CS 254: DATABASE MANAGEMENT SYSTEMS

STOCK MARKET DATABASE SYSTEM

Prepared By:

Rashwin Musuku (**22CSB0F07**)

Pradhyumna Kasam (**22CSB0F09**)

Vision:

In this project, we have developed a stock market database system tailored to efficiently handle the complexity of stock trading and market analysis. The database serves as a centralized repository for storing crucial data related to various entities within the stock market ecosystem, facilitating seamless interaction and information retrieval for market participants.

Our database encompasses essential components such as customers, brokers, companies, and stocks, each meticulously represented with detailed attributes capturing pertinent information. For instance, customer details including names, passwords, and contact information are stored, enabling smooth communication and account management. Similarly, information about companies, such as headquarters location, CEO, and sector classification, provides valuable insights into the corporate landscape.

Crucially, the database records financial metrics such as earnings per share (EPS), dividend yield, and market valuation for stocks, empowering investors with the data needed to make informed investment decisions. Transaction details, including order IDs, quantities, prices, and timestamps, are meticulously logged, ensuring accurate tracking and management of exchange transactions.

Idea:

One of the key features of our database is the establishment of relationships among different entities, facilitated by unique identifiers such as Ticker Codes and Portfolio IDs. These relationships enable seamless data retrieval along with efficiency within the stock market ecosystem.

Furthermore, our database management system incorporates measures to ensure compliance with regulatory protocols. Controls are implemented to protect data integrity.

In conclusion, our stock market database management system provides a comprehensive solution for managing stock market data, facilitating investment analysis, transaction processing, and market monitoring. With its user friendly interface and robust functionality, the database empowers stakeholders to navigate the complexities of the stock market with confidence and efficiency.

Table of Contents:

ENTITIES

1. DBA: Database administrators are registered with unique login IDs and provide essential details such as passwords, dates of birth, phone numbers, and names for managing database access and administration.

2. Customer: Customers are identified by unique IDs and provide essential information such as names, passwords, contact details, and dates of birth for account management and communication purposes.

3. Broker: Brokers are registered with unique IDs and provide key details including names, passwords, dates of birth, and contact numbers for facilitating trades and managing client accounts.

4. Portfolio: Portfolios are assigned unique IDs and track essential financial metrics such as profit/loss and amount invested to monitor investment performance and manage diversified holdings.

5. Stock: Stocks are represented by ticker codes and linked to corporate IDs, allowing for the identification of listed securities and their respective issuing companies within the stock market database.

6. Common: Common stocks are associated with ticker codes and include attributes such as votes/share, EPS, dividend yield, outstanding shares, market valuation, and float value for analyzing and managing common stock data.

7. Preferred: Preferred stocks are identified by ticker codes and include attributes such as call dates, cumulative status, par value, float value, outstanding shares, and market valuation for tracking preferred stock performance and characteristics.

8. Technicals: Technical indicators for stocks are recorded with attributes such as all-time high and low prices, daily closing and opening prices, and IPO prices associated with ticker codes for analyzing stock price movements and trends.

9. Exchange: Exchange transactions are logged with unique order IDs, quantity, price, timestamp, buyer and seller portfolio IDs, and associated ticker codes for tracking and managing stock exchange transactions.

10. Listings: Stock listings are recorded with unique listing numbers, prices, timestamps, and associated ticker codes for tracking and managing stock listings and market activity.

11. Company: Companies are listed with names, headquarters locations, CEO names, corporate IDs, and sector affiliations to identify corporate entities and their operational details within specific sectors.

12. Company Fundamentals: Company fundamentals include financial metrics such as income, revenue, EBITDA, debt, and dividend payouts associated with corporate IDs to assess company performance and financial health.

13. Sector: Sectors are categorized by names and recorded with market capitalization, price-to-earnings ratio, price-to-book ratio, and dividend yield to analyze sector-wise market trends and performance.

RELATIONSHIPS

One-to-One Relationships: 1 -> 1

- Customer-Portfolio
- Stock-Common
- Stock-Preferred
- Company-Company_Fundamentals
- Stock-Technicals

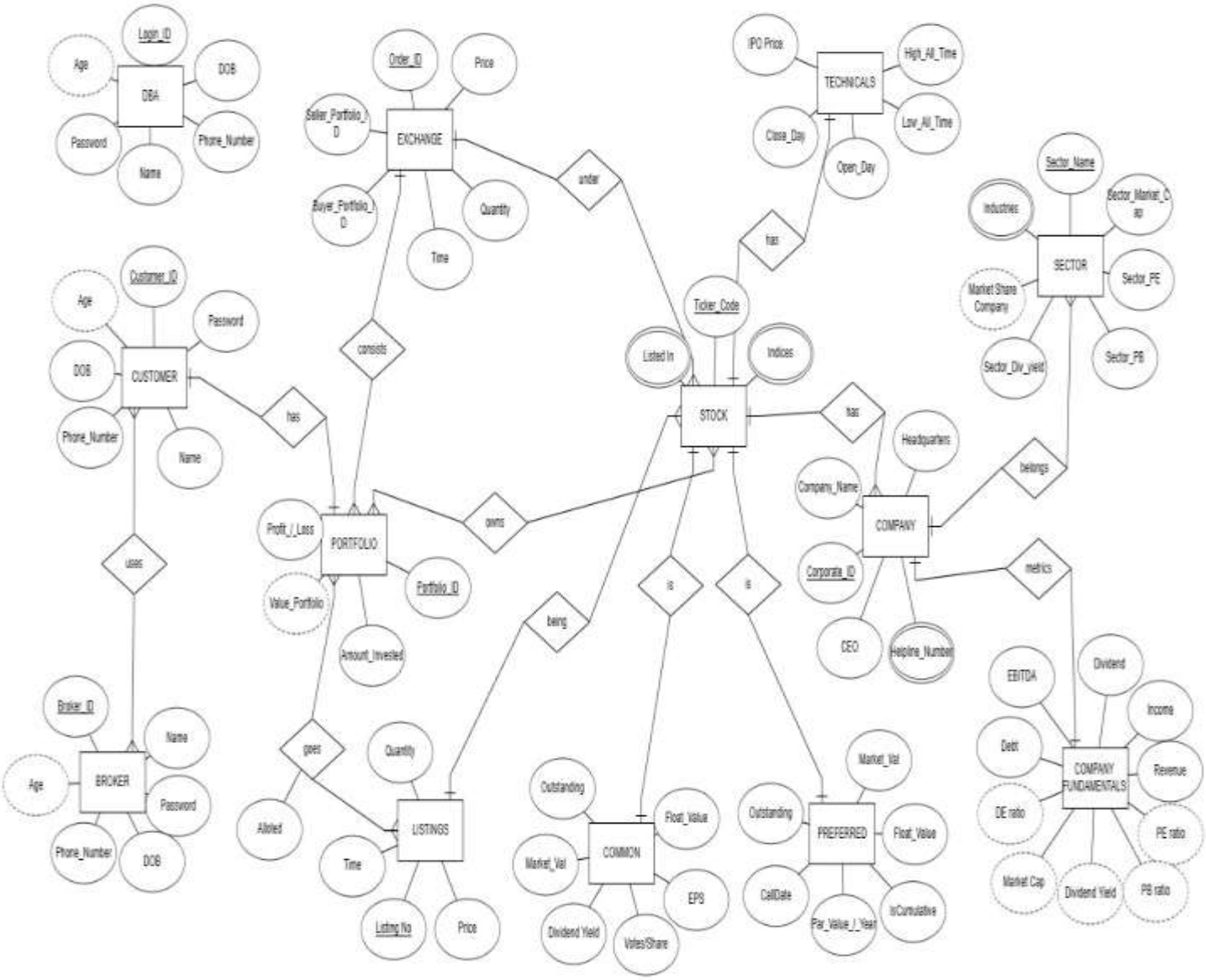
One-to-Many Relationships: 1 -> N

- Portfolio-Exchange
- Stock-Listings
- Stock-Exchange
- Sector-Company
- Company-Stock

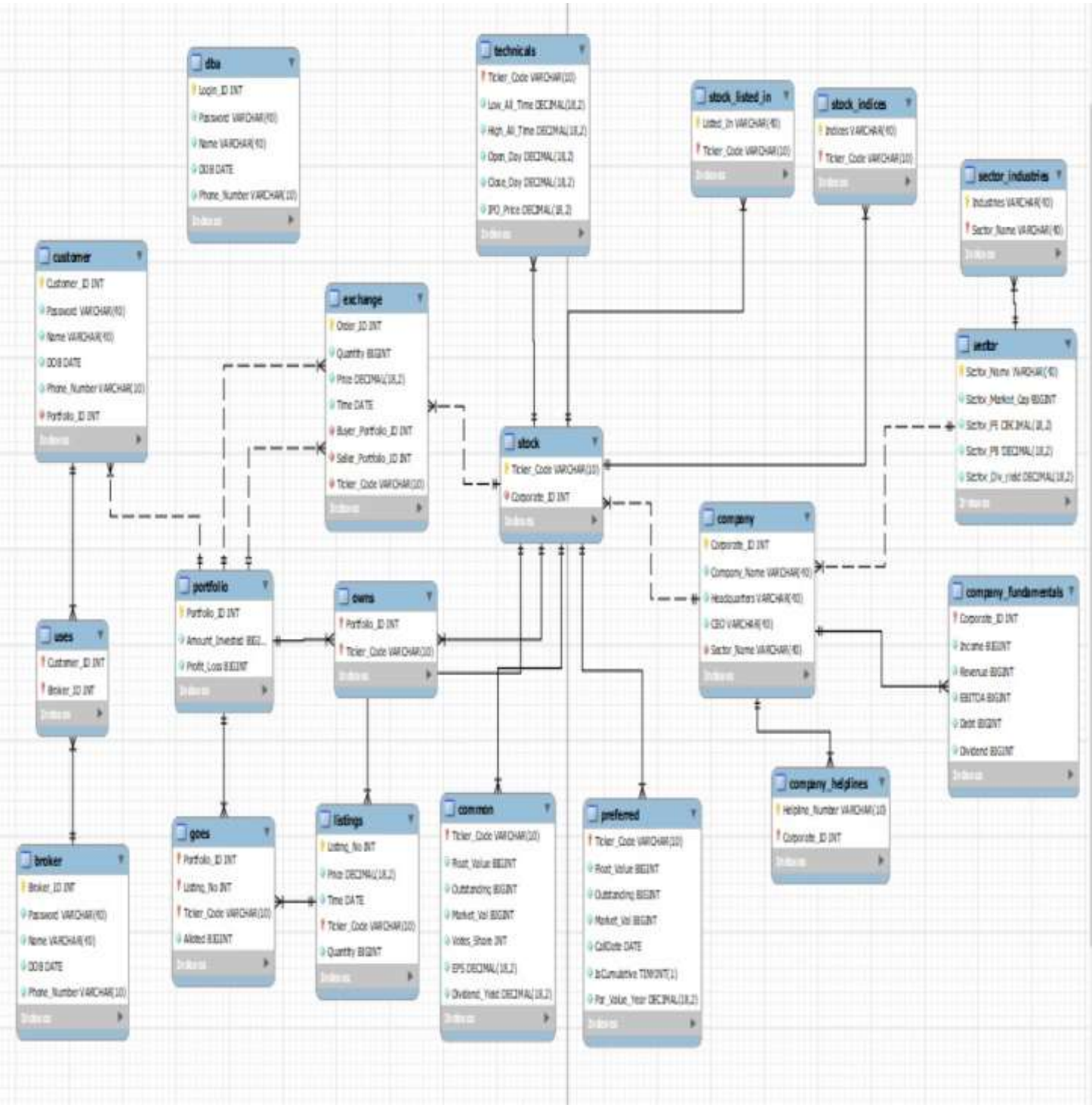
Many-to-Many relationships: N -> M

- Customer-Broker
- Portfolio-Listings
- Portfolio-Stock

ENTITY RELATIONAL MODEL



RELATIONAL SCHEMA



TABLES, KEYS, NORMALIZATION

Primary Key: Determines All Attributes In The Table.

BCNF: FD|X -> Y. X Is Primary Key.

ATTRIBUTE	DATATYPE

```
CREATE DATABASE STOCK_MARKET;

USE  STOCK_MARKET;
```

1. DBA:

Login_ID	INT
Password	VARCHAR(40)
Phone_Number	VARCHAR(10)
Name	VARCHAR(40)
DOB	DATE

Primary Key: (Login_ID).

Foreign Key: None.

Table is in BCNF.

```
CREATE TABLE DBA
(
Login_ID INT NOT NULL,
Password VARCHAR(40) NOT NULL,
Name VARCHAR(40) NOT NULL,
DOB DATE NOT NULL,
Phone_Number VARCHAR(10) NOT NULL,
PRIMARY KEY (Login_ID)
);
```

2. Broker:

Broker_ID	INT
Password	VARCHAR(40)
Phone_Number	VARCHAR(10)
Name	VARCHAR(40)
DOB	DATE

Primary Key: (Broker_ID).

Foreign Key: None.

Table is in BCNF.

```
CREATE TABLE BROKER
(
Broker_ID INT NOT NULL,
Password VARCHAR(40) NOT NULL,
Name VARCHAR(40) NOT NULL,
DOB DATE NOT NULL,
Phone_Number VARCHAR(10) NOT NULL,
PRIMARY KEY (Broker_ID)
);
```

3. Portfolio:

Portfolio_ID	INT
Amount_Invested	BIGINT
Profit_Loss	BIGINT

Primary Key: (Portfolio_ID).

Foreign Key: None.

Table is in BCNF.

```
CREATE TABLE PORTFOLIO
(
Portfolio_ID INT NOT NULL,
Amount_Invested BIGINT NOT NULL,
Profit_Loss BIGINT NOT NULL,
PRIMARY KEY (Portfolio_ID)
);
```

4. Sector:

Sector_Name	VARCHAR(40)
Sector_Market_Cap	BIGINT
Sector_PE	DECIMAL(18,2)
Sector_PB	DECIMAL(18,2)
Sector_Div_yield	DECIMAL(18,2)

Primary Key: (Portfolio_ID).

Foreign Key: None.

Table is in BCNF.

```
CREATE TABLE SECTOR
(
Sector_Name VARCHAR(40) NOT NULL,
Sector_Market_Cap BIGINT NOT NULL,
Sector_PE DECIMAL(18, 2) NOT NULL,
```

Sector_PB DECIMAL(18, 2) NOT NULL,

Sector_Div_yield DECIMAL(18, 2) NOT NULL,

PRIMARY KEY (Sector_Name)

);

5. Sector_Industries:

Industries	VARCHAR(40)
Sector_Name	VARCHAR(40)

Primary Key: (Industries, Sector_Name).

Foreign Key: Sector_Name.

Sector(Sector_Name) -> Sector_Name.

Table is in BCNF.

CREATE TABLE SECTOR_INDUSTRIES

(

Industries VARCHAR(40) NOT NULL,

Sector_Name VARCHAR(40) NOT NULL,

PRIMARY KEY (Industries, Sector_Name),

FOREIGN KEY (Sector_Name) REFERENCES SECTOR(Sector_Name)

);

6. Company:

Corporate_ID	INT
Company_Name	VARCHAR(40)
Headquaters	VARCHAR(40)
CEO	VARCHAR(40)
Sector_Name	VARCHAR(40)

Primary Key: (Corporate_ID).

Foreign Key: Sector_Name.

Sector(Sector_Name) -> Sector_Name.

Table is in BCNF.

```
CREATE TABLE COMPANY
(
Corporate_ID INT NOT NULL,
Company_Name VARCHAR(40) NOT NULL,
Headquarters VARCHAR(40) NOT NULL,
CEO VARCHAR(40) NOT NULL,
Sector_Name VARCHAR(40) NOT NULL,
PRIMARY KEY (Corporate_ID),
FOREIGN KEY (Sector_Name) REFERENCES
SECTOR(Sector_Name)
);
```

7. Company_Fundamentals:

ATTRIBUTES	DATATYPE
Corporate_ID	INT
Income	BIGINT
Revenue	BIGINT
EBITA	BIGINT
Debt	BIGINT
Dividend	BIGINT

Primary Key: (Corporate_ID).

Foreign Key: Corporate_ID.

Company(Corporate_ID) -> Corporate_ID.

Table is in BCNF.

```
CREATE TABLE COMPANY_FUNDAMENTALS
(
Corporate_ID INT NOT NULL,
Income BIGINT NOT NULL,
Revenue BIGINT NOT NULL,
EBITDA BIGINT NOT NULL,
Debt BIGINT NOT NULL,
Dividend BIGINT NOT NULL,
PRIMARY KEY (Corporate_ID),
FOREIGN KEY (Corporate_ID) REFERENCES
COMPANY(Corporate_ID)
);
```

8. Company_Helplines:

Helpline_Number	VARCHAR(10)
Corporate_ID	INT

Primary Key: (Helpline_Number, Corporate_ID).

Foreign Key: Corporate_ID.

Company(Corporate_ID) -> Corporate_ID.

Table is in BCNF.

```
CREATE TABLE COMPANY_HELPLINES
(
Helpline_Number VARCHAR(10) NOT NULL,
Corporate_ID INT NOT NULL,
PRIMARY KEY (Helpline_Number, Corporate_ID),
FOREIGN KEY (Corporate_ID) REFERENCES
COMPANY(Corporate_ID)
);
```

9. Stock:

Ticker_Code	VARCHAR(10)
Corporate_ID	INT

Primary Key: (Ticker_Code).

Foreign Key: Corporate_ID.

Company(Corporate_ID) -> Corporate_ID.

Table is in BCNF.

```
CREATE TABLE STOCK
(
Ticker_Code VARCHAR(10) NOT NULL,
Corporate_ID INT NOT NULL,
PRIMARY KEY (Ticker_Code),
FOREIGN KEY (Corporate_ID) REFERENCES
COMPANY(Corporate_ID)
);
```

10. Common:

Ticker_Code	VARCHAR(10)
Float_Value	BIGINT
Outstanding	BIGINT
Market_Val	BIGINT
Votes_Share	INT
EPS	DECIMAL(18,2)
Dividend_Yield	DECIMAL(18,2)

Primary Key: (Ticker_Code).

Foreign Key: Ticker_Code.

Stock(Ticker_Code) -> Ticker_Code.

Table is in BCNF.

```
CREATE TABLE COMMON
(
Ticker_Code VARCHAR(10) NOT NULL,
Float_Value BIGINT NOT NULL,
Outstanding BIGINT NOT NULL,
Market_Val BIGINT NOT NULL,
Votes_Share INT NOT NULL,
EPS DECIMAL(18, 2) NOT NULL,
Dividend_Yield DECIMAL(18, 2) NOT NULL,
PRIMARY KEY (Ticker_Code),
FOREIGN KEY (Ticker_Code) REFERENCES
STOCK(Ticker_Code)
);
```

11. Preferred:

Ticker_Code	VARCHAR(10)
Float_Value	BIGINT
Outstanding	BIGINT
Market_Val	BIGINT
CallDate	DATE
IsCumulative	BOOL
Par_Value_Year	DECIMAL(18,2)

Primary Key: (Ticker_Code).

Foreign Key: Ticker_Code.

Stock(Ticker_Code) -> Ticker_Code.

Table is in BCNF.

```
CREATE TABLE PREFERRED
(
Ticker_Code VARCHAR(10) NOT NULL,
Float_Value BIGINT NOT NULL,
Outstanding BIGINT NOT NULL,
Market_Val BIGINT NOT NULL,
CallDate DATE NOT NULL,
IsCumulative BOOL NOT NULL,
Par_Value_Year DECIMAL(18, 2) NOT NULL,
PRIMARY KEY (Ticker_Code),
FOREIGN KEY (Ticker_Code) REFERENCES
STOCK(Ticker_Code)
);
```

12. Technicals:

Ticker_Code	VARCHAR(10)
Low_All_Time	DECIMAL(18, 2)
High_All_Time	DECIMAL(18, 2)
Open_Day	DECIMAL(18, 2)
Close_Day	DECIMAL(18, 2)
IPO_Price	DECIMAL(18, 2)

Primary Key: (Ticker_Code).

Foreign Key: Ticker_Code.

Stock(Ticker_Code) -> Ticker_Code.

Table is in BCNF.

```
CREATE TABLE TECHNICALS
(
Ticker_Code VARCHAR(10) NOT NULL,
Low_All_Time DECIMAL(18, 2) NOT NULL,
High_All_Time DECIMAL(18, 2) NOT NULL,
Open_Day DECIMAL(18, 2) NOT NULL,
Close_Day DECIMAL(18, 2) NOT NULL,
IPO_Price DECIMAL(18, 2) NOT NULL,
PRIMARY KEY (Ticker_Code),
FOREIGN KEY (Ticker_Code) REFERENCES
STOCK(Ticker_Code)
);
```

13. STOCK_LISTED_IN:

Ticker_Code	VARCHAR(10)
Listed_In	VARCHAR(40)

Primary Key: (Ticker_Code, Listed_In).

Foreign Key: Ticker_Code.

Stock(Ticker_Code) -> Ticker_Code.

Table is in BCNF.

```
CREATE TABLE STOCK_LISTED_IN
(
Listed_In VARCHAR(40) NOT NULL,
Ticker_Code VARCHAR(10) NOT NULL,
PRIMARY KEY (Listed_In, Ticker_Code),
FOREIGN KEY (Ticker_Code) REFERENCES
STOCK(Ticker_Code)
);
```

14. STOCK_INDICES:

Ticker_Code	VARCHAR(10)
Indices	VARCHAR(40)

Primary Key: (Ticker_Code, Indices).

Foreign Key: Ticker_Code.

Stock(Ticker_Code) -> Ticker_Code.

Table is in BCNF.

```
CREATE TABLE STOCK_INDICES
(
Indices VARCHAR(40) NOT NULL,
Ticker_Code VARCHAR(10) NOT NULL,
PRIMARY KEY (Indices, Ticker_Code),
FOREIGN KEY (Ticker_Code) REFERENCES
STOCK(Ticker_Code)
);
```

15. Customer:

Customer_ID	INT
Password	VARCHAR(40)
Name	VARCHAR(40)
DOB	DATE
Phone_Number	VARCHAR(10)
Portfolio_ID	INT

Primary Key: (Customer_ID).

Foreign Key: Portfolio_ID.

Portfolio(Portfolio_ID) -> Portfolio_ID.

Table is in BCNF.

```
CREATE TABLE CUSTOMER
(
Customer_ID INT NOT NULL,
Password VARCHAR(40) NOT NULL,
Name VARCHAR(40) NOT NULL,
DOB DATE NOT NULL,
```

Phone_Number VARCHAR(10) NOT NULL,

Portfolio_ID INT NOT NULL,

PRIMARY KEY (Customer_ID),

FOREIGN KEY (Portfolio_ID) REFERENCES PORTFOLIO(Portfolio_ID)

);

16. Uses:

Customer_ID	INT
Broker_ID	INT

Primary Key: (Customer_ID, Broker_ID).

Foreign Key: Customer_ID, Broker_ID.

Customer(Customer_ID) -> Customer_ID.

Broker(Broker_ID) -> Broker_ID.

Table is in BCNF.

CREATE TABLE USES

(

Customer_ID INT NOT NULL,

Broker_ID INT NOT NULL,

PRIMARY KEY (Customer_ID, Broker_ID),

FOREIGN KEY (Customer_ID) REFERENCES CUSTOMER(Customer_ID),

FOREIGN KEY (Broker_ID) REFERENCES BROKER(Broker_ID)

);

17. Exchange:

Order_ID	INT
Quantity	BIGINT
Price	DECIMAL(18,2)
Time	DATE
Buyer_Portfolio_ID	INT
Seller_Portfolio_ID	INT
Ticker_Code	VARCHAR(10)

Primary Key: (Order_ID).

Foreign Key: Buyer_Portfolio_ID, Seller_Portfolio_ID, Ticker_Code.

Portfolio(Portfolio_ID) -> Buyer_Portfolio_ID.

Portfolio(Portfolio_ID) -> Seller_Portfolio_ID.

Stock(Ticker_Code) -> Ticker_Code.

Table is in BCNF.

```
CREATE TABLE EXCHANGE
(
Order_ID INT NOT NULL,
Quantity BIGINT NOT NULL,
Price DECIMAL(18, 2) NOT NULL,
Time DATE NOT NULL,
Buyer_Portfolio_ID INT NOT NULL,
Seller_Portfolio_ID INT NOT NULL,
Ticker_Code VARCHAR(10) NOT NULL,
PRIMARY KEY (Order_ID),
```

FOREIGN KEY (Buyer_Portfolio_ID) REFERENCES PORTFOLIO(Portfolio_ID),

FOREIGN KEY (Seller_Portfolio_ID) REFERENCES PORTFOLIO(Portfolio_ID),

FOREIGN KEY (Ticker_Code) REFERENCES STOCK(Ticker_Code)

);

18. Listings:

Listing_No	INT
Quantity	BIGINT
Price	DECIMAL(18,2)
Time	DATE
Ticker_Code	VARCHAR(10)

Primary Key: (Listing_No, Ticker_Code).

Foreign Key: Ticker_Code.

Stock(Ticker_Code) -> Ticker_Code.

Table is in BCNF.

CREATE TABLE LISTINGS

(

Listing_No INT NOT NULL,

Price DECIMAL(18, 2) NOT NULL,

Time DATE NOT NULL,

Ticker_Code VARCHAR(10) NOT NULL,

Quantity BIGINT NOT NULL,

PRIMARY KEY (Listing_No, Ticker_Code),

FOREIGN KEY (Ticker_Code) REFERENCES
STOCK(Ticker_Code)

);

19. Goes:

Portfolio_ID	INT
Listing_No	INT
Ticker_Code	VARCHAR(10)
Alloted	BIGINT

Primary Key: (Portfolio_ID ,Listing_No, Ticker_Code).

Foreign Key: Portfolio_ID, Listing_No, Ticker_Code.

Portfolio(Portfolio_ID) -> Portfolio_ID.

Listings(Listing_No) -> Listing_No.

Listings(Ticker_Code) -> Ticker_Code.

Table is in BCNF.

CREATE TABLE GOES

(

Portfolio_ID INT NOT NULL,

Listing_No INT NOT NULL,

Ticker_Code VARCHAR(10) NOT NULL,

Alloted BIGINT NOT NULL,

PRIMARY KEY (Portfolio_ID,Listing_No, Ticker_Code),

FOREIGN KEY (Portfolio_ID) REFERENCES
PORTFOLIO(Portfolio_ID),

FOREIGN KEY (Listing_No, Ticker_Code) REFERENCES LISTINGS(Listing_No, Ticker_Code)

);

20. Owns:

Portfolio_ID	INT
Ticker_Code	VARCHAR(10)

Primary Key: (Portfolio_ID, Ticker_Code).

Foreign Key: Portfolio_ID, Ticker_Code.

Portfolio(Portfolio_ID) -> Portfolio_ID.

Stock(Ticker_Code) -> Ticker_Code.

Table is in BCNF.

CREATE TABLE OWNS

(

Portfolio_ID INT NOT NULL,

Ticker_Code VARCHAR(10) NOT NULL,

PRIMARY KEY (Portfolio_ID,Ticker_Code),

FOREIGN KEY (Portfolio_ID) REFERENCES PORTFOLIO(Portfolio_ID),

FOREIGN KEY (Ticker_Code) REFERENCES STOCK(Ticker_Code)

);

DATA INSERTION

```
INSERT INTO DBA (Login_ID, Password, Name, DOB, Phone_Number)
VALUES
(1001, 'securepass', 'Ajay Kumar', '1980-05-15', '9876543210'),
(1002, 'adminpass', 'Neha Sharma', '1985-09-20', '9876543211'),
(1003, 'root123', 'Rajesh Singh', '1990-11-10', '9876543212'),
(1004, 'sysadminpass', 'Pooja Patel', '1988-03-25', '9876543213'),
(1005, 'pass123', 'Amit Gupta', '1992-07-08', '9876543214');

INSERT INTO BROKER (Broker_ID, Password, Name, DOB, Phone_Number)
VALUES
(2001, 'brkr@123', 'Rahul Verma', '1982-08-18', '9876543220'),
(2002, 'sakshi@456', 'Sakshi Singh', '1987-10-25', '9876543221'),
(2003, 'anil@789', 'Anil Kumar', '1984-12-05', '9876543222'),
(2004, 'neha@xyz', 'Neha Sharma', '1989-04-15', '9876543223'),
(2005, 'raj@123', 'Raj Patel', '1986-06-30', '9876543224');

INSERT INTO PORTFOLIO (Portfolio_ID, Amount_Invested, Profit_Loss)
VALUES
(3001, 356000 , -13000),
(3002,307150, -72900),
(3003, 237500, 56000),
(3004, 635000, -421000),
(3005, 474000, -292000),
(3006, 178100, 41900),
(3007,191750, 32750);
```

```
INSERT INTO SECTOR (Sector_Name, Sector_Market_Cap, Sector_PE,
Sector_PB, Sector_Div_yield)
VALUES
('IT', 50000000000, 25.5, 4.5, 2.0),
('Automobile', 60000000000, 20.0, 3.8, 1.8),
('Pharmaceuticals', 40000000000, 30.0, 5.0, 2.5),
('Banking', 70000000000, 22.0, 3.2, 2.2),
('Telecom', 30000000000, 18.5, 3.0, 1.5);

INSERT INTO SECTOR_INDUSTRIES (Industries, Sector_Name)
VALUES
('Software', 'IT'),
('Software Service', 'IT'),
('Automobiles', 'Automobile'),
('Car Parts','Automobile'),
('Pharma Services', 'Pharmaceuticals'),
('Medicines','Pharmaceuticals'),
('Banks', 'Banking'),
('Telecommunication', 'Telecom');

INSERT INTO COMPANY (Corporate_ID, Company_Name, Headquarters, CEO,
Sector_Name)
VALUES
(5001, 'TCS', 'Mumbai', 'Rajesh Gopinathan', 'IT'),
(5002, 'Maruti Suzuki', 'New Delhi', 'Kenichi Ayukawa', 'Automobile'),
(5003, 'Sun Pharma', 'Mumbai', 'Dilip Shanghvi', 'Pharmaceuticals'),
(5004, 'HDFC Bank', 'Mumbai', 'Sashidhar Jagdishan', 'Banking'),
(5005, 'Bharti Airtel', 'New Delhi', 'Gopal Vittal', 'Telecom');
```

```
INSERT INTO COMPANY_FUNDAMENTALS (Corporate_ID, Income, Revenue, EBITDA, Debt, Dividend)
VALUES
(5001, 10000000000, 5000000000, 3000000000, 2000000000, 50000000),
(5002, 15000000000, 7500000000, 4500000000, 3000000000, 75000000),
(5003, 8000000000, 4000000000, 2000000000, 1500000000, 40000000),
(5004, 12000000000, 6000000000, 3500000000, 2500000000, 60000000),
(5005, 9000000000, 4500000000, 2500000000, 1800000000, 45000000);

INSERT INTO COMPANY_HELPLINES (Helpline_Number, Corporate_ID)
VALUES
('1800123456', 5001),
('1800456789', 5002),
('1800567890', 5003),
('1800789456', 5004),
('1800789457', 5004),
('1800987654', 5005),
('1800987655', 5005);

INSERT INTO STOCK (Ticker_Code, Corporate_ID)
VALUES
('TCS1', 5001),
('MARUTI1', 5002),
('SUNPHARMA1', 5003),
('HDFCBANK1', 5004),
('BHARTIART1', 5005),
('TCS2', 5001),
('MARUTI2', 5002),
```

```
('SUNPHARMA2', 5003),
('HDFCBANK2', 5004),
('BHARTIART2', 5005);

INSERT INTO COMMON (Ticker_Code, Float_Value, Outstanding, Market_Val,
Votes_Share, EPS, Dividend_Yield)
VALUES
('TCS1', 100000, 500000, 700, 10, 50.75, 2.0),
('MARUTI1', 250000, 400000, 600, 20, 35.50, 1.5),
('SUNPHARMA1', 20000, 30000, 4500, 10, 25.25, 1.0),
('HDFCBANK1', 35000, 60000, 9000, 15, 60.00, 2.5),
('BHARTIART1', 30000, 55000, 1800, 5, 55.75, 2.2);

INSERT INTO PREFERRED (Ticker_Code, Float_Value, Outstanding, Market_Val,
CallDate, IsCumulative, Par_Value_Year)
VALUES
('TCS2', 10000, 15000, 500, '2026-01-01', true, 5.00),
('MARUTI2', 40000, 80000, 400, '2025-02-01', false, 4.00),
('SUNPHARMA2', 6000, 10000, 4000, '2025-01-15', true, 3.00),
('HDFCBANK2', 1200, 5000, 7500, '2028-06-01', false, 6.00),
('BHARTIART2', 5000, 10000, 1000, '2030-01-01', true, 5.50);

INSERT INTO TECHNICALS (Ticker_Code, Low_All_Time, High_All_Time,
Open_Day, Close_Day, IPO_Price)
VALUES
('TCS1', 610, 720, 640, 700, 630),
('MARUTI1', 500, 600, 550, 600, 540),
('SUNPHARMA1', 4050, 4700, 4050, 4500, 4000),
('HDFCBANK1', 7000, 9000, 8000, 9000, 8100),
```

```
('BHARTIART1', 1600, 2000, 1650, 1800, 1620),
('TCS2', 400, 520, 470, 500, 450),
('MARUTI2', 320, 400, 380, 400, 360),
('SUNPHARMA2', 3500, 4500, 3200, 4000, 3600),
('HDFCBANK2', 6000, 8000, 6650, 7500, 6750),
('BHARTIART2', 700, 1010, 970, 1000, 900);

INSERT INTO STOCK_LISTED_IN (Listed_In, Ticker_Code)
VALUES
('NSE', 'TCS1'),
('BSE', 'TCS1'),
('BSE', 'MARUTI1'),
('NSE', 'SUNPHARMA1'),
('BSE', 'HDFCBANK1'),
('NSE', 'BHARTIART1'),
('BSE', 'MARUTI2'),
('NSE', 'SUNPHARMA2'),
('BSE', 'HDFCBANK2'),
('NSE', 'BHARTIART2');

INSERT INTO STOCK_INDICES (Indices, Ticker_Code)
VALUES
('NIFTY50', 'TCS1'),
('SENSEX', 'MARUTI1'),
('SENSEX', 'SUNPHARMA1'),
('NIFTY50', 'HDFCBANK1'),
('SENSEX', 'BHARTIART1'),
('SENSEX', 'MARUTI2'),
```

```
('SENSEX', 'SUNPHARMA2'),
('NIFTY50', 'HDFCBANK2'),
('SENSEX', 'BHARTIART2');

INSERT INTO CUSTOMER (Customer_ID, Password, Name, DOB,
Phone_Number, Portfolio_ID)
VALUES
(6001, 'sneha@123', 'Sneha Gupta', '1993-07-20', '9876543330', 3001),
(6002, 'rakesh@456', 'Rakesh Patel', '1990-04-15', '9876543331', 3002),
(6003, 'priya@789', 'Priyanka Sharma', '1988-10-05', '9876543332', 3003),
(6004, 'manish@987', 'Manish Kumar', '1995-01-25', '9876543333', 3004),
(6005, 'neha@xyz', 'Neha Verma', '1992-03-30', '9876543334', 3005),
(6006, 'amit@123', 'Amit Singh', '1994-09-12', '9876543335', 3006),
(6007, 'riya@456', 'Riya Gupta', '1991-06-18', '9876543336', 3007);

INSERT INTO USES (Customer_ID, Broker_ID)
VALUES
(6001, 2004),
(6001, 2005),
(6002, 2001),
(6003, 2005),
(6004, 2002),
(6005, 2003),
(6006, 2004),
(6006, 2001);
```

```
INSERT INTO EXCHANGE (Order_ID, Quantity, Price, Time, Buyer_Portfolio_ID,
Seller_Portfolio_ID, Ticker_Code)
VALUES
(7001, 100, 650, '2023-12-01', 3002, 3001, 'TCS1'),
(7002, 50, 350, '2024-02-20', 3003, 3002, 'MARUTI2'),
(7003, 200, 420, '2024-01-15', 3005, 3003, 'SUNPHARMA1'),
(7004, 150, 800, '2023-12-05', 3002, 3004, 'HDFCBANK1'),
(7005, 80, 850, '2022-12-05', 3004, 3005, 'BHARTIART2'),
(7006, 60, 100, '2021-08-12', 3001, 3003, 'SUNPHARMA1'),
(7007, 30, 700, '2023-11-05', 3001, 3007, 'TCS2'),
(7008, 20, 370, '2022-03-28', 3002, 3006, 'MARUTI1'),
(7009, 9, 750, '2021-09-17',  3002, 3007, 'MARUTI1'),
(7010,100, 450, '2023-05-09', 3006,3003, 'SUNPHARMA1'),
(7011, 40, 350, '2022-12-20', 3001, 3004, 'BHARTIART2'),
(7012, 100, 400, '2022-07-04', 3007,3005,'SUNPHARMA1'),
(7013,50, 400, '2021-02-14', 3003,3004, 'HDFCBANK1'),
(7014, 10, 1200 , '2023-10-30', 3007,3006,'BHARTIART1'),
(7015, 20, 1500, '2022-06-08', 3005, 3007, 'TCS2');

INSERT INTO LISTINGS (Listing_No, Price, Time, Ticker_Code, Quantity)
VALUES
(1, 630, '2023-11-30', 'TCS1', 50000),
(2, 680, '2024-01-30', 'TCS1', 50000),
(1, 360, '2024-01-15', 'MARUTI2', 40000),
(1, 200, '2023-10-30', 'SUNPHARMA1', 20000),
(1, 810, '2022-12-30', 'HDFCBANK1' , 35000),
(1, 900, '2021-06-30', 'BHARTIART2' , 5000),
(1, 650,'2022-07-20', 'MARUTI1', 20000),
```

```
(1,450,'2023-04-04','SUNPHARMA2',7000),
(1,700,'2021-05-06','HDFCBANK2', 16500),
(1,710,'2022-03-23','BHARTIART1',13000),
(2,550,'2024-08-19','MARUTI1',12000),
(1,500,'2024-03-30','TCS2',25000);

INSERT INTO GOES (Portfolio_ID, Listing_No, Ticker_Code, Alloted)
VALUES
(3001, 1, 'TCS1', 500),
(3002, 1, 'MARUTI2', 300),
(3003, 1, 'SUNPHARMA1', 1000),
(3004, 1, 'HDFCBANK1', 700),
(3005, 1, 'BHARTIART2', 400),
(3006, 1, 'BHARTIART1', 110),
(3007, 1, 'MARUTI1', 79),
(3006, 2, 'MARUTI1', 100),
(3007, 1, 'TCS2', 130);

INSERT INTO OWNS (Portfolio_ID, Ticker_Code)
VALUES
(3001, 'TCS1'),
(3001, 'BHARTIART2'),
(3001, 'TCS2'),
(3001, 'SUNPHARMA1'),
(3002,'MARUTI2'),
(3002,'TCS1'),
(3002,'HDFCBANK1'),
(3002,'MARUTI1'),
```

```
(3003, 'SUNPHARMA1'),
(3003,'MARUTI2'),
(3003,'HDFCBANK1'),
(3004,'HDFCBANK1'),
(3004,'BHARTIART2'),
(3005,'BHARTIART2'),
(3005,'SUNPHARMA1'),
(3005,'TCS2'),
(3006,'BHARTIART1'),
(3006,'MARUTI1'),
(3006,'SUNPHARMA1'),
(3007,'TCS2'),
(3007,'MARUTI1'),
(3007,'SUNPHARMA1'),
(3007,'BHARTIART1');
```

QUERIES

1.Top 5 Portfolios by Profit

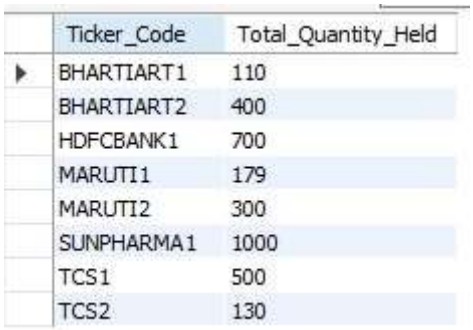
```
SELECT Portfolio_ID, Profit_Loss
FROM PORTFOLIO
ORDER BY Profit_Loss DESC
LIMIT 5;
```

Result Grid | Filter Rows

Portfolio_ID	Profit_Loss
3003	56000
3006	41900
3007	32750
3001	-13000
3002	-72900

2.Total Quantity Held for Each Ticker Code

```
SELECT O.Ticker_Code, SUM(G.Alloted) AS Total_Quantity_Held
FROM OWNS AS O
JOIN GOES AS G ON O.Portfolio_ID = G.Portfolio_ID AND O.Ticker_Code =
G.Ticker_Code
GROUP BY O.Ticker_Code;
```



Ticker_Code	Total_Quantity_Held
BHARTIART1	110
BHARTIART2	400
HDFCBANK1	700
MARUTI1	179
MARUTI2	300
SUNPHARMA1	1000
TCS1	500
TCS2	130

3.Portfolios Holding Stocks of Company X (here X = Sun Pharma)

```
SELECT DISTINCT Portfolio_ID
FROM OWNS
WHERE Ticker_Code IN (
   SELECT Ticker_Code
   FROM STOCK
   WHERE Corporate_ID = (
      SELECT Corporate_ID
      FROM COMPANY
      WHERE Company_Name = 'Sun Pharma'
   )
);
```

Portfolio_ID
3001
3003
3005
3006
3007

4.Distinct company names of the stocks owned by the portfolio with the ID 3001

```
SELECT DISTINCT c.Company_Name
FROM COMPANY c
JOIN STOCK s ON c.Corporate_ID = s.Corporate_ID
JOIN OWNS o ON s.Ticker_Code = o.Ticker_Code
WHERE o.Portfolio_ID = 3001;
```

Company_Name
Bharti Airtel
Sun Pharma
TCS

5.Current Price of Bharti Airtel Stocks

```
SELECT s.Ticker_Code, t.Close_Day AS Current_Price
FROM STOCK s
JOIN TECHNICALS t ON s.Ticker_Code = t.Ticker_Code
JOIN COMPANY c ON s.Corporate_ID = c.Corporate_ID
WHERE c.Company_Name = 'Bharti Airtel';
```

Ticker_Code	Current_Price
BHARTIART1	1800.00
BHARTIART2	1000.00

6.List all the companies traded on BSE and NSE both

```
SELECT com.Company_Name
FROM (
SELECT s.Corporate_ID,s.Ticker_Code
FROM STOCK s
JOIN STOCK_LISTED_IN sli1 ON s.Ticker_Code = sli1.Ticker_Code AND
sli1.Listed_In = 'BSE'
JOIN STOCK_LISTED_IN sli2 ON s.Ticker_Code = sli2.Ticker_Code AND
sli2.Listed_In = 'NSE'
) p
JOIN COMPANY com ON p.Corporate_ID = com.Corporate_ID;
```

Company_Name
TCS

7.Stocks Owned by Customer X (here X = Sneha Gupta)

```
SELECT DISTINCT s.Ticker_Code
FROM STOCK s
JOIN OWNS o ON s.Ticker_Code = o.Ticker_Code
JOIN CUSTOMER c ON o.Portfolio_ID = c.Portfolio_ID
WHERE c.Name = 'Sneha Gupta';
```

Ticker_Code
BHARTIART2
SUNPHARMA1
TCS1
TCS2

8.Which stock has the highest number of holders?

```
SELECT s.Ticker_Code, COUNT(o.Portfolio_ID) AS Holder_Count
FROM STOCK s
JOIN OWNS o ON s.Ticker_Code = o.Ticker_Code
GROUP BY s.Ticker_Code
ORDER BY Holder_Count DESC
LIMIT 1;
```

Ticker_Code	Holder_Count
SUNPHARMA1	5

9.Which stock grew the most from their Lowest price?

```
SELECT s.Ticker_Code, (MAX(t.Close_Day) - MIN(t.Low_All_Time)) AS Growth
FROM STOCK s
JOIN TECHNICALS t ON s.Ticker_Code = t.Ticker_Code
GROUP BY s.Ticker_Code
ORDER BY Growth DESC
LIMIT 1;
```

Ticker_Code	Growth
HDFCBANK1	2000.00

LEARNINGS AND OUTCOMES

Creating a stock database project prived to be a **valuable learning opportunity** in both database design and finance. Through this project, **we learnt the need of structuring schemas** to accurately represent real-world entities such as companies, stocks, customers, portfolios, exchanges, and transactions.

This **involved refining our data modeling skills** to identify entities, attributes, and relationships while organizing the database efficiently to avoid redundancy and errors. By **implementing data integrity measures** like primary and foreign keys, we ensured consistency in our database.

Additionally, we **improve our SQL skills** by writing queries for **tasks like data retrieval, filtering, and table joining**. We also **deepened our understanding of financial concepts**, including stock listing, trading, and market pricing. This project **helps us understand real-world database applications** in finance, preparing us for future industry roles. Overall, working on a stock database project integrates database management principles with practical finance applications, **enhancing our understanding in both fields.**

CONCLUSION

In conclusion, the development and implementation of the Stock Market database project have proven to be an invaluable lesson highlighting the **importance of planning and design**. Through meticulous planning, diligent execution, and iterative refinement, we believe we have successfully constructed a **comprehensive solution** that addresses the complexities and demands of the modern stock market landscape. This project's **design ensures normalization principles, entity-relationship modelling and data integrity** constraints to ensure an effective database. Moreover, we believe the **scalability and flexibility of the database design** allows it to be expandable to account for any intricacies uncovered.

Thank You.