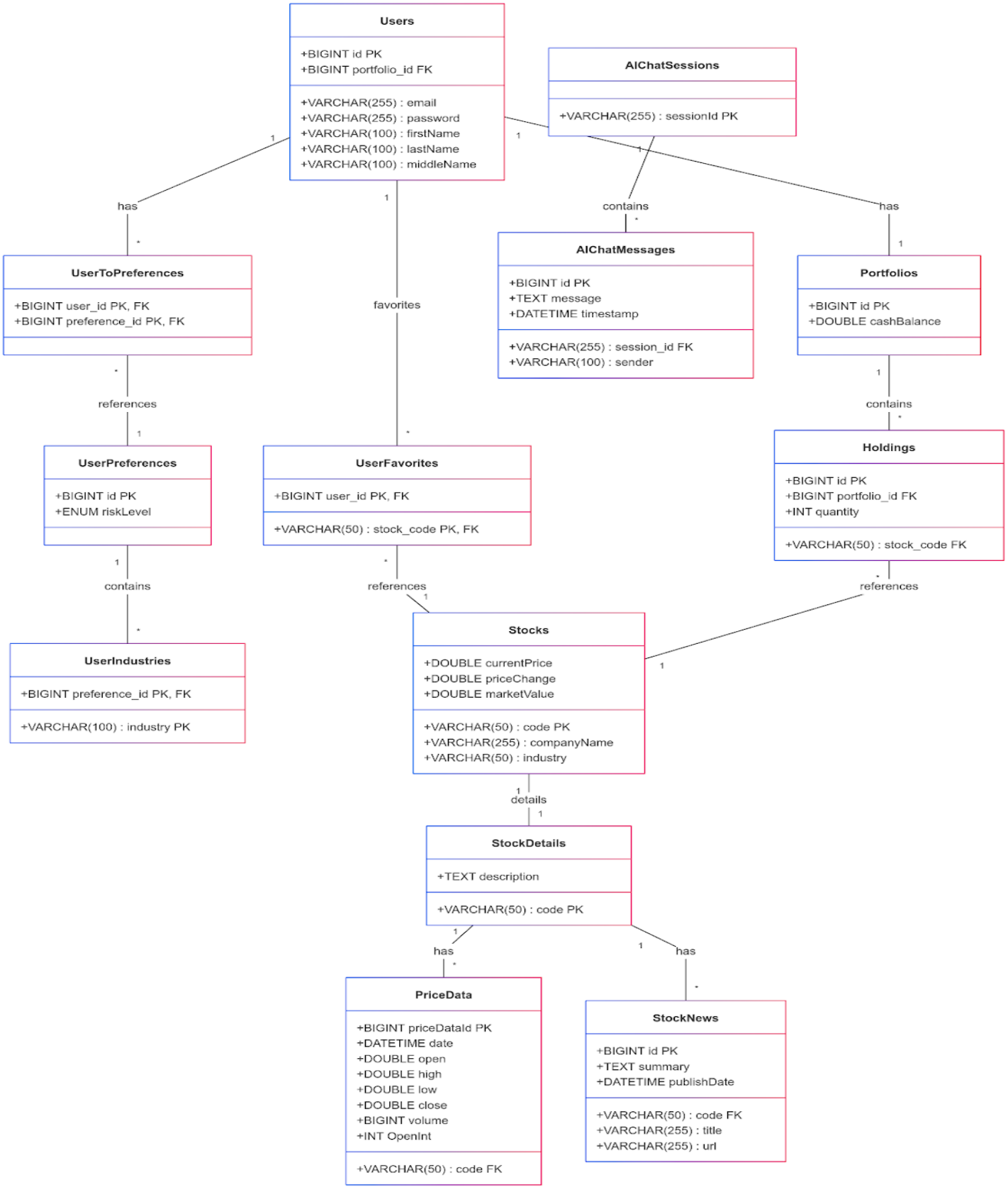
Milestone 3: Business Insights

T1: Yanyue Wang, Yixia Xu, Xiaoyan Cai, Fengying Zeng, Zichen Zhang, Liu Lu, Hanwen Guo, Yulin Cheng

Updated the UML:



1.What are the top 10 most valuable companies by market value?

```
SELECT code, companyName, marketValue
FROM stocks
ORDER BY marketValue DESC
LIMIT 10;
```

code	companyName	marketValue
amzn	Amazon.com, Inc.	1125350000
googl	Alphabet Inc. Class A	1044150000.0000001
bcr	Company BCR	331890000
tsla	Tesla, Inc.	302990000
nvda	NVIDIA Corporation	216140000
brk-b	Berkshire Hathaway Inc. Class B	183680000
asml	ASML Holding N.V.	179600000
fb	Facebook, Inc.	178460000
aapl	Apple Inc.	174670000
jnj	Johnson & Johnson	139560000
NULL	NULL	NULL

2.Which stocks have the highest market value in each industry?

```
SELECT
    s.industry,
    s.companyName,
    s.marketValue
FROM
    stocks s
WHERE
    s.marketValue = (
        SELECT MAX(s1.marketValue)
        FROM stocks s1
        WHERE s1.industry = s.industry
    );
```

industry	companyName	marketValue
Energy	Adaro Energy	7950000
Consumer Goods	Amazon.com, Inc.	1125350000
Financials	Banco de Chile	90040000
Materials	Balchem Corporation	82820000
Other	Company BCR	331890000
Finance	Berkshire Hathaway Inc. Class B	183680000
Communication Services	Alphabet Inc. Class A	1044150000.0000001
Healthcare	Johnson & Johnson	139560000
Technology	NVIDIA Corporation	216140000

3. Which users prefer 'Technology' and have the highest cash balance in their portfolio?

```
SELECT DISTINCT u.id, u.firstName, u.lastName, p.cashBalance
FROM users u
JOIN user_preferences up ON u.id = up.user_id
JOIN user_industries ui ON up.id = ui.preference_id
JOIN portfolios p ON u.id = p.users_id
WHERE ui.industry = 'Technology'
ORDER BY p.cashBalance DESC
LIMIT 5;
```

id	firstName	lastName	cashBalance
71	Frank	Jones	9649.23
98	Henry	Johnson	9313.12
80	Emma	Garcia	8357.83
77	Emma	Johnson	7514.97
89	Grace	Williams	6935.31

4. What is the most volatile stock based on the largest difference between high and low prices over the past 30 days?

```
SELECT code,
    MAX(high) AS max_price,
    MIN(low) AS min_price,
    (MAX(high) - MIN(low)) AS volatility
FROM price_data
WHERE date >= (SELECT MAX(date) FROM price_data) - INTERVAL 30 DAY
GROUP BY code
ORDER BY volatility DESC
LIMIT 1;
```

code	max_price	min_price	volatility
amzn	1135.54	962.5	173.03999999999996

5. Which stocks have a positive price trend over the last 7 days?

```
SELECT p1.code
FROM price_data p1
JOIN price_data p2
    ON p1.code = p2.code
WHERE p1.date = (SELECT MAX(date) FROM price_data)
  AND p2.date = (SELECT MAX(date) FROM price_data) - INTERVAL 7 DAY
  AND p1.close > p2.close;
```

code
aapl
abcd
amzn
baa
bce
bcrx
jpm_a
mc
nvda
v

6. Which stocks are most favored by users (frequently added to users' favorites)?

```
SELECT stock_code, COUNT(*) AS frequency
FROM user_favorites
GROUP BY stock_code
ORDER BY frequency DESC
LIMIT 10;
```

stock_code	frequency
nvda	15
msft	15
googl	14
tsla	13
aapl	12
fb	12
jnj	11
brk-b	11
v	11
adro	9

7. Which industries have the most users with preferences?

```
SELECT ui.industry, COUNT(DISTINCT ui.preference_id) AS num_users
FROM user_industries ui
GROUP BY ui.industry
ORDER BY num_users DESC
LIMIT 3;
```

industry	num_users
Finance	45
Materials	43
Energy	43

8. Which stocks are most popular among users in the 'Consumer Goods' industry?

```
SELECT uf.stock_code, COUNT(*) AS favorite_count
FROM user_favorites uf
JOIN stocks s ON uf.stock_code = s.code
WHERE s.industry = 'Consumer Goods'
GROUP BY uf.stock_code
ORDER BY favorite_count DESC
LIMIT 1;
```

stock_code	favorite_count
tsla	13

9. Which stocks in the 'Technology' industry have the highest price change in the last month?

```
SELECT s.code, s.companyName, s.priceChange
FROM stocks s
WHERE s.industry = 'Technology'
ORDER BY ABS(s.priceChange) DESC
LIMIT 5;
```

code	companyName	priceChange
nvda	NVIDIA Corporation	104.25999999999999
bcor	Blucora, Inc.	-91.83
asml	ASML Holding N.V.	67.72
aapl	Apple Inc.	62.78999999999999
msft	Microsoft Corporation	-28.00999999999999
NULL	NULL	NULL

10. List users with a 'LOW' risk preference who have a portfolio balance greater than $5,000.

```
SELECT u.firstName, u.lastName, p.cashBalance
FROM users u
JOIN user_preferences up ON u.id = up.user_id
JOIN portfolios p ON u.id = p.users_id
WHERE up.riskLevel = 'LOW' AND p.cashBalance > 5000;
```

firstName	lastName	cashBalance
Charlie	Miller	9240.78
Bob	Smith	7945.43
Frank	Martinez	6133.55
Alice	Jones	6181.22
Emma	Brown	6156.99
Emma	Brown	6156.99
Grace	Johnson	6953.15
Frank	Jones	6559.58
Emma	Brown	6881.03
Jack	Rodriguez	6306.91
Jack	Rodriguez	6306.91
David	Jones	7232.44
Alice	Williams	7241.49
Charlie	Jones	8179.19
Frank	Garcia	5409.11
Emma	Davis	8166.53
Emma	Davis	8166.53
Jack	Smith	9312.5
Jack	Williams	5697.14
Ivy	Williams	9679.12
David	Davis	7964.28
Emma	Johnson	7514.97
Emma	Johnson	7514.97
Frank	Garcia	5503.75
Emma	Davis	5722.56
Jack	Brown	9958.23
Jack	Brown	9958.23
Grace	Williams	6935.31
Jack	Davis	6962.12
David	Davis	8370.36
Ivy	Williams	9273.22
Henry	Johnson	9313.12
Grace	Johnson	8745.96

SQL Code :

```
CREATE SCHEMA IF NOT EXISTS StockRecommender;

DROP TABLE IF EXISTS ai_chat_messages;
DROP TABLE IF EXISTS ai_chat_sessions;
DROP TABLE IF EXISTS holdings;
DROP TABLE IF EXISTS user_favorites;
DROP TABLE IF EXISTS stock_news;
DROP TABLE IF EXISTS price_data;
DROP TABLE IF EXISTS stock_details;
DROP TABLE IF EXISTS user_industries;
DROP TABLE IF EXISTS user_preferences;
DROP TABLE IF EXISTS portfolios;
DROP TABLE IF EXISTS users;
DROP TABLE IF EXISTS stocks;

SELECT * FROM holdings;
SELECT * FROM stock_news;
SELECT * FROM ai_chat_sessions;
SELECT * FROM ai_chat_messages;

CREATE TABLE stocks (
  code VARCHAR(50) NOT NULL,
  companyName VARCHAR(255),
  currentPrice DOUBLE,
  priceChange DOUBLE,
  marketValue DOUBLE,
  industry VARCHAR(50),
  CONSTRAINT pk_stocks_code PRIMARY KEY (code)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE users (
  id BIGINT NOT NULL AUTO_INCREMENT,
  email VARCHAR(255),
  password VARCHAR(255),
  firstName VARCHAR(100),
  lastName VARCHAR(100),
  middleName VARCHAR(100),
  PRIMARY KEY (id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE portfolios (
  id BIGINT NOT NULL AUTO_INCREMENT,
  users_id BIGINT NOT NULL,
  cashBalance DOUBLE,
  PRIMARY KEY (id),
  FOREIGN KEY (users_id) REFERENCES users(id) on DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE user_preferences (
```

```
  id BIGINT NOT NULL AUTO_INCREMENT,
  user_id BIGINT NOT NULL,
  riskLevel ENUM('LOW', 'MEDIUM', 'HIGH') NOT NULL,
  #marketSector VARCHAR(100) NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE user_industries (
  preference_id BIGINT NOT NULL,
  industry VARCHAR(100) NOT NULL,
  PRIMARY KEY (preference_id, industry),
  FOREIGN KEY (preference_id) REFERENCES user_preferences(id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE user_favorites (
  user_id BIGINT NOT NULL,
  stock_code VARCHAR(50) NOT NULL,
  PRIMARY KEY (user_id, stock_code),
  FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
  FOREIGN KEY (stock_code) REFERENCES stocks(code) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE stock_details (
  #stockDetailsId BIGINT NOT NULL AUTO_INCREMENT,
  code VARCHAR(50) NOT NULL,
  description TEXT,
  CONSTRAINT pk_stock_details_code PRIMARY KEY (code)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE price_data (
  priceDataId BIGINT NOT NULL AUTO_INCREMENT,
  code VARCHAR(50), #codestock_detail_id BIGINT, #To store the filename as the stock code
  date DATETIME,
  open DOUBLE,
  high DOUBLE,
  low DOUBLE,
  close DOUBLE,
  volume BIGINT,
  OpenInt INT,
  CONSTRAINT pk_price_data_priceDataId PRIMARY KEY (priceDataId),
  CONSTRAINT fk_price_data_code FOREIGN KEY(code) REFERENCES stock_details(code)
  ON UPDATE CASCADE ON DELETE SET NULL
  #FOREIGN KEY (stock_detail_id) REFERENCES stock_details(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE stock_news (
  id BIGINT NOT NULL AUTO_INCREMENT,
  code VARCHAR(50), #stock_detail_id BIGINT,
  title VARCHAR(255),
  summary TEXT,
  url VARCHAR(255),
```

```
  publishDate DATETIME,
  PRIMARY KEY (id),
  FOREIGN KEY (code) REFERENCES stock_details(code) #id—>code
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE holdings (
  id BIGINT NOT NULL AUTO_INCREMENT,
  portfolio_id BIGINT,
  stock_code VARCHAR(50),
  quantity INT,
  PRIMARY KEY (id),
  FOREIGN KEY (portfolio_id) REFERENCES portfolios(id),
  FOREIGN KEY (stock_code) REFERENCES stocks(code)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE ai_chat_sessions (
  sessionId VARCHAR(255) NOT NULL,
  PRIMARY KEY (sessionId)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE ai_chat_messages (
  id BIGINT NOT NULL AUTO_INCREMENT,
  session_id VARCHAR(255),
  sender VARCHAR(100),
  message TEXT,
  timestamp DATETIME,
  PRIMARY KEY (id),
  FOREIGN KEY (session_id) REFERENCES ai_chat_sessions(sessionId)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Insert Data :

```
#STEP 2 INSERT DATA
# insert the selected stock details into table stock_details
INSERT INTO stock_details(code,description)
  VALUES('aapl','description1'),
  ('abcd','description2'),
  ('adro','description3'),
  ('amzn','description4'),
  ('asml','description5'),
  ('baa','description6'),
  ('bc','description7'),
  ('bcacr','description8'),
  ('bcc','description9'),
  ('bcd','description10'),
  ('bce','description11'),
  ('bcei','description12'),
  ('bch','description13'),
  ('bcor','description14'),
```

```
  ('bcpc','description15'),
  ('bcr','description16'),
  ('bcrh','description17'),
  ('bcrx','description18'),
  ('bcs_d','description19'),
  ('brk-b','description20'),
  ('corp','description21'),
  ('fb','description22'),
  ('googl','description23'),
  ('jnj','description24'),
  ('jpm_a','description25'),
  ('mc','description26'),
  ('msft','description27'),
  ('nvda','description28'),
  ('tsla','description29'),
  ('v','description30');

#Insert related data into stocks according to stock's price_data
INSERT INTO stocks (code, companyName, currentPrice, priceChange, marketValue, industry)
SELECT
    price_data.code,
    'Unknown CompanyName', -- You can update this with actual company names
    price_data.close AS currentPrice,
    (price_data.close - COALESCE((SELECT close FROM price_data WHERE code = price_data.code ORDER
BY date DESC LIMIT 1 OFFSET 1), price_data.close)) AS priceChange,
    price_data.close * 1000000 AS marketValue, #Assuming 1,000,000 outstanding shares (replace with real
data)
    'Unknown Industry'
FROM price_data
WHERE price_data.date = (SELECT MAX(date) FROM price_data WHERE code = price_data.code);

#update company name and industry
UPDATE stocks
SET companyName = CASE code
    WHEN 'aapl' THEN 'Apple Inc.'
    WHEN 'abcd' THEN 'Company ABCD'
    WHEN 'adro' THEN 'Adaro Energy'
    WHEN 'amzn' THEN 'Amazon.com, Inc.'
    WHEN 'asml' THEN 'ASML Holding N.V.'
    WHEN 'baa' THEN 'Company BAA'
    WHEN 'bc' THEN 'Brunswick Corporation'
    WHEN 'bcacr' THEN 'Company BCACR'
    WHEN 'bcc' THEN 'Boise Cascade Company'
    WHEN 'bcd' THEN 'Company BCD'
    WHEN 'bce' THEN 'BCE Inc.'
    WHEN 'bcei' THEN 'Company BCEI'
    WHEN 'bch' THEN 'Banco de Chile'
    WHEN 'bcor' THEN 'Blucora, Inc.'
    WHEN 'bcpc' THEN 'Balchem Corporation'
    WHEN 'bcr' THEN 'Company BCR'
    WHEN 'bcrh' THEN 'Blue Capital Reinsurance Holdings Ltd.'
    WHEN 'bcrx' THEN 'BioCryst Pharmaceuticals, Inc.'
```

```
    WHEN 'bcs_d' THEN 'Company BCS_D'
    WHEN 'brk-b' THEN 'Berkshire Hathaway Inc. Class B'
    WHEN 'corp' THEN 'Company CORP'
    WHEN 'fb' THEN 'Facebook, Inc.'
    WHEN 'googl' THEN 'Alphabet Inc. Class A'
    WHEN 'jnj' THEN 'Johnson & Johnson'
    WHEN 'jpm_a' THEN 'Company JPM_A'
    WHEN 'mc' THEN 'Moelis & Company'
    WHEN 'msft' THEN 'Microsoft Corporation'
    WHEN 'nvda' THEN 'NVIDIA Corporation'
    WHEN 'tsla' THEN 'Tesla, Inc.'
    WHEN 'v' THEN 'Visa Inc.'
    ELSE companyName
END,
industry = CASE code
    WHEN 'aapl' THEN 'Technology'
    WHEN 'abcd' THEN 'Other'
    WHEN 'adro' THEN 'Energy'
    WHEN 'amzn' THEN 'Consumer Goods'
    WHEN 'asml' THEN 'Technology'
    WHEN 'baa' THEN 'Other'
    WHEN 'bc' THEN 'Consumer Goods'
    WHEN 'bcacr' THEN 'Other'
    WHEN 'bcc' THEN 'Materials'
    WHEN 'bcd' THEN 'Other'
    WHEN 'bce' THEN 'Communication Services'
    WHEN 'bcei' THEN 'Other'
    WHEN 'bch' THEN 'Financials'
    WHEN 'bcor' THEN 'Technology'
    WHEN 'bcpc' THEN 'Materials'
    WHEN 'bcr' THEN 'Other'
    WHEN 'bcrh' THEN 'Finance'
    WHEN 'bcrx' THEN 'Healthcare'
    WHEN 'bcs_d' THEN 'Other'
    WHEN 'brk-b' THEN 'Finance'
    WHEN 'corp' THEN 'Other'
    WHEN 'fb' THEN 'Communication Services'
    WHEN 'googl' THEN 'Communication Services'
    WHEN 'jnj' THEN 'Healthcare'
    WHEN 'jpm_a' THEN 'Other'
    WHEN 'mc' THEN 'Finance'
    WHEN 'msft' THEN 'Technology'
    WHEN 'nvda' THEN 'Technology'
    WHEN 'tsla' THEN 'Consumer Goods'
    WHEN 'v' THEN 'Finance'
    ELSE industry
END
WHERE code IN ('aapl', 'abcd', 'adro', 'amzn', 'asml', 'baa', 'bc', 'bcacr',
               'bcc', 'bcd', 'bce', 'bcei', 'bch', 'bcor', 'bcpc', 'bcr',
               'bcrh', 'bcrx', 'bcs_d', 'brk-b', 'corp', 'fb', 'googl',
               'jnj', 'jpm_a', 'mc', 'msft', 'nvda', 'tsla', 'v');
```

```
SELECT * FROM users;
-- Insert 100 users with random names, emails, and link them to preferences and portfolios
SET @row_number = 0;
INSERT INTO users (email, password, firstName, lastName, middleName)
SELECT
   CONCAT('user', (@row_number := @row_number + 1), '@example.com') AS email,  -- Ensure unique user
emails
   MD5(CONCAT('password', @row_number)) AS password,  -- Ensure unique passwords
   ELT(1 + FLOOR(RAND() * 10), 'Alice', 'Bob', 'Charlie', 'David', 'Emma', 'Frank', 'Grace', 'Henry', 'Ivy',
'Jack') AS firstName,
   ELT(1 + FLOOR(RAND() * 10), 'Smith', 'Johnson', 'Brown', 'Williams', 'Jones', 'Garcia', 'Miller', 'Davis',
'Rodriguez', 'Martinez') AS lastName,
   IF(RAND() > 0.7, NULL, ELT(1 + FLOOR(RAND() * 5), 'Lee', 'Ann', 'James', 'Marie', 'Paul')) AS
middleName
   #user_pref.id,  -- Assign a valid preference_id
   #(SELECT id FROM portfolios ORDER BY RAND() LIMIT 1) AS portfolio_id  -- Randomly select a
portfolio for each user
FROM
        (SELECT @row_number := 0) AS init,  -- Initialize row counter
   (SELECT 1 UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION
SELECT 6 UNION SELECT 7 UNION SELECT 8 UNION SELECT 9 UNION SELECT 10) a,
   (SELECT 1 UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION
SELECT 6 UNION SELECT 7 UNION SELECT 8 UNION SELECT 9 UNION SELECT 10) b
LIMIT 100;

   #(SELECT id FROM user_preferences ORDER BY RAND() LIMIT 1),  -- Pick an existing preference_id
   #(SELECT id FROM portfolios ORDER BY RAND() LIMIT 1)        -- Pick an existing portfolio_id
#FROM (SELECT 1 AS id UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5)
AS temp  -- Assign alias 'id'
#LIMIT 100;

SELECT * FROM portfolios;
-- Insert 100 portfolios with random cash balances
INSERT INTO portfolios (users_id, cashBalance)
SELECT id, ROUND(RAND() * 10000, 2) AS cashBalance
FROM users;

SELECT * FROM user_preferences
ORDER BY user_id;
-- Insert 100 user preferences with random risk levels for each user
INSERT INTO user_preferences (user_id, riskLevel)
SELECT
   u.id AS user_id,  -- Get user_id from the users table
   ELT(1 + FLOOR(RAND() * 3), 'LOW', 'MEDIUM', 'HIGH') AS riskLevel  -- Randomly select a risk level
(LOW, MEDIUM, HIGH)
FROM users u
JOIN (SELECT 1 AS preference UNION SELECT 2 UNION SELECT 3) AS temp  -- Creates 3 entries per
user
ORDER BY RAND()  -- Randomize the order of user preferences
LIMIT 300;

SELECT * FROM user_industries
```

```
ORDER BY preference_id ASC;
-- -- Insert random industries into the user_industries table ensuring each preference_id has 3 unique industries.
-- Set counter variables to manage row numbering
-- SET @counter := 0;
-- SET @prev_preference_id := NULL;
INSERT INTO user_industries (preference_id, industry)
SELECT new_data.preference_id, new_data.industry
FROM (
   SELECT
      up.id AS preference_id,  -- Corrected alias for user_preferences
      ELT(1 + FLOOR(RAND() * 8),
         'Technology', 'Healthcare', 'Finance', 'Energy',
         'Communication Services', 'Consumer Goods', 'Materials', 'Other') AS industry
   FROM user_preferences up  -- Use 'up' as alias for user_preferences table
   ORDER BY RAND()  -- Randomize the order
   LIMIT 1000
) AS new_data
LEFT JOIN user_industries ui
   ON new_data.preference_id = ui.preference_id
   AND new_data.industry = ui.industry
WHERE ui.preference_id IS NULL;  -- Ensure no duplicates are inserted

-- INSERT INTO user_industries (preference_id, industry)
-- SELECT user_pref.id AS preference_id, industry
-- FROM (
--    SELECT
--       user_pref.id AS preference_id,
--       ELT(1 + FLOOR(RAND() * 8),
--          'Technology', 'Healthcare', 'Finance', 'Energy',
--          'Communication Services', 'Consumer Goods', 'Materials', 'Other') AS industry
--    FROM user_preferences user_pref
--    ORDER BY RAND()
--    LIMIT 100
-- ) AS new_data
-- LEFT JOIN user_industries ui
-- ON new_data.preference_id = ui.preference_id AND new_data.industry = ui.industry
-- WHERE ui.preference_id IS NULL;

-- Inserting data into user_favorites with stock codes and company names
INSERT INTO user_favorites (user_id, stock_code)
VALUES
   (1, 'aapl'),
   (1, 'amzn'),
   (1, 'msft'),
   (2, 'googl'),
   (2, 'tsla'),
   (2, 'v'),
   (3, 'nvda'),
   (3, 'fb'),
   (3, 'jnj'),
   (4, 'brk-b'),
   (4, 'bce'),
```

(4, 'bch'),
(5, 'bc'),
(5, 'adro'),
(5, 'asml'),
(6, 'aapl'),
(6, 'nvda'),
(6, 'fb'),
(7, 'msft'),
(7, 'tsla'),
(7, 'googl'),
(8, 'bcacr'),
(8, 'bcor'),
(8, 'bcc'),
(9, 'bce'),
(9, 'bcei'),
(9, 'bch'),
(10, 'bcrx'),
(10, 'bcpc'),
(10, 'bcr'),
(11, 'bcrh'),
(11, 'bcs_d'),
(11, 'brk-b'),
(12, 'corp'),
(12, 'jnj'),
(12, 'v'),
(13, 'tsla'),
(13, 'msft'),
(13, 'googl'),
(14, 'aapl'),
(14, 'nvda'),
(14, 'adro'),
(15, 'asml'),
(15, 'fb'),
(15, 'brk-b'),
(16, 'bce'),
(16, 'bcor'),
(16, 'msft'),
(17, 'bcacr'),
(17, 'bcrx'),
(17, 'bcei'),
(18, 'v'),
(18, 'jnj'),
(18, 'googl'),
(19, 'nvda'),
(19, 'aapl'),
(19, 'tsla'),
(20, 'bcpc'),
(20, 'bcr'),
(20, 'bch'),
(21, 'adro'),
(21, 'msft'),
(21, 'fb'),

(22, 'googl'),
(22, 'nvda'),
(22, 'brk-b'),
(23, 'jnj'),
(23, 'bce'),
(23, 'v'),
(24, 'corp'),
(24, 'bc'),
(24, 'bcei'),
(25, 'msft'),
(25, 'aapl'),
(25, 'fb'),
(26, 'tsla'),
(26, 'googl'),
(26, 'brk-b'),
(27, 'adro'),
(27, 'bcor'),
(27, 'nvda'),
(28, 'msft'),
(28, 'jnj'),
(28, 'v'),
(29, 'bce'),
(29, 'corp'),
(29, 'tsla'),
(30, 'googl'),
(30, 'brk-b'),
(30, 'nvda'),
(31, 'bcrx'),
(31, 'fb'),
(31, 'msft'),
(32, 'bcei'),
(32, 'jnj'),
(32, 'tsla'),
(33, 'aapl'),
(33, 'nvda'),
(33, 'bcor'),
(34, 'bcacr'),
(34, 'v'),
(34, 'bch'),
(35, 'adro'),
(35, 'msft'),
(35, 'googl'),
(36, 'fb'),
(36, 'brk-b'),
(36, 'jnj'),
(37, 'nvda'),
(37, 'corp'),
(37, 'tsla'),
(38, 'v'),
(38, 'aapl'),
(38, 'msft'),
(39, 'bc'),

(39, 'bcei'),
(39, 'brk-b'),
(40, 'adro'),
(40, 'googl'),
(40, 'nvda'),
(41, 'bcrx'),
(41, 'fb'),
(41, 'v'),
(42, 'msft'),
(42, 'aapl'),
(42, 'corp'),
(43, 'bce'),
(43, 'tsla'),
(43, 'bcor'),
(44, 'googl'),
(44, 'nvda'),
(44, 'brk-b'),
(45, 'fb'),
(45, 'jnj'),
(45, 'bch'),
(46, 'bce'),
(46, 'v'),
(46, 'corp'),
(47, 'nvda'),
(47, 'bcei'),
(47, 'adro'),
(48, 'tsla'),
(48, 'aapl'),
(48, 'msft'),
(49, 'bcacr'),
(49, 'bcrx'),
(49, 'googl'),
(50, 'v'),
(50, 'fb'),
(50, 'jnj'),
(51, 'msft'),
(51, 'aapl'),
(51, 'tsla'),
(52, 'adro'),
(52, 'nvda'),
(52, 'bcor'),
(53, 'googl'),
(53, 'bce'),
(53, 'brk-b'),
(54, 'aapl'),
(54, 'fb'),
(54, 'msft'),
(55, 'nvda'),
(55, 'v'),
(55, 'jnj'),
(56, 'bcacr'),
(56, 'bcei'),

```
    (56, 'bch'),
    (57, 'googl'),
    (57, 'tsla'),
    (57, 'adro'),
    (58, 'fb'),
    (58, 'nvda'),
    (58, 'brk-b'),
    (59, 'aapl'),
    (59, 'msft'),
    (59, 'googl'),
    (60, 'jnj'),
    (60, 'tsla'),
    (60, 'bcei');
```

Load Data:

```
#Load data into stock's price data

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/aapl.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'aapl',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/abcd.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'abcd',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;
```

```
LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/adro.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'adro',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

#Load all selected stock txt data into price data
LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/amzn.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'amzn',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/asml.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'asml',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/baa.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
```

```
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'baa',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bc.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'bc',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bcacr.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'bcacr',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bcc.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
```

```
    code = 'bcc',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bcd.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'bcd',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bce.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'bce',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bcei.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'bcei',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
```

```
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bch.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'bch',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bcor.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'bcor',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bcpc.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'bcpc',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;
```

```
LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bcr.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'bcr',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bcrh.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'bcrh',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bcrx.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'bcrx',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/bcs_d.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
```

```
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'bcs_d',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/brk-b.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'brk-b',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/corp.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'corp',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/fb.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
```

```
    code = 'fb',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/googl.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'googl',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/jnj.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'jnj',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/jpm_a.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'jpm_a',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
```

```
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/mc.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'mc',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/msft.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'msft',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/nvda.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'nvda',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;
```

```
LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/tsla.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'tsla',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;

LOAD DATA LOCAL INFILE '/Users/anne/Desktop/data/v.us.txt'
INTO TABLE price_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS -- Skip the header row
(@date, @open, @high, @low, @close, @volume, @openInt)
SET
    code = 'v',  -- Manually setting the stock code from the filename
    date = STR_TO_DATE(@date, '%Y-%m-%d'),
    open = @open,
    high = @high,
    low = @low,
    close = @close,
    volume = @volume,
    openInt = @openInt;
```