**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.

**PL/SQL Query:**

SQL> DECLARE

2 CURSOR cur\_customers IS

3 SELECT CustomerID, DOB

4 FROM Customers;

5 v\_customer\_id Customers.CustomerID%TYPE;

6 v\_dob Customers.DOB%TYPE;

7 v\_age NUMBER;

8 BEGIN

9 FOR rec IN cur\_customers LOOP

10 v\_customer\_id := rec.CustomerID;

11 v\_dob := rec.DOB;

12

13 v\_age := TRUNC((SYSDATE - v\_dob) / 365);

14

15 IF v\_age > 60 THEN

16 UPDATE Loans

17 SET InterestRate = InterestRate - 1

18 WHERE CustomerID = v\_customer\_id;

19 END IF;

20 END LOOP;

21

22 COMMIT;

23 END;

24 /

**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.

**PL/SQL Query:**

SQL> CREATE GLOBAL TEMPORARY TABLE TempVIPCustomers (

2 CustomerID NUMBER,

3 IsVIP VARCHAR2(3)

4 ) ON COMMIT DELETE ROWS;

Table created.

SQL> BEGIN

2 INSERT INTO TempVIPCustomers (CustomerID, IsVIP)

3 SELECT CustomerID, 'TRUE'

4 FROM Customers

5 WHERE Balance > 10000;

6

7 COMMIT;

8 END;

9 /

**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.

**PL/SQL Query:**

SQL> SET SERVEROUTPUT ON;

SQL>

SQL> DECLARE

2 CURSOR cur\_loans IS

3 SELECT LoanID, CustomerID, EndDate

4 FROM Loans

5 WHERE EndDate BETWEEN SYSDATE AND SYSDATE + 30;

6

7 v\_loan\_id Loans.LoanID%TYPE;

8 v\_customer\_id Loans.CustomerID%TYPE;

9 v\_end\_date Loans.EndDate%TYPE;

10 v\_customer\_name Customers.Name%TYPE;

11 BEGIN

12 FOR rec IN cur\_loans LOOP

13 v\_loan\_id := rec.LoanID;

14 v\_customer\_id := rec.CustomerID;

15 v\_end\_date := rec.EndDate;

16

17 BEGIN

18 SELECT Name INTO v\_customer\_name

19 FROM Customers

20 WHERE CustomerID = v\_customer\_id;

21

22 DBMS\_OUTPUT.PUT\_LINE('Reminder: Dear ' || v\_customer\_name || ', your loan with LoanID ' || v\_loan\_id || ' is due on ' || TO\_CHAR(v\_end\_date, 'YYYY-MM-DD') || '.');

23 EXCEPTION

24 WHEN NO\_DATA\_FOUND THEN

25 DBMS\_OUTPUT.PUT\_LINE('Reminder: LoanID ' || v\_loan\_id || ' is due soon, but customer details are not available.');

26 END;

27 END LOOP;

28 END;

29 /