**Exercise 5: Triggers**

**Scenario 1:** Automatically update the last modified date when a customer's record is updated.

**Question:** Write a trigger **UpdateCustomerLastModified** that updates the LastModified column of the Customers table to the current date whenever a customer's record is updated.

**PL/SQL Procedure:**

DELIMITER //

CREATE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON Customers

FOR EACH ROW

BEGIN

SET NEW.LastModified = NOW();

END;

//

DELIMITER ;

**Scenario 2:** Maintain an audit log for all transactions.

**Question:** Write a trigger **LogTransaction** that inserts a record into an AuditLog table whenever a transaction is inserted into the Transactions table

**PL/SQL Procedure:**

First, create the AuditLog table to store audit records.

CREATE TABLE AuditLog (

AuditID INT AUTO\_INCREMENT PRIMARY KEY,

TransactionID INT,

AccountID INT,

TransactionDate DATETIME,

Amount DECIMAL(10, 2),

TransactionType VARCHAR(10),

LogDate DATETIME DEFAULT CURRENT\_TIMESTAMP

);

DELIMITER //

CREATE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

INSERT INTO AuditLog (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

VALUES (NEW.TransactionID, NEW.AccountID, NEW.TransactionDate, NEW.Amount, NEW.TransactionType);

END;

//

DELIMITER ;

**Scenario 3:** Enforce business rules on deposits and withdrawals.

**Question:** Write a trigger **CheckTransactionRules** that ensures withdrawals do not exceed the balance and deposits are positive before inserting a record into the Transactions table

**PL/SQL Procedure:**

DELIMITER //

CREATE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

BEGIN

DECLARE v\_Balance DECIMAL(10, 2);

SELECT Balance INTO v\_Balance FROM Accounts WHERE AccountID = NEW.AccountID;

IF NEW.TransactionType = 'Withdrawal' AND NEW.Amount > v\_Balance THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Insufficient balance for withdrawal';

END IF;

IF NEW.Amount <= 0 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Transaction amount must be positive';

END IF;

END;

//

DELIMITER ;

**Testing the Triggers:**

**1.Update a customer record to test UpdateCustomerLastModified:**

UPDATE Customers SET Name = 'John Doe Updated' WHERE CustomerID = 1;

SELECT \* FROM Customers WHERE CustomerID = 1;

**2.Insert a transaction to test LogTransaction and CheckTransactionRules**:

-- Successful deposit

INSERT INTO Transactions (AccountID, TransactionDate, Amount, TransactionType) VALUES (1, NOW(), 200, 'Deposit');

SELECT \* FROM Transactions;

SELECT \* FROM AuditLog;

-- Unsuccessful withdrawal due to insufficient balance

INSERT INTO Transactions (AccountID, TransactionDate, Amount, TransactionType) VALUES (1, NOW(), 2000, 'Withdrawal');

-- Unsuccessful deposit with negative amount

INSERT INTO Transactions (AccountID, TransactionDate, Amount, TransactionType) VALUES (1, NOW(), -100, 'Deposit');