Quinn Smith
Bennett Lou
Alfred Timperley

CSC 436 MYSQL Stock Database

1) Project idea

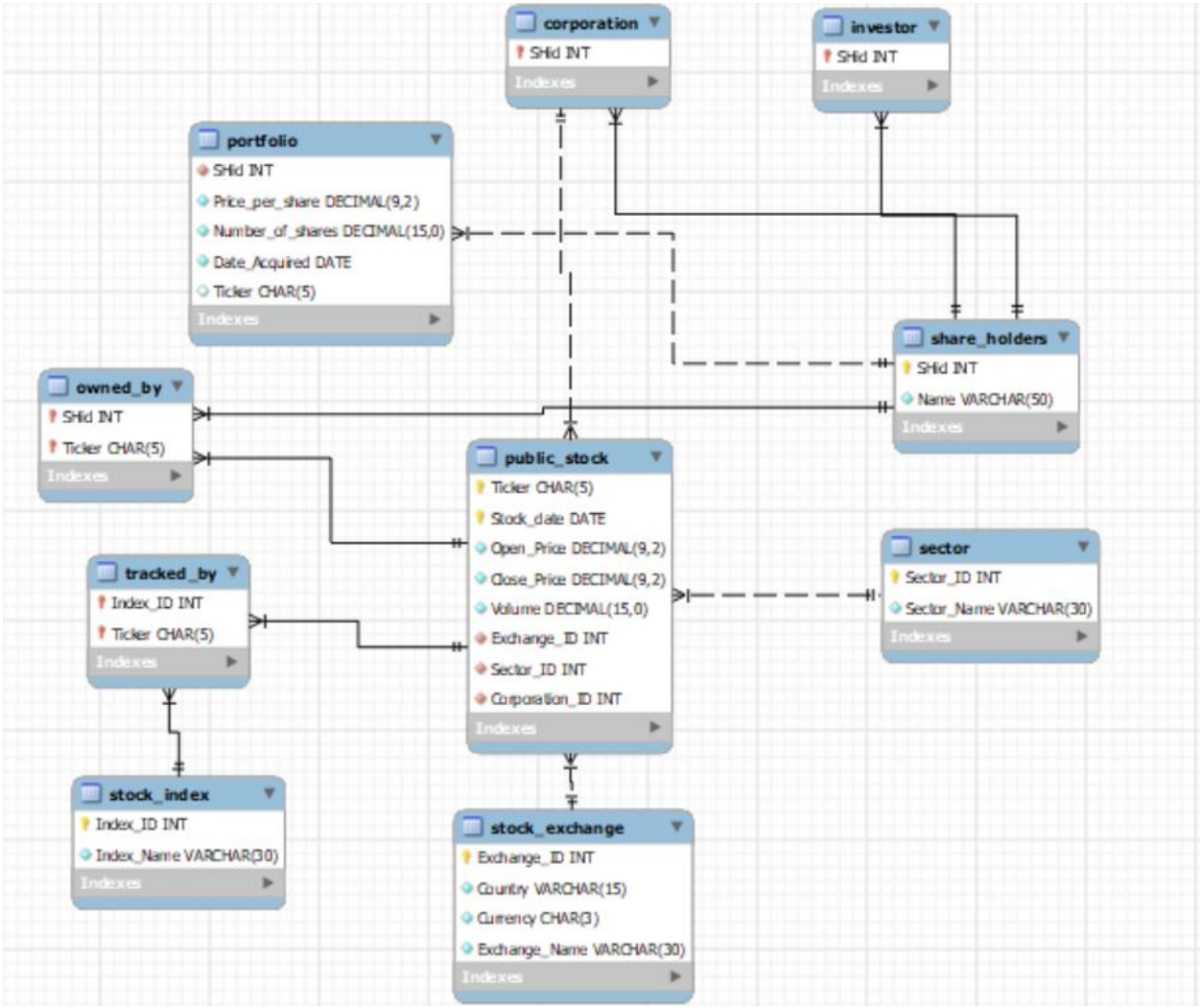
Create a database that stores all relevant information about equities
Stores information like:

- Shareholders for a stock
- What exchange the stock is listed on
- What indexes is the stock apart of
- What sector the stock is in
- Create a UI to be able to easily add / remove / update the database

2) User requirements (Basic operations and questions)

The Stock Database GUI is similar to various stock trading applications. Within it, you can collect or add data regarding shareholders, corporations, and their portfolios. The user enters in information in the blank fields, and it will be added to the database. On top of this you can directly access many queries and alter the MYSQL tables directly from the GUI, but it requires knowing proper SQL syntax.

3) ERD with Database tables, relationships, constraints etc.



4) Data sources, if exists

yfinance. A python library that scraps data off of the yahoo finance website. It contains historical stock data, so things like pricing, volume, dividends, major shareholders, etc. We used python to extract about 10 days of data of some selected stocks like Apple, Amazon, CVD, and more. The rest of the information like shareholders names and stock ownership is made up.

5) SQL queries and sample results for each, (cover different types of queries DDL, DML, DCL,)

Create table

```
CREATE TABLE Stock_Exchange
(
  Exchange_ID INT PRIMARY KEY,
  Country VARCHAR(15) NOT NULL,
  Currency CHAR(3) NOT NULL,
  Exchange_Name VARCHAR(30) NOT NULL
);
```

Insert

```
-- A stock holder buying a new stock
INSERT INTO Portfolio
VALUES (2, 141.32, 10, '2022-10-07', 'AAPL');
```

SHid	Price_per_share	Number_of_shares	Date_Acquired	Ticker
6	205.82	100	2022-10-24	TSLA
7	279.71	5	2022-10-13	ADBE
7	116.00	10	2022-10-26	AMZN
7	91.70	15	2022-10-18	CVS
7	233.94	70	2022-10-07	TSLA
8	220.95	10	2022-10-11	TSLA
9	206.42	50	2022-10-21	TSLA
9	96.43	20	2022-09-30	CVS
10	64.24	50	2022-10-25	BBY
10	118.00	60	2022-10-07	AMZN
2	141.32	10	2022-10-07	AAPL

Update

```
-- Removing only a few stocks from a share holder
UPDATE Portfolio
SET Number_of_shares = 5
WHERE SHid = 1 AND Ticker = 'ADBE' AND Date_Acquired = '2022-10-13';
```

SHid	Price_per_share	Number_of_shares	Date_Acquired	Ticker
1	294.74	5	2022-10-13	ADBE
1	57.33	25	2022-10-19	AMD
1	282.76	100	2022-09-29	TSLA
2	118.00	15	2022-10-12	IBM
2	100.00	50	2022-10-20	GOOGL
3	205.82	1000	2022-10-24	TSLA
3	87.18	300	2022-10-10	CVS
3	65.30	100	2022-10-14	BBY
4	114.10	50	2022-10-14	AMZN
5	308.43	40	2022-10-24	ADBE
5	57.47	70	2022-10-20	AMD

Delete

```
-- Removing the stock from a share holder
DELETE FROM Portfolio
WHERE SHid = 2 AND Ticker = 'AAPL';

-- See what corporations are investing in
SELECT Name, Ticker, Number_of_shares
FROM Corporation NATURAL LEFT JOIN Share_Holders NATURAL LEFT JOIN Portfolio
WHERE Corporation.SHid = Share_Holders.SHid AND Ticker IS NOT NULL;
```

Name	Ticker	Number_of_shares
Newport Investment Group	TSLA	70
Newport Investment Group	ADBE	5
Newport Investment Group	CVS	15
Newport Investment Group	AMZN	10
Bank of America	TSLA	10

Using Average

```
-- Getting the average purchase price of all stocks in every portfolio
SELECT Ticker, AVG(Price_per_share) as Average_purchase_price
FROM Portfolio
GROUP BY Ticker
ORDER BY Average_purchase_price DESC;
```

Ticker	Average_purchase_price
ADBE	294.293333
TSLA	225.951667
AAPL	141.320000
IBM	126.775000
AMZN	116.033333
GOOGL	101.450000
CVS	92.935000
BBY	64.770000
AMD	57.400000

Joins

```
-- Getting the name of every shareholder, the company they are invested in, and what industry that company is in
SELECT Name, Ticker, Sector_Name
FROM Share_Holders
NATURAL JOIN Portfolio
LEFT JOIN (SELECT DISTINCT Ticker, Sector_Name
FROM Public_Stock
LEFT JOIN Sector
USING(Sector_ID)) as sub
USING(Ticker);
```

Name	Ticker	Sector_Name
Harvey Fisher	ADBE	Information Technology
Harvey Fisher	AMD	Information Technology
Harvey Fisher	TSLA	Consumer Discretionary
Jose Mason	IBM	Information Technology
Jose Mason	GOOGL	Communication Services
Joseph Rhodes	TSLA	Consumer Discretionary
Joseph Rhodes	CVS	Health Care
Joseph Rhodes	BBY	Consumer Discretionary
Sadie Gibbs	AMZN	Consumer Discretionary
Joann Lambert	ADBE	Information Technology
Joann Lambert	AMD	Information Technology
Joann Lambert	CVS	Health Care
Joann Lambert	GOOGL	Communication Services
Joann Lambert	IBM	Information Technology
Dominic Mcdaniel	TSLA	Consumer Discretionary
Newport Inve...	ADBE	Information Technology
Newport Inve...	AMZN	Consumer Discretionary
Newport Inve...	CVS	Health Care
Newport Inve...	TSLA	Consumer Discretionary
Bank of America	TSLA	Consumer Discretionary
Franklin Burns	TSLA	Consumer Discretionary
Franklin Burns	CVS	Health Care
Ana Farmer	BBY	Consumer Discretionary

Select transactions

```
#Find transactions done by shareholders in certain month
SELECT  EXTRACT(Month FROM Date_acquired) AS month,
        COUNT(number_of_shares) AS Shareholder_transactions,
        Sum(number_of_shares) AS Shareholder_stock_purchase_number
FROM portfolio
GROUP BY EXTRACT(Month FROM Date_acquired)
ORDER BY EXTRACT(Month FROM Date_acquired) ASC;
```

month	Shareholder_transactions	Shareholder_stock_purchase_number
9	3	150
10	22	2145

6) Indexes

```
CREATE INDEX When_Bought
ON Portfolio (SHid, Date_Acquired);

SELECT SHid
FROM Portfolio USE INDEX(When_Bought)
WHERE Date_Acquired = '2022-10-13';
```

Result Grid

	SHid
▶	1
	7

7) Views

```
#View of total amount in money shareholders spent
CREATE VIEW Total_spent AS
SELECT SHid, SUM(Price_per_share * Number_of_shares) as Total_spent
FROM Portfolio
GROUP BY SHid;
```

Result Grid | Filter

	SHid	Total_spent
▶	1	31182.95
	2	6770.00
	3	238504.00
	4	5705.00
	5	32041.25
	6	20582.00
	7	20309.85
	8	2209.50
	9	12249.60
	10	10292.00

8) Description of the Technology used for your interface Implementation.

For the Project we used a combination of yfinance, Node and React to create a properly functioning stock database with a functional user interface. yfinance is Yahoo Finances API and can be used to comb through their data for information such as stock names, prices, volumes, market cap, and many other data points. Python scripts can pull data from Yahoo into Excel sheets that can then be imported into MYSQL. React is a free and open-source front-end

JavaScript library for building user interfaces. In React, applications are developed by creating reusable components. These components are individual pieces of a final interface. When assembled they form the application’s entire user interface.

```
LOAD DATA LOCAL INFILE 'C:/Users/qsmit/Downloads/436Project/CSC436-main/CSC436-main/data/Stock_Exchange.cs
INTO TABLE Stock_Exchange
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```