University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ___2___

Date: __03/01/2024__

Group Number: ___97___

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Ayush Joshi	20443560	i9h9l	ayushmehuljoshi@gmail.com
Dhruv Ganesh	90793209	t0b2m	Dhruvganesh307@gmail.com
Stallon Pinto	88077979	s7f6k	Stallonpinto03@gmail.com

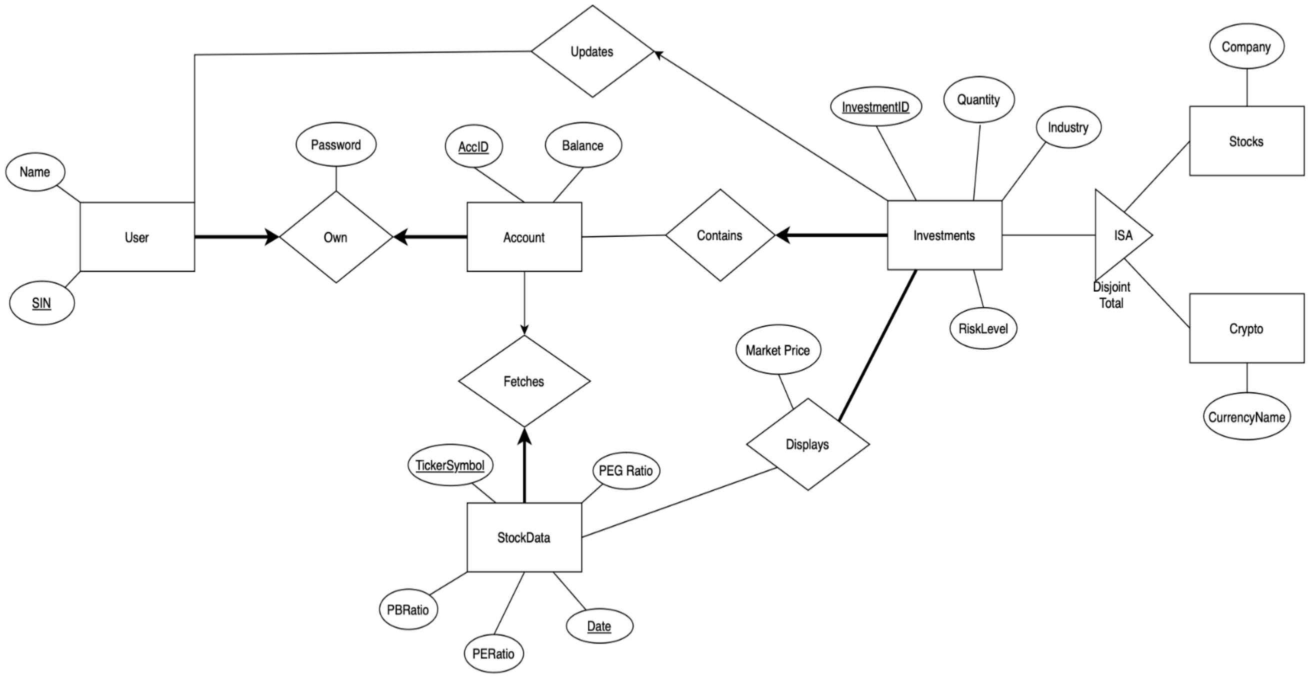
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

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

Summary

The database facilitates a comprehensive financial management system, encompassing User, Account, Stock Data, and Investments such as Stocks and Cryptocurrency. Users can register and log in, utilising the account to fetch market data to provide some technical indicators, which would aid in investment decision-making. The system allows users to make investment decisions by updating their portfolio based on the data-driven insights provided by the account.

ER Diagram



Our decisions on the ER-Diagram were based on the need to simplify the Relational Schema by removing any unnecessary entities and relations which complicated our conceptual schema without adding much value. For example, we removed the Company Analysis Entity since we recognised that the company analysis being referred to didn't require its own relation. We also removed the Bank Account Weak-Entity since we didn't see the need for our model to have access to the bank details of the user. Additionally, we removed the aggregate relation, after removing the need for a MarketData entity by integrating it together with the FinancialIndicator entity into the StockData Entity. Finally, we also added additional attributes to the ISA relations, the Owns relationship and we added Participation and Key Constraints into our ER-Diagram.

Relational Schema

Entities:

User (SIN : varchar (PK) , Name : varchar)
Account (AccID : varchar (PK), Balance : float)
Investments (InvestmentID : varchar (PK), Industry : varchar , RiskLevel : varchar, Quantity : float)
Stocks (InvestmentID : int (PK), Industry: varchar, RiskLevel: varchar, Company : varchar)
Crypto (InvestmentID : int (PK), Industry: varchar, RiskLevel: varchar, CurrencyName : varchar)
StockData (TickerSymbol : varchar (PK), Date : date (PK), MarketPrice : float , PERatio : float , P/BRatio : float, PEGRatio: float)

Relations:

Owns (**SIN** : Varchar (PK) (FK), **AcctID**: Varchar (FK), Password: Varchar)
Contains (**AcctId** : Varchar (PK) (FK), **InvestmentId**: Varchar (FK))
Updates (**InvestmentID** : int (PK) (FK), **SIN**: Varchar (FK))
Fetches (**AcctID** : Varchar (PK) (FK), **TickerSymbol** : Varchar (FK) (CK), **Date**: date (FK) (CK))
Displays(**TickerSymbol** : Varchar (PK) (FK), **Date**: date (PK) (FK), **InvestmentID**: int (PK) (FK))

Functional Dependencies

User:
SIN → Name

Account:
AcctID → Balance

Investments:
InvestmentID → Industry, RiskLevel, Quantity

Stocks:
InvestmentID → Industry, RiskLevel, Quantity, Company
Company → Industry, RiskLevel

Crypto:
InvestmentID → Industry, RiskLevel, Quantity, CurrencyName
CurrencyName → Industry, RiskLevel

StockData:
TickerSymbol, Date → PERatio, MarketPrice, PEGRatio, PBRatio
MarketPrice → PERatio, PEGRatio, PBRatio
PERatio → PEGRatio

Updates, Contains, Fetches, Displays:
Only trivial FDs

Owns:
AcctID → SIN, Password
SIN → AcctID, Password

Normalization

Stocks:
CompanyDetails (Company (PK), Industry)
Stocks (InvestmentID (PK), RiskLevel, Quantity, **Company** (FK))

Crypto:
CryptoDetails (CurrencyName (PK), Industry)
Crypto (InvestmentID (PK), RiskLevel, Quantity, **CurrencyName** (FK))

StockData:
DependantRatio (PERatio (PK), PEGRatio)
MarketPriceRatios(MarketPrice (PK), **PERatio** (FK), PBRatio)
StockData (TickerSymbol (PK), Date (PK), **MarketPrice** (FK))

After Normalization, we get the following schema in BCNF- Form :

User (SIN (PK) , Name)
Account (AccID (PK), Balance)
Investments (InvestmentID (PK), Industry, RiskLevel, Quantity)

Stocks:
CompanyDetails (Company, Industry)
Stocks (InvestmentID, RiskLevel, Quantity, **Company** (FK))

Crypto:
CryptoDetails (CurrencyName, Industry,)
Crypto (InvestmentID, RiskLevel, Quantity, **CurrencyName** (FK))

StockData:
DependantRatio (PERatio (PK), PEGRatio)
MarketPriceRatios(MarketPrice (PK), **PERatio** (FK), PBRatio)
StockData (TickerSymbol (PK), Date (PK), **MarketPrice** (FK))

Owns (**SIN** (PK) (FK), **AcctID** (FK), Password)
Contains (**AcctId** (PK) (FK), **InvestmentId** (FK))
Updates (**InvestmentID** (PK) (FK), **SIN** (FK))
Fetches (**AcctID** (PK) (FK), **TickerSymbol** (FK) (CK), **Date** (FK) (CK))
Displays(**TickerSymbol** (PK) (FK), **Date**(PK) (FK), **InvestmentID** (PK) (FK))

SQL Code

```
CREATE TABLE User (
   SIN VARCHAR(20) PRIMARY KEY,
   Name VARCHAR(100)
);


CREATE TABLE Account (
   AccID VARCHAR(20) PRIMARY KEY,
   Balance FLOAT
);


CREATE TABLE Investments (
   InvestmentID INT PRIMARY KEY,
   Industry VARCHAR(100),
   RiskLevel VARCHAR(50),
   Quantity FLOAT
);


CREATE TABLE CompanyDetails (
   Company VARCHAR(50) PRIMARY KEY,
   Industry VARCHAR(100),
);


CREATE TABLE CryptoDetails (
   Company VARCHAR(50) PRIMARY KEY,
   Industry VARCHAR(100),
);


CREATE TABLE Stocks (
   InvestmentID INT PRIMARY KEY,
   Company VARCHAR(100),
   RiskLevel VARCHAR(50),
   Quantity FLOAT,
   FOREIGN KEY (Company) references CompanyDetails(Company)
);


CREATE TABLE Crypto (
   InvestmentID INT PRIMARY KEY,
   CurrencyName VARCHAR(100),
   RiskLevel VARCHAR(50),
   Quantity FLOAT,
   FOREIGN KEY (CurrencyName) REFERENCES CompanyDetails(Company)
);


CREATE TABLE DependantRatio (
   PERatio FLOAT PRIMARY KEY,
   PEGRatio FLOAT,
);


CREATE TABLE MarketPriceRatio (
   MarketPrice VARCHAR(50),
   PERatio FLOAT,
   PBRatio FLOAT,
   PRIMARY KEY (MarketPrice),
   FOREIGN KEY (PERatio) REFERENCES DependantRatio(PERatio)
);



CREATE TABLE StockData (
   TickerSymbol VARCHAR(50),
   Date DATE,
   MarketPrice VARCHAR(50),
   PRIMARY KEY (TickerSymbol, Date),
   FOREIGN KEY (MarketPrice) REFERENCES MarketPriceRatio(MarketPrice)
);



CREATE TABLE Owns (
   SIN VARCHAR(20),
   AcctID VARCHAR(20),
   Password VARCHAR(50),
   PRIMARY KEY (SIN, AcctID),
   FOREIGN KEY (SIN) REFERENCES User(SIN),
   FOREIGN KEY (AcctID) REFERENCES Account(AccID)
);


CREATE TABLE Contains (
   AcctID VARCHAR(20),
   InvestmentID INT NOT NULL,
   PRIMARY KEY (InvestmentID),
   FOREIGN KEY (AcctID) REFERENCES Account(AccID),
   FOREIGN KEY (InvestmentID) REFERENCES Investments(InvestmentID) ON DELETE
CASCADE
);
```

```
CREATE TABLE Updates (
   InvestmentID INT,
   SIN VARCHAR(20),
   PRIMARY KEY (InvestmentID),
   FOREIGN KEY (InvestmentID) REFERENCES Investments(InvestmentID) ON DELETE CASCADE,
   FOREIGN KEY (SIN) REFERENCES User(SIN)
);
```

```
CREATE TABLE Fetches (
   AcctID VARCHAR(20),
   TickerSymbol VARCHAR(20),
   Date DATE,
   PRIMARY KEY (AcctID),
   UNIQUE (TickerSymbol, Date),
   FOREIGN KEY (AcctID) REFERENCES Account(AccID),
   FOREIGN KEY (TickerSymbol, Date) REFERENCES StockData(TickerSymbol, Date) ON DELETE CASCADE
);
```

```
CREATE TABLE Displays (
   TickerSymbol VARCHAR(20),
   Date DATE,
   InvestmentID INT NOT NULL,
   PRIMARY KEY (TickerSymbol, Date, InvestmentID),
   FOREIGN KEY (TickerSymbol, Date) REFERENCES StockData(TickerSymbol, Date),
   FOREIGN KEY (InvestmentID) REFERENCES Investments(InvestmentID)
);
```

Tuples

User:

```
INSERT INTO User (SIN, Name)
VALUES ('123456789', 'John Smith'),
            ('987654321', 'Alice Johnson'),
            ('456789123', 'Michael Brown'),
            ('789123456', 'Emily Davis'),
            ('321654987', 'David Wilson');
```

Account:

```
INSERT INTO Account (AccID, Balance)
VALUES ('123', 1000.50),
            ('456', 2500.75),
            ('789', 500.00),
            ('012', 3500.25),
            ('345', 200.00);
```

Investments:

```
INSERT INTO Investments (InvestmentID, Industry, RiskLevel, Quantity)
VALUES (1, 'Technology', 'High', 500),
            (2, 'Healthcare', 'Medium', 300),
            (3, 'Finance', 'Low', 700),
            (4, 'Energy', 'High', 250),
            (5, 'Consumer Goods', 'Medium', 400);
```

CompanyDetails:

```
INSERT INTO CompanyDetails (Company, Industry)
VALUES ('Apple Inc.', 'Technology');
INSERT INTO CompanyDetails (Company, Industry)
VALUES ('Amazon.com Inc.', 'E-commerce');
INSERT INTO CompanyDetails (Company, Industry)
VALUES ('Tesla Inc.', 'Automotive');
INSERT INTO CompanyDetails (Company, Industry)
VALUES ('Johnson & Johnson', 'Healthcare');
INSERT INTO CompanyDetails (Company, Industry)
VALUES ('Procter & Gamble Co.', 'Consumer Goods');
```

CryptoDetails:

```
INSERT INTO CryptoDetails (Company, Industry)
VALUES ('Bitcoin', 'Cryptocurrency'),
            ('Ethereum', 'Cryptocurrency'),
            ('Ripple', 'Cryptocurrency'),
            ('Litecoin', 'Cryptocurrency'),
            ('Cardano', 'Cryptocurrency');
```

Stocks:

```
INSERT INTO Stocks (InvestmentID, RiskLevel, Quantity, Company)
VALUES ('34', 'Low', 50, 'Apple Inc.');
INSERT INTO Stocks (InvestmentID, RiskLevel, Quantity, Company)
VALUES ('48', 'Medium', 20, 'Amazon.com Inc.');
INSERT INTO Stocks (InvestmentID, RiskLevel, Quantity, Company)
VALUES ('54', 'High', 10, 'Tesla Inc.');
INSERT INTO Stocks (InvestmentID, RiskLevel, Quantity, Company)
VALUES ('62', 'Low', 30, 'Johnson & Johnson');
INSERT INTO Stocks (InvestmentID, RiskLevel, Quantity, Company)
VALUES ('71', 'Low', 25, 'Procter & Gamble Co.');
```

Crypto:

```
INSERT INTO Crypto (InvestmentID, CurrencyName, RiskLevel, Quantity)
VALUES (1, 'Bitcoin', 'High', 2.5),
            (2, 'Ethereum', 'Medium', 5.0),
            (3, 'Cardano', 'Low', 10.0),
            (4, 'Ripple', 'High', 3.8),
            (5, 'Litecoin', 'Medium', 7.2);
```

DependantRatio:

```
INSERT INTO DependantRatio (PERatio, PEGRatio)
VALUES (10.5, 1.2),
       (15.2, 1.5),
       (8.7, 0.9),
       (20.3, 1.8),
       (12.6, 1.3);
```

MarketPriceRatio:

```
INSERT INTO MarketPriceRatio (MarketPrice, PERatio, PBRatio)
VALUES ('High', 20.5, 2.1),
       ('Medium', 15.3, 1.8),
       ('Low', 10.7, 1.5),
       ('Very High', 25.8, 2.5),
       ('Very Low', 8.9, 1.2);
```

StockData:

```
INSERT INTO StockData (TickerSymbol, Date, MarketPrice)
VALUES ('AAPL', '2024-02-28', 'High'),
       ('GOOGL', '2024-02-28', 'Medium'),
       ('MSFT', '2024-02-28', 'High'),
       ('AMZN', '2024-02-28', 'Very High'),
       ('FB', '2024-02-28', 'Medium');
```

Owns:

```
INSERT INTO Owns (SIN, AcctID, Password)
VALUES ('123456789', '123', 'password123'),
       ('987654321', '456', 'pass456'),
       ('456789123', '789', 'secure789'),
       ('789123456', '012', 'myacct012'),
       ('321654987', '345', 'password345');
```

Contains:

```
INSERT INTO Contains (AcctID, InvestmentID)
VALUES ('123', 1),
       ('123', 2),
       ('456', 3),'
       ('456', 4),
       ('789', 5);
```

Updates:

```
INSERT INTO Updates (InvestmentID, SIN)
VALUES (1, '123456789'),
       (2, '987654321'),
       (3, '456789123'),
       (4, '789123456'),
       (5, '321654987'); Fetches
```

Fetches:

```
INSERT INTO Fetches (AcctID, TickerSymbol, Date)
VALUES ('123', 'AAPL', '2024-02-28'),
       ('456', 'GOOGL', '2024-02-28'),
       ('789', 'MSFT', '2024-02-28'),
       ('012', 'AMZN', '2024-02-28'),
       ('345', 'FB', '2024-02-28');
```

Displays

```
INSERT INTO Displays (TickerSymbol, Date, InvestmentID)
VALUES ('AAPL', '2024-02-28', 1),
       ('GOOGL', '2024-02-28', 2),
       ('MSFT', '2024-02-28', 3),
       ('AMZN', '2024-02-28', 4),
       ('FB', '2024-02-28', 5);
```