**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.

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON customers

FOR EACH ROW

BEGIN

:NEW.LastModified := SYSDATE;

END;

**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.

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT ON transactions

FOR EACH ROW

BEGIN

INSERT INTO AuditLog (transaction\_id, account\_id, amount, transaction\_date, log\_date)

VALUES (:NEW.transaction\_id, :NEW.account\_id, :NEW.amount, :NEW.transaction\_date, 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.

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON transactions

FOR EACH ROW

DECLARE

v\_balance NUMBER;

BEGIN

IF :NEW.amount > 0 THEN

NULL;

ELSIF :NEW.amount < 0 THEN

SELECT balance INTO v\_balance

FROM accounts

WHERE account\_id = :NEW.account\_id

FOR UPDATE;

IF v\_balance + :NEW.amount < 0 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance for withdrawal.');

END IF;

ELSE

RAISE\_APPLICATION\_ERROR(-20002, 'Transaction amount must be non-zero.');

END IF;

END;