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 TABLE AuditLog (  AuditID NUMBER PRIMARY KEY,  TransactionID NUMBER,  ChangeDate DATE,  ChangeType VARCHAR2(50) );   CREATE SEQUENCE AuditLogSeq START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;   CREATE OR REPLACE TRIGGER LogTransaction AFTER INSERT ON Transactions FOR EACH ROW BEGIN  INSERT INTO AuditLog (AuditID, TransactionID, ChangeDate, ChangeType)  VALUES (AuditLogSeq.NEXTVAL, :NEW.TransactionID, SYSDATE, 'INSERT'); 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.TransactionType = 'Withdrawal' THEN  SELECT Balance INTO v\_balance  FROM Accounts  WHERE AccountID = :NEW.AccountID;    IF v\_balance < :NEW.Amount THEN  RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds for withdrawal');  END IF;  END IF;    IF :NEW.TransactionType = 'Deposit' THEN  IF :NEW.Amount <= 0 THEN  RAISE\_APPLICATION\_ERROR(-20002, 'Deposit amount must be positive');  END IF;  END IF; END; |
| --- |