**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.

**CODE:**

SET SERVEROUTPUT ON;

DECLARE

CURSOR cust\_cursor IS

SELECT customer\_id, age, loan\_interest\_rate

FROM customers

FOR UPDATE;

v\_new\_rate NUMBER;

BEGIN

FOR cust\_rec IN cust\_cursor LOOP

IF cust\_rec.age > 60 THEN

v\_new\_rate := cust\_rec.loan\_interest\_rate - (cust\_rec.loan\_interest\_rate \* 0.01);

UPDATE customers

SET loan\_interest\_rate = v\_new\_rate

WHERE customer\_id = cust\_rec.customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || cust\_rec.customer\_id ||

' | New Interest Rate: ' || v\_new\_rate);

END IF;

END LOOP;

COMMIT;

END;

/

**OUTPUT:**

Customer ID: 101 | New Interest Rate: 6.93

Customer ID: 109 | New Interest Rate: 5.94

Customer ID: 120 | New Interest Rate: 7.92

**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.

**CODE:**

SET SERVEROUTPUT ON;

DECLARE

CURSOR cust\_cursor IS

SELECT customer\_id, balance

FROM customers

FOR UPDATE;

BEGIN

FOR cust\_rec IN cust\_cursor LOOP

IF cust\_rec.balance > 10000 THEN

UPDATE customers

SET IsVIP = 'TRUE'

WHERE customer\_id = cust\_rec.customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || cust\_rec.customer\_id || ' promoted to VIP.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || cust\_rec.customer\_id || ' not eligible for VIP.');

END IF;

END LOOP;

COMMIT;

END;

/

**OUTPUT:**

Customer ID: 101 promoted to VIP.

Customer ID: 102 not eligible for VIP.

Customer ID: 103 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.

**CODE:**

SET SERVEROUTPUT ON;

DECLARE

CURSOR due\_loan\_cursor IS

SELECT customer\_id, loan\_id, due\_date

FROM loans

WHERE due\_date BETWEEN SYSDATE AND SYSDATE + 30;

BEGIN

FOR loan\_rec IN due\_loan\_cursor LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Customer ID ' || loan\_rec.customer\_id ||

' has Loan ID ' || loan\_rec.loan\_id ||

' due on ' || TO\_CHAR(loan\_rec.due\_date, 'DD-MON-YYYY') ||

'. Please make the payment on time.');

END LOOP;

END;

/

**OUTPUT:**

Reminder: Customer ID 101 has Loan ID L001 due on 28-JUN-2025. Please make the payment on time.

Reminder: Customer ID 108 has Loan ID L014 due on 10-JUL-2025. Please make the payment on time.