CS306 Project Phase III Bartu Sisman 28038

Using the two tables created in the second phase of the project, I developed a Python application that provides a user interface to perform CRUD (Create, Read, Update, Delete) operations on these tables. The implementation adheres to the atomicity principle by keeping each function separate and straightforward, ensuring clarity, simplicity, and maintainability.

2. Tables and Data

Company Table

Attributes:

- **symbol**: VARCHAR(10) PRIMARY KEY
- **name**: VARCHAR(100) NOT NULL
- **outstanding_shares**: BIGINT NOT NULL

Data:

```
INSERT INTO Company (Symbol, Name, Outstanding_Shares) VALUES

('AAPL', 'Apple Inc.', 4371000000),

('GOOGL', 'Alphabet Inc.', 300900000),

('MSFT', 'Microsoft Corp.', 7603000000),

('AMZN', 'Amazon.com Inc.', 504000000),

('FB', 'Meta Platforms Inc.', 2400000000),

('TSLA', 'Tesla Inc.', 1000000000),

('BRK.A', 'Berkshire Hathaway Inc.', 1382000),

('V', 'Visa Inc.', 2107000000),

('JNJ', 'Johnson & Johnson', 2601000000),

('WMT', 'Walmart Inc.', 2800000000);
```

Stock Table

Attributes:

- **symbol**: VARCHAR(10)
- **date**: DATE
- **highest**: FLOAT
- **lowest**: FLOAT
- **opening**: FLOAT
- **closing**: FLOAT
- **volume**: BIGINT
- PRIMARY KEY (**symbol**, **date**)
- FOREIGN KEY (**symbol**) REFERENCES **Company**(**symbol**)

Data:

```
INSERT INTO Stock (Symbol, Date, Highest, Lowest, Opening, Closing, Volume) VALUES
('AAPL', '2023-10-01', 150.25, 147.30, 148.00, 149.55, 90000000),
('AAPL', '2023-10-02', 152.00, 148.20, 149.80, 150.70, 88000000),
('GOOGL', '2023-10-01', 120.50, 117.75, 118.00, 119.50, 42000000),
('MSFT', '2023-10-01', 305.20, 300.10, 301.50, 304.00, 78000000),
('AMZN', '2023-10-01', 125.80, 123.00, 124.30, 125.40, 50000000),
('TSLA', '2023-10-01', 900.00, 880.00, 890.00, 895.00, 80000000),
('FB', '2023-10-01', 310.00, 305.00, 307.50, 309.45, 60000000),
('BRK.A', '2023-10-01', 450000.00, 440000.00, 445000.00, 448000.00, 500),
('V', '2023-10-01', 230.00, 225.00, 228.00, 229.50, 45000000),
('JNJ', '2023-10-01', 175.00, 170.00, 171.00, 174.50, 40000000);
```

Insert Operations

Insert into Company Table:

- **Query**: **INSERT INTO Company (symbol, name, outstanding_shares) VALUES (%s, %s, %s)**
- **Generated Query**:

  cursor.execute("INSERT INTO Company (symbol, name, outstanding_shares) VALUES (%s, %s, %s)", ('NFLX', 'Netflix Inc.', 450000000))

  cursor.execute("INSERT INTO Company (symbol, name, outstanding_shares) VALUES (%s, %s, %s)", ('DIS', 'Walt Disney Co.', 1820000000))

Insert into Stock Table:

- **Query**: **INSERT INTO Stock (symbol, date, highest, lowest, opening, closing, volume) VALUES (%s, %s, %s, %s, %s, %s, %s)**
- **Generated Query**:

  cursor.execute("INSERT INTO Stock (symbol, date, highest, lowest, opening, closing, volume) VALUES (%s, %s, %s, %s, %s, %s, %s)", ('NFLX', '2023-10-01', 490.25, 485.30, 487.00, 489.55, 1000000))

  cursor.execute("INSERT INTO Stock (symbol, date, highest, lowest, opening, closing, volume) VALUES (%s, %s, %s, %s, %s, %s, %s)", ('DIS', '2023-10-01', 135.50, 130.75, 132.00, 134.50, 5000000))

Select Operations

Select from Company Table:

- **Query**: **SELECT * FROM Company**
- **Generated Query**: cursor.execute("SELECT * FROM Company")

Select from Stock Table:

- **Query**: **SELECT * FROM Stock**
- **Generated Query**: cursor.execute("SELECT * FROM Stock")

Update Operations

Update Company Table:

- **Query**: **UPDATE Company SET name = %s, outstanding_shares = %s WHERE symbol = %s**
- **Generated Query**:

  ```
  cursor.execute("UPDATE Company SET name = %s, outstanding_shares = %s WHERE symbol = %s", ('Netflix Corporation', 460000000, 'NFLX'))
  ```

Update Stock Table:

- **Query**: **UPDATE Stock SET highest = %s, lowest = %s, opening = %s, closing = %s, volume = %s WHERE symbol = %s AND date = %s**
- **Generated Query**:

  ```
  cursor.execute("UPDATE Stock SET closing = %s WHERE symbol = %s AND date = %s", (490.00, 'NFLX', '2023-10-01'))
  ```

Delete Operations

Delete from Company Table:

- **Query**: **DELETE FROM Company WHERE symbol = %s**
- **Generated Query**:

  ```
  cursor.execute("DELETE FROM Company WHERE symbol = %s", ('NFLX',))
  ```

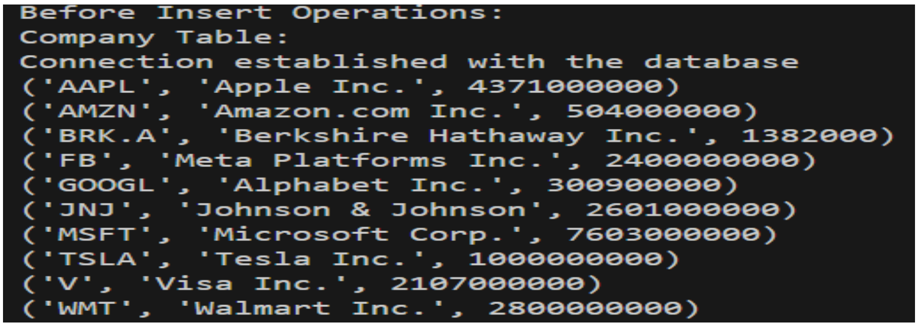
Delete from Stock Table:

- **Query**: **DELETE FROM Stock WHERE symbol = %s AND date = %s**
- **Generated Query**:

  ```
  cursor.execute("DELETE FROM Stock WHERE symbol = %s AND date = %s", ('NFLX', '2023-10-01'))
  ```

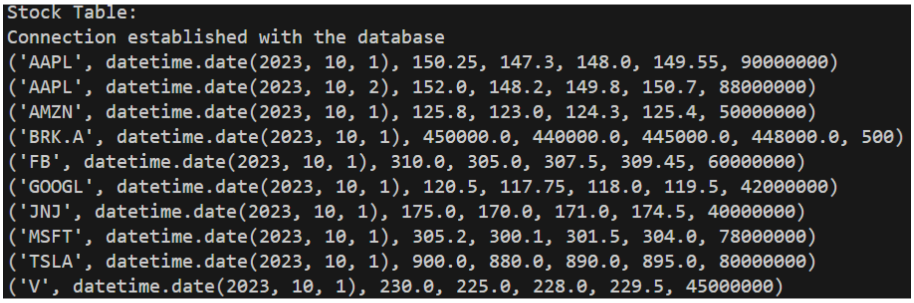
Before Insert Operations

Company Table:



```
Before Insert Operations:
Company Table:
Connection established with the database
('AAPL', 'Apple Inc.', 4371000000)
('AMZN', 'Amazon.com Inc.', 504000000)
('BRK.A', 'Berkshire Hathaway Inc.', 1382000)
('FB', 'Meta Platforms Inc.', 2400000000)
('GOOGL', 'Alphabet Inc.', 300900000)
('JNJ', 'Johnson & Johnson', 2601000000)
('MSFT', 'Microsoft Corp.', 7603000000)
('TSLA', 'Tesla Inc.', 1000000000)
('V', 'Visa Inc.', 2107000000)
('WMT', 'Walmart Inc.', 2800000000)
```

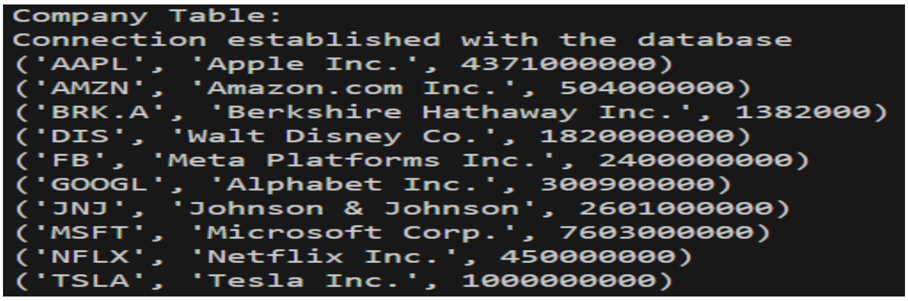
Stock Table:



```
Stock Table:
Connection established with the database
('AAPL', datetime.date(2023, 10, 1), 150.25, 147.3, 148.0, 149.55, 90000000)
('AAPL', datetime.date(2023, 10, 2), 152.0, 148.2, 149.8, 150.7, 88000000)
('AMZN', datetime.date(2023, 10, 1), 125.8, 123.0, 124.3, 125.4, 50000000)
('BRK.A', datetime.date(2023, 10, 1), 450000.0, 440000.0, 445000.0, 448000.0, 500)
('FB', datetime.date(2023, 10, 1), 310.0, 305.0, 307.5, 309.45, 60000000)
('GOOGL', datetime.date(2023, 10, 1), 120.5, 117.75, 118.0, 119.5, 42000000)
('JNJ', datetime.date(2023, 10, 1), 175.0, 170.0, 171.0, 174.5, 40000000)
('MSFT', datetime.date(2023, 10, 1), 305.2, 300.1, 301.5, 304.0, 78000000)
('TSLA', datetime.date(2023, 10, 1), 900.0, 880.0, 890.0, 895.0, 80000000)
('V', datetime.date(2023, 10, 1), 230.0, 225.0, 228.0, 229.5, 45000000)
```

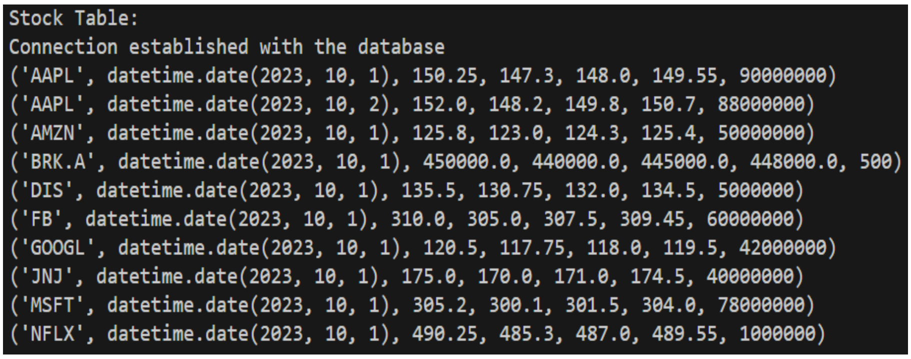
After Insert Operations

Company Table:



```
Company Table:
Connection established with the database
('AAPL', 'Apple Inc.', 4371000000)
('AMZN', 'Amazon.com Inc.', 504000000)
('BRK.A', 'Berkshire Hathaway Inc.', 1382000)
('DIS', 'Walt Disney Co.', 1820000000)
('FB', 'Meta Platforms Inc.', 2400000000)
('GOOGL', 'Alphabet Inc.', 300900000)
('JNJ', 'Johnson & Johnson', 2601000000)
('MSFT', 'Microsoft Corp.', 7603000000)
('NFLX', 'Netflix Inc.', 450000000)
('TSLA', 'Tesla Inc.', 1000000000)
```

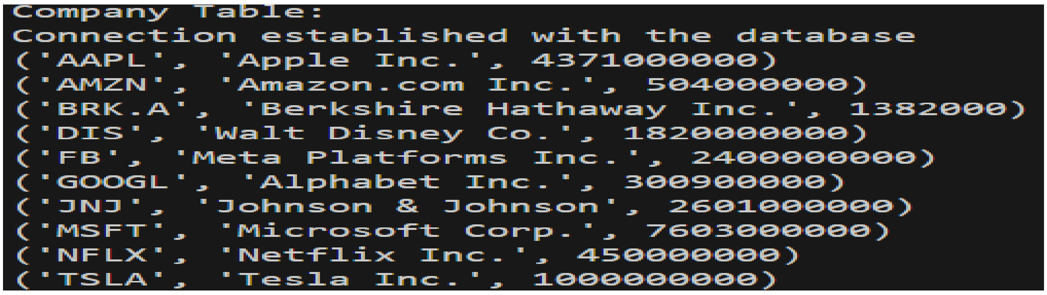
Stock Table:



```
Stock Table:
Connection established with the database
('AAPL', datetime.date(2023, 10, 1), 150.25, 147.3, 148.0, 149.55, 90000000)
('AAPL', datetime.date(2023, 10, 2), 152.0, 148.2, 149.8, 150.7, 88000000)
('AMZN', datetime.date(2023, 10, 1), 125.8, 123.0, 124.3, 125.4, 50000000)
('BRK.A', datetime.date(2023, 10, 1), 450000.0, 440000.0, 445000.0, 448000.0, 500)
('DIS', datetime.date(2023, 10, 1), 135.5, 130.75, 132.0, 134.5, 5000000)
('FB', datetime.date(2023, 10, 1), 310.0, 305.0, 307.5, 309.45, 60000000)
('GOOGL', datetime.date(2023, 10, 1), 120.5, 117.75, 118.0, 119.5, 42000000)
('JNJ', datetime.date(2023, 10, 1), 175.0, 170.0, 171.0, 174.5, 40000000)
('MSFT', datetime.date(2023, 10, 1), 305.2, 300.1, 301.5, 304.0, 78000000)
('NFLX', datetime.date(2023, 10, 1), 490.25, 485.3, 487.0, 489.55, 1000000)
```

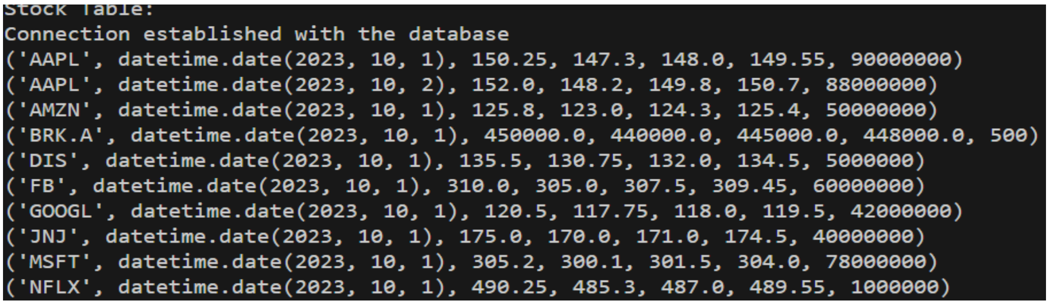
Before Update Operations

Company Table:



```
Company Table:
Connection established with the database
('AAPL', 'Apple Inc.', 4371000000)
('AMZN', 'Amazon.com Inc.', 504000000)
('BRK.A', 'Berkshire Hathaway Inc.', 1382000)
('DIS', 'Walt Disney Co.', 1820000000)
('FB', 'Meta Platforms Inc.', 2400000000)
('GOOGL', 'Alphabet Inc.', 300900000)
('JNJ', 'Johnson & Johnson', 2601000000)
('MSFT', 'Microsoft Corp.', 7603000000)
('NFLX', 'Netflix Inc.', 450000000)
('TSLA', 'Tesla Inc.', 1000000000)
```

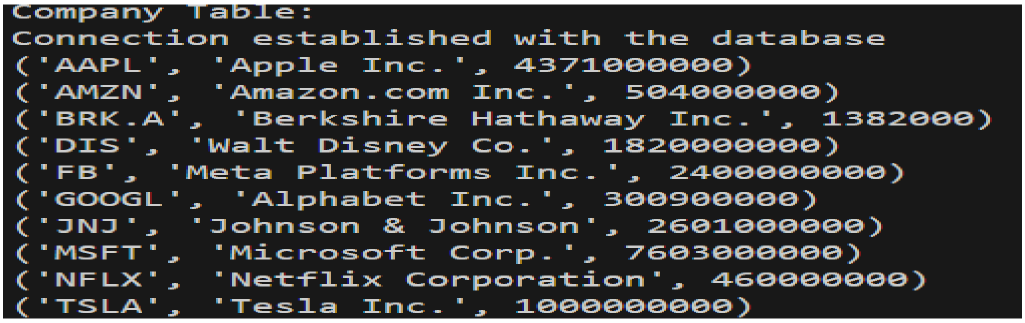
Stock Table:



```
Stock Table:
Connection established with the database
('AAPL', datetime.date(2023, 10, 1), 150.25, 147.3, 148.0, 149.55, 90000000)
('AAPL', datetime.date(2023, 10, 2), 152.0, 148.2, 149.8, 150.7, 88000000)
('AMZN', datetime.date(2023, 10, 1), 125.8, 123.0, 124.3, 125.4, 50000000)
('BRK.A', datetime.date(2023, 10, 1), 450000.0, 440000.0, 445000.0, 448000.0, 500)
('DIS', datetime.date(2023, 10, 1), 135.5, 130.75, 132.0, 134.5, 5000000)
('FB', datetime.date(2023, 10, 1), 310.0, 305.0, 307.5, 309.45, 60000000)
('GOOGL', datetime.date(2023, 10, 1), 120.5, 117.75, 118.0, 119.5, 42000000)
('JNJ', datetime.date(2023, 10, 1), 175.0, 170.0, 171.0, 174.5, 40000000)
('MSFT', datetime.date(2023, 10, 1), 305.2, 300.1, 301.5, 304.0, 78000000)
('NFLX', datetime.date(2023, 10, 1), 490.25, 485.3, 487.0, 489.55, 1000000)
```

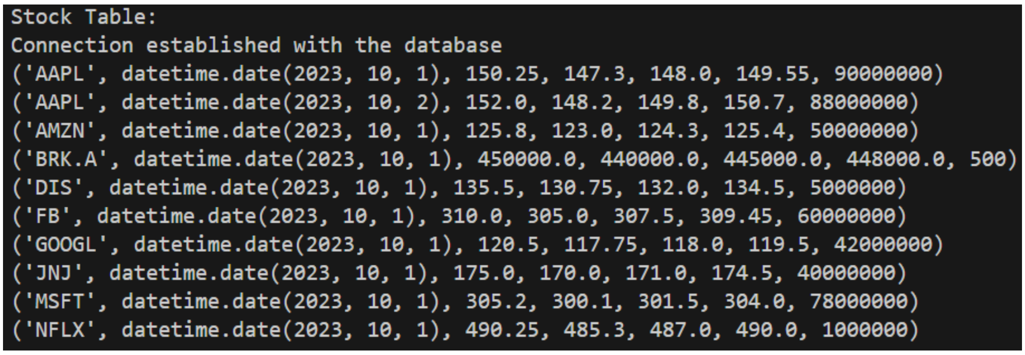
After Update Operations

Company Table:



```
Company Table:
Connection established with the database
('AAPL', 'Apple Inc.', 4371000000)
('AMZN', 'Amazon.com Inc.', 504000000)
('BRK.A', 'Berkshire Hathaway Inc.', 1382000)
('DIS', 'Walt Disney Co.', 1820000000)
('FB', 'Meta Platforms Inc.', 2400000000)
('GOOGL', 'Alphabet Inc.', 300900000)
('JNJ', 'Johnson & Johnson', 2601000000)
('MSFT', 'Microsoft Corp.', 7603000000)
('NFLX', 'Netflix Corporation', 460000000)
('TSLA', 'Tesla Inc.', 1000000000)
```

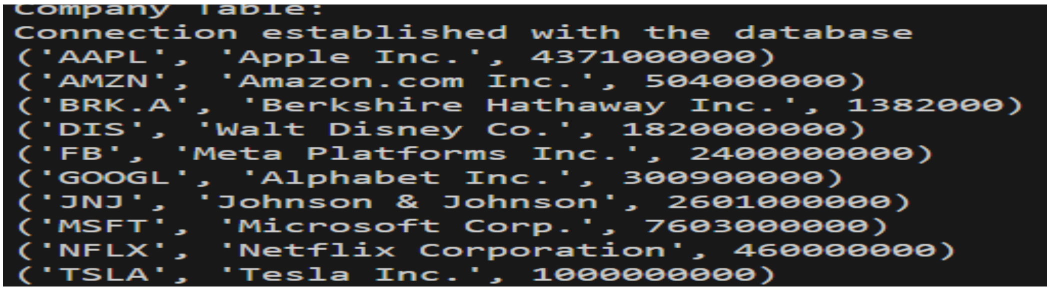
Stock Table:



```
Stock Table:
Connection established with the database
('AAPL', datetime.date(2023, 10, 1), 150.25, 147.3, 148.0, 149.55, 90000000)
('AAPL', datetime.date(2023, 10, 2), 152.0, 148.2, 149.8, 150.7, 88000000)
('AMZN', datetime.date(2023, 10, 1), 125.8, 123.0, 124.3, 125.4, 50000000)
('BRK.A', datetime.date(2023, 10, 1), 450000.0, 440000.0, 445000.0, 448000.0, 500)
('DIS', datetime.date(2023, 10, 1), 135.5, 130.75, 132.0, 134.5, 5000000)
('FB', datetime.date(2023, 10, 1), 310.0, 305.0, 307.5, 309.45, 60000000)
('GOOGL', datetime.date(2023, 10, 1), 120.5, 117.75, 118.0, 119.5, 42000000)
('JNJ', datetime.date(2023, 10, 1), 175.0, 170.0, 171.0, 174.5, 40000000)
('MSFT', datetime.date(2023, 10, 1), 305.2, 300.1, 301.5, 304.0, 78000000)
('NFLX', datetime.date(2023, 10, 1), 490.25, 485.3, 487.0, 490.0, 1000000)
```

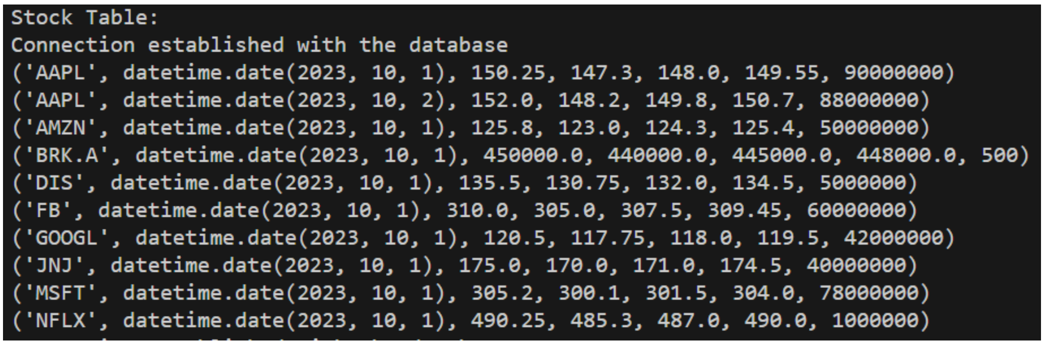
Before Delete Operations

Company Table:



```
Company Table:
Connection established with the database
('AAPL', 'Apple Inc.', 4371000000)
('AMZN', 'Amazon.com Inc.', 504000000)
('BRK.A', 'Berkshire Hathaway Inc.', 1382000)
('DIS', 'Walt Disney Co.', 1820000000)
('FB', 'Meta Platforms Inc.', 2400000000)
('GOOGL', 'Alphabet Inc.', 300900000)
('JNJ', 'Johnson & Johnson', 2601000000)
('MSFT', 'Microsoft Corp.', 7603000000)
('NFLX', 'Netflix Corporation', 460000000)
('TSLA', 'Tesla Inc.', 1000000000)
```

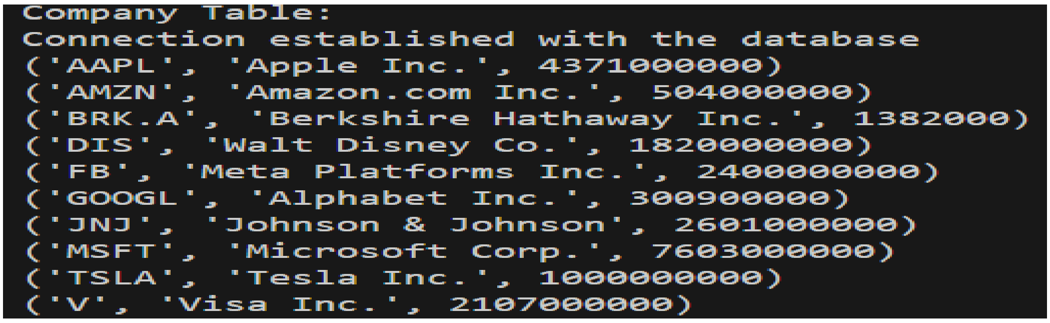
Stock Table:



```
Stock Table:
Connection established with the database
('AAPL', datetime.date(2023, 10, 1), 150.25, 147.3, 148.0, 149.55, 90000000)
('AAPL', datetime.date(2023, 10, 2), 152.0, 148.2, 149.8, 150.7, 88000000)
('AMZN', datetime.date(2023, 10, 1), 125.8, 123.0, 124.3, 125.4, 50000000)
('BRK.A', datetime.date(2023, 10, 1), 450000.0, 440000.0, 445000.0, 448000.0, 500)
('DIS', datetime.date(2023, 10, 1), 135.5, 130.75, 132.0, 134.5, 5000000)
('FB', datetime.date(2023, 10, 1), 310.0, 305.0, 307.5, 309.45, 60000000)
('GOOGL', datetime.date(2023, 10, 1), 120.5, 117.75, 118.0, 119.5, 42000000)
('JNJ', datetime.date(2023, 10, 1), 175.0, 170.0, 171.0, 174.5, 40000000)
('MSFT', datetime.date(2023, 10, 1), 305.2, 300.1, 301.5, 304.0, 78000000)
('NFLX', datetime.date(2023, 10, 1), 490.25, 485.3, 487.0, 490.0, 1000000)
```

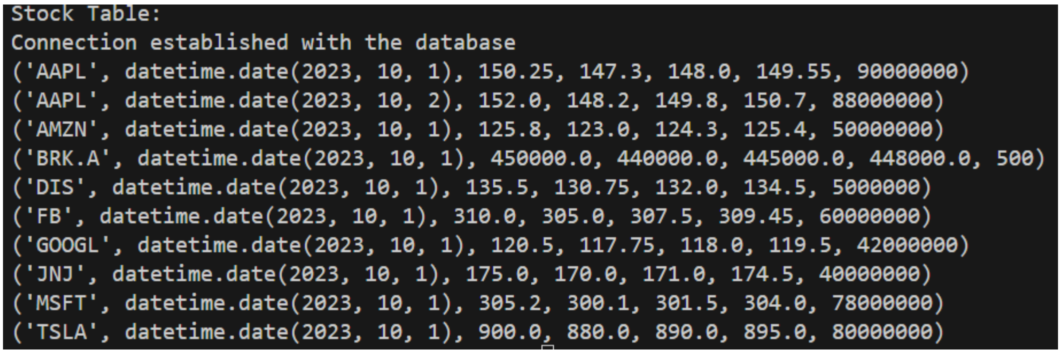
After Delete Operations

Company Table:



```
Company Table:
Connection established with the database
('AAPL', 'Apple Inc.', 4371000000)
('AMZN', 'Amazon.com Inc.', 504000000)
('BRK.A', 'Berkshire Hathaway Inc.', 1382000)
('DIS', 'Walt Disney Co.', 1820000000)
('FB', 'Meta Platforms Inc.', 2400000000)
('GOOGL', 'Alphabet Inc.', 300900000)
('JNJ', 'Johnson & Johnson', 2601000000)
('MSFT', 'Microsoft Corp.', 7603000000)
('TSLA', 'Tesla Inc.', 1000000000)
('V', 'Visa Inc.', 2107000000)
```

Stock Table:



```
Stock Table:
Connection established with the database
('AAPL', datetime.date(2023, 10, 1), 150.25, 147.3, 148.0, 149.55, 90000000)
('AAPL', datetime.date(2023, 10, 2), 152.0, 148.2, 149.8, 150.7, 88000000)
('AMZN', datetime.date(2023, 10, 1), 125.8, 123.0, 124.3, 125.4, 50000000)
('BRK.A', datetime.date(2023, 10, 1), 450000.0, 440000.0, 445000.0, 448000.0, 500)
('DIS', datetime.date(2023, 10, 1), 135.5, 130.75, 132.0, 134.5, 5000000)
('FB', datetime.date(2023, 10, 1), 310.0, 305.0, 307.5, 309.45, 60000000)
('GOOGL', datetime.date(2023, 10, 1), 120.5, 117.75, 118.0, 119.5, 42000000)
('JNJ', datetime.date(2023, 10, 1), 175.0, 170.0, 171.0, 174.5, 40000000)
('MSFT', datetime.date(2023, 10, 1), 305.2, 300.1, 301.5, 304.0, 78000000)
('TSLA', datetime.date(2023, 10, 1), 900.0, 880.0, 890.0, 895.0, 80000000)
```