Financial Portfolio Management System

Objective

Design and implement a comprehensive financial portfolio management system using MySQL. This system should allow users to track and manage their investment portfolios, providing useful information on asset allocation, performance, and risk.The primary objective of developing this financial portfolio management system is to empower small family portfolios with a comprehensive and user-friendly tool for effective management and analysis of our investment portfolio

Data Generation

To simulate real-world financial data, we'll create tables for stocks, bonds, and cash, along with historical price data. For simplicity, let's consider a portfolio with four stocks, four bonds,four user families and cash(savings,checkings,emergency,investment).

Data

This real-world financial information comes from yahoo finance.We will select these 4 companies as our stock choices.

Stocks:

1	AAPL	Apple Inc.	150.50	Technology
2	MSFT	Microsoft Corp.	300.75	Technology
3	JPM	JPMorgan Chase	120.25	Finance
4	GOOGL	Alphabet Inc.	2500.00	Technology

This real-world financial information comes from yahoo finance.We will select these 4 bonds as our bond choices for this project.

Bonds:

1	US Treasury	Treasury Bond 2025	1000.00	2.5	2025-12-31
2	IBM	Corporate Bond A	950.50	3.0	2024-08-15
3	GE	Corporate Bond B	800.25	2.0	2023-05-20
4	Ford	Auto Bond	1100.75	2.75	2026-02-28

These numbers are randomly generated using a random module from python standard library.

Cash:

1	Savings	50000.00
2	Checking	10000.00
3	Emergency	25000.00
4	Investment	75000.00

This real-world financial information comes from yahoo finance.

AssetHistory:

1 | AAPL | 160.00 | 2023-01-02 |
2 | MSFT | 305.00 | 2023-01-02 |
3 | JPM | 122.00 | 2023-01-02 |
4 | GOOGL | 2525.00| 2023-01-02 |
5 | AAPL | 155.00 | 2023-01-03 |
6 | MSFT | 298.00 | 2023-01-03 |
7 | JPM | 118.00 | 2023-01-03 |
8 | GOOGL | 2490.00| 2023-01-03

These numbers are randomly generated using a random module from python standard library.

Portfolio:

1 (1, 1, 160.00, '2023-01-02'),
2 (2, 2, 305.00, '2023-01-02'),
3 (3, 3, 122.00, '2023-01-02'),
4 (4, 4, 2525.00, '2023-01-02'),
5 (5, 1, 155.00, '2023-01-03'),
6 (6, 2, 298.00, '2023-01-03'),
7 (7, 3, 118.00, '2023-01-03'),
8 (8, 4, 2490.00, '2023-01-03');

Tables

- **Stocks Table:**
  - StockID (int, primary key)
  - Symbol (varchar)
  - CompanyName (varchar)
  - CurrentPrice (decimal)
  - Sector (varchar)
- **Bonds Table:**
  - BondID (int, primary key)
  - Issuer (varchar)
  - BondName (varchar)
  - CurrentPrice (decimal)
  - CouponRate (decimal)
  - MaturityDate (date)
- **Cash Table:**
  - CashID (int, primary key)
  - AccountName (varchar)

    - Balance (decimal)
- **AssetHistory Table:**
    - HistoryID (int, primary key)
    - AssetID (int, foreign key referencing stocks,bonds,cash table)
    - Price (decimal)
    - Date (date)
- **Portfolio Table:**
    - PortfolioID (int, primary key)
    - UserID (int, foreign key referencing user table)
    - StockID (int, foreign key referencing stocks table)
    - BondID (int, foreign key referencing bonds table)
    - CashID (int, foreign key referencing cash table)
    - Quantity (int)
    - PurchasePrice (decimal)
    - PurchaseDate (date)

Questions

1. What is the current value of the portfolio?
2. Implement metrics like portfolio return.
3. What is the percentage allocation of stocks, bonds, and cash in the portfolio?
4. What is the total investment value for each user's portfolio?
5. Which stocks have the highest and lowest current prices in the portfolio?
6. Which user has the highest total portfolio value?

Answers

1.

```
SELECT

    p.PortfolioID,

    u.Username,

    (COALESCE(s.CurrentPrice, 0) * p.Quantity) AS StockValue,

    (COALESCE(b.CurrentPrice, 0) * p.Quantity) AS BondValue,

    (COALESCE(c.Balance, 0)) AS CashValue,
```

((COALESCE(s.CurrentPrice, 0) * p.Quantity) + (COALESCE(b.CurrentPrice, 0) * p.Quantity) + COALESCE(c.Balance, 0)) AS TotalValue

FROM Portfolio p

JOIN User u ON p.UserID = u.UserID

LEFT JOIN Stocks s ON p.StockID = s.StockID

LEFT JOIN Bonds b ON p.BondID = b.BondID

LEFT JOIN Cash c ON p.CashID = c.CashID;

2.

SELECT

p.PortfolioID,

u.Username,

((COALESCE(s.CurrentPrice, 0) * p.Quantity + COALESCE(b.CurrentPrice, 0) * p.Quantity + COALESCE(c.Balance, 0)) - p.InitialInvestment) / p.InitialInvestment * 100 AS PortfolioReturn

FROM Portfolio p

JOIN User u ON p.UserID = u.UserID

LEFT JOIN Stocks s ON p.StockID = s.StockID

LEFT JOIN Bonds b ON p.BondID = b.BondID

LEFT JOIN Cash c ON p.CashID = c.CashID;

3.

SELECT

p.PortfolioID,

u.Username,

((COALESCE(s.CurrentPrice, 0) * p.Quantity) / TotalValue) * 100 AS StockAllocation,

```
    ((COALESCE(b.CurrentPrice, 0) * p.Quantity) / TotalValue) * 100 AS BondAllocation,

    (COALESCE(c.Balance, 0) / TotalValue) * 100 AS CashAllocation

FROM Portfolio p

    JOIN User u ON p.UserID = u.UserID

    LEFT JOIN Stocks s ON p.StockID = s.StockID

    LEFT JOIN Bonds b ON p.BondID = b.BondID

    LEFT JOIN Cash c ON p.CashID = c.CashID

  CROSS JOIN ( SELECT SUM((COALESCE(s.CurrentPrice, 0) * p.Quantity) +
(COALESCE(b.CurrentPrice, 0) * p.Quantity) + COALESCE(c.Balance, 0)) AS TotalValue

        FROM Portfolio p

            LEFT JOIN Stocks s ON p.StockID = s.StockID

            LEFT JOIN Bonds b ON p.BondID = b.BondID

            LEFT JOIN Cash c ON p.CashID = c.CashID

    ) AS Total;
```

4.

```
SELECT

    u.Username,

    SUM((COALESCE(s.CurrentPrice, 0) * p.Quantity) + (COALESCE(b.CurrentPrice, 0) *
p.Quantity) + COALESCE(c.Balance, 0)) AS TotalInvestmentValue

FROM Portfolio p

    JOIN User u ON p.UserID = u.UserID

    LEFT JOIN Stocks s ON p.StockID = s.StockID

    LEFT JOIN Bonds b ON p.BondID = b.BondID

    LEFT JOIN Cash c ON p.CashID = c.CashID
```

```
GROUP BY u.Username;
```

5.

```
SELECT
        Symbol,
        MAX(CurrentPrice) AS HighestCurrentPrice,
        MIN(CurrentPrice) AS LowestCurrentPrice
FROM Stocks
GROUP BY Symbol;
```

6.

```
SELECT
   u.Username,
SUM((COALESCE(s.CurrentPrice, 0) * p.Quantity) + (COALESCE(b.CurrentPrice, 0) *
p.Quantity) + COALESCE(c.Balance, 0)) AS TotalPortfolioValue
FROM Portfolio p
   JOIN User u ON p.UserID = u.UserID
   LEFT JOIN Stocks s ON p.StockID = s.StockID
   LEFT JOIN Bonds b ON p.BondID = b.BondID
   LEFT JOIN Cash c ON p.CashID = c.CashID
GROUP BY u.Username
ORDER BY TotalPortfolioValue DESC
LIMIT 1;
```