**Exercise 3: Stored Procedures**

**Scenario1:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

-- Update savings accounts by adding 1% interest to balance

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountType = 'Savings';

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Monthly interest of 1% applied to all savings accounts.');

END;

/

BEGIN

ProcessMonthlyInterest;

END;

/

**OUTPUT:**



**Scenario2:**

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_department IN VARCHAR2,

p\_bonus\_percent IN NUMBER

) AS

BEGIN

-- Update salary by adding bonus percentage for employees in given department

UPDATE Employees

SET Salary = Salary + (Salary \* p\_bonus\_percent / 100)

WHERE Department = p\_department;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Salaries updated with a bonus of ' || p\_bonus\_percent || '% for department ' || p\_department);

END;

/

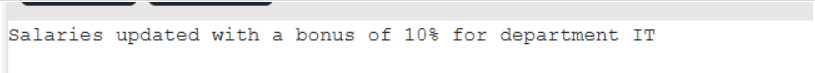
BEGIN

UpdateEmployeeBonus('IT', 10);

END;

/

**OUTPUT**:



**Scenario3:**

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_from\_account IN NUMBER,

p\_to\_account IN NUMBER,

p\_amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

-- Check balance of source account

SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = p\_from\_account;

IF v\_balance < p\_amount THEN

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

ELSE

-- Deduct from source

UPDATE Accounts SET Balance = Balance - p\_amount WHERE AccountID = p\_from\_account;

-- Add to destination

UPDATE Accounts SET Balance = Balance + p\_amount WHERE AccountID = p\_to\_account;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Transferred $' || p\_amount || ' from account ' || p\_from\_account || ' to account ' || p\_to\_account);

END IF;

END;

/

BEGIN

TransferFunds(1, 2, 100);

END;

/

**OUTPUT:**

