# Exercise 1: Control Structures

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

**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.

**Script:**

DECLARE

v\_discount\_rate NUMBER := 1;

v\_customer\_age NUMBER;

v\_loan\_count NUMBER;

v\_new\_rate NUMBER;

CURSOR c\_customers IS

SELECT CustomerID, Name, DOB FROM Customers;

CURSOR c\_loans(p\_customer\_id NUMBER) IS

SELECT LoanID, InterestRate

FROM Loans

WHERE CustomerID = p\_customer\_id

FOR UPDATE;

BEGIN

FOR cust\_rec IN c\_customers LOOP

v\_customer\_age := FLOOR(MONTHS\_BETWEEN(SYSDATE, cust\_rec.DOB)/12);

IF v\_customer\_age > 60 THEN

SELECT COUNT(\*) INTO v\_loan\_count

FROM Loans

WHERE CustomerID = cust\_rec.CustomerID;

IF v\_loan\_count > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Processing senior customer: ' || cust\_rec.Name ||

' (Age: ' || v\_customer\_age || ')');

FOR loan\_rec IN c\_loans(cust\_rec.CustomerID) LOOP

v\_new\_rate := GREATEST(loan\_rec.InterestRate - v\_discount\_rate, 0);

UPDATE Loans

SET InterestRate = v\_new\_rate

WHERE CURRENT OF c\_loans;

DBMS\_OUTPUT.PUT\_LINE('Applied 1% discount to loan ' || loan\_rec.LoanID ||

' for customer ' || cust\_rec.CustomerID ||

'. Old rate: ' || loan\_rec.InterestRate || '%' ||

', New rate: ' || v\_new\_rate || '%');

END LOOP;

END IF;

END IF;

END LOOP;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Senior customer loan discount processing complete.');

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

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

END;

/

**Script output:**

Senior customer loan discount processing complete.

PL/SQL procedure successfully completed.

**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.

**Script :**

BEGIN

EXECUTE IMMEDIATE 'ALTER TABLE Customers ADD IsVIP VARCHAR2(5) DEFAULT ''FALSE''';

EXCEPTION

WHEN OTHERS THEN

IF SQLCODE = -1430 THEN

NULL;

ELSE

RAISE;

END IF;

END;

/

DECLARE

v\_vip\_threshold NUMBER := 10000;

v\_updated\_rows NUMBER := 0;

BEGIN

UPDATE Customers

SET IsVIP = 'TRUE',

LastModified = SYSDATE

WHERE Balance > v\_vip\_threshold

AND (IsVIP IS NULL OR IsVIP = 'FALSE');

v\_updated\_rows := SQL%ROWCOUNT;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Promoted ' || v\_updated\_rows || ' customers to VIP status.');

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

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

END;

/

**Script output:**

Promoted 0 customers to VIP status.

PL/SQL procedure successfully completed.

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

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

**Script:**

DECLARE

v\_reminder\_days NUMBER := 30;

v\_reminder\_count NUMBER := 0;

CURSOR c\_due\_loans IS

SELECT l.LoanID,

l.CustomerID,

c.Name AS CustomerName,

l.EndDate,

FLOOR(l.EndDate - SYSDATE) AS DaysRemaining

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + v\_reminder\_days

ORDER BY l.EndDate;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('=== Loan Payment Reminders (Next ' || v\_reminder\_days || ' Days) ===');

DBMS\_OUTPUT.PUT\_LINE('');

FOR loan\_rec IN c\_due\_loans LOOP

DBMS\_OUTPUT.PUT\_LINE('Customer: ' || loan\_rec.CustomerName || ' (ID: ' || loan\_rec.CustomerID || ')');

DBMS\_OUTPUT.PUT\_LINE('Loan ID: ' || loan\_rec.LoanID);

DBMS\_OUTPUT.PUT\_LINE('Due Date: ' || TO\_CHAR(loan\_rec.EndDate, 'YYYY-MM-DD'));

DBMS\_OUTPUT.PUT\_LINE('Days Remaining: ' || loan\_rec.DaysRemaining);

DBMS\_OUTPUT.PUT\_LINE('Message: Dear ' || loan\_rec.CustomerName || ', your loan payment is due soon.');

DBMS\_OUTPUT.PUT\_LINE('----------------------------------------');

v\_reminder\_count := v\_reminder\_count + 1;

END LOOP;

IF v\_reminder\_count = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('No loans due in the next ' || v\_reminder\_days || ' days.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Total reminders sent: ' || v\_reminder\_count);

END IF;

END;

/

**Script output:**

=== Loan Payment Reminders (Next 30 Days) ===

No loans due in the next 30 days.

PL/SQL procedure successfully completed.

**DBMS\_OUTPUT for scenario 1,2,3:**

