**Control Structures**

**Scenario 1: Discount on Loan Interest Rates for Seniors**

**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.

DECLARE

CURSOR customer\_cursor IS

SELECT c.CustomerID, l.InterestRate

FROM Customers c

JOIN Loans l ON c.CustomerID = l.CustomerID

WHERE EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM c.DOB) > 60;

v\_customerID Customers.CustomerID%TYPE;

v\_interestRate Loans.InterestRate%TYPE;

BEGIN

OPEN customer\_cursor;

LOOP

FETCH customer\_cursor INTO v\_customerID, v\_interestRate;

EXIT WHEN customer\_cursor%NOTFOUND;

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE CustomerID = v\_customerID;

DBMS\_OUTPUT.PUT\_LINE('Updated interest rate for Customer ID: ' || v\_customerID);

END LOOP;

CLOSE customer\_cursor;

END;

**Scenario 2: Promote Customers to VIP Status**

**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.  
  
DECLARE

CURSOR vip\_cursor IS

SELECT CustomerID

FROM Customers

WHERE Balance > 10000;

v\_customerID Customers.CustomerID%TYPE;

BEGIN

OPEN vip\_cursor;

LOOP

FETCH vip\_cursor INTO v\_customerID;

EXIT WHEN vip\_cursor%NOTFOUND;

-- Assuming there's an IsVIP column in Customers table

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = v\_customerID;

DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || v\_customerID || ' promoted to VIP.');

END LOOP;

CLOSE vip\_cursor;

END;  
  
  
  
  
  
  
  
**Scenario 3: Loan Reminders**

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

DECLARE

CURSOR reminder\_cursor IS

SELECT l.LoanID, c.Name, l.EndDate

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30;

v\_loanID Loans.LoanID%TYPE;

v\_customerName Customers.Name%TYPE;

v\_dueDate Loans.EndDate%TYPE;

BEGIN

OPEN reminder\_cursor;

LOOP

FETCH reminder\_cursor INTO v\_loanID, v\_customerName, v\_dueDate;

EXIT WHEN reminder\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || v\_loanID || ' for ' || v\_customerName || ' is due on ' || TO\_CHAR(v\_dueDate, 'YYYY-MM-DD'));

END LOOP;

CLOSE reminder\_cursor;

END;