**Exercise 3: Stored Procedures**

**Scenario 1:**

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE Accounts';

EXCEPTION

WHEN OTHERS THEN

NULL;

END;

/

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

AccountType VARCHAR2(20),

Balance NUMBER,

LastModified DATE

);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (1, 101, 'Savings', 10000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (2, 102, 'Checking', 5000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (3, 103, 'Savings', 15000, SYSDATE);

COMMIT;

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

UPDATE Accounts

SET Balance = Balance \* 1.01,

LastModified = SYSDATE

WHERE AccountType = 'Savings';

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

END;

/

BEGIN

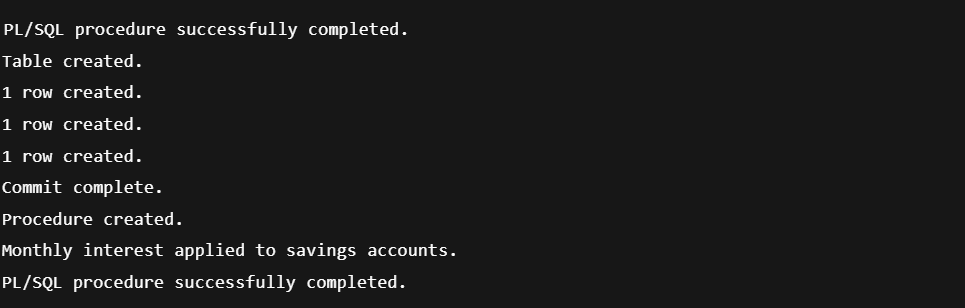
DBMS\_OUTPUT.ENABLE;

ProcessMonthlyInterest;

END;

/

**Output:**

****

**Scenario 2:**

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE Employees';

EXCEPTION

WHEN OTHERS THEN

NULL;

END;

/

CREATE TABLE Employees (

EmployeeID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Position VARCHAR2(50),

Salary NUMBER,

Department VARCHAR2(50),

HireDate DATE

);

INSERT INTO Employees VALUES (1, 'Alice', 'Manager', 70000, 'HR', SYSDATE);

INSERT INTO Employees VALUES (2, 'Bob', 'Developer', 60000, 'IT', SYSDATE);

INSERT INTO Employees VALUES (3, 'Charlie', 'Clerk', 40000, 'HR', SYSDATE);

COMMIT;

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

dept\_name IN VARCHAR2,

bonus\_pct IN NUMBER

) AS

BEGIN

UPDATE Employees

SET Salary = Salary + (Salary \* bonus\_pct / 100)

WHERE Department = dept\_name;

DBMS\_OUTPUT.PUT\_LINE('Bonus applied to ' || dept\_name || ' department.');

END;

/

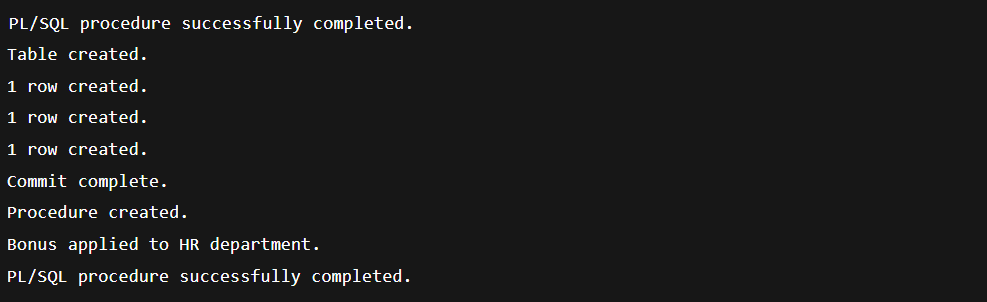
BEGIN

DBMS\_OUTPUT.ENABLE;

UpdateEmployeeBonus('HR', 10);

END;

/**Output:**



**Scenario 3:**

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE Accounts';

EXCEPTION

WHEN OTHERS THEN

NULL;

END;

/

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

AccountType VARCHAR2(20),

Balance NUMBER,

LastModified DATE

);

INSERT INTO Accounts VALUES (1, 101, 'Savings', 10000, SYSDATE);

INSERT INTO Accounts VALUES (2, 101, 'Checking', 2000, SYSDATE);

COMMIT;

CREATE OR REPLACE PROCEDURE TransferFunds (

from\_account\_id IN NUMBER,

to\_account\_id IN NUMBER,

amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance

FROM Accounts

WHERE AccountID = from\_account\_id

FOR UPDATE;

IF v\_balance < amount THEN

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

END IF;

UPDATE Accounts

SET Balance = Balance - amount,

LastModified = SYSDATE

WHERE AccountID = from\_account\_id;

UPDATE Accounts

SET Balance = Balance + amount,

LastModified = SYSDATE

WHERE AccountID = to\_account\_id;

DBMS\_OUTPUT.PUT\_LINE(

'Transferred ' || amount || ' from Account ' || from\_account\_id || ' to Account ' || to\_account\_id

);

END;

/

BEGIN

DBMS\_OUTPUT.ENABLE;

TransferFunds(1, 2, 1500);

END;

/**Output:**

