README for the Stock Portfolio MYSQL Database Management Script

Team Formation
Team name: HealthEdBot
Names & USC ID:
Cici Chang (8706354957)
Daoyang Li (7168949829)
Yifan Yang(8386626967)
https://github.com/DSCI560

Introduction

This Python script is designed for managing stock portfolios through interactions with a MySQL database and fetching real-time stock data using the yfinance library. It enables users to add and remove stocks from portfolios, display portfolio contents, and ensure the uniqueness of portfolio names.

Requirements

- Linux ubuntu
- Python 3.x
- MySQL Database
- Libraries: yfinance, mysql.connector

Install the necessary Python libraries using pip:

```
pip install yfinance mysql-connector-python
```

Setup

MySQL Database Configuration:

Ensure the MySQL database server is running.
Create a database named stock_portfolio and the tables (portfolio, stocks).

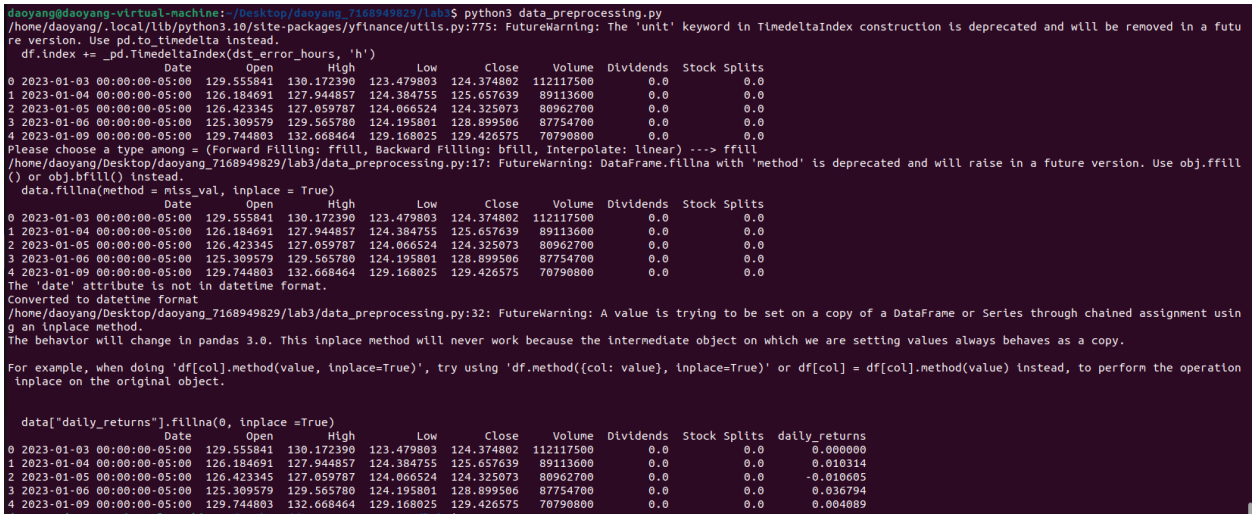
```
mysql> USE stock_database;
Database changed
mysql> CREATE TABLE IF NOT EXISTS portfolios(
    -> id INT AUTO_INCREMENT PRIMARY KEY,
    -> name VARCHAR(255) NOT NULL,
    -> creation_date DATE NOT NULL
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> SHOW TABLES;
+--------------------------+
| Tables_in_stock_database |
+--------------------------+
| portfolios               |
+--------------------------+
1 row in set (0.00 sec)

mysql> CREATE TABLE IF NOT EXISTS stocks(
    -> id INT AUTO_INCREMENT PRIMARY KEY,
    -> symbol VARCHAR(10) NOT NULL,
    -> date DATE NOT NULL,
    -> open DECIMAL(10,2),
    -> high DECIMAL(10,2),
    -> low DECIMAL(10,2),
    -> close DECIMAL(10,2),
    -> dividends DECIMAL(10,2),
    -> portfolio_id INT,
    -> FOREIGN KEY (portfolio_id) REFERENCES portfolios(id) ON DELETE CASCADE);
Query OK, 0 rows affected (0.03 sec)

mysql> SHOW TABLES;
+--------------------------+
| Tables_in_stock_database |
+--------------------------+
| portfolios               |
| stocks                   |
+--------------------------+
2 rows in set (0.00 sec)

mysql>
```

Database Connection:

Establish the database connection parameters (host, user, password, database) in the Python script to match MySQL server credentials.

```
daoyang@daoyang-virtual-machine:~/Desktop/daoyang_7168949829/lab3$ python3 stock.py
(1, 'apple', datetime.date(2023, 8, 20))
(2, '[value-2]', datetime.date(2023, 1, 31))
(3, 'tech', datetime.date(2024, 2, 1))
(4, 'tech', datetime.date(2024, 2, 1))
(5, 'new', datetime.date(2024, 2, 1))
/home/daoyang/.local/lib/python3.10/site-packages/yfinance/utils.py:775: FutureWarni
re version. Use pd.to_timedelta instead.
  df.index += _pd.TimedeltaIndex(dst_error_hours, 'h')
                                Open        High  ...  Dividends  Stock Splits
Date                                              ...
2023-01-03 00:00:00-05:00  129.555841  130.172390  ...        0.0           0.0
2023-01-04 00:00:00-05:00  126.184691  127.944857  ...        0.0           0.0
2023-01-05 00:00:00-05:00  126.423353  127.059795  ...        0.0           0.0
2023-01-06 00:00:00-05:00  125.309594  129.565795  ...        0.0           0.0
2023-01-09 00:00:00-05:00  129.744788  132.668449  ...        0.0           0.0

[5 rows x 7 columns]
Stock added successfully
Stock added successfully
Stock added successfully
Stock added successfully
Stock added successfully
Stock added successfully
Stock added successfully
Stock added successfully
Stock added successfully
Stock added successfully
```

```
Portfolio ID: 1, Creation Date: apple
Stocks included:

Portfolio ID: 2, Creation Date: [value-2]
Stocks included:

Portfolio ID: 3, Creation Date: tech
Stocks included: AAPL, MSFT, AMZN, GOOGL, JPM, JNJ, KO, BA, NVDA, AAPL

Portfolio ID: 4, Creation Date: tech
Stocks included:

Portfolio ID: 5, Creation Date: new
Stocks included: AAPL, MSFT, AMZN, GOOGL, TSLA, JPM, JNJ, KO, BA, NVDA, AAPL, MSFT, AMZN, GOOGL,

Database error: Unread result found
Portfolio ID: 1, Creation Date: apple
Stocks included:

Portfolio ID: 2, Creation Date: [value-2]
Stocks included:

Portfolio ID: 3, Creation Date: tech
Stocks included: AAPL, MSFT, AMZN, GOOGL, JPM, JNJ, KO, BA, NVDA, AAPL

Portfolio ID: 4, Creation Date: tech
Stocks included:

Portfolio ID: 5, Creation Date: new
Stocks included: AAPL, MSFT, AMZN, GOOGL, TSLA, JPM, JNJ, KO, BA, NVDA, AAPL, MSFT, AMZN, GOOGL,
```



Usage

Run the python script ‘stock.py’ directly through the terminal. The script includes a test function **test_portfolio_management()** demonstrating the functionalities, such as adding stocks to a portfolio, removing a stock, and displaying portfolios.

Run the python script ‘data_preprocessing.py’ directly through the terminal.
The script consists of a function `fetch_stock_data` that downloads stock data for a given symbol between specified start and end dates. It then processes this data by filling missing values as specified by the user, ensures the date column is in datetime format, calculates daily returns, and saves the processed data to a CSV file.

Main Functions

- **fetch_stock_data(stock_symbol, start_date, end_date):** Fetches historical data for a specified stock symbol from Yahoo Finance.
- **check_stock_validity(stock_symbol):** Validates the existence of a stock symbol in the market data.
- **add_portfolio(portfolio_name, creation_date):** Adds a new portfolio to the database if the name is unique.
- **add_stock_to_portfolio(portfolio_name, stock_symbol):** Adds a stock to an existing portfolio after validating the stock symbol and checking if the portfolio exists.
- **remove_stock_from_portfolio(portfolio_name, stock_symbol):** Removes a stock from a specified portfolio.
- **display_portfolios():** Displays all portfolios and their associated stocks.