**Scenario 1:** The bank wants to apply a discount to loan interest rates for customers above 60 years old.

**o Question:** Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates.

Code

-- Initializing database

CREATE TABLE CustomerLoans (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(50),

Age NUMBER,

Balance NUMBER(10,2),

IsVIP CHAR(1) DEFAULT 'F', -- 'T' or 'F'

InterestRate NUMBER(5,2),

DueDate DATE

);

-- Inserting Sample Records

INSERT INTO CustomerLoans VALUES (1, 'Alice', 65, 15000.00, 'N', 7.5, SYSDATE + 20);

INSERT INTO CustomerLoans VALUES (2, 'Bob', 45, 8000.00, 'N', 8.0, SYSDATE + 40);

INSERT INTO CustomerLoans VALUES (3, 'Charlie', 70, 12000.00, 'N', 7.0, SYSDATE + 15);

INSERT INTO CustomerLoans VALUES (4, 'Diana', 30, 5000.00, 'N', 9.0, SYSDATE + 5);

INSERT INTO CustomerLoans VALUES (5, 'Edward', 61, 11000.00, 'N', 8.5, SYSDATE + 50);

COMMIT;

-- PL/SQL Procedure

CREATE PROCEDURE ProcessMonthlyInterest

IS

i NUMBER := 1;

name VARCHAR2(50);

bal NUMBER(10,2);

BEGIN

LOOP

SELECT Name,Balance INTO name,bal FROM CustomerLoans WHERE CustomerID=i;

UPDATE CustomerLoans SET Balance = Balance + (Balance \* 0.01) WHERE CustomerID=i;

bal := bal + (bal\*0.01);

dbms\_output.put\_line('Dear ' || name || ',');

dbms\_output.put\_line('1% Interest Rate is applied to your current balance. Your new balance is ' || bal);

i := i+1;

exit when i>5;

END LOOP;

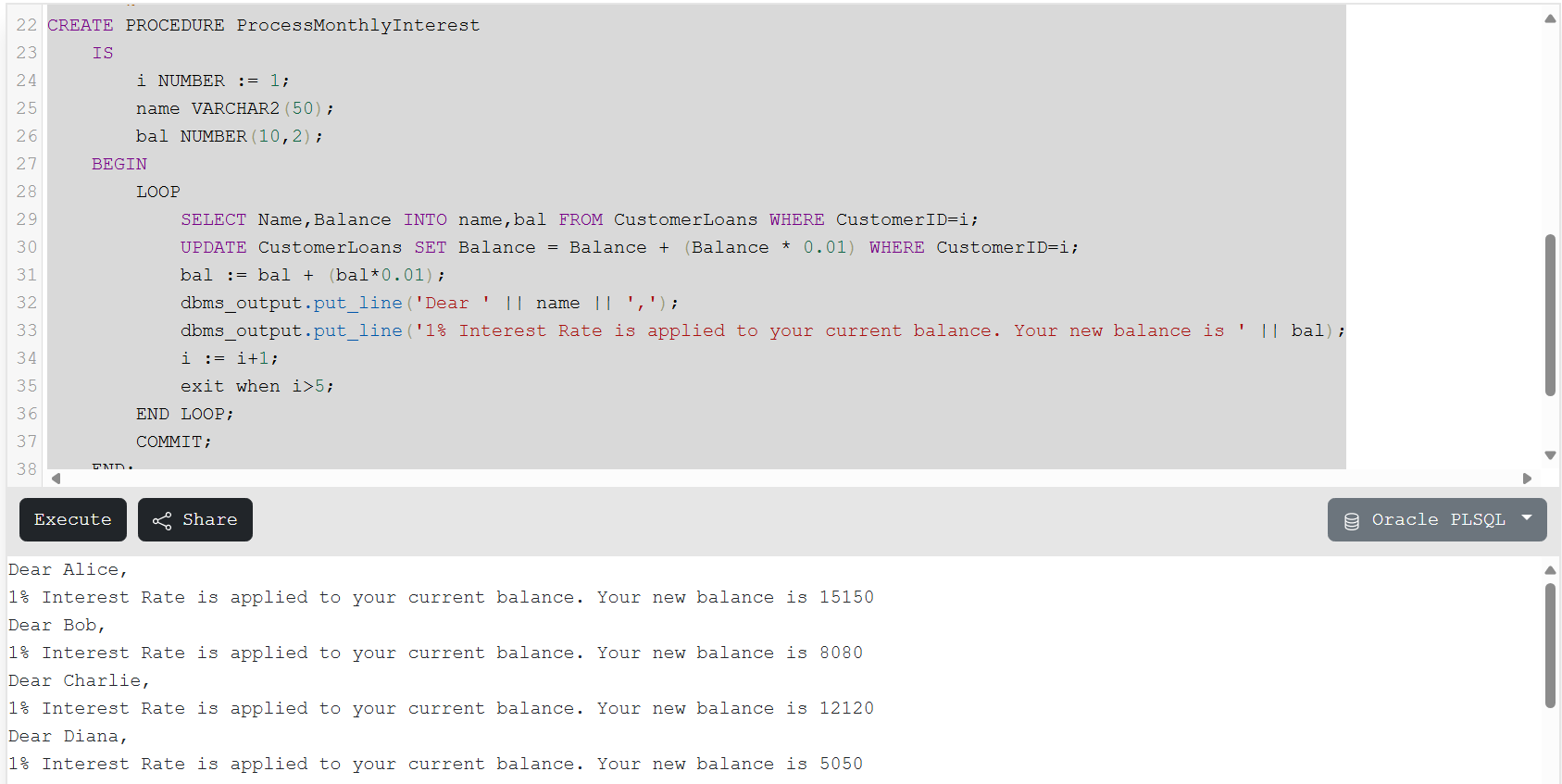
COMMIT;

END;

/

--PL/SQL main

EXECUTE ProcessMonthlyInterest;

Output  
  


**Scenario 2:** A customer can be promoted to VIP status based on their balance.

**Question:** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.  
  
  
  
  
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (dept IN VARCHAR2, bonus IN NUMBER)

IS

i NUMBER := 101;

name VARCHAR2(50);

d VARCHAR2(50);

sal NUMBER(10,2);

b NUMBER := bonus/100;

BEGIN

LOOP

SELECT Name,Department,Salary INTO name,d,sal FROM Employees WHERE EmployeeID=i;

If(dept = d)THEN

UPDATE Employees SET Salary = Salary + (Salary \* b) WHERE EmployeeID=i;

sal := sal + (sal\*b);

dbms\_output.put\_line('EMPLOYEE NAME: ' || name || ',');

dbms\_output.put\_line('We have added a bonus of ' || bonus || '% to your current salary. Your new salary is ' || sal);

END IF;

i := i+1;

exit when i>107;

END LOOP;

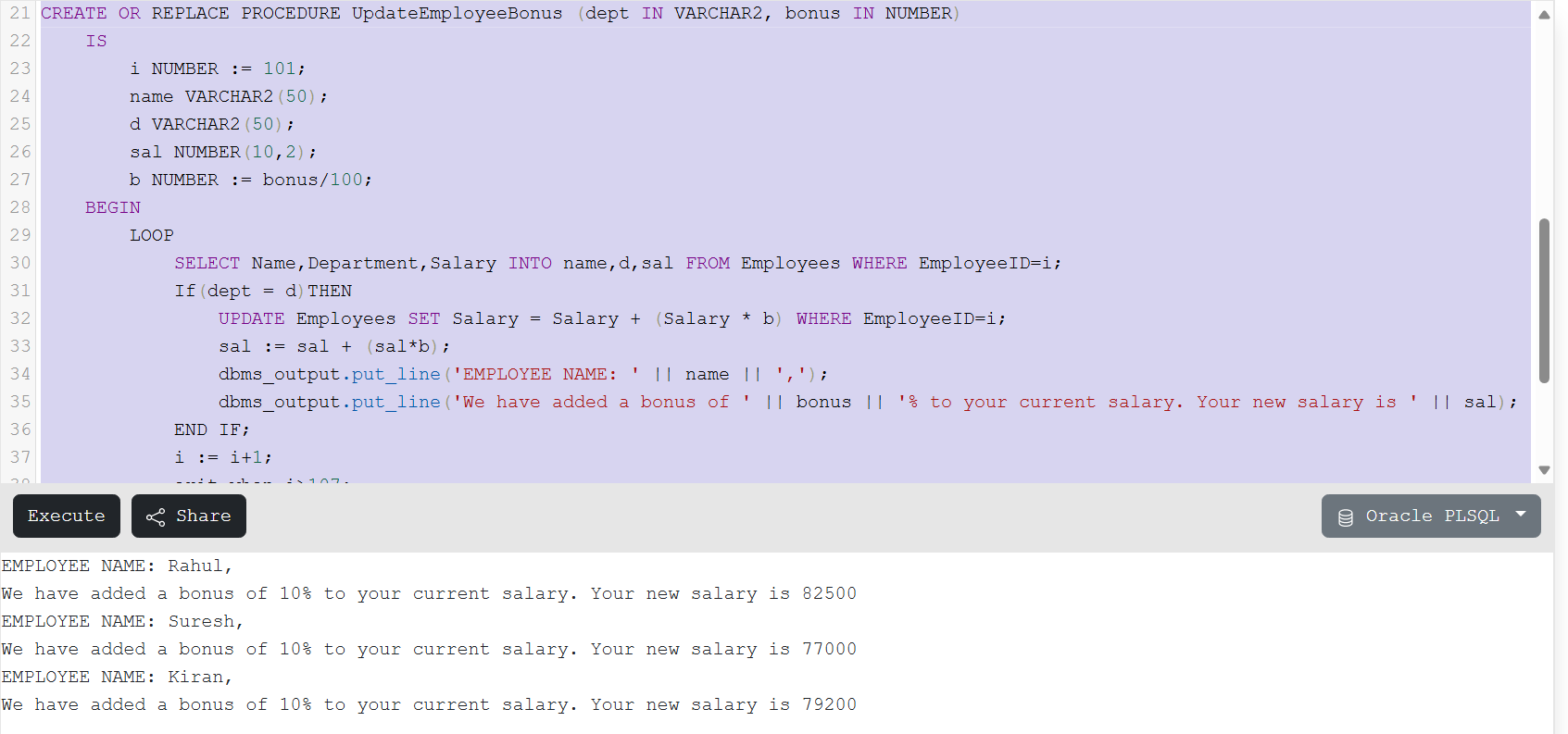
COMMIT;

END;

/

--PL/SQL main

EXECUTE UpdateEmployeeBonus('IT',10);

Output  
  


**Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days.

**o Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

CODE

-- Initializing database

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerName VARCHAR2(50),

Balance NUMBER(10,2)

);

-- Inserting Sample Records

INSERT INTO Accounts VALUES (201, 'Alice', 10000.00);

INSERT INTO Accounts VALUES (202, 'Bob', 8000.00);

INSERT INTO Accounts VALUES (203, 'Charlie', 12000.00);

INSERT INTO Accounts VALUES (204, 'Diana', 5000.00);

INSERT INTO Accounts VALUES (205, 'Edward', 15000.00);

COMMIT;

-- PL/SQL Procedure

CREATE OR REPLACE PROCEDURE TransferFunds(srcid IN NUMBER, destid IN NUMBER, amt NUMBER)

IS

srcname VARCHAR2(50);

destname VARCHAR2(50);

srcbal NUMBER(10,2);

destbal NUMBER(10,2);

BEGIN

SELECT CustomerName,Balance INTO srcname,srcbal FROM Accounts WHERE AccountID=srcid;

SELECT CustomerName,Balance INTO destname,destbal FROM Accounts WHERE AccountID=destid;

IF(srcbal-amt>0)THEN

srcbal := srcbal - amt;

destbal := destbal + amt;

dbms\_output.put\_line('TRANSFER SUCCESSFUL');

dbms\_output.put\_line('Amount Transferred: ' || amt);

dbms\_output.put\_line('FROM Account ID: ' || srcid || ' |Name: '|| srcname || ' |Remaining Balance: ' || srcbal);

dbms\_output.put\_line('TO Account ID: ' || destid || ' |Name: '|| destname || ' |Remaining Balance: ' || destbal);

ELSE

dbms\_output.put\_line('TRANSFER FAILED DUE TO INSUFFICIENT BALANCE');

dbms\_output.put\_line('Available Balance: ' || srcbal || ' |Requested Amount: ' || amt);

END IF;

COMMIT;

END;

/

--PL/SQL main

EXECUTE TransferFunds(201,203, 2000);

EXECUTE TransferFunds(204,201, 6000);

