**PL/SQL Programming**

**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.

DECLARE

v\_customer\_id Customers.CustomerID%TYPE;

v\_dob Customers.DOB%TYPE;

v\_age NUMBER;

v\_loan\_id Loans.LoanID%TYPE;

v\_interest\_rate Loans.InterestRate%TYPE;

v\_discount CONSTANT NUMBER := 0.01; -- 1% discount

BEGIN

FOR customer IN (SELECT CustomerID, DOB FROM Customers) LOOP

*-- Calculate the age of the customer*

v\_age := TRUNC(MONTHS\_BETWEEN(SYSDATE, customer.DOB) / 12);

IF v\_age > 60 THEN

*-- Loop through the loans of the customer*

FOR loan IN (SELECT LoanID, InterestRate FROM Loans WHERE CustomerID =

customer.CustomerID) LOOP

v\_loan\_id := loan.LoanID;

v\_interest\_rate := loan.InterestRate;

*-- Apply the 1% discount to the interest rate*

v\_interest\_rate := v\_interest\_rate - v\_discount;

*-- Update the loan with the new interest rate*

UPDATE Loans

SET InterestRate = v\_interest\_rate

WHERE LoanID = v\_loan\_id;

END LOOP;

END IF;

END LOOP;

COMMIT;

END;

/

**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.

Add a new column ‘IsVIP’ to the ‘Customers’ table to store the VIP status.

ALTER TABLE Customers ADD IsVIP VARCHAR2(3);

DECLARE

v\_customer\_id Customers.CustomerID%TYPE;

v\_balance Customers.Balance%TYPE;

BEGIN

FOR customer IN (SELECT CustomerID, Balance FROM Customers) LOOP

v\_customer\_id := customer.CustomerID;

v\_balance := customer.Balance;

IF v\_balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = v\_customer\_id;

ELSE

UPDATE Customers

SET IsVIP = 'FALSE'

WHERE CustomerID = v\_customer\_id;

END IF;

END LOOP;

COMMIT;

END;

/

**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.

DECLARE

CURSOR c\_due\_loans IS

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

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30;

v\_loan\_id Loans.LoanID%TYPE;

v\_customer\_id Loans.CustomerID%TYPE;

v\_end\_date Loans.EndDate%TYPE;

v\_customer\_name Customers.Name%TYPE;

BEGIN

OPEN c\_due\_loans;

LOOP

FETCH c\_due\_loans INTO v\_loan\_id, v\_customer\_id, v\_end\_date, v\_customer\_name;

EXIT WHEN c\_due\_loans%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Reminder: Dear ' || v\_customer\_name ||

', your loan (ID: ' || v\_loan\_id ||

') is due on ' || TO\_CHAR(v\_end\_date, 'DD-MON-YYYY') ||

'. Please ensure timely payment.');

END LOOP;

CLOSE c\_due\_loans;

END;

/