Database Project

**Mehmet Turhan**

# Domain Description

Title: Stock Market Investing and Advising Database

We will create a database for a brokerage firm that manages investment portfolios for its clients. The firm has a number of financial advisors, who each have a set of clients they work with. Clients can hold multiple accounts with the firm, each of which can contain multiple investments in various stocks. The firm also tracks various financial metrics for each investment, such as its current price and its historical performance.

Entities and Relationships:

Clients: This entity will contain information about individual clients, including their name, contact information, and other relevant details. Each client will be assigned a unique identifier, client\_id, to ensure that their information can be easily tracked and accessed.

Accounts: This entity will contain information about individual accounts held by clients, including the account\_id, client\_id (foreign key), balance, and performance. Each account will be assigned a unique identifier, account\_id, to ensure that its information can be easily tracked and accessed. It also has an advisor\_id (foreign key) assigned to see which advisor is helping to make decisions on the investments for that account.

Investments: This entity will contain information about individual investments made by clients, including the investment\_id, stock\_id (foreign key), account\_id (foreign key), quantity, purchase\_price, and purchase\_date. Each investment will be assigned a unique identifier, investment\_id, to ensure that its information can be easily tracked and accessed.

Advisors: This entity will contain information about individual advisors, including their name, contact information, and other relevant details. Each advisor will be assigned a unique identifier, advisor\_id, to ensure that their information can be easily tracked and accessed. It will also contain an attribute called performance, which tracks the overall performance of the advisor's investments.

Transactions: This entity will contain information about individual transactions made by clients, including the transaction\_id, investment\_id (foreign key), exchange\_id (foreign key), transaction\_type, transaction\_date, price\_per\_share, and quantity. Each transaction will be assigned a unique identifier, transaction\_id, to ensure that its information can be easily tracked and accessed.

Metrics: This entity will contain information about individual metrics for each investment, including the metric\_id, investment\_id (foreign key), metric\_type, metric\_date, and value. Each metric will be assigned a unique identifier, metric\_id, to ensure that its information can be easily tracked and accessed.

# Exchanges: This entity will contain information about individual exchanges, including the exchange\_id and exchange\_name. Each exchange will be assigned a unique identifier, exchange\_id, to ensure that its information can be easily tracked and accessed. It will also contain an attribute called location, which shows where the exchange is located.

# Stocks: This entity will contain information about individual stocks, including the stock\_id, exchange\_id (foreign key), stock\_symbol, stock\_name, description, and industry. Each stock will be assigned a unique identifier, stock\_id, to ensure that its information can be easily tracked and accessed. It will also contain an attribute called industry, which shows which industry the stock is in.

# Entity-Relationship Diagram

Diagram

Description automatically generated

# Relational Schema

Client (client\_id, first\_name, last\_name, email, phone, preferred\_industry(FK))

Account (account\_id, client\_id (FK), advisor\_id (FK), balance, account\_performance)

Investment (investment\_id, stock\_id (FK), account\_id (FK), quantity, purchase\_price, purchase\_date)

Advisor (advisor\_id, first\_name, last\_name, email, phone, performance)

Transactions (transaction\_id, investment\_id (FK), exchange\_id (FK), transaction\_type, transaction\_date, price\_per\_share, quantity)

Metric (metric\_id, investment\_id (FK), metric\_type, metric\_date, value)

Exchange (exchange\_id, exchange\_name, location)

Stock (stock\_id, exchange\_id (FK), stock\_symbol, stock\_name, description, industry)

Diagram

Description automatically generated

Boyce–Codd Normal Form Decomposition

First we identify all functional dependencies in the schema.

Client\_id → {first\_name, last\_name, email, phone, preferred\_industry}

Account\_id → {client\_id, advisor\_id, balance, account\_performance}

Investment\_id → {stock\_id, account\_id, quantity, purchase\_price, purchase\_date}

Advisor\_id → {first\_name, last\_name, email, phone, performance}

Investment\_id {transaction\_id, exchange\_id, transaction\_type, transaction\_date, price\_per\_share, quantity}

Investment\_id → {metric\_id, metric\_type, metric\_date, value}

Exchange\_id → {exchange\_name, location}

Stock\_id → {exchange\_id, stock\_symbol, stock\_name, description, industry}

Then we check whether any non-prime attribute is functionally dependent on a proper subset of a candidate key.

The candidate keys:

{client\_id}

{account\_id}

{investment\_id}

{advisor\_id}

{transaction\_id}

{metric\_id}

{exchange\_id}

{stock\_id}

The non-prime attributes:

{first\_name, last\_name, email, phone, preferred\_industry} (from the Client relation)

{balance, account\_performance} (from the Account relation)

{quantity, purchase\_price, purchase\_date} (from the Investment relation)

{performance} (from the Advisor relation)

{transaction\_type, transaction\_date, price\_per\_share, quantity} (from the Transactions relation)

{metric\_type, metric\_date, value} (from the Metric relation)

{exchange\_name, location} (from the Exchange relation)

{stock\_symbol, stock\_name, description, industry} (from the Stock relation)

We can see that all non-prime attributes are fully dependent on the candidate keys, and there are no partial dependencies. Therefore, the schema is in 2NF.

Then we check whether any non-prime attribute is transitively dependent on a candidate key.

There are no non-prime attributes that are transitively dependent on a candidate key. Therefore, the schema is in 3NF.

Lastly we check whether any non-prime attribute is functionally dependent on another non-prime attribute.

There are no non-prime attributes that are functionally dependent on another non-prime attribute. Therefore, the schema is in BCNF.

# Transaction and Query Executions

--1: Find the total balance of all accounts grouped by preferred industry of clients.

SELECT c.preferred\_industry, SUM(a.balance) as total\_balance

FROM Client c

INNER JOIN Account a ON c.client\_id = a.client\_id

GROUP BY c.preferred\_industry;

| **preferred\_industry** | **total\_balance** |
| --- | --- |
| Energy | 15000 |
| Finance | 34000 |
| Healthcare | 30500 |
| Real Estate | 68000 |
| Retail | 12000 |
| Technology | 40000 |

--2: Find the advisors whose performance is greater than the average performance of all advisors and their associated clients.

SELECT DISTINCT ad.first\_name, ad.last\_name, ad.performance

FROM Advisor ad

WHERE ad.performance > (

SELECT AVG(performance)

FROM Advisor

) ORDER BY ad.performance DESC;

| **first\_name** | **last\_name** | **performance** |
| --- | --- | --- |
| Cathie | Wood | 9.5 |
| David | Swensen | 9.4 |
| Larry | Fink | 9.3 |
| Paul | Tudor Jones | 9.2 |
| Ray | Dalio | 9.1 |

--3: Give the first and last name of clients and their advisors who invested in the Real Estate industry and have an account performance greater than 4.0.

SELECT c.first\_name, c.last\_name, a.first\_name, a.last\_name

FROM Client c

INNER JOIN Account ac ON c.client\_id = ac.client\_id

INNER JOIN Advisor a ON ac.advisor\_id = a.advisor\_id

WHERE c.preferred\_industry = 'Real Estate'

AND ac.account\_performance > 4.0;

| **first\_name** | **last\_name** | **first\_name** | **last\_name** |
| --- | --- | --- | --- |
| Not | Sure | Paul | Tudor Jones |

--4: Give the first and last name of advisors who have no clients.

SELECT a.first\_name, a.last\_name

FROM Advisor a

LEFT OUTER JOIN Account ac ON a.advisor\_id = ac.advisor\_id

WHERE ac.account\_id IS NULL;

NO RESULT

--5: Find the first and last name of the clients and the advisors who have performance greater than 4.0

SELECT c.first\_name AS client\_first\_name, c.last\_name AS client\_last\_name,

adv.first\_name AS advisor\_first\_name, adv.last\_name AS advisor\_last\_name

FROM Client c

JOIN Account a ON c.client\_id = a.client\_id

JOIN Advisor adv ON a.advisor\_id = adv.advisor\_id

WHERE a.account\_performance > 4.0;

| **client\_first\_name** | **client\_last\_name** | **advisor\_first\_name** | **advisor\_last\_name** |
| --- | --- | --- | --- |
| Mehmet | Turhan | Jamie | Dimon |
| Lincoln | Chapata | Larry | Fink |
| Nick | Price | Abigail | Johnson |
| Rawleigh | Pollock | David | Swensen |
| Abdulrahman | Nassar | Warren | Buffett |
| Not | Sure | Paul | Tudor Jones |

--6: Find the advisor's first and last name and their performance who have clients with account balance less than 15000.

SELECT DISTINCT adv.first\_name, adv.last\_name, adv.performance

FROM Advisor adv

JOIN Account a ON adv.advisor\_id = a.advisor\_id

WHERE a.balance < 15000;

| **first\_name** | **last\_name** | **performance** |
| --- | --- | --- |
| Abigail | Johnson | 9 |
| Howard | Marks | 8.8 |
| Ken | Griffin | 8.9 |

--7: Find the purchase\_price of all the investments made in NASDAQ and purchase\_price of more than 3000$.

SELECT s.stock\_name, i.purchase\_price

FROM Stock s

INNER JOIN Investment i ON s.stock\_id = i.stock\_id

WHERE s.exchange\_id = (SELECT exchange\_id FROM Exchange WHERE exchange\_name = 'NASDAQ')

AND i.purchase\_price > 3000;

| **stock\_name** | **purchase\_price** |
| --- | --- |
| Tesla, Inc. | 6240 |
| INDUS Realty Trust, Inc. | 8945 |
| Biogen Inc. | 9230 |

--8: Find all investments and their associated metrics.

SELECT Investment.investment\_id, Investment.purchase\_price, Metric.metric\_type, Metric.value

FROM Investment

LEFT OUTER JOIN Metric ON Investment.investment\_id = Metric.investment\_id;

| **investment\_id** | **purchase\_price** | **metric\_type** | **value** |
| --- | --- | --- | --- |
| 1 | 12550 | Dividend Yield | 0.005 |
| 1 | 12550 | PE ratio | 25.7 |
| 2 | 17640 | Dividend Yield | 0.008 |
| 2 | 17640 | PE ratio | 35.6 |
| 3 | 14560 | Dividend Yield | 0.015 |
| 3 | 14560 | PE ratio | 20.9 |
| 4 | 6240 | Dividend Yield | 0.025 |
| 4 | 6240 | PE ratio | 15.3 |
| 5 | 8945 | Dividend Yield | 0.03 |
| 5 | 8945 | PE ratio | 12.8 |
| 6 | 9230 | Dividend Yield | 0.02 |
| 6 | 9230 | PE ratio | 18.4 |
| 7 | 7812.5 | Dividend Yield | 0.012 |
| 7 | 7812.5 | PE ratio | 21.6 |
| 8 | 5820 | Dividend Yield | 0.007 |
| 8 | 5820 | PE ratio | 30.2 |
| 9 | 6502.5 | Dividend Yield | 0.02 |
| 9 | 6502.5 | PE ratio | 16.5 |
| 10 | 5280 | Dividend Yield | 0.008 |
| 10 | 5280 | PE ratio | 28.6 |