**EXERCISE 5**-**TRIGGERS**

**SCENARIO 1:** **UpdateCustomerLastModified**

**CREATING PROCEDURE:**

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE

ON Customers

FOR EACH ROW

BEGIN

    :NEW.LastModified := SYSDATE;

END;

/

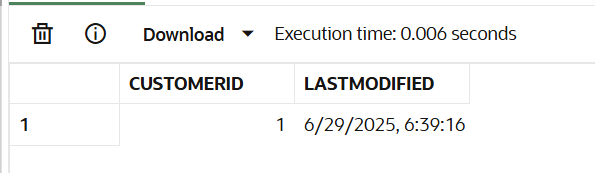
UPDATE Customers

SET Balance = Balance + 1000

WHERE CustomerID = 1;

SELECT CustomerID, LastModified FROM Customers WHERE CustomerID=1;

**OUTPUT:**

****

**SCENARIO 2: LogTransaction**

**TABLE CREATION:**

CREATE TABLE Transactions (

TransactionID NUMBER PRIMARY KEY,

AccountID NUMBER,

TransactionDate DATE,

Amount NUMBER,

TransactionType VARCHAR2(10),

FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)

);

CREATE TABLE AuditLog (

AuditID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

TransactionID NUMBER,

ActionDate DATE

);

**PROCEDURE:**

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT

ON Transactions

FOR EACH ROW

BEGIN

    INSERT INTO AuditLog (TransactionID, ActionDate)

    VALUES (:NEW.TransactionID, SYSDATE);

END;

/

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT

ON Transactions

FOR EACH ROW

BEGIN

    INSERT INTO AuditLog (TransactionID, ActionDate)

    VALUES (:NEW.TransactionID, SYSDATE);

END;

/

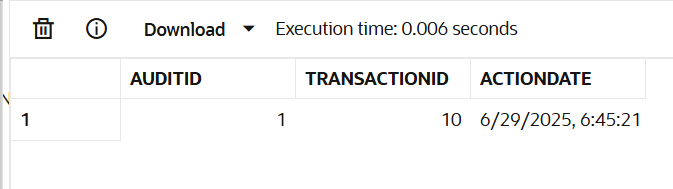
INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

VALUES (10, 101, SYSDATE, 500, 'Deposit');

COMMIT;

SELECT \* FROM AuditLog;

**OUTPUT:**

****

**SCENARIO 3:** **CheckTransactionRules**

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT

ON Transactions

FOR EACH ROW

DECLARE

    v\_balance NUMBER;

BEGIN

    -- get current account balance

    SELECT Balance INTO v\_balance

    FROM Accounts

    WHERE AccountID = :NEW.AccountID;

    IF :NEW.TransactionType = 'Withdrawal' THEN

        IF :NEW.Amount > v\_balance THEN

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

        END IF;

    ELSIF :NEW.TransactionType = 'Deposit' THEN

        IF :NEW.Amount <= 0 THEN

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

        END IF;

    END IF;

END;

/

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

VALUES (1, 101, SYSDATE, 1000, 'Withdrawal');

COMMIT;

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

VALUES (8, 101, SYSDATE, 0, 'Deposit');

**OUTPUT:**

