**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 Block:**

SET SERVEROUTPUT ON;

DECLARE

CURSOR c IS

SELECT LoanID, CustomerID, InterestRate

FROM Loans

WHERE CustomerID IN (

SELECT CustomerID

FROM Customers

WHERE DOB <= ADD\_MONTHS(TRUNC(SYSDATE), -60\*12)

);

BEGIN

FOR r IN c LOOP

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE LoanID = r.LoanID;

DBMS\_OUTPUT.PUT\_LINE(

'Loan ' || r.LoanID ||

' Customer ' || r.CustomerID ||

' rate ' || TO\_CHAR(r.InterestRate) ||

' -> ' || TO\_CHAR(r.InterestRate - 1)

);

END LOOP;

END;

/

**Table:**

A black background with a black border

AI-generated content may be incorrect.

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**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 Block:**

ALTER TABLE Customers

ADD IsVIP CHAR(1) DEFAULT 'N' CHECK (IsVIP IN ('Y','N'));

SET SERVEROUTPUT ON;

DECLARE

CURSOR c IS

SELECT CustomerID, Balance

FROM Customers;

BEGIN

FOR r IN c LOOP

IF r.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'Y'

WHERE CustomerID = r.CustomerID;

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

END IF;

END LOOP;

END;

/

**Table:**

**A black background with white lines

AI-generated content may be incorrect.**

**Output:**

**A computer screen shot of text

AI-generated content may be incorrect.**

**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 Block:**

SET SERVEROUTPUT ON;

DECLARE

CURSOR c IS

SELECT l.LoanID,

l.CustomerID,

l.EndDate,

c.Name

FROM Loans l

JOIN Customers c ON c.CustomerID = l.CustomerID

WHERE l.EndDate BETWEEN TRUNC(SYSDATE) AND TRUNC(SYSDATE) + 30;

BEGIN

FOR r IN c LOOP

DBMS\_OUTPUT.PUT\_LINE(

'Reminder: Loan ' || r.LoanID ||

' for Customer ' || r.CustomerID || ' (' || r.Name || ')' ||

' is due on ' || TO\_CHAR(r.EndDate, 'YYYY-MM-DD')

);

END LOOP;

END;

/

**Table:**

A screen shot of a black and white screen

AI-generated content may be incorrect.

**Output:**

A computer screen shot of a black screen

AI-generated content may be incorrect.