



UE22CS341A
Database Management System

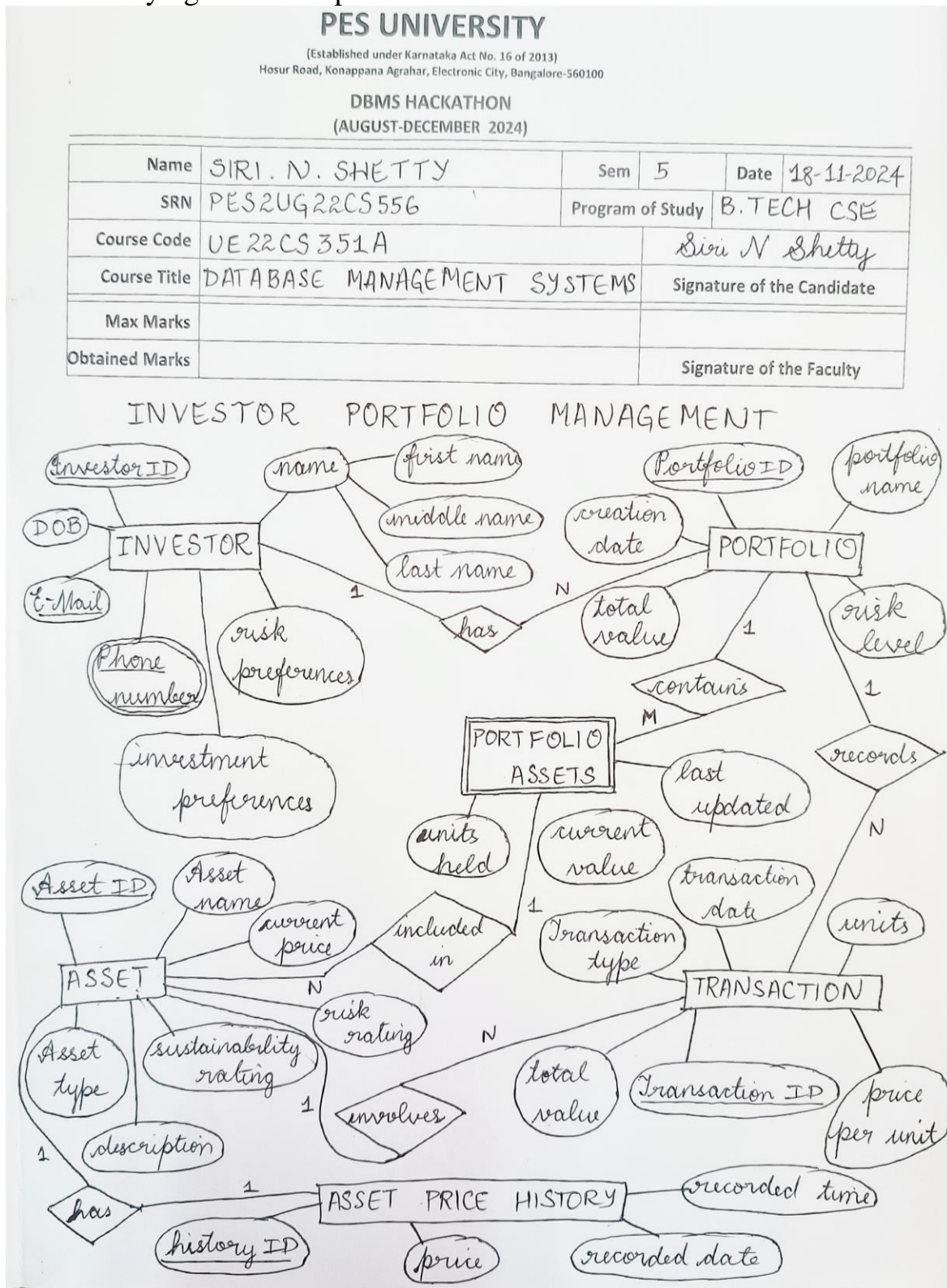
Investor Portfolio Management System
(Set-8 Version-1)

Prepared by:

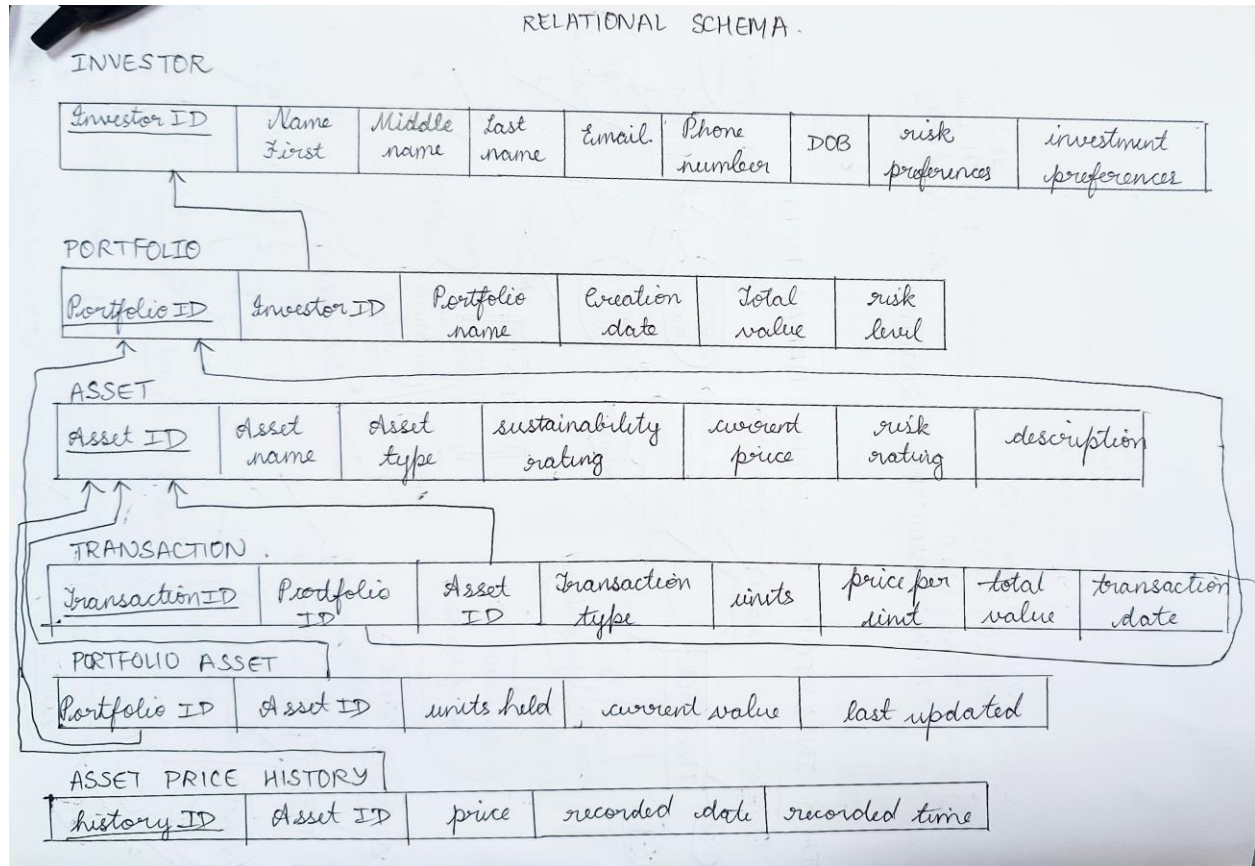
Siri N Shetty	PES2UG22CS556
Suprith S	PES2UG22CS600
Sri Vidya M	PES2UG22CS575
Soni Arya Rakeshkumar	PES2UG22CS568

Semester-5
Section-J

1. Draw an E-R Diagram for EcoVenture Investments that allows them to effectively manage their growing needs. Illustrate the various components in the system and their underlying relationships.



2. Convert the E-R Diagram to a relational schema depicting the constraints, providing a clear view of the system.



```
mysql> prompt 556_568_575_600>
PROMPT set to '556_568_575_600>'
```

3. Develop the database with all relevant tables based on the schema, ensuring proper relationships and constraints. Populate each table with at least 5 records. Attach screenshots showing the SQL queries for table creation and queries displaying the populated data.

```
556_568_575_600>CREATE DATABASE hackathon;
Query OK, 1 row affected (0.14 sec)

556_568_575_600>use hackathon;
Database changed
556_568_575_600>CREATE TABLE Investor ( investor_id INT AUTO_INCREMENT PRIMARY KEY, full_name VARCHAR(100) NOT NULL, email VARCHAR(150) UNIQUE NOT NULL, phone_number VARCHAR(20) UNIQUE NOT NULL, address TEXT, risk_tolerance ENUM('Low', 'Medium', 'High') DEFAULT 'Medium', created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP);
Query OK, 0 rows affected (0.07 sec)

556_568_575_600>CREATE TABLE Portfolio ( portfolio_id INT AUTO_INCREMENT PRIMARY KEY, investor_id INT NOT NULL, portfolio_name VARCHAR(100) NOT NULL, initial_investment DECIMAL(15, 2) NOT NULL, created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, status ENUM('Active', 'Inactive') DEFAULT 'Active', FOREIGN KEY (investor_id) REFERENCES Investor(investor_id) ON DELETE CASCADE);
Query OK, 0 rows affected (0.12 sec)

556_568_575_600>CREATE TABLE Asset ( asset_id INT AUTO_INCREMENT PRIMARY KEY, asset_name VARCHAR(100) NOT NULL, asset_type ENUM('Stock', 'Bond', 'ETF', 'Real Estate', 'Cryptocurrency') NOT NULL, market_value DECIMAL(15, 2) NOT NULL, risk_level ENUM('Low', 'Medium', 'High') NOT NULL, created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP);
Query OK, 0 rows affected (0.21 sec)

556_568_575_600>CREATE TABLE Transaction ( transaction_id INT AUTO_INCREMENT PRIMARY KEY, portfolio_id INT NOT NULL, asset_id INT NOT NULL, transaction_type ENUM('Buy', 'Sell') NOT NULL, transaction_date DATE NOT NULL, units DECIMAL(12, 4) NOT NULL, price_per_unit DECIMAL(15, 2) NOT NULL, total_value DECIMAL(15, 2) NOT NULL, FOREIGN KEY (portfolio_id) REFERENCES Portfolio(portfolio_id) ON DELETE CASCADE, FOREIGN KEY (asset_id) REFERENCES Asset(asset_id) ON DELETE CASCADE);
Query OK, 0 rows affected (0.20 sec)

556_568_575_600>CREATE TABLE Market_History ( history_id INT AUTO_INCREMENT PRIMARY KEY, asset_id INT NOT NULL, date DATE NOT NULL, market_value DECIMAL(15, 2) NOT NULL, FOREIGN KEY (asset_id) REFERENCES Asset(asset_id) ON DELETE CASCADE);
Query OK, 0 rows affected (0.24 sec)

556_568_575_600>CREATE TABLE Portfolio_Asset ( portfolio_id INT NOT NULL, asset_id INT NOT NULL, units_owned DECIMAL(12, 4) NOT NULL, purchase_price DECIMAL(15, 2) NOT NULL, current_value DECIMAL(15, 2) NOT NULL, last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP, PRIMARY KEY (portfolio_id, asset_id), FOREIGN KEY (portfolio_id) REFERENCES Portfolio(portfolio_id) ON DELETE CASCADE, FOREIGN KEY (asset_id) REFERENCES Asset(asset_id) ON DELETE CASCADE);
Query OK, 0 rows affected (0.24 sec)

556_568_575_600>show tables;
+-----+
| Tables_in_hackathon |
+-----+
| asset                |
| investor              |
| market_history        |
| portfolio              |
| portfolio_asset        |
| transaction            |
+-----+
6 rows in set (0.00 sec)
```

```
556_568_575_600>-- Insert records into Investor table
556_568_575_600>INSERT INTO Investor (full_name, email, phone_number, address, risk_tolerance) VALUES
-> ('Alice Johnson', 'alice.johnson@example.com', '1234567890', '123 Elm Street, NY', 'Low'),
-> ('Bob Smith', 'bob.smith@example.com', '0987654321', '456 Oak Avenue, CA', 'Medium'),
-> ('Charlie Brown', 'charlie.brown@example.com', '1122334455', '789 Pine Lane, TX', 'High'),
-> ('Diana Prince', 'diana.prince@example.com', '2233445566', '101 Maple Drive, FL', 'Medium'),
-> ('Edward Stark', 'edward.stark@example.com', '3344556677', '202 Birch Boulevard, WA', 'High');
```

```
Query OK, 5 rows affected (0.10 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
556_568_575_600>
```

```
556_568_575_600>-- Insert records into Portfolio table
```

```
556_568_575_600>INSERT INTO Portfolio (investor_id, portfolio_name, initial_investment) VALUES
-> (1, 'Retirement Fund', 50000.00),
-> (2, 'Growth Portfolio', 75000.00),
-> (3, 'High-Risk Portfolio', 100000.00),
-> (4, 'Diversified Portfolio', 60000.00),
-> (5, 'Crypto Focus', 45000.00);
```

```
Query OK, 5 rows affected (0.09 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
556_568_575_600>INSERT INTO Asset (asset_name, asset_type, market_value, risk_level) VALUES
```

```
-> ('Apple Inc.', 'Stock', 150.50, 'Low'),
-> ('US Treasury Bond', 'Bond', 102.75, 'Low'),
-> ('S&P 500 ETF', 'ETF', 420.30, 'Medium'),
-> ('Downtown Office', 'Real Estate', 500000.00, 'High'),
-> ('Bitcoin', 'Cryptocurrency', 27000.00, 'High');
```

```
Query OK, 5 rows affected (0.09 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
556_568_575_600>
```

```
556_568_575_600>-- Insert records into Portfolio_Asset table
```

```
556_568_575_600>INSERT INTO Portfolio_Asset (portfolio_id, asset_id, units_owned, purchase_price, current_value) VALUES
-> (1, 1, 100.00, 145.00, 150.50),
-> (1, 2, 50.00, 100.00, 102.75),
-> (2, 3, 75.00, 400.00, 420.30),
-> (3, 5, 1.50, 20000.00, 27000.00),
-> (4, 4, 0.10, 450000.00, 500000.00);
```

```
Query OK, 5 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
556_568_575_600>
```

```
556_568_575_600>-- Insert records into Transaction table
```

```
556_568_575_600>INSERT INTO Transaction (portfolio_id, asset_id, transaction_type, transaction_date, units, price_per_unit, total_value) VALUES
-> (1, 1, 'Buy', '2024-11-01', 100.00, 145.00, 14500.00),
-> (1, 2, 'Buy', '2024-11-02', 50.00, 100.00, 5000.00),
-> (2, 3, 'Buy', '2024-11-03', 75.00, 400.00, 30000.00),
-> (3, 5, 'Buy', '2024-11-04', 1.50, 20000.00, 30000.00),
-> (4, 4, 'Buy', '2024-11-05', 0.10, 450000.00, 45000.00);
```

```
Query OK, 5 rows affected (0.09 sec)
```

```
556_568_575_600>-- Insert records into Market_History table
```

```
556_568_575_600>INSERT INTO Market_History (asset_id, date, market_value) VALUES
-> (1, '2024-11-01', 145.00),
-> (1, '2024-11-15', 150.50),
-> (5, '2024-11-01', 20000.00),
-> (5, '2024-11-15', 27000.00),
-> (4, '2024-11-01', 450000.00);
```

```
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```



```

556_568_575_600>select * from Investor;
+-----+-----+-----+-----+-----+-----+-----+
| investor_id | full_name | email | phone_number | address | risk_tolerance | created_at |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice Johnson | alice.johnson@example.com | 1234567890 | 123 Elm Street, NY | Low | 2024-11-18 10:11:46 |
| 2 | Bob Smith | bob.smith@example.com | 0987654321 | 456 Oak Avenue, CA | Medium | 2024-11-18 10:11:46 |
| 3 | Charlie Brown | charlie.brown@example.com | 1122334455 | 789 Pine Lane, TX | High | 2024-11-18 10:11:46 |
| 4 | Diana Prince | diana.prince@example.com | 2233445566 | 101 Maple Drive, FL | Medium | 2024-11-18 10:11:46 |
| 5 | Edward Stark | edward.stark@example.com | 3344556677 | 202 Birch Boulevard, WA | High | 2024-11-18 10:11:46 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

556_568_575_600>select * from Portfolio;
+-----+-----+-----+-----+-----+-----+
| portfolio_id | investor_id | portfolio_name | initial_investment | created_at | status |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | Retirement Fund | 50000.00 | 2024-11-18 10:11:46 | Active |
| 2 | 2 | Growth Portfolio | 75000.00 | 2024-11-18 10:11:46 | Active |
| 3 | 3 | High-Risk Portfolio | 100000.00 | 2024-11-18 10:11:46 | Active |
| 4 | 4 | Diversified Portfolio | 60000.00 | 2024-11-18 10:11:46 | Active |
| 5 | 5 | Crypto Focus | 45000.00 | 2024-11-18 10:11:46 | Active |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

556_568_575_600>select * from Transaction;
+-----+-----+-----+-----+-----+-----+-----+-----+
| transaction_id | portfolio_id | asset_id | transaction_type | transaction_date | units | price_per_unit | total_value |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | Buy | 2024-11-01 | 100.0000 | 145.00 | 14500.00 |
| 2 | 1 | 1 | Buy | 2024-11-02 | 50.0000 | 100.00 | 5000.00 |
| 3 | 2 | 3 | Buy | 2024-11-03 | 75.0000 | 400.00 | 30000.00 |
| 4 | 3 | 5 | Buy | 2024-11-04 | 1.5000 | 20000.00 | 30000.00 |
| 5 | 4 | 4 | Buy | 2024-11-05 | 0.1000 | 450000.00 | 45000.00 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

556_568_575_600>select * from Asset;
+-----+-----+-----+-----+-----+-----+
| asset_id | asset_name | asset_type | market_value | risk_level | created_at |
+-----+-----+-----+-----+-----+-----+
| 1 | Apple Inc. | Stock | 150.50 | Low | 2024-11-18 10:11:47 |
| 2 | US Treasury Bond | Bond | 102.75 | Low | 2024-11-18 10:11:47 |
| 3 | S&P 500 ETF | ETF | 420.30 | Medium | 2024-11-18 10:11:47 |
| 4 | Downtown Office | Real Estate | 500000.00 | High | 2024-11-18 10:11:47 |
| 5 | Bitcoin | Cryptocurrency | 27000.00 | High | 2024-11-18 10:11:47 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

556_568_575_600>select * from Portfolio_Asset;
+-----+-----+-----+-----+-----+-----+
| portfolio_id | asset_id | units_owned | purchase_price | current_value | last_updated |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 100.0000 | 145.00 | 150.50 | 2024-11-18 10:11:47 |
| 1 | 2 | 50.0000 | 100.00 | 102.75 | 2024-11-18 10:11:47 |
| 2 | 3 | 75.0000 | 400.00 | 420.30 | 2024-11-18 10:11:47 |
| 3 | 5 | 1.5000 | 20000.00 | 27000.00 | 2024-11-18 10:11:47 |
| 4 | 4 | 0.1000 | 450000.00 | 500000.00 | 2024-11-18 10:11:47 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

556_568_575_600>select * from Market_History;
+-----+-----+-----+-----+
| history_id | asset_id | date | market_value |
+-----+-----+-----+-----+
| 1 | 1 | 2024-11-01 | 145.00 |
| 2 | 1 | 2024-11-15 | 150.50 |
| 3 | 5 | 2024-11-01 | 20000.00 |
| 4 | 5 | 2024-11-15 | 27000.00 |
| 5 | 4 | 2024-11-01 | 450000.00 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

4. Demonstrate the use of Federated Storage Engine in MySQL on the “asset” table created in your database, establishing a connection between the server and the client. You need to clean up the asset database by removing all the assets with risk level = “Low” to focus on Medium and High risk investments.

Write an SQL query to perform the above operation and attach relevant screenshots demonstrating the federated storage engine.

Remote

```
556_568_575_600->select*from asset;
+-----+-----+-----+-----+-----+-----+-----+
| asset_id | asset_name | asset_type | sustainability_rating | current_price | risk_rating | description |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Tesla Stock | Stock | A | 250.00 | High | Electric vehicle manufacturer stock |
| 2 | Bitcoin | Cryptocurrency | C | 45000.00 | High | Decentralized digital currency |
| 3 | Green Bond | Bond | A+ | 1000.00 | Low | Eco-friendly bond supporting green projects |
| 4 | Amazon Stock | Stock | B | 3300.00 | Medium | E-commerce and cloud computing stock |
| 5 | Solar Energy ETF | ETF | A+ | 150.00 | Low | ETF focused on solar energy companies |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

556_568_575_600->DELETE FROM ASSET WHERE risk_rating = 'Low';
Query OK, 2 rows affected (0.01 sec)
```

```
556_568_575_600->DELETE FROM ASSET WHERE risk_rating = 'Low';
Query OK, 2 rows affected (0.01 sec)

556_568_575_600->select*from asset;
+-----+-----+-----+-----+-----+-----+-----+
| asset_id | asset_name | asset_type | sustainability_rating | current_price | risk_rating | description |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Tesla Stock | Stock | A | 250.00 | High | Electric vehicle manufacturer stock |
| 2 | Bitcoin | Cryptocurrency | C | 45000.00 | High | Decentralized digital currency |
| 4 | Amazon Stock | Stock | B | 3300.00 | Medium | E-commerce and cloud computing stock |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Client:

```
MySQL 8.0 Command Line Client
556_568_575_600>create database dbmshack;
Query OK, 1 row affected (0.01 sec)

556_568_575_600>use dbmshack;
Database changed
556_568_575_600>CREATE TABLE INVESTOR (
->   investor_id INT PRIMARY KEY,
->   name VARCHAR(255),
->   email VARCHAR(255),
->   phone VARCHAR(20),
->   dob DATE,
->   risk_preference VARCHAR(100),
->   investment_preferences TEXT
-> ) ENGINE=FEDERATED
-> CONNECTION='mysql://fed-user:fed-pswd@10.1.18.48:3306/dbmshack/INVESTOR';
Query OK, 0 rows affected (0.01 sec)

556_568_575_600>
556_568_575_600>CREATE TABLE PORTFOLIO (
->   portfolio_id INT PRIMARY KEY,
->   investor_id INT,
->   portfolio_name VARCHAR(255),
->   creation_date DATE,
->   total_value DECIMAL(18, 2),
->   risk_level VARCHAR(100)
-> ) ENGINE=FEDERATED
-> CONNECTION='mysql://fed-user:fed-pswd@10.1.18.48:3306/dbmshack/PORTFOLIO';
Query OK, 0 rows affected (0.01 sec)

556_568_575_600>
556_568_575_600>CREATE TABLE ASSET (
->   asset_id INT PRIMARY KEY,
->   asset_name VARCHAR(255),
->   asset_type VARCHAR(100),
->   sustainability_rating VARCHAR(100),
->   current_price DECIMAL(18, 2),
->   risk_rating VARCHAR(100),
->   description TEXT
-> ) ENGINE=FEDERATED
-> CONNECTION='mysql://fed-user:fed-pswd@10.1.18.48:3306/dbmshack/ASSET';
Query OK, 0 rows affected (0.01 sec)
```


Select MySQL 8.0 Command Line Client

```
556_568_575_600>CREATE TABLE ASSET_PRICE_HISTORY (  
->     history_id INT PRIMARY KEY,  
->     asset_id INT,  
->     price DECIMAL(18, 2),  
->     recorded_date DATE,  
->     recorded_time TIME  
-> ) ENGINE=FEDERATED  
-> CONNECTION='mysql://fed-user:fed-pswd@10.1.18.48:3306/dbmshack/ASSET_PRICE_HISTORY';  
Query OK, 0 rows affected (0.01 sec)
```

556_568_575_600>

```
556_568_575_600>CREATE TABLE PORTFOLIO_ASSET (  
->     portfolio_id INT,  
->     asset_id INT,  
->     units_held DECIMAL(18, 2),  
->     current_value DECIMAL(18, 2),  
->     last_updated DATE,  
->     PRIMARY KEY (portfolio_id, asset_id),  
->     FOREIGN KEY (portfolio_id) REFERENCES PORTFOLIO(portfolio_id) ON DELETE CASCADE,  
->     FOREIGN KEY (asset_id) REFERENCES ASSET(asset_id) ON DELETE CASCADE  
-> ) ENGINE=FEDERATED  
-> CONNECTION='mysql://fed-user:fed-pswd@10.1.18.48:3306/dbmshack/PORTFOLIO_ASSET';  
Query OK, 0 rows affected (0.01 sec)
```

556_568_575_600>

```
556_568_575_600>CREATE TABLE TRANSACTION (  
->     transaction_id INT PRIMARY KEY,  
->     portfolio_id INT NOT NULL,  
->     asset_id INT NOT NULL,  
->     transaction_type VARCHAR(50) CHECK (transaction_type IN ('BUY', 'SELL')),  
->     units DECIMAL(18, 2),  
->     price_per_unit DECIMAL(18, 2),  
->     total_value DECIMAL(18, 2),  
->     transaction_date DATE NOT NULL,  
->     FOREIGN KEY (portfolio_id) REFERENCES PORTFOLIO(portfolio_id) ON DELETE CASCADE,  
->     FOREIGN KEY (asset_id) REFERENCES ASSET(asset_id) ON DELETE CASCADE  
-> ) ENGINE=FEDERATED  
-> CONNECTION='mysql://fed-user:fed-pswd@10.1.18.48:3306/dbmshack/TRANSACTIONS';  
Query OK, 0 rows affected (0.03 sec)
```

```

Select MySQL 8.0 Command Line Client

556_568_575_600>show tables;
+-----+
| Tables_in_dbmshack |
+-----+
| asset                |
| asset_price_history  |
| investor             |
| portfolio            |
| portfolio_asset      |
| transaction          |
+-----+
6 rows in set (0.01 sec)

556_568_575_600>select * from asset;
+-----+-----+-----+-----+-----+-----+-----+
| asset_id | asset_name | asset_type | sustainability_rating | current_price | risk_rating | description |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Tesla Stock | Stock | A | 250.00 | High | Electric vehicle manufacturer stock |
| 2 | Bitcoin | Cryptocurrency | C | 45000.00 | High | Decentralized digital currency |
| 3 | Green Bond | Bond | A+ | 1000.00 | Low | Eco-friendly bond supporting green projects |
| 4 | Amazon Stock | Stock | B | 3300.00 | Medium | E-commerce and cloud computing stock |
| 5 | Solar Energy ETF | ETF | A+ | 150.00 | Low | ETF focused on solar energy companies |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.99 sec)

556_568_575_600>select * from asset;
+-----+-----+-----+-----+-----+-----+-----+
| asset_id | asset_name | asset_type | sustainability_rating | current_price | risk_rating | description |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Tesla Stock | Stock | A | 250.00 | High | Electric vehicle manufacturer stock |
| 2 | Bitcoin | Cryptocurrency | C | 45000.00 | High | Decentralized digital currency |
| 4 | Amazon Stock | Stock | B | 3300.00 | Medium | E-commerce and cloud computing stock |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.09 sec)

556_568_575_600>

```

- Assuming that you have created (investor, portfolio, asset, portfolio_asset, transactions) as part of the database.

You have been tasked with analysing the growth of various assets owned by investors over a period of 20 years. Each asset has a fixed current value and grows at a specified annual compound interest rate.

Using the given asset values and a 5% annual growth rate, calculate the projected values of all assets after 20 years. The output should include the following details (columns) for each asset:

Investor's full name

Portfolio name

Asset name

Projected asset value after 20 years

full_name	portfolio_name	asset_name	projected_value
-----------	----------------	------------	-----------------

```

556_568_575_600>SELECT
->   inv.full_name AS "Investor's Full Name",
->   port.portfolio_name AS "Portfolio Name",
->   a.asset_name AS "Asset Name",
->   ROUND(pa.current_value * POWER(1 + 0.05, 20), 2) AS "Projected Asset Value After 20 Years"
-> FROM
->   Portfolio_Asset pa
-> JOIN
->   Portfolio port ON pa.portfolio_id = port.portfolio_id
-> JOIN
->   Investor inv ON port.investor_id = inv.investor_id
-> JOIN
->   Asset a ON pa.asset_id = a.asset_id;
+-----+-----+-----+-----+
| Investor's Full Name | Portfolio Name | Asset Name | Projected Asset Value After 20 Years |
+-----+-----+-----+-----+
| Alice Johnson       | Retirement Fund | Apple Inc. | 399.32 |
| Alice Johnson       | Retirement Fund | US Treasury Bond | 272.63 |
| Bob Smith           | Growth Portfolio | S&P 500 ETF | 1115.18 |
| Charlie Brown       | High-Risk Portfolio | Bitcoin | 71639.04 |
| Diana Prince        | Diversified Portfolio | Downtown Office | 1326648.85 |
+-----+-----+-----+-----+
5 rows in set (0.04 sec)

```

6. Assuming that you have created (investor, portfolio, asset, portfolio_asset, transactions) tables as part of the database.

Create a stored procedure that handles the process of transferring an asset from one portfolio to another. The procedure should:

- Deduct the asset units from the source portfolio.
- Add the asset units to the destination portfolio.
- Update the current values in both portfolios accordingly.
- Ensure that both portfolios belong to the same investor.

```

556_568_575_688->
556_568_575_688->CREATE PROCEDURE TransferAsset2(
->    IN source_portfolio_id INT, -- Source Portfolio ID
->    IN destination_portfolio_id INT, -- Destination Portfolio ID
->    IN asset_id INT, -- Asset ID to be transferred
->    IN transfer_units DECIMAL(18, 2) -- Number of units to transfer
-> )
-> BEGIN
->    DECLARE source_investor_id INT;
->    DECLARE destination_investor_id INT;
->    DECLARE current_price DECIMAL(18, 2);
->    DECLARE transfer_value DECIMAL(18, 2);
->    DECLARE source_units DECIMAL(18, 2);
->
->    -- 1. Verify that both portfolios belong to the same investor
->    SELECT investor_id INTO source_investor_id
->    FROM PORTFOLIO_3
->    WHERE portfolio_id = source_portfolio_id
->    LIMIT 1;
->
->    SELECT investor_id INTO destination_investor_id
->    FROM PORTFOLIO_3
->    WHERE portfolio_id = destination_portfolio_id
->    LIMIT 1;
->
->    IF source_investor_id != destination_investor_id THEN
->        SIGNAL SQLSTATE '45000'
->        SET MESSAGE_TEXT = 'Both portfolios must belong to the same investor.';
->    END IF;
->
->    -- 2. Check if the source portfolio has the asset and enough units of the asset
->    SELECT units_held INTO source_units
->    FROM PORTFOLIO_ASSET
->    WHERE portfolio_id = source_portfolio_id AND asset_id = asset_id
->    LIMIT 1;
->
->    IF source_units IS NULL THEN
->        SIGNAL SQLSTATE '45000'
->        SET MESSAGE_TEXT = 'The asset does not exist in the source portfolio.';
->    END IF;
->
->    IF source_units < transfer_units THEN
->        SIGNAL SQLSTATE '45000'
->        SET MESSAGE_TEXT = 'Insufficient units in the source portfolio.';
->    END IF;
->
->    -- 3. Get the current price of the asset
->    SELECT current_price INTO current_price
->    FROM ASSET
->    WHERE asset_id = asset_id
->    LIMIT 1;
->
->    -- 4. Calculate the transfer value
->    SET transfer_value = current_price * transfer_units;
->
->    -- 5. Deduct units from the source portfolio (only if the asset exists in this portfolio)
->    UPDATE PORTFOLIO_ASSET
->    SET units_held = units_held - transfer_units,
->        current_value = current_value - transfer_value
->    WHERE portfolio_id = source_portfolio_id AND asset_id = asset_id;
->
->    -- 6. Add units to the destination portfolio
->    INSERT INTO PORTFOLIO_ASSET (portfolio_id, asset_id, units_held, current_value, last_updated)
->    VALUES (destination_portfolio_id, asset_id, transfer_units, transfer_value, CURDATE())
->    ON DUPLICATE KEY UPDATE
->        units_held = units_held + transfer_units,
->        current_value = current_value + transfer_value,
->        last_updated = CURDATE();
->
->    -- 7. Update total values in both portfolios
->    UPDATE PORTFOLIO_3
->    SET total_value = total_value - transfer_value
->    WHERE portfolio_id = source_portfolio_id;
->
->    UPDATE PORTFOLIO_3
->    SET total_value = total_value + transfer_value
->    WHERE portfolio_id = destination_portfolio_id;
->
-> END$$
Query OK, 0 rows affected (0.01 sec)

```

```

556_568_575_600->select *from portfolio_asset;
+-----+-----+-----+-----+-----+
| portfolio_id | asset_id | units_held | current_value | last_updated |
+-----+-----+-----+-----+-----+
| 1 | 1 | 0.00 | NULL | 2023-12-01 |
| 2 | 1 | 10.00 | NULL | 2024-11-18 |
| 2 | 2 | 0.50 | 22500.00 | 2023-12-01 |
| 4 | 4 | 5.00 | 5000.00 | 2023-12-01 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

556_568_575_600->CALL TransferAsset(2, 1, 1, 10);
Query OK, 0 rows affected (0.01 sec)

556_568_575_600->select *from portfolio_asset;
+-----+-----+-----+-----+-----+
| portfolio_id | asset_id | units_held | current_value | last_updated |
+-----+-----+-----+-----+-----+
| 1 | 1 | 10.00 | NULL | 2024-11-18 |
| 2 | 1 | 0.00 | NULL | 2024-11-18 |

```

7. Assuming that you have created (investor, portfolio, asset, portfolio_asset, transactions) as part of the database.

Write a SQL trigger that will automatically update the current value of assets in the portfolio_asset table following any buy or sell transactions recorded in the transaction table.

Ensure that the trigger:

- Increases the current value of an asset when a 'Buy' transaction is recorded.
- Decreases the current value of an asset when a 'Sell' transaction is recorded.
- Updates the last_updated timestamp for the affected asset record in the portfolio_asset table immediately after the transaction is processed.

Create a front-end using streamlit demonstrating the above triggers execution and attach the relevant screenshots.

```

556_568_575_600>DELIMITER $$
556_568_575_600>
556_568_575_600>CREATE TRIGGER update_portfolio_asset
-> AFTER INSERT ON Transaction
-> FOR EACH ROW
-> BEGIN
->     DECLARE updated_value DECIMAL(15, 2);
->
->     -- Fetch the current value of the asset
->     SELECT current_value INTO updated_value
->     FROM Portfolio_Asset
->     WHERE portfolio_id = NEW.portfolio_id AND asset_id = NEW.asset_id;
->
->     -- Update the current value based on transaction type
->     IF NEW.transaction_type = 'Buy' THEN
->         UPDATE Portfolio_Asset
->         SET current_value = current_value + (NEW.units * NEW.price_per_unit),
->             last_updated = CURRENT_TIMESTAMP
->         WHERE portfolio_id = NEW.portfolio_id AND asset_id = NEW.asset_id;
->     ELSEIF NEW.transaction_type = 'Sell' THEN
->         UPDATE Portfolio_Asset
->         SET current_value = current_value - (NEW.units * NEW.price_per_unit),
->             last_updated = CURRENT_TIMESTAMP
->         WHERE portfolio_id = NEW.portfolio_id AND asset_id = NEW.asset_id;
->     END IF;
-> END$$
Query OK, 0 rows affected (0.02 sec)

```

Code to execute frontend in Streamlit:

```

import streamlit as st
import mysql.connector
import pandas as pd

from dotenv import load_dotenv
import os

# Load environment variables from .env file
load_dotenv()

# Database connection
def get_db_connection():
    return mysql.connector.connect(
        host="localhost",
        user="root", # Replace with your username
        password=os.getenv("DB_PASSWORD"), # Replace with your
password
        database="hackathon" # Replace with your database name
    )

```



```

# Function to display Portfolio_Asset table
def display_portfolio_assets():
    conn = get_db_connection()
    query = "SELECT * FROM Portfolio_Asset"
    df = pd.read_sql(query, conn)
    conn.close()
    return df

# Function to add a transaction
def add_transaction(portfolio_id, asset_id, transaction_type,
units, price_per_unit):
    conn = get_db_connection()
    cursor = conn.cursor()
    query = """
        INSERT INTO Transaction (portfolio_id, asset_id,
transaction_type, transaction_date, units, price_per_unit,
total_value)
        VALUES (%s, %s, %s, NOW(), %s, %s, %s)
    """
    total_value = units * price_per_unit
    cursor.execute(query, (portfolio_id, asset_id,
transaction_type, units, price_per_unit, total_value))
    conn.commit()
    conn.close()

# Streamlit UI
st.title("Portfolio Asset Management")
st.write("Demonstrating SQL Trigger Execution")

# Display Portfolio_Asset table
st.header("Portfolio Assets")
st.write("Current state of Portfolio_Asset table:")
df = display_portfolio_assets()
st.dataframe(df)

# Add a transaction
st.header("Add a Transaction")
portfolio_id = st.number_input("Portfolio ID", min_value=1,
step=1)
asset_id = st.number_input("Asset ID", min_value=1, step=1)
transaction_type = st.selectbox("Transaction Type", ["Buy",
"Sell"])
units = st.number_input("Units", min_value=0.01, step=0.01,
format="%.2f")

```

```

price_per_unit = st.number_input("Price Per Unit",
min_value=0.01, step=0.01, format="%.2f")

if st.button("Add Transaction"):
    add_transaction(portfolio_id, asset_id, transaction_type,
units, price_per_unit)
    st.success("Transaction added successfully!")
    st.write("Updated Portfolio_Asset table:")
    df = display_portfolio_assets()
    st.dataframe(df)

```

Output:

Portfolio Asset Management

Demonstrating SQL Trigger Execution

Portfolio Assets

Current state of Portfolio_Asset table:

	portfolio_id	asset_id	units_owned	purchase_price	current_value	last_updated
0	1	1	100	145	150.5	2024-11-18 10:11:47
1	1	2	50	100	102.75	2024-11-18 10:11:47
2	2	3	75	400	420.3	2024-11-18 10:11:47
3	3	5	1.5	20,000	27,000	2024-11-18 10:11:47
4	4	4	0.1	450,000	500,000	2024-11-18 10:11:47
5	6	1	50	140	150.5	2024-11-18 11:14:58

Add a Transaction

Portfolio ID

1

-

+

Asset ID

1

-

+

Transaction Type

Buy

▼

Units

10.00

-

+

Price Per Unit

100.00

-

+

Add Transaction

Add Transaction

Transaction added successfully!

Updated Portfolio_Asset table:

	portfolio_id	asset_id	units_owned	purchase_price	current_value	last_updated
0	1	1	100	145	1,150.5	2024-11-18 11:33:05
1	1	2	50	100	102.75	2024-11-18 10:11:47
2	2	3	75	400	420.3	2024-11-18 10:11:47
3	3	5	1.5	20,000	27,000	2024-11-18 10:11:47
4	4	4	0.1	450,000	500,000	2024-11-18 10:11:47
5	6	1	50	140	150.5	2024-11-18 11:14:58

8. Assuming that you have created (investor, portfolio, asset, portfolio_asset, transactions) as part of the database.

EcoVenture Investments wants to project the future value of assets within investor portfolios, accounting for a constant annual growth rate. You are required to implement a function that takes the asset's current value, interest rate, and number of years as input and returns the projected future value.

- Create a **user-defined SQL function** calculate_future_value that takes:
 1. current_value (DECIMAL) – the current market value of the asset from the portfolio_asset table.
 2. interest_rate (DECIMAL) – a constant annual growth rate.
 3. years (INT) – the number of years over which the asset value is projected.
- The function should return the **projected future value** using the formula:
$$\text{Future Value} = \text{Current Value} * (1 + \text{interest_rate}) ^ \text{years}$$
- Use the function to display for each asset:
 - Investor's full name (from the investor table)
 - Portfolio name (from the portfolio table)
 - Asset name (from the asset table)
 - Current asset value (from the portfolio_asset table)
 - Projected asset value after 10 years (using the created function)

```

556_568_575_600>
556_568_575_600>DELIMITER ;
556_568_575_600>DELIMITER $$
556_568_575_600>
556_568_575_600>CREATE FUNCTION calculate_future_value(current_value DECIMAL, interest_rate DECIMAL, years INT)
-> RETURNS DECIMAL
-> DETERMINISTIC
-> BEGIN
->   DECLARE future_value DECIMAL;
->   SET future_value = current_value * POWER(1 + interest_rate, years);
->   RETURN future_value;
-> END$$

```

Query OK, 0 rows affected (0.05 sec)

```

556_568_575_600>
556_568_575_600>DELIMITER ;
556_568_575_600>SELECT
->   i.full_name AS investor_name,
->   p.portfolio_name,
->   a.asset_name,
->   pa.current_value AS current_asset_value,
->   calculate_future_value(pa.current_value, 0.05, 10) AS projected_asset_value_10_years
-> FROM
->   Portfolio_Asset pa
-> JOIN
->   Portfolio p ON pa.portfolio_id = p.portfolio_id
-> JOIN
->   Investor i ON p.investor_id = i.investor_id
-> JOIN
->   Asset a ON pa.asset_id = a.asset_id;

```

investor_name	portfolio_name	asset_name	current_asset_value	projected_asset_value_10_years
Alice Johnson	Retirement Fund	Apple Inc.	1150.50	1151
Alice Johnson	Tech Growth	Apple Inc.	150.50	151
Alice Johnson	Retirement Fund	US Treasury Bond	102.75	103
Bob Smith	Growth Portfolio	S&P 500 ETF	420.30	420
Diana Prince	Diversified Portfolio	Downtown Office	500000.00	500000
Charlie Brown	High-Risk Portfolio	Bitcoin	27000.00	27000

6 rows in set, 10 warnings (0.26 sec)

9. Demonstrate Transaction Isolation and identify its type by implementing the following Create a new portfolio for an investor with an initial investment of ₹5000 under the name 'Retirement Fund'. Start a transaction and update the portfolio by adding ₹1000 to the initial investment. In a separate session, attempt to update the same portfolio by adding ₹500. Please submit screenshots of the portfolio insertion, the transaction and the update of ₹1000, as well as the error received when trying to update from the separate session.

```
556_568_575_600>select * from Portfolio;
+-----+-----+-----+-----+-----+-----+
| portfolio_id | investor_id | portfolio_name | initial_investment | created_at | status |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | Retirement Fund | 51500.00 | 2024-11-18 10:11:46 | Active |
| 2 | 2 | Growth Portfolio | 75000.00 | 2024-11-18 10:11:46 | Active |
| 3 | 3 | High-Risk Portfolio | 100000.00 | 2024-11-18 10:11:46 | Active |
| 4 | 4 | Diversified Portfolio | 60000.00 | 2024-11-18 10:11:46 | Active |
| 5 | 5 | Crypto Focus | 45000.00 | 2024-11-18 10:11:46 | Active |
| 6 | 1 | Tech Growth | 30000.00 | 2024-11-18 11:14:51 | Active |
| 7 | 1 | Tech Growth | 30000.00 | 2024-11-18 11:16:56 | Active |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

556_568_575_600>INSERT INTO Portfolio (investor_id, portfolio_name, initial_investment)VALUES (1, 'Retirement Fund', 5000);
Query OK, 1 row affected (0.04 sec)

556_568_575_600>select * from Portfolio;
+-----+-----+-----+-----+-----+-----+
| portfolio_id | investor_id | portfolio_name | initial_investment | created_at | status |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | Retirement Fund | 51500.00 | 2024-11-18 10:11:46 | Active |
| 2 | 2 | Growth Portfolio | 75000.00 | 2024-11-18 10:11:46 | Active |
| 3 | 3 | High-Risk Portfolio | 100000.00 | 2024-11-18 10:11:46 | Active |
| 4 | 4 | Diversified Portfolio | 60000.00 | 2024-11-18 10:11:46 | Active |
| 5 | 5 | Crypto Focus | 45000.00 | 2024-11-18 10:11:46 | Active |
| 6 | 1 | Tech Growth | 30000.00 | 2024-11-18 11:14:51 | Active |
| 7 | 1 | Tech Growth | 30000.00 | 2024-11-18 11:16:56 | Active |
| 9 | 1 | Retirement Fund | 5000.00 | 2024-11-18 11:44:24 | Active |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```



```
556_568_575_600>select * from Portfolio;
```

portfolio_id	investor_id	portfolio_name	initial_investment	created_at	status
1	1	Retirement Fund	51500.00	2024-11-18 10:11:46	Active
2	2	Growth Portfolio	75000.00	2024-11-18 10:11:46	Active
3	3	High-Risk Portfolio	100000.00	2024-11-18 10:11:46	Active
4	4	Diversified Portfolio	60000.00	2024-11-18 10:11:46	Active
5	5	Crypto Focus	45000.00	2024-11-18 10:11:46	Active
6	1	Tech Growth	30000.00	2024-11-18 11:14:51	Active
7	1	Tech Growth	30000.00	2024-11-18 11:16:56	Active
9	1	Retirement Fund	5000.00	2024-11-18 11:44:24	Active

```
8 rows in set (0.00 sec)
```

```
556_568_575_600>START TRANSACTION;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
556_568_575_600>UPDATE Portfolio
```

```
-> SET initial_investment = initial_investment + 1000
```

```
-> WHERE portfolio_name = 'Retirement Fund';
```

```
Query OK, 2 rows affected (0.00 sec)
```

```
Rows matched: 2 Changed: 2 Warnings: 0
```

```
556_568_575_600>select * from Portfolio;
```

portfolio_id	investor_id	portfolio_name	initial_investment	created_at	status
1	1	Retirement Fund	52500.00	2024-11-18 10:11:46	Active
2	2	Growth Portfolio	75000.00	2024-11-18 10:11:46	Active
3	3	High-Risk Portfolio	100000.00	2024-11-18 10:11:46	Active
4	4	Diversified Portfolio	60000.00	2024-11-18 10:11:46	Active
5	5	Crypto Focus	45000.00	2024-11-18 10:11:46	Active
6	1	Tech Growth	30000.00	2024-11-18 11:14:51	Active
7	1	Tech Growth	30000.00	2024-11-18 11:16:56	Active
9	1	Retirement Fund	6000.00	2024-11-18 11:44:24	Active

```
556_568_575_600>UPDATE Portfolio
```

```
-> SET initial_investment = initial_investment + 500
```

```
-> WHERE portfolio_name = 'Retirement Fund';
```

```
Query OK, 2 rows affected (0.00 sec)
```

```
Rows matched: 2 Changed: 2 Warnings: 0
```

```
556_568_575_600>select * from Portfolio;
```

portfolio_id	investor_id	portfolio_name	initial_investment	created_at	status
1	1	Retirement Fund	53000.00	2024-11-18 10:11:46	Active
2	2	Growth Portfolio	75000.00	2024-11-18 10:11:46	Active
3	3	High-Risk Portfolio	100000.00	2024-11-18 10:11:46	Active
4	4	Diversified Portfolio	60000.00	2024-11-18 10:11:46	Active
5	5	Crypto Focus	45000.00	2024-11-18 10:11:46	Active
6	1	Tech Growth	30000.00	2024-11-18 11:14:51	Active
7	1	Tech Growth	30000.00	2024-11-18 11:16:56	Active
9	1	Retirement Fund	6500.00	2024-11-18 11:44:24	Active

```
8 rows in set (0.00 sec)
```