ITI Data Visualization Track - Graduation Project

Stock Market Analysis

March 2023

Contents

1 Introduction

The aim of this project is to design, architect, and implement a data warehouse (DWH) that stores stock market data effectively. The business requirement here is to evaluate companies' stock market performance to be considered a potential investment. Hence, the DWH needs to accommodate historical data and relevant measures. The data stored consist of various dimensions, attributes, and measures. The model shall be utilized to make dashboards that tell a story about the data. Our DWH model comprises five dimension tables and two fact tables. The DWH is designed with generality in mind. That is, it is possible to add companies from different industries, different stock markets, or different countries. It is also possible to track data in atomic and aggregated forms as our model offers both.

2 Our Story

Our story is focused on the real estate industry in Egypt. In particular, four companies; El Obour Real Estate, Amer Group, Arab Real Estate, and Arab Developers Holding. The big question is, given multiple companies in the same industry how to take a decision to invest in one of them? To answer this, one needs to evaluate the companies' performance in the stock market from various angles. First off, we want to carefully inspect how the different companies perform compared to each other. This shall be done using the available historical data and relevant measures utilizing them. Second, once a single company is chosen based on the previous step, we shall observe how this company performs within its stock market. Third, we shall track its historical performance by comparing the most recent year with the previous ones. This will give us insight into whether or not the company is making progress which in turn will allow us to consider it as a potential investment or not.

3 Data Collection

While collecting the data needed to carry out our story, we found out we needed two levels of data. A daily level that includes information about the price of a stock; closing price, open price, high, low...etc. The second level is aggregated on the quarter of each year. This includes numbers that companies release on a quarterly basis such as market capitalization, revenue, operating cashflow...etc.

3.1 Data Sources

We used the Egyptian Exchange website to select companies in the real estate industry.

3.1.1 Historical Daily Data

For the four real estate companies we focus on as well as Fawry and SWVL, we relied on investing.com to get their daily data. These data were formatted in csv files. A file for each company.

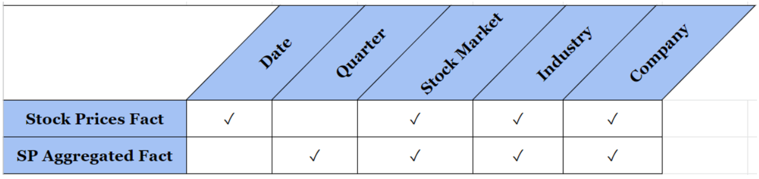
3.1.2 Historical Quarterly Data

For the quarterly aggregated data we used a pro account also on investing.com. There we can search for the company and the measure we need. Here we got the data for price ratios, outstanding shares, and market capitalization. And here we used the data in the balance sheet and the income statement. The aggregate data were mostly not tabular. They were separate values here and there that we had to haunt. Hence, they were copied into a spreadsheet for usage.

The data collected has a time range from 2019 to 2023. This means we have around 900 days and almost 17 quarters.

4 Data Modelling

We devised a galaxy schema data model with five dimension tables and two fact tables. We wanted to offer a model that can be generalized to answer any query. This meant we need an atomic-level fact table. The grain here is the day, this captures the smallest level of detail. However, most of the significant measures are reported or calculated on a quarterly basis. For this reason, we designed a one-way aggregated fact table that is dependent on the atomic fact table. The key distinction between the two tables is that the aggregate table is connected to a quarter date dimension while the atomic table is connected to a daily date dimension as shown in the model's bus matrix.



	Date	Quarter	Stock Market	Industry	Company
Stock Prices Fact	✓		✓	✓	✓
SP Aggregated Fact		✓	✓	✓	✓

Figure 1: Bus Matrix

4.1 Dimension Tables

We have the following dimensions with the corresponding columns in our model.
Date: date_key, year, quarter, month_name, month, week_of_year, week_of_month, day,

day_name
Company: company_key, name, stock_symbol, headquarters_location, type, equity_type, web_page
Stock market: smarket_key, name, country, picked_index
Industry: industry_key, name, sector
Quarter: quarter_key, quarter, year

4.2 Fact Tables

The model has these two fact tables, atomic and aggregate respectively, with their corresponding columns.

Stock_prices_fact: spf_pkey, full_date, company_key, stock_market_key, industry_key, stock_price, open, high, low, volume, com_change_pct, market_change_pct, daily_return, beta, quarter_key

Sp_aggregated_fact: agg_pkey, quarter_key, company_key, stock_market_key, industry_key, avg_sp, avg_volume, earnings, revenue, net_profit_margin, p_to_e, p_op_cashflow, price_to_book, price_to_sales, market_cap, outstanding_shares, share_turnover, current_assets, current_liabilities, inventory, cash_equivalents, current_ratio, quick_ratio, cash_ratio

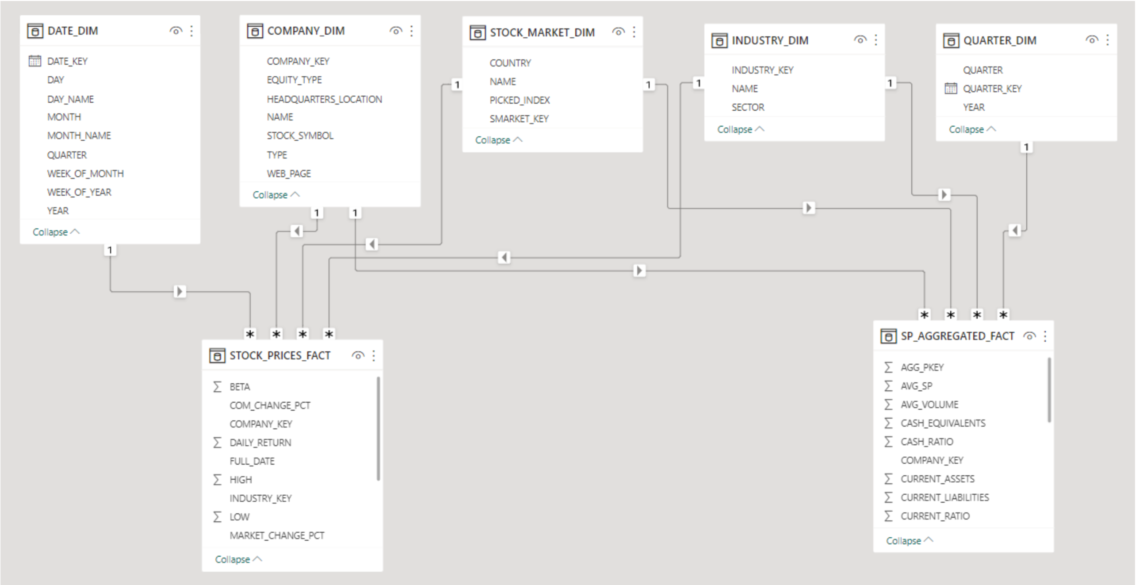


Figure 2: Logical Model

Note: Full explanation of important columns and how they are calculated, when valid, is included in the Measures section.

5 Data Loading

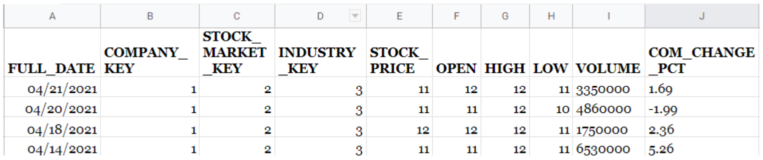
5.1 Data Insertion in the Dimension Tables

We have five dimensions in our model. Two of them are dates, the date, and the quarter dimensions. Two scripts looping on the range of values we wanted to insert are used. For the date dimension, rows are incremented by a day. For the quarter dimension, rows are incremented by 3 months (a quarter). The code for the two scripts is here.

The dimensions company, stock market, and industry are inserted with the normal insert statements since they are few. Their primary keys are generated by a sequence-trigger pair. Had there been many of them, we would have inserted them through a spreadsheet into the toad directly. Their code is also found here.

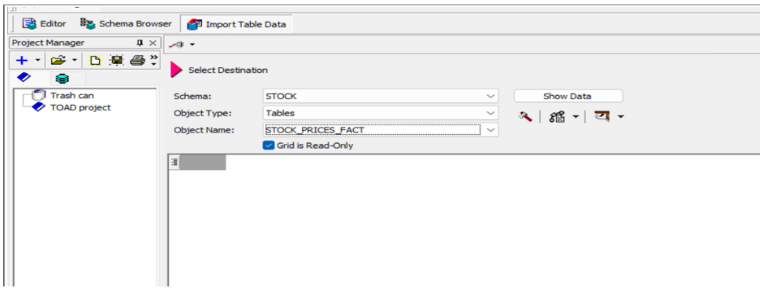
5.2 Daily Data

As stated previously, the daily data (full_date, price, open, high, low, volume, change_pct) come in a csv file for each company. In a spreadsheet, we added all the referential keys (company_key, stock_market_key, industry_key) as columns and inserted the corresponding keys for each company as shown in the figure. The spreadsheet is then imported into Toad to populate the stock_prices_fact table as shown in the figure. We used a sequence-trigger pair to generate a surrogate key that distinguishes each row.



A	B	C	D	E	F	G	H	I	J
FULL_DATE	COMPANY_ KEY	STOCK_ MARKET _KEY	INDUSTRY _KEY	STOCK_ PRICE	OPEN	HIGH	LOW	VOLUME	COM_CHANGE _PCT
04/21/2021	1	2	3	11	12	12	11	3350000	1.69
04/20/2021	1	2	3	11	11	12	10	4860000	-1.99
04/18/2021	1	2	3	12	12	12	11	1750000	2.36
04/14/2021	1	2	3	11	11	12	11	6530000	5.26

Figure 3: Adding Referential Keys

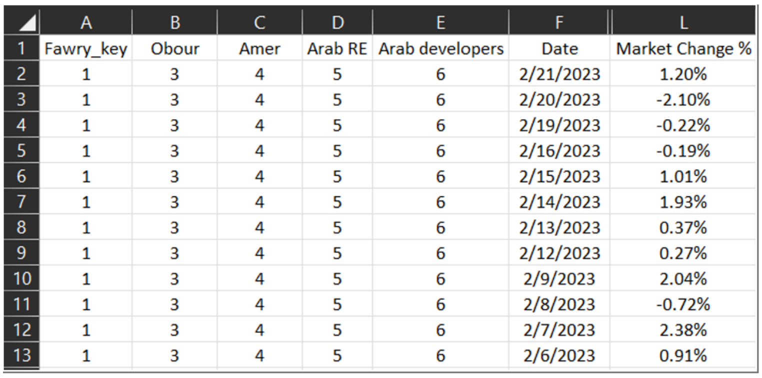


Schema: STOCK
Object Type: Tables
Object Name: STOCK_PRICES_FACT

Figure 4: Importing into Toad

5.2.1 Market_change_pct Column

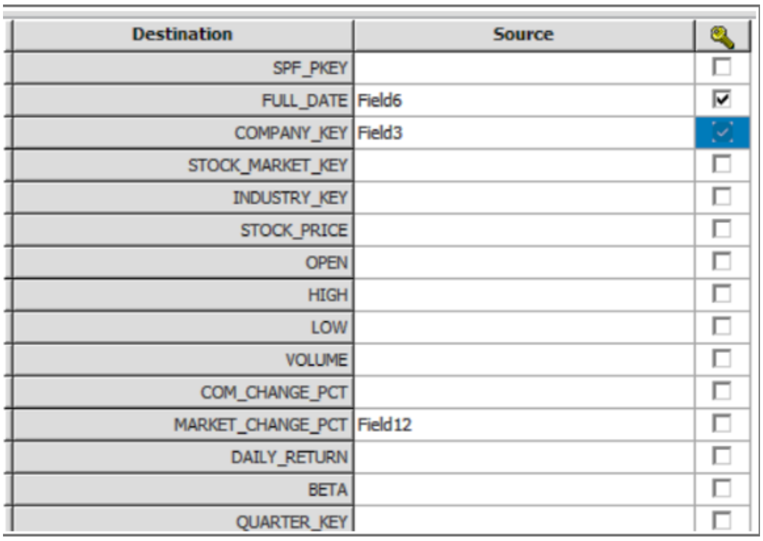
This column has the change percentage of the price of the stock market index we used to evaluate a company's performance in its market. For instance, SWVL is listed on NASDAQ stock exchange, so the market_change_pct corresponding to SWVL would be that of NASDAQ composite. NASDAQ composite is a stock market index that includes almost all stocks listed on the Nasdaq stock exchange. To accomplish this, we added the companies' keys, dates along with the corresponding market_change_pct. This way the company key and the date constitute a unique identifier we can use to insert correct values into the market_change_pct column by importing the spreadsheet in the following figure into Toad.



	A	B	C	D	E	F	L
1	Fawry_key	Obour	Amer	Arab RE	Arab developers	Date	Market Change %
2	1	3	4	5	6	2/21/2023	1.20%
3	1	3	4	5	6	2/20/2023	-2.10%
4	1	3	4	5	6	2/19/2023	-0.22%
5	1	3	4	5	6	2/16/2023	-0.19%
6	1	3	4	5	6	2/15/2023	1.01%
7	1	3	4	5	6	2/14/2023	1.93%
8	1	3	4	5	6	2/13/2023	0.37%
9	1	3	4	5	6	2/12/2023	0.27%
10	1	3	4	5	6	2/9/2023	2.04%
11	1	3	4	5	6	2/8/2023	-0.72%
12	1	3	4	5	6	2/7/2023	2.38%
13	1	3	4	5	6	2/6/2023	0.91%

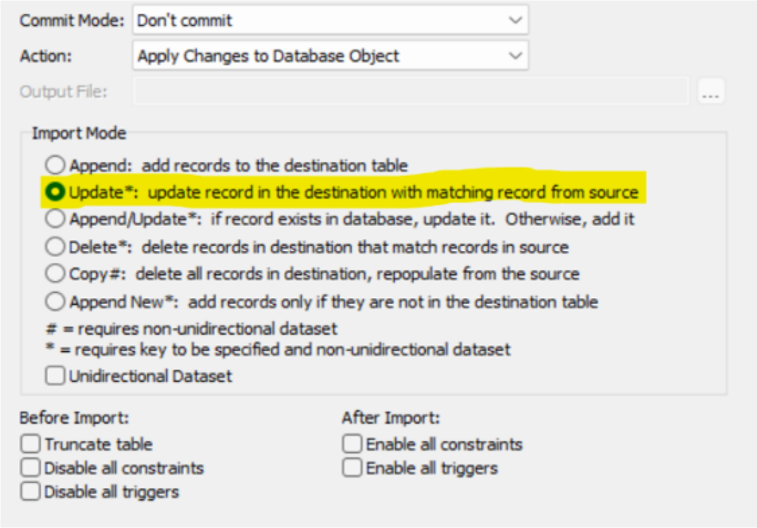
Figure 5: Market_change_pct Preparation

From the database option in the menu bar in Toad we selected 'import' then 'import table data'. At this point, we choose the spreadsheet we want to import and the columns we want and proceed with the rest of the steps. The following two figures show how we specified the columns to match with while importing into Toad.



Destination	Source	🔑
SPF_PKEY		□
FULL_DATE	Field6	☑
COMPANY_KEY	Field3	☑
STOCK_MARKET_KEY		□
INDUSTRY_KEY		□
STOCK_PRICE		□
OPEN		□
HIGH		□
LOW		□
VOLUME		□
COM_CHANGE_PCT		□
MARKET_CHANGE_PCT	Field12	□
DAILY_RETURN		□
BETA		□
QUARTER_KEY		□

Figure 6: Matching on company key and date



Commit Mode: Don't commit

Action: Apply Changes to Database Object

Output File:

Import Mode

○ Append: add records to the destination table

● Update*: update record in the destination with matching record from source

○ Append/Update*: if record exists in database, update it. Otherwise, add it

○ Delete*: delete records in destination that match records in source

○ Copy#: delete all records in destination, repopulate from the source

○ Append New*: add records only if they are not in the destination table

= requires non-unidirectional dataset

* = requires key to be specified and non-unidirectional dataset

□ Unidirectional Dataset

Before Import:

□ Truncate table

□ Disable all constraints

□ Disable all triggers

After Import:

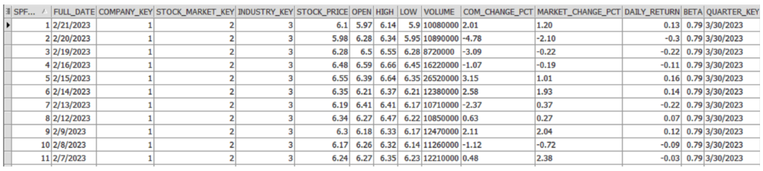
□ Enable all constraints

□ Enable all triggers

Figure 7: Selecting update option not append

5.2.2 Quarter_key Column

This column is the linking step between the two fact tables. It belongs to the daily fact table (stock_prices_fact). We wrote a script to extract the quarter part from the full_date column and insert them into the quarter_key column. This way the quarter_key column contains the end date of every quarter in the dataset, this is when the data of the quarter is released. This is a sample of the final look of the stock_prices_fact table in Toad.

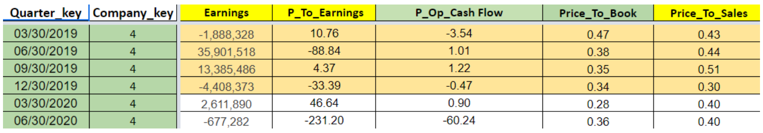


SPF...	FULL_DATE	COMPANY_KEY	STOCK_MARKET_KEY	INDUSTRY_KEY	STOCK_PRICE	OPEN	HIGH	LOW	VOLUME	COM_CHANGE_PCT	MARKET_CHANGE_PCT	DAILY_RETURN	BETA	QUARTER_KEY
1	2/21/2023	1	2	3	6.1	5.97	6.14	5.9	10080000	2.01	1.20	0.13	0.79	3/30/2023
2	2/20/2023	1	2	3	5.98	6.28	6.34	5.95	10890000	-4.78	-2.10	-0.3	0.79	3/30/2023
3	2/19/2023	1	2	3	6.28	6.5	6.55	6.28	8720000	-3.09	-0.22	-0.22	0.79	3/30/2023
4	2/16/2023	1	2	3	6.48	6.59	6.66	6.45	16220000	-1.07	-0.19	-0.11	0.79	3/30/2023
5	2/15/2023	1	2	3	6.55	6.39	6.64	6.35	26520000	3.15	1.01	0.16	0.79	3/30/2023
6	2/14/2023	1	2	3	6.35	6.21	6.37	6.21	12380000	2.58	1.93	0.14	0.79	3/30/2023
7	2/13/2023	1	2	3	6.19	6.41	6.41	6.17	10710000	-2.37	0.37	-0.22	0.79	3/30/2023
8	2/12/2023	1	2	3	6.34	6.27	6.47	6.22	10850000	0.63	0.27	0.07	0.79	3/30/2023
9	2/9/2023	1	2	3	6.3	6.18	6.33	6.17	12470000	2.11	2.04	0.12	0.79	3/30/2023
10	2/8/2023	1	2	3	6.17	6.26	6.32	6.14	11260000	-1.12	-0.72	-0.09	0.79	3/30/2023
11	2/7/2023	1	2	3	6.24	6.27	6.35	6.23	12210000	0.48	2.38	-0.03	0.79	3/30/2023

Figure 8: Stock_prices_fact Table

5.3 Quarterly Data

The aggregated fact table has all the referential keys (company_key, stock_market_key, industry_key) and the quarter dimension key passed from the stock_prices_fact table as shown in the previous step. We wrote a script that inserts all the referential keys and calculates the average stock price and the average volume on a quarterly basis for each company. For this, we used analytic SQL functions to partition by the company and the quarter key and get the averages we need. The script is here in the appendix.
The rest of the quarterly data are collected, as stated in the data collection section, from investing pro as discrete values or from the companies' websites themselves. We gathered their values and arrange them in a spreadsheet. Then imported the spreadsheet into Toad by matching the company key and the quarter key the same way we imported the market_change_pct column previously.



Quarter_key	Company_key	Earnings	P_To_Earnings	P_Op_Cash Flow	Price_To_Book	Price_To_Sales
03/30/2019	4	-1,888,328	10.76	-3.54	0.47	0.43
06/30/2019	4	35,901,518	-88.84	1.01	0.38	0.44
09/30/2019	4	13,385,486	4.37	1.22	0.35	0.51
12/30/2019	4	-4,408,373	-33.39	-0.47	0.34	0.30
03/30/2020	4	2,611,890	46.64	0.90	0.28	0.40
06/30/2020	4	-677,282	-231.20	-60.24	0.36	0.40

Figure 9: Sample of the Data Arranged in the Spreadsheet

6 Data Transformation

The data did not need many transformations. Only 3 columns were transformed.

The volume column in the stock_prices_fact table initially had data with the thousand abbreviation 'K' and the million abbreviation 'M'. That would prevent any calculations on the column. To transform this column, a backup column is created and populated with the volume column values. We then removed the 'K' or 'M' characters from the volume column. Now we needed to multiply the volume column by 1 thousand or 1 million. The backup column is used here such that the values with 'K' are multiplied by 1 thousand and the values with 'M' are multiplied by 1 million. Now the backup column is dropped and we end up with the correct values in the volume column. Refer to the first block of code in the data transformation section in the appendix.

The other two columns are Com_Change_pct and Market_Change_pct. They had the percentage sign '%' in every value. These two columns are used in the Beta measure in the next section. That is why it is needed to have them in numeric format. Using the replace function, we removed the percentage sing '%'. Refer to the second block of code in the data transformation section in the appendix.

7 Measures

In this section, we provide explanations for all measures used in the two fact tables.

7.1 Stock_prices_fact Table

Measure	Description
Stock_price	The daily close price for the stock
Open	The daily open price for the stock
High	Maximum price of a stock through the whole day
Low	Minimum price of a stock through the whole day
Volume	The number of shares exchanged throughout the whole day.
Com_change_pct	The percentage change of the company's stock price over the day.
Market_change_pct	The percentage change of the market index we picked stock price over the day.
Beta	A measure used to rank stocks according to how much they deviate from the market.

7.2 Sp_aggregate_fact Table

Avg_sp	The average stock price over the quarter
Avg_volume	The average volume over the quarter
Earnings	The net income during a quarter
Revenue	The total amount of income generated by sales throughout a quarter
Net_profit_margin	A measure of how much net income or profit is generated as a percentage of revenue
P_to_E	The ratio of price per share to earnings per share. It measures whether the company is undervalued or overvalued
P_op_cashflow	measures the value of a stock's price relative to its operating cash flow per share
Price_to_book	The ratio of the market value of a company's shares (share price) over its book value of equity
Price__to_sales	the price the investors put on each 1 Dollar of the revenue of the company
Market_cap	The total market value of a company's outstanding shares of stock
Outstanding_shares	The total number of shares issued and actively held by stockholders
Share_turnover	The ratio of the trading volume of shares to the number of shares outstanding
Current_assets	Cash and other assets that are expected to be converted to cash within a year
Current_liabilities	The amounts due to be paid to creditors within twelve months
Inventory	a current asset account found on the balance sheet, consisting of all raw materials, work-in-progress, and finished goods that a company has
Cash_equivalents	the line item on the balance sheet that reports the value of a company's assets that are cash or can be converted into cash immediately
Current_ratio	The ratio between current assets and current liabilities
Quick_ratio	The ratio between current assets minus inventory, and current liabilities
Cash_ratio	The ratio between cash equivalents and current liabilities

Note: All queries used to calculate some of the measures are in the Measures Calculation section in the appendix.

8 Conclusion

Given the analysis done in our dashboard, it is promising that the model is reliable and can be used to take an investment decision.

9 Appendix

This appendix includes all the SQL, PL/SQL code used in this project.

9.1 Tables Creation

9.1.1 Dimension Tables

```
CREATE TABLE Date_dim
 (
    Date_key Date CONSTRAINT Date_pk_cons PRIMARY KEY,
    Year Varchar2(4),
    Quarter Varchar2(4),
    month_name varchar(25),
    Month Varchar2(10),
    week_of_year varchar(4),
    week_of_month varchar(4),
    Day Varchar(4),
    day_name varchar(20)
 );
 --------------------------------------------
CREATE TABLE Stock_Market_Dim
 (
    SMarket_key NUMBER(4) CONSTRAINT SM_pk_cons PRIMARY KEY,
    Name Varchar2(150),
    Country Varchar2(100),
    picked_index varchar2(150)
 );
-------------------------------------------------
CREATE TABLE Industry_Dim
 (
    Industry_key NUMBER(4) CONSTRAINT Ind_pk_cons PRIMARY KEY,
    Name Varchar2(100),
    Sector Varchar2(100)
 );

---------------------------------
CREATE TABLE Company_Dim
 (
    Company_key NUMBER(4) CONSTRAINT Comp_pk_cons PRIMARY KEY,
    Name Varchar2(100),
    Stock_Symbol Varchar2(10),
    Headquarters_Location Varchar2(250),
    Type Varchar2(10),
    Equity_Type Varchar2(20),
    Web_Page Varchar2(400)
 );
|----------------------------------
CREATE TABLE Quarter_dim
 (
    Quarter_key date CONSTRAINT Quarter_pk_cons PRIMARY KEY,
    Quarter Varchar2(4),
    Year Varchar2(4)
 );
```

9.1.2 Fact Tables

```
CREATE TABLE Stock_Prices_Fact (
Spf_pkey NUMBER(20) CONSTRAINT SPF_pk_cons PRIMARY KEY,
Full_date DATE,
Company_key NUMBER(4),
Stock_Market_key NUMBER(4),
Industry_key NUMBER(4),
Stock_Price NUMBER(8,2) ,
Open NUMBER(8,2),
High NUMBER(8,2),
Low NUMBER(8,2),
Volume VARCHAR2(15) ,
COM_Change_Pct VARCHAR2(15),
Market_Change_Pct VARCHAR2(15) ,
Daily_Return NUMBER(8,2),
Beta number(8,2)
);
-----------------------------------------
ALTER TABLE Stock_Prices_Fact
ADD(
CONSTRAINT Date_fk_cons FOREIGN KEY(Full_date) REFERENCES
Date_Dim(Date_key),
CONSTRAINT Comp_fk_cons FOREIGN KEY(Company_key) REFERENCES
Company_Dim(Company_key),
CONSTRAINT SM_fk_cons FOREIGN KEY(Stock_Market_key) REFERENCES
Stock_Market_Dim(SMarket_key),
CONSTRAINT Ind_fk_cons FOREIGN KEY(Industry_key) REFERENCES
```

```
Industry_Dim(Industry_key)
);
--------------------------------------------
/*Adding Quarter_key column to stock_prices_fact
(This column will be used to create the aggregate fact table)*/
alter table stock_prices_fact
add quarter_key date;
-----------------------------------------------
Create table sp_aggregated_fact
    (Agg_Pkey number(10) CONSTRAINT Agg_pk_cons PRIMARY KEY,
    Quarter_key date,
    Company_key NUMBER(4),
    Stock_Market_key NUMBER(4),
    Industry_key NUMBER(4),
    Avg_SP Number(10,2),
    Avg_Volume Number(10,2),
    Earnings NUMBER(20,2),
    Revenue NUMBER(20,2),
    Net_Profit_Margin NUMBER(20,2),
    P_To_E NUMBER(20,2),
    P_Op_CashFlow NUMBER(20,2),
    Price_To_Book NUMBER(20,2),
    Price_To_Sales NUMBER(20,2),
    Market_Cap NUMBER(20,2),
    Outstanding_Shares NUMBER(20),
    Share_Turnover NUMBER(20,4),
    Current_assets NUMBER(20,2),
    Current_liabilities NUMBER(20,2),
    Inventory NUMBER(20,2),
    Cash_equivalents NUMBER(20,2),
    Current_Ratio NUMBER(20,4),
    Quick_Ratio NUMBER(20,4),
    Cash_Ratio NUMBER(20,4)
);
------------------------------------------
ALTER TABLE SP_Aggregated_Fact
ADD(
    CONSTRAINT Quarter_fk_cons FOREIGN KEY(Quarter_key) REFERENCES
    Quarter_dim(Quarter_key),
    CONSTRAINT Comp_fk1_cons FOREIGN KEY(Company_key) REFERENCES
    Company_Dim(Company_key),
    CONSTRAINT SM_fk1_cons FOREIGN KEY(Stock_Market_key) REFERENCES
    Stock_Market_Dim(SMarket_key),
    CONSTRAINT Ind_fk1_cons FOREIGN KEY(Industry_key) REFERENCES
    Industry_Dim(Industry_key)
);
```

9.2 Data Insertion

9.2.1 Dimension Tables Insertions

Inserting Dates into Date_dim

```
    Declare
i date := to_date( '01/01/2019', 'MM/DD/YYYY');
max_date date := to_date('12/31/2023', 'MM/DD/YYYY');

Begin
 while i <= max_date
 loop

 insert into date_dim (DATE_KEY, YEAR, QUARTER, MONTH_NAME, MONTH,
 WEEK_OF_YEAR, WEEK_OF_MONTH, DAY, DAY_NAME)
 VALUES (i, to_char(i, 'YYYY'), 'Q'||to_char(i, 'Q'), to_char(i, 'Month'),
 to_char(i, 'MM'), to_char(i, 'WW'), to_char(i, 'W'),
 to_char(i, 'DD'),  to_char(i, 'Day'));
 i := i + interval '1' day;

 end loop;

 End;
 --------------------------------------
```

Inserting Quarters and Year into Quarter_dim

```
    Declare
i date := to_date( '03/30/2019', 'MM/DD/YYYY');
max_date date := to_date('12/30/2023', 'MM/DD/YYYY');

begin
 while i <= max_date
 loop

 insert into Quarter_dim (Quarter_key, Quarter,Year)
 VALUES (i,  'Q'||to_char(i, 'Q'),to_char(i, 'YYYY'));
 i := i + interval '3' Month ;
 end loop;

 end;
----------------------------------------------
```

Populating the stock market dimension

```
 insert into stock_market_dim (NAME, COUNTRY, PICKED_INDEX)
 values ('NASDAQ', 'United States', 'NASDAQ Composite');
 ----------------------------
 insert into stock_market_dim (NAME, COUNTRY, PICKED_INDEX)
 values ('EGX', 'Egypt', 'EGX100');
 ----------------------------
  insert into stock_market_dim (NAME, COUNTRY, PICKED_INDEX)
 values  ('Frankfurt Stock Exchange' , 'Germany', 'DAX30');
 ----------------------------
   insert into stock_market_dim (NAME, COUNTRY, PICKED_INDEX)
 values  ('JPX', 'Japan', 'Nikkei 225');
--------------------------------------------
```

Populating the industry dimension

```
insert into industry_dim (NAME, SECTOR)
 values ('Real Estate Operations', 'Real Estate');
 ---------------------------------------
 insert into industry_dim ( NAME, SECTOR)
 values ('Passenger Transportation Services', 'Industrials');
 -----------------------------
  insert into industry_dim ( NAME, SECTOR)
 values ('Professional and Commercial Services',  'Industrials');
 --------------------------
 insert into industry_dim (NAME, SECTOR)
 values ( 'Telecommunications Services', 'Technology');
---------------------------------------
```

Populating The Company dimension

```
Insert into Company_dim
    (NAME, STOCK_SYMBOL, HEADQUARTERS_LOCATION, TYPE, EQUITY_TYPE, WEB_PAGE)
Values
    ('Fawry', 'FWRY', 'Egypt', 'Equity', 'ORD', 'fawry.com');
 ------------------------- -------------------------- ---------------------------
Insert into Company_dim
    (NAME, STOCK_SYMBOL, HEADQUARTERS_LOCATION, TYPE, EQUITY_TYPE, WEB_PAGE)
Values
    'Swvl', 'SWVL', 'United Arab Emirates', 'Equity', 'ORD', 'swvl.com');
 -------------------------- -------------------------- --------------------------
Insert into Company_dim
    (NAME, STOCK_SYMBOL, HEADQUARTERS_LOCATION, TYPE, EQUITY_TYPE, WEB_PAGE)
Values
    (’El Obour Real Estate’, ’OBRI’, ’Egypt’, ’Equity’, ’ORD’, ’www.elobouregy.com’);
  -------------------------- -------------------------- --------------------------
Insert into Company_dim
     (NAME, STOCK_SYMBOL, HEADQUARTERS_LOCATION, TYPE, EQUITY_TYPE, WEB_PAGE)
Values
    (’Amer Group’, ’AMER’,  ’Egypt’, ’Equity’, ’ORD’, ’amer-group.com’);
 -------------------------- -------------------------- --------------------------
Insert into Company_dim
     (NAME, STOCK_SYMBOL, HEADQUARTERS_LOCATION, TYPE, EQUITY_TYPE, WEB_PAGE)
Values
    (’Arab Real Estate’, ’RREI’, ’Egypt’, ’Equity’, ’ORD’,
    ’arabianrealestate-aleco.com’);

-------------------------- -------------------------- --------------------------


Insert into Company_dim
    (NAME, STOCK_SYMBOL, HEADQUARTERS_LOCATION, TYPE, EQUITY_TYPE, WEB_PAGE)
Values
    ( ’Arab Developers Holding’, ’ARAB’, ’Egypt’, ’Equity’, ’ORD’,
    ’ad-holding.com’);

--------------------------------------------------------------------------------
```

9.2.2 Fact Tables Insertions

Extracting quarters from dates and inserting them into the quarter_key column in Stock_prices_fact

```
declare
min_key number(20);
max_key number(20);

begin
    select min(spf_pkey) into min_key from stock_prices_fact;
    select max(spf_pkey) into max_key from stock_prices_fact;
    while min_key <= max_key
    loop
        update stock_prices_fact
        set quarter_key = ( select case when month in (to_char(’01’), to_char(’02’),
    to_char(’03’))
    then to_date(’03/30’||extract(YEAR from full_date), ’MM/DD/YYYY’)
```

```
      when month in (to_char('04'), to_char('05'),
         to_char('06')) then
     to_date('06/30/'||extract(YEAR from full_date), 'MM/DD/YYYY')
     when month in (to_char('07'), to_char('08'),
     to_char('09'))
     then to_date('09/30/'||extract(YEAR from full_date), 'MM/DD/YYYY')
     when month in (to_char('10'), to_char('11'),
     to_char('12'))
     then to_date('12/30/'||extract(YEAR from full_date), 'MM/DD/YYYY')
      end as test
    from (
        select full_date, to_char(full_date,'Mm') as month
        from stock_prices_fact where spf_pkey = min_key))
        where  spf_pkey = min_key ;

        min_key:= min_key +1;
    end loop;
end;
-------------------------------------------------
```

Inserting data into sp_aggregated_fact

```
    Insert into sp_aggregated_fact (quarter_key, company_key,
    stock_market_key, industry_key, Avg_SP, Avg_Volume)
select *  from (
   select quarter_key, company_key, stock_market_key,
   industry_key,
     max(round(avg_stock_price,2)) as Avg_SP_Q,
      max(round(avg_volume))  as Avg_Volume_Q from (
        select quarter_key, company_key, stock_market_key,
        industry_key,
            avg(stock_price) over(partition by company_key,
            quarter_key) as avg_stock_price,
            avg(volume) over(partition by company_key,
            quarter_key) as avg_volume
            from stock_prices_fact)
            group by quarter_key, company_key,
            stock_market_key, industry_key
            order by company_key, quarter_key);
```

9.3 Data Transformation

Transforming the volume column

```
--adding a backup column for the volume so I can distinguish
between values that are in million and thousand and removing
(M and K) in the original column

Alter table stock_prices_fact
add
    Volume_bckup varchar2 (20) ;

Update stock_prices_fact
set Volume_bckup = volume ;
--Removing M and K from the column
        Update stock_prices_fact
            Set volume = replace (volume, ’M’,’’);

        Update stock_prices_fact
            Set volume = replace (volume, ’K’,’’);

---Multiplying by 1000 and 1000000
         Update stock_prices_fact
            Set volume = volume*1000
         Where Volume_bckup like ’%K’

         Update stock_prices_fact
            Set volume = volume*1000000
         Where Volume_bckup like ’%M’


-Deleting the backup volume colum
Alter table stock_prices_fact
    Drop column Volume_bckup
-----------------------------------------
```

Transforming the Market_Change_pct and Com_Change_pct columns

```
    ---Removing ’%’ from both columns [ com_change_pct , market_change_pct]

Update stock_prices_fact
    Set com_change_pct = replace (com_change_pct, ’%’,’’);

Update stock_prices_fact
    Set market_change_pct = replace (market_change_pct, ’%’,’’);
-----------------------------------------------------
```

9.4 Measures Calculations

Calculating the beta measure in the stock_prices_fact

```
    declare
min_com number(4);
max_com number(4);
begin
    select min(company_key) into min_com from stock_prices_fact;
    select max(company_key) into max_com from stock_prices_fact;
    while min_com <= max_com
    loop
    update stock_prices_fact
    set beta =  (select distinct cov/var from (
                select company_key, covar_pop(com_change_pct, market_change_pct)
                over(partition by company_key) as cov,
                variance(market_change_pct) over(partition by company_key) as var
                from stock_prices_fact
                where company_key = min_com))
     where company_key = min_com;

     min_com := min_com+1;
     end loop;
end;
```

Calculating the share turnover measure

```
update sp_aggregated_fact
set share_turnover = avg_volume/outstanding_shares;
```

Calculating the current ratio measure

```
update sp_aggregated_fact
set current_ratio = current_assets/current_liabilities;
```

Calculating the quick ratio measure

```
update sp_aggregated_fact
set quick_ratio = (current_assets - inventory)/current_liabilities;
```

Calculating the cash ratio measure

```
update sp_aggregated_fact
set cash_ratio = cash_equivalents/current_liabilities;
```