

# Ringer – Virtual stock app

ECE-GY-9953-BK30

Group C :

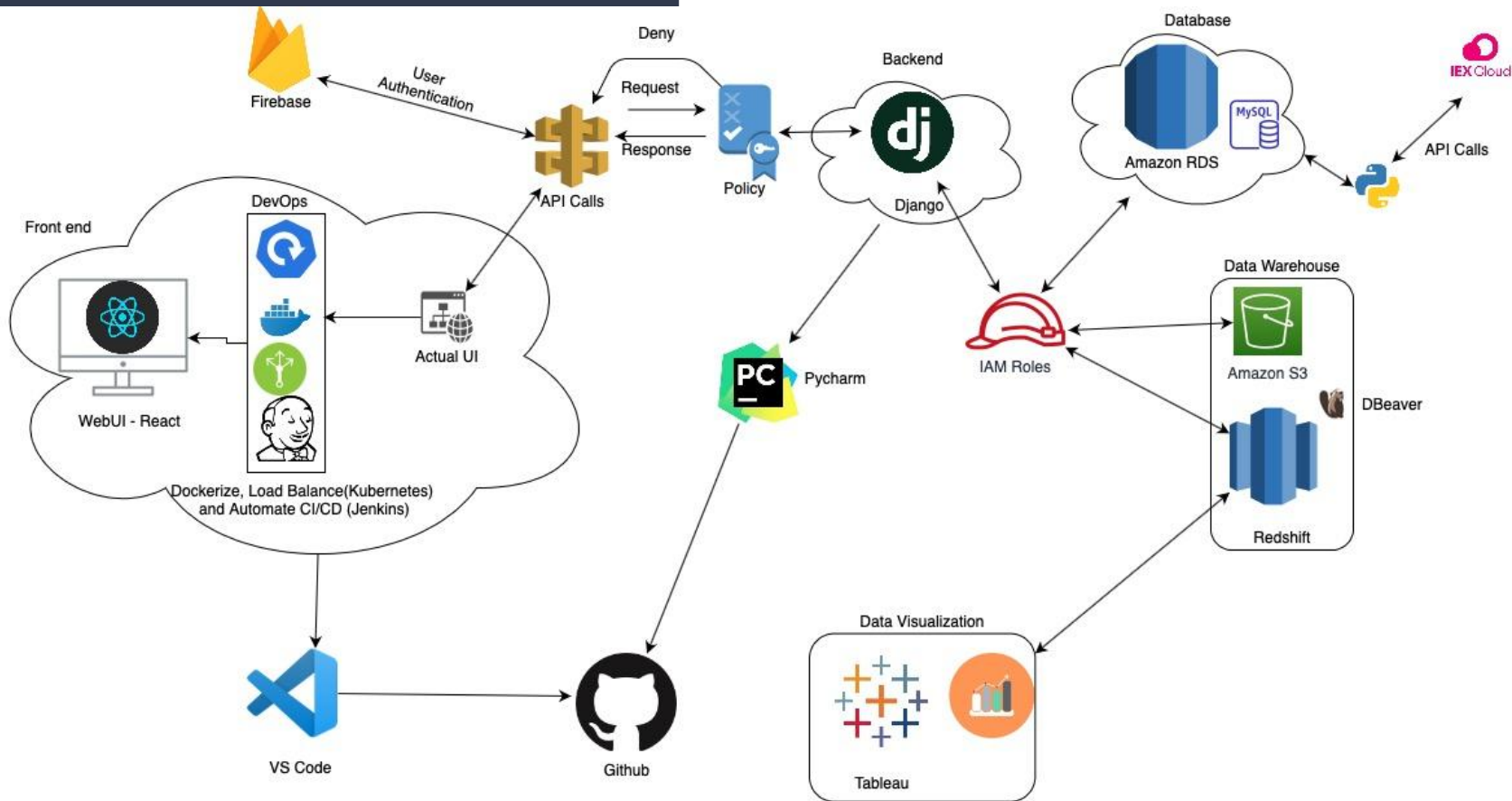
Chia-An Kuo(Net ID: cak580)

Sai Kiran(Net ID: vc2118)

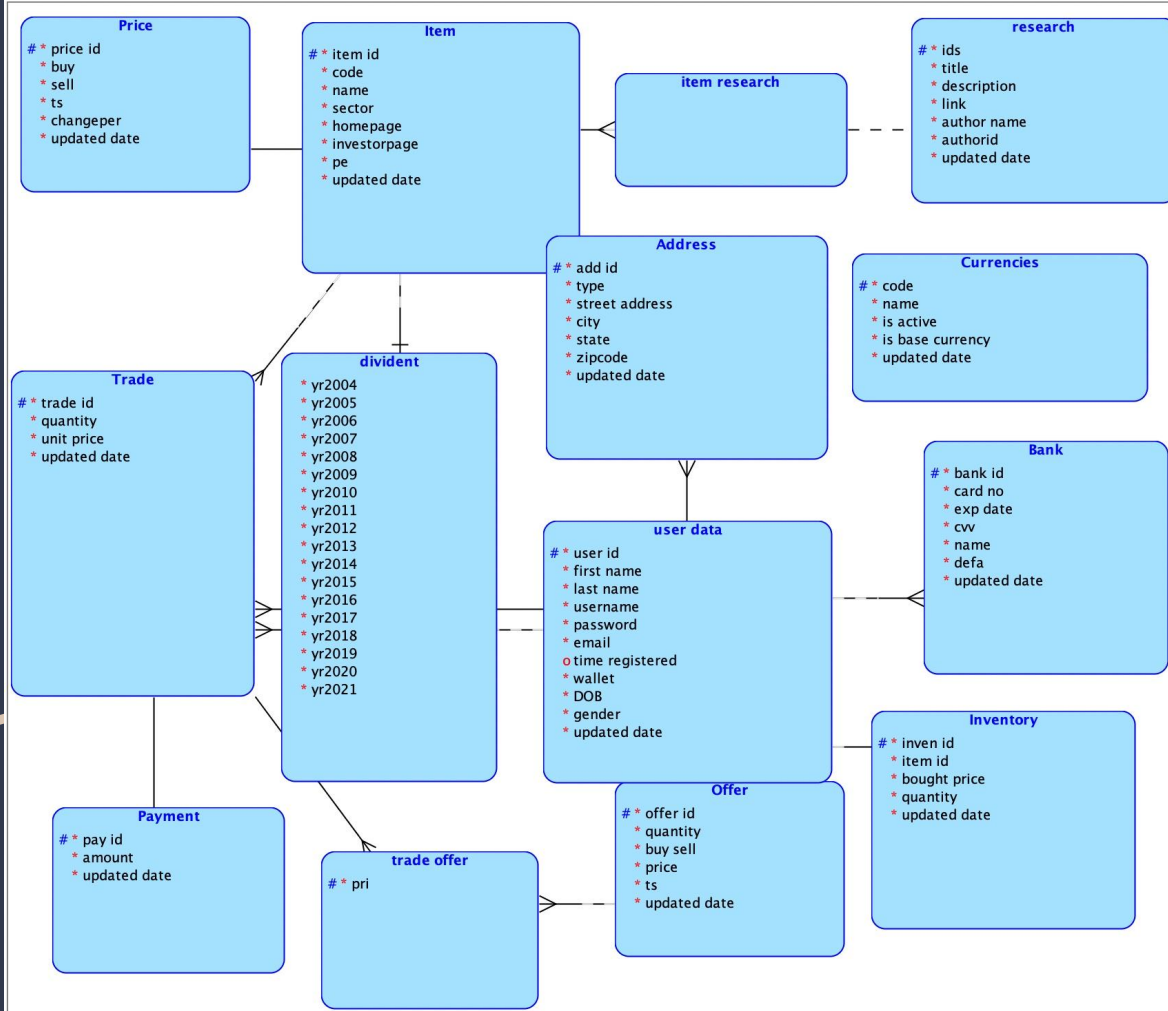
# Business Case

This is a virtual stock trading web application made using React.js hook. This application intends to help future investors learn the ups and downs of the stock market and build a portfolio of stocks for themselves. Beginning with a sum of \$1000 you get a chance to get accustomed to the buying and selling of stocks, track your progress over a period of time and make yourself ready to conquer the stock market. The app will be containerized, and could be deployed on kubernetes. We are using jenkins for the CI/CD pipeline. Whenever the code is modified, the app will deploy the latest version automatically.

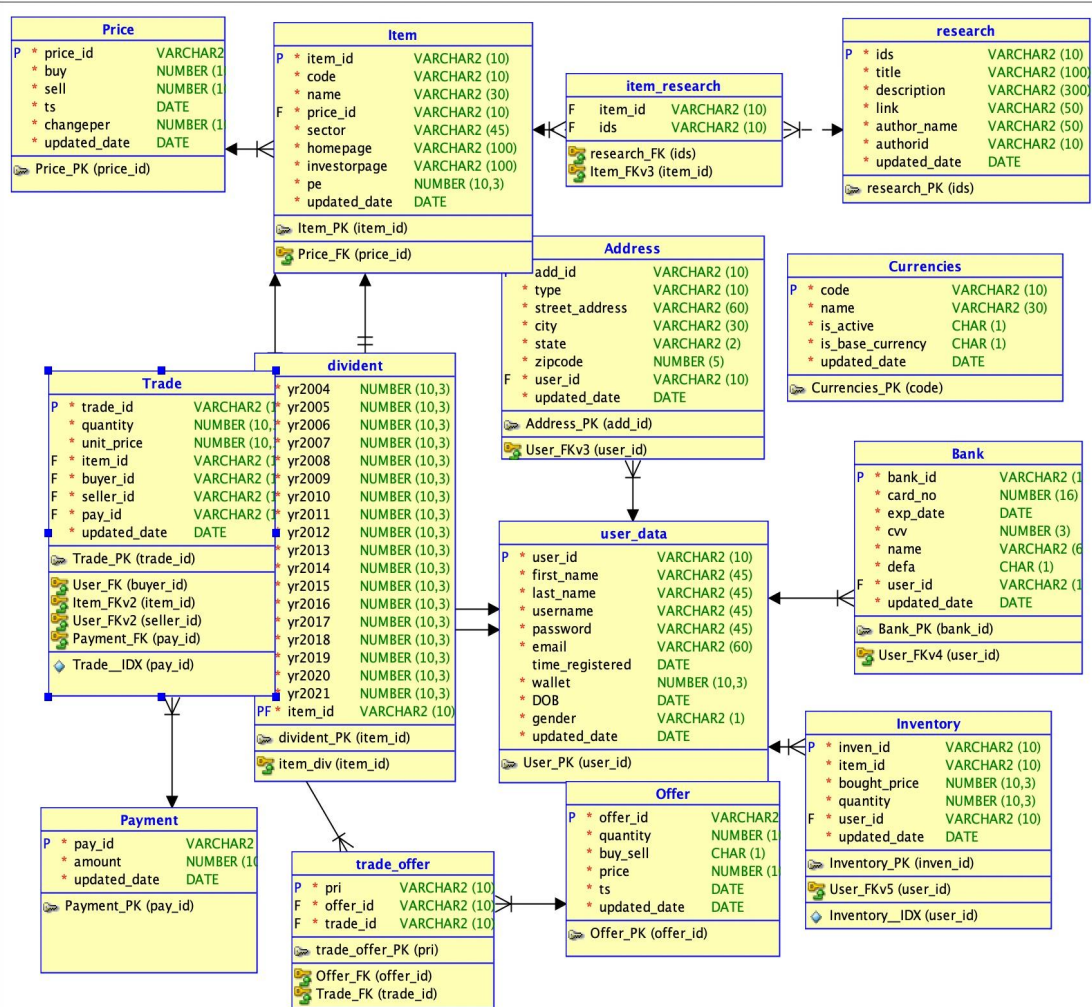
# Topology:



# Logical model of OLTP



# Relational model of OLTP



# Web application

## First research

This is the description. It is usually a lot longer than the title and contains information about the research that has been done



Source



Ann

## Add research

Title

Link

Description

Add research

StockApp

Hello chiaan456789@gmail.com | Sign Out

Exchanges

ann

123

0.50

sean

123

0.50

Eref

123

0.50

Hang

123

0.50

Top five stocks

AAPL

35 USD

TSLA

24 USD

AMZN

0.19 USD

NVDA

0.18 USD

MFST

0.159 USD

Highest dividend yield in current year

AAPL

35 USD

Highest dividend yield all time

AAPL

237 USD

Highest dividend yield growth in past 3 years

AAPL

17 USD

Ring Exchange

Highest dividend yield 2020 ▾

Name	Ticker	Ask	Bid	Dividend p/s	Dividend per 1000 spent	P/E	Sector
<div>Buy</div> Apple	AAPL	129	128.5	35 USD	271.32 USD	15	Technology
<div>Buy</div> Tesla	TSLA	129	128.5	24 USD	186.05 USD	15	Transport
<div>Buy</div> Amazon	AMZN	129	128.5	0.19 USD	1.47 USD	15	Sales
<div>Buy</div> Nvidia	NVDA	129	128.5	0.18 USD	1.40 USD	15	Industry
<div>Buy</div> Microsoft	MFST	129	128.5	0.159 USD	1.23 USD	15	Technology

## Dividend per share

2021	35
2020	20
2019	19
2018	18
2017	15
2016	13
2015	10
2014	12
2013	15
2012	10
2011	15
2010	12
2009	3
2008	5
2007	12

## Dividend Calculator

Amount in USD

0

Amount of shares

0

Current ask

129 USD

Projected dividend

0.00 USD

## Aggregated Dividend

Average past 20 years

13.17 USD

Total yield

237 USD

Stock number 1 in dividends per share in aggregate (20 years).

1

# Web application

Amount to Buy

Amount in USD

6

Amount of shares

0.05

Buy

Current ask

129 USD

Projected dividend

1.75 USD

Your Transaction History

Balance :  
9991

Buy  
Name: Apple  
pe / bid: 15/128.5  
amount in USD: \$ 6  
amount of Shares: 0.05

Buy  
Name: Apple  
pe / bid: 15/128.5  
amount in USD: \$ 3  
amount of Shares: 0.02

Profile

Username :  
First name :  
Last name :  
E - mail :  
Gender :  
Age :  
Address :  
City :  
State :  
Zip :  
Card No :  
Cvv :  
Phone :  
Balance :  
9991

Edit your Profile

Ring

Your Stock amount:

Balance :  
9991

Sell Name: Apple  
amount in USD: \$ 9

Your Transaction History

Balance :  
9991

Buy  
Name: Apple  
pe / bid: 15/128.5  
amount in USD: \$ 6  
amount of Shares: 0.05

Buy  
Name: Apple  
pe / bid: 15/128.5  
amount in USD: \$ 3  
amount of Shares: 0.02



## Edit Profile

Username :

First name :

Last name :

E - mail :

chiaan0426@gmail.com

Gender:

Age:

Address:

City:

# Docker

```
# pull official base image
FROM node:13.12.0-alpine

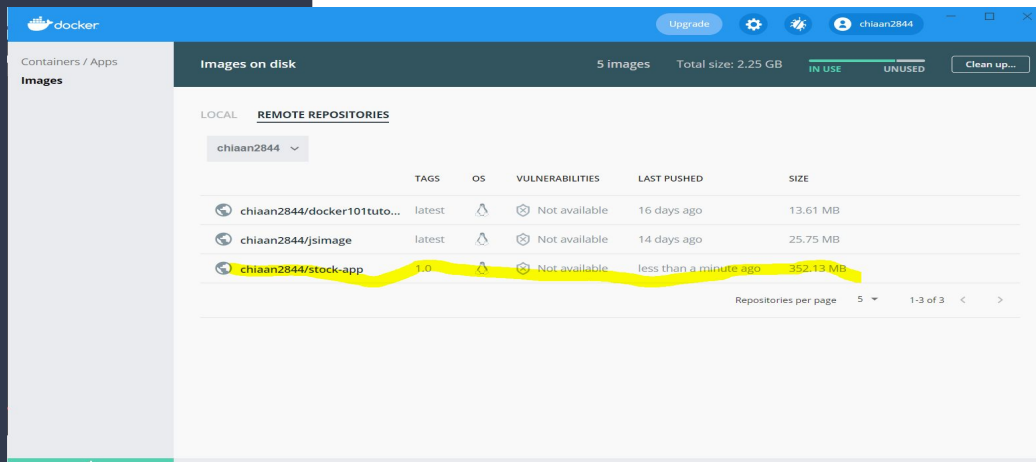
# set working directory
WORKDIR /app

# add `/app/node_modules/.bin` to $PATH
ENV PATH /app/node_modules/.bin:$PATH

# install app dependencies
COPY package.json ./
COPY package-lock.json ./
RUN npm install --silent
RUN npm install react-scripts@3.4.1 -g --silent
RUN npm install react-router-dom --silent
RUN npm install classnames --silent
RUN npm install react-util-kit --silent
RUN npm install --save firebase

# add app
COPY . ./

# start app
CMD ["npm", "start"]
```

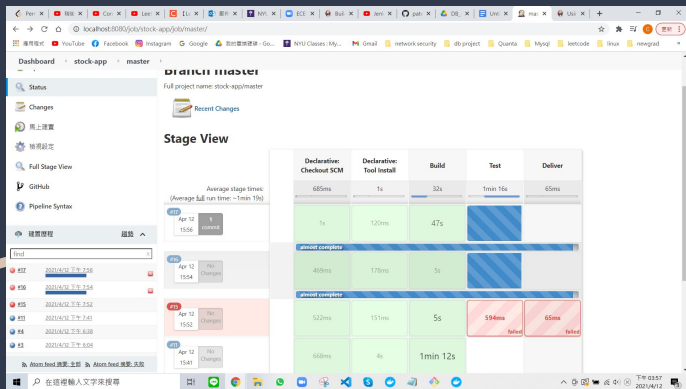


```
ann@LAPTOP-Q0DGEKUF MINGW64 ~/Desktop/db_project/my-stock-market (master)
$ docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
stock-app           1.0             d006757bf49d   About a minute ago   1.1GB
mongo               latest          30b3be246e39   5 days ago       449MB
mongo-express       latest          e5a1f58bcef1   6 days ago       128MB
jenkins/jenkins     lts             d457516b229f   6 days ago       571MB

ann@LAPTOP-Q0DGEKUF MINGW64 ~/Desktop/db_project/my-stock-market (master)
$ |
```



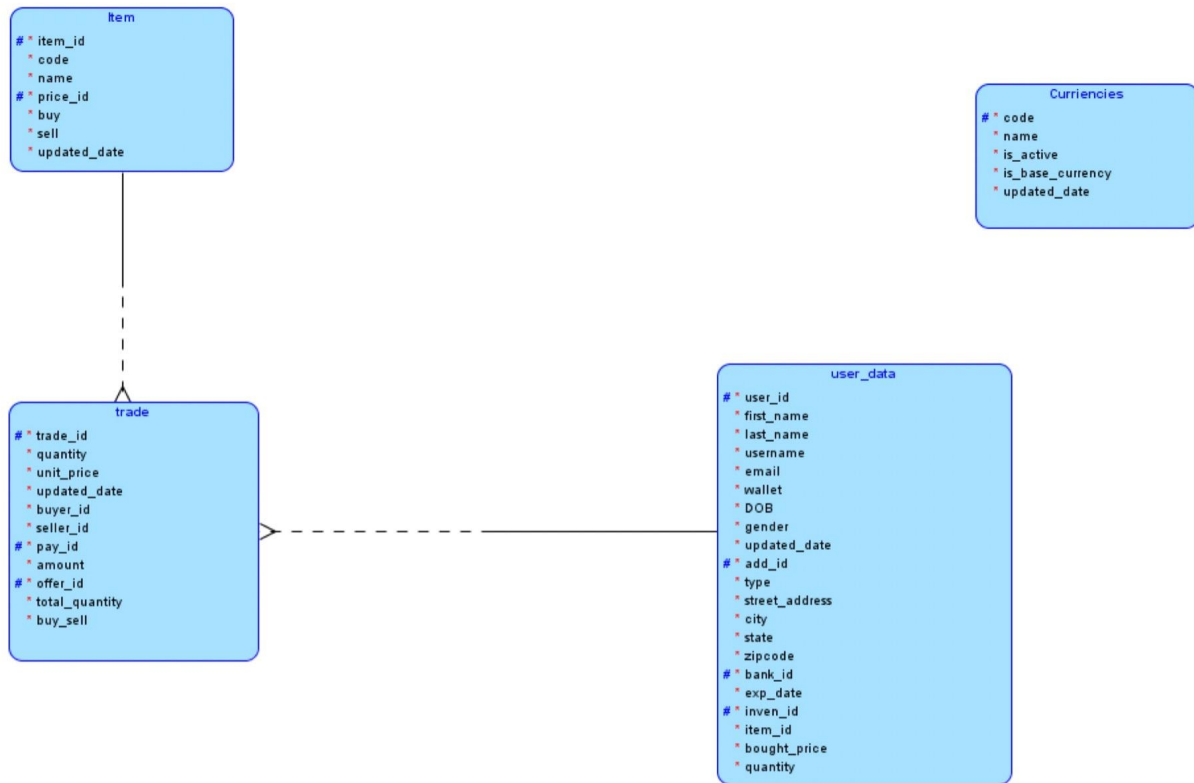
# CI/CD pipeline



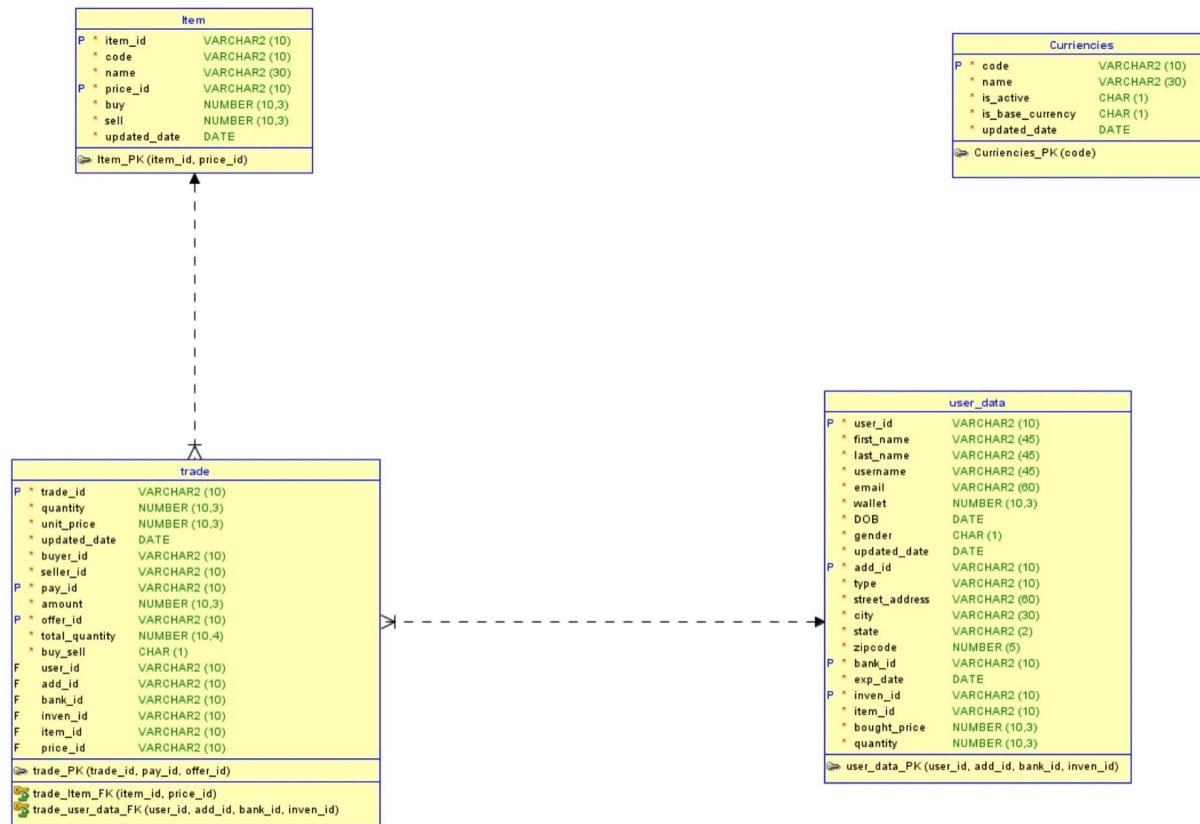
```
---
apiVersion: v1
kind: Service
metadata:
  name: stock-app-service
  labels:
    app: stock-app
spec:
  type: NodePort
  ports:
    - port: 8000
      protocol: TCP
      targetPort: 8000
      nodePort: 32121
  selector:
    app: stock-app
```

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: stock-app
  labels:
    app: stock-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: stock-app
  template:
    metadata:
      labels:
        app: stock-app
    spec:
      containers:
        - name: stock-app
          image: chiaan2844/stock-app
          ports:
            - containerPort: 8000
```

# Logical model of DW systems



# Relational model of DW systems



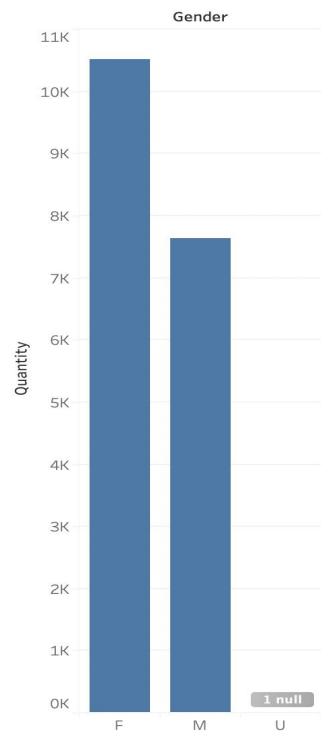
# ETL approach

- Extract the columns in the same order as the DW tables by using joins on OLTP and save it as a csv file.
- Copy the CSV file to Amazon S3.
- Truncate the External tables.
- Load the CSV file to the Staging table in Redshift using COPY command.
- For initial load, load the whole data into DW tables.
- For incremental load, run a Procedure to check with the already present rows and, update them if they are modified or, insert the new rows.

# Data analytics

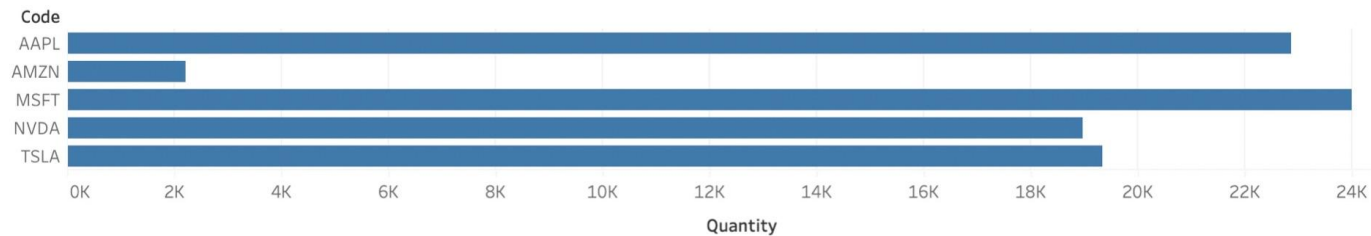
Columns	Gender
Rows	SUM(Quantity)

Sheet 3



Columns	SUM(Quantity)
Rows	Code

Sheet 1



# Lesson learned

1. Learned a lot about Stock Trading.
2. Implementation and design of Data Warehouse.
3. Creation of History, External and Partition Tables.
4. Implementing ETL, and Visualising using BI tools.
5. AWS - RDS, S3, IAM Roles, Redshift.
6. Technologies like React, Firebase, Docker, Kubernetes, Jenkins (CI/CD), Django, Tableau, DBeaver, etc.
7. Learned a lot about cross platform communication using APIs.
8. Last but not the least, how to manage time, work with teammates and importance of deadlines.

# Things didn't go well:

We spent a lot of time choosing the right DB for the use case, as we spent almost  $\frac{3}{4}$  th of the time working on MongoDB as a backend.