**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.

**MY ANSWERS :**

SET SERVEROUTPUT ON;

BEGIN

FOR rec IN (

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

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

)

LOOP

IF MONTHS\_BETWEEN(SYSDATE, rec.DOB) / 12 > 60 THEN

UPDATE Loans

SET InterestRate = rec.InterestRate - 1

WHERE LoanID = rec.LoanID;

DBMS\_OUTPUT.PUT\_LINE('Discount applied for LoanID: ' || rec.LoanID);

ELSE

DBMS\_OUTPUT.PUT\_LINE('No discount (customer age below 60) for LoanID: ' || rec.LoanID);

END IF;

END LOOP;

END;

/

**OUTPUT :**

**Before :**

1. John Doe was born in 1985 → Age = 39
2. Jane Smith was born in 1990 → Age = 34

**After :**

No discount (customer age below 60) for LoanID: 1

**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.

**MY ANSWERS :**

SET SERVEROUTPUT ON;

BEGIN

FOR rec IN (SELECT CustomerID, Balance FROM Customers)

LOOP

IF rec.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = rec.CustomerID;

DBMS\_OUTPUT.PUT\_LINE('Customer ID ' || rec.CustomerID || ' promoted to VIP.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Customer ID ' || rec.CustomerID || ' NOT eligible for VIP.');

END IF;

END LOOP;

END;

/

**OUTPUT :**

Test Case 1:

1) Customer ID 1 NOT eligible for VIP.

2)Customer ID 2 NOT eligible for VIP.

Test Case 2 :

Table :

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (3, 'High Balance User', TO\_DATE('1970-01-01', 'YYYY-MM-DD'), 20000, SYSDATE);

OutPut :

Customer ID 1 NOT eligible for VIP.

Customer ID 2 NOT eligible for VIP.

Customer ID 3 promoted to VIP.

**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.

**MY ANSWERS :**

SET SERVEROUTPUT ON;

BEGIN

FOR rec IN (

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

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

WHERE l.EndDate <= SYSDATE + 30

)

LOOP

DBMS\_OUTPUT.PUT\_LINE('📢 Reminder: Loan ' || rec.LoanID || ' for ' || rec.Name || ' is due on ' || TO\_CHAR(rec.EndDate, 'YYYY-MM-DD'));

END LOOP;

END;

/

**OUTPUT :**

Case 1 :

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (3, 1, 3000, 5, SYSDATE, SYSDATE + 25); -- due in 25 days

Output :

Reminder: Loan 3 for John Doe is due on 2025-07-16

Case 2 :

Loan due in 10 days (for Jane)

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (4, 2, 4500, 4, SYSDATE, SYSDATE + 10);

Output :

Reminder: Loan 3 for John Doe is due on 2025-07-16

Reminder: Loan 4 for Jane Smith is due on 2025-07-01