**EXERCISE 5:**

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;

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,

transaction\_type,

amount,

transaction\_date,

created\_by,

created\_at

) VALUES (

:NEW.transaction\_id,

:NEW.account\_id,

:NEW.transaction\_type,

:NEW.amount,

:NEW.transaction\_date,

USER,

SYSDATE

);

END;  
  
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

-- Check if the transaction is a withdrawal

IF :NEW.transaction\_type = 'WITHDRAWAL' THEN

-- Get the current balance of the account

SELECT balance INTO v\_balance FROM accounts WHERE account\_id = :NEW.account\_id FOR UPDATE;

-- Ensure the balance is sufficient

IF v\_balance < :NEW.amount THEN

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

END IF;

ELSIF :NEW.transaction\_type = 'DEPOSIT' THEN

-- Ensure the deposit amount is positive

IF :NEW.amount <= 0 THEN

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

END IF;

END IF;

END;