-- Drop tables if they already exist

DROP TABLE Accounts CASCADE CONSTRAINTS;

DROP TABLE Employees CASCADE CONSTRAINTS;

-- Create Accounts table

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

AccountType VARCHAR2(20), -- e.g., 'Savings', 'Current'

Balance NUMBER(10, 2)

);

-- Create Employees table

CREATE TABLE Employees (

EmployeeID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Department VARCHAR2(50),

Salary NUMBER(10, 2)

);

-- Insert sample data into Accounts

INSERT INTO Accounts VALUES (101, 1, 'Savings', 5000);

INSERT INTO Accounts VALUES (102, 2, 'Savings', 8000);

INSERT INTO Accounts VALUES (103, 1, 'Current', 3000);

INSERT INTO Accounts VALUES (104, 3, 'Savings', 10000);

-- Insert sample data into Employees

INSERT INTO Employees VALUES (1, 'Alice', 'HR', 50000);

INSERT INTO Employees VALUES (2, 'Bob', 'Finance', 60000);

INSERT INTO Employees VALUES (3, 'Charlie', 'HR', 55000);

COMMIT;  
  
  
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountType = 'Savings';

DBMS\_OUTPUT.PUT\_LINE('Monthly interest processed for all savings accounts.');

COMMIT;

END;

/  
  
EXEC ProcessMonthlyInterest;

SELECT \* FROM Accounts;

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_Department IN VARCHAR2,

p\_BonusPercent IN NUMBER

) IS

BEGIN

UPDATE Employees

SET Salary = Salary + (Salary \* (p\_BonusPercent / 100))

WHERE Department = p\_Department;

DBMS\_OUTPUT.PUT\_LINE('Bonus updated for department: ' || p\_Department);

COMMIT;

END;

/  
  
EXEC UpdateEmployeeBonus('HR', 10); -- 10% bonus for HR

SELECT \* FROM Employees;

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_FromAccount IN NUMBER,

p\_ToAccount IN NUMBER,

p\_Amount IN NUMBER

) IS

v\_Balance NUMBER;

BEGIN

-- Get current balance of source account

SELECT Balance INTO v\_Balance

FROM Accounts

WHERE AccountID = p\_FromAccount

FOR UPDATE;

-- Check if balance is sufficient

IF v\_Balance < p\_Amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in source account.');

END IF;

-- Deduct from source account

UPDATE Accounts

SET Balance = Balance - p\_Amount

WHERE AccountID = p\_FromAccount;

-- Add to destination account

UPDATE Accounts

SET Balance = Balance + p\_Amount

WHERE AccountID = p\_ToAccount;

DBMS\_OUTPUT.PUT\_LINE('Transfer of ' || p\_Amount || ' from Account ' || p\_FromAccount || ' to Account ' || p\_ToAccount || ' successful.');

COMMIT;

END;

/  
  
EXEC TransferFunds(101, 103, 1000);

SELECT \* FROM Accounts;