advertising

Top 10 Market Cap Companies Across S&P500

Company	Market Cap
AAPL	0.81T
GOOGL	0.73T
GOOG	0.73T
MSFT	0.69T
AMZN	0.69T
JPM	0.39T
JNJ	0.35T
XOM	0.33T
BAC	0.32T
WMT	0.30T

Bottom 10 Market Cap Companies Across S&P500

Company	Market Cap
ALK	90bn
NI	7.78bn
CMG	7.69bn
ALLE	7.60bn
RHI	7.05bn
AES	6.92bn
KIM	6.18bn
BF.B	5.50bn
PWR	5.33bn
AIZ	4.65bn

Top 10 Market Cap Companies Prices

Bottom 10 Market Cap Companies Prices

Activate Window

Companies Analysis

TOP All   BOTTOM All

3M

Communication Services

1.62K	5.20K	283bn	22.21K	$9.19T
PRICE_PER_SALE	PRICE_PER_BOOK	Outstanding Shares	Dividend_Payout	Revenue

Price VS 52 Week High and Low

43,575.67

34.82K   51.14K

Earning Share and Dividend Yield For Each Company

● EARNING_SHARE ● DIVIDEND_YIELD

Companies (x-axis): Alphabet Inc. (Cl..., Alphabet Inc. (Class C), Apple Inc., Assurant, Alaska Air Group, JPMorgan Chase, Amazon, Walmart, Microsoft, Allegion, Robert Half, ExxonMobil, Brown-Forman, Bank of America, Quanta Services, NiSource, Chipotle Mexican Grill, Kimco Realty, AES Corporation
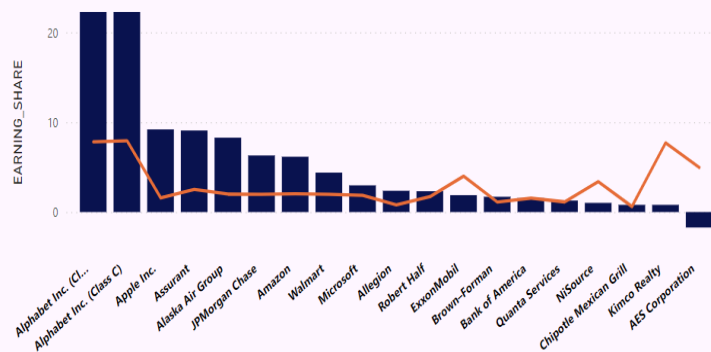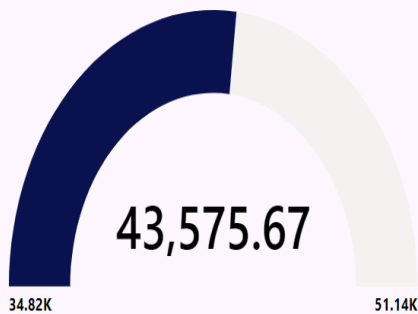
Activate Window

Correlation Of Stock And Gold Prices

● Stock_Price ● Gold_Price

3,000

2,500

2,000

1,500

1,000

2014 2015 2016 2017 2018

Gold Price By Years

34.66K (2.18%)

316.33K (19.91%)

311.53K (19.61%)

315.42K (19.85%)

319.1K (20.08%)

292.01K (18.38%)

● 2013 ● 2014 ● 2015 ● 2016 ● 2017 ● 2018

S&P 500 To Gold Ratio

2.0

1.5

1.00

1.0

0.5

0.0

2014 2015 2016 2017 2018

Stock Price By Years

72.2K (2.78%)

614.72K (23.68%)

375.61K (14.47%)

527.85K (20.33%)

486.71K (18.74%)

519.39K (20%)

● 2013 ● 2014 ● 2015 ● 2016 ● 2017 ● 2018
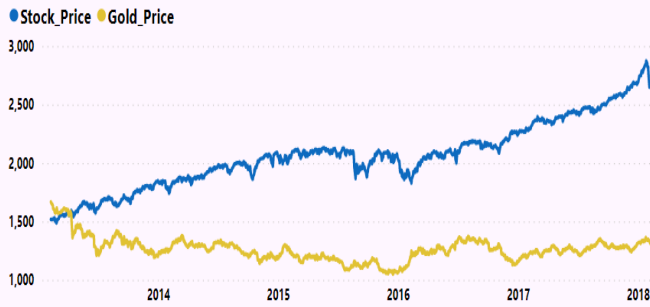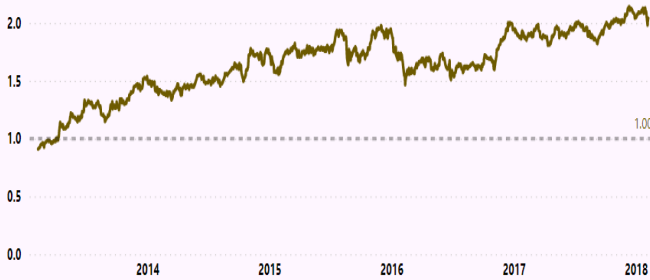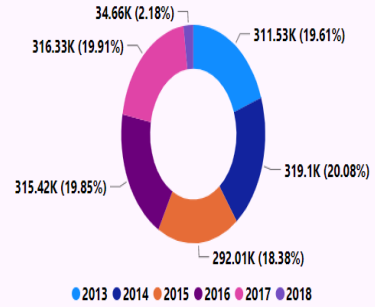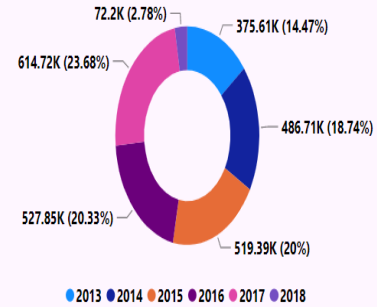
Activate Window

Conclusion

Project walkthrough:

1- Data understanding

2- Data gathering

3- Creating logical design (DWH model)

4- Physical design (Create tables on DBMS)

5- Import data on database

6- Building queries to answer the questions

7- Import files into power BI to build interactive dashboard

Technologies Used In Project :

- **Microsoft Excel** (using to import data from web and cleaning data)
- **SQL developer** (to create physical design for tables and queries)
- **Power BI** (Data visualization)

Appendix

1- SQL queries for creation tables

Company Dimension:

```
CREATE TABLE COMPANY_DIMENSION
(
 SYMBOL      VARCHAR2(100 BYTE) primary key,
 COMPANY_NAME VARCHAR2(100 BYTE),
 SECTOR      VARCHAR2(100 BYTE),
 INDUSTRY    VARCHAR2(100 BYTE),
 CITY        VARCHAR2(100 BYTE),
 STATE       VARCHAR2(100 BYTE)
)
```

Date Dimension:

```
CREATE TABLE DATE_DIMENSION
(

 FULL_DATE   DATE   primary key,
 YEAR       VARCHAR2(100 BYTE),
 MONTH      VARCHAR2(100 BYTE),
 MONTH_NAME  VARCHAR2(100 BYTE),
 WEEK       VARCHAR2(100 BYTE),
 DAY        VARCHAR2(100 BYTE),
 DAY_NAME   VARCHAR2(100 BYTE),
 QUARTER    VARCHAR2(100 BYTE)
)
```

Daily Monitoring Fact:

```
CREATE TABLE DAILY_MONITORING_FACT
(
 Full_date date,
 SYMBOL   varchar2(100 BYTE),
 OPEN      NUMBER(20,2),
 HIGH     NUMBER(20,2),
 LOW      NUMBER(20,2),
 CLOSE    NUMBER(20,2),
 VOLUME    NUMBER(20,2),
);
```

S&P Index Fact:

```
CREATE TABLE S&P_Index_Fact
(
 Full_date date,
 OPEN      NUMBER(20,2),
 HIGH      NUMBER(20,2),
 LOW       NUMBER(20,2),
 CLOSE     NUMBER(20,2),
 VOLUME    NUMBER(20,2),
);
```

Gold Fact

```
CREATE TABLE GOLD_FACT
(
 Full_date date,

 OPEN      NUMBER(20,2),
 HIGH      NUMBER(20,2),
 LOW       NUMBER(20,2),
 CLOSE     NUMBER(20,2),
 VOLUME    NUMBER(20,2)
)
```

Stock Market Fact

```
CREATE TABLE STOCK_MARKET_FACT
(
 SYMBOL          varchar2(100 Byte),
 PRICE           NUMBER(20,2),
 PRICE_EARNING       NUMBER(20,2),
 DIVIDEND_YIELD      NUMBER(20,2),
 EARNING_SHARE       NUMBER(20,2),
 year_high        number(20,2),
 year_low         number(20,2),
 MARKET_CAP        NUMBER(20,2),
 EBITDA          NUMBER(20,2),
 PRICE_per_sale     NUMBER(20,2),
 price_per_book      number(20,2)
 )
```

```sql
ALTER TABLE DAILY_MONITORING_FACT ADD (
 CONSTRAINT DATE_FK
 FOREIGN KEY (full_date)
 REFERENCES DATE_DIMENSION(full_date),
 CONSTRAINT SYMB_FK
 FOREIGN KEY (SYMBOL)
 REFERENCES COMPANY_DIMENSION(symbol));

ALTER TABLE GOLD_FACT ADD (
 CONSTRAINT DAT_FK
 FOREIGN KEY (full_date)
 REFERENCES DATE_DIMENSION(full_date));

ALTER TABLE STOCK_MARKET_FACT ADD (
 CONSTRAINT SYM_FK
 FOREIGN KEY (SYMBOL)
 REFERENCES COMPANY_DIMENSION(symbol));
```

DAX Formulas:

```
Change% = ((Daily_Monitoring_fact[Close_Stock] -
        Daily_Monitoring_fact[Open_Stock])/Daily_Monitoring_fact[Open_Stock])


SP500 To Gold Ratio = DIVIDE(MAX(SP500_index_fact[Close]),MAX('gold_Fact'[Close_Gold]))


Revenue = ROUNDUP(Stock_market_fact[MARKET_CAP]/Stock_market_fact[PRICE_PER_SALE],0)


Outstanding Shares = ROUNDUP(Stock_market_fact[MARKET_CAP]/Stock_market_fact[PRICE],0)


Dividend_Payout = ROUND(Stock_market_fact[Annual_Dividend]/
                        Stock_market_fact[EARNING_SHARE],2)


Annual_Dividend = ROUND(Stock_market_fact[DIVIDEND_YIELD]* Stock_market_fact[PRICE],2)
```

```
Market cap group = SWITCH(

                TRUE(),
                        Stock_market_fact[MARKET_CAP] <= 10000000000, "Low Market Cap",
                        Stock_market_fact[MARKET_CAP] > 10000000000
&&  Stock_market_fact[MARKET_CAP] <= 20000000000 , "Mid Market Cap",
                        Stock_market_fact[MARKET_CAP] > 20000000000, "High Market Cap",
                        BLANK())
```

Tasks:

- Data Modeling: All Team members

- Data Preparation & Creation: Abdelrahman and ahmed osama

- creation tables and Data Loading in SQL Developer: Ahmed Osama

- Ideas of Queries and analysis: Abdelrahman and Mazen
-Implementing Queries: Ahmed Osama

- Visualization --> Mazen and Abdelrahman

-Writing Documentation: All Team

-Presentation: Ahmed osama