Financial Market Data Pipeline

<u>Problem Setting</u>

In recent times, especially with the covid pandemic the global financial situation has seen a lot of difficulties. Companies and organizations have had to improvise their profit-making strategies. Thus, it has become more important now than ever for investors and companies to keep a constant check on their financial health. One way of doing this is keeping track of stock prices. This must be done daily, as stock prices keep fluctuating frequently. But this involves a large amount of data and might not be handled efficiently by traditional on-premise Datawarehouses, this makes cloud the best solution to store and analyze data.

<u>Problem Definition</u>

The large amount of data created by the stock market needs to be assessed. This must be done in a computationally efficient manner. To handle this data, we must look for scalable and robust solutions. Our proposal involves using cloud-based warehouses to store, manipulate and analyze the stock market data, which is at a daily level. We intend to analyze trends and interpret them in our project.

<u>Project Pipeline:</u>

- Data Collection: Data is collected from the 2 different data sources mentioned above. AWS Lambda is used to collect data and store it in respective formats.
- Data Storing: The data is stored in AWS S3 bucket. AWS lambda is used to upload data to the bucket.
- Data Transformation: Transformations are performed to the data using AWS Glue jobs.
- Data Visualization: The data is analyzed through AWS Athena and then visualized to a dashboard by connecting Athena to Tableau

Data Sources

The data sources are as follows:

1. Yahoo Finance Data: The data is obtained from yahoo finance API through python https://pypi.org/project/yfinance/

2. DJI Index data: The DJI index data is web scraped from the website https://www.slickcharts.com/dowjones

Data Description

The yahoo finance data gives the historical financial data for various stocks and indexes. Using the yahoo finance API in python, we are getting the data for DJI index, Google stock and Amazon stock from 01/01/2020 to 12/31/2022.

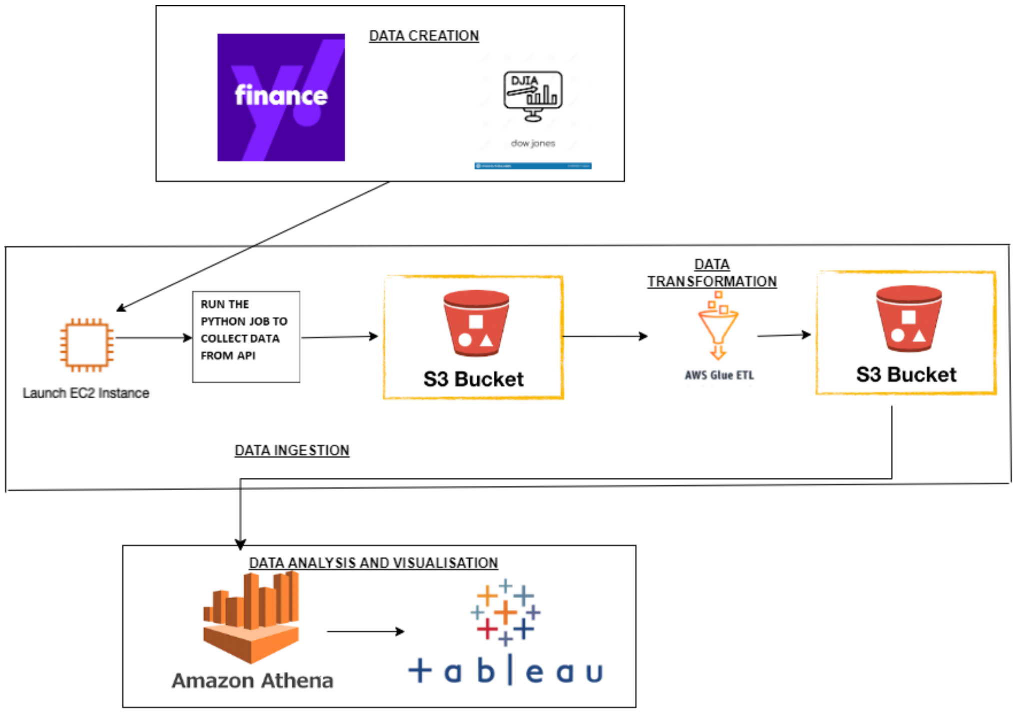
The DJI index data contains financial data about the 30 companies that are a part of the DJI Index. Dow Jones Industrial Average is a stock market index made up of 30 prominent companies listed on American stock exchanges.

The data obtained from both data sources is concatenated using company symbol.

Analysis is done on 6 dimensions of stock price for 32 companies over the timeframe 01/01/2020 - 12/31/2022:

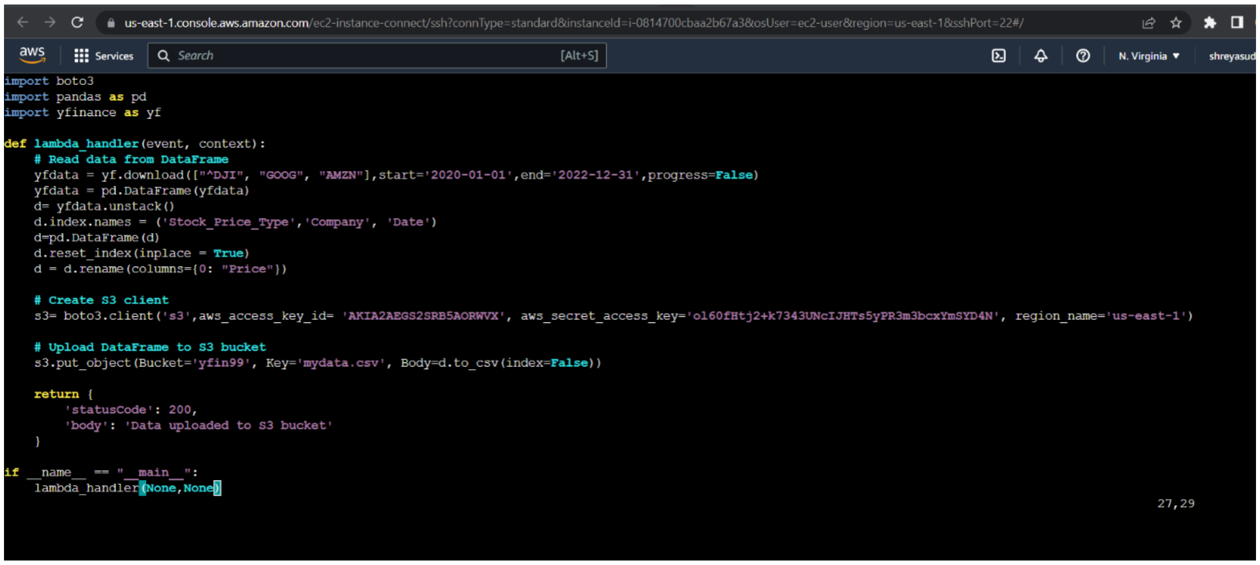
- Open: A stock's opening price for the start of the day
- Close: A stock's final price at the end of the day
- Adj Close: The closing price after considering all relevant splits and dividend payouts
- High: A stock's highest trading price during the day
- Low: A stock's lowest trading price during the day
- Volume: The number of shares being traded for a particular stock

<u>Pipeline Architecture:</u>

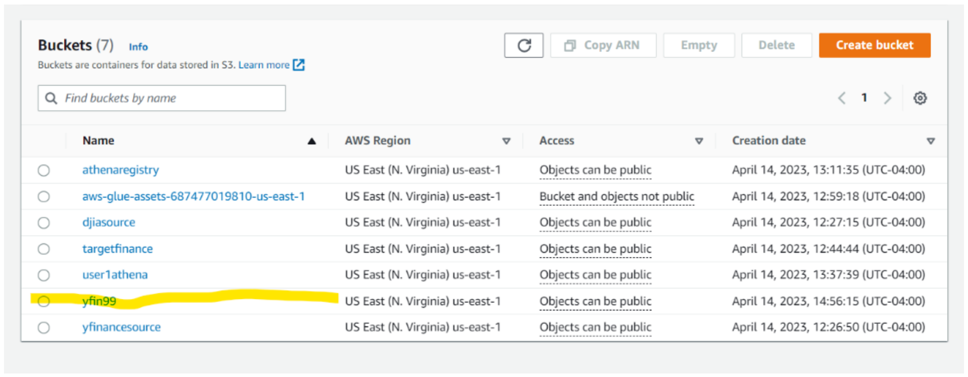


<u>PIPELINE PROCESS:</u>

- **Data Ingestion :**
  - In this step we use the EC2 instance to get data from the Yahoo finance API and the DOW Jones API. This data is then stored in the S3 bucket. We make use of the BOTO3 library to write data into S3. For the purpose of this project, we use two data sources in our S3, csv and json.



```
import boto3
import pandas as pd
import yfinance as yf

def lambda_handler(event, context):
    # Read data from DataFrame
    yfdata = yf.download(["^DJI", "GOOG", "AMZN"],start='2020-01-01',end='2022-12-31',progress=False)
    yfdata = pd.DataFrame(yfdata)
    d= yfdata.unstack()
    d.index.names = ('Stock_Price_Type','Company', 'Date')
    d=pd.DataFrame(d)
    d.reset_index(inplace = True)
    d = d.rename(columns={0: "Price"})

    # Create S3 client
    s3= boto3.client('s3',aws_access_key_id= 'AKIA2AEGS2SRB5AORWVX', aws_secret_access_key='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', region_name='us-east-1')

    # Upload DataFrame to S3 bucket
    s3.put_object(Bucket='yfin99', Key='mydata.csv', Body=d.to_csv(index=False))

    return {
        'statusCode': 200,
        'body': 'Data uploaded to S3 bucket'
    }

if __name__ == "__main__":
    lambda_handler(None,None)
                                                                                             27,29
```



Buckets (7) Info

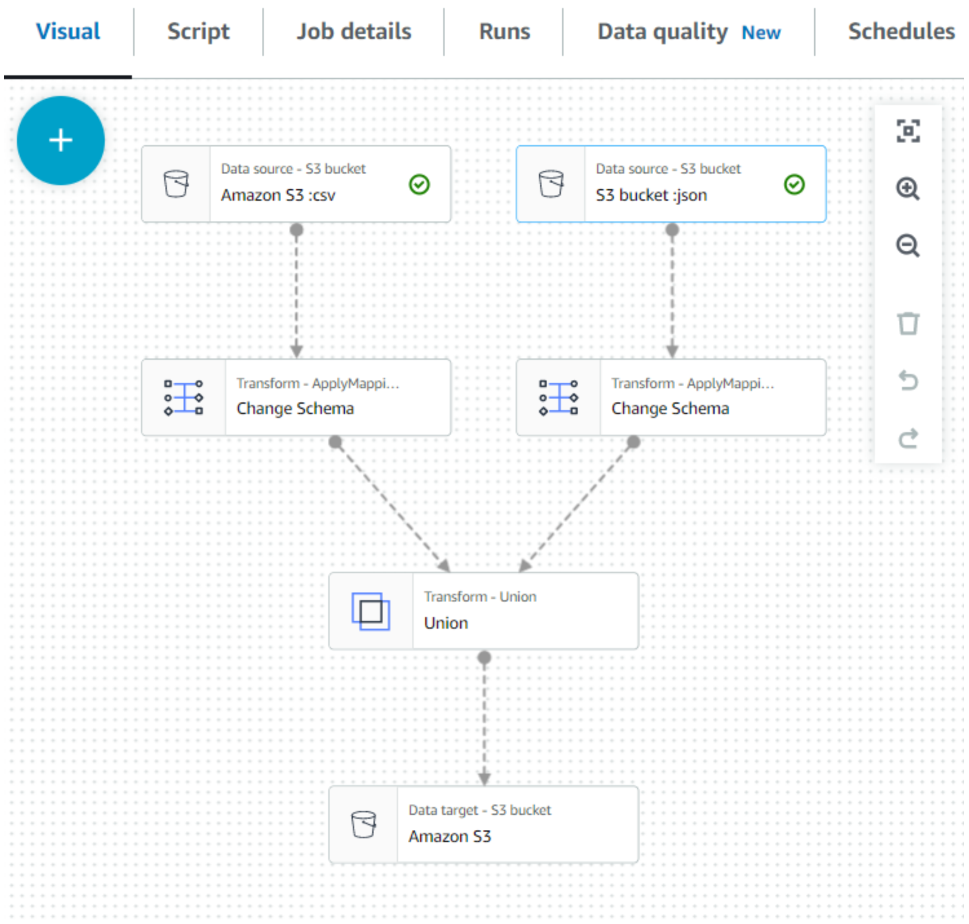
Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	athenaregistry	US East (N. Virginia) us-east-1	Objects can be public	April 14, 2023, 13:11:35 (UTC-04:00)
○	aws-glue-assets-687477019810-us-east-1	US East (N. Virginia) us-east-1	Bucket and objects not public	April 14, 2023, 12:59:18 (UTC-04:00)
○	djiasource	US East (N. Virginia) us-east-1	Objects can be public	April 14, 2023, 12:27:15 (UTC-04:00)
○	targetfinance	US East (N. Virginia) us-east-1	Objects can be public	April 14, 2023, 12:44:44 (UTC-04:00)
○	user1athena	US East (N. Virginia) us-east-1	Objects can be public	April 14, 2023, 13:37:39 (UTC-04:00)
○	yfin99	US East (N. Virginia) us-east-1	Objects can be public	April 14, 2023, 14:56:15 (UTC-04:00)
○	yfinancesource	US East (N. Virginia) us-east-1	Objects can be public	April 14, 2023, 12:26:50 (UTC-04:00)

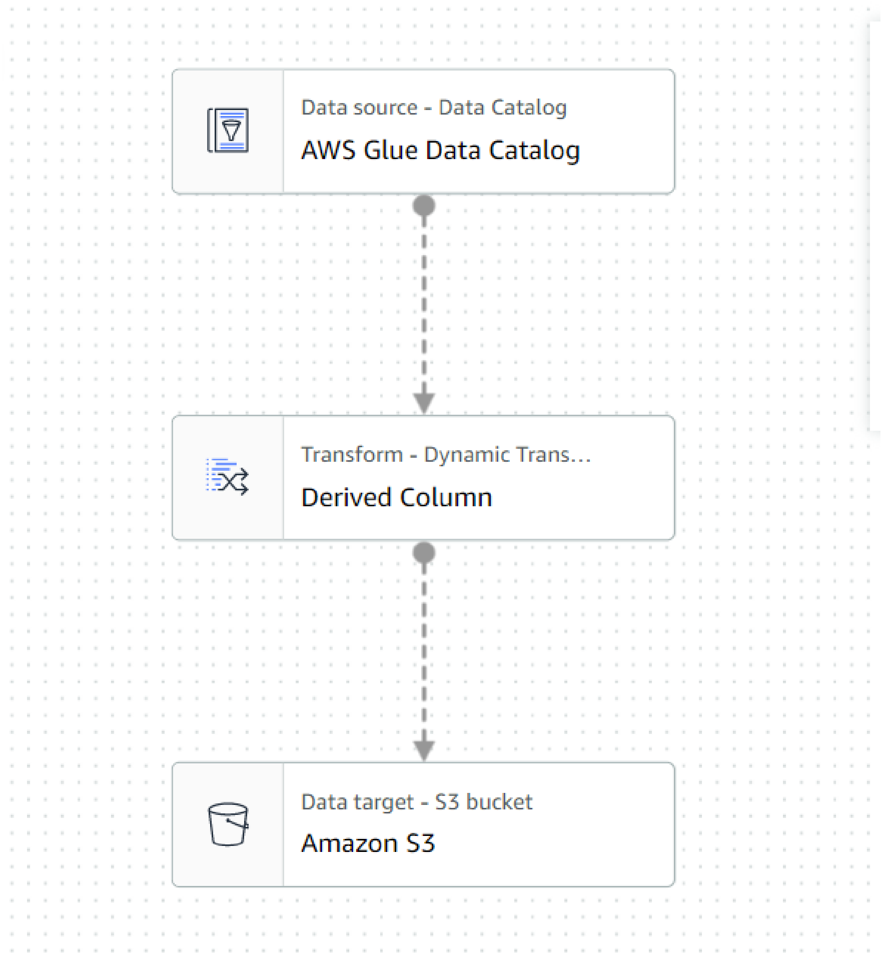
Data Transformation:

We perform two glue jobs:

- Union Table:
    - The data from S3 buckets are taken as the data sources for the glue . Mapping is applied to both the data sources from string to their respective data types. After applying ping, the transformed data are combined using union transform. The union data is then stored in the target S3 bucket, and a data catalog table named union_table is made for Athena query usage.

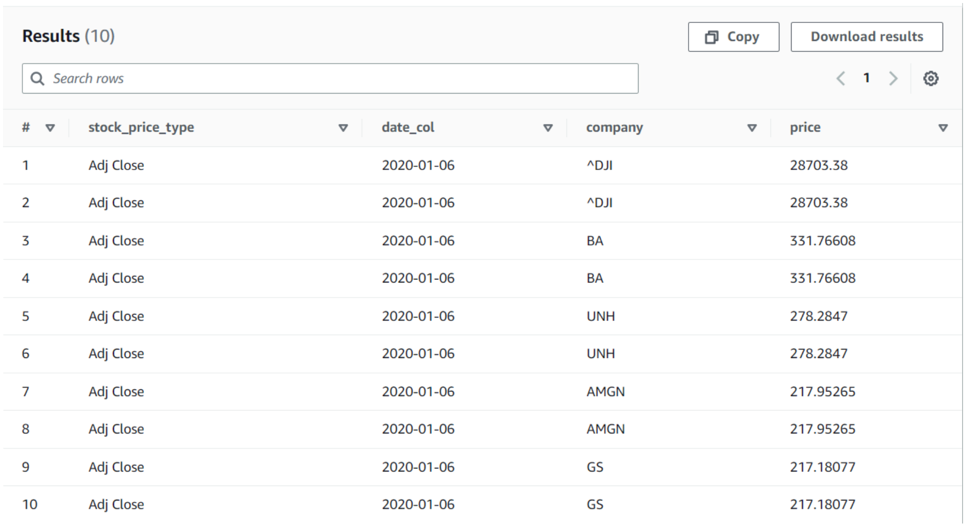


- Rank Job:
    - A new column is added to the union table from the data catalog. The derived column is a rank column representing a dense rank for the stocks and their price. The transformed is then stored in the target S3 bucket and a data catalogue table named rank_job is made for Athena query usage.


Data source - Data Catalog
AWS Glue Data Catalog
Transform - Dynamic Trans...
Derived Column
Data target - S3 bucket
Amazon S3

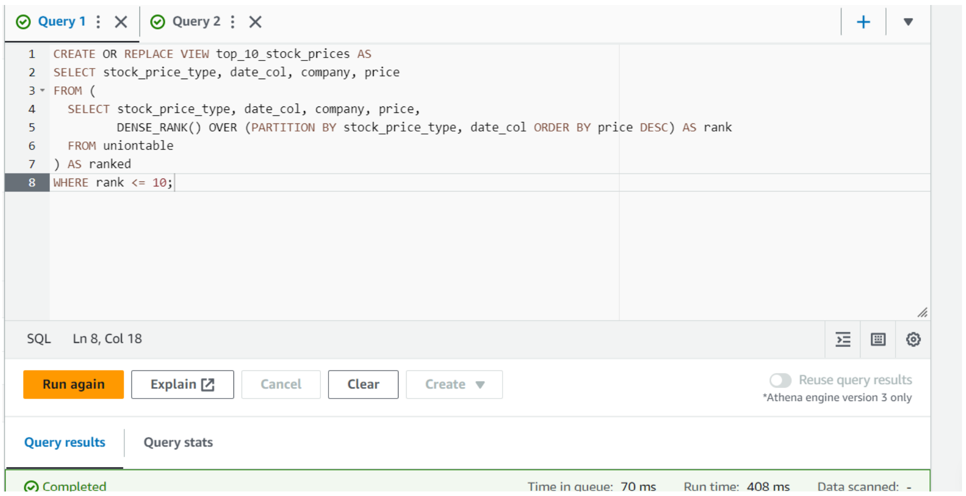
Data Analyzing:

We use Athena to analyze the data in our S3 bucket and try to understand the trends in the same.The financedata table has data, which is day on day, it can be be analysed by type and day on day trend.

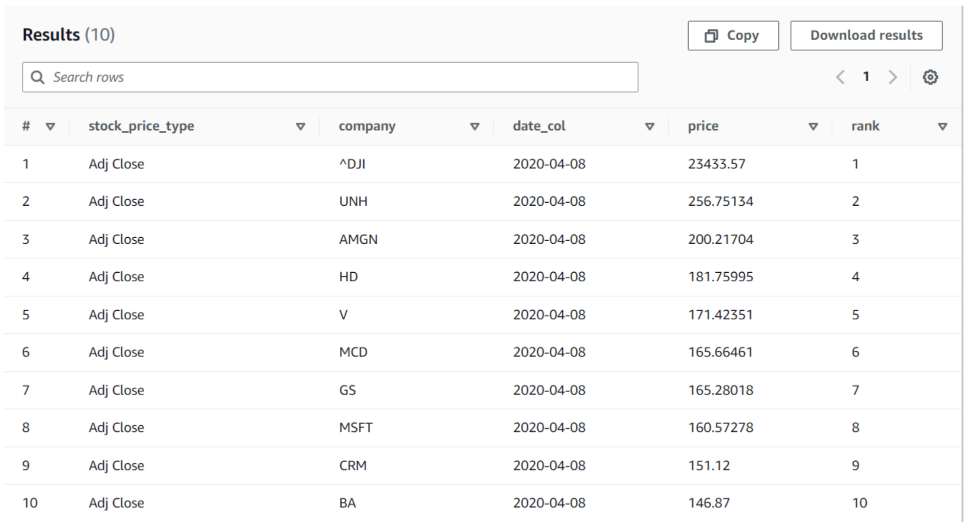

Results (10)

#	stock_price_type	date_col	company	price
1	Adj Close	2020-01-06	^DJI	28703.38
2	Adj Close	2020-01-06	^DJI	28703.38
3	Adj Close	2020-01-06	BA	331.76608
4	Adj Close	2020-01-06	BA	331.76608
5	Adj Close	2020-01-06	UNH	278.2847
6	Adj Close	2020-01-06	UNH	278.2847
7	Adj Close	2020-01-06	AMGN	217.95265
8	Adj Close	2020-01-06	AMGN	217.95265
9	Adj Close	2020-01-06	GS	217.18077
10	Adj Close	2020-01-06	GS	217.18077

- View to analyze top performing companies in each stock type:


```
CREATE OR REPLACE VIEW top_10_stock_prices AS
SELECT stock_price_type, date_col, company, price
FROM (
  SELECT stock_price_type, date_col, company, price,
         DENSE_RANK() OVER (PARTITION BY stock_price_type, date_col ORDER BY price DESC) AS rank
  FROM uniontable
) AS ranked
WHERE rank <= 10;
```

Completed — Time in queue: 70 ms — Run time: 408 ms — Data scanned: -



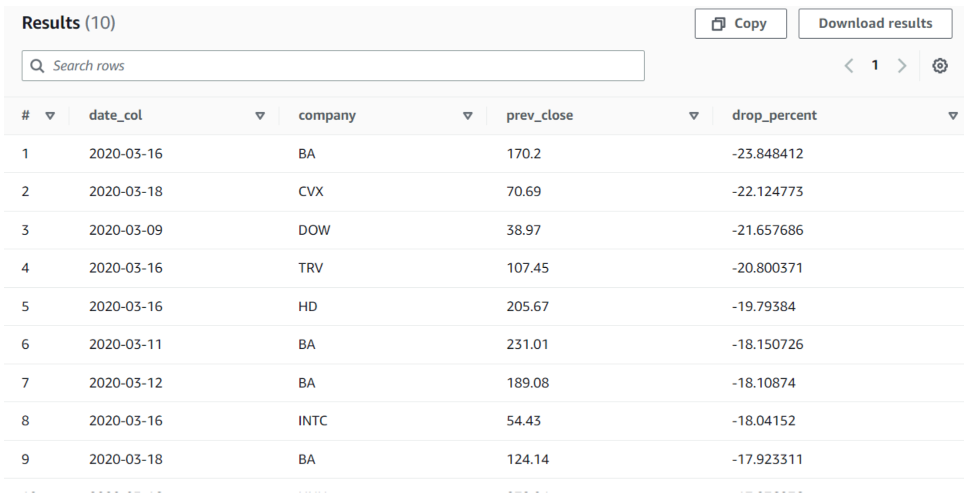
Results (10)

Copy | Download results

Search rows

#	stock_price_type	company	date_col	price	rank
1	Adj Close	^DJI	2020-04-08	23433.57	1
2	Adj Close	UNH	2020-04-08	256.75134	2
3	Adj Close	AMGN	2020-04-08	200.21704	3
4	Adj Close	HD	2020-04-08	181.75995	4
5	Adj Close	V	2020-04-08	171.42351	5
6	Adj Close	MCD	2020-04-08	165.66461	6
7	Adj Close	GS	2020-04-08	165.28018	7
8	Adj Close	MSFT	2020-04-08	160.57278	8
9	Adj Close	CRM	2020-04-08	151.12	9
10	Adj Close	BA	2020-04-08	146.87	10

- View for companies whose stock price fell the most during the day:



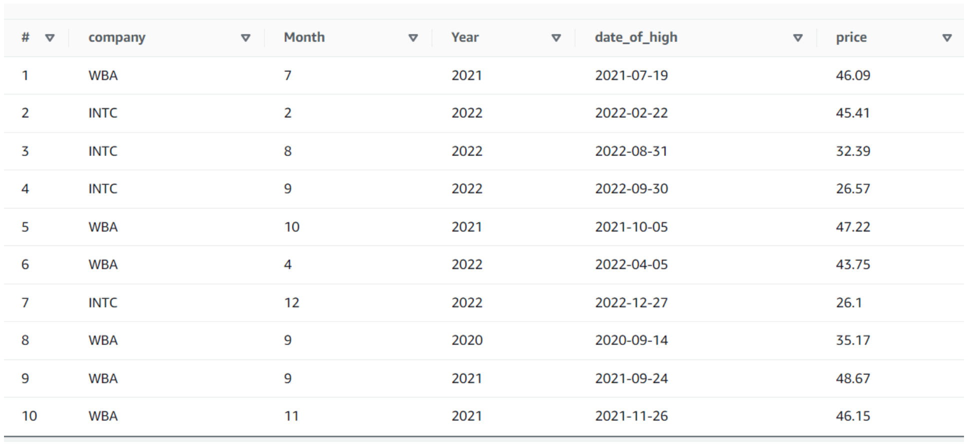
Results (10)

Copy | Download results

Search rows

#	date_col	company	prev_close	drop_percent
1	2020-03-16	BA	170.2	-23.848412
2	2020-03-18	CVX	70.69	-22.124773
3	2020-03-09	DOW	38.97	-21.657686
4	2020-03-16	TRV	107.45	-20.800371
5	2020-03-16	HD	205.67	-19.79384
6	2020-03-11	BA	231.01	-18.150726
7	2020-03-12	BA	189.08	-18.10874
8	2020-03-16	INTC	54.43	-18.04152
9	2020-03-18	BA	124.14	-17.923311

- View for highest stock price of the company in the given month-year period



#	company	Month	Year	date_of_high	price
1	WBA	7	2021	2021-07-19	46.09
2	INTC	2	2022	2022-02-22	45.41
3	INTC	8	2022	2022-08-31	32.39
4	INTC	9	2022	2022-09-30	26.57
5	WBA	10	2021	2021-10-05	47.22
6	WBA	4	2022	2022-04-05	43.75
7	INTC	12	2022	2022-12-27	26.1
8	WBA	9	2020	2020-09-14	35.17
9	WBA	9	2021	2021-09-24	48.67
10	WBA	11	2021	2021-11-26	46.15