**Exercise 1: Control Structures**

Scenario 1

SQL> DECLARE

2 v\_age NUMBER; -- Variable to store customer age

3 v\_loan\_id NUMBER; -- Variable to store loan ID

4 v\_current\_rate NUMBER; -- Variable to store current loan interest rate

5 BEGIN

6 -- Loop through all customers and their loan information

7 FOR customer IN (

8 SELECT c.CustomerID, c.DOB, l.LoanID, l.LoanAmount, l.InterestRate

9 FROM Customers c

10 JOIN Loans l ON c.CustomerID = l.CustomerID

11 ) LOOP

12 -- Calculate age

13 v\_age := TRUNC(MONTHS\_BETWEEN(SYSDATE, customer.DOB) / 12);

14

15 -- Get current loan interest rate

16 v\_loan\_id := customer.LoanID;

17 v\_current\_rate := customer.InterestRate;

18

19 -- Apply 1% discount if the customer is older than 60

20 IF v\_age > 60 THEN

21 UPDATE Loans

22 SET InterestRate = v\_current\_rate \* 0.99

23 WHERE LoanID = v\_loan\_id;

24

25 -- Output to show which loan was updated

26 DBMS\_OUTPUT.PUT\_LINE('Applied 1% discount for loan ' || v\_loan\_id || ' of customer

' || customer.CustomerID);

27 END IF;

28 END LOOP;

29

30 COMMIT; -- Commit the changes to the database

31 END;

32 /

Applied 1% discount for loan 1001 of customer 1

Applied 1% discount for loan 1005 of customer 2

PL/SQL procedure successfully completed.

Scenario 2

SQL> DECLARE

2 v\_balance NUMBER; -- Variable to store customer balance

3 v\_customer\_id NUMBER; -- Variable to store customer ID

4 BEGIN

5 -- Loop through all customers

6 FOR customer IN (

7 SELECT CustomerID, Balance

8 FROM Customers

9 ) LOOP

10 v\_balance := customer.Balance;

11 v\_customer\_id := customer.CustomerID;

12

13 -- Set IsVIP flag to 'TRUE' if the customer's balance is above $10,000

14 IF v\_balance > 10000 THEN

15 UPDATE Customers

16 SET IsVIP = 'T'

17 WHERE CustomerID = v\_customer\_id;

18

19 -- Output to show the promotion

20 DBMS\_OUTPUT.PUT\_LINE('Customer ' || v\_customer\_id || ' promoted to VIP status.');

21 END IF;

22 END LOOP;

23

24 COMMIT; -- Commit the changes to the database

25 END;

26 /

Customer 4 promoted to VIP status.

Customer 5 promoted to VIP status.

PL/SQL procedure successfully completed.

SQL> DECLARE

2 v\_due\_date DATE; -- Variable to store loan due date

3 v\_customer\_id NUMBER; -- Variable to store customer ID

4 v\_loan\_id NUMBER; -- Variable to store loan ID

5 BEGIN

6 -- Loop through all loans due in the next 30 days

7 FOR loan IN (

8 SELECT LoanID, CustomerID, EndDate

9 FROM Loans

10 WHERE EndDate BETWEEN SYSDATE AND SYSDATE + 30

11 ) LOOP

12 v\_due\_date := loan.EndDate;

13 v\_customer\_id := loan.CustomerID;

14 v\_loan\_id := loan.LoanID;

15

16 -- Send reminder for the loan due soon

17 DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ' || v\_loan\_id || ' for customer ' || v\_customer\_id ||

18 ' is due on ' || TO\_CHAR(v\_due\_date, 'MM/DD/YYYY') || '.');

19 END LOOP;

20 END;

21 /

Reminder: Loan 1001 for customer 1 is due on 07/15/2025.

Reminder: Loan 1004 for customer 2 is due on 07/20/2025.

PL/SQL procedure successfully completed.