

CPSC 304 2015W2 Project Part II

Group Members

| Name | Student No. | Unix ID | Email |
|--------------|-------------|---------|----------------------------|
| Albert Xing | 40640104 | z6k8 | albert.xing@alumni.ubc.ca |
| Calvin Cheng | 36090132 | o7x8 | calvin.cheng@alumni.ubc.ca |
| Kyle Stadnyk | 52749025 | b5p5 | b5p5@ugrad.cs.ubc.ca |

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above.

In addition, we indicate that we are fully aware of the hrules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia.

ER Diagram

Please find the ER diagram in a separate folder ER-Diagram.

Database Schemas

Stock(symbol: CHAR(6), exchange: CHAR(6), name: VARCHAR(64))

Primary key: (symbol, exchange)

Functional dependencies: $symbol, exchange \rightarrow name$

The Stock entity represents a single stock, identified by a symbol and the exchange it is traded in. The given FD asserts that a unique symbol and exchange implies the name of the stock.

Holding(holding_id: INT, portfolio_id: INT, symbol: CHAR(6), exchange: CHAR(6), num_shares: INT, date: DATE, time: TIME, price: INT)

Primary key: (holding_id, portfolio_id)

Foreign key: portfolio_id references Portfolio

Foreign key: (symbol, exchange) references Stock

Functional dependencies: $holding_id, portfolio_id \rightarrow symbol, exchange, date, time, price$

A Holding is a number of shares of a stock inside a portfolio, with the date, time, and price (in USD) the shares were bought for. The FD states that a holding can be uniquely identified by a holding_id and the ID of the portfolio in which it belongs.

Portfolio(id: INT, purpose: VARCHAR(64), creation_date: DATE, principal: INT, cash: INT, owner_id: INT, manager_id: INT)

Primary key: id

Foreign key: owner_id references User

Foreign key: manager_id references User

Functional dependencies: $id \rightarrow purpose, creation_date, principal, cash, owner_id, manager_id$

A **Portfolio** is a collection of stock holdings for a particular purpose. The FD states that a portfolio can be identified by its unique ID. Also note that portfolios for self-directed investors are both owned and managed by the same investor, while managed portfolios are owned by an investor and managed by an advisor.

User(id: INT, name: VARCHAR(64), email: VARCHAR(64), phone: CHAR(10), password: BINARY(60), address_id: INT)
Primary key: id
Foreign key: address_id references Address
Functional dependencies: $id \rightarrow name, email, phone, password, address_id$

Users are either investors (who provide principal), advisors (who trade on behalf of investors), or self-directed investors (who assume both roles). Each user has a unique ID, as the FD states, through which they can be identified. We decided not to include **email** as a candidate key because it may be the case that a single individual would wish to open multiple accounts, for example a financial advisor who wants to open a separate individual account for their own use.

Address(id: INT, number: CHAR(5), street: VARCHAR(32), city: VARCHAR(32), country: VARCHAR(32), postal_code: CHAR(6))
Primary key: id
Functional dependencies: $id \rightarrow number, street, city, country, postal_code$

An **Address** represents a particular mailing address with a street number, street, city, country, and postal code. For a more efficient implementation, a unique ID is assigned to each address which allows for faster queries through a single column instead of multiple columns. Note that although postal codes often uniquely identify cities, this is not generally the case especially internationally. We decided not to include the FDs $postal_code \rightarrow city$ and $number, street, city \rightarrow postal_code$ on this basis.

Normalization

All of our tables are already in BCNF, and thus in 3NF. This is because the entities on the left-hand-side of all FDs are exactly the primary keys of each table.

Also note that there are no candidate keys other than the primary keys already given.

SQL DDL

```
CREATE TABLE User (  
    Id INT PRIMARY KEY,  
    Name VARCHAR(64) NOT NULL,  
    Email VARCHAR(64) NOT NULL,  
    Phone CHAR(10),  
    Password BINARY(60) NOT NULL,  
    Address_Id INT  
)  
  
CREATE TABLE Holding (  
    Holding_Id INT,  
    Portfolio_Id INT REFERENCES Portfolio ON DELETE RESTRICT,  
    Symbol CHAR(6),
```

```

        Exchange CHAR(6),
        Num_Shares INT NOT NULL,
        Date DATE NOT NULL,
        Time TIME NOT NULL,
        Price INT NOT NULL,
        PRIMARY KEY (Holding_Id, Portfolio_Id),
        FOREIGN KEY (Symbol, Exchange) REFERENCES Stock
    )

CREATE TABLE Stock (
    Symbol CHAR(6),
    Exchange CHAR(6),
    Name VARCHAR(64) NOT NULL,
    PRIMARY KEY (Symbol, Exchange)
)

CREATE TABLE Portfolio (
    Id INT PRIMARY KEY,
    Purpose VARCHAR(64),
    Creation_Date DATE,
    Principal INT NOT NULL,
    Cash INT NOT NULL,
    Owner_Id INT REFERENCES User,
    Manager_Id INT REFERENCES User
)

CREATE TABLE Address (
    Id INT PRIMARY KEY,
    Number CHAR(5),
    Street VARCHAR(32),
    City VARCHAR(32) NOT NULL,
    Country VARCHAR(32) NOT NULL,
    Postal_Code CHAR(6)
)

```

Note the ON DELETE RESTRICT clause in Holding. This prevents a portfolio from being removed from the table while it still contains active holdings - the investor should sell all assets before closing a portfolio.

Tables

Stock

| symbol | exchange | name |
|--------|----------|-----------------------------------|
| FB | NASDAQ | Facebook Inc |
| AAPL | NASDAQ | Apple Inc |
| MSFT | NASDAQ | Microsoft Corporation |
| GOOGL | NASDAQ | Alphabet Inc |
| 300431 | SHE | Beijing Baofeng Technology Co Ltd |

Holding

| holding_id | portfolio_id | num_shares | symbol | exchange | date | time | price |
|------------|--------------|------------|--------|----------|------------|-------|--------|
| 1 | 1 | FB | NASDAQ | 300 | 02/01/2016 | 13:23 | 98.63 |
| 2 | 1 | AAPL | NASDAQ | 8000 | 13/06/2011 | 20:34 | 46.56 |
| 1 | 2 | MSFT | NASDAQ | 232 | 27/03/2009 | 12:13 | 18.13 |
| 2 | 3 | GOOGL | NASDAQ | 2938 | 25/06/2010 | 13:08 | 236.57 |
| 3 | 5 | 300431 | SHE | 97832 | 27/03/2015 | 09:03 | 6.22 |

Portfolio

| id | purpose | creation_date | principal | cash | owner_id | manager_id |
|----|------------|---------------|-----------|-------|----------|------------|
| 1 | investing | 03/01/2015 | 10230 | 124 | 5 | 3 |
| 2 | investing | 08/13/2012 | 400000 | 3728 | 4 | 3 |
| 3 | trading | 10/04/2008 | 23000 | 12039 | 4 | 4 |
| 4 | retirement | 03/29/2007 | 340000 | 28377 | 5 | 2 |
| 5 | bored | 12/25/2004 | 4932000 | 0 | 1 | 1 |

User

| id | name | email | phone | address |
|----|------------------|---------------------|------------|---------|
| 1 | George Preece | gkp@example.com | 6392840427 | 6271 |
| 2 | Erik Combs | ecombs@example.com | 4382719285 | 3827 |
| 3 | Natalie Bolduc | natalie@example.com | 5837163859 | 285 |
| 4 | Madeleine Nilsen | mnilsen@example.com | 5829774637 | 6834 |
| 5 | Patrik Larsson | patrik@example.com | 7947252898 | 982 |

Address

| id | number | street | city | country | postal_code |
|------|--------|----------------|----------------|---------|-------------|
| 6271 | 6271 | 32nd St. | New York | US | 10001 |
| 3827 | 3827 | Main St. N | North Royalton | US | 44133 |
| 285 | 285 | Dewhurst Blvd. | Toronto | CA | M4J3J7 |
| 6834 | 6834 | Burrard St. | Vancouver | CA | V6C2E8 |
| 982 | 982 | De Castro St | Road Town | VGB | NULL |

Note that in these examples the various IDs may seem to exhibit external constraints, for example the `id` and `number` fields in `Address` are equal. However these patterns will not be enforced or implemented and are only for demonstrative purposes.