**WEEK – 2**

**PLSQL**

**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.

**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.

**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.

**Scenario 1: Apply 1% Discount for Customers Above 60**

DECLARE

TYPE CustomerRec IS RECORD (

customer\_id NUMBER,

age NUMBER,

interest\_rate NUMBER

);

TYPE CustomerTable IS TABLE OF CustomerRec INDEX BY PLS\_INTEGER;

customers CustomerTable;

BEGIN

-- Sample data

customers(1) := CustomerRec(1, 65, 7.5);

customers(2) := CustomerRec(2, 45, 6.8);

customers(3) := CustomerRec(3, 70, 8.2);

-- Loop through and apply discount

FOR i IN customers.FIRST .. customers.LAST LOOP

IF customers(i).age > 60 THEN

customers(i).interest\_rate := customers(i).interest\_rate - 1;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || customers(i).customer\_id ||

' | Age: ' || customers(i).age ||

' | New Interest Rate: ' || customers(i).interest\_rate);

END LOOP;

END;

**Output:**

Customer ID: 1 | Age: 65 | New Interest Rate: 6.5

Customer ID: 2 | Age: 45 | New Interest Rate: 6.8

Customer ID: 3 | Age: 70 | New Interest Rate: 7.2

**Scenario 2: Promote to VIP Based on Balance**

DECLARE

TYPE CustomerRec IS RECORD (

customer\_id NUMBER,

balance NUMBER,

is\_vip BOOLEAN

);

TYPE CustomerTable IS TABLE OF CustomerRec INDEX BY PLS\_INTEGER;

customers CustomerTable;

BEGIN

-- Sample data

customers(1) := CustomerRec(1, 12000, FALSE);

customers(2) := CustomerRec(2, 5000, FALSE);

customers(3) := CustomerRec(3, 20000, FALSE);

-- Loop to set VIP flag

FOR i IN customers.FIRST .. customers.LAST LOOP

IF customers(i).balance > 10000 THEN

customers(i).is\_vip := TRUE;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || customers(i).customer\_id ||

' | Balance: ' || customers(i).balance ||

' | Is VIP: ' || CASE WHEN customers(i).is\_vip THEN 'YES' ELSE 'NO' END);

END LOOP;

END;

**Output:**

Customer ID: 1 | Balance: 12000 | Is VIP: YES

Customer ID: 2 | Balance: 5000 | Is VIP: NO

Customer ID: 3 | Balance: 20000 | Is VIP: YES

**Scenario 3: Send Reminders for Loans Due in Next 30 Days**

DECLARE

TYPE LoanRec IS RECORD (

customer\_name VARCHAR2(100),

due\_date DATE

);

TYPE LoanTable IS TABLE OF LoanRec INDEX BY PLS\_INTEGER;

loans LoanTable;

BEGIN

-- Sample data (Assume today is 2025-06-27)

loans(1) := LoanRec('Alice', SYSDATE + 10); -- due in 10 days

loans(2) := LoanRec('Bob', SYSDATE + 40); -- due in 40 days

loans(3) := LoanRec('Charlie', SYSDATE + 25);-- due in 25 days

FOR i IN loans.FIRST .. loans.LAST LOOP

IF loans(i).due\_date <= SYSDATE + 30 THEN

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan for ' || loans(i).customer\_name ||

' is due on ' || TO\_CHAR(loans(i).due\_date, 'DD-Mon-YYYY'));

END IF;

END LOOP;

END;

**Output:**

Reminder: Loan for Alice is due on 07-Jul-2025

Reminder: Loan for Charlie is due on 22-Jul-2025