**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.

-> **Solution:**

BEGIN

FOR customer\_rec IN (SELECT CustomerID, TRUNC(MONTHS\_BETWEEN(SYSDATE, DOB) / 12) AS Age FROM Customers)

LOOP

IF customer\_rec.Age > 60 THEN

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE CustomerID = customer\_rec.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

/

**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.

-> **Solution:**

BEGIN

FOR customer\_rec IN (SELECT CustomerID, Balance FROM Customers)

LOOP

IF customer\_rec.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = TRUE

WHERE CustomerID = customer\_rec.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

/

x

**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.

-> **Solution:**

BEGIN

FOR loan\_rec IN (SELECT LoanID, CustomerID, EndDate FROM Loans WHERE EndDate <= SYSDATE + 30)

LOOP

DECLARE

customer\_name VARCHAR2(100);

BEGIN

SELECT Name INTO customer\_name FROM Customers WHERE CustomerID = loan\_rec.CustomerID;

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || loan\_rec.LoanID || ' for customer ' || customer\_name || ' is due on ' || TO\_CHAR(loan\_rec.EndDate, 'YYYY-MM-DD'));

END;

END LOOP;

END;

/