**Technical Report for Stock Analysis Using News Articles and PESTLE Data**

****

**By**

**Neeraj Chavan**

**10521393**

**CONTENTS**

**1. Introduction.**

**1.1. Reasons for selecting the subject area.**

**1.2. Vision and Goals.**

**1.3. Key StakeHolders.**

**1.4. Business requirements.**

**2. DATASET AND SOURCES**

**2. SCHEMA.**

**3. ETL.**

**4. VISUALIZATIONS AND REPORTS.**

**4.1. Visualizations.**

**4.2. Reports.**

**5. Include XML and Schema.**

**6. Graph Databases.**

**6.1. Comparison to relational databases.**

**7. Conclusions.**

**8. Bibliography.**

**Appendix A – VISUALIZATIONS Code.**

**Appendix B – Neo4J code.**

# 

# 1. Introduction

Earlier studies on stock market prediction are based on the historical stock prices. Later studies have debunked the approach of predicting stock market movements using historical prices. Stock market prices are largely fluctuating. The efficient market hypothesis (EMH) states that financial market movements depend on news, current events and product releases and all these factors will have a significant impact on a company’s stock value. Because of the lying unpredictability in news and current events, stock market prices follow a random walk pattern and cannot be predicted with more than 50% accuracy.( Justin Kuepper,2019)

With the advent of media sources like the news paper, the information about the public sector has become abundant. Social media is transforming like a perfect platform to share public emotions about any topic and has a significant impact on overall public opinion. Social media platform like Twitter, has received a lot of attention from researchers in recent times. Twitter is a micro-blogging application that allows users to follow and comment on other user’s thoughts or share their opinions in real time. The social information exploited are very useful for making predictions.

The data used in this project is gathered from various sources such as Yahoo finances, PESTLE News and Irish Times News Archive. The data has been scraped using python code (see Appendix C) to ensure that the data is accurate and not obtained from redundant sources.

## 

## 

## 1.1. Reasons for selecting the subject area

In the finance field, the stock market and its trends are extremely volatile in nature. It attracts researchers to capture the volatility and predicting its next moves. Investors and market analysts study the market behavior and plan their buy or sell strategies accordingly. As the stock market produces a large amount of data every day, it is very difficult for an individual to consider all the current and past information for predicting future trend of a stock.

Mainly there are two methods for Forecasting market trends. One is Technical analysis and the other is Fundamental analysis. Technical analysis considers past price and volume to predict the future trend. Fundamental analysis of a business involves analyzing its financial data to get some insights. The efficacy of both technical and fundamental analysis is disputed by efficient-market hypothesis which states that stock market prices are essentially unpredictable.

## 1.2. Vision and Goals

This research follows the Fundamental analysis technique to discover future trend of a stock by considering news articles about a company as prime information and tries to classify news as good (positive) and bad (negative). If the news sentiment is positive, there are more chances that the stock price will go up and if the news sentiment is negative, then stock price may go down. This research is an attempt to build a model that predicts news polarity which may affect changes in stock trends. In other words, check the impact of news articles on stock prices. We are using supervised machine learning as classification and other text mining techniques to check news polarity.

## 1.3. Key StakeHolders

Key stakeholder of this idea would be the following:

* Investors
* Colleges and Institutions
* Any individual/s with keen interest in the stock market

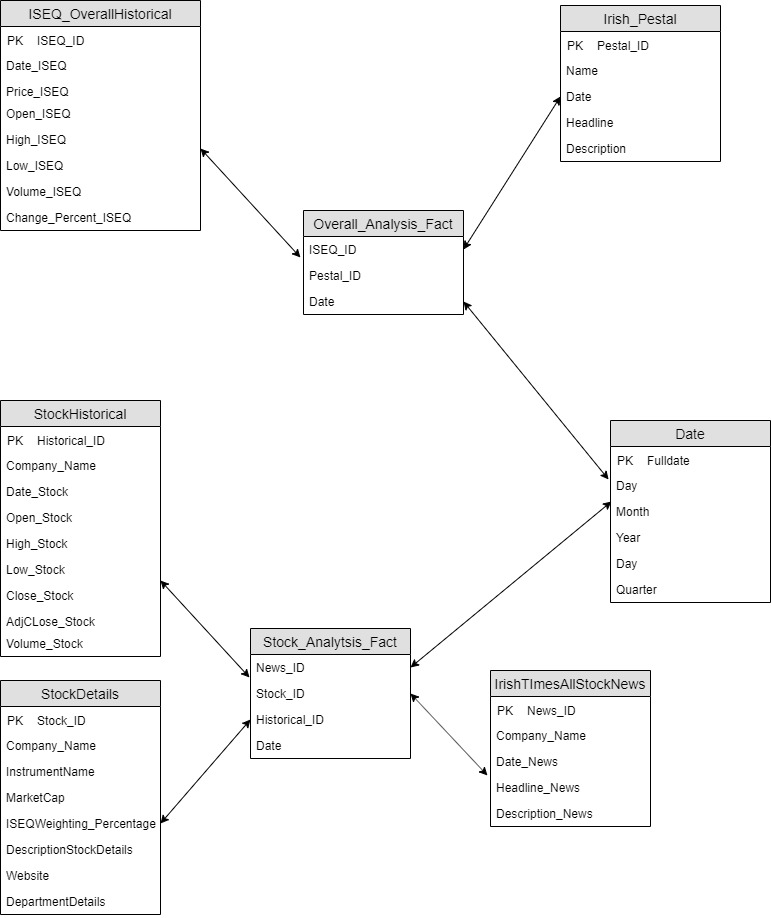
## 1.4. Business requirements

The Business requirements are as follows:

* Understanding the various factors and its business impact.
* Which PESTLE sector/data affects the stock markets the most.
* Average variations in price stock prices.
* Overall market trends over the last 15 years.
* Comparison of competitors stock prices over last 100 days.

# 2. SCHEMA

·



The schema that we have chosen is a star schema to fit the model adequately.A star schema is composed of one or more central fact tables, a set of dimension tables, and the joins that relate the dimension tables to the fact tables. This section describes these components and outlines some of the decisions you need to make before designing a decision-support schema.

The database contains 2 fact tables, Overall\_Analysis\_Fact and Stock\_Analysis\_Fact that connects the various dimension tables by various relations. A fact table contains data columns for the numeric measurements of a business. It also includes a set of columns that form a concatenated, or composite key. Each column of the concatenated key is a foreign key drawn from a dimensional table primary key. Fact tables usually have few columns and many rows, which result in relatively long and narrowly shaped tables.Dimension tables store descriptions of the characteristics of a business. A dimension is usually descriptive information that qualifies a fact. For example, each record in a product dimension represents a specific product. In the star schema shown at the beginning of this chapter, the product, customer, promotion, and time dimensions describe the measurements in the fact table. Dimensions do not change, or change slowly over time.

Our database contains data scraped from various sources which contain data from Irish times news, PESTLE news and data from ISEQ which is the Irish stock market database holding all the data entries of all the Irish stock Market.

Some of the advantages of star schema are as follows:

* Query performance  
  Because a star schema database has a small number of tables and clear join paths, queries run faster than they do against an OLTP system. Small single-table queries, usually of dimension tables, are almost instantaneous. Large join queries that involve multiple tables take only seconds or minutes to run.
* Load performance and administration  
  Structural simplicity also reduces the time required to load large batches of data into a star schema database.
* Easily understood  
  A star schema is easy to understand and navigate, with dimensions joined only through the fact table. These joins are more significant to the end user, because they represent the fundamental relationship between parts of the underlying business. Users can also browse dimension table attributes before constructing a query.

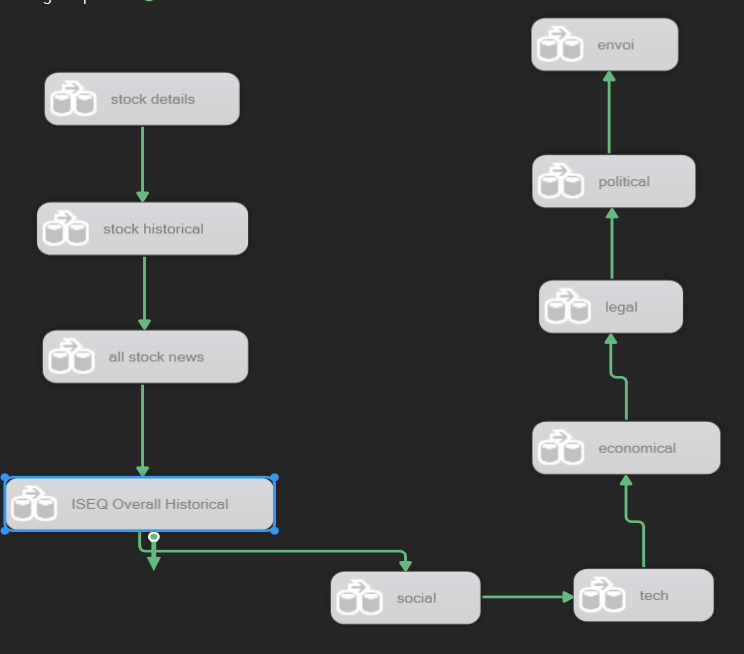
# 3. ETL

ETL (Extract, Transform and Load ) is the process of taking data from one source and transforming it to load it to another place. ETL could be just as simple as mapping fields between two databases or even as complicated as performing mathematical operations on some fields, combining it with other data sources and then loading them into a final location. It’s a process of interfacing data from one department or an outside source and loading it into another database for another department.(Extract, Transform and Load, 2011)

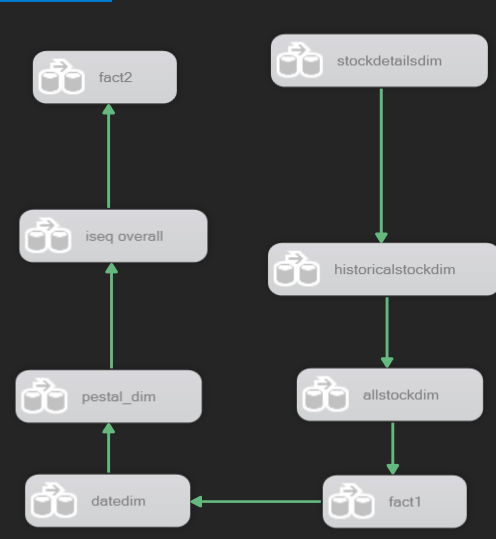
The primary advantage of this being that each department will be able to set their own technical specifications for their projects, then having the ETL team transform the data they need from other departments according to their project requirements.

For this project, initially we created the source database StockPrediction\_DS using SQL Server Management Studio. Then using SSIS (SQL Server Integration Project), we loaded the data into an ADO Destination. The next step was to load the dimension and fact tables using SSIS into the data warehouse (StockPrediction\_DW) for analysis purposes.

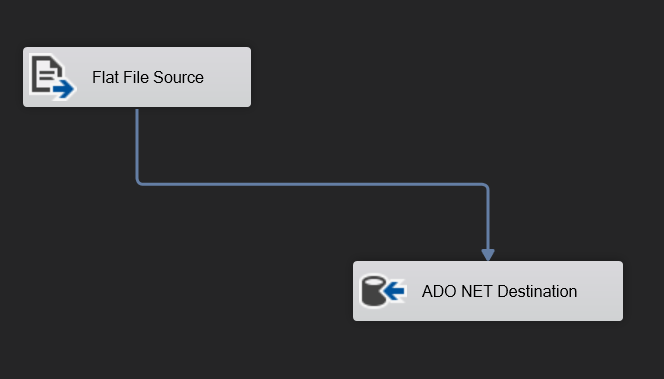
**SSIS Structure for loading data in source Database:**



**SSIS structure for loading data in dimension and fact table:**

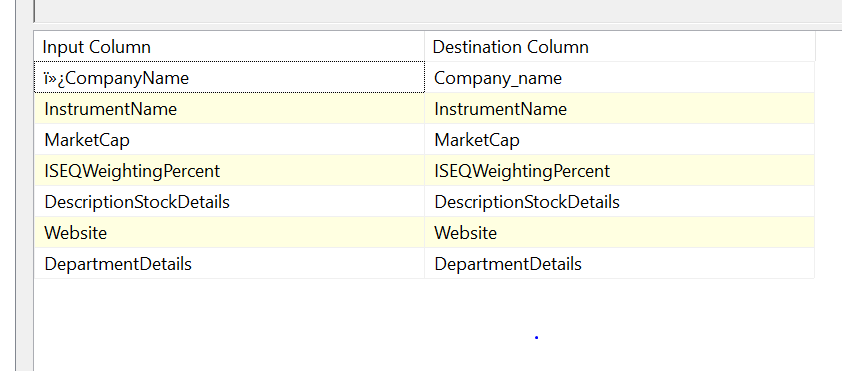


Loading data in single data source table:



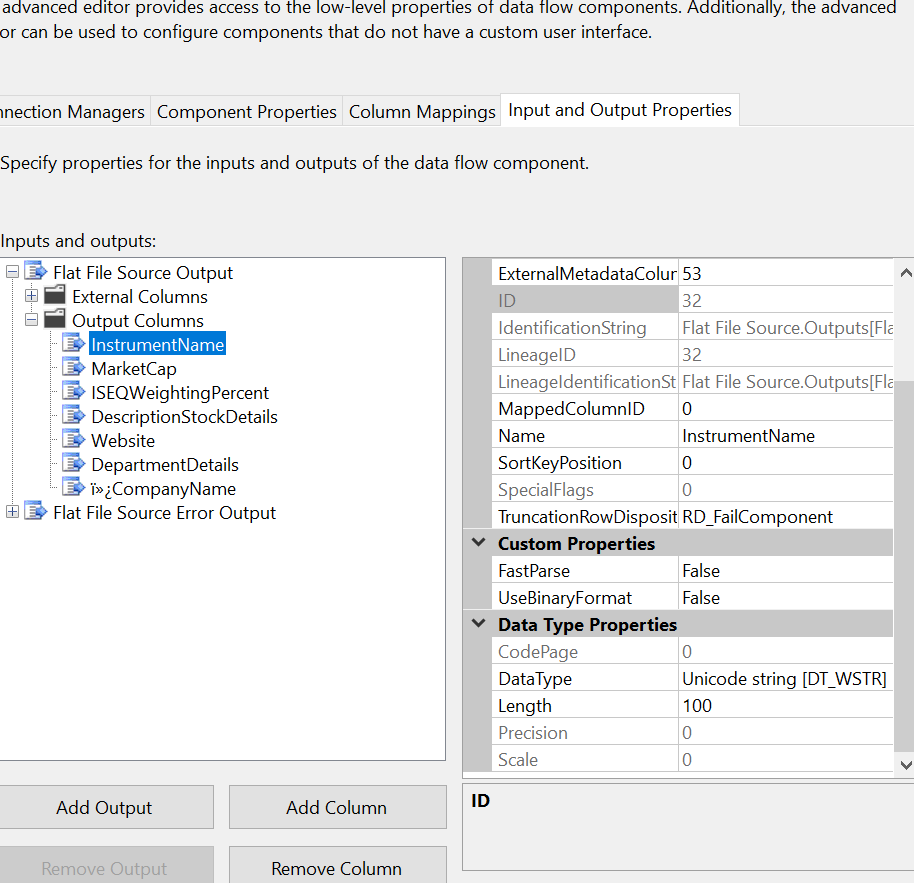
For loading data in our data source tables we used Flat File Source to load our scrapped CSV’s, defined a Flat file connection for the same and passed its output to ADO NET Destination where we created a Database connection with our local server(DESKTOP-JAG58LJ) and selected the Database name and respective table.

**Checking data types for suitable mapping:**



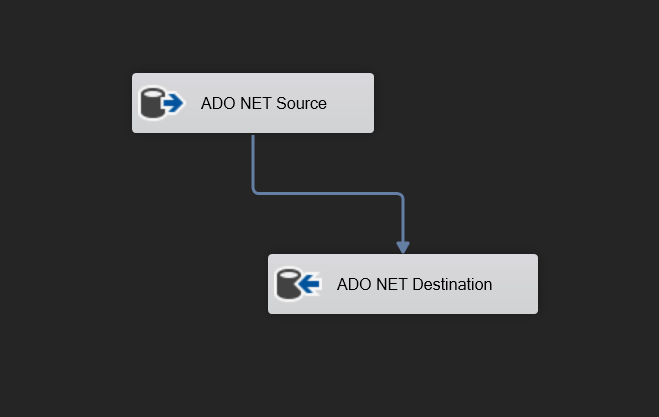
Once our ADO NET Destination is connected to our local database, we go in mapping section where we note down the suitable data types from Destination\_column, which would be used in the next step for data conversion,

**Converting Data Types:**



Once we have noted respective Data Structures for destination columns we right click and select advance filters where we go to Input and Output Properties -> Flat File Source Output -> Output Columns and change the Data types for the output columns with respect to ADO NET Destination destination columns.

**Loading Data in dimension table:**



Once all the data in Data Source database has been loaded, we start loading data in respective Dimensions. To do the same we used ADO NET Source as input where we established a connection with our local database and selected the Database name and respective table where the data needs to be mapped from. Then we connect the output of ADO NET Source to ADO NET Destination as input where we established a connection with our local database and selected the Database name and respective table where the data needs to be mapped from.

Once all the dimensions are loaded we move to create fact table. For the same we first built respective queries In SQL

**Fact table queries:**

SELECT IrishTimesAllStockNews\_dim.date,StockHistorical\_dim.Company\_name,StockHistorical\_dim.Historical\_ID,IrishTimesAllStockNews\_dim.News\_ID,

StockHistorical\_dim.High\_Stock - (StockHistorical\_dim.Low\_Stock) as high\_low

FROM IrishTimesAllStockNews\_dim INNER JOIN

StockHistorical\_dim ON IrishTimesAllStockNews\_dim.date = StockHistorical\_dim.date

SELECT ISEQOverallHistorical\_dim.date,pestal\_dim.pestal\_ID,ISEQOverallHistorical\_dim.ChangePercent\_ISEQ,

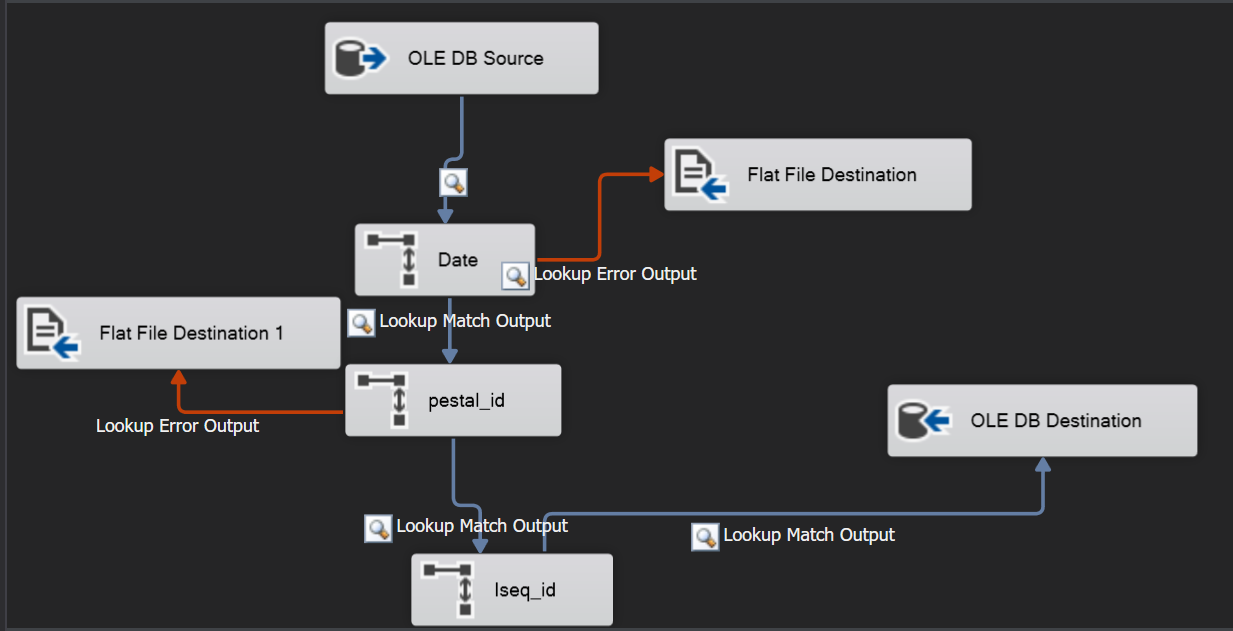
ISEQOverallHistorical\_dim.ISEQ\_ID,ISEQOverallHistorical\_dim.High\_ISEQ - (ISEQOverallHistorical\_dim.Low\_ISEQ) as high\_low,

ISEQOverallHistorical\_dim.Price\_ISEQ - coalesce(lag(ISEQOverallHistorical\_dim.Price\_ISEQ)

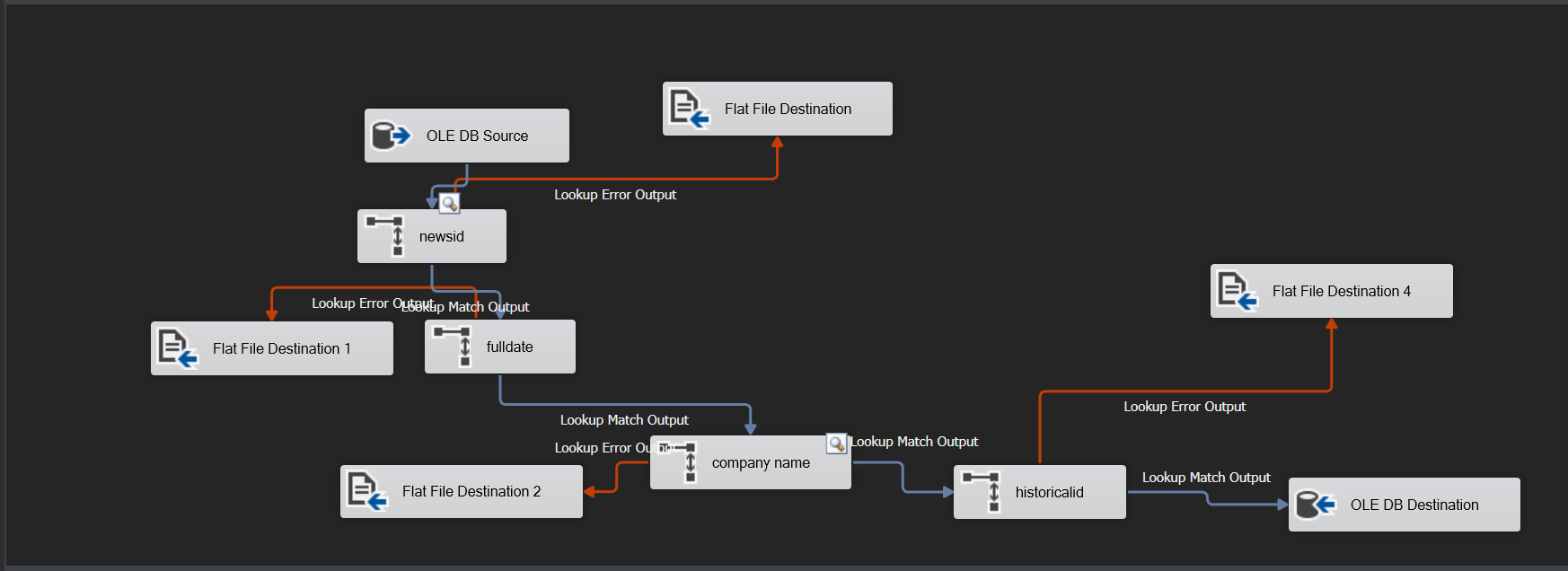
over (order by ISEQOverallHistorical\_dim.ISEQ\_ID), 1) as Price\_diff

From ISEQOverallHistorical\_dim INNER JOIN pestal\_dim ON ISEQOverallHistorical\_dim.date=pestal\_dim.date

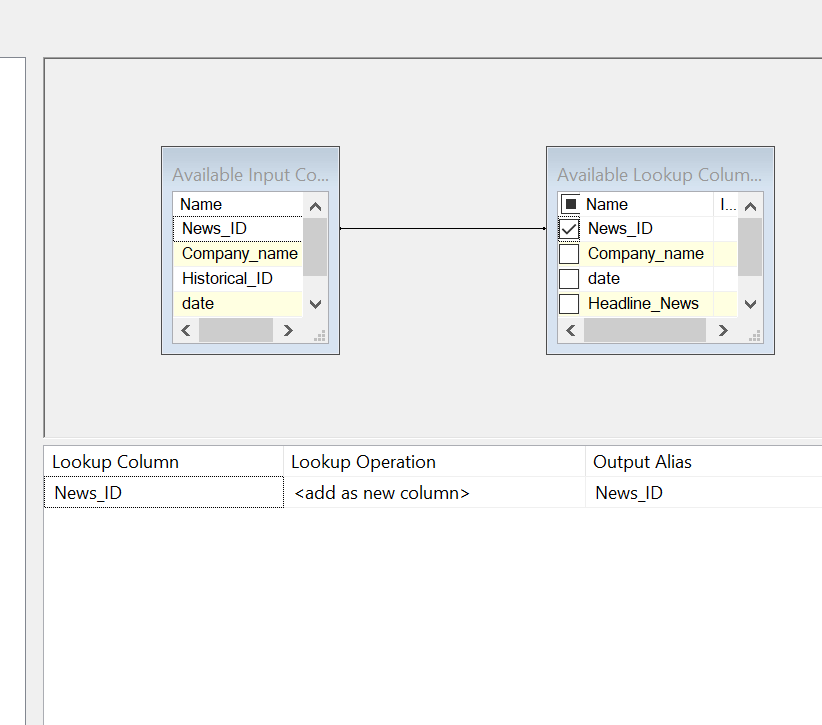
**Overall\_ISEQ\_fact SSIS Structure:**



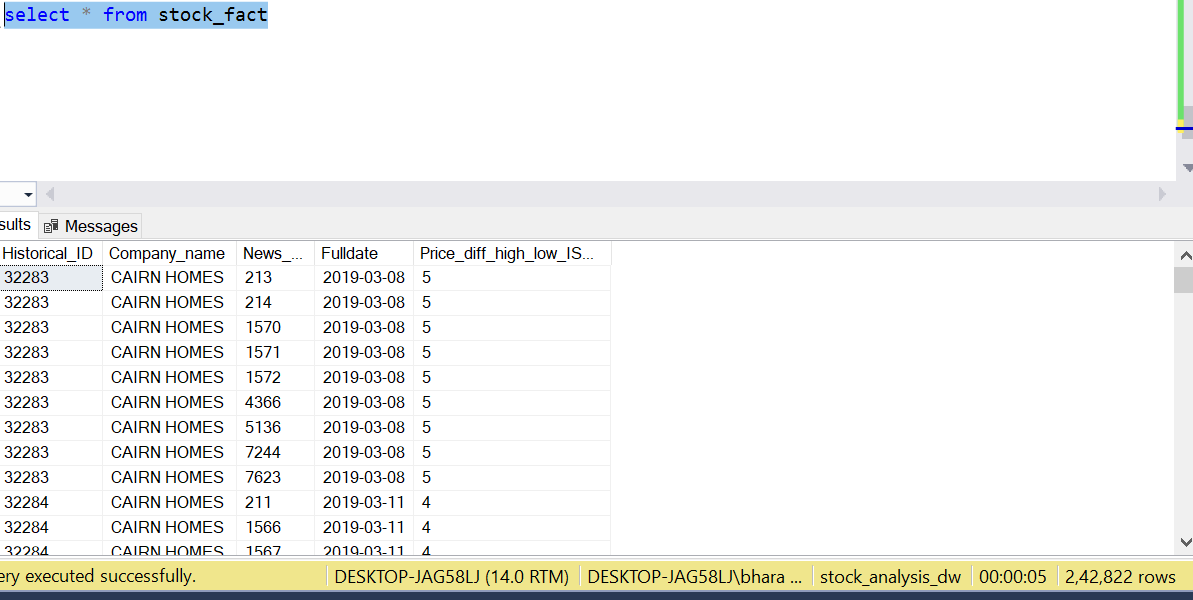
**Stock\_fact SSIS Structure:**



Once queries are created, we select OLE DB Source and establish connection and paste our resp query in Build Query section and connect it to lookups. We use one Lookup to map one column from fact table, so if you have 5 columns in fact table you should have 5 lookups, also we stored unmatched lookups in flat file using Flat File Destination. The output of last lookup goes to OLE DB Destination where we established a connection with our local database and selected the Database name and respective table where the data needs to be mapped.

**Mapping Lookups:**

Once Done, you can select rows from your resp fact tables:



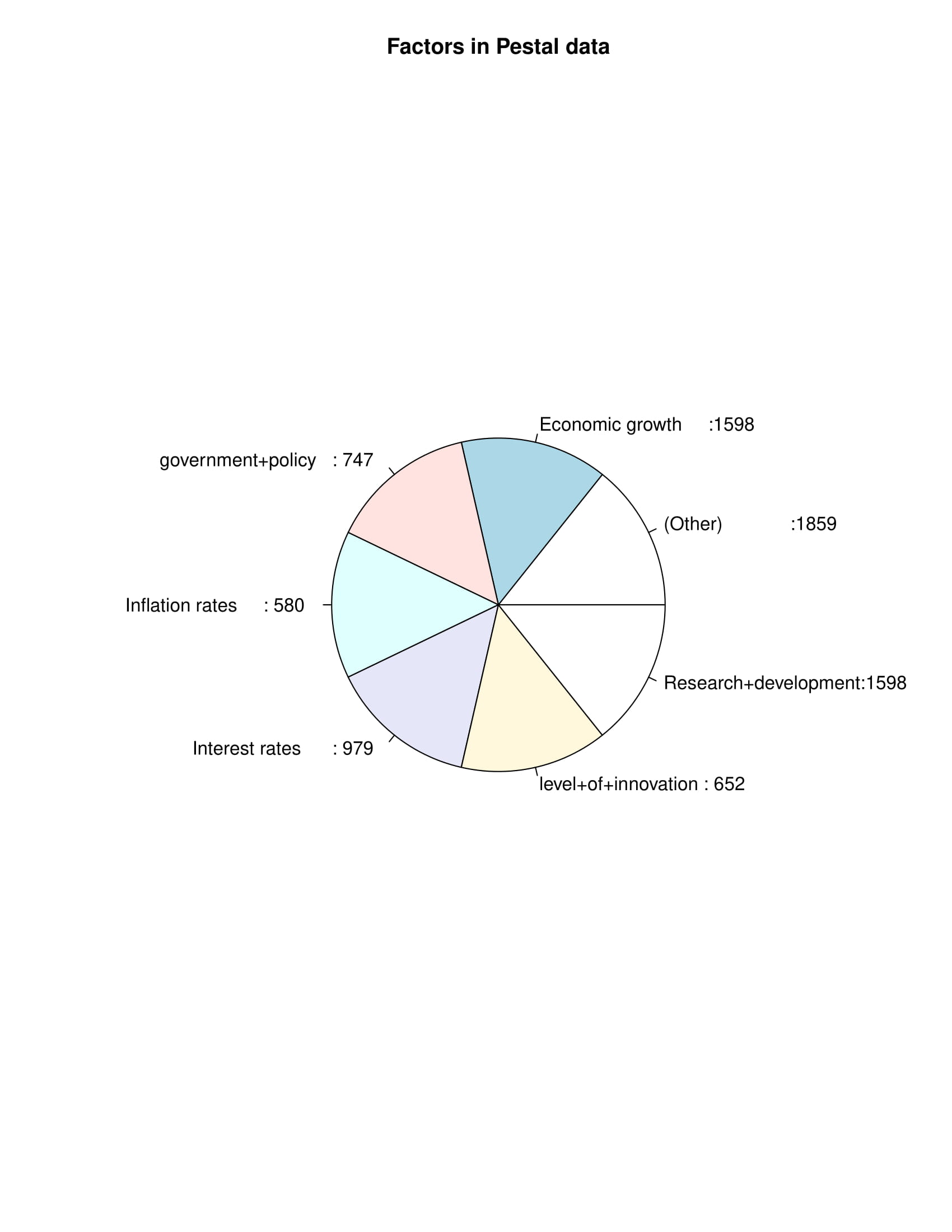
# 

# 4. VISUALIZATIONS AND REPORTS

# 4.1. Visualizations

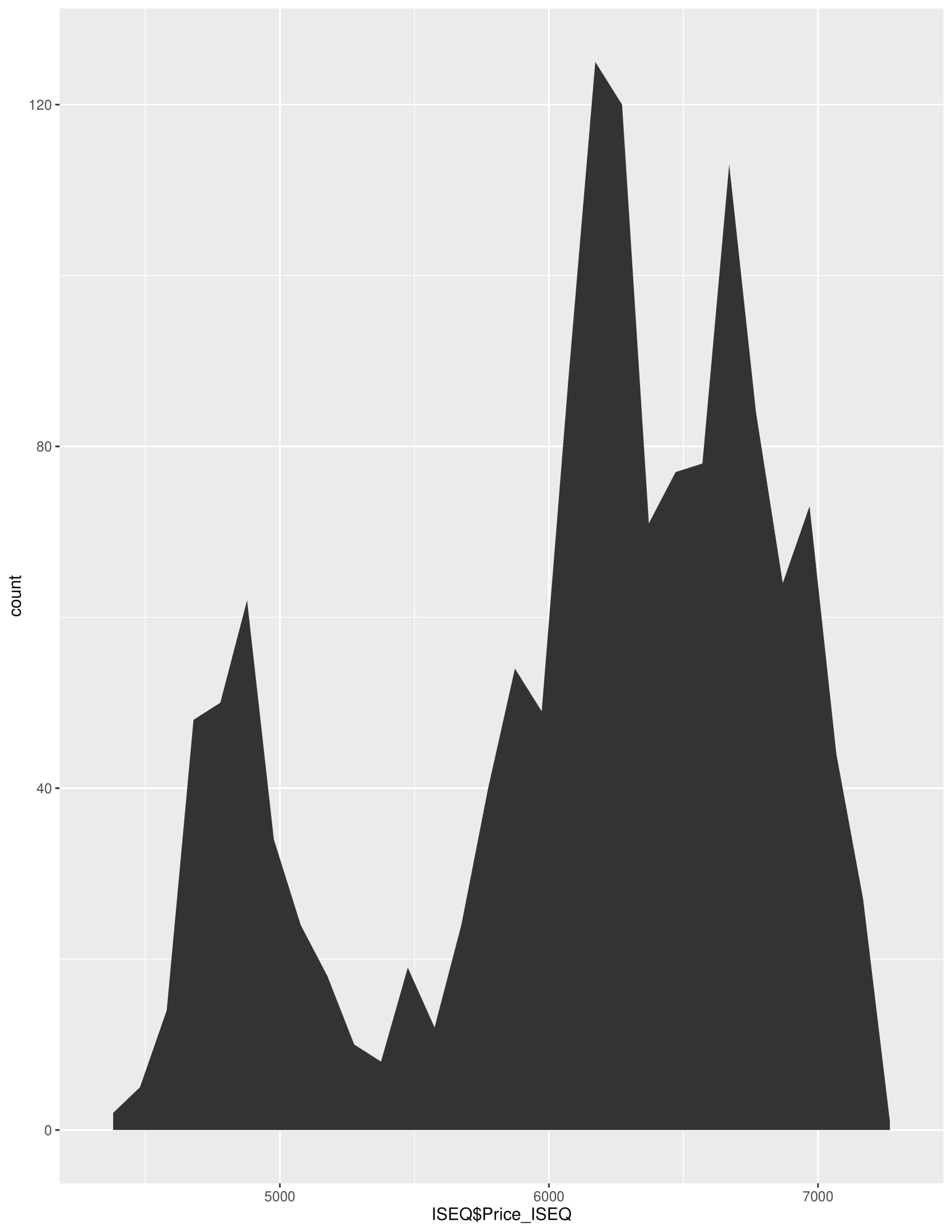
(See Appendix A for code)

**Visualization 1**



Pie Chart visualization displaying the various PESTLE factors that affect the overall Irish Stock Market.

**Visualization 2**



Visualization describing the overall performance of Irish Stock Market - Weights as a single attribute.

**Visualization 3**

## 

Visualization describing the top performers amongst since past 100 days

**Visualization 4**

****

Visualization describing the WordCloud of keywords affecting the Irish Stock Market

**Visualization 5**

## 

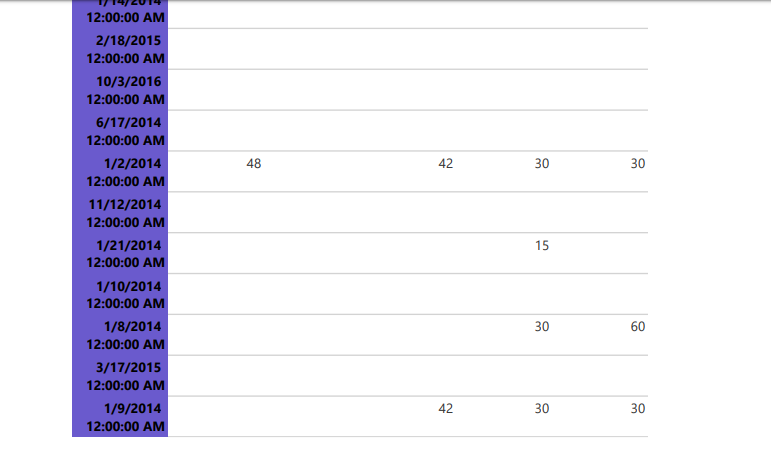
Price fluctuation of ISEQ over the past 15 years

## 4.2. Reports

**Report 1**

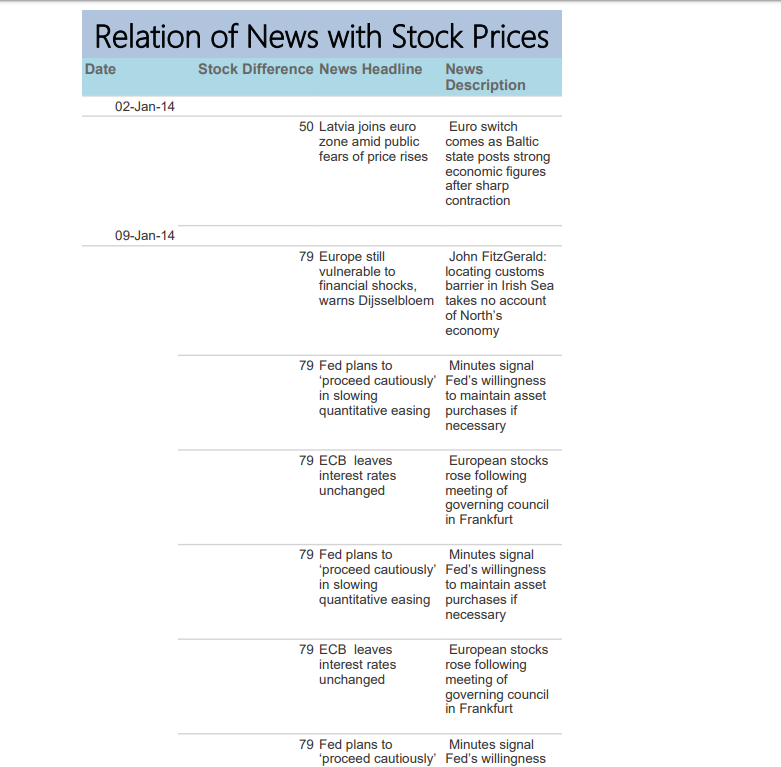


Shown above is a matrix report created using SSRS tool from Microsoft Visual Studio 2019. The factors involved are the Dates of the Articles published, PESTLE sectors involved and the data is the total count of articles on that particular date.



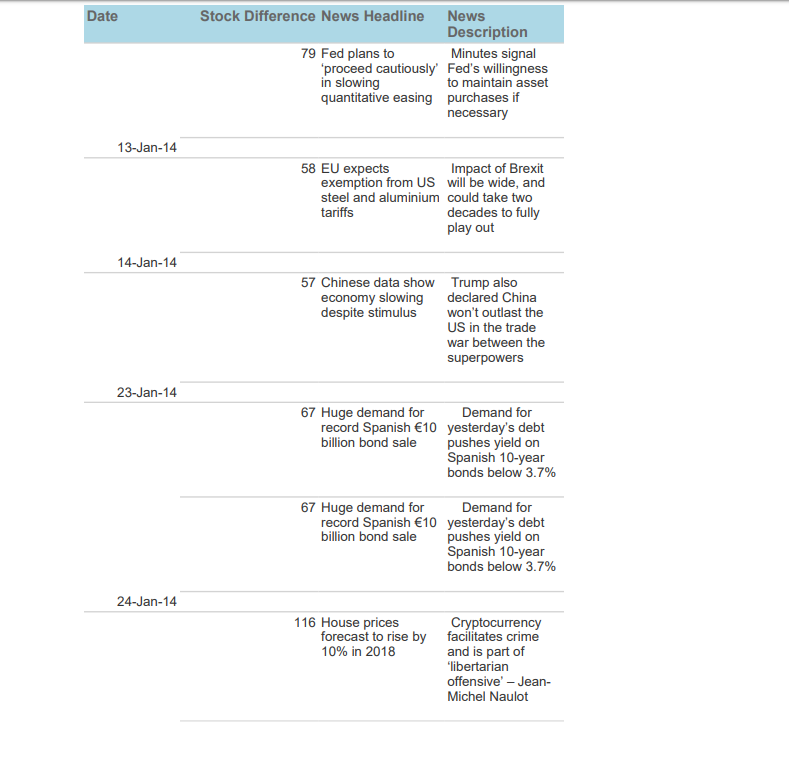
This answer the business requirement “Which factors affect the stock market?”

**Report 2**



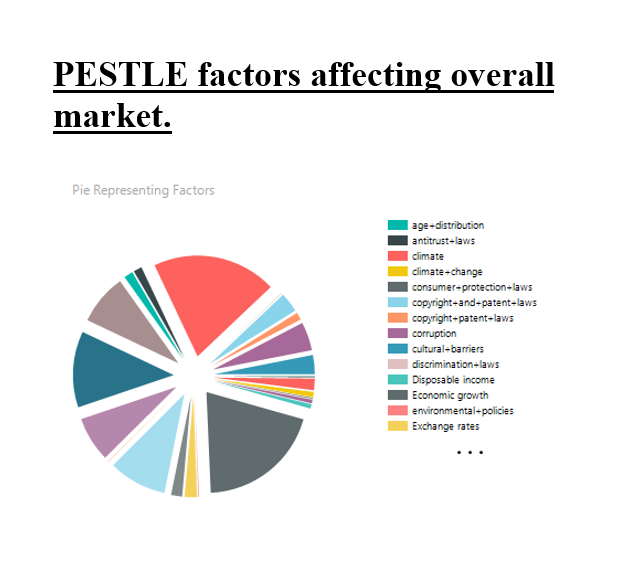
The above given report is another type of report called a Tabular report. This report helps in understanding the data in a tabular format. The factors involved in the above given report are:

1. Date
2. Difference in the stock value (High stock minus low stock)
3. News Headlines for that particular day.
4. News Description for the same day.



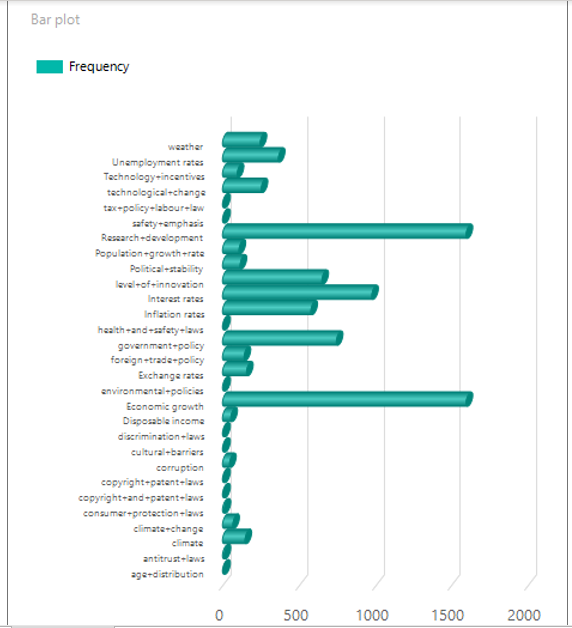
This report shows the difference in the stock value and the corresponding news articles related to it. This is an attempt to understand the average variation in stock prices and the role of news articles in it.

**Report 3**



Since Visualisation of data is easier for the stakeholders to understand it better, the above report is a graphical report which helps us understand which factors influence the stock market.

**Report 4**



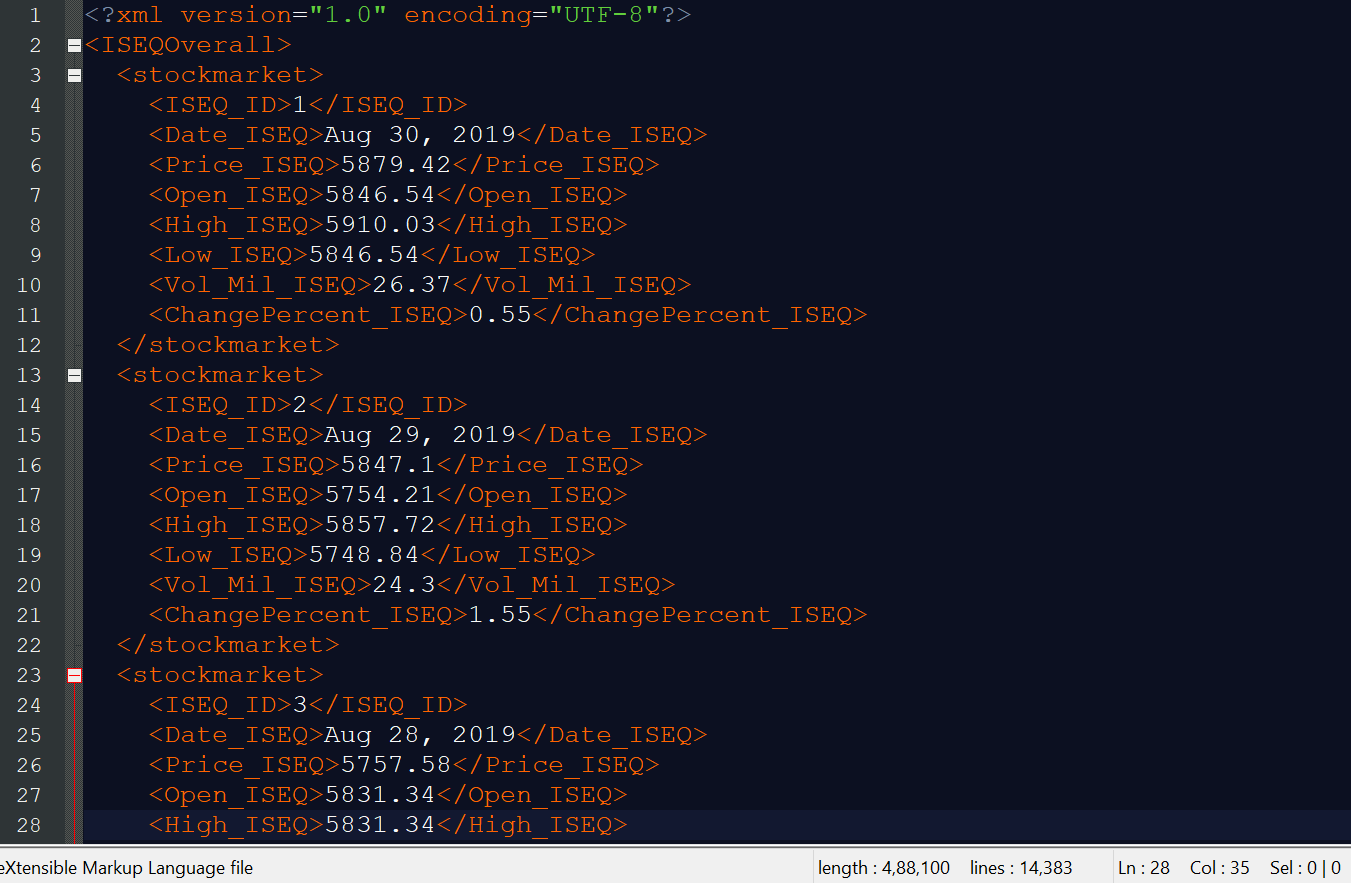
The above report is another graphical representation of the count of which articles affect the Irish stock market most.

# 5. XML and Schema

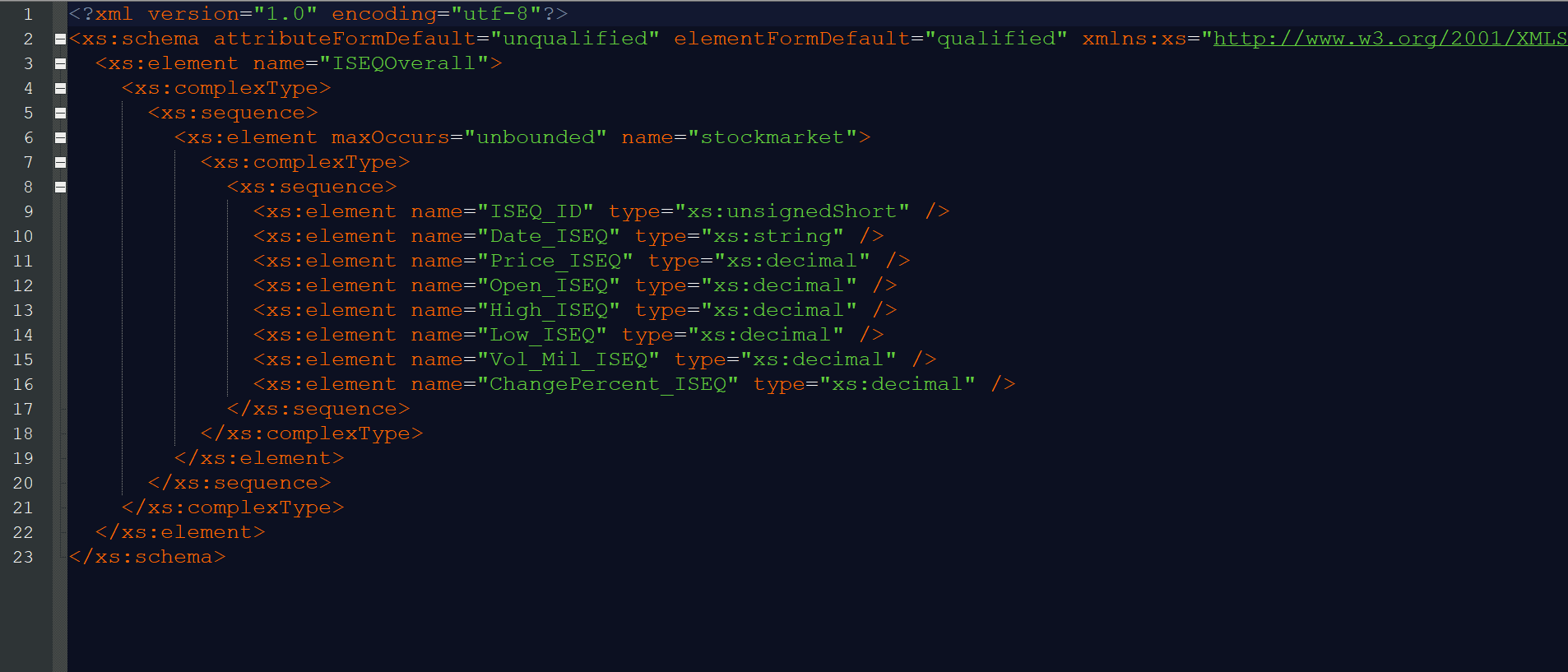
XML is a well defined, general purpose, data transfer language that's designed to facilitate the transfer of data from one system to another in a standard way.

It is a markup language for describing data structures, the contents of those data structures, and the various relationships therein in a universal and abstract format.

## 5.1. Dimension Table - ISEQOverallDim XML:



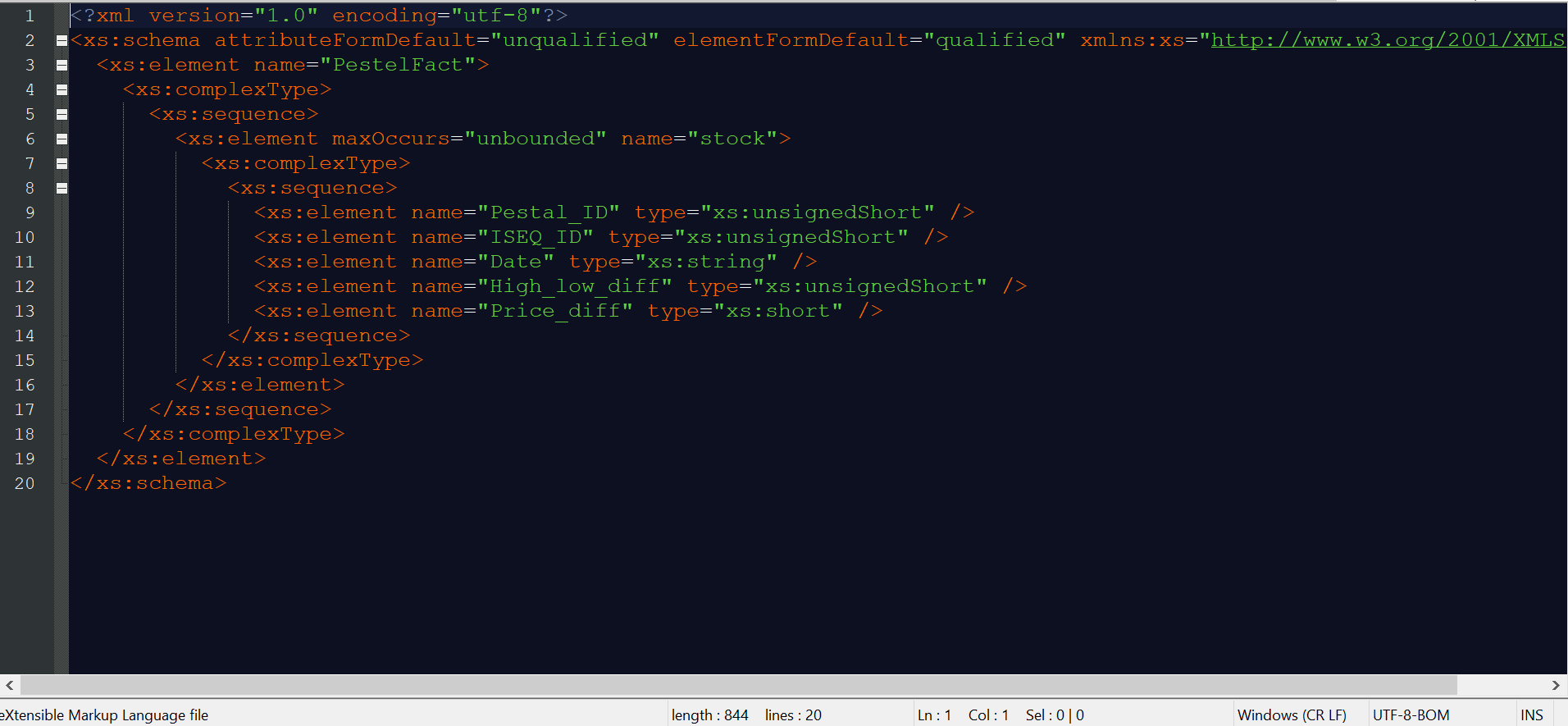
## 5.2. Dimension Table - ISEQOverallDim XSD:



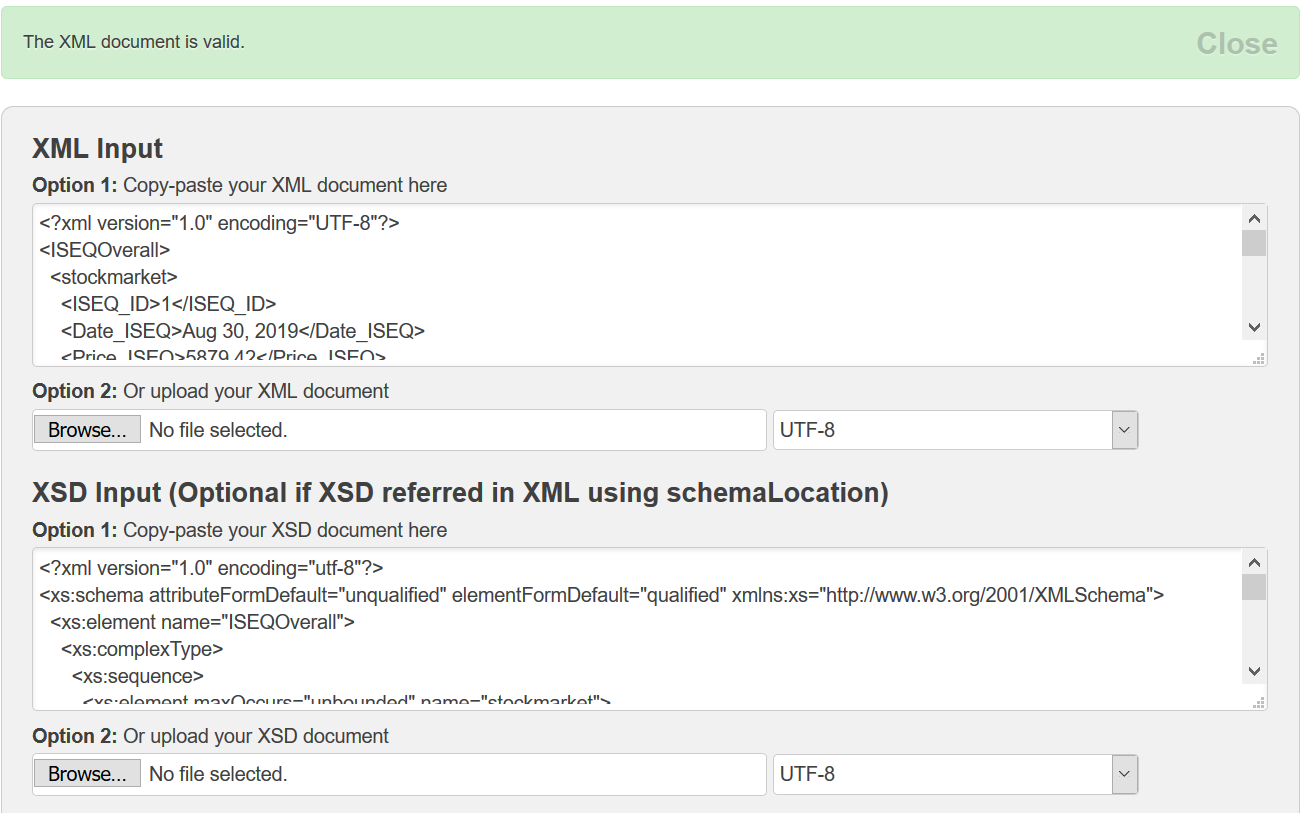
## 5.3. Fact Table - PESTELFact XML:



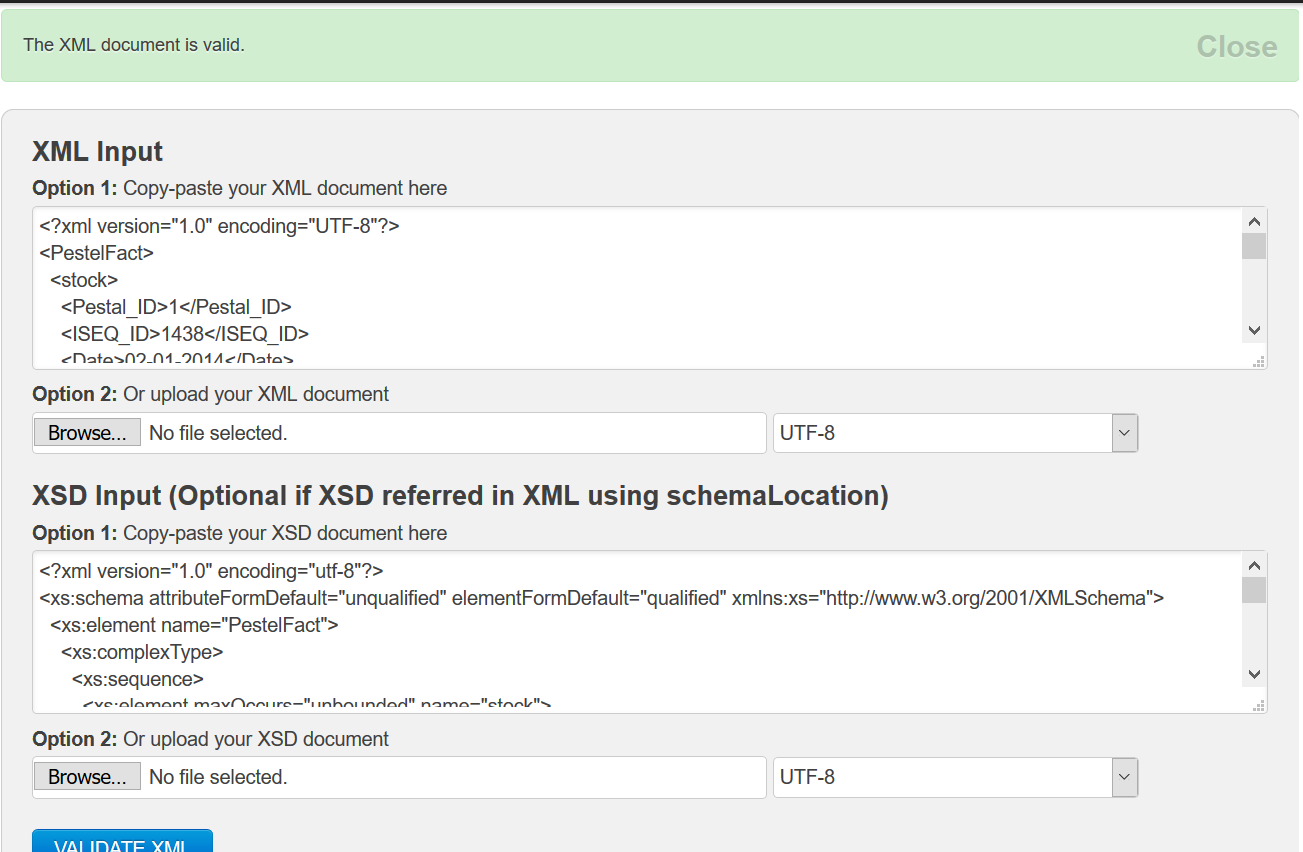
## 5.4. Fact Table - PESTELFact XSD:



## 5.5. XML Validation ISEQOverall

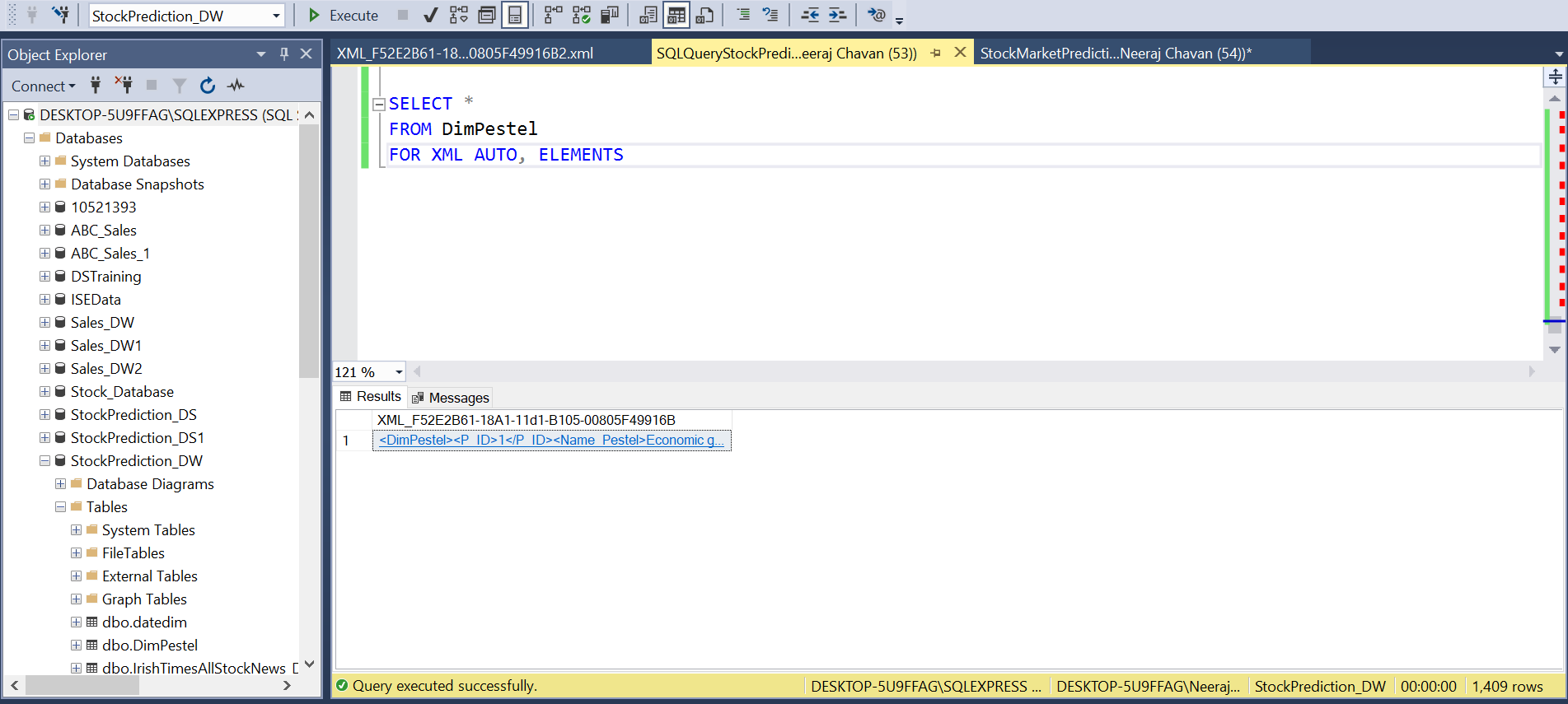


## 5.5. XML Validation PESTELFact

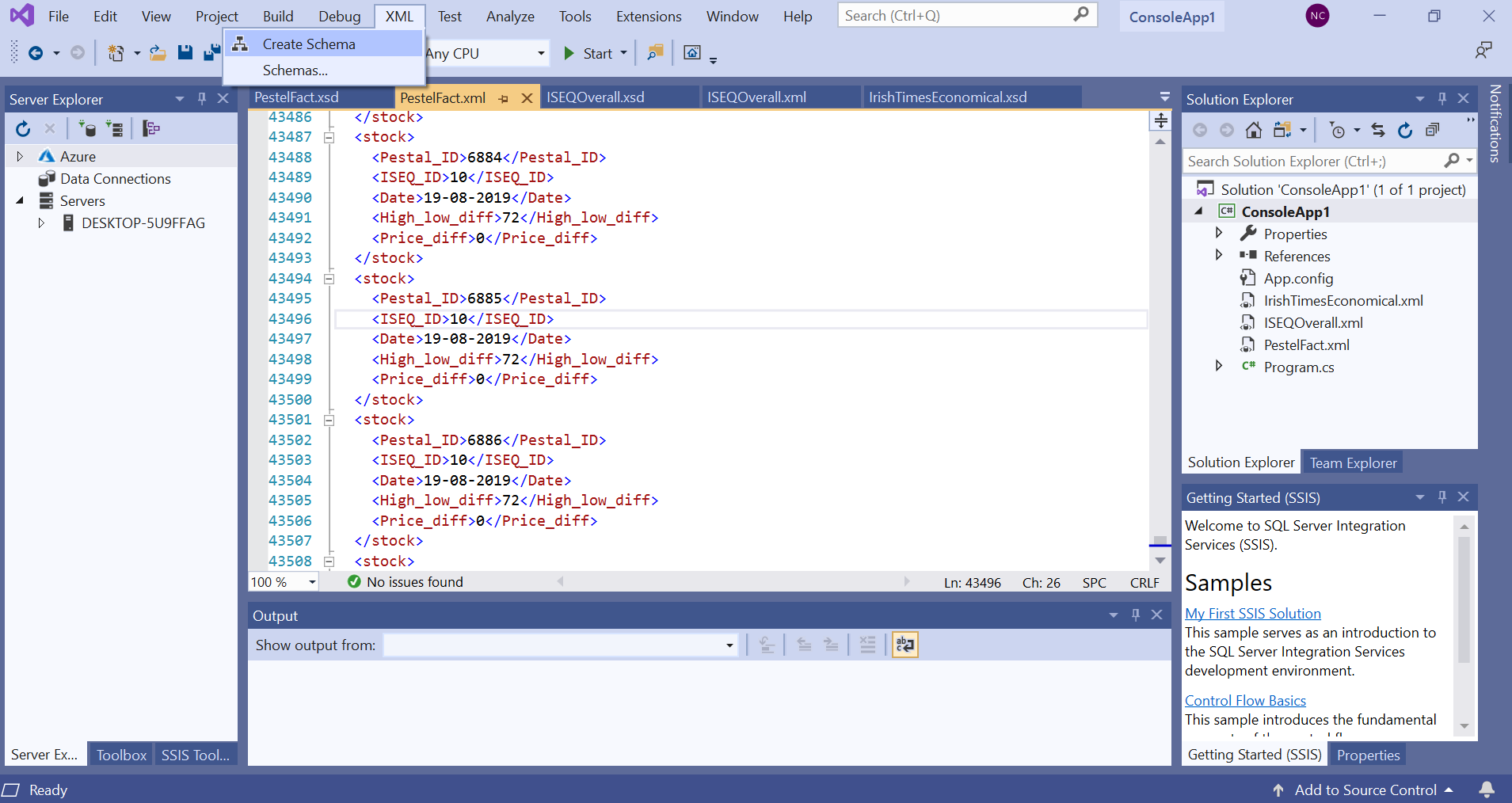


Validator Link: <https://www.freeformatter.com/xml-validator-xsd.html>

## 5.5. Generation of XML:



## 5.6. Generation of XSD from ConsoleApp Using Visual Studio



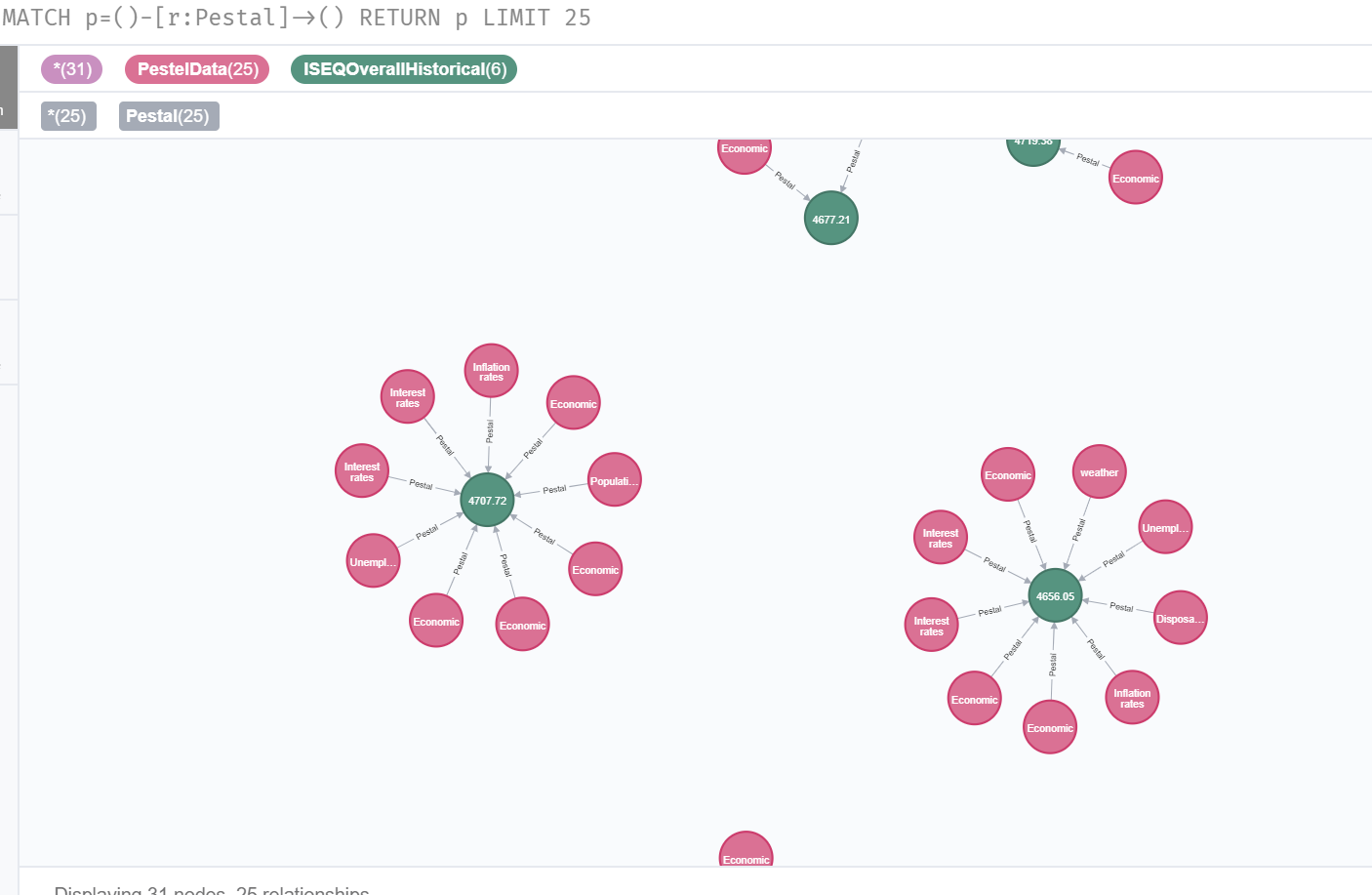
# 6. Graph Databases

6.1 **Implementing Relational Database into Graphical Database**.

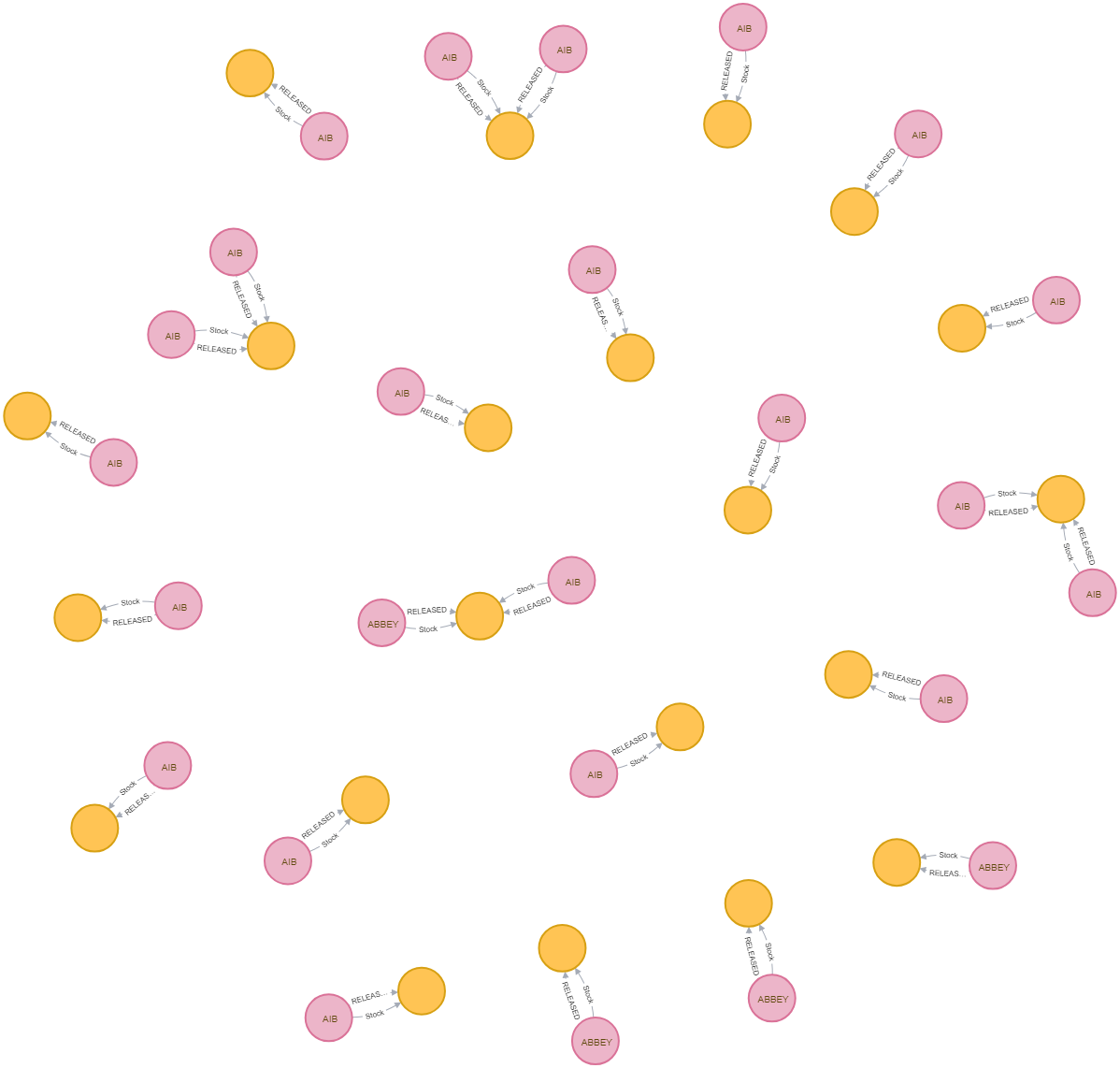
Neo4j is graphical representation of database where data presents in the form of nodes and relationships between them.

Cypher is most commonly used graphical query language.

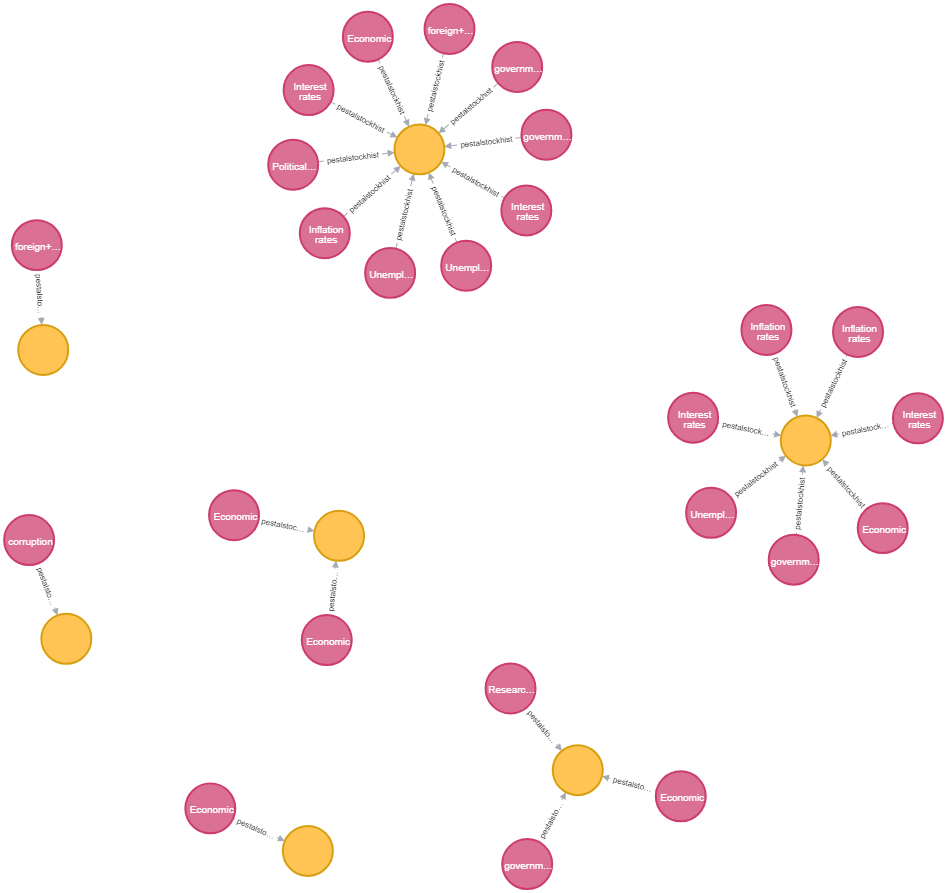
1. Relationship with date between Pestle and ISEQOverAll where we are able to see the entities related to the Pestle Over ISEQOverAll nodes.



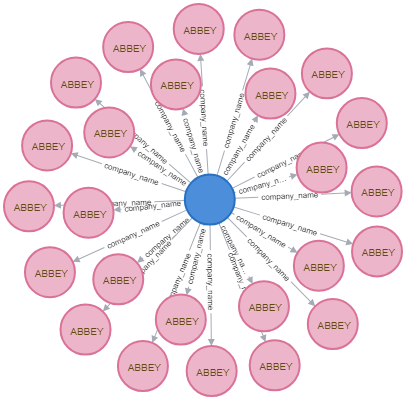
1. Relationship with Date between StockHistorical and IrishTimesAllStockNews where we are able to see the Date entities related to the StockHistorical IrishTimesAllStockNews nodes periocially.



c)Relationship with Date between StockHistorical and Pestle where we are able to see the Date entities related to the StockHistorical over Pestle nodes by date.



d)Relationship of CompanyName between IrishTimesAllStockNews and Stock Details where we are able to see the Company entities related to the Stock Details and IrishTimesAllStockNews nodes.



## 6.1. COMPARISON to relational databases

**Use of Graph database vs Relational database**

a) Graph database is suitable for when data are highly connected and for static tables..

b) Social Networking sites are best example for graphical database where time will be less to construct data and views found in social networking sites.

c) Also,it is best suitable for fraud detection system, access management system and risk management system.

|  |  |
| --- | --- |
| Graphical Database | Relational Database |
| a)Data is denoted in the form of Static tables. | a)Data is denoted in form of Nodes and relationships between each other. |
| b)It reduces efficiency and performance when data set is loaded. | b)The size of the data set does not affect the database operation. |
| c)Adding new constraint to the existing table is difficult. | c)Additional properties and relationship can be done easily. |

# 7. Conclusion

In this project, we have aimed to analyse the impact of real world events and news headlines on the stock market. Having used data scraped from multiple sources for our analysis, we understood how to effectively clean and integrate the data and then store it in the SQL database. Using SSIS we performed the ETL and learnt the various SSIS tools available in Visual Studio. Later we generated reports and visualizations in SSRS and R, to gather insights from the data, as per the business requirements.

We created the relevant XML and Schema. Also, we explored Neo4j graph databases and understood the advantages of graph databases.

Moving on, this project can be further expanded to store data in real time and generate predictions regarding stock prices accordingly.

# 

# 

# 

# 

# 

# 8. Bibliography

# Justin Kuepper (2019) ‘Efficient Market Hypothesis (EMH) Definition’ available at <https://www.investopedia.com/terms/e/efficientmarkethypothesis.asp>

# Jinjian Zhai Nicholas Cohen Anand Atreya (2011) ‘CS224N Final Project: Sentiment analysis of news articles for financial signal prediction’ available at <https://nlp.stanford.edu/courses/cs224n/2011/reports/nccohen-aatreya-jameszjj.pdf>

# Extract, Transform and Load (2011) available at

# <https://en.wikipedia.org/wiki/Extract,_transform,_load>

# 

# 

# Appendix A – VISUALIZATIONS Code

#Installing Package

install.packages("RODBC")

#loading the package

require(RODBC)

#establishing connection to database

conn = odbcDriverConnect("Driver={SQL Server};

server=DESKTOP-JAG58LJ;

database=stock\_analysis\_dw;

trusted\_connection=true")

#storing all the neccesary columns to respective variable

pestal\_ID=sqlQuery(conn,"select pestal\_ID from pestal\_dim" )

name\_pestal=sqlQuery(conn,"select name\_pestal from pestal\_dim" )

date=sqlQuery(conn,"select date from pestal\_dim" )

Headline\_pestal=sqlQuery(conn,"select Headline\_pestal from pestal\_dim" )

Description\_pestal=sqlQuery(conn,"select Headline\_pestal from pestal\_dim" )

Price\_ISEQ=sqlQuery(conn,"select Price\_ISEQ from ISEQOverallHistorical\_dim" )

date\_ISEQ=sqlQuery(conn,"select date from ISEQOverallHistorical\_dim" )

ISEQ=sqlQuery(conn,"select \* from ISEQOverallHistorical\_dim" )

abbey=sqlQuery(conn,"Select Close\_Stock from StockHistorical\_dim where Company\_name='ABBEY'")

aib=sqlQuery(conn,"Select Close\_Stock from StockHistorical\_dim where Company\_name='AIB'")

stockHistorical=sqlQuery(conn,"Select \* from StockHistorical\_dim")

fact1=sqlQuery(conn,"Select \* from stock\_fact")

fact2=sqlQuery(conn,"Select \* from Overall\_ISEQ\_fact")

write.csv(fact2,"C:/Users/bhara/OneDrive/Desktop/fact1.csv", row.names = FALSE)

summary=summary(name\_pestal)

#plot1

table=table(summary)

pie=pie(table,main="Factors in Pestal data")

#plot2

a <- ggplot(ISEQ, aes(x = ISEQ$Price\_ISEQ))

a + geom\_area(stat = "bin")

# creating Default scatter plot of ISEQQ price over last 15 years

p <- ggplot(ISEQ, aes(ISEQ$Price\_ISEQ, ISEQ$date))

p + geom\_point()

abbey$Close\_Stock

#creating Line plot of all the availabe shares with respect to their prices for last 115 days

ds=data.frame(abbey$Close\_Stock[1:115],aib$Close\_Stock)

a=a <- ggplot(ds, aes(abbey$Close\_Stock[1:115],aib$Close\_Stock))

a+geom\_line()

ggplot(stockHistorical, aes(x=stockHistorical$date[100], y=stockHistorical$Close\_Stock[100], group=stockHistorical$Company\_name[100])) +

geom\_line(aes(linetype=stockHistorical$Company\_name[100], color = stockHistorical$Company\_name))+

geom\_point(aes(shape=stockHistorical$Company\_name[100], color = stockHistorical$Company\_name))

# 

# 

# 

# 

# 

# 

# 

# Appendix B – Neo 4J code

6.1) Steps to Convert Relational database into Graphical database.

a) Create NODES for each relational database tables.

LOAD CSV WITH HEADERS FROM "file:///IrishTimesAllStockNews.csv" as row CREATE (c:IrishTimesAllStockNews) SET c=row{News\_ID:row.News\_ID,CompanyName\_News:row.CompanyName\_News,Date\_News:row.Date\_News,Headline\_News:row.Headline\_News,Description\_News:row.Description\_News} return c

#IrishTimesAllStockNews Node Created.

LOAD CSV WITH HEADERS FROM "file:///StockHistorical.csv" as row CREATE (c:StockHistorical) SET c=row{Historical\_ID:row.Historical\_ID,CompanyName:row.CompanyName,Date\_Stock:row.Date\_Stock,Open\_Stock:row.Open\_Stock,High\_Stock:row.High\_Stock,Low\_Stock:row.Low\_Stock,Close\_Stock:row.Close\_Stock,AdjClose\_Stock:row.AdjClose\_Stock,Volume\_Stock:row.Volume\_Stock} return c

#StockHistorical Node Created.

LOAD CSV WITH HEADERS FROM "file:///PestelData.csv" as row CREATE (c:PestelData) SET c=row{P\_ID:row.P\_ID,Name\_Pestal:row.Name\_Pestal,Date\_Pestel:row.Date\_Pestel,Headline\_Pestel:row.Headline\_Pestel,Description\_Pestel:row.Description\_Pestel} return c

#PestelData Node Created.

LOAD CSV WITH HEADERS FROM "file:///StockDetails.csv" as row CREATE (c:StockDetails) SET c=row{CompanyName:row.CompanyName,InstrumentName:row.InstrumentName,MarketCap:row.MarketCap,ISEQWeightingPercent:row.ISEQWeightingPercent,DescriptionStockDetails:row.DescriptionStockDetails,Website:row.Website,DepartmentDetails:row.DepartmentDetails} return c

#StockDetails Node Created.

LOAD CSV WITH HEADERS FROM "file:///ISEQOverallHistorical.csv" as row CREATE (c:ISEQOverallHistorical) SET c=row{ISEQ\_ID:row.ISEQ\_ID,Date\_ISEQ:row.Date\_ISEQ,Price\_ISEQ:row.Price\_ISEQ,Open\_ISEQ:row.Open\_ISEQ,High\_ISEQ:row.High\_ISEQ,Low\_ISEQ:row.Low\_ISEQ,Vol\_Mil\_ISEQ:row.Vol\_Mil\_ISEQ,ChangePercent\_ISEQ:row.ChangePercent\_ISEQ} return c

#ISEQOverallHistorical Node Created.

LOAD CSV WITH HEADERS FROM "file:///FullDateTable.csv" as row CREATE (c:FullDateTable) SET c=row{FullDate:row.FullDate,Date:row.Date,Month:row.Month,Year:row.Year,Day:row.Day,Quater:row.Quater} return c

#FullDateTable Node Created.

b) Create INDEX For each NODE.

$CREATE INDEX ON :IrishTimesAllStockNews(News\_ID)

#Index created for IrishTimesAllStockNews Node.

$CREATE INDEX ON :StockHistorical(Historical\_ID)

#Index created for StockHistorical Node.

$CREATE INDEX ON :PestelData(P\_ID)

#Index created for PestelData Node.

$CREATE INDEX ON :StockDetails(CompanyName)

#Index created for StockDetails Node.

$CREATE INDEX ON :ISEQOverallHistorical(ISEQ\_ID)

#Index created for ISEQOverallHistorical Node.

$CREATE INDEX ON :FullDateTable(FullDate)

#Index created for FullDateTable Node.

c) Create RELATIONSHIP between NODE.

#Relationship with date between Pestal and ISEQOverAll

MATCH (a:PestelData),(b:ISEQOverallHistorical)

WHERE a.Date\_Pestel = b.Date\_ISEQ

CREATE (a)-[r:Pestal]->(b)

RETURN r limit 500

#Relationship with Date between StockHistorical and IrishTimesAllStockNews

MATCH (a:StockHistorical),(b:IrishTimesAllStockNews)

WHERE a.Date\_Stock = b.Date\_News

CREATE (a)-[r:Pestal]->(b)

RETURN r limit 500

#Relationship with Date between StockHistorical and Pestal

MATCH (a:StockHistorical),(b:PestelData)

WHERE a.Date\_Stock = b.Date\_Pestel

CREATE (a)-[r:Pestal]->(b)

RETURN r limit 500

#Relationship of CompanyName between IrishTimesAllStockNews and Stock Details

MATCH (a:IrishTimesAllStockNews),(b:StockDetails)

WHERE a.Date\_News = b.CompanyName

CREATE (a)-[r:Pestal]->(b)

RETURN r limit 500

# 

# Appendix C- Scraping code in python

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

import bs4

import re

import requests

from bs4 import BeautifulSoup

import csv

csv\_file= open('Irish\_times\_legal.csv' , 'w')

csv\_writer= csv.writer(csv\_file)

csv\_writer.writerow(['Name','Date','Headline','Description','Body'])

share\_list=['discrimination+laws','antitrust+laws','consumer+protection+laws','copyright+and+patent+laws','copyright+patent+laws','health+and+safety+laws']

for shares in share\_list:

shares=shares

try:

source= requests.get('https://www.irishtimes.com/search/search-7.4195619?q='+shares+'+market&fromDate=01%2F01%2F2014&toDate=24%2F10%2F2019&page=0').text

#print(source)

soup= BeautifulSoup(source, 'lxml' )

check=soup.find\_all('a', class\_='button-link')

print(check)

length=len(check)-1

last=check[length]['href'].split('=')

#print(last)

leng=len(last)-1

#print(leng)

loop\_end=int(last[leng])+1

#print(loop\_end)

except Exception:

pass

page\_list=[]

for j in range(0,loop\_end):

page\_link="https://www.irishtimes.com/search/search-7.4195619?q="+shares+"+market&fromDate=01%2F01%2F2014&toDate=24%2F10%2F2019&pageId=2.709&page="+str(j)+""

page\_list.append(page\_link)

print(len(page\_list))

article\_list=[]

for k in range(1,len(page\_list)):

source1= requests.get(page\_list[k]).text

soup1= BeautifulSoup(source1, 'lxml' )

for a1 in soup1.find\_all('div', class\_='search\_items\_title'):

link1=a1.span.a['href']

article\_list.append('https://www.irishtimes.com/'+link1+'')

print(len(article\_list))

print(article\_list)

for i in range(1,len(article\_list)):

try:

print(article\_list[i])

data= requests.get(article\_list[i]).text

soup2 = bs4.BeautifulSoup(data, 'html.parser')

date=soup2.find('div', class\_='time-metadata').time.text

results = soup2.body.find\_all(string=re.compile('.\*{0}.\*'.format(shares)), recursive=True)

row = soup2.find\_all('div', string=shares)

article=soup2.find(name='hgroup')

headline= article.h1.text

description=article.h2.text

pData1=''

body=soup2.find\_all("div", class\_="article\_bodycopy", id=False)

for tag in body:

for element in tag.find\_all("p"):

pData = element.text

pData1= pData1+'\n'+pData

print(date)

print(headline)

print(description)

#print(pData1)

csv\_writer.writerow([shares,date,headline,description,pData1])

except Exception:

pass

except ConnectionError:

pass

except:

pass

csv\_file.close()

# Appendix D- SQL Query:

use StockPrediction\_DS

--adding data in source tables

CREATE TABLE ISEQOverallHistorical (

ISEQ\_ID int NOT NULL,

Date\_ISEQ date,

Price\_ISEQ int,

Open\_ISEQ int,

High\_ISEQ int,

Low\_ISEQ int,

Volume\_ISEQ int,

ChangePercent\_ISEQ decimal (3,2),

CONSTRAINT pk\_ISEQ\_ID PRIMARY KEY (ISEQ\_ID),

);

CREATE TABLE StockDetails (

Company\_name varchar(100) NOT NULL,

InstrumentName nvarchar(100),

MarketCap bigint,

ISEQWeightingPercent int,

DescriptionStockDetails nvarchar(800),

Website nvarchar(500),

DepartmentDetails nvarchar(100),

CONSTRAINT pk\_Company\_name PRIMARY KEY (Company\_name),

);

CREATE TABLE StockHistorical (

Historical\_ID int NOT NULL,

Company\_name varchar(100) NOT NULL,

Date\_Stock date,

Open\_Stock int,

High\_Stock int,

Low\_Stock int,

Close\_Stock int,

AdjClose\_Stock int,

Volume\_Stock bigint,

CONSTRAINT pk\_Historical\_ID PRIMARY KEY (Historical\_ID),

);

CREATE TABLE IrishTimesAllStockNews (

News\_ID int NOT NULL,

Company\_name\_News varchar(100),

Date\_News date,

Headline\_News nvarchar(500),

Description\_News nvarchar(500),

CONSTRAINT pk\_News\_ID PRIMARY KEY (News\_ID)

);

CREATE TABLE IrishTimesPolitical (

Political\_ID int NOT NULL,

Name\_political varchar(50),

Date\_Political date,

Headline\_Political nvarchar(500),

Description\_Political nvarchar(500),

CONSTRAINT pk\_Political\_ID PRIMARY KEY (Political\_ID),

);

CREATE TABLE IrishTimesEconomical (

Economical\_ID int NOT NULL,

Name\_Economical varchar(50),

Date\_Economical date,

Headline\_Economical nvarchar(500),

Description\_Economical nvarchar(500),

CONSTRAINT pk\_Economical\_ID PRIMARY KEY (Economical\_ID),

);

CREATE TABLE IrishTimesSocial (

Social\_ID int NOT NULL,

Name\_Social varchar(50),

Date\_Social date,

Headline\_Social nvarchar(500),

Description\_Social nvarchar(500),

CONSTRAINT pk\_Social\_ID PRIMARY KEY (Social\_ID),

);

CREATE TABLE IrishTimesTech (

Tech\_ID int NOT NULL,

Name\_Tech varchar(50),

Date\_Tech date,

Headline\_Tech nvarchar(500),

Description\_Tech nvarchar(500),

CONSTRAINT pk\_Tech\_ID PRIMARY KEY (Tech\_ID),

);

CREATE TABLE IrishTimesEnvior (

Envior\_ID int NOT NULL,

Envior\_Tech varchar(50),

Date\_Envior date,

Headline\_Envior nvarchar(500),

Description\_Envior nvarchar(500),

CONSTRAINT pk\_Envior\_ID PRIMARY KEY (Envior\_ID),

);

CREATE TABLE IrishTimesLegal (

Legal\_ID int NOT NULL,

Legal\_Tech varchar(50),

Date\_Legal date,

Headline\_Legal nvarchar(500),

Description\_Legal nvarchar(500),

CONSTRAINT pk\_Legal\_ID PRIMARY KEY (Legal\_ID),

);

use stock\_analysis\_dw

---creating data base dimentions

create table pestal\_dim(

pestal\_ID int NOT NULL,

name\_pestal nvarchar(50),

date date,

Headline\_pestal nvarchar(500),

Description\_pestal nvarchar(500),

CONSTRAINT pk\_pestal\_ID PRIMARY KEY (pestal\_ID),

);

select \* from ISEQOverallHistorical\_dim

CREATE TABLE ISEQOverallHistorical\_dim (

ISEQ\_ID int NOT NULL,

date date,

Price\_ISEQ int,

Open\_ISEQ int,

High\_ISEQ int,

Low\_ISEQ int,

Volume\_ISEQ int,

ChangePercent\_ISEQ decimal (3,2),

CONSTRAINT pk\_ISEQ\_ID PRIMARY KEY (ISEQ\_ID),

);

CREATE TABLE StockDetails\_dim (

Company\_name varchar(100) NOT NULL,

InstrumentName nvarchar(100),

MarketCap bigint,

ISEQWeightingPercent int,

DescriptionStockDetails nvarchar(800),

Website nvarchar(500),

DepartmentDetails nvarchar(100),

CONSTRAINT pk\_Company\_name PRIMARY KEY (Company\_name),

);

CREATE TABLE StockHistorical\_dim (

Historical\_ID int NOT NULL,

Company\_name varchar(100) NOT NULL,

date date,

Open\_Stock int,

High\_Stock int,

Low\_Stock int,

Close\_Stock int,

AdjClose\_Stock int,

Volume\_Stock bigint,

CONSTRAINT pk\_Historical\_ID PRIMARY KEY (Historical\_ID),

);

CREATE TABLE IrishTimesAllStockNews\_dim (

News\_ID int NOT NULL,

Company\_name varchar(100),

date date,

Headline\_News nvarchar(500),

Description\_News nvarchar(500),

CONSTRAINT pk\_News\_ID PRIMARY KEY (News\_ID)

);

select \* from IrishTimesAllStockNews\_dim

CREATE TABLE datedim (

Fulldate date NOT NULL,

date int,

Month int,

Year int,

Day nvarchar(100),

Quater nvarchar(100),

CONSTRAINT pk\_Fulldate PRIMARY KEY (Fulldate)

);

CREATE TABLE stock\_fact (

Historical\_ID int,

Company\_name varchar(100) ,

News\_ID int,

Fulldate date,

Price\_diff\_high\_low\_ISEQ int,

);

CREATE TABLE Overall\_ISEQ\_fact(

pestal\_ID int ,

ISEQ\_ID int ,

Date date,

High\_low\_diff int,

Price\_diff int

);

---Fact table creation queries

SELECT IrishTimesAllStockNews\_dim.date,StockHistorical\_dim.Company\_name,StockHistorical\_dim.Historical\_ID,IrishTimesAllStockNews\_dim.News\_ID,

StockHistorical\_dim.High\_Stock - (StockHistorical\_dim.Low\_Stock) as high\_low

FROM IrishTimesAllStockNews\_dim INNER JOIN

StockHistorical\_dim ON IrishTimesAllStockNews\_dim.date = StockHistorical\_dim.date

SELECT ISEQOverallHistorical\_dim.date,pestal\_dim.pestal\_ID,ISEQOverallHistorical\_dim.ChangePercent\_ISEQ,

ISEQOverallHistorical\_dim.ISEQ\_ID,ISEQOverallHistorical\_dim.High\_ISEQ - (ISEQOverallHistorical\_dim.Low\_ISEQ) as high\_low,

ISEQOverallHistorical\_dim.Price\_ISEQ - coalesce(lag(ISEQOverallHistorical\_dim.Price\_ISEQ)

over (order by ISEQOverallHistorical\_dim.ISEQ\_ID), 1) as Price\_diff

From ISEQOverallHistorical\_dim INNER JOIN pestal\_dim ON ISEQOverallHistorical\_dim.date=pestal\_dim.date

---SSRS Report Queries

select Overall\_ISEQ\_fact.High\_low\_diff,Overall\_ISEQ\_fact.date,pestal\_dim.Headline\_pestal,pestal\_dim.Description\_pestal from Overall\_ISEQ\_fact

INNER JOIN pestal\_dim on Overall\_ISEQ\_fact.pestal\_ID=pestal\_dim.pestal\_ID where Overall\_ISEQ\_fact.High\_low\_diff>49

select pestal\_dim.name\_pestal,stock\_fact.Fulldate,count(stock\_fact.Price\_diff\_high\_low\_ISEQ) as 'Number of Articles'

from stock\_fact inner join pestal\_dim on pestal\_dim.date=stock\_fact.Fulldate where

stock\_fact.Price\_diff\_high\_low\_ISEQ>4.9 group by stock\_fact.Fulldate,pestal\_dim.name\_pestal

select p.date,p.name\_pestal,count(p.name\_pestal) as cnt from pestal\_dim p group by date,name\_pestal

select name\_pestal,count(\*) as Frequency from pestal\_dim group by name\_pestal