

# 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.

## CODE:

```
DECLARE
    v_updated_count NUMBER := 0;
BEGIN
    FOR res IN (
        SELECT c.name, c.customerID, l.loanId, l.interestRate
        FROM loans l
        JOIN customers c ON l.customerID = c.customerID
        WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, c.dob)/12) > 60
    ) LOOP
        UPDATE loans
        SET interestRate = res.interestRate - 1
        WHERE loanId = res.loanId;

        v_updated_count := v_updated_count + 1;

        DBMS_OUTPUT.PUT_LINE('Updated loan ' || res.loanId ||
                              ' for customer ' || res.name ||
                              ' (ID: ' || res.customerID ||
                              '). New rate: ' || (res.interestRate - 1) || '%');
    END LOOP;

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Total loans updated: ' || v_updated_count);
END;
/
```

## OUTPUT:

**Initial Customers Table:**

Live SQL

Worksheet

Library

[ SQL Worksheet ]\*

1

2

SELECT \* FROM CUSTOMERS;

Query result

Script output

DBMS output

Explain Plan

SQL history

Download

Execution time: 0.011 seconds

|   | CUSTOMERID | NAME       | DOB                    | BALANCE | LASTMODIFIED           |
|---|------------|------------|------------------------|---------|------------------------|
| 1 | 1          | John Doe   | 5/15/1985, 12:00:00 AM | 1000    | 6/26/2025, 10:18:03 AM |
| 2 | 2          | Jane Smith | 7/20/1990, 12:00:00 AM | 1500    | 6/26/2025, 10:18:03 AM |

Initial Loans Table:

Live SQL

Worksheet

Library

[ SQL Worksheet ]\*

1

SELECT \* FROM LOANS;

Query result

Script output

DBMS output

Explain Plan

SQL history

Download

Execution time: 0.001 seconds

|   | LOANID | CUSTOMERID | LOANAMOUNT | INTERESTRATE | STARTDATE              | ENDDATE                |
|---|--------|------------|------------|--------------|------------------------|------------------------|
| 1 | 1      | 1          | 5000       | 5            | 6/26/2025, 10:18:03 AM | 6/26/2030, 10:18:03 AM |

Query Output:

Query result

Script output

DBMS output

Explain Plan

SQL history

SQL> DECLARE

v\_updated\_count NUMBER := 0;

BEGIN

FOR res IN (...

Show more...

-----

Total loans updated: 0

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.004

**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.**

### CODE:

ADD isVIP column to the Customers Table.

```
ALTER TABLE CUSTOMERS ADD ISVIP VARCHAR2(5);
```

PL/SQL query to find VIP customers.

```
DECLARE
    vip_count NUMBER := 0;
BEGIN
    FOR account IN (
        SELECT customerid, accountid, balance FROM ACCOUNTS
    ) LOOP
        IF account.balance >= 10000 THEN
            UPDATE CUSTOMERS
            SET isvip = 'TRUE'
            WHERE customerid = account.customerid;
            vip_count := vip_count + 1;
        ELSE
            UPDATE CUSTOMERS
            SET isvip = 'FALSE'
            WHERE customerid = account.customerid;
        END IF;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Total ' || vip_count || ' VIPs found.');
```

### Output:

#### Modified Customers Table:

```
SQL> ALTER TABLE CUSTOMERS ADD ISVIP VARCHAR (5)
```



```
Table CUSTOMERS altered.
```

```
Elapsed: 00:00:00.025
```

Query resultScript outputDBMS outputExplain PlanSQL history

Download

Execution time: 0.008 seconds

|   | CUSTOMERID | NAME       | DOB                    | BALANCE | LASTMODIFIED           | ISVIP  |
|---|------------|------------|------------------------|---------|------------------------|--------|
| 1 | 1          | John Doe   | 5/15/1985, 12:00:00 AM | 1000    | 6/26/2025, 10:18:03 AM | (null) |
| 2 | 2          | Jane Smith | 7/20/1990, 12:00:00 AM | 1500    | 6/26/2025, 10:18:03 AM | (null) |

PL/SQL Query Result:

Live SQLWorksheetLibrary

[ SQL Worksheet ]\*

1

2

3

4

5

6

7

8

9

10

DECLARE

vip\_count NUMBER := 0;

BEGIN

FOR account IN (

SELECT customerid, accountid, balance FROM ACCOUNTS

) LOOP

IF account.balance >= 10000 THEN

UPDATE CUSTOMERS

SET isvip = 'TRUE'

WHERE customerid = account.customerid;

END LOOP;

END;

Query resultScript outputDBMS outputExplain PlanSQL history

SQL> DECLARE  
    vip\_count NUMBER := 0;  
BEGIN  
    FOR account IN (...  
Show more...

Total 0 VIPs found.

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.013

Query resultScript outputDBMS outputExplain PlanSQL history

Download

Execution time: 0.005 seconds

|   | CUSTOMERID | NAME       | DOB                    | BALANCE | LASTMODIFIED           | ISVIP |
|---|------------|------------|------------------------|---------|------------------------|-------|
| 1 | 1          | John Doe   | 5/15/1985, 12:00:00 AM | 1000    | 6/26/2025, 10:18:03 AM | FALSE |
| 2 | 2          | Jane Smith | 7/20/1990, 12:00:00 AM | 1500    | 6/26/2025, 10:18:03 AM | FALSE |

**Scenario 3: The bank wants to send reminders to customers whose loans are due within the next 30 days.**

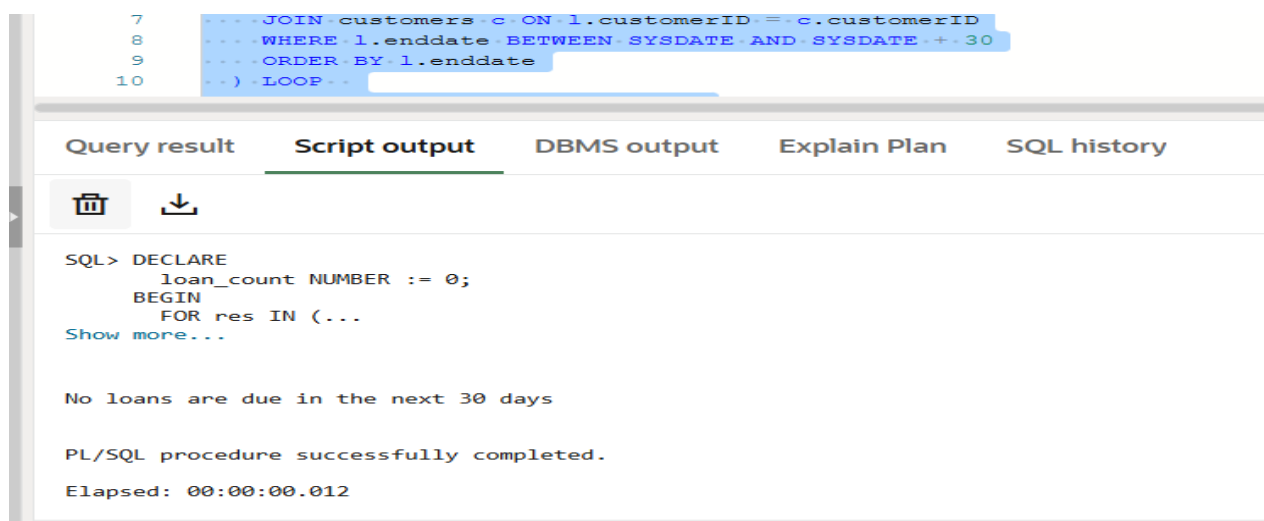
**Question: Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.**

### CODE:

```
DECLARE
    loan_count NUMBER := 0;
BEGIN
    FOR res IN (
        SELECT c.name, c.customerID, l.loanID, l.enddate
        FROM loans l
        JOIN customers c ON l.customerID = c.customerID
        WHERE l.enddate BETWEEN SYSDATE AND SYSDATE + 30
        ORDER BY l.enddate
    ) LOOP
        loan_count := loan_count + 1;
        DBMS_OUTPUT.PUT_LINE(res.name || ', your loan ' || res.loanID ||
            ' is due on ' || TO_CHAR(res.enddate, 'DD-MON-YYYY'));
    END LOOP;



    IF loan_count = 0 THEN
        DBMS_OUTPUT.PUT_LINE('No loans are due in the next 30 days');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Total loans due in next 30 days: ' || loan_count);
    END IF;
END;
/
```

### OUTPUT:



7    -- JOIN customers c ON l.customerID = c.customerID  
8    -- WHERE l.enddate BETWEEN SYSDATE AND SYSDATE + 30  
9    -- ORDER BY l.enddate  
10  -- ) LOOP --

Query result    Script output    DBMS output    Explain Plan    SQL history

SQL> DECLARE  
      loan\_count NUMBER := 0;  
      BEGIN  
      FOR res IN (...  
Show more...

No loans are due in the next 30 days

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.012