CPSC 304 Project Cover Page

Milestone #: ___4___

Date: ___04/06/2024___

Group Number: ___97___

Name	Student Number	CS Alias (User id)	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.

Link to Github Repository

https://github.students.cs.ubc.ca/CPSC304-2023W-T2/project_i9h9l_s7f6k_t0b2m

SQL Script

```
CREATE TABLE Accounts (
    SIN VARCHAR2 (20) PRIMARY KEY, fullName VARCHAR2 (100), Balance NUMBER DEFAULT 1000 NOT NULL , pswd VARCHAR2 (20)
);


CREATE TABLE Investments (
    InvestmentID NUMBER PRIMARY KEY, Industry VARCHAR2 (100), RiskLevel VARCHAR2 (50), Quantity NUMBER
);


CREATE TABLE CompanyDetails (
    Company VARCHAR2 (50) PRIMARY KEY, Industry VARCHAR2 (100)
);


CREATE TABLE CryptoDetails (
    Company VARCHAR2 (50) PRIMARY KEY, Industry VARCHAR2 (100)
);


CREATE TABLE Stocks (
    InvestmentID NUMBER PRIMARY KEY, Company VARCHAR2 (100), RiskLevel VARCHAR2 (50), Quantity NUMBER, CONSTRAINT fk_stocks_company FOREIGN KEY (Company) REFERENCES CompanyDetails (Company)
);


CREATE TABLE Crypto (
    InvestmentID NUMBER PRIMARY KEY, CurrencyName VARCHAR2 (100), RiskLevel VARCHAR2 (50), Quantity NUMBER, CONSTRAINT fk_crypto_currencyname FOREIGN KEY (CurrencyName) REFERENCES CryptoDetails (Company)
);
```

```
CREATE TABLE DependantRatio (
   PERatio NUMBER PRIMARY KEY, PEGRatio NUMBER
);


CREATE TABLE MarketPriceRatio (
   MarketPrice VARCHAR2 (50) PRIMARY KEY, PERatio NUMBER, PBRatio NUMBER,
CONSTRAINT fk_marketpriceratio_peratio FOREIGN KEY (PERatio) REFERENCES
DependantRatio (PERatio)
);


CREATE TABLE StockData (
   TickerSymbol VARCHAR2 (50), DateOfTicker DATE, MarketPrice VARCHAR2 (50),
PRIMARY KEY (TickerSymbol, DateOfTicker), CONSTRAINT fk_stockdata_marketprice
FOREIGN KEY (MarketPrice) REFERENCES MarketPriceRatio (MarketPrice)
);


CREATE TABLE Contains (
   SIN VARCHAR2 (20), InvestmentID NUMBER NOT NULL, CONSTRAINT pk_contains
PRIMARY KEY (InvestmentID), CONSTRAINT fk_contains_sin FOREIGN KEY (SIN)
REFERENCES Accounts (SIN), CONSTRAINT fk_contains_investmentid FOREIGN KEY
(InvestmentID) REFERENCES Investments (InvestmentID) ON DELETE CASCADE
);


CREATE TABLE Updates (
   InvestmentID NUMBER, SIN VARCHAR2 (20), CONSTRAINT pk_updates PRIMARY
KEY (InvestmentID), CONSTRAINT fk_updates_investmentid FOREIGN KEY
(InvestmentID) REFERENCES Investments (InvestmentID), CONSTRAINT
fk_updates_sin FOREIGN KEY (SIN) REFERENCES Accounts (SIN) ON DELETE
CASCADE
);


CREATE TABLE Fetches (
   SIN VARCHAR2 (20), TickerSymbol VARCHAR2 (50), DateOfFetch DATE,
CONSTRAINT pk_fetches PRIMARY KEY (
      SIN, TickerSymbol, DateOfTicker
   ), CONSTRAINT unq_fetches_ticker_date UNIQUE (TickerSymbol, DateOfFetch),
CONSTRAINT fk_fetches_sin FOREIGN KEY (SIN) REFERENCES Accounts (SIN),
CONSTRAINT fk_fetches_stockdata FOREIGN KEY (TickerSymbol, DateOfFetch)
REFERENCES StockData (TickerSymbol, DateOfTicker) ON DELETE CASCADE
);
```

```
CREATE TABLE Displays (
   TickerSymbol VARCHAR2(20),
   DateOfTicker DATE,
   InvestmentID NUMBER NOT NULL,
   CONSTRAINT pk_displays PRIMARY KEY (TickerSymbol, DateOfTicker,
InvestmentID),
   CONSTRAINT fk_displays_stockdata FOREIGN KEY (TickerSymbol, DateOfTicker)
REFERENCES StockData (TickerSymbol, DateOfTicker) ON DELETE CASCADE,
   CONSTRAINT fk_displays_investments FOREIGN KEY (InvestmentID)
REFERENCES Investments (InvestmentID) ON DELETE CASCADE
);
```

Inserting data into Accounts table

```
INSERT INTO Accounts (SIN, fullName, Balance, pswd)
VALUES ('123456789', 'John Smith', 1000.50, 'password123');

INSERT INTO Accounts (SIN, fullName, Balance, pswd)
VALUES ('987654321', 'Alice Johnson', 2500.75, 'password234');

INSERT INTO Accounts (SIN, fullName, Balance, pswd)
VALUES ('456789123', 'Michael Brown', 500.00, 'password345');

INSERT INTO Accounts (SIN, fullName, Balance, pswd)
VALUES ('789123456', 'Emily Davis', 3765.00, 'password456');

INSERT INTO Accounts (SIN, fullName, Balance, pswd)
VALUES ('321654987', 'David Wilson', 200.00, 'password567');

INSERT INTO Accounts (SIN, fullName, Balance, pswd)
VALUES ('234567890', 'Olivia Moore', 3200.75, 'oliviaM2024');

INSERT INTO Accounts (SIN, fullName, Balance, pswd)
VALUES ('345678901', 'Liam Johnson', 4150.50, 'liamJ2024');

INSERT INTO Accounts (SIN, fullName, Balance, pswd)
VALUES ('456789012', 'Emma Williams', 5275.25, 'emmaW2024');

INSERT INTO Accounts (SIN, fullName, Balance, pswd)
VALUES ('567890123', 'Noah Smith', 1900.00, 'noahS2024');

INSERT INTO Accounts (SIN, fullName, Balance, pswd)
VALUES ('678901234', 'Ava Davis', 2850.00, 'avaD2024');
```

Inserting data into Investments table

```
INSERT INTO Investments (InvestmentID, Industry, RiskLevel, Quantity)
VALUES (1, 'Technology', 'High', 500);

INSERT INTO Investments (InvestmentID, Industry, RiskLevel, Quantity)
VALUES (2, 'Healthcare', 'Medium', 300);

INSERT INTO Investments (InvestmentID, Industry, RiskLevel, Quantity)
VALUES (3, 'Finance', 'Low', 700);

INSERT INTO Investments (InvestmentID, Industry, RiskLevel, Quantity)
VALUES (4, 'Energy', 'High', 250);

INSERT INTO Investments (InvestmentID, Industry, RiskLevel, Quantity)
VALUES (5, 'Consumer Goods', 'Medium', 400);

INSERT INTO Investments (InvestmentID, Industry, RiskLevel, Quantity)
VALUES (6, 'Technology', 'Medium', 600);

INSERT INTO Investments (InvestmentID, Industry, RiskLevel, Quantity)
VALUES (7, 'Energy', 'Low', 800);

INSERT INTO Investments (InvestmentID, Industry, RiskLevel, Quantity)
VALUES (8, 'Healthcare', 'High', 400);

INSERT INTO Investments (InvestmentID, Industry, RiskLevel, Quantity)
VALUES (9, 'Technology', 'High', 300);

INSERT INTO Investments (InvestmentID, Industry, RiskLevel, Quantity) VALUES (10,
'Consumer Goods', 'Low', 500);
```

Inserting data into CompanyDetails table

```
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');

INSERT INTO CompanyDetails (Company, Industry)
VALUES ('Microsoft Corp.', 'Technology');

INSERT INTO CompanyDetails (Company, Industry)
VALUES ('BP plc', 'Energy');

INSERT INTO CompanyDetails (Company, Industry)
VALUES ('Pfizer Inc.', 'Healthcare');

INSERT INTO CompanyDetails (Company, Industry)
VALUES ('Nvidia Corp.', 'Technology');

INSERT INTO CompanyDetails (Company, Industry)
VALUES ('Coca-Cola Co.', 'Consumer Goods');
```

Inserting data into CryptoDetails table

```
INSERT INTO CryptoDetails (Company, Industry)
VALUES ('Bitcoin', 'Cryptocurrency');

INSERT INTO CryptoDetails (Company, Industry)
VALUES ('Ethereum', 'Cryptocurrency');

INSERT INTO CryptoDetails (Company, Industry)
VALUES ('Ripple', 'Cryptocurrency');

INSERT INTO CryptoDetails (Company, Industry)
VALUES ('Litecoin', 'Cryptocurrency');

INSERT INTO CryptoDetails (Company, Industry)
VALUES ('Cardano', 'Cryptocurrency');

INSERT INTO CryptoDetails (Company, Industry)
VALUES ('Polkadot', 'Cryptocurrency');

INSERT INTO CryptoDetails (Company, Industry)
VALUES ('Chainlink', 'Cryptocurrency');

INSERT INTO CryptoDetails (Company, Industry)
VALUES ('Binance Coin', 'Cryptocurrency');

INSERT INTO CryptoDetails (Company, Industry)
VALUES ('Solana', 'Cryptocurrency');
```

```
INSERT INTO CryptoDetails (Company, Industry)
VALUES ('Monero', 'Cryptocurrency');
```

Inserting data into Stocks table

```
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.');

INSERT INTO Stocks (InvestmentID, Company, RiskLevel, Quantity)
VALUES (72, 'Microsoft Corp.', 'Medium', 45);

INSERT INTO Stocks (InvestmentID, Company, RiskLevel, Quantity)
VALUES (73, 'BP plc', 'Low', 110);

INSERT INTO Stocks (InvestmentID, Company, RiskLevel, Quantity)
VALUES (74, 'Pfizer Inc.', 'High', 90);

INSERT INTO Stocks (InvestmentID, Company, RiskLevel, Quantity)
VALUES (75, 'Nvidia Corp.', 'High', 30);

INSERT INTO Stocks (InvestmentID, Company, RiskLevel, Quantity)
VALUES (76, 'Coca-Cola Co.', 'Low', 120);
```

Inserting data into Crypto table

```
INSERT INTO Crypto (InvestmentID, CurrencyName, RiskLevel, Quantity)
VALUES (1, 'Bitcoin', 'High', 2.5);

INSERT INTO Crypto (InvestmentID, CurrencyName, RiskLevel, Quantity)
VALUES (2, 'Ethereum', 'Medium', 5.0);

INSERT INTO Crypto (InvestmentID, CurrencyName, RiskLevel, Quantity)
VALUES (3, 'Cardano', 'Low', 10.0);
```

```
INSERT INTO Crypto (InvestmentID, CurrencyName, RiskLevel, Quantity)
VALUES (4, 'Ripple', 'High', 3.8);

INSERT INTO Crypto (InvestmentID, CurrencyName, RiskLevel, Quantity)
VALUES (5, 'Litecoin', 'Medium', 7.2);

INSERT INTO Crypto (InvestmentID, CurrencyName, RiskLevel, Quantity)
VALUES (77, 'Polkadot', 'High', 4.0);

INSERT INTO Crypto (InvestmentID, CurrencyName, RiskLevel, Quantity)
VALUES (78, 'Chainlink', 'Medium', 6.5);

INSERT INTO Crypto (InvestmentID, CurrencyName, RiskLevel, Quantity)
VALUES (79, 'Binance Coin', 'Low', 1.2);

INSERT INTO Crypto (InvestmentID, CurrencyName, RiskLevel, Quantity)
VALUES (80, 'Solana', 'High', 2.7);

INSERT INTO Crypto (InvestmentID, CurrencyName, RiskLevel, Quantity)
VALUES (81, 'Monero', 'Medium', 3.5);
```

Inserting data into DependantRatio table

```
INSERT INTO DependantRatio (PERatio, PEGRatio)
VALUES (10.5, 1.2);

INSERT INTO DependantRatio (PERatio, PEGRatio)
VALUES (15.2, 1.5);

INSERT INTO DependantRatio (PERatio, PEGRatio)
VALUES (8.7, 0.9);

INSERT INTO DependantRatio (PERatio, PEGRatio)
VALUES (20.3, 1.8);

INSERT INTO DependantRatio (PERatio, PEGRatio)
VALUES (12.6, 1.3);

INSERT INTO DependantRatio (PERatio, PEGRatio)
 VALUES (22.5, 2.1);

INSERT INTO DependantRatio (PERatio, PEGRatio)
VALUES (18.7, 1.9);

INSERT INTO DependantRatio (PERatio, PEGRatio)
VALUES (14.3, 1.4);
```

Inserting data into MarketPriceRatio table

```
INSERT INTO MarketPriceRatio (MarketPrice, PERatio, PBRatio)
VALUES ('High', 12.6, 1.3);

INSERT INTO MarketPriceRatio (MarketPrice, PERatio, PBRatio)
VALUES ('Medium', 15.2, 1.5);

INSERT INTO MarketPriceRatio (MarketPrice, PERatio, PBRatio)
VALUES ('Low', 10.5, 1.2);

INSERT INTO MarketPriceRatio (MarketPrice, PERatio, PBRatio)
VALUES ('Very High', 20.3, 1.8);

INSERT INTO MarketPriceRatio (MarketPrice, PERatio, PBRatio)
VALUES ('Very Low', 8.7, 0.9);

INSERT INTO MarketPriceRatio (MarketPrice, PERatio, PBRatio)
VALUES ('Moderate', 22.5, 2.1);

INSERT INTO MarketPriceRatio (MarketPrice, PERatio, PBRatio)
VALUES ('Speculative', 18.7, 1.9);

INSERT INTO MarketPriceRatio (MarketPrice, PERatio, PBRatio)
VALUES ('Conservative', 14.3, 1.4);
```

Inserting data into StockData table

```
INSERT INTO StockData (TickerSymbol, DateOfTicker, MarketPrice)
VALUES ('AAPL', TO_DATE('2024-02-28', 'YYYY-MM-DD'), 'High');

INSERT INTO StockData (TickerSymbol, DateOfTicker, MarketPrice)
VALUES ('GOOGL', TO_DATE('2024-02-28', 'YYYY-MM-DD'), 'Medium');

INSERT INTO StockData (TickerSymbol, DateOfTicker, MarketPrice)
VALUES ('MSFT', TO_DATE('2024-02-28', 'YYYY-MM-DD'), 'High');

INSERT INTO StockData (TickerSymbol, DateOfTicker, MarketPrice)
VALUES ('AMZN', TO_DATE('2024-02-28', 'YYYY-MM-DD'), 'Very High');

INSERT INTO StockData (TickerSymbol, DateOfTicker, MarketPrice)
VALUES ('FB', TO_DATE('2024-02-28', 'YYYY-MM-DD'), 'Medium');

INSERT INTO StockData (TickerSymbol, DateOfTicker, MarketPrice)
VALUES ('NVDA', TO_DATE('2024-03-01', 'YYYY-MM-DD'), 'Speculative');

INSERT INTO StockData (TickerSymbol, DateOfTicker, MarketPrice)
```

```
 VALUES ('PFE', TO_DATE('2024-03-01', 'YYYY-MM-DD'), 'Moderate');

INSERT INTO StockData (TickerSymbol, DateOfTicker, MarketPrice)
VALUES ('BP', TO_DATE('2024-03-01', 'YYYY-MM-DD'), 'Conservative');
```

Inserting data into Contains table

```
INSERT INTO Contains (SIN, InvestmentID)
VALUES ('123456789', 1);

INSERT INTO Contains (SIN, InvestmentID)
VALUES ('987654321', 2);

INSERT INTO Contains (SIN, InvestmentID)
VALUES ('456789123', 3);

INSERT INTO Contains (SIN, InvestmentID)
VALUES ('789123456', 4);

INSERT INTO Contains (SIN, InvestmentID)
VALUES ('321654987', 5);

INSERT INTO Contains (SIN, InvestmentID)
VALUES ('234567890', 6);

INSERT INTO Contains (SIN, InvestmentID)
 VALUES ('345678901', 7);

INSERT INTO Contains (SIN, InvestmentID)
VALUES ('456789012', 8);
```

Inserting data into Updates table

```
INSERT INTO Updates (InvestmentID, SIN)
VALUES (1, '123456789');

INSERT INTO Updates (InvestmentID, SIN)
VALUES (2, '987654321');

INSERT INTO Updates (InvestmentID, SIN)
VALUES (3, '456789123');

INSERT INTO Updates (InvestmentID, SIN)
VALUES (4, '789123456');

INSERT INTO Updates (InvestmentID, SIN)
VALUES (5, '321654987');
```

```
INSERT INTO Updates (InvestmentID, SIN)
VALUES (9, '567890123');

INSERT INTO Updates (InvestmentID, SIN)
VALUES (10, '678901234');
```

Inserting data into Fetches table

Inserting for ticker AAPL

```
INSERT INTO Fetches (SIN, TickerSymbol, DateOfFetch)
SELECT '123456789', 'AAPL', DATE '2024-02-28'
FROM StockData
WHERE TickerSymbol = 'AAPL' AND DateOfTicker = DATE '2024-02-28';
```

Inserting for ticker GOOGL

```
INSERT INTO Fetches (SIN, TickerSymbol, DateOfFetch)
SELECT '987654321', 'GOOGL', DATE '2024-02-28'
FROM StockData
WHERE TickerSymbol = 'GOOGL' AND DateOfTicker = DATE '2024-02-28';
```

Inserting for ticker MSFT

```
INSERT INTO Fetches (SIN, TickerSymbol, DateOfFetch)
SELECT '456789123', 'MSFT', DATE '2024-02-28'
FROM StockData
WHERE TickerSymbol = 'MSFT' AND DateOfTicker = DATE '2024-02-28';
```

Inserting for ticker AMZN

```
INSERT INTO Fetches (SIN, TickerSymbol, DateOfFetch)
SELECT '789123456', 'AMZN', DATE '2024-02-28'
FROM StockData
WHERE TickerSymbol = 'AMZN' AND DateOfTicker = DATE '2024-02-28';
```

Inserting for ticker FB

```
INSERT INTO Fetches (SIN, TickerSymbol, DateOfFetch)
SELECT '321654987', 'FB', DATE '2024-02-28'
FROM StockData
WHERE TickerSymbol = 'FB' AND DateOfTicker = DATE '2024-02-28';
```

Inserting data into Displays table

```
INSERT INTO Displays (TickerSymbol, DateOfTicker, InvestmentID)
VALUES ('AAPL', TO_DATE('2024-02-28', 'YYYY-MM-DD'), 1);

INSERT INTO Displays (TickerSymbol, DateOfTicker, InvestmentID)
VALUES ('GOOGL', TO_DATE('2024-02-28', 'YYYY-MM-DD'), 2);
```

```
INSERT INTO Displays (TickerSymbol, DateOfTicker, InvestmentID)
VALUES ('MSFT', TO_DATE('2024-02-28', 'YYYY-MM-DD'), 3);

INSERT INTO Displays (TickerSymbol, DateOfTicker, InvestmentID)
VALUES ('AMZN', TO_DATE('2024-02-28', 'YYYY-MM-DD'), 4);

INSERT INTO Displays (TickerSymbol, DateOfTicker, InvestmentID)
VALUES ('FB', TO_DATE('2024-02-28', 'YYYY-MM-DD'), 5);

INSERT INTO Displays (TickerSymbol, DateOfTicker, InvestmentID)
VALUES ('NVDA', TO_DATE('2024-03-01', 'YYYY-MM-DD'), 6);

INSERT INTO Displays (TickerSymbol, DateOfTicker, InvestmentID)
VALUES ('PFE', TO_DATE('2024-03-01', 'YYYY-MM-DD'), 7);

INSERT INTO Displays (TickerSymbol, DateOfTicker, InvestmentID)
VALUES ('BP', TO_DATE('2024-03-01', 'YYYY-MM-DD'), 8);
```

Project Description :

Our website functions as a financial management platform, catering to users who aim to maintain and grow their personal or business finances. With features enabling users to create and update personal accounts, the platform emphasizes ease of access and the importance of up-to-date financial information. It offers an interactive snapshot of an individual's financial health by allowing users to review accounts with balances above a certain value, potentially highlighting opportunities for investment or savings.

The essence of the platform extends into the realm of investment tracking, allowing users to sift through various industries to assess and manage their investment portfolios. The seamless integration of account information with specific investments suggests a personalized approach to financial management, enabling users to make informed decisions based on comprehensive data analytics.

Advanced features of the website provide an aggregation of investment data, giving users a bird's-eye view of industry-specific financial engagement, as well as the ability to pinpoint industries with significant investment activities. This level of insight, coupled with the ability to identify users with diversified investments across multiple key sectors, positions the platform as a tool not only for tracking finances but also for strategic investment planning.

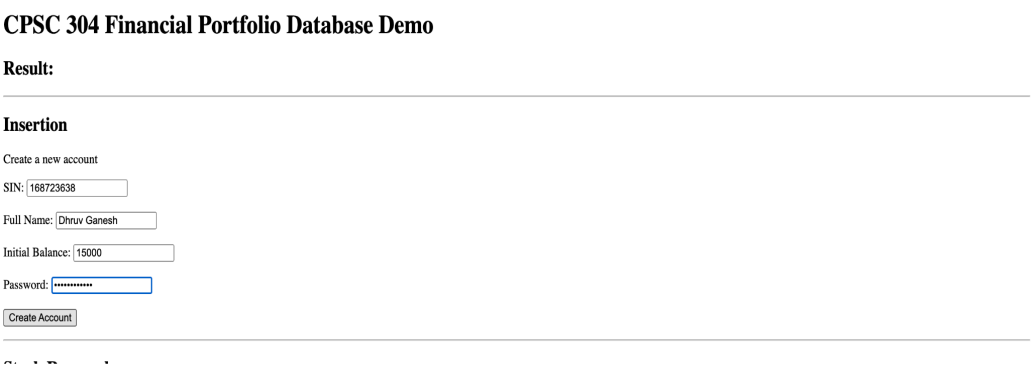
The changes from our schema in milestone 2 is that we removed the user entity and the and owns relations connecting it to the account entity in order to reduce some redundancy. Instead, we moved the necessary attributes and relations that were previously connected to the user entity, to the account table itself. We also removed financial indicators and avoided using crypto for easier implementation of our website. This had to happen mainly due to several difficulties met by the team during implementation of an alternative stack- the PERN Stack (PostgreSQL, Express, React, Node), and other issues, including a teammate falling ill. As a result we had to make a lastt-minute switch to a different stack which was better supported by the tutorials (PHP/Oracle).

Queries

Insert Query- Insert a new account

INSERT INTO Accounts (SIN, fullName, Balance, pswd)
VALUES ('1122z334455', 'Jane Doe', 5000, 'janeDoePass');

```
<h2>Insertion</h2>
<p>Create a new account</p>
<form method="POST" action="project.php">
    <input type="hidden" id="insertAccountRequest" name="insertAccountRequest">
    SIN: <input type="text" name="accountSIN"> <br /><br />
    Full Name: <input type="text" name="accountFullName"> <br /><br />
    Initial Balance: <input type="text" name="accountBalance"> <br /><br />
    Password: <input type="password" name="accountPassword"> <br /><br />
    <input type="submit" value="Create Account" name="createAccountSubmit"></p>
</form>
<hr />
    </div>
```



ACCOUNTS BEFORE INSERT:

Retrieved data from Accounts table:

SIN	fullName	Balance	pswd
123456789	John Smith	16872	password123
987654321	Alice Johnson	2500.75	password234
456789123	Michael Brown	500	password345
789123456	Emily Davis	3765	password456
321654987	David Wilson	200	password567
16789263	Jason Maj	1000	TEST
234567890	Olivia Moore	3200.75	oliviaM2024
345678901	Liam Johnson	4150.5	liamJ2024
456789012	Emma Williams	5275.25	emmaW2024
567890123	Noah Smith	1900	noahS2024
678901234	Ava Davis	2850	avaD2024

ACCOUNTS AFTER INSERT:

Retrieved data from Accounts table:

SIN	fullName	Balance	pswd
123456789	John Smith	16872	password123
987654321	Alice Johnson	2500.75	password234
456789123	Michael Brown	500	password345
789123456	Emily Davis	3765	password456
321654987	David Wilson	200	password567
16789263	Jason Maj	1000	TEST
168723638	Dhruv Ganesh	15000	passworddrhv
234567890	Olivia Moore	3200.75	oliviaM2024
345678901	Liam Johnson	4150.5	liamJ2024
456789012	Emma Williams	5275.25	emmaW2024
567890123	Noah Smith	1900	noahS2024
678901234	Ava Davis	2850	avaD2024

Delete Query- Remove a stock entry

DELETE FROM Stocks
WHERE Company = 'Tesla Inc.';

```
<h2>Stock Removal</h2>
<p>Remove a stock entry from the database</p>
<form method="POST" action="project.php">
    <input type="hidden" id="deleteStockRequest" name="deleteStockRequest">
    Investment ID of Stock to Remove: <input type="text" name="deleteStockID"> <br /><br />
    <input type="submit" value="Remove Stock" name="removeStockSubmit"></p>
</form>
<hr />
```

Stock Removal

Remove a stock entry from the database

Investment ID of Stock to Remove: 76

Remove Stock

Result:

BEFORE DELETE REQUEST:

Retrieved data from Stocks table:

InvestmentID	RiskLevel	Quantity	Company
48	Amazon.com Inc.	Medium	20
54	Tesla Inc.	High	10
62	Johnson	Low	30
71	Procter Co.	Low	25
34	Apple Inc.	Low	50
72	Microsoft Corp.	Medium	45
73	BP plc	Low	110
74	Pfizer Inc.	High	90
75	Nvidia Corp.	High	30
76	Coca-Cola Co.	Low	120

AFTER DELETE REQUEST:

Retrieved data from Stocks table:

InvestmentID	RiskLevel	Quantity	Company
48	Amazon.com Inc.	Medium	20
54	Tesla Inc.	High	10
62	Johnson	Low	30
71	Procter Co.	Low	25
34	Apple Inc.	Low	50
72	Microsoft Corp.	Medium	45
73	BP plc	Low	110
74	Pfizer Inc.	High	90
75	Nvidia Corp.	High	30

Update Query - Update balance for a specific account

UPDATE Accounts
SET Balance = Balance + 1000
WHERE SIN = '123456789';

```
<h2>Update Account Balance</h2>
<p>Update the balance for an existing account</p>
<form method="POST" action="project.php">
    <input type="hidden" id="updateBalanceRequest" name="updateBalanceRequest">
    SIN of Account: <input type="text" name="updateSin" required> <br /><br />
    New Balance: <input type="number" name="updateBalance" step="0.01" required> <br /><br />
    <input type="submit" value="Update Balance" name="updateBalanceSubmit">
</form>
<hr />
```

Update Account Balance

Update the balance for an existing account

SIN of Account: 168723638

New Balance: 20000

Update Balance

RESULT BEFORE UPDATE:

Retrieved data from Accounts table:

SIN	fullName	Balance	pswd
123456789	John Smith	16872	password123
987654321	Alice Johnson	2500.75	password234
456789123	Michael Brown	500	password345
789123456	Emily Davis	3765	password456
321654987	David Wilson	200	password567
16789263	Jason Maj	1000	TEST
168723638	Dhruv Ganesh	15000	passworddrhv
234567890	Olivia Moore	3200.75	oliviaM2024
345678901	Liam Johnson	4150.5	liamJ2024
456789012	Emma Williams	5275.25	emmaW2024
567890123	Noah Smith	1900	noahS2024
678901234	Ava Davis	2850	avaD2024

RESULT AFTER UPDATE:

Retrieved data from Accounts table:

SIN	fullName	Balance	pswd
123456789	John Smith	16872	password123
987654321	Alice Johnson	2500.75	password234
456789123	Michael Brown	500	password345
789123456	Emily Davis	3765	password456
321654987	David Wilson	200	password567
16789263	Jason Maj	1000	TEST
168723638	Dhruv Ganesh	20000	passworddrhv
234567890	Olivia Moore	3200.75	oliviaM2024
345678901	Liam Johnson	4150.5	liamJ2024
456789012	Emma Williams	5275.25	emmaW2024
567890123	Noah Smith	1900	noahS2024
678901234	Ava Davis	2850	avaD2024

Selection Query- Find all accounts with a balance greater than 1000

SELECT * FROM Accounts
WHERE Balance > 1000;

```
<h2>Find Accounts with High Balance</h2>
<p>View all high balance accounts.</p>
<form method="GET", action="project.php">
    <input type="hidden" id="highBalanceAccounts" name="highBalanceRequest">
    <input type="submit" value="Find High Balance Accounts" name="submitHighBalance">
</form>

<hr />
```

Retrieved data from Accounts table:

SIN	fullName	Balance	pswd
123456789	John Smith	16872	password123
987654321	Alice Johnson	2500.75	password234
789123456	Emily Davis	3765	password456
16789263	Jason Maj	1000	TEST
168723638	Dhruv Ganesh	20000	passworddrhv
234567890	Olivia Moore	3200.75	oliviaM2024
345678901	Liam Johnson	4150.5	liamJ2024
456789012	Emma Williams	5275.25	emmaW2024
567890123	Noah Smith	1900	noahS2024
678901234	Ava Davis	2850	avaD2024

Projection Query - Find distinct industries of investments

SELECT DISTINCT Industry
FROM Investments;

```
<h2>View Distinct Investment Industries</h2>
<p>View all unique industries represented in our investments.</p>
<form method="GET" action="project.php">
    <input type="hidden" id="distinctIndustriesRequest" name="distinctIndustriesRequest">
    <input type="submit" value="View Industries" name="viewIndustriesSubmit">
</form>
<hr />
```

Result:

Retrieved data from Investment table:

Industry
Healthcare
Finance
Technology
Energy
Consumer Goods

Join Query - Select all investments made by a particular account

```
SELECT I.*
FROM Investments I
JOIN Contains C ON I.InvestmentID = C.InvestmentID
WHERE C.SIN = '123456789';
```

```
<h2>Investments by Account</h2>
<form method="GET", action="project.php">
    <input type="hidden" id="joinQueryRequest" name="joinQueryRequest">
    SIN: <input type="text" name="sinJoin"> <br /><br />
    <input type="submit" name="joinInvestments" value="Find Investments">
</form>
<hr />
```

Investments by Account

SIN: 123456789

Find Investments

Retrieved data from Join:

SIN	InvestmentID
123456789	1

Aggregation with Group by - Find the total quantity of investments by industry

SELECT Industry, SUM(Quantity) AS TotalQuantity
FROM Investments
GROUP BY Industry;

```
<h2>Total Investments by Industry</h2>
<form method="GET", action="project.php">
    <input type="hidden" id="aggregationRequest" name="aggregationRequest">
    <input type="submit" name="aggregateInvestments" value="Aggregate Investments">
</form>
<hr />
```

Retrieved aggregate data from Investments:

Industry	Number of Investments
Technology	3
Consumer Goods	2
Energy	2
Healthcare	2
Finance	1

Aggregation with Group by and Having - Find industries where the total quantity of investments exceeds 5

SELECT Industry, SUM(Quantity) AS TotalQuantity
FROM Investments
GROUP BY Industry
HAVING SUM(Quantity) > 5;

```
<h2>Industries with Total Investments Exceeding 5</h2>
<form method="GET", action="project.php">
    <input type="hidden" id="aggregationRequest2" name="aggregationRequest2">
    <input type="submit" name="aggregateInvestments2" value="Aggregate Investments 2">
</form>
<hr />
```

Result:

Retrieved division data:

SIN

Nested Aggregation- Find the maximum average quantity of investments across all industries

```
SELECT MAX(AvgQuantity) AS MaxAvgQuantity
FROM (SELECT Industry, AVG(Quantity) AS AvgQuantity
    FROM Investments
    GROUP BY Industry) SubQuery;
```

```
<h2>Maximum Average Quantity by Industry</h2>
<form method="GET", action="project.php">
    <input type="hidden" id="nestedAggregationRequest" name="nestedAggregationRequest">
    <input type="submit" name="nestedAggregationSubmit" value="Find Max Avg Quantity">
</form>
<hr />
```

Result:

Retrieved nested aggregate data:

Maximum Average Quantity

700

Division - Selects accounts that have investments in both specified industries

```
SELECT SIN FROM Accounts
WHERE SIN IN (
   SELECT C.SIN
   FROM Contains C
   JOIN Investments I ON C.InvestmentID = I.InvestmentID
   WHERE I.Industry = 'Technology'
)
INTERSECT
SELECT SIN FROM Accounts
WHERE SIN IN (
   SELECT C.SIN
   FROM Contains C
   JOIN Investments I ON C.InvestmentID = I.InvestmentID
   WHERE I.Industry = 'Finance'
);
```

```
<h2>Accounts Investing in Both Specified Industries</h2>
<form method="GET", action="project.php">
    <input type="hidden" id="divisionRequest" name="divisionRequest">
    Industry 1: <input type="text" name="industry1" required> <br /><br />
    Industry 2: <input type="text" name="industry2" required> <br /><br />
    <input type="submit" name="divisionSubmit" value="Find Accounts">
</form>
<hr />
```

Accounts Investing in Both Specified Industries

Industry 1: Technology

Industry 2: Finance

Find Accounts

Result:

Retrieved division data:

SIN