

**FOUNDATION FOR ORGANIZATION RESEARCH AND EDUCATION (FORE) SCHOOL OF MANAGEMENT**

**NEW DELHI**

**Academic Year 2023 – 25**

**BIG DATA FOR MANAGERS (BDDA – 2)**

**PROGECT REPORT**

**Submitted to: Prof. AMARNATH MITRA**

**Submitted by:**

**SUBHAJIT PAUL**

**ROLL: 321173**

**SEC: C**

**Normal Forms in Database Design**

Normal forms are a set of rules used to design relational databases to minimize data redundancy and anomalies. They help ensure data integrity and consistency. Let's explore the normal forms up to Boyce-Codd Normal Form (BCNF).

**First Normal Form (1NF)**

* **Rule:** Each attribute in a relation should be atomic (indivisible).
* **Violation:** A single attribute contains multiple values (e.g., storing multiple phone numbers in a single field).
* **Example:** If a "Customer" table has a field "Address" that stores both street address and city, it violates 1NF.

**Second Normal Form (2NF)**

* **Rule:** A relation is in 2NF if it is in 1NF and every non-key attribute is fully dependent on the primary key.
* **Violation:** A non-key attribute is dependent on only a part of the primary key (partial dependency).
* **Example:** If a "Order" table has fields "Order ID," "Customer ID," and "Customer Name," and "Customer Name" depends only on "Customer ID," it violates 2NF.

**Third Normal Form (3NF)**

* **Rule:** A relation is in 3NF if it is in 2NF and every non-key attribute is non-transitively dependent on the primary key.
* **Violation:** A non-key attribute is dependent on another non-key attribute (transitive dependency).
* **Example:** If a "Employee" table has fields "Employee ID," "Department ID," and "Department Name," and "Department Name" depends on "Department ID" (which is a non-key attribute), it violates 3NF.

**Boyce-Codd Normal Form (BCNF)**

* **Rule:** A relation is in BCNF if every determinant is a superkey.
* **Determinant:** A set of attributes that determine other attributes.
* **Superkey:** A set of attributes that uniquely identify a tuple.
* **Violation:** A non-key attribute determines a superkey.
* **Example:** If a "Product" table has fields "Product ID," "Supplier ID," and "Supplier Name," and "Supplier ID" determines "Supplier Name" (which is a superkey), it violates BCNF.

**In summary:**

* 1NF ensures atomic attributes.
* 2NF eliminates partial dependencies.
* 3NF eliminates transitive dependencies.
* BCNF is a stricter form of 3NF, requiring that every determinant be a superkey.

By adhering to these normal forms, you can create well-structured databases that are more efficient, maintainable, and less prone to data anomalies.

**Analyzing the Normal Forms of the Tables**

**1. Investor**

* **1NF:** Yes, all attributes contain atomic values.
* **2NF:** Yes, all non-key attributes (Name, Email, RiskToleranceLevel, InvestmentGoals) are fully dependent on the primary key (InvestorID).
* **3NF:** Yes, there are no transitive dependencies.
* **BCNF:** Yes, since the table is in 3NF and there are no non-trivial functional dependencies.

**2. Portfolio**

* **1NF:** Yes, all attributes contain atomic values.
* **2NF:** Yes, all non-key attributes (Name, InceptionDate, Value) are fully dependent on the primary key (PortfolioID).
* **3NF:** Yes, there are no transitive dependencies.
* **BCNF:** Yes, since the table is in 3NF and there are no non-trivial functional dependencies.

**3. AssetClass**

* **1NF:** Yes, all attributes contain atomic values.
* **2NF:** Yes, all non-key attributes (Name, Description) are fully dependent on the primary key (AssetClassID).
* **3NF:** Yes, there are no transitive dependencies.
* **BCNF:** Yes, since the table is in 3NF and there are no non-trivial functional dependencies.

**4. Asset**

* **1NF:** Yes, all attributes contain atomic values.
* **2NF:** Yes, all non-key attributes (Name, TickerSymbol, CurrentPrice) are fully dependent on the primary key (AssetID).
* **3NF:** Yes, there are no transitive dependencies.
* **BCNF:** Yes, since the table is in 3NF and there are no non-trivial functional dependencies.

**5. PortfolioHolding**

* **1NF:** Yes, all attributes contain atomic values.
* **2NF:** Yes, all non-key attributes (Quantity, PurchasePrice, PurchaseDate) are fully dependent on the composite primary key (PortfolioID, AssetID).
* **3NF:** Yes, there are no transitive dependencies.
* **BCNF:** Yes, since the table is in 3NF and there are no non-trivial functional dependencies.

**6. Transaction**

* **1NF:** Yes, all attributes contain atomic values.
* **2NF:** Yes, all non-key attributes (Quantity, Price, Date) are fully dependent on the composite primary key (TransactionID).
* **3NF:** Yes, there are no transitive dependencies.
* **BCNF:** Yes, since the table is in 3NF and there are no non-trivial functional dependencies.

**7. Benchmark**

* **1NF:** Yes, all attributes contain atomic values.
* **2NF:** Yes, all non-key attributes (Name, Description) are fully dependent on the primary key (BenchmarkID).
* **3NF:** Yes, there are no transitive dependencies.
* **BCNF:** Yes, since the table is in 3NF and there are no non-trivial functional dependencies.

**8. PortfolioBenchmark**

* **1NF:** Yes, all attributes contain atomic values.
* **2NF:** Yes, all non-key attributes are fully dependent on the composite primary key (PortfolioID, BenchmarkID).
* **3NF:** Yes, there are no transitive dependencies.
* **BCNF:** Yes, since the table is in 3NF and there are no non-trivial functional dependencies.

**Conclusion**

All of the tables in your database design appear to be in Boyce-Codd Normal Form (BCNF). This is a good indication that your database is well-normalized and has minimal redundancy and anomalies.

**CRUD Operation in Tables**

* 1. **Investor**

**-- Create Investor table**

CREATE TABLE Investor (

InvestorID INT PRIMARY KEY,

Name VARCHAR(255) NOT NULL,

Email VARCHAR(255) UNIQUE NOT NULL,

RiskToleranceLevel ENUM('Low', 'Medium', 'High') NOT NULL,

InvestmentGoals VARCHAR(255)

);

**--Insert Values**

INSERT INTO Investor (InvestorID, Name, Email, RiskToleranceLevel, InvestmentGoals)

VALUES

(100, 'Aryan Sharma', 'aryan.sharma@email.com', 'Medium', 'Retirement'),

(101, 'Isha Patel', 'isha.patel@email.com', 'Low', 'Education'),

(102, 'Rahul Verma', 'rahul.verma@email.com', 'High', 'Business Growth'),

(103, 'Priya Gupta', 'priya.gupta@email.com', 'Medium', 'Property Purchase'),

(104, 'Rajesh Kumar', 'rajesh.kumar@email.com', 'Low', 'Savings'),

(105, 'Neha Singh', 'neha.singh@email.com', 'High', 'Startup Investment'),

(106, 'Amit Kapoor', 'amit.kapoor@email.com', 'Medium', 'Travel'),

(107, 'Anjali Reddy', 'anjali.reddy@email.com', 'Low', 'Childrens Education'),

(108, 'Vijay Patel', 'vijay.patel@email.com', 'High', 'Luxury Purchase'),

(109, 'Sona Kaur', 'sona.kaur@email.com', 'Medium', 'Financial Independence'),

(110, 'Vikram Singh', 'vikram.singh@email.com', 'Low', 'Emergency Fund'),

(111, 'Maya Sharma', 'maya.sharma@email.com', 'High', 'Real Estate Investment'),

(112, 'Deepak Kumar', 'deepak.kumar@email.com', 'Medium', 'Debt Repayment'),

(113, 'Anika Rao', 'anika.rao@email.com', 'Low', 'Health Care'),

(114, 'Sanjay Gupta', 'sanjay.gupta@email.com', 'High', 'Philanthropy'),

(115, 'Meera Patel', 'meera.patel@email.com', 'Medium', 'Early Retirement'),

(116, 'Kabir Khan', 'kabir.khan@email.com', 'Low', 'Wedding Expenses'),

(117, 'Rhea Kapoor', 'rhea.kapoor@email.com', 'High', 'Art Collection'),

(118, 'Aditya Verma', 'aditya.verma@email.com', 'Medium', 'Childrens Future'),

(119, 'Riya Singh', 'riya.singh@email.com', 'Low', 'Down Payment on House'),

(120, 'Arjun Reddy', 'arjun.reddy@email.com', 'High', 'Technology Stocks'),

(121, 'Pooja Sharma', 'pooja.sharma@email.com', 'Medium', 'Career Development'),

(122, 'Rohan Kumar', 'rohan.kumar@email.com', 'Low', 'Long-term Investment'),

(123, 'Nisha Patel', 'nisha.patel@email.com', 'High', 'Venture Capital'),

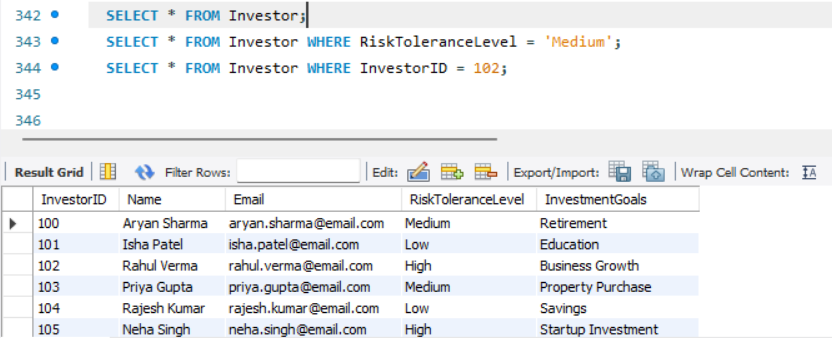
(124, 'Mira Mehta', 'mira.mehta@email.com', 'Medium', 'Retirement');

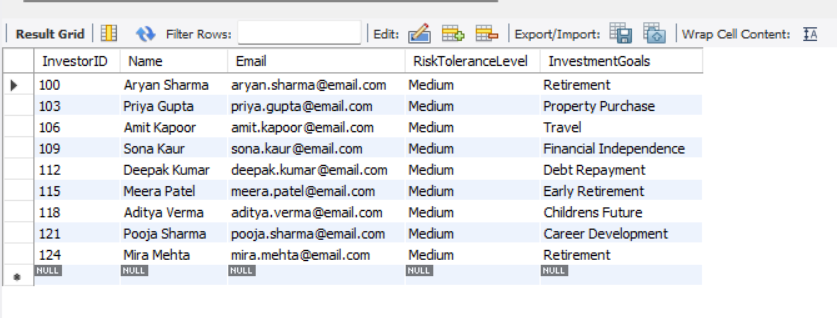
**--Read Values**

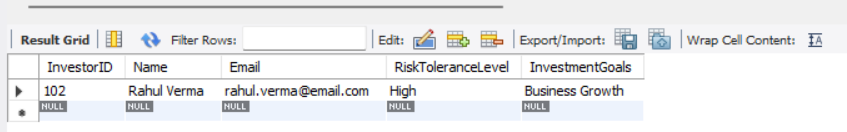
SELECT \* FROM Investor;

SELECT \* FROM Investor WHERE RiskToleranceLevel = 'Medium';

SELECT \* FROM Investor WHERE InvestorID = 102;





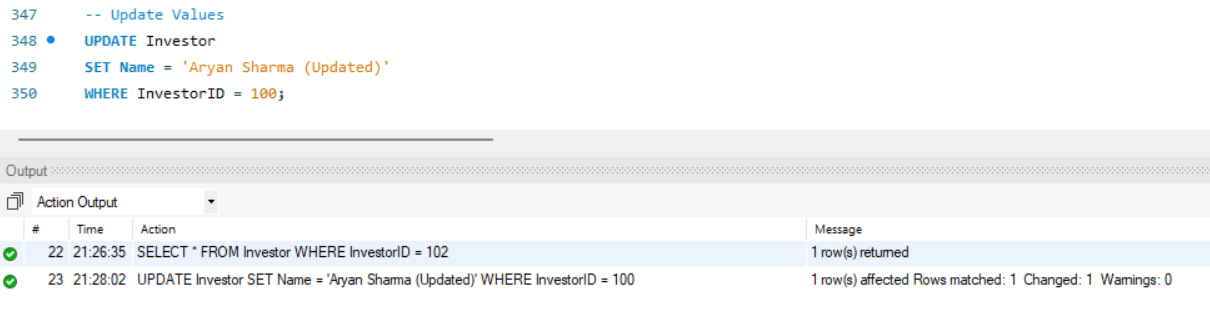


**-- Update Values**

UPDATE Investor

SET Name = 'Aryan Sharma (Updated)'

WHERE InvestorID = 100;



**--Delete Values**

DELETE FROM Investor WHERE InvestorID = 110;

21:29:42 DELETE FROM Investor WHERE InvestorID = 110 Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`wealth\_builder\_compass`.`portfolio`, CONSTRAINT `portfolio\_ibfk\_1` FOREIGN KEY (`InvestorID`) REFERENCES `investor` (`InvestorID`)) 0.000 sec

* 1. **Portfolio**

**-- Create Portfolio table**

CREATE TABLE Portfolio (

PortfolioID INT PRIMARY KEY,

InvestorID INT,

Name VARCHAR(255) NOT NULL,

InceptionDate DATE NOT NULL,

Value DECIMAL(15,2),

FOREIGN KEY (InvestorID) REFERENCES Investor(InvestorID)

);

**--Insert Values**

INSERT INTO Investor (InvestorID, Name, Email, RiskToleranceLevel, InvestmentGoals)

VALUES

(100, 'Aryan Sharma', 'aryan.sharma@email.com', 'Medium', 'Retirement'),

(101, 'Isha Patel', 'isha.patel@email.com', 'Low', 'Education'),

(102, 'Rahul Verma', 'rahul.verma@email.com', 'High', 'Business Growth'),

(103, 'Priya Gupta', 'priya.gupta@email.com', 'Medium', 'Property Purchase'),

(104, 'Rajesh Kumar', 'rajesh.kumar@email.com', 'Low', 'Savings'),

(105, 'Neha Singh', 'neha.singh@email.com', 'High', 'Startup Investment'),

(106, 'Amit Kapoor', 'amit.kapoor@email.com', 'Medium', 'Travel'),

(107, 'Anjali Reddy', 'anjali.reddy@email.com', 'Low', 'Childrens Education'),

(108, 'Vijay Patel', 'vijay.patel@email.com', 'High', 'Luxury Purchase'),

(109, 'Sona Kaur', 'sona.kaur@email.com', 'Medium', 'Financial Independence'),

(110, 'Vikram Singh', 'vikram.singh@email.com', 'Low', 'Emergency Fund'),

(111, 'Maya Sharma', 'maya.sharma@email.com', 'High', 'Real Estate Investment'),

(112, 'Deepak Kumar', 'deepak.kumar@email.com', 'Medium', 'Debt Repayment'),

(113, 'Anika Rao', 'anika.rao@email.com', 'Low', 'Health Care'),

(114, 'Sanjay Gupta', 'sanjay.gupta@email.com', 'High', 'Philanthropy'),

(115, 'Meera Patel', 'meera.patel@email.com', 'Medium', 'Early Retirement'),

(116, 'Kabir Khan', 'kabir.khan@email.com', 'Low', 'Wedding Expenses'),

(117, 'Rhea Kapoor', 'rhea.kapoor@email.com', 'High', 'Art Collection'),

(118, 'Aditya Verma', 'aditya.verma@email.com', 'Medium', 'Childrens Future'),

(119, 'Riya Singh', 'riya.singh@email.com', 'Low', 'Down Payment on House'),

(120, 'Arjun Reddy', 'arjun.reddy@email.com', 'High', 'Technology Stocks'),

(121, 'Pooja Sharma', 'pooja.sharma@email.com', 'Medium', 'Career Development'),

(122, 'Rohan Kumar', 'rohan.kumar@email.com', 'Low', 'Long-term Investment'),

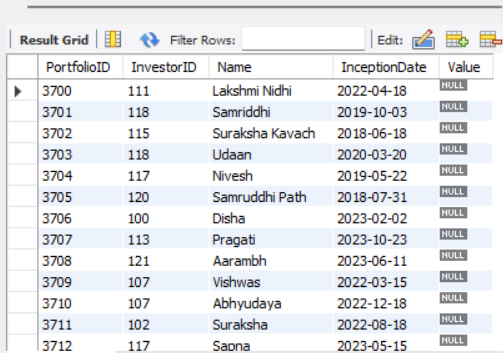
(123, 'Nisha Patel', 'nisha.patel@email.com', 'High', 'Venture Capital'),

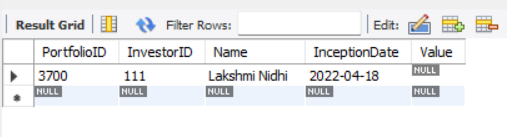
(124, 'Mira Mehta', 'mira.mehta@email.com', 'Medium', 'Retirement');

**--Read Values**

SELECT \* FROM Portfolio;

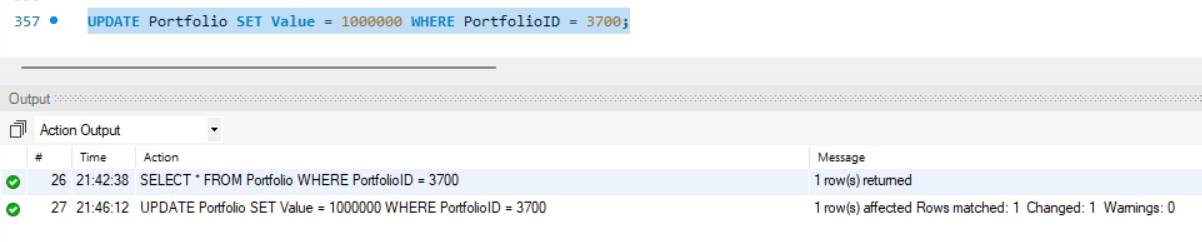
SELECT \* FROM Portfolio WHERE PortfolioID = 3700;





**-- Update Values**

UPDATE Portfolio SET Value = 1000000 WHERE PortfolioID = 3700;



**--Delete Values**

DELETE FROM Portfolio WHERE PortfolioID = 3701;

21:46:57 DELETE FROM Portfolio WHERE PortfolioID = 3701 Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`wealth\_builder\_compass`.`portfolioholding`, CONSTRAINT `portfolioholding\_ibfk\_1` FOREIGN KEY (`PortfolioID`) REFERENCES `portfolio` (`PortfolioID`)) 0.032 sec

* 1. **AssetClass**

**-- Create Asset table**

CREATE TABLE AssetClass (

AssetClassID INT PRIMARY KEY,

Name VARCHAR(255) NOT NULL,

Description TEXT

);

**--Insert Values**

INSERT INTO AssetClass (AssetClassID, Name, Description)

VALUES

(10, 'Stocks', 'Shares of ownership in a company'),

(11, 'Bonds', 'Debt instruments issued by companies or governments'),

(12, 'Mutual Funds', 'Pooled investment funds that invest in a variety of assets'),

(13, 'ETFs', 'Exchange-traded funds that track a specific index or sector'),

(14, 'Real Estate', 'Property investments, including residential and commercial'),

(15, 'Commodities', 'Raw materials like gold, oil, or agricultural products'),

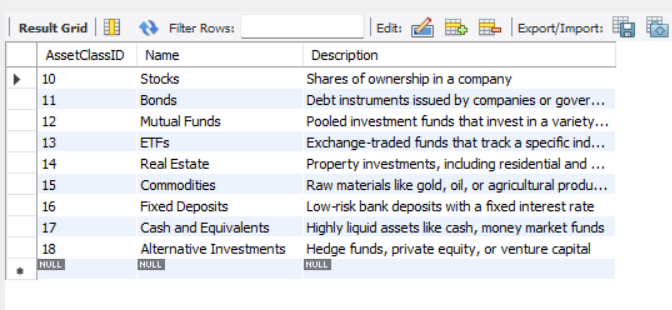
(16, 'Fixed Deposits', 'Low-risk bank deposits with a fixed interest rate'),

(17, 'Cash and Equivalents', 'Highly liquid assets like cash, money market funds'),

(18, 'Alternative Investments', 'Hedge funds, private equity, or venture capital');

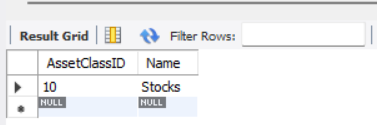
**-- Read (Select) all data from the AssetClass table**

SELECT \* FROM AssetClass;



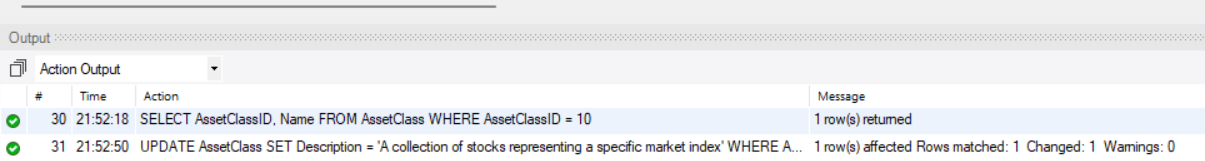
**-- Read (Select) specific data from the AssetClass table based on conditions**

SELECT AssetClassID, Name FROM AssetClass WHERE AssetClassID = 10;



**-- Update a specific record in the AssetClass table**

UPDATE AssetClass SET Description = 'A collection of stocks representing a specific market index' WHERE AssetClassID = 13;



**-- Delete a specific record from the AssetClass table**

DELETE FROM AssetClass WHERE AssetClassID = 15;

21:53:27 DELETE FROM AssetClass WHERE AssetClassID = 15 Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`wealth\_builder\_compass`.`asset`, CONSTRAINT `asset\_ibfk\_1` FOREIGN KEY (`AssetClassID`) REFERENCES `assetclass` (`AssetClassID`)) 0.015 sec

* 1. **Asset**

**-- Create Asset table**

CREATE TABLE Asset (

AssetID INT PRIMARY KEY,

AssetClassID INT,

Name VARCHAR(255) NOT NULL,

TickerSymbol VARCHAR(10),

CurrentPrice DECIMAL(15,2),

FOREIGN KEY (AssetClassID) REFERENCES AssetClass(AssetClassID)

);

**-- Insert Values**

INSERT INTO Asset (AssetID, AssetClassID, Name, TickerSymbol, CurrentPrice)

VALUES

-- Stocks

(2000, 10, 'Reliance Industries', 'RELIANCE', 2500.50),

(2001, 10, 'HDFC Bank', 'HDFCBANK', 1600.25),

(2002, 10, 'Infosys', 'INFY', 1450.75),

(2003, 10, 'Tata Consultancy Services', 'TCS', 3300.00),

-- Bonds

(2004, 11, 'Government Bond 2030', 'GOVT2030', 102.50),

(2005, 11, 'SBI Corporate Bond', 'SBICORP', 98.75),

-- Mutual Funds

(2006, 12, 'Axis Bluechip Fund', 'AXISBLUE', 55.20),

(2007, 12, 'HDFC Balanced Advantage Fund', 'HDFCBAL', 48.95),

(2008, 12, 'SBI Small Cap Fund', 'SBISMALL', 62.30),

-- ETFs

(2009, 13, 'NIFTY 50 ETF', 'NIFTYBEES', 180.50),

(2010, 13, 'SENSEX ETF', 'SETFNIF50', 375.00),

-- Real Estate (Indicative Prices)

(2011, 14, 'Apartment in Mumbai', 'MUMAPT', 15000000),

(2012, 14, 'Land in Bangalore', 'BLRLAND', 8000000),

-- Commodities (Indicative Prices)

(2013, 15, 'Gold', 'GOLD', 50000),

(2014, 15, 'Crude Oil', 'CRUDE', 80),

-- Fixed Deposits

(2015, 16, 'SBI Fixed Deposit', 'SBIFD', NULL), -- Interest rate varies

-- Cash and Equivalents

(2016, 17, 'Savings Account', 'SAVINGS', NULL),

(2017, 17, 'Liquid Fund', 'LIQUIDF', 10.00), -- NAV (Net Asset Value)

-- Alternative Investments

(2018, 18, 'Private Equity Fund A', 'PEFUNDA', NULL), -- Illiquid, value not readily available

(2019, 18, 'Hedge Fund B', 'HEDGEB', NULL);

SET SQL\_SAFE\_UPDATES = 0;

UPDATE Asset

SET CurrentPrice = CASE

WHEN AssetClassID = 16 THEN 6.5 -- Fixed Deposits (Indicative Interest Rate)

WHEN AssetClassID = 17 AND AssetID = 2016 THEN 1000000 -- Savings Account Balance (Example)

WHEN AssetClassID = 17 AND AssetID = 2017 THEN 10.00 -- Liquid Fund NAV

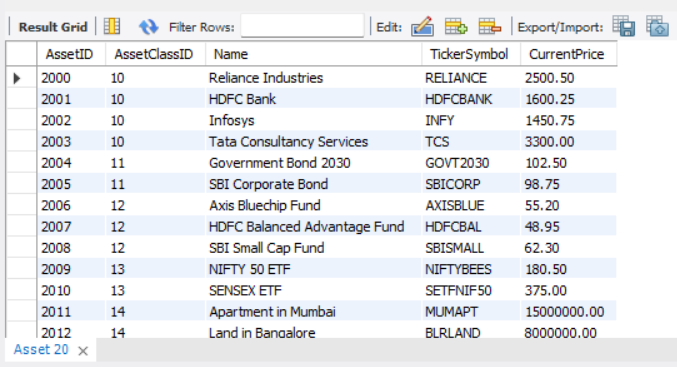
WHEN AssetClassID = 18 THEN 500000 -- Alternative Investments (Estimated Value)

END

WHERE AssetID IN (2015, 2016, 2017, 2018, 2019);

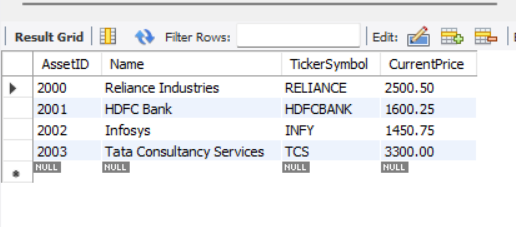
**-- Read (Select) all data from the Asset table**

SELECT \* FROM Asset;



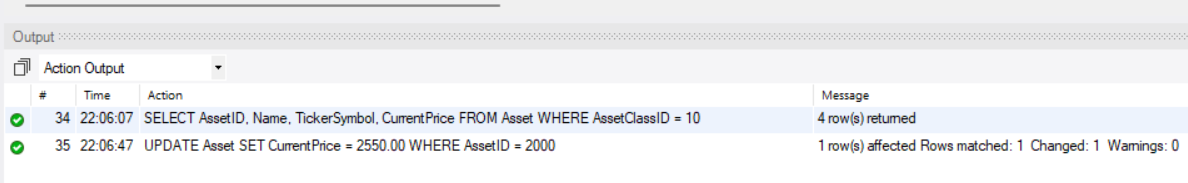
**-- Read (Select) specific data from the Asset table based on conditions**

SELECT AssetID, Name, TickerSymbol, CurrentPrice FROM Asset WHERE AssetClassID = 10;



**-- Update a specific record in the Asset table**

UPDATE Asset SET CurrentPrice = 2550.00 WHERE AssetID = 2000;



**-- Delete a specific record from the Asset table**

DELETE FROM Asset WHERE AssetID = 2001;

22:07:11 DELETE FROM Asset WHERE AssetID = 2001 Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`wealth\_builder\_compass`.`portfolioholding`, CONSTRAINT `portfolioholding\_ibfk\_2` FOREIGN KEY (`AssetID`) REFERENCES `asset` (`AssetID`)) 0.016 sec

* 1. **PortfolioHolding**

**-- Create PortfolioHolding table**

CREATE TABLE PortfolioHolding (

PortfolioID INT,

AssetID INT,

Quantity DECIMAL(15,2) NOT NULL,

PurchasePrice DECIMAL(15,2) NOT NULL,

PurchaseDate DATE NOT NULL,

PRIMARY KEY (PortfolioID, AssetID),

FOREIGN KEY (PortfolioID) REFERENCES Portfolio(PortfolioID),

FOREIGN KEY (AssetID) REFERENCES Asset(AssetID)

);

**-- Insert Values**

INSERT INTO PortfolioHolding (PortfolioID, AssetID, Quantity, PurchasePrice, PurchaseDate)

VALUES

(3701, 2000, 100, 2450.00, '2023-05-15'),

(3702, 2001, 50, 1550.00, '2022-08-20'),

(3703, 2002, 75, 1380.00, '2021-11-10'),

(3704, 2003, 120, 3200.00, '2020-02-25'),

(3705, 2004, 200, 101.50, '2023-03-08'),

(3706, 2005, 150, 97.25, '2022-06-12'),

(3707, 2006, 80, 54.00, '2021-09-20'),

(3708, 2007, 100, 47.50, '2020-12-30'),

(3709, 2008, 60, 61.00, '2023-04-18'),

(3710, 2009, 150, 178.00, '2022-07-25'),

(3711, 2010, 80, 370.00, '2021-10-15'),

(3712, 2011, 2, 14800000.00, '2023-01-20'),

(3713, 2012, 3, 7800000.00, '2022-04-28'),

(3714, 2013, 100, 49000.00, '2021-08-12'),

(3715, 2014, 50, 78.00, '2020-11-23'),

(3716, 2015, 1000000, 6.50, '2023-02-01'),

(3717, 2016, 500000, 1000000.00, '2022-05-18'),

(3718, 2017, 200, 9.80, '2021-09-02'),

(3719, 2018, 100, 500000.00, '2020-12-15'),

(3720, 2019, 50, 500000.00, '2023-03-22'),

(3721, 2000, 80, 2400.00, '2022-07-10'),

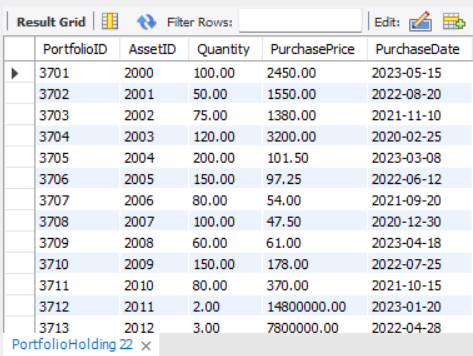
(3722, 2001, 120, 1520.00, '2021-10-25'),

(3723, 2002, 50, 1400.00, '2020-01-30'),

(3724, 2003, 100, 3150.00, '2023-05-05');

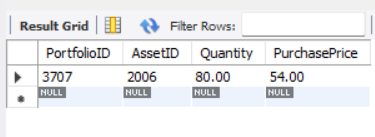
**-- Read (Select) all data from the PortfolioHolding table**

SELECT \* FROM PortfolioHolding;



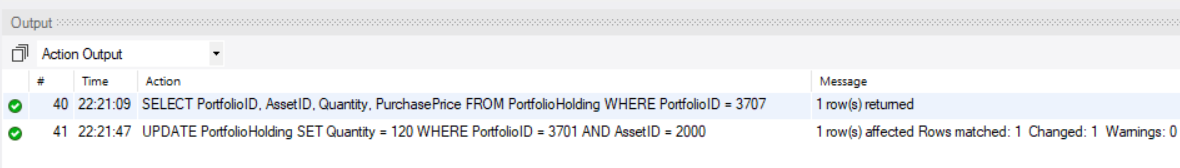
**-- Read (Select) specific data from the PortfolioHolding table based on conditions**

SELECT PortfolioID, AssetID, Quantity, PurchasePrice FROM PortfolioHolding WHERE PortfolioID = 3707;



**-- Update a specific record in the PortfolioHolding table**

UPDATE PortfolioHolding SET Quantity = 120 WHERE PortfolioID = 3701 AND AssetID = 2000;



**-- Delete a specific record from the PortfolioHolding table**

DELETE FROM PortfolioHolding WHERE PortfolioID = 3702 AND AssetID = 2001;

22:22:15 DELETE FROM PortfolioHolding WHERE PortfolioID = 3702 AND AssetID = 2001 1 row(s) affected 0.016 sec

* 1. **Transaction**

**-- Create Transaction table**

CREATE TABLE Transaction (

TransactionID INT PRIMARY KEY,

PortfolioID INT,

AssetID INT,

TransactionType ENUM('Buy', 'Sell') NOT NULL,

Quantity DECIMAL(15,2) NOT NULL,

Price DECIMAL(15,2) NOT NULL,

Date DATE NOT NULL,

FOREIGN KEY (PortfolioID) REFERENCES Portfolio(PortfolioID),

FOREIGN KEY (AssetID) REFERENCES Asset(AssetID)

);

**-- Insert values**

INSERT INTO Transaction (TransactionID, PortfolioID, AssetID, TransactionType, Quantity, Price, Date)

VALUES

(12345, 3701, 2000, 'Buy', 100, 2450.00, '2023-05-15'),

(12346, 3702, 2001, 'Sell', 50, 1550.00, '2022-08-20'),

(12347, 3703, 2002, 'Buy', 75, 1380.00, '2021-11-10'),

(12348, 3704, 2003, 'Sell', 120, 3200.00, '2020-02-25'),

(12349, 3705, 2004, 'Buy', 200, 101.50, '2023-03-08'),

(12350, 3706, 2005, 'Sell', 150, 97.25, '2022-06-12'),

(12351, 3707, 2006, 'Buy', 80, 54.00, '2021-09-20'),

(12352, 3708, 2007, 'Sell', 100, 47.50, '2020-12-30'),

(12353, 3709, 2008, 'Buy', 60, 61.00, '2023-04-18'),

(12354, 3710, 2009, 'Sell', 150, 178.00, '2022-07-25'),

(12355, 3711, 2010, 'Buy', 80, 370.00, '2021-10-15'),

(12356, 3712, 2011, 'Buy', 2, 14800000.00, '2023-01-20'),

(12357, 3713, 2012, 'Sell', 3, 7800000.00, '2022-04-28'),

(12358, 3714, 2013, 'Buy', 100, 49000.00, '2021-08-12'),

(12359, 3715, 2014, 'Sell', 50, 78.00, '2020-11-23'),

(12360, 3716, 2015, 'Buy', 1000000, 6.50, '2023-02-01'),

(12361, 3717, 2016, 'Sell', 500000, 1000000.00, '2022-05-18'),

(12362, 3718, 2017, 'Buy', 200, 9.80, '2021-09-02'),

(12363, 3719, 2018, 'Sell', 100, 500000.00, '2020-12-15'),

(12364, 3720, 2019, 'Buy', 50, 500000.00, '2023-03-22'),

(12365, 3721, 2000, 'Sell', 80, 2400.00, '2022-07-10'),

(12366, 3722, 2001, 'Buy', 120, 1520.00, '2021-10-25'),

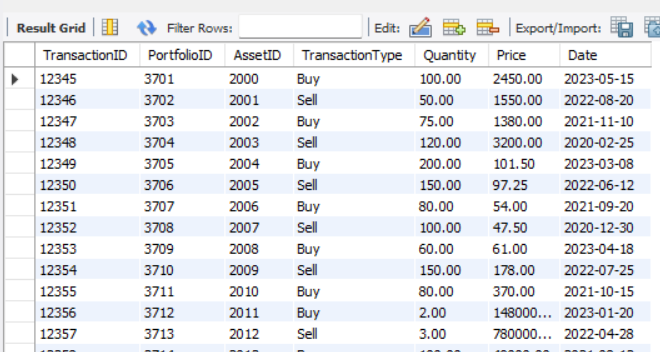
(12367, 3723, 2002, 'Sell', 50, 1400.00, '2020-01-30'),

(12368, 3724, 2003, 'Buy', 100, 3150.00, '2023-05-05'),

(12369, 3725, 2004, 'Sell', 100, 100.00, '2023-06-01');

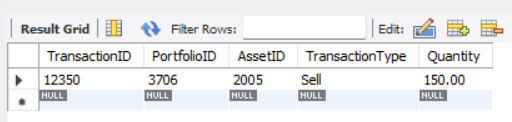
**-- Read (Select) all data from the Transaction table**

SELECT \* FROM Transaction;



**-- Read (Select) specific data from the Transaction table based on conditions**

SELECT TransactionID, PortfolioID, AssetID, TransactionType, Quantity FROM Transaction WHERE PortfolioID = 3700;



**-- Update a specific record in the Transaction table**

UPDATE Transaction SET Price = 2500.00 WHERE TransactionID = 12345;

22:25:14 UPDATE Transaction SET Price = 2500.00 WHERE TransactionID = 12345 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.015 sec

**-- Delete a specific record from the Transaction table**

DELETE FROM Transaction WHERE TransactionID = 12346;

22:25:44 DELETE FROM Transaction WHERE TransactionID = 12346 1 row(s) affected 0.016 sec

* 1. **Benchmark**

**-- Create Benchmark table**

CREATE TABLE Benchmark (

BenchmarkID INT PRIMARY KEY,

Name VARCHAR(255) NOT NULL,

Description TEXT

);

**-- Insert values**

INSERT INTO Benchmark (BenchmarkID, Name, Description)

VALUES

(1, 'NIFTY 50', 'A benchmark representing the top 50 companies listed on the National Stock Exchange of India'),

(2, 'Sensex', 'A benchmark representing the 30 largest companies listed on the Bombay Stock Exchange'),

(3, 'MSCI Emerging Markets Index', 'A benchmark tracking the performance of emerging market stocks globally'),

(4, 'S&P 500', 'A benchmark representing the 500 largest companies listed on the New York Stock Exchange'),

(5, 'Nasdaq Composite', 'A benchmark representing all common stocks listed on the Nasdaq stock market'),

(6, 'FTSE 100', 'A benchmark representing the 100 largest companies listed on the London Stock Exchange'),

(7, 'Hang Seng Index', 'A benchmark representing the top 50 companies listed on the Hong Kong Stock Exchange'),

(8, 'Nikkei 225', 'A benchmark representing the 225 largest companies listed on the Tokyo Stock Exchange'),

(9, 'CAC 40', 'A benchmark representing the 40 largest companies listed on the Paris Stock Exchange'),

(10, 'DAX 30', 'A benchmark representing the 30 largest companies listed on the Frankfurt Stock Exchange'),

(11, 'Gold Price', 'The price of gold per ounce'),

(12, 'Crude Oil Price', 'The price of crude oil per barrel'),

(13, 'US 10-Year Treasury Yield', 'The interest rate on 10-year US Treasury bonds'),

(14, 'MSCI World Index', 'A benchmark representing the performance of developed market stocks globally'),

(15, 'Russell 2000', 'A benchmark representing the smallest 2000 companies listed on the US stock market'),

(16, 'Bloomberg Barclays US Aggregate Bond Index', 'A benchmark representing the performance of US investment-grade bonds'),

(17, 'Dow Jones Industrial Average', 'A benchmark representing 30 large, well-known US companies'),

(18, 'MSCI Emerging Markets Bond Index', 'A benchmark representing the performance of emerging market bonds'),

(19, 'FTSE All-World Index', 'A benchmark representing the performance of global stocks'),

(20, 'S&P GSCI Commodities Index', 'A benchmark tracking the performance of a basket of commodities'),

(21, 'MSCI Asia Pacific Index', 'A benchmark representing the performance of Asian Pacific stocks'),

(22, 'JPMorgan Emerging Markets Bond Index', 'A benchmark representing the performance of emerging market bonds'),

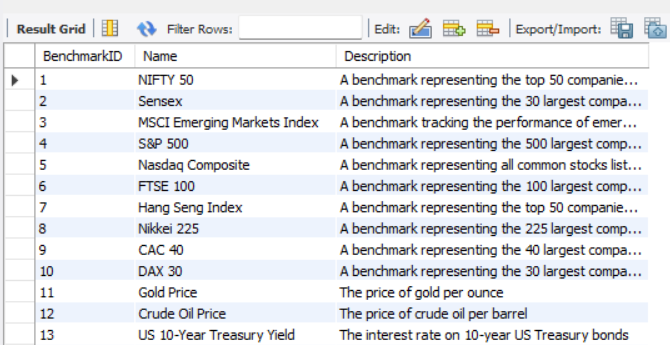
(23, 'Russell 3000', 'A benchmark representing the 3000 largest US companies'),

(24, 'MSCI Latin America Index', 'A benchmark representing the performance of Latin American stocks'),

(25, 'Bloomberg Barclays US Corporate Bond Index', 'A benchmark representing the performance of US corporate bonds');

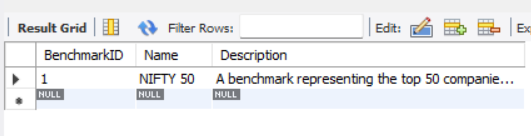
**-- Read (Select) all data from the Benchmark table**

SELECT \* FROM Benchmark;



**-- Read (Select) specific data from the Benchmark table based on conditions**

SELECT BenchmarkID, Name, Description FROM Benchmark WHERE BenchmarkID = 1;



**-- Update a specific record in the Benchmark table**

UPDATE Benchmark SET Description = 'A modified description' WHERE BenchmarkID = 2;

22:29:08 UPDATE Benchmark SET Description = 'A modified description' WHERE BenchmarkID = 2 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.000 sec

**-- Delete a specific record from the Benchmark table**

DELETE FROM Benchmark WHERE BenchmarkID = 3;

22:29:49 DELETE FROM Benchmark WHERE BenchmarkID = 3 1 row(s) affected 0.016 sec

* 1. **PortfolioBenchmark**

**-- Create PortfolioBenchmark table**

CREATE TABLE PortfolioBenchmark (

PortfolioID INT,

BenchmarkID INT,

PRIMARY KEY (PortfolioID, BenchmarkID),

FOREIGN KEY (PortfolioID) REFERENCES Portfolio(PortfolioID) ON DELETE CASCADE,

FOREIGN KEY (BenchmarkID) REFERENCES Benchmark(BenchmarkID) ON DELETE CASCADE

);

**-- Insert Values**

INSERT INTO PortfolioBenchmark (PortfolioID, BenchmarkID)

VALUES

(3701, 1),

(3702, 2),

(3703, 3),

(3704, 4),

(3705, 5),

(3706, 6),

(3707, 7),

(3708, 8),

(3709, 9),

(3710, 10),

(3711, 11),

(3712, 12),

(3713, 13),

(3714, 14),

(3715, 15),

(3716, 16),

(3717, 17),

(3718, 18),

(3719, 19),

(3720, 20),

(3721, 21),

(3722, 22),

(3723, 23),

(3724, 24),

(3725, 25);

**-- Read (Select) all data from the PortfolioBenchmark table**

SELECT \* FROM PortfolioBenchmark;

**-- Read (Select) specific data from the PortfolioBenchmark table based on conditions**

SELECT PortfolioID, BenchmarkID FROM PortfolioBenchmark WHERE PortfolioID = 3700;

* + - Update table

UPDATE PortfolioBenchmark SET BenchmarkID = 2 WHERE PortfolioID = 3701 AND BenchmarkID = 1;

22:32:07 UPDATE PortfolioBenchmark SET BenchmarkID = 2 WHERE PortfolioID = 3701 AND BenchmarkID = 1 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.000 sec

**-- Delete a specific record from the PortfolioBenchmark table**

DELETE FROM PortfolioBenchmark WHERE PortfolioID = 3701 AND BenchmarkID = 1;

22:32:45 DELETE FROM PortfolioBenchmark WHERE PortfolioID = 3701 AND BenchmarkID = 1 0 row(s) affected 0.000 sec