**Exercise 1: Control Structures**

**Scenario 1:**

CREATE TABLE Customers (

    CustomerID NUMBER PRIMARY KEY,

    Name VARCHAR2(100),

    DOB DATE,

    Balance NUMBER,

    LastModified DATE

);

CREATE TABLE Accounts (

    AccountID NUMBER PRIMARY KEY,

    CustomerID NUMBER,

    AccountType VARCHAR2(20),

    Balance NUMBER,

    LastModified DATE,

    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

CREATE TABLE Transactions (

    TransactionID NUMBER PRIMARY KEY,

    AccountID NUMBER,

    TransactionDate DATE,

    Amount NUMBER,

    TransactionType VARCHAR2(10),

    FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)

);

CREATE TABLE Loans (

    LoanID NUMBER PRIMARY KEY,

    CustomerID NUMBER,

    LoanAmount NUMBER,

    InterestRate NUMBER,

    StartDate DATE,

    EndDate DATE,

    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

CREATE TABLE Employeess (

    EmployeeID NUMBER PRIMARY KEY,

    Name VARCHAR2(100),

    Position VARCHAR2(50),

    Salary NUMBER,

    Department VARCHAR2(50),

    HireDate DATE

);

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (1, 'John Doe', TO\_DATE('1985-05-15', 'YYYY-MM-DD'), 1000, SYSDATE);

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (2, 'Jane Smith', TO\_DATE('1990-07-20', 'YYYY-MM-DD'), 1500, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (1, 1, 'Savings', 1000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (2, 2, 'Checking', 1500, SYSDATE);

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

VALUES (1, 1, SYSDATE, 200, 'Deposit');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

VALUES (2, 2, SYSDATE, 300, 'Withdrawal');

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (1, 1, 5000, 5, SYSDATE, ADD\_MONTHS(SYSDATE, 60));

INSERT INTO Employeess (EmployeeID, Name, Position, Salary, Department, HireDate)

VALUES (1, 'Alice Johnson', 'Manager', 70000, 'HR', TO\_DATE('2015-06-15', 'YYYY-MM-DD'));

INSERT INTO Employeess (EmployeeID, Name, Position, Salary, Department, HireDate)

VALUES (2, 'Bob Brown', 'Developer', 60000, 'IT', TO\_DATE('2017-03-20', 'YYYY-MM-DD'));

SET SERVEROUTPUT ON;

BEGIN

    FOR cust IN (

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

        FROM Customers c

        JOIN Loans l ON c.CustomerID = l.CustomerID

    )

    LOOP

        IF MONTHS\_BETWEEN(SYSDATE, cust.DOB) / 12 > 60 THEN

            UPDATE Loans

            SET InterestRate = InterestRate - 1

            WHERE LoanID = cust.LoanID;

            DBMS\_OUTPUT.PUT\_LINE('1% discount applied to Customer ID: ' || cust.CustomerID ||

                                 ', Loan ID: ' || cust.LoanID ||

                                 ', New Interest Rate: ' || (cust.InterestRate - 1));

        ELSE

            DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || cust.CustomerID || ' is not eligible (Age <= 60)');

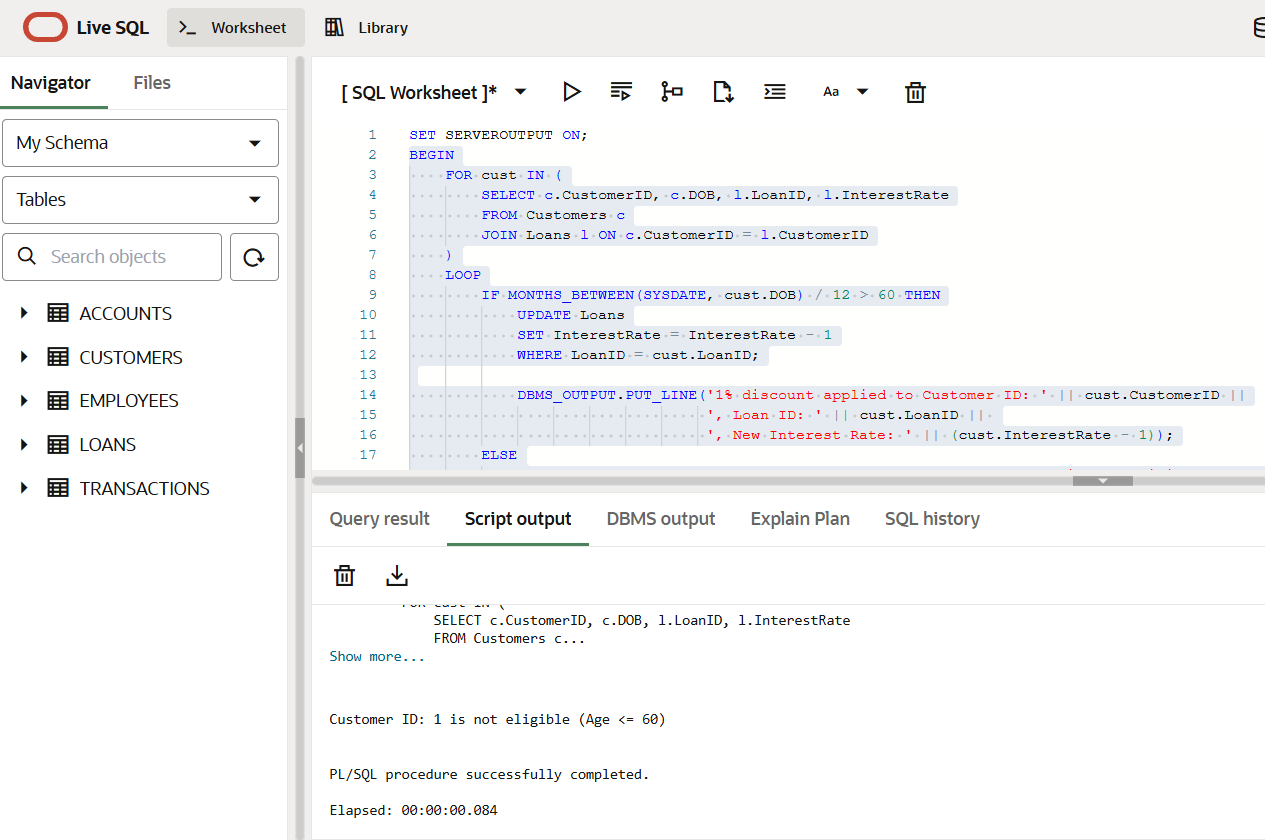
        END IF;

    END LOOP;

END;

/

**Output:**



**Scenario 2:**

ALTER TABLE Customers ADD IsVIP CHAR(1); -- 'Y' for VIP, NULL or 'N' for non-VIP

SET SERVEROUTPUT ON;

BEGIN

    FOR cust IN (

        SELECT CustomerID, Balance FROM Customers

    )

    LOOP

        IF cust.Balance > 10000 THEN

            UPDATE Customers

            SET IsVIP = 'Y'

            WHERE CustomerID = cust.CustomerID;

            DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || cust.CustomerID || ' promoted to VIP (Balance: ' || cust.Balance || ')');

        ELSE

            DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || cust.CustomerID || ' not eligible for VIP (Balance: ' || cust.Balance || ')');

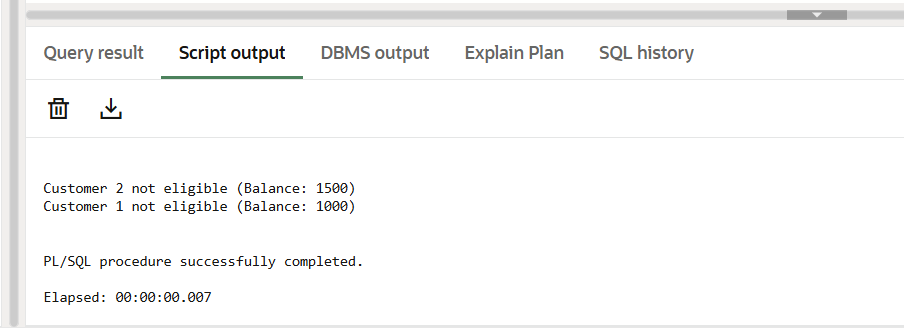
        END IF;

    END LOOP;

END;

/

**Output:**



**Scenario 3:**

SET SERVEROUTPUT ON;

BEGIN

    FOR rec IN (

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

        FROM Loans l

        JOIN Customers c ON l.CustomerID = c.CustomerID

        WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30

    )

    LOOP

        DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || rec.LoanID ||

                             ' for Customer "' || rec.Name || '" is due on ' ||

                             TO\_CHAR(rec.EndDate, 'DD-MON-YYYY'));

    END LOOP;

END;

/

**Output:**

