**Scenario 1:** The bank wants to apply a discount to loan interest rates for customers above 60 years old. 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.

**Query :**

DECLARE

CURSOR cust\_cursor IS

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

FROM Customers c

LEFT JOIN Loans l ON c.CustomerID = l.CustomerID;

v\_customer\_id NUMBER;

v\_dob DATE;

v\_loan\_id NUMBER;

v\_interest\_rate NUMBER;

v\_age NUMBER;

BEGIN

OPEN cust\_cursor;

LOOP

FETCH cust\_cursor INTO v\_customer\_id, v\_dob, v\_loan\_id, v\_interest\_rate;

EXIT WHEN cust\_cursor%NOTFOUND;

-- Calculate age

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

-- Apply discount if customer is above 60

IF v\_age > 60 AND v\_loan\_id IS NOT NULL THEN

UPDATE Loans SET InterestRate = InterestRate \* 0.99 WHERE LoanID = v\_loan\_id;

END IF;

END LOOP;

CLOSE cust\_cursor;

END;

**Scenario 2:** A customer can be promoted to VIP status based on their balance. 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.

**Query :**

**Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days. Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

**Query :**

DECLARE

CURSOR loan\_cursor IS

SELECT c.Name, l.LoanID, l.EndDate

FROM Customers c

JOIN Loans l ON c.CustomerID = l.CustomerID

WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30;

v\_name VARCHAR2(100);

v\_loan\_id NUMBER;

v\_end\_date DATE;

BEGIN

OPEN loan\_cursor;

LOOP

FETCH loan\_cursor INTO v\_name, v\_loan\_id, v\_end\_date;

EXIT WHEN loan\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Reminder for ' || v\_name || ': Loan ' || v\_loan\_id || ' is due on ' || v\_end\_date);

END LOOP;

CLOSE loan\_cursor;

END;