US Stock Database

Provide daily and monthly market and corporate action data for active securities
Help investors make well-informed investment decisions

URL: http://ugrad.cs.jhu.edu/~nju1/HomePage.html

Ning Ju (615) nju1@jhu.edu

Hongqiao Mu (615) hmu3@jhu.edu

Jiawen Bai (615) jiawen.bai@jhu.edu

1. Introduction

The US Stock Databases contain daily and monthly market and corporate action data for active securities with primary listings on the NYSE, NASDAQ exchanges. Investors have access to latest data to make well-informed investment decisions. The US Stock Database provides the following information. Before reading we highly recommend you to try our pilot product here: http://ugrad.cs.jhu.edu/~nju1/HomePage.html

Data for individual securities:

- Price and quote data: open, close, bid/low, ask/high (We only include data from September 9, 2016 to December 13, 2016 due to space limitation.)
- Trading volume
- Corporate actions
- Identifiers, descriptors, and supplemental data items

Exchange Coverage:

- NYSE - data series begin on January 1, 2016
- NASDAQ - data series begin on January 1, 2016

Market Indexes:

- NASDAQ Composite Index
- Dow Jones Industrial Average
- S&P 500 Composite Index
- Russell 2000 Index

2. Data Extraction

We extract raw data from Wikipedia and official websites of exchanges and companies using Python. Part of raw data sources are listed due to space limitation.

- https://finance.yahoo.com/
- https://www.nasdaq.com/
- https://www.nyse.com/index
- http://www.morganstanley.com/
- https://www.sec.gov/edgar.shtml

We clean the raw data using Python and Excel and utilize a python script to generate SQL insert into statements. Here is a sample:

```
filename = "REPORT"
data = pd.read_excel(filename+".xlsx")
output = open(filename+".txt", "w")
N, d = data.shape
begin = "insert into " + filename + " values ("
for i in range(N):
    command = begin
    for j in range(d):
```

```
    if pd.isnull(data.iloc[i,j]):
      break
    if j == d-1:
      command += "'" + str(data.iloc[i,j]) + "');\n"
      output.write(command)
    else:
      if np.isreal(data.iloc[i,j]):
        command += "'" + str(data.iloc[i,j]) + "', "
      else:
        command += "'" + str(data.iloc[i,j]) + "', "
output.close()
```

A formal SQL Database definition of the relational database is in CREATE_TABLE.sql. Here is a sample:

```
CREATE TABLE SECURITIES (
                Symbol varchar(10)
               ,CIK varchar(10)
               ,Security_Name varchar(50)
               ,ExchangeID varchar(4)
               ,CityID varchar(4)
               ,SectorID varchar(8)
               ,Founded VARCHAR(20)
               ,PRIMARY KEY (Symbol)
               );

CREATE TABLE EXCHANGES (
                 ExchangeID varchar(4)
               , Exchange_Name varchar(50)
               , CityID VARCHAR(4)
               , Market_Cap int(15)
               , MOnthly_Trade_Volume int(10)
               , Open_UTC time
               , Close_UTC time
               , PRIMARY KEY (ExchangeID)
              );
```

3. User Guide

This database aims to provide information on US stock market including individual securities, exchanges and major indexes. Based on users' need and preference, we build a user-friendly interface which could achieve many functions. You could visit our home page: http://ugrad.cs.jhu.edu/~nju1/HomePage.html . Figure 1 shows the home page.

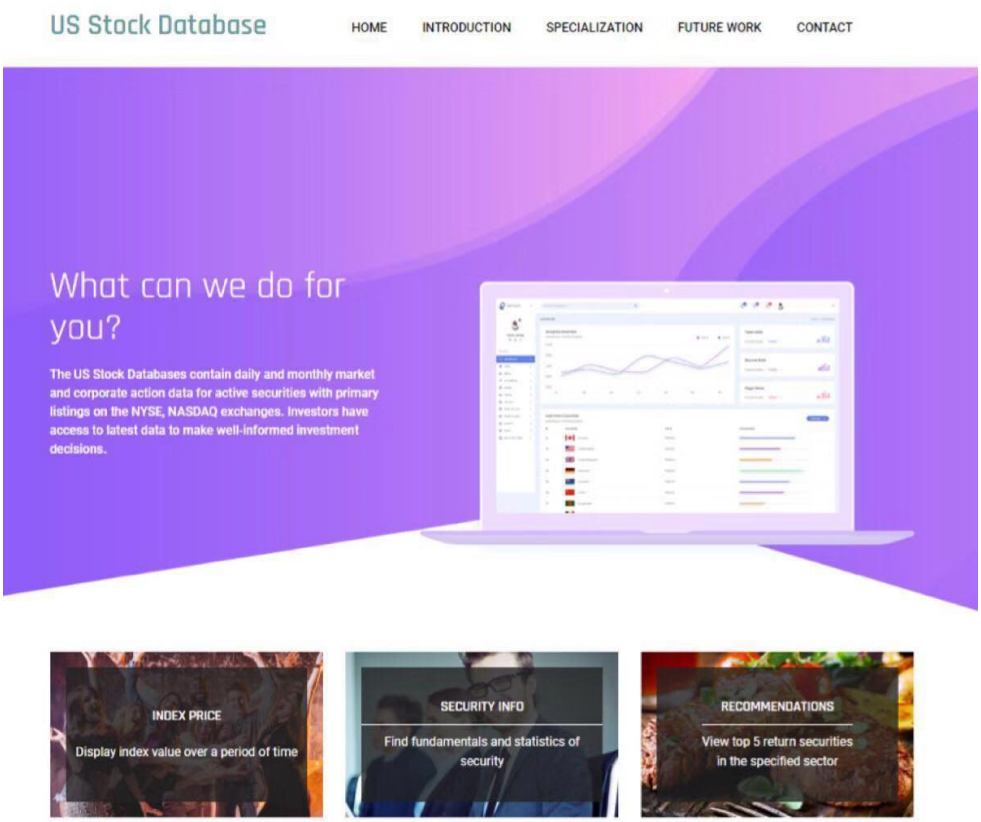


Figure 1 Home Page of Interface

Our interface could achieve the following functions:

- **Display the value of a specified index over a period of time**
  In this interface, users could choose an index which is included in our database and specify the begin date and end date of query period as Figure 2 suggested. The output is the index value over the query period and a line chart illustrating the trend. The following screenshot of our website illustrates an example in which the index is S&P 500 and the query period is from December 1 2018 to January 31 2018. The output also includes a line chart of index value over the selected period, general information of the index and daily price information. Figure 3 illustrates the line chart.

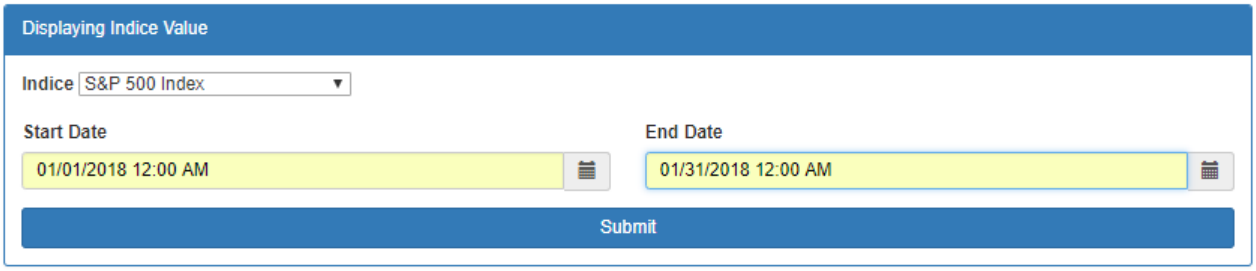


Figure 2 Interface of displaying the value of index

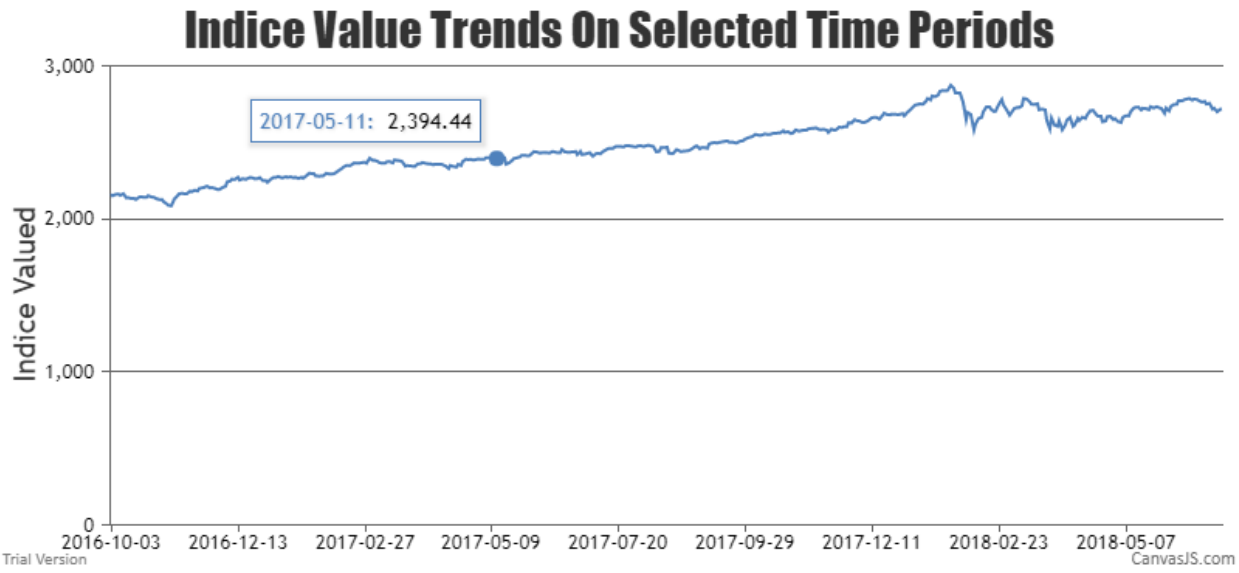


Figure 3 Output of function1: Indice Value Trend

- **Find the statistics and fundamentals of a specified security**
  Users could enter symbol of a security (such as AAPL) and the outputs are fundamentals including assets, liabilities, revenue, earnings and EPS, and some additional information about the security. This function gives users a brief understanding of the selected security.

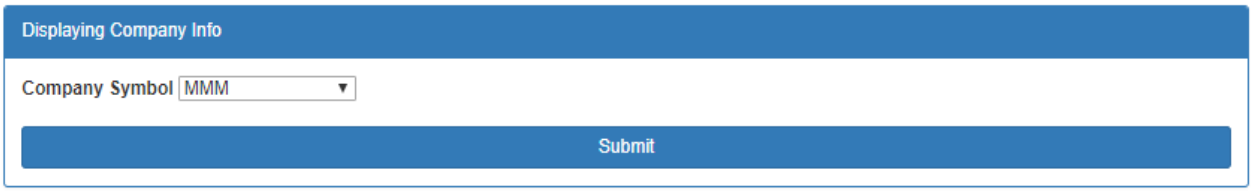


Figure 4 Interface of finding statistics and fundamentals of security

Figure 5 illustrates the output when the input symbol is 'MMM'.

General Information About MMM

Symbol	CIK	Security_Name	Exchange_Name	GICS_Sector	GICS_Industry	GICS_Sub_Industry	City	Founded
MMM	0000066740	3M Company	New York Stock EXCHANGES	Industrials	Industrial Conglomerates	Industrial Conglomerates	St. Paul	1902

Symbol	Holding_Percentage	No_of_Holders	Total_Shares_Hold	Total_value_of_Holding	Date_observed
MMM	0.67570	1995	393433732	80618506024	2018-11-16

Symbol	IndiceID	IndiceID	Indice	Type_Name	Constituents	Weighting
MMM	SPX	SPX	S&P 500 Index	Large cap	505	Free-float capitalization-weighted
MMM	DJIA	DJIA	Dow Jones Industrial Average	Large cap	30	Price-weighted

Symbol	Date_Observed	Assets	Liabilities	Revenue	Eanings	EPS
MMM	2013-12-31	33550000000.0	16048000000.0	30871000000.0	32416000000.0	6.83
MMM	2014-12-31	31209000000.0	18100000000.0	31821000000.0	34317000000.0	7.63
MMM	2015-12-31	32883000000.0	21454000000.0	30274000000.0	36296000000.0	7.72

Figure 5 Output of function2

- **View top 5 highest return securities in a given sector**
  Users could choose a sector (such as Health Care) and a date and the return value is the

top 5 highest return securities in the given sector. For example, if we ask to view top 5 highest return securities in Health Care sector on November 17, 2016 as Figure 6 suggested, the output will be Figure 7.

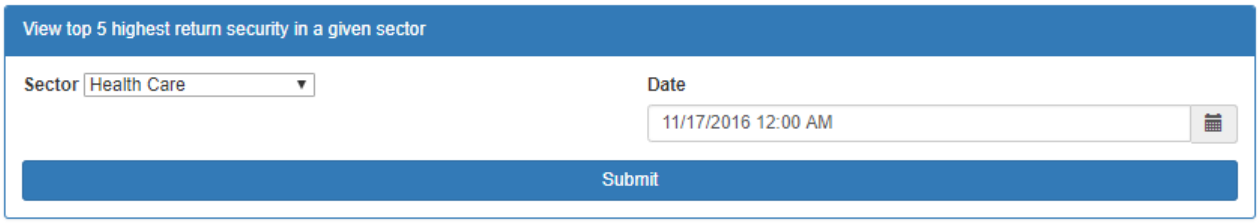


Figure 6 Interface of viewing top 5 securities in sector

General Information About Health Care

Security_Name	dayreturn
Centene Corporation	0.0305206637
Universal Health Services, Inc.	0.0295852816
Hologic	0.0271565754
DaVita Inc.	0.0263245024
AmerisourceBergen Corp	0.0250640291

Figure 7 Function3 output

- **Recommend the best fund satisfying users' need**
  Users could choose fund type like Equity, and the system would recommend the best fund of the selected type.

Besides, the following 16 query questions could be done in our database. The implementations of these queries are included in 'SQL_QUERY.sql'

- List the institutions holding shares in more than 5 companies which are components of S&P 500.
- List the institutions which invest the highest weight in healthcare.
- List the number of securities in Information Technology sector which was founded before 2000.
- List the company names which posted a file on November 1, 2018.
- List the NYSE companies which locate in New York City.
- List the companies which has the highest daily return on November 9, 2018 and the return.
- List the companies which are components of at least two indices.
- List the companies which have more liabilities than assets in 2018.
- List all investment institutions and its highest investment-weighted sector.
- List the File Number of Apple which was posted after November 1, 2018.
- List the number of 8-K file of Apple in 2018.
- List the number of 8-K file of securities in NYSE in 2018.
- List the Institution which hold the most shares in Twitter.
- List the securities holding by institutions whose fund have highest YTD on November 9, 2018.
- List the major institutional holder of the security which has the highest daily return on

November 9,2018.

- List the assets, liabilities, revenues of securities whose location is the same as Paypal.

4. Area of Specialization

In this section, we mainly discuss our areas of specialization. Our major work is to build an interesting interface and to extract complex real data from online sources. We also works on data visualization as discussed in section 3 where we plot a line chart of index value trend.

- **Interesting interface**
  We develop a user-friendly interface modifying a template downloaded from https://colorlib.com/wp/free-html-website-templates/ . The user interface works well on both PC and mobile. Besides, we provide multiple functions based on investors' concerns offering information about securities, exchanges and indexes as discussed above. We have shown the user interface of PC in Section 3 and Figure 8 shows the mobile user interface. Also, instead of barely displaying the results in a table, we provide line plots to help user to analyze the price behavior of indices and securities.

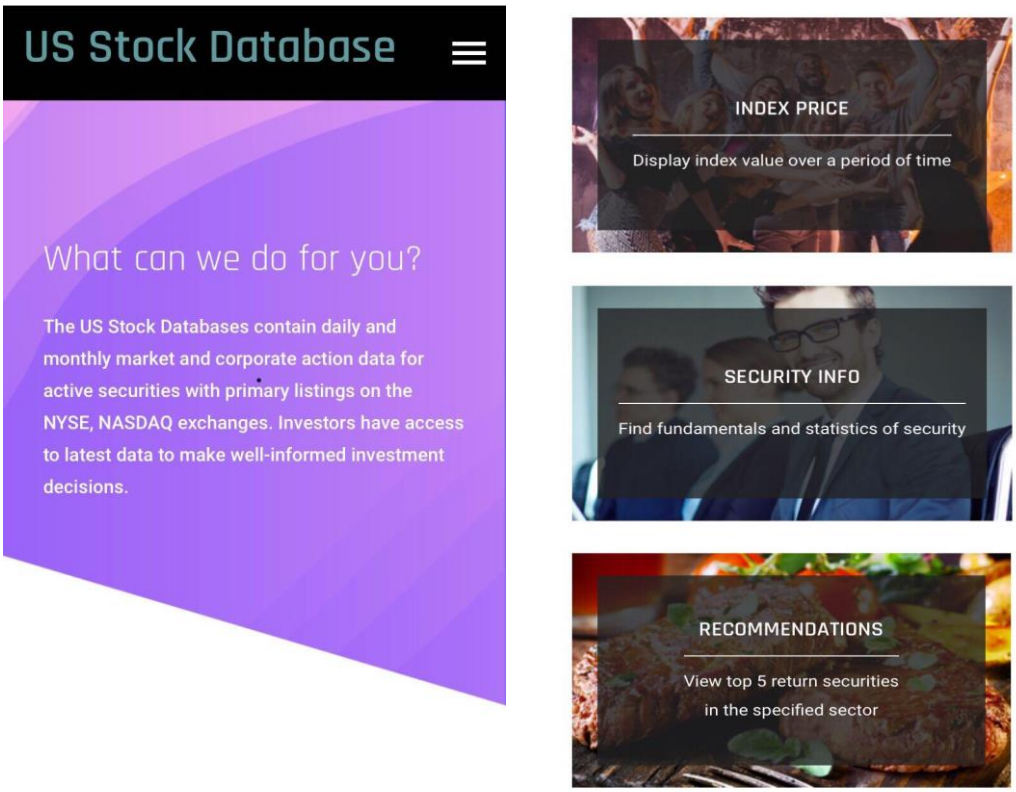


Figure 8 Mobile User Interface

Figure 9 illustrates the query and output on mobile interface.

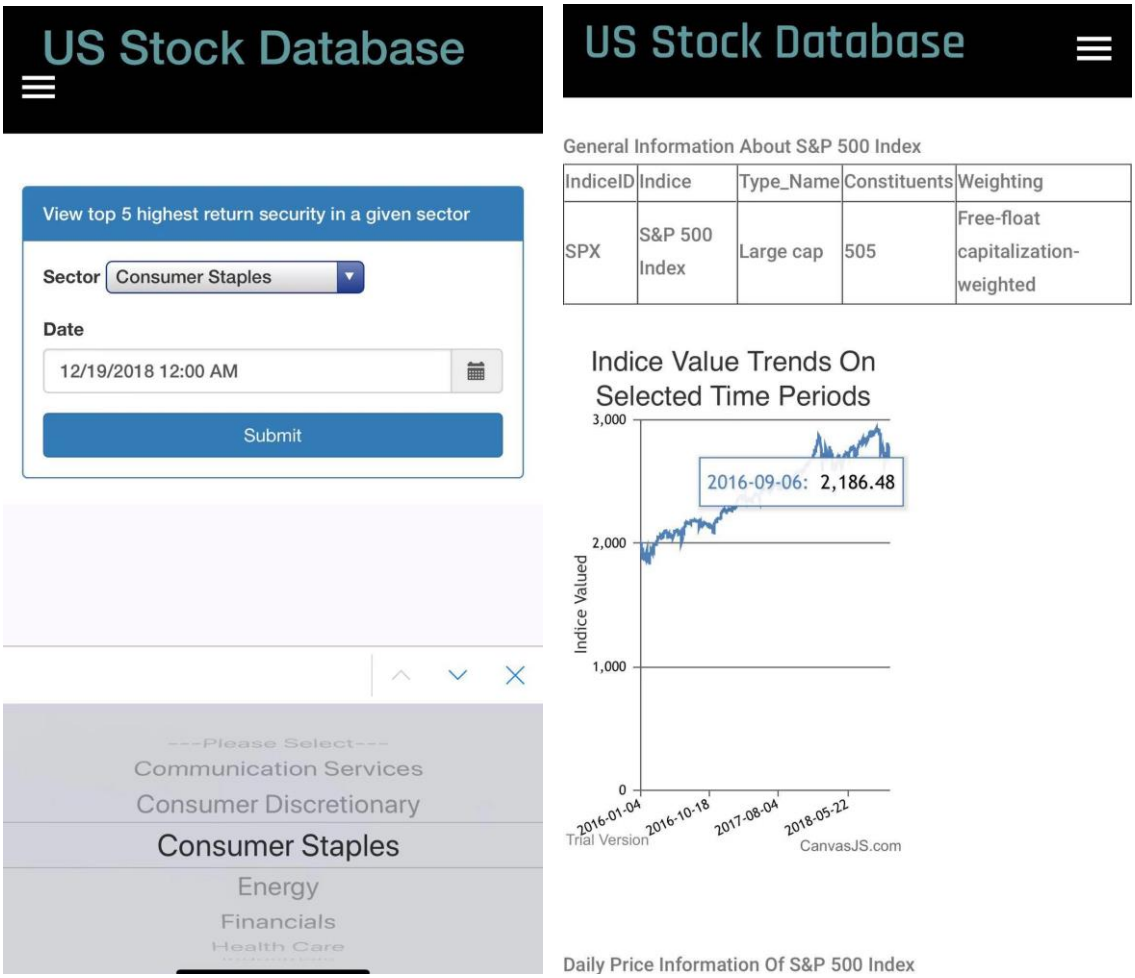


Figure 9 Mobile User Interface query and output

- **Complex extraction of real data from online sources**
  The data extraction, cleanse and SQL code generation of this project takes significant amount of time in our work. It's difficulty for us to find well-prepared initial data so we extract raw data from Wikipedia and official websites of exchanges and companies using Python and spend much time on data extraction and cleaning. The sample code we developed for extracting data is followed.

```
from bs4 import BeautifulSoup
import requests
import pandas as pd

url = 'https://xxxxx'
res = requests.get(url)
soup = BeautifulSoup(res.text, 'lxml')
tables = soup.select('table')
df_list = []
for table in tables:
df_list.append(pd.concat(pd.read_html(table.prettify())))
df = pd.concat(df_list)
```

```
df.to_excel('report1.xlsx')
```

5. Achievements and Strengths

Raw data is downloaded from multiple online free sources. The initial data is unstructured and ugly so we spend much time on data extraction and cleanse. We reduce redundancy and then extract useful information from the large dataset. Also, we provide several functions to meet investors' need.

Besides, it is difficult for an individual investor to get free access to stock information. Investors need to pay a lot to investment companies such as Morningstar to get valuable information, but our database provide individual investors with free and accessible data. Individual investors could use these data to make informed decisions.

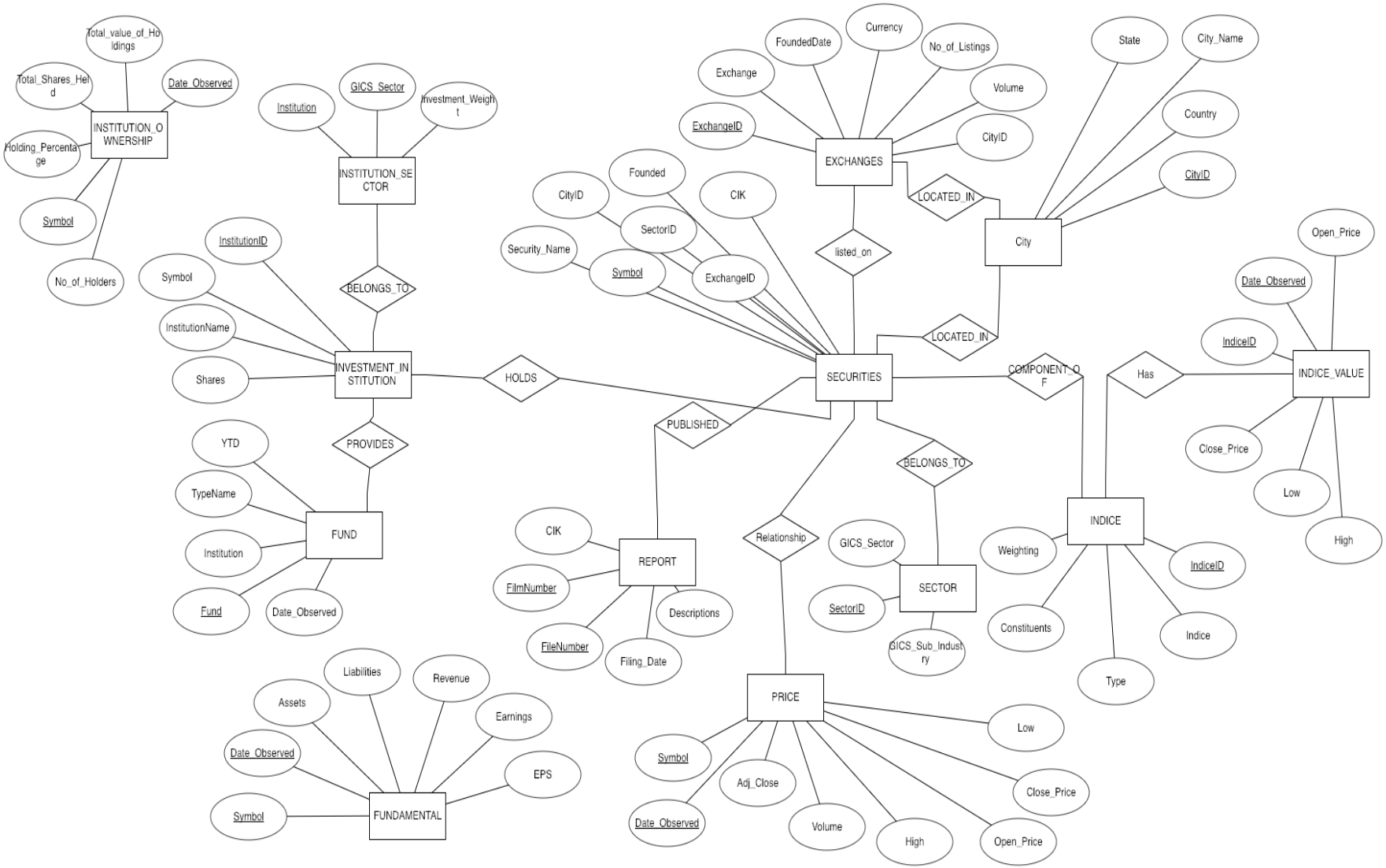
6. Limitation and Future Works

Since the initial data is unstructured it takes us a lot of time to extract and clean data. We don't insert all the data into our database due to time limitation. Also, we only include price data in 2016 due to space limitation. We will keep working on this in the future to enrich our database.

Besides, we want to build a decision support system that help investors to make investment decisions. It is hard for individual investors to make scientific and reasonable decisions since they are easily influenced by emotions. For improvement, we can use data mining tools to build a decision system and offer investors these suggestions.

7. Appendix

(1) Here's the entity relation diagram used in our project:


Total_value_of_Holdings
Total_Shares_Held
Date_Observed
INSTITUTION_OWNERSHIP
Holding_Percentage
Symbol
No_of_Holders
Institution
GICS_Sector
Investment_Weight
INSTITUTION_SECTOR
BELONGS_TO
InstitutionID
Symbol
InstitutionName
Shares
INVESTMENT_INSTITUTION
HOLDS
PROVIDES
YTD
TypeName
Institution
Fund
Date_Observed
FUND
Liabilities
Revenue
Assets
Earnings
Date_Observed
EPS
Symbol
FUNDAMENTAL
FoundedDate
Currency
Exchange
No_of_Listings
Volume
ExchangeID
CityID
EXCHANGES
State
City_Name
Country
CityID
LOCATED_IN
City
Founded
CityID
CIK
SectorID
Security_Name
Symbol
ExchangeID
listed_on
LOCATED_IN
SECURITIES
PUBLISHED
CIK
FilmNumber
FileNumber
Filing_Date
Descriptions
REPORT
Relationship
PRICE
Symbol
Adj_Close
Date_Observed
Volume
High
Low
Close_Price
Open_Price
BELONGS_TO
GICS_Sector
SectorID
GICS_Sub_Industry
SECTOR
COMPONENT_OF
Has
INDICE
Weighting
IndiceID
Constituents
Indice
Type
Open_Price
Date_Observed
IndiceID
INDICE_VALUE
Close_Price
Low
High

(2) Attached is the full relational table specification of our database in the SQL Database Definition Language.

```
DROP TABLE IF EXISTS SECURITIES;
CREATE TABLE SECURITIES (
                Symbol varchar(10)
               ,CIK varchar(10)
               ,Security_Name varchar(50)
               ,ExchangeID varchar(4)
               ,CityID varchar(4)
               ,SectorID varchar(8)
               ,Founded VARCHAR(20)
               ,PRIMARY KEY (Symbol)
               ,FOREIGN KEY (CIK) REFERENCES REPORT
               );

DROP TABLE IF EXISTS EXCHANGES;
CREATE TABLE EXCHANGES (
                 ExchangeID varchar(4)
               , Exchange_Name varchar(50)
               , CityID VARCHAR(4)
               , Market_Cap int(15)
               , MOnthly_Trade_Volume int(10)
               , Open_UTC time
               , Close_UTC time
               , PRIMARY KEY (ExchangeID)
              );

DROP TABLE IF EXISTS SECTOR;
CREATE TABLE SECTOR  (
               SectorID varchar(8)
              , GICS_Sector varchar(70)
              , GICS_Industry varchar(70)
              , GICS_Sub_Industry varchar(70)
              , PRIMARY KEY (SectorID)
               );


DROP TABLE IF EXISTS PRICE;
CREATE TABLE PRICE   (
               Symbol varchar(10)
              ,Date_Observed Date
              ,Open_Price decimal(10,6)
              ,High_Price decimal(10,6)
              ,Low_Price decimal(10,6)
              ,Close_Price decimal(10,6)
              ,Volume int(20)
              ,PRIMARY KEY (Symbol,Date_observed)
              );
DROP TABLE IF EXISTS FUNDAMENTAL;
CREATE TABLE FUNDAMENTAL (
                 Symbol varchar(10)
                ,Date_Observed Date
                ,Assets Decimal(21,1)
                ,Liabilities Decimal(21,1)
```

```
                ,Revenue Decimal(21,1)
                ,Eanings Decimal(21,1)
                ,EPS decimal(5,2)
                ,PRIMARY KEY (Symbol,Date_observed)
                );
DROP TABLE IF EXISTS REPORT;
CREATE TABLE REPORT (
             Filling varchar(15)
                    ,Descriptions varchar(200)
                    ,Filling_Date Date
                    ,FileNumber varchar(15)
            ,FilmNumber varchar(15)
            ,CIK varchar(15)
            ,PRIMARY KEY (FileNumber,FilmNumber)
            );

DROP TABLE IF EXISTS INSTITUTIONOWNERSHIP;
CREATE TABLE INSTITUTIONOWNERSHIP (
                      Symbol varchar(10)
                     ,Holding_Percentage DECIMAL(6,5)
                     ,No_of_Holders int(10)
                     ,Total_Shares_Hold bigint(50)
                     ,Total_value_of_Holding bigint(50)
                     ,Date_observed Date
                     ,PRIMARY KEY (Symbol)
                     );

DROP TABLE IF EXISTS MAIN_HOLDER;
CREATE TABLE MAIN_HOLDER (
                Symbol varchar(10)
               ,Institution varchar(50)
               ,Shares int(20)
               ,PRIMARY KEY (Symbol,Institution)
               );

DROP TABLE IF EXISTS INSTITUTION_SECTOR;
CREATE TABLE INSTITUTION_SECTOR (
   INSTITUTION VARCHAR(50),
   GICS_Sector VARCHAR(30),
   Investment_Weight DECIMAL(5 , 4 ),
   PRIMARY KEY (INSTITUTION , GICS_Sector)
);
DROP TABLE IF EXISTS COMPONENT_OF;
CREATE TABLE COMPONENT_OF (
                Symbol VARCHAR(10)
               ,IndiceID VARCHAR(4)
               ,PRIMARY KEY (IndiceID,Symbol)
               );

DROP TABLE IF EXISTS INDICE;
CREATE TABLE INDICE (
             IndiceID VARCHAR(4)
            ,Indice VARCHAR(40)
            ,Type_Name VARCHAR(10)
            ,Constituents int(5)
```

```
            ,Weighting VARCHAR(100)
            ,PRIMARY KEY (IndiceID)
            );

DROP TABLE IF EXISTS INDICE_VALUE;
CREATE TABLE INDICE_VALUE (
                IndiceID VARCHAR(4)
                ,Date_Observed Date
                ,Open_Price DECIMAL(8,2)
                ,High DECIMAL(8,2)
                ,Low DECIMAL(8,2)
                ,Close_Price DECIMAL(8,2)
                ,PRIMARY KEY (IndiceID, Date_Observed)
                );

DROP TABLE IF EXISTS FUND;
CREATE TABLE FUND (
            Fund VARCHAR(60)
            ,Institution VARCHAR(20)
            ,Type_Name VARCHAR(20)
            ,YTD DECIMAL(4,2)
            ,Date_Observed Date
            ,PRIMARY KEY (Fund)
            );

DROP TABLE IF EXISTS CITY;
CREATE TABLE CITY (
            CityID VARCHAR(4)
            ,City VARCHAR(20)
            ,State VARCHAR(20)
            ,Country VARCHAR(10)
            ,PRIMARY KEY (CityID)
            );
```

(3) Attached is representative samples of SQL insert code

```
insert into SECURITIES values ('MMM', '0000066740', '3M Company', '0001', '0299', '20105010',
'1902');
insert into SECURITIES values ('ABT', '0000001800', 'Abbott Laboratories', '0001', '0300',
'35101010', '1888');
insert into SECURITIES values ('ABBV', '0001551152', 'AbbVie Inc.', '0001', '0300', '35202010',
'2013 (1888)');
insert into SECURITIES values ('ABMD', '0000815094', 'ABIOMED Inc', '0002', '0302', '35101010',
'1981');
insert into SECURITIES values ('ACN', '0001467373', 'Accenture plc', '0001', '0303', '45102010',
'1989');
insert into SECURITIES values ('ADBE', '0000796343', 'Adobe Systems Inc', '0002', '0010',
'45103010', '1982');
insert into SECURITIES values ('AMD', '0000002488', 'Advanced Micro Devices Inc', '0002',
'0168', '45301020', '1969');
insert into SECURITIES values ('AAP', '0001158449', 'Advance Auto Parts', '0001', '0305',
'25504050', '1932');
insert into SECURITIES values ('AES', '0000874761', 'AES Corp', '0001', '0051', '55105010', '1981');
insert into SECURITIES values ('AET', '0001122304', 'Aetna Inc', '0001', '0218', '35102030', '1810');
```

```
insert into SECURITIES values ('AMG', '0001004434', 'Affiliated Managers Group Inc', '0001',
'0312', '40203010', '1993');
insert into SECURITIES values ('AFL', '0000004977', 'AFLAC Inc', '0001', '0015', '40301020',
'1955');
insert into SECURITIES values ('A', '0001090872', 'Agilent Technologies Inc', '0001', '0219',
'35101010', '1999');
insert into SECURITIES values ('ALK', '0000766421', 'Alaska Air Group Inc', '0001', '0020',
'20302010', '1985');
insert into SECURITIES values ('ARE', '0001035443', 'Alexandria Real Estate Equities  Inc', '0001',
'0163', '60101040', '1994');
insert into SECURITIES values ('ALXN', '0000899866', 'Alexion Pharmaceuticals', '0002', '0321',
'35201010', '1992');
insert into SECURITIES values ('ALGN', '0001097149', 'Align Technology', '0002', '0010',
'35101020', '1997');
insert into SECURITIES values ('ALLE', '0001579241', 'Allegion', '0001', '0303', '20102010', '1908');
insert into SECURITIES values ('AGN', '0001578845', 'Allergan, Plc', '0001', '0303', '35202010',
'1983');
insert into SECURITIES values ('ADS', '0001101215', 'Alliance Data Systems', '0001', '0070',
'45102020', '1996');

insert into SECTOR values ('10101010', 'Energy', 'Energy Equipment & Services', 'Oil & Gas
Drilling');
insert into SECTOR values ('10101020', 'Energy', 'Energy Equipment & Services', 'Oil & Gas
Equipment & Services');
insert into SECTOR values ('10102010', 'Energy', 'Oil, Gas & Consumable Fuels', 'Integrated Oil &
Gas');
insert into SECTOR values ('10102020', 'Energy', 'Oil, Gas & Consumable Fuels', 'Oil & Gas
Exploration & Production');
insert into SECTOR values ('10102030', 'Energy', 'Oil, Gas & Consumable Fuels', 'Oil & Gas
Refining & Marketing opportunities');
insert into SECTOR values ('10102040', 'Energy', 'Oil, Gas & Consumable Fuels', 'Oil & Gas Storage
& Transportation');
insert into SECTOR values ('10102050', 'Energy', 'Oil, Gas & Consumable Fuels', 'Coal &
Consumable Fuels');
insert into SECTOR values ('15102010', 'Material', 'Chemicals', 'Construction Materials');
insert into SECTOR values ('15103010', 'Material', 'Construction Materials', 'Metal & Glass
Containers');
insert into SECTOR values ('15103020', 'Material', 'Construction Materials', 'Paper Packaging');
insert into SECTOR values ('15104010', 'Material', 'Metals & Mining', 'Aluminum');
insert into SECTOR values ('15104020', 'Material', 'Metals & Mining', 'Diversified Metals &
Mining');
insert into SECTOR values ('15104025', 'Material', 'Metals & Mining', 'Copper');
insert into SECTOR values ('15104030', 'Material', 'Metals & Mining', 'Gold');
insert into SECTOR values ('15104040', 'Material', 'Metals & Mining', 'Precious Metals & Minerals');

insert into PRICE values ('WLTW', '2016-12-30', 122.589996, 123.559998, 121.389999, 122.279999,
'466400');
insert into PRICE values ('A', '2016-12-30', 45.759998, 45.82, 45.380001, 45.560001, '1216100');
insert into PRICE values ('AAL', '2016-12-30', 47.419998, 47.66, 46.470001, 46.689999,
'4495000');
insert into PRICE values ('AAP', '2016-12-30', 171.320007, 172.0, 168.600006, 169.119995,
'489300');
insert into PRICE values ('AAPL', '2016-12-30', 116.650002, 117.199997, 115.43, 115.82,
'30586300');
insert into PRICE values ('ABBV', '2016-12-30', 62.73, 62.93, 62.41, 62.619999, '5999200');
```

```
insert into PRICE values ('ABC', '2016-12-30', 79.349998, 79.489998, 77.959999, 78.190002,
'1387500');
insert into PRICE values ('ABT', '2016-12-30', 38.330002, 38.869999, 38.25, 38.41, '10445600');
insert into PRICE values ('ACN', '2016-12-30', 117.559998, 117.949997, 116.589996, 117.129997,
'1739500');
insert into PRICE values ('ADBE', '2016-12-30', 104.07, 104.220001, 102.470001, 102.949997,
'2079100');
insert into PRICE values ('ADI', '2016-12-30', 73.800003, 74.019997, 72.389999, 72.620003,
'1758500');
insert into PRICE values ('ADM', '2016-12-30', 45.419998, 45.689999, 45.029999, 45.650002,
'2509000');
insert into PRICE values ('ADP', '2016-12-30', 103.379997, 103.440002, 102.0, 102.779999,
'2008400');
insert into PRICE values ('ADS', '2016-12-30', 229.830002, 230.460007, 227.389999, 228.5,
'338700');
insert into PRICE values ('ADSK', '2016-12-30', 75.400002, 75.419998, 73.599998, 74.010002,
'1529600');
insert into PRICE values ('AEE', '2016-12-30', 52.490002, 52.880001, 52.25, 52.459999, '950700');
insert into PRICE values ('AEP', '2016-12-30', 63.32, 63.419998, 62.720001, 62.959999, '1677000');
insert into PRICE values ('AES', '2016-12-30', 11.73, 11.79, 11.59, 11.62, '3245200');
insert into PRICE values ('AET', '2016-12-30', 124.559998, 124.739998, 123.169998, 124.010002,
'1588100');
insert into PRICE values ('AFL', '2016-12-30', 70.019997, 70.139999, 69.43, 69.599998, '1193900');
insert into PRICE values ('AGN', '2016-12-30', 207.600006, 212.449997, 207.509995, 210.009995,
'3968000');
insert into PRICE values ('AIG', '2016-12-30', 65.400002, 65.720001, 65.120003, 65.309998,
'4218700');
insert into PRICE values ('AIZ', '2016-12-30', 93.129997, 93.400002, 92.470001, 92.860001,
'287800');
insert into PRICE values ('AJG', '2016-12-30', 51.959999, 52.16, 51.630001, 51.959999, '1219500');
insert into PRICE values ('AKAM', '2016-12-30', 67.25, 67.550003, 66.349998, 66.68, '966400');
insert into PRICE values ('ALB', '2016-12-30', 87.339996, 87.739998, 85.599998, 86.080002,
'449600');
insert into PRICE values ('ALK', '2016-12-30', 88.739998, 89.139999, 88.019997, 88.730003,
'931000');
insert into PRICE values ('ALL', '2016-12-30', 74.540001, 74.559998, 73.93, 74.120003, '1705500');
insert into PRICE values ('ALLE', '2016-12-30', 65.040001, 65.199997, 63.709999, 64.0, '501800');
insert into PRICE values ('ALXN', '2016-12-30', 123.620003, 124.230003, 121.809998, 122.349998,
'1427900');
insert into PRICE values ('AMAT', '2016-12-30', 32.860001, 32.91, 32.18, 32.27, '5848300');
insert into PRICE values ('AME', '2016-12-30', 48.709999, 48.830002, 48.209999, 48.599998,
'1045600');
insert into PRICE values ('AMG', '2016-12-30', 144.940002, 146.210007, 144.369995, 145.300003,
'669800');
insert into PRICE values ('AMGN', '2016-12-30', 147.630005, 148.75, 145.619995, 146.210007,
'3193100');
insert into PRICE values ('AMP', '2016-12-30', 110.620003, 111.599998, 110.230003, 110.940002,
'798100');
insert into PRICE values ('AMT', '2016-12-30', 106.459999, 106.690002, 105.410004, 105.68,
'1628100');
insert into PRICE values ('AMZN', '2016-12-30', 766.469971, 767.400024, 748.280029,
749.869995, '4125300');
insert into PRICE values ('AN', '2016-12-30', 48.740002, 48.950001, 48.299999, 48.650002,
'524500');
```

```
insert into PRICE values ('ANTM', '2016-12-30', 144.880005, 145.5, 143.110001, 143.770004,
'1085700');
insert into PRICE values ('AON', '2016-12-30', 111.769997, 111.959999, 111.059998, 111.529999,
'634300');
insert into PRICE values ('APA', '2016-12-30', 63.869999, 64.199997, 63.27, 63.470001,
'2238300');
insert into PRICE values ('APC', '2016-12-30', 70.07, 70.790001, 69.510002, 69.730003,
'2441500');
insert into PRICE values ('APD', '2016-12-30', 144.529999, 144.820007, 143.229996, 143.820007,
'725900');
insert into PRICE values ('APH', '2016-12-30', 67.879997, 67.879997, 66.970001, 67.199997,
'876700');
insert into PRICE values ('ARNC', '2016-12-30', 18.780001, 19.08, 18.469999, 18.540001,
'5054000');
insert into PRICE values ('ATVI', '2016-12-30', 36.529999, 36.560001, 35.950001, 36.110001,
'5090500');
insert into PRICE values ('AVB', '2016-12-30', 175.259995, 177.770004, 174.889999, 177.149994,
'756300');
insert into PRICE values ('AVGO', '2016-12-30', 180.369995, 180.899994, 176.020004,
176.770004, '2542100');
insert into PRICE values ('AVY', '2016-12-30', 70.940002, 71.230003, 70.019997, 70.220001,
'585400');
insert into PRICE values ('AWK', '2016-12-30', 72.660004, 72.889999, 72.029999, 72.360001,
'775400');
insert into PRICE values ('AXP', '2016-12-30', 73.959999, 74.339996, 73.839996, 74.080002,
'3378800');
insert into PRICE values ('AYI', '2016-12-30', 233.690002, 234.119995, 229.940002, 230.860001,
'388000');
```

(4) Attached is SQL query code and output for the 16 questions

#1

```
SELECT m.institution
FROM MAIN_HOLDER AS m LEFT JOIN COMPONENT_OF AS c ON m.Symbol = c.Symbol
WHERE IndiceID = 'SPX'
GROUP BY m.institution
HAVING COUNT(DISTINCT m.symbol) >5;
```

```
+----------------------------------+
| institution                      |
+----------------------------------+
| BERKSHIRE HATHAWAY INC           |
| BLACKROCK INC.                   |
| FMR LLC                          |
| JPMORGAN CHASE & CO              |
| PRICE T ROWE ASSOCIATES INC /MD/ |
| STATE STREET CORP                |
| VANGUARD GROUP INC               |
| WELLINGTON MANAGEMENT GROUP LLP  |
+----------------------------------+
```

#2

```
SELECT r.institution
FROM (SELECT
        i.institution
      , i.Investment_Weight
      , rank() over (ORDER BY investment_weight DESC) AS weight_rank
        FROM INSTITUTION_SECTOR AS i
        WHERE i.GICS_Sector = 'Healthcare') AS r
        WHERE r.weight_rank =1;
```

```
+---------------+
| Institution   |
+---------------+
| BLACKROCK INC |
+---------------+
```

#3

```
SELECT COUNT(DISTINCT a.symbol)
FROM SECURITIES AS a LEFT JOIN SECTOR as b ON a.sectorID = b.sectorID
WHERE CAST(a.founded AS SIGNED) < 2000
 AND b.GICS_sector = 'Information Technology';
```

```
    +-------------------------+
    | COUNT(DISTINCT a.symbol) |
    +-------------------------+
    |                      60 |
    +-------------------------+
```

#4

```
SELECT DISTINCT s.symbol
FROM SECURITIES AS s INNER JOIN REPORT AS r ON s.CIK = r.CIK
WHERE r.filling_date = '2018-11-1';
```

```
+--------+
| symbol |
+--------+
| ABMD   |
| AAPL   |
+--------+
```

#5

```
SELECT s.symbol
FROM SECURITIES AS s INNER JOIN EXCHANGES as e ON s.exchangeid = e.exchangeid
        INNER JOIN CITY AS c ON c.cityid = s.cityid
WHERE e.exchange_name = 'New York Stock EXCHANGES'
        AND c.country = 'US'
        AND c.state = 'New York'
        AND c.city = 'New York';
```

```
+--------+
| symbol |
+--------+
| AXP    |
| AIG    |
| ARNC   |
| AIZ    |
| BK     |
| BLK    |
| BMY    |
| CBS    |
| C      |
| CL     |
| ED     |
| COTY   |
| EL     |
| FL     |
| GS     |
| HES    |
| IPG    |
| JEF    |
| JPM    |
| LLL    |
| L      |
| MMC    |
| MET    |
| KORS   |
| MCO    |
| MS     |
| MSCI   |
| NLSN   |
| OMC    |
| PFE    |
| PM     |
| RL     |
| PVH    |
| SLG    |
| SPGI   |
| TPR    |
| TIF    |
| TRV    |
| VZ     |
| VNO    |
+--------+
```

6

```
SELECT s.Security_Name
        ,(p.Close_Price-p.Open_Price)/p.Open_Price AS dayreturn
FROM PRICE AS p
INNER JOIN SECURITIES AS s ON s.Symbol=p.Symbol
WHERE p.Date_Observed='2016-12-29'
ORDER BY (p.Close_Price-p.Open_Price)/p.Open_Price DESC
LIMIT 1;
```

```
+--------------------+--------------+
| Security_Name      | dayreturn    |
+--------------------+--------------+
| Nvidia Corporation | 0.0622497414 |
+--------------------+--------------+
```

7

```
SELECT s.Security_Name
FROM (
        SELECT c.Symbol
                ,COUNT(c.IndiceID) AS countindice
         FROM COMPONENT_OF AS c
        GROUP BY c.Symbol
        ) AS a
```

```
INNER JOIN SECURITIES AS s ON s.Symbol=a.Symbol
WHERE a.countindice >= 2;
```

```
+--------------------------------+
| Security_Name                  |
+--------------------------------+
| Apple Inc.                     |
| American Express Co            |
| Boeing Company                 |
| Caterpillar Inc.               |
| Cisco Systems                  |
| Chevron Corp.                  |
| The Walt Disney Company        |
| Goldman Sachs Group            |
| Home Depot                     |
| International Business Machines |
| Intel Corp.                    |
| Johnson & Johnson              |
| JPMorgan Chase & Co.           |
| Coca-Cola Company  (The)       |
| McDonalds Corp.                |
| 3M Company                     |
| Merck & Co.                    |
| Microsoft Corp.                |
| Nike                           |
| Pfizer Inc.                    |
| Procter & Gamble               |
| The Travelers Companies Inc.   |
| United Health Group Inc.       |
| United Technologies            |
| Visa Inc.                      |
| Verizon Communications         |
| Walgreens Boots Alliance       |
| Walmart                        |
| Exxon Mobil Corp.              |
+--------------------------------+
```

```
# 8
SELECT s.Security_Name
FROM FUNDAMENTAL AS f
INNER JOIN SECURITIES AS s ON s.Symbol=f.Symbol
WHERE f.Liabilities >= f.Assets
AND YEAR(f.Date_Observed)=2016;
```

```
+-----------------+
| Security_Name   |
+-----------------+
| AutoZone Inc    |
| HP Inc.         |
| L Brands Inc.   |
| TransDigm Group |
+-----------------+
```

```
# 9
SELECT i.Institution
        ,i.GICS_Sector
        ,i.Investment_Weight
FROM (
        SELECT Institution
                ,MAX(Investment_Weight) AS maxweight
        FROM INSTITUTION_SECTOR
        GROUP BY Institution
                        ) AS a
INNER JOIN INSTITUTION_SECTOR AS i
ON i.Institution=a.Institution
AND a.maxweight=i.Investment_Weight;
```

```
+----------------------------------+-------------+--------------------+
| Institution                      | GICS_Sector | Investment_Weight  |
+----------------------------------+-------------+--------------------+
| BLACKROCK INC                    | Technology  |             0.2223 |
| GOLDMAN SACHS GROUP INC          | Financials  |             0.3184 |
| JPMORGAN CHASE & CO              | Financials  |             0.3338 |
| morgan-stanley                   | Financials  |             0.3725 |
| ubs-asset-management-americas-inc | Technology  |             0.2745 |
| WELLS FARGO & COMPANY            | Financials  |             0.4084 |
+----------------------------------+-------------+--------------------+
```

10

```
SELECT DISTINCT(r.FileNumber)
FROM REPORT AS r
INNER JOIN SECURITIES AS s ON s.CIK=r.CIK
WHERE s.Security_Name='Apple Inc.'
AND r.Filling_Date>'2018-11-01';
```

```
+------------+
| FileNumber |
+------------+
| 333-228159 |
| 001-36743  |
+------------+
2 rows in set (0.00 sec)
```

11

```
SELECT S.Symbol AS Symbol, COUNT(*) AS Num_Of_8K_Files
FROM SECURITIES AS S, REPORT AS R
WHERE S.CIK = R.CIK AND S.Symbol = 'AAPL' AND R.Filling = '8-K' AND YEAR(R.filling_date) = 2018;
```

```
+--------+-----------------+
| Symbol | Num_Of_8K_Files |
+--------+-----------------+
| AAPL   |               6 |
+--------+-----------------+
```

12

```
SELECT E.Exchange_Name AS
Exchange_Name, COUNT(*) AS Num_Of_8K_Files
FROM SECURITIES AS S, REPORT AS R, EXCHANGES AS E
WHERE S.CIK = R.CIK
        AND E.Exchange_Name= 'New York Stock EXCHANGES'
        AND E.ExchangeID = S.ExchangeID
        AND R.Filling = '8-K'
        AND YEAR(R.filling_date) = 2018;
```

```
+-------------------------+-----------------+
| Exchange_Name           | Num_Of_8K_Files |
+-------------------------+-----------------+
| New York Stock EXCHANGES |              74 |
+-------------------------+-----------------+
```

13

```
SELECT *
FROM MAIN_HOLDER AS M
WHERE M.Symbol = 'TWTR'
ORDER BY M.Shares DESC
LIMIT 1;
```

```
+--------+--------------------+----------+
| Symbol | Institution        | Shares   |
+--------+--------------------+----------+
| TWTR   | VANGUARD GROUP INC | 70255589 |
+--------+--------------------+----------+
```

14

```
SELECT *
FROM MAIN_HOLDER AS M
```

```
WHERE M.Institution LIKE  (SELECT CONCAT(F.Institution, '%')
                           FROM FUND AS F
                           WHERE F.Date_Observed = '2018-11-15'
                           ORDER BY F.YTD DESC
                           LIMIT 1);
```

Symbol	Institution	Shares
AAPL	BLACKROCK INC.	295793280
TWTR	BLACKROCK INC.	47379873
ABT	BLACKROCK INC.	117203489
MSFT	BLACKROCK INC.	503748643
AMZN	BLACKROCK INC.	25546629
EBAY	BLACKROCK INC.	57424891
DIS	BLACKROCK INC.	87254234
V	BLACKROCK INC.	122485390
GPS	BLACKROCK INC.	15672359
HPQ	BLACKROCK INC.	105799419
MS	BLACKROCK INC.	108632289
NLSN	BLACKROCK INC.	26816620
JWN	BLACKROCK INC.	9865556
PYPL	BLACKROCK INC.	73378920
RL	BLACKROCK INC.	3937063
SBUX	BLACKROCK INC.	90368884
TROW	BLACKROCK INC.	17444147
TGT	BLACKROCK INC.	41887592
UA	BLACKROCK INC.	11571835
BABA	BLACKROCK INC.	63377267
WFC	BLACKROCK INC.	290798071
AAL	BLACKROCK INC.	23112849
GS	BLACKROCK INC.	22576744
HSY	BLACKROCK INC.	12694041
MTB	BLACKROCK INC.	9986587
MMC	BLACKROCK INC.	39673471
MCD	BLACKROCK INC.	51902035
MSCI	BLACKROCK INC.	6378690
NFLX	BLACKROCK INC.	27397999
TIF	BLACKROCK INC.	8900821
MCK	BLACKROCK INC.	15414724
BK	BLACKROCK INC.	54163051
KO	BLACKROCK INC.	246593679
C	BLACKROCK INC.	177564297
CVS	BLACKROCK INC.	95750621
FRT	BLACKROCK INC.	7927797
FE	BLACKROCK INC.	58738683
GE	BLACKROCK INC.	536996868
GPC	BLACKROCK INC.	14236823

15

```
SELECT *
FROM SECURITIES AS S, MAIN_HOLDER AS M
WHERE S.Symbol = M.Symbol
        AND S.Security_Name = (SELECT S.Security_Name
                               FROM PRICE AS P, SECURITIES AS S
                               WHERE P.Date_Observed='2016-12-29' AND S.Symbol=P.Symbol
                               ORDER BY (P.Close_Price-P.Open_Price)/P.Open_Price DESC
                               LIMIT 1)
ORDER BY M.Shares DESC
LIMIT 1;
```

Empty set (0.14 sec)

16

```
SELECT S2.Symbol, F.Assets, F.Liabilities, F.Revenue
FROM SECURITIES AS S1, SECURITIES AS S2, FUNDAMENTAL AS F
WHERE S1.Security_Name = 'PayPal' AND S1.CityID = S2.CityID AND S1.Symbol <> S2.Symbol
        AND F.Symbol = S2.Symbol;
```

Symbol	Assets	Liabilities	Revenue
ADBE	10380298000.0	3655664000.0	4055240000.0
ADBE	10785829000.0	4009924000.0	4147065000.0
ADBE	11726472000.0	4724892000.0	4795511000.0
CSCO	101000000000.0	42071000000.0	48607000000.0
CSCO	105000000000.0	48416000000.0	47142000000.0
CSCO	113000000000.0	53675000000.0	49161000000.0
CSCO	122000000000.0	58066000000.0	49247000000.0
EBAY	41488000000.0	17841000000.0	8257000000.0
EBAY	45132000000.0	25226000000.0	8790000000.0
EBAY	17755000000.0	11179000000.0	8592000000.0
XLNX	4729451000.0	1766155000.0	2168652000.0
XLNX	5037349000.0	2284667000.0	2382531000.0
XLNX	4898065000.0	2286471000.0	2377344000.0
XLNX	4823154000.0	2233261000.0	2213881000.0