

# BDDA -2 Project Report

## Topic - Investment Management

### Entities:

#### 1. Customer

- **CustomerID** (Primary Key)
- Name
- Email
- ContactNumber
- Address

#### 2. Portfolio

- **PortfolioID** (Primary Key)
- CustomerID (Foreign Key)
- PortfolioName
- CreationDate

#### 3. Investment

- **InvestmentID** (Primary Key)
- PortfolioID (Foreign Key)
- InvestmentType (e.g., Stock, Bond, Mutual Fund)
- AmountInvested
- DateOfInvestment

#### 4. Transaction

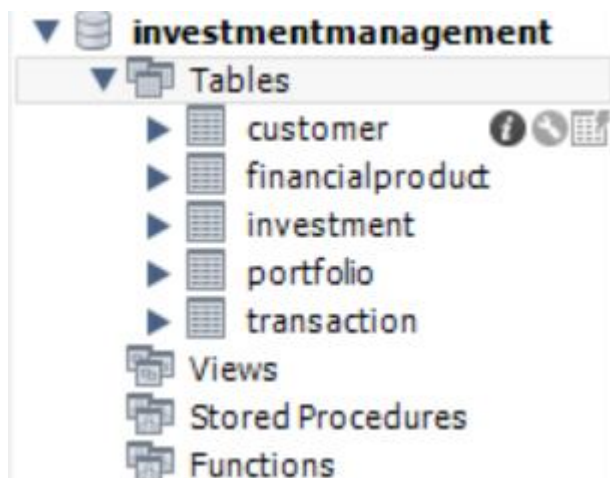
- **TransactionID** (Primary Key)
- InvestmentID (Foreign Key)
- TransactionType (e.g., Buy, Sell)
- TransactionDate
- Quantity
- PricePerUnit

#### 5. FinancialProduct

- **ProductID** (Primary Key)
- ProductName
- ProductType (e.g., Equity, Debt, Hybrid)
- RiskLevel

#### Relationships:

- A **Customer** can have multiple **Portfolios** (1 to many).
- Each **Portfolio** belongs to one **Customer** (many to 1).
- A **Portfolio** can contain multiple **Investments** (1 to many).
- Each **Investment** is part of one **Portfolio** (many to 1).
- An **Investment** can have multiple **Transactions** (1 to many).
- Each **Transaction** is associated with one **Investment** (many to 1).
- An **Investment** is related to one or more **FinancialProducts** (many to many).



#### SQL Script to make Database

```
-- Create the database
CREATE DATABASE InvestmentManagement;

-- Use the database
USE InvestmentManagement;

-- Create the Customer table
CREATE TABLE Customer (
    CustomerID INT AUTO_INCREMENT PRIMARY KEY,
```

```

    Name VARCHAR(100) NOT NULL,

    Email VARCHAR(100) UNIQUE NOT NULL,

    ContactNumber VARCHAR(15),

    Address VARCHAR(255)

);

-- Create the Portfolio table
CREATE TABLE Portfolio (

    PortfolioID INT AUTO_INCREMENT PRIMARY KEY,

    CustomerID INT,

    PortfolioName VARCHAR(100) NOT NULL,

    CreationDate DATE,

    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

-- Create the Investment table
CREATE TABLE Investment (

    InvestmentID INT AUTO_INCREMENT PRIMARY KEY,

    PortfolioID INT,

    InvestmentType VARCHAR(50),

    AmountInvested DECIMAL(10, 2),

    DateOfInvestment DATE,

    FOREIGN KEY (PortfolioID) REFERENCES Portfolio(PortfolioID)

);

-- Create the Transaction table
CREATE TABLE Transaction (

    TransactionID INT AUTO_INCREMENT PRIMARY KEY,

    InvestmentID INT,

    TransactionType VARCHAR(10),

    TransactionDate DATE,

    Quantity INT,

    PricePerUnit DECIMAL(10, 2),

    FOREIGN KEY (InvestmentID) REFERENCES Investment(InvestmentID)

);

-- Create the FinancialProduct table
CREATE TABLE FinancialProduct (

    ProductID INT AUTO_INCREMENT PRIMARY KEY,

```

```

        ProductName VARCHAR(100) NOT NULL,

        ProductType VARCHAR(50),

        RiskLevel VARCHAR(20)

    );

-- Insert dummy data into Customer table
INSERT INTO Customer (Name, Email, ContactNumber, Address) VALUES

('Alice Johnson', 'alice.johnson@example.com', '123-456-7890', '123 Elm St, Springfield'),

('Bob Smith', 'bob.smith@example.com', '234-567-8901', '456 Oak St, Springfield'),

('Charlie Brown', 'charlie.brown@example.com', '345-678-9012', '789 Maple St, Springfield');

-- Insert dummy data into Portfolio table
INSERT INTO Portfolio (CustomerID, PortfolioName, CreationDate) VALUES

(1, 'Alice\'s Retirement Fund', '2023-01-15'),

(2, 'Bob\'s Growth Portfolio', '2023-02-20'),

(3, 'Charlie\'s College Savings', '2023-03-10');

-- Insert dummy data into Investment table
INSERT INTO Investment (PortfolioID, InvestmentType, AmountInvested, DateOfInvestment) VALUES

(1, 'Stock', 5000.00, '2023-01-16'),

(1, 'Bond', 2000.00, '2023-02-01'),

(2, 'Mutual Fund', 3000.00, '2023-03-05'),

(3, 'Stock', 4000.00, '2023-03-15');

-- Insert dummy data into Transaction table
INSERT INTO Transaction (InvestmentID, TransactionType, TransactionDate, Quantity,
PricePerUnit) VALUES

(1, 'Buy', '2023-01-17', 50, 100.00),

(1, 'Sell', '2023-02-15', 20, 110.00),

(2, 'Buy', '2023-02-02', 100, 20.00),

(3, 'Buy', '2023-03-06', 150, 20.00),

(4, 'Buy', '2023-03-16', 40, 100.00);

-- Insert dummy data into FinancialProduct table
INSERT INTO FinancialProduct (ProductName, ProductType, RiskLevel) VALUES

('Tech Stock', 'Equity', 'High'),

('Government Bond', 'Debt', 'Low'),

('Balanced Mutual Fund', 'Hybrid', 'Medium');

```

Adding Dummy data

```

-- Use the database
USE InvestmentManagement;

-- Insert additional dummy data into Customer table
INSERT INTO Customer (Name, Email, ContactNumber, Address) VALUES
('David Lee', 'david.lee@example.com', '456-789-0123', '101 Pine St, Springfield'),
('Emma Davis', 'emma.davis@example.com', '567-890-1234', '102 Pine St, Springfield'),
('Frank Wright', 'frank.wright@example.com', '678-901-2345', '103 Pine St, Springfield'),
('Grace Green', 'grace.green@example.com', '789-012-3456', '104 Pine St, Springfield'),
('Hannah White', 'hannah.white@example.com', '890-123-4567', '105 Pine St, Springfield'),
('Ian Black', 'ian.black@example.com', '901-234-5678', '106 Pine St, Springfield'),
('Jack Hill', 'jack.hill@example.com', '012-345-6789', '107 Pine St, Springfield'),
('Karen King', 'karen.king@example.com', '123-456-7891', '108 Pine St, Springfield'),
('Liam Scott', 'liam.scott@example.com', '234-567-8910', '109 Pine St, Springfield'),
('Mia Young', 'mia.young@example.com', '345-678-9101', '110 Pine St, Springfield');

-- Insert additional dummy data into Portfolio table
INSERT INTO Portfolio (CustomerID, PortfolioName, CreationDate) VALUES
(4, 'David\'s Savings', '2023-04-01'),
(5, 'Emma\'s Investment Fund', '2023-04-15'),
(6, 'Frank\'s Retirement Fund', '2023-04-20'),
(7, 'Grace\'s Wealth Portfolio', '2023-05-01'),
(8, 'Hannah\'s Long-term Investment', '2023-05-10'),
(9, 'Ian\'s Growth Portfolio', '2023-05-15'),
(10, 'Jack\'s Balanced Portfolio', '2023-06-01'),
(11, 'Karen\'s Income Fund', '2023-06-10'),
(12, 'Liam\'s College Fund', '2023-06-15'),
(13, 'Mia\'s Retirement Fund', '2023-07-01');

-- Insert additional dummy data into FinancialProduct table
INSERT INTO FinancialProduct (ProductName, ProductType, RiskLevel) VALUES
('Healthcare Stock', 'Equity', 'High'),
('Corporate Bond', 'Debt', 'Medium'),
('Aggressive Growth Fund', 'Equity', 'High'),
('Municipal Bond', 'Debt', 'Low'),
('International Stock', 'Equity', 'High'),
('Real Estate Fund', 'Hybrid', 'Medium'),

```

```

('Treasury Bond', 'Debt', 'Low'),
('Index Fund', 'Equity', 'Medium'),
('Dividend Fund', 'Equity', 'Low'),
('Small Cap Fund', 'Equity', 'High');

-- Insert additional dummy data into Investment table

INSERT INTO Investment (PortfolioID, InvestmentType, AmountInvested, DateOfInvestment) VALUES

(4, 'Stock', 6000.00, '2023-04-02'),
(5, 'Bond', 2500.00, '2023-04-16'),
(6, 'Mutual Fund', 3500.00, '2023-04-21'),
(7, 'Stock', 4500.00, '2023-05-02'),
(8, 'Bond', 3000.00, '2023-05-11'),
(9, 'Mutual Fund', 5000.00, '2023-05-16'),
(10, 'Stock', 5500.00, '2023-06-02'),
(11, 'Bond', 2000.00, '2023-06-11'),
(12, 'Mutual Fund', 4000.00, '2023-06-16'),
(13, 'Stock', 7000.00, '2023-07-02');

-- Insert additional dummy data into Transaction table

INSERT INTO Transaction (InvestmentID, TransactionType, TransactionDate, Quantity,
PricePerUnit) VALUES

(5, 'Buy', '2023-04-03', 60, 100.00),
(6, 'Sell', '2023-04-17', 25, 110.00),
(7, 'Buy', '2023-04-22', 120, 20.00),
(8, 'Buy', '2023-05-03', 200, 20.00),
(9, 'Buy', '2023-05-12', 50, 100.00),
(10, 'Sell', '2023-05-17', 40, 110.00),
(11, 'Buy', '2023-06-03', 70, 20.00),
(12, 'Buy', '2023-06-12', 150, 20.00),
(13, 'Buy', '2023-06-17', 80, 100.00),
(14, 'Buy', '2023-07-03', 100, 70.00);

```

Customer Table

	CustomerID	Name	Email	ContactNumber	Address
▶	1	Alice Johnson	alice.johnson@example.com	123-456-7890	123 Elm St, Springfield
	2	Bob Smith	bob.smith@example.com	234-567-8901	456 Oak St, Springfield
	3	Charlie Brown	charlie.brown@example.com	345-678-9012	789 Maple St, Springfield
	4	David Lee	david.lee@example.com	456-789-0123	101 Pine St, Springfield
	5	Emma Davis	emma.davis@example.com	567-890-1234	102 Pine St, Springfield
	6	Frank Wright	frank.wright@example.com	678-901-2345	103 Pine St, Springfield
	7	Grace Green	grace.green@example.com	789-012-3456	104 Pine St, Springfield
	8	Hannah White	hannah.white@example.com	890-123-4567	105 Pine St, Springfield
	9	Ian Black	ian.black@example.com	901-234-5678	106 Pine St, Springfield
	10	Jack Hill	jack.hill@example.com	012-345-6789	107 Pine St, Springfield
	11	Karen King	karen.king@example.com	123-456-7891	108 Pine St, Springfield
	12	Liam Scott	liam.scott@example.com	234-567-8910	109 Pine St, Springfield
	13	Mia Young	mia.young@example.com	345-678-9101	110 Pine St, Springfield
•	NULL	NULL	NULL	NULL	NULL

Financial product Table

	ProductID	ProductName	ProductType	RiskLevel
▶	1	Tech Stock	Equity	High
	2	Government Bond	Debt	Low
	3	Balanced Mutual Fund	Hybrid	Medium
	4	Healthcare Stock	Equity	High
	5	Corporate Bond	Debt	Medium
	6	Aggressive Growth Fund	Equity	High
	7	Municipal Bond	Debt	Low
	8	International Stock	Equity	High
	9	Real Estate Fund	Hybrid	Medium
	10	Treasury Bond	Debt	Low
	11	Index Fund	Equity	Medium
	12	Dividend Fund	Equity	Low
	13	Small Cap Fund	Equity	High
•	NULL	NULL	NULL	NULL

Investment Table

	InvestmentID	PortfolioID	InvestmentType	AmountInvested	DateOfInvestment
▶	1	1	Stock	5000.00	2023-01-16
	2	1	Bond	2000.00	2023-02-01
	3	2	Mutual Fund	3000.00	2023-03-05
	4	3	Stock	4000.00	2023-03-15
	5	4	Stock	6000.00	2023-04-02
	6	5	Bond	2500.00	2023-04-16
	7	6	Mutual Fund	3500.00	2023-04-21
	8	7	Stock	4500.00	2023-05-02
	9	8	Bond	3000.00	2023-05-11
	10	9	Mutual Fund	5000.00	2023-05-16
	11	10	Stock	5500.00	2023-06-02
	12	11	Bond	2000.00	2023-06-11
	13	12	Mutual Fund	4000.00	2023-06-16
	14	13	Stock	7000.00	2023-07-02
•	NULL	NULL	NULL	NULL	NULL

Portfolio Table

	PortfolioID	CustomerID	PortfolioName	CreationDate
▶	1	1	Alice's Retirement Fund	2023-01-15
	2	2	Bob's Growth Portfolio	2023-02-20
	3	3	Charlie's College Savings	2023-03-10
	4	4	David's Savings	2023-04-01
	5	5	Emma's Investment Fund	2023-04-15
	6	6	Frank's Retirement Fund	2023-04-20
	7	7	Grace's Wealth Portfolio	2023-05-01
	8	8	Hannah's Long-term Investment	2023-05-10
	9	9	Ian's Growth Portfolio	2023-05-15
	10	10	Jack's Balanced Portfolio	2023-06-01
	11	11	Karen's Income Fund	2023-06-10
	12	12	Liam's College Fund	2023-06-15
	13	13	Mia's Retirement Fund	2023-07-01
*	NULL	NULL	NULL	NULL

Transaction Table

	TransactionID	InvestmentID	TransactionType	TransactionDate	Quantity	PricePerUnit
▶	1	1	Buy	2023-01-17	50	100.00
	2	1	Sell	2023-02-15	20	110.00
	3	2	Buy	2023-02-02	100	20.00
	4	3	Buy	2023-03-06	150	20.00
	5	4	Buy	2023-03-16	40	100.00
	6	5	Buy	2023-04-03	60	100.00
	7	6	Sell	2023-04-17	25	110.00
	8	7	Buy	2023-04-22	120	20.00
	9	8	Buy	2023-05-03	200	20.00
	10	9	Buy	2023-05-12	50	100.00
	11	10	Sell	2023-05-17	40	110.00
	12	11	Buy	2023-06-03	70	20.00
	13	12	Buy	2023-06-12	150	20.00
	14	13	Buy	2023-06-17	80	100.00
	15	14	Buy	2023-07-03	100	70.00
*	NULL	NULL	NULL	NULL	NULL	NULL

All the tables in the provided design are in Third Normal Form (3NF):

- **Customer Table:** Each customer is uniquely identified by CustomerID, and all attributes depend only on CustomerID.
- **Portfolio Table:** Each portfolio is uniquely identified by PortfolioID, and all attributes depend only on PortfolioID.
- **Investment Table:** Each investment is uniquely identified by InvestmentID, and all attributes depend only on InvestmentID.



- **Transaction Table:** Each transaction is uniquely identified by TransactionID, and all attributes depend only on TransactionID.
- **FinancialProduct Table:** Each financial product is uniquely identified by ProductID, and all attributes depend only on ProductID.

## Performing CRUD Operations

### Create Operation

```
-- Insert new customer
INSERT INTO Customer (Name, Email, ContactNumber, Address)
VALUES ('Test Customer', 'test.customer@example.com', '123-456-7890', '456 Test St, Springfield');

-- Insert new portfolio
INSERT INTO Portfolio (CustomerID, PortfolioName, CreationDate)
VALUES (1, 'Test Portfolio', CURDATE());

-- Insert new investment
INSERT INTO Investment (PortfolioID, InvestmentType, AmountInvested, DateOfInvestment)
VALUES (1, 'Stock', 1000.00, CURDATE());

-- Insert new transaction
INSERT INTO Transaction (InvestmentID, TransactionType, TransactionDate, Quantity, PricePerUnit)
VALUES (1, 'Buy', CURDATE(), 10, 100.00);

-- Insert new financial product
INSERT INTO FinancialProduct (ProductName, ProductType, RiskLevel)
VALUES ('Test Product', 'Equity', 'High');
```

### Read Operation

```
-- Read customers
SELECT * FROM Customer LIMIT 10;

-- Read portfolios
SELECT * FROM Portfolio LIMIT 10;

-- Read investments
SELECT * FROM Investment LIMIT 10;

-- Read transactions
SELECT * FROM Transaction LIMIT 10;

-- Read financial products
SELECT * FROM FinancialProduct LIMIT 10;
```

### Update Operation

```
-- Update customer
UPDATE Customer
SET Name = 'Updated Customer'
WHERE CustomerID = 1;

-- Update portfolio
UPDATE Portfolio
SET PortfolioName = 'Updated Portfolio'
```

```

WHERE PortfolioID = 1;

-- Update investment
UPDATE Investment
SET AmountInvested = 2000.00
WHERE InvestmentID = 1;

-- Update transaction
UPDATE Transaction
SET Quantity = 20
WHERE TransactionID = 1;

-- Update financial product
UPDATE FinancialProduct
SET RiskLevel = 'Low'
WHERE ProductID = 1;

```

## Delete Operation

```

-- Delete customer
DELETE FROM Customer
WHERE CustomerID = 1;

-- Delete portfolio
DELETE FROM Portfolio
WHERE PortfolioID = 1;

-- Delete investment
DELETE FROM Investment
WHERE InvestmentID = 1;

-- Delete transaction
DELETE FROM Transaction
WHERE TransactionID = 1;

-- Delete financial product
DELETE FROM FinancialProduct
WHERE ProductID = 1;

```

## Automation of CRUD Operations for stress test

```

-- Insert 1000 new customers
DELIMITER $$

CREATE PROCEDURE BulkInsertCustomers()
BEGIN
    DECLARE i INT DEFAULT 1;
    WHILE i <= 1000 DO
        INSERT INTO Customer (Name, Email, ContactNumber, Address)
        VALUES (CONCAT('Customer', 10000 + i), CONCAT('customer', 10000 + i, '@example.com'), CONCAT('123-456-', LPAD(10000 + i, 4, '0')),
        CONCAT('Address ', 10000 + i, ' St, Springfield'));
        SET i = i + 1;
    END WHILE;
END$$

DELIMITER ;

CALL BulkInsertCustomers();

```

```
-- Update 1000 customers
DELIMITER $$
```

```
CREATE PROCEDURE BulkUpdateCustomers()
BEGIN
    DECLARE i INT DEFAULT 1;
    WHILE i <= 1000 DO
        UPDATE Customer
        SET Name = CONCAT('Updated Customer', i)
        WHERE CustomerID = 10000 + i;
        SET i = i + 1;
    END WHILE;
END$$
```

```
DELIMITER ;
```

```
CALL BulkUpdateCustomers();
```

```
-- Delete 1000 customers
DELIMITER $$
```

```
CREATE PROCEDURE BulkDeleteCustomers()
BEGIN
    DECLARE i INT DEFAULT 1;
    WHILE i <= 1000 DO
        DELETE FROM Customer
        WHERE CustomerID = 10000 + i;
        SET i = i + 1;
    END WHILE;
END$$
```

```
DELIMITER ;
```

```
CALL BulkDeleteCustomers();
```

26	13:08:02	CALL BulkInsertCustomers()	1 row(s) affected	6.235 sec
27	13:08:09	CREATE PROCEDURE BulkUpdateCustomers() BEGIN DECLARE i INT DEFAULT 1; WHILE i <= 1000 DO UPDATE...	0 row(s) affected	0.015 sec
28	13:08:09	CALL BulkUpdateCustomers()	0 row(s) affected	0.016 sec
29	13:08:09	CREATE PROCEDURE BulkDeleteCustomers() BEGIN DECLARE i INT DEFAULT 1; WHILE i <= 1000 DO DELETE ...	0 row(s) affected	0.016 sec
30	13:08:09	CALL BulkDeleteCustomers()	0 row(s) affected	0.015 sec
31	13:08:19	SELECT * FROM investmentmanagement.customer	1014 row(s) returned	0.000 sec / 0.000 sec

1014 Bulk Customers added to the database using stress testing

The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the 'investmentmanagement' database schema with tables like 'customer', 'financialproduct', 'investment', 'portfolio', and 'transaction'. The central pane shows a query editor with the statement: `SELECT * FROM investmentmanagement.customer;`. The bottom pane shows the execution output of the bulk insert and update procedures, including the final SELECT statement that returned 1014 rows.

#	Time	Action	Message	Duration / Fetch
26	13:08:02	CALL BulkInsertCustomers()	1 row(s) affected	6.235 sec
27	13:08:09	CREATE PROCEDURE BulkUpdateCustomers() BEGIN DECLARE i INT DEFAULT 1; WHILE i <= 1000 DO UPDATE...	0 row(s) affected	0.015 sec
28	13:08:09	CALL BulkUpdateCustomers()	0 row(s) affected	0.016 sec
29	13:08:09	CREATE PROCEDURE BulkDeleteCustomers() BEGIN DECLARE i INT DEFAULT 1; WHILE i <= 1000 DO DELETE ...	0 row(s) affected	0.016 sec
30	13:08:09	CALL BulkDeleteCustomers()	0 row(s) affected	0.015 sec
31	13:08:19	SELECT * FROM investmentmanagement.customer	1014 row(s) returned	0.000 sec / 0.000 sec

