**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.

SQL> CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

2 BEFORE UPDATE ON Customers

3 FOR EACH ROW

4 BEGIN

5 :NEW.LastModified := SYSDATE;

6 END;

7 /

Trigger created.

**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.

SQL> CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

2 BEFORE UPDATE ON Customers

3 FOR EACH ROW

4 BEGIN

5 :NEW.LastModified := SYSDATE;

6 END;

7 /

Trigger created.

**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.

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

BEGIN

IF :NEW.transactiontype = 'WITHDRAWAL' THEN

SELECT balance

INTO :new.balance

FROM accounts

WHERE accountid = :NEW.accountid;

IF :new.amount > :new.balance THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds');

END IF;

ELSIF :NEW.transactiontype = 'DEPOSIT' THEN

IF :NEW.amount <= 0 THEN

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

END IF;

END IF;

END;

/