**Exercise 5: Triggers**

**Scenario 1: Update Last Modified Date**

**Trigger**:

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON Customers

FOR EACH ROW

BEGIN

:NEW.LastModified := SYSDATE;

END;

/

**Verification Query**:

UPDATE Customers SET Name = 'Johnathan Doe' WHERE CustomerID = 1;

SELECT \* FROM Customers WHERE CustomerID = 1;

**Sample Output**:

CUSTOMERID | NAME | DOB | BALANCE | LASTMODIFIED

---------------------------------------------------------------

1 | Johnathan Doe | 1985-05-15 | 1000 | 2024-08-08

**Scenario 2: Maintain an Audit Log**

**Trigger**:

CREATE TABLE AuditLog (

LogID NUMBER PRIMARY KEY,

TransactionID NUMBER,

ActionType VARCHAR2(50),

ActionDate DATE

);

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

INSERT INTO AuditLog (LogID, TransactionID, ActionType, ActionDate)

VALUES (AuditLog\_SEQ.NEXTVAL, :NEW.TransactionID, 'INSERT', SYSDATE);

END; /

**SEQUENCE VERIFICATION:**CREATE SEQUENCE AuditLog\_SEQ

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

**RECOMPILE TRIGGER:**

ALTER TRIGGER LogTransaction COMPILE;

**Verification Query**:

BEGIN

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

VALUES (3, 1, SYSDATE, 300, 'Withdrawal');

END;

/

SELECT \* FROM AuditLog;

**Sample Output**:

LOGID | TRANSACTIONID | ACTIONTYPE | ACTIONDATE

------|---------------|------------|------------

1 | 3 | INSERT | <current\_date>

**Scenario 3: Check Transaction Rules**

**Trigger**:

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

BEGIN

IF :NEW.TransactionType = 'Withdrawal' THEN

DECLARE

v\_balance NUMBER;

BEGIN

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

IF v\_balance < :NEW.Amount THEN

RAISE\_APPLICATION\_ERROR(-20004, 'Insufficient funds for withdrawal');

END IF;

END;

ELSIF :NEW.TransactionType = 'Deposit' THEN

IF :NEW.Amount <= 0 THEN

RAISE\_APPLICATION\_ERROR(-20005, 'Deposit amount must be positive');

END IF;

END IF;

END;

/

**Verification Query**:

BEGIN

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

VALUES (3, 1, SYSDATE, 300, 'Withdrawal');

END;

/

**Sample Output**:

If there are insufficient funds or invalid deposit amount:

Error: Insufficient funds for withdrawal / Deposit amount must be positive