**Exercise 5: Triggers**

**ANS1:**

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON CUSTOMERS

FOR EACH ROW

BEGIN

:NEW.LASTMODIFIED:=SYSDATE;

END;

/

**ANS2:**

CREATE OR REPLACE TRIGGER LogTransaction

BEFORE INSERT ON TRANSACTIONS

FOR EACH ROW

BEGIN

INSERT INTO AUDITLOG VALUES(

1,:NEW.TRANSACTIONID,:NEW.ACCOUNTID,:NEW.TRANSACTIONDATE,

:NEW.AMOUNT,:NEW.TRANSACTIONTYPE,SYSDATE);

END;

/

**ANS3:**

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON TRANSACTIONS

FOR EACH ROW

DECLARE

VBAL ACCOUNTS.BALANCE%TYPE;

E1 EXCEPTION;

E2 EXCEPTION;

E3 EXCEPTION;

BEGIN

SELECT BALANCE INTO VBAL

FROM ACCOUNTS WHERE

ACCOUNTID= :NEW.ACCOUNTID

FOR UPDATE;

IF :NEW.TRANSACTIONTYPE='WITHDRAWAL' THEN

IF :NEW.AMOUNT>VBAL THEN

RAISE E1;

END IF;

ELSIF :NEW.TRANSACTIONTYPE='DEPOSIT' THEN

IF :NEW.AMOUNT<0 THEN

RAISE E2;

END IF;

ELSE

RAISE E3;

END IF;

EXCEPTION

WHEN E1 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'WITHDRAWAL AMOUNT EXCEEDS CURRENT BALANCE.');

WHEN E2 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'DEPOSIT AMOUNT MUST BE POSITIVE.');

WHEN E3 THEN

RAISE\_APPLICATION\_ERROR(-20003, 'INVALID TRANSACTION TYPE.');

END;

/