**Exercise 1: Control Structures**

**Code:**

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

BEGIN

FOR rec IN (

SELECT l.LoanID, l.InterestRate

FROM Customers c

JOIN Loans l ON c.CustomerID = l.CustomerID

WHERE c.Age > 60

)

LOOP

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE LoanID = rec.LoanID;

DBMS\_OUTPUT.PUT\_LINE('Applied 1% discount to Loan ID ' || rec.LoanID);

END LOOP;

COMMIT;

END;

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

BEGIN

FOR rec IN (

SELECT CustomerID

FROM Customers

WHERE Balance > 10000

)

LOOP

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = rec.CustomerID;

DBMS\_OUTPUT.PUT\_LINE('Customer ID ' || rec.CustomerID || ' marked as VIP');

END LOOP;

COMMIT;

END;

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

DECLARE

v\_message VARCHAR2(200);

BEGIN

FOR rec IN (

SELECT c.CustomerName, l.DueDate

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

WHERE l.DueDate BETWEEN SYSDATE AND SYSDATE + 30

)

LOOP

v\_message := 'Reminder: Dear ' || rec.CustomerName ||

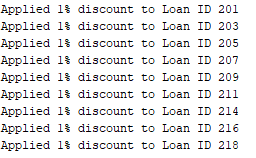
', your loan is due on ' || TO\_CHAR(rec.DueDate, 'DD-MON-YYYY');

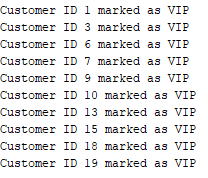
DBMS\_OUTPUT.PUT\_LINE(v\_message);

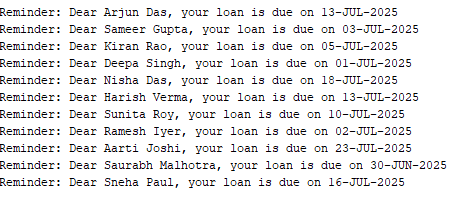
END LOOP;

END;

**Output**:







**Exercise 3: Stored Procedures**

**Code:**

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

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR rec IN (SELECT AccountID, Balance FROM SavingsAccounts)

LOOP

UPDATE SavingsAccounts

SET Balance = Balance + (rec.Balance \* 0.01)

WHERE AccountID = rec.AccountID;

DBMS\_OUTPUT.PUT\_LINE('Interest added to Account ID: ' || rec.AccountID);

END LOOP;

COMMIT;

END;

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

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_dept\_id IN NUMBER,

p\_bonus\_pct IN NUMBER -- e.g., 10 for 10%

) IS

BEGIN

FOR rec IN (

SELECT EmpID, Salary FROM Employees WHERE DepartmentID = p\_dept\_id

)

LOOP

UPDATE Employees

SET Salary = Salary + (rec.Salary \* (p\_bonus\_pct / 100))

WHERE EmpID = rec.EmpID;

DBMS\_OUTPUT.PUT\_LINE('Bonus applied to Emp ID: ' || rec.EmpID);

END LOOP;

COMMIT;

END;

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

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_from\_acc IN NUMBER,

p\_to\_acc IN NUMBER,

p\_amount IN NUMBER

) IS

v\_from\_balance NUMBER;

BEGIN

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

IF v\_from\_balance < p\_amount THEN

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

END IF;

UPDATE Accounts

SET Balance = Balance - p\_amount

WHERE AccountID = p\_from\_acc;

UPDATE Accounts

SET Balance = Balance + p\_amount

WHERE AccountID = p\_to\_acc;

DBMS\_OUTPUT.PUT\_LINE('Transferred ₹' || p\_amount || ' from Acc ' || p\_from\_acc || ' to Acc ' || p\_to\_acc);

COMMIT;

END;