**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.

**CODE**

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON Customers

FOR EACH ROW

BEGIN

    :NEW.LastModified := SYSDATE;

END UpdateCustomerLastModified;

/

**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.

**CODE**

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

    INSERT INTO AuditLog (

        TransactionID,

        AccountID,

        TransactionDate,

        Amount,

        TransactionType,

        LogTime

    )

    VALUES (

        :NEW.TransactionID,

        :NEW.AccountID,

        :NEW.TransactionDate,

        :NEW.Amount,

        :NEW.TransactionType,

        SYSDATE

    );

END;

/

**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.

**CODE**

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

DECLARE

    current\_balance NUMBER;

BEGIN

    -- Check if the transaction is a withdrawal

    IF :NEW.transaction\_type = 'WITHDRAWAL' THEN

        -- Retrieve the current balance of the account

        SELECT balance INTO current\_balance FROM Accounts WHERE account\_id = :NEW.account\_id FOR UPDATE;

        -- Ensure the withdrawal amount does not exceed the current balance

        IF :NEW.amount > current\_balance THEN

            RAISE\_APPLICATION\_ERROR(-20001, 'Withdrawal amount exceeds current balance.');

        END IF;

    ELSIF :NEW.transaction\_type = 'DEPOSIT' THEN

        -- Ensure the deposit amount is positive

        IF :NEW.amount <= 0 THEN

            RAISE\_APPLICATION\_ERROR(-20002, 'Deposit amount must be positive.');

        END IF;

    END IF;

END CheckTransactionRules;

/