Normalization and Functional Dependency analysis for each table

1. Advisor Table
   - Attributes: `AdvisorID`, `FirstName`, `LastName`, `PhoneNumber`, `Email`, `Experience`, `Fees`
   - Functional Dependencies:
     - `AdvisorID → FirstName, LastName, PhoneNumber, Email, Experience, Fees`
   - Normalization:
     - 1NF: All values are atomic, so the table is in 1NF.
     - 2NF: No partial dependencies (as `AdvisorID` fully determines all attributes).
     - 3NF: No transitive dependencies; all non-key attributes depend directly on the primary key (`AdvisorID`).
     - BCNF: This table is in BCNF as `AdvisorID` is a candidate key with no partial or transitive dependencies.

2. Investor Table
   - Attributes: `InvestorID`, `FirstName`, `LastName`, `DOB`, `Age`, `Email`, `PhoneNumber`, `Address`, `DateJoined`, `AdvisorID`
   - Functional Dependencies:
     - `InvestorID → FirstName, LastName, DOB, Age, Email, PhoneNumber, Address, DateJoined, AdvisorID`
   - Normalization:
     - 1NF: The table has atomic values.
     - 2NF: `InvestorID` is the primary key and fully determines all other attributes, so there are no partial dependencies.
     - 3NF: There are no transitive dependencies since all non-key attributes depend directly on `InvestorID`.
     - BCNF: The table is in BCNF.

3. StockInformation Table
   - Attributes: `Symbol`, `CompanyName`, `CurrentPrice`, `MarketCap`, `Sector`, `DividendYield`, `PriceEarningsRatio`, `52WeekHigh`, `52WeekLow`
   - Functional Dependencies:
     - `Symbol → CompanyName, CurrentPrice, MarketCap, Sector, DividendYield, PriceEarningsRatio, 52WeekHigh, 52WeekLow`
   - Normalization:
     - 1NF: Atomic values in all attributes.
     - 2NF: `Symbol` is the primary key, and it fully determines each attribute.

- 3NF: No transitive dependencies exist.
- BCNF: This table is in BCNF.

4. BondInformation Table
- Attributes: `BondID`, `BondName`, `BondPrice`, `CouponRate`, `MaturityDate`, `CreditRating`
- Functional Dependencies:
  - `BondID → BondName, BondPrice, CouponRate, MaturityDate, CreditRating`
- Normalization:
  - 1NF: All values are atomic.
  - 2NF: `BondID` is the primary key and determines all other attributes, so no partial dependencies exist.
  - 3NF: No transitive dependencies are present.
  - BCNF: The table is in BCNF.

5. Fund Table
- Attributes: `FundID`, `FundName`, `FundType`, `NAV`, `ExpenseRatio`, `FundManager`, `InceptionDate`
- Functional Dependencies:
  - `FundID → FundName, FundType, NAV, ExpenseRatio, FundManager, InceptionDate`
- Normalization:
  - 1NF: Atomic values.
  - 2NF: No partial dependencies, as `FundID` is the primary key.
  - 3NF: No transitive dependencies exist.
  - BCNF: The table is in BCNF.

6. StockHoldings Table
- Attributes: `StockHoldingID`, `InvestorID`, `Symbol`, `Quantity`
- Functional Dependencies:
  - `StockHoldingID → InvestorID, Symbol, Quantity`
- Normalization:
  - 1NF: Atomic values.
  - 2NF: All attributes depend fully on the primary key `StockHoldingID`.
  - 3NF: No transitive dependencies.
  - BCNF: The table is in BCNF.

7. BondHoldings Table
   - Attributes: `BondHoldingID`, `InvestorID`, `BondID`, `Quantity`
   - Functional Dependencies:
     - `BondHoldingID → InvestorID, BondID, Quantity`
   - Normalization:
     - 1NF: Atomic values.
     - 2NF: No partial dependencies.
     - 3NF: No transitive dependencies.
     - BCNF: Already in BCNF.

8. MutualFundHoldings Table
   - Attributes: `MutualFundHoldingID`, `InvestorID`, `FundID`, `Quantity`
   - Functional Dependencies:
     - `MutualFundHoldingID → InvestorID, FundID, Quantity`
   - Normalization:
     - 1NF: Atomic values.
     - 2NF: All attributes depend on `MutualFundHoldingID`.
     - 3NF: No transitive dependencies.
     - BCNF: The table is in BCNF.

9. StockTransaction Table
   - Attributes: `TransactionID`, `InvestorID`, `Symbol`, `TransactionType`, `Quantity`, `Price`, `TransactionDate`, `TransactionValue`
   - Functional Dependencies:
     - `TransactionID → InvestorID, Symbol, TransactionType, Quantity, Price, TransactionDate, TransactionValue`
   - Normalization:
     - 1NF: Atomic values.
     - 2NF: All attributes depend fully on `TransactionID`.
     - 3NF: No transitive dependencies.
     - BCNF: The table is in BCNF.

10. BondTransaction Table
    - Attributes: `TransactionID`, `InvestorID`, `BondID`, `TransactionType`, `Quantity`, `Price`, `TransactionDate`, `TransactionValue`
    - Functional Dependencies:
      - `TransactionID → InvestorID, BondID, TransactionType, Quantity, Price, TransactionDate, TransactionValue`

- Normalization:
  - 1NF: Atomic values.
  - 2NF: `TransactionID` fully determines all attributes.
  - 3NF: No transitive dependencies exist.
  - BCNF: The table is in BCNF.

11. FundTransaction Table
- Attributes: `TransactionID`, `InvestorID`, `FundID`, `TransactionType`, `Quantity`, `Price`, `TransactionDate`, `TransactionValue`
- Functional Dependencies:
  - `TransactionID → InvestorID, FundID, TransactionType, Quantity, Price, TransactionDate, TransactionValue`
- Normalization:
  - 1NF: Atomic values.
  - 2NF: No partial dependencies as `TransactionID` fully determines all other attributes.
  - 3NF: No transitive dependencies exist.
  - BCNF: The table is in BCNF.

12. MarketData Table
- Attributes: `MarketDataID`, `Date`, `AssetType`, `AssetID`, `Price`, `Volume`
- Functional Dependencies:
  - `MarketDataID → Date, AssetType, AssetID, Price, Volume`
- Normalization:
  - 1NF: Atomic values.
  - 2NF: `MarketDataID` fully determines all other attributes.
  - 3NF: No transitive dependencies.
  - BCNF: Already in BCNF.

Update, Delete, and Insert Anomalies

Update Anomalies
1. Investor Contact Information: Updating contact details across tables could lead to inconsistencies.
2. Advisor Fees: Changes to advisor fees require updates in related investor and transaction records.
3. Market Prices: Price changes for stocks or bonds need to be updated in `StockInformation` and `BondInformation` to avoid inconsistency.

Delete Anomalies
1. Investor Deletion: Deleting an investor cascades to holdings and transactions, losing valuable financial history.
2. Stock or Bond Deletion: Removing a stock or bond causes deletions in holdings and transactions, potentially erasing critical data.
3. Fund Deletion: Deleting a fund erases associated holdings and transactions, leading to data loss.

Insert Anomalies
1. Holdings without Investor: Creating holdings without a valid `InvestorID` causes foreign key constraint violations.
2. Transaction without Asset: Transactions without valid stock, bond, or fund entries violate foreign key constraints.
3. Advisor Assignment without Advisor: Assigning a non-existent advisor prevents the creation of investor-advisor relationships.

Scrutiny of Redundancies

1. Investor Data: Redundant investor details in multiple tables. Solution: Store investor data only in `Investor` and reference by `InvestorID`.
2. Market Data and Asset Info: Duplicated `price` data in transactions and asset tables. Solution: Keep current prices in asset tables; transaction tables should store only transaction prices.
3. Fund and Investment: Fund information is repeated in holdings and transactions. Solution: Use `FundID` to reference fund details from `Fund` table.
4. Transaction Data: Investor and price info repeated in transactions. Solution: Only store transaction-specific details, referencing investor and asset tables as needed.
5. Advisor and Investor Association: `AdvisorID` repeated in `Investor` for advisor associations. Solution: Create `AdvisorAssignment` table to map `AdvisorID` to `InvestorID`, making updates easier.