**PL/SQL EXERCISES:**

**Exercise 1: Control Structures**

**Scenario 1:**

DECLARE

    CURSOR customer\_loans IS

        SELECT l.LoanID, l.InterestRate, c.CustomerID, c.DOB

        FROM Loans l

        JOIN Customers c ON l.CustomerID = c.CustomerID;

    v\_age NUMBER;

    v\_discount\_rate NUMBER := 1; -- 1% discount

BEGIN

    FOR loan\_rec IN customer\_loans LOOP

        -- Calculate customer age

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

        -- Check if customer is above 60

        IF v\_age > 60 THEN

            -- Apply discount to interest rate

            UPDATE Loans

            SET InterestRate = InterestRate - v\_discount\_rate

            WHERE LoanID = loan\_rec.LoanID;

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

                                 ' for customer ' || loan\_rec.CustomerID);

        END IF;

    END LOOP;

    COMMIT;

    DBMS\_OUTPUT.PUT\_LINE('Discount application completed.');

EXCEPTION

    WHEN OTHERS THEN

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

        ROLLBACK;

END;

/

**Scenario 2:**

DECLARE

    CURSOR customer\_cursor IS

        SELECT CustomerID, Balance

        FROM Customers;

    v\_vip\_threshold NUMBER := 10000; -- $10,000 threshold

BEGIN

    FOR cust\_rec IN customer\_cursor LOOP

        -- Check if customer balance exceeds threshold

        IF cust\_rec.Balance > v\_vip\_threshold THEN

            -- Update IsVIP flag (assuming the column exists)

            -- If the column doesn't exist, you would need to add it first

            UPDATE Customers

            SET IsVIP = 'TRUE'

            WHERE CustomerID = cust\_rec.CustomerID;

            DBMS\_OUTPUT.PUT\_LINE('Promoted customer ' || cust\_rec.CustomerID ||

                                 ' to VIP status');

        END IF;

    END LOOP;

    COMMIT;

    DBMS\_OUTPUT.PUT\_LINE('VIP promotion completed.');

EXCEPTION

    WHEN OTHERS THEN

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

        ROLLBACK;

END;

/

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

**Scenario 3:**

DECLARE

    CURSOR due\_loans IS

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

        FROM Loans l

        JOIN Customers c ON l.CustomerID = c.CustomerID

        WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30;

    v\_days\_remaining NUMBER;

BEGIN

    FOR loan\_rec IN due\_loans LOOP

        -- Calculate days remaining

        v\_days\_remaining := loan\_rec.EndDate - SYSDATE;

        -- Print reminder message

        DBMS\_OUTPUT.PUT\_LINE('Reminder: Customer ' || loan\_rec.Name ||

                             ' (ID: ' || loan\_rec.CustomerID || ') has loan ' ||

                             loan\_rec.LoanID || ' due in ' || v\_days\_remaining ||

                             ' days on ' || TO\_CHAR(loan\_rec.EndDate, 'YYYY-MM-DD'));

    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE('Reminder process completed. ' ||

                         SQL%ROWCOUNT || ' reminders sent.');

EXCEPTION

    WHEN OTHERS THEN

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

END;

/

