**Exercise 3: Stored Procedures**

**Scenario 1:** The bank needs to process monthly interest for all savings accounts.

* + Question: Write a stored procedure ProcessMonthlyInterest that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

**Scenario 2:** The bank wants to implement a bonus scheme for employees based on their performance.

* + Question: Write a stored procedure UpdateEmployeeBonus that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

**Scenario 3:** Customers should be able to transfer funds between their accounts.

* + Question: Write a stored procedure TransferFunds that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

**Scenario 1: Process Monthly Interest for Savings Accounts**

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, 1, 'SAVINGS', 1000, SYSDATE);

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR acc IN (

SELECT AccountID, Balance

FROM Accounts

WHERE UPPER(AccountType) = 'SAVINGS'

)

LOOP

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01),

LastModified = SYSDATE

WHERE AccountID = acc.AccountID;

END LOOP;

COMMIT;

END;

/

BEGIN

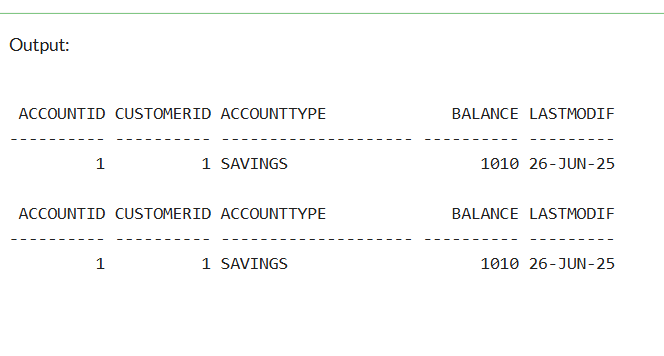
ProcessMonthlyInterest;

END;

/

-- View result

SELECT \* FROM Accounts;



**Scenario 2: Employee Bonus Based on Department and Performance**

CREATE TABLE Employees (

EmployeeID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Position VARCHAR2(50),

Salary NUMBER,

Department VARCHAR2(50),

HireDate DATE

);

INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)

VALUES (1, 'Alice Johnson', 'Manager', 70000, 'HR', TO\_DATE('2015-06-15', 'YYYY-MM-DD'));

INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)

VALUES (2, 'Bob Brown', 'Developer', 60000, 'IT', TO\_DATE('2017-03-20', 'YYYY-MM-DD'));

INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)

VALUES (3, 'Chirag Patel', 'Analyst', 58000, 'HR', TO\_DATE('2018-09-10', 'YYYY-MM-DD'));

INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)

VALUES (4, 'Sneha Reddy', 'QA Engineer', 61000, 'IT', TO\_DATE('2019-12-01', 'YYYY-MM-DD'));

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

dept\_name IN VARCHAR2,

bonus\_percent IN NUMBER

) AS

BEGIN

UPDATE Employees

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

WHERE UPPER(Department) = UPPER(dept\_name);

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

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/

BEGIN

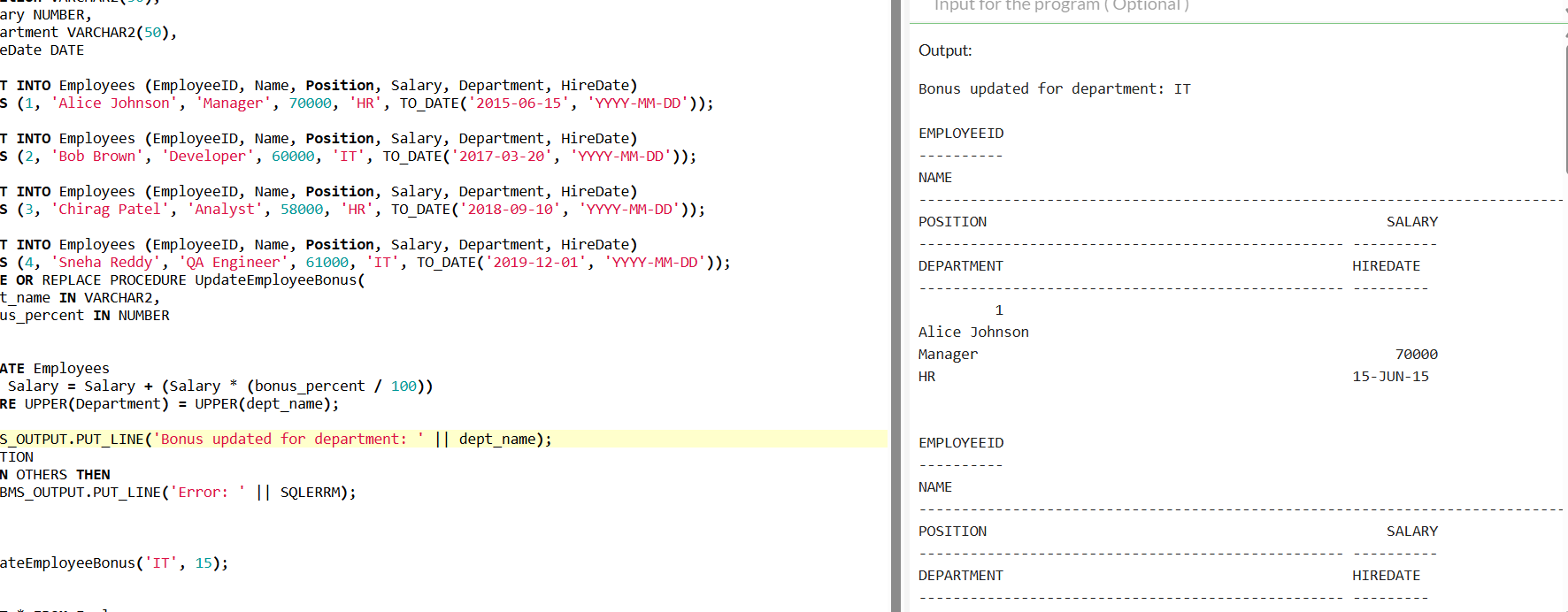
UpdateEmployeeBonus('IT', 15);

END;

/

SELECT \* FROM Employees;--

**Output:**



**Scenario 3: Transfer Funds Between Accounts**

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

AccountType VARCHAR2(20),

Balance NUMBER,

LastModified DATE

);

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

VALUES (1, 'Savings', 1000, SYSDATE);

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

VALUES (2, 'Checking', 1500, SYSDATE);

CREATE OR REPLACE PROCEDURE TransferFunds(

from\_account\_id IN NUMBER,

to\_account\_id IN NUMBER,

amount IN NUMBER

) AS

insufficient\_balance EXCEPTION;

v\_balance NUMBER;

BEGIN

-- Check balance of source account

SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = from\_account\_id;

IF v\_balance < amount THEN

RAISE insufficient\_balance;

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('Transfer successful!');

EXCEPTION

WHEN insufficient\_balance THEN

DBMS\_OUTPUT.PUT\_LINE('Transfer failed: Insufficient balance.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/

BEGIN

TransferFunds(1, 2, 500);

END;

/

