

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

CURSOR customer_cursor IS

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

    FROM Customers c

    JOIN Loans l ON c.CustomerID = l.CustomerID

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

BEGIN

    FOR loan_record IN customer_cursor LOOP

        UPDATE Loans

        SET InterestRate = GREATEST(0, InterestRate - 1)

        WHERE LoanID = loan_record.LoanID;

        DBMS_OUTPUT.PUT_LINE('Applied 1% discount to loan ID: ' || loan_record.LoanID);

    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.

```
ALTER TABLE Customers ADD IsVIP CHAR(1);

DECLARE

    CURSOR customer_cursor IS

        SELECT CustomerID, Balance

        FROM Customers;

BEGIN

    FOR customer_record IN customer_cursor LOOP

        IF customer_record.Balance > 10000 THEN

            UPDATE Customers

            SET IsVIP = 'Y'

            WHERE CustomerID = customer_record.CustomerID;

        ELSE

            UPDATE Customers

            SET IsVIP = 'N'

            WHERE CustomerID = customer_record.CustomerID;

        END IF;

        DBMS_OUTPUT.PUT_LINE('Customer ID: ' || customer_record.CustomerID ||

            ' Balance: ' || customer_record.Balance ||

            ' IsVIP: ' || (CASE WHEN customer_record.Balance > 10000 THEN 'Y' ELSE 'N' END));

    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 loan_cursor IS

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

    FROM Customers c

    JOIN Loans l ON c.CustomerID = l.CustomerID

    WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30;

BEGIN

FOR loan_record IN loan_cursor LOOP

    DBMS_OUTPUT.PUT_LINE(

        'Reminder: Loan ID ' || loan_record.LoanID ||

        ' for customer ' || loan_record.Name ||

        ' is due on ' || TO_CHAR(loan_record.EndDate, 'YYYY-MM-DD') || ' '

    );

END LOOP;

END;
```