**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:**

CREATE TABLE Customers (

    CustomerID NUMBER PRIMARY KEY,

    Name VARCHAR2(100),

    DOB DATE,

    isVIP VARCHAR(5),

    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 Employees (

    EmployeeID NUMBER PRIMARY KEY,

    Name VARCHAR2(100),

    Position VARCHAR2(50),

    Salary NUMBER,

    Department VARCHAR2(50),

    HireDate DATE

);

INSERT INTO Customers (CustomerID, Name, DOB, isVIP, Balance, LastModified) VALUES (1, 'Alice Kumar', DATE '1985-05-12', NULL, 9200, SYSDATE);

INSERT INTO Customers (CustomerID, Name, DOB, isVIP, Balance, LastModified) VALUES (2, 'Rahul Singh', DATE '1960-03-20', NULL, 15000, SYSDATE);

INSERT INTO Customers (CustomerID, Name, DOB, isVIP, Balance, LastModified) VALUES (3, 'Fatima Reddy', DATE '1978-11-30', NULL, 3000, SYSDATE);

INSERT INTO Customers (CustomerID, Name, DOB, isVIP, Balance, LastModified) VALUES (4, 'John Doe', DATE '1995-01-01', NULL, 7500, SYSDATE);

INSERT INTO Customers (CustomerID, Name, DOB, isVIP, Balance, LastModified) VALUES (5, 'Sneha Mehta', DATE '1958-09-15', NULL, 12000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (101, 1, 'Savings', 22500, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (102, 2, 'Savings', 91000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (103, 3, 'Current', 1500, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (104, 4, 'Savings', 6000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (105, 5, 'Current', 10000, SYSDATE);

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType) VALUES (1001, 101, SYSDATE - 5, 500, 'Deposit');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType) VALUES (1002, 102, SYSDATE - 3, 1000, 'Withdrawal');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType) VALUES (1003, 103, SYSDATE - 10, 200, 'Deposit');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType) VALUES (1004, 104, SYSDATE - 1, 700, 'Deposit');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType) VALUES (1005, 105, SYSDATE - 2, 1500, 'Withdrawal');

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

VALUES (201, 1, 50000, 7.5, DATE '2022-01-01', DATE '2025-01-01');

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

VALUES (202, 2, 100000, 8.0, DATE '2021-06-15', DATE '2026-06-15');

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

VALUES (203, 3, 30000, 6.0, DATE '2023-03-10', DATE '2024-03-10');

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

VALUES (204, 4, 20000, 7.0, DATE '2022-11-01', DATE '2024-11-01');

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

VALUES (205, 5, 75000, 7.8, DATE '2020-05-01', DATE '2023-05-01');

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

VALUES (301, 'Arun Nair', 'Manager', 75000, 'Finance', DATE '2015-02-01');

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

VALUES (302, 'Priya Iyer', 'Analyst', 55000, 'Risk', DATE '2018-07-10');

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

VALUES (303, 'Vinod Kumar', 'Clerk', 30000, 'Accounts', DATE '2020-01-15');

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

VALUES (304, 'Rita Sharma', 'Executive', 48000, 'Customer Service', DATE '2019-04-20');

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

VALUES (305, 'Manoj Desai', 'IT Support', 40000, 'IT', DATE '2021-09-01');

select \* from Customers;

select COUNT(CustomerID) as empn from Customers;

BEGIN

    FOR rec IN (SELECT CustomerID, DOB FROM Customers) LOOP

        -- dbms\_output.put\_line(TRUNC(MONTHS\_BETWEEN(SYSDATE, rec.DOB) / 12));

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

            UPDATE Loans

            SET InterestRate = InterestRate - 1

            WHERE CustomerID = rec.CustomerID;

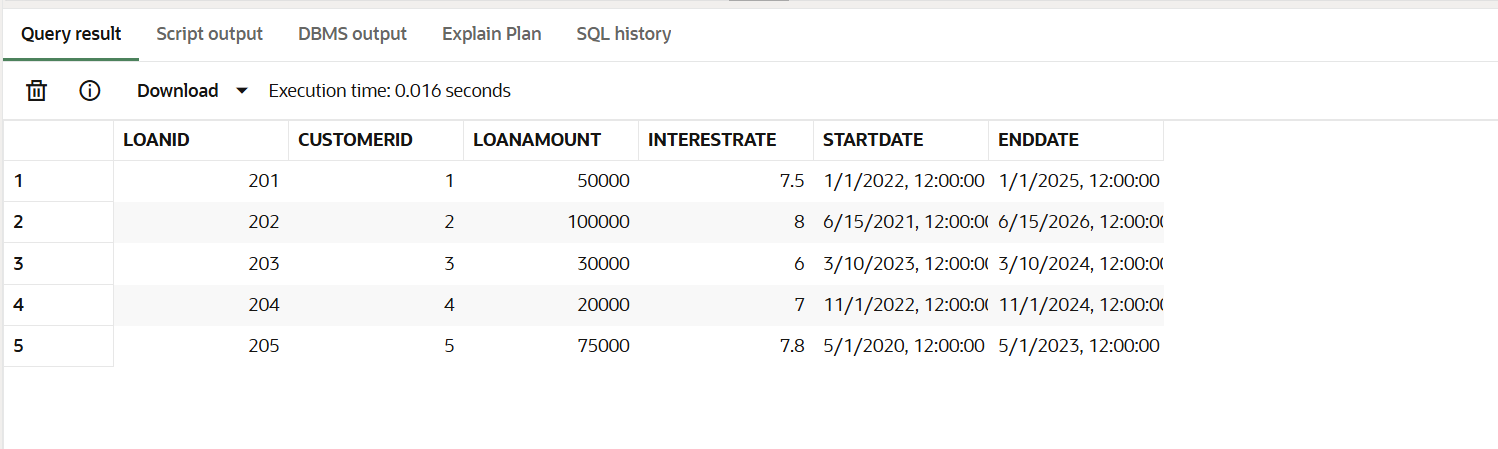
        END IF;

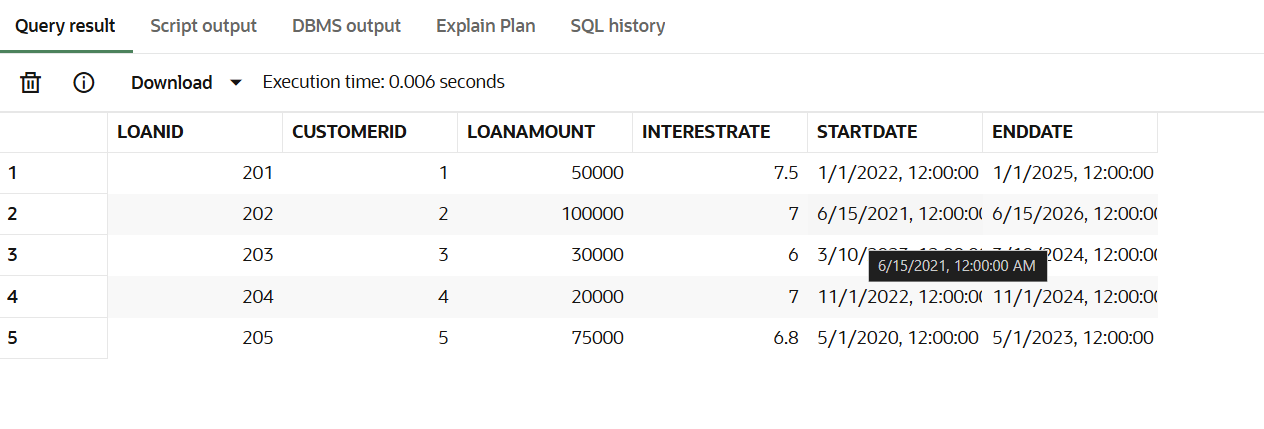
    END LOOP;

END;

/

**OUTPUT:**





**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:**

BEGIN

    for rec in (select isVIP,Balance,CustomerID from Customers) LOOP

        if rec.Balance > 10000 then

            update CUSTOMERS

            set ISVIP = 'true'

            where CUSTOMERID = rec.CUSTOMERID;

        else

            update CUSTOMERS

            set ISVIP = 'false'

            where CUSTOMERID = rec.CUSTOMERID;

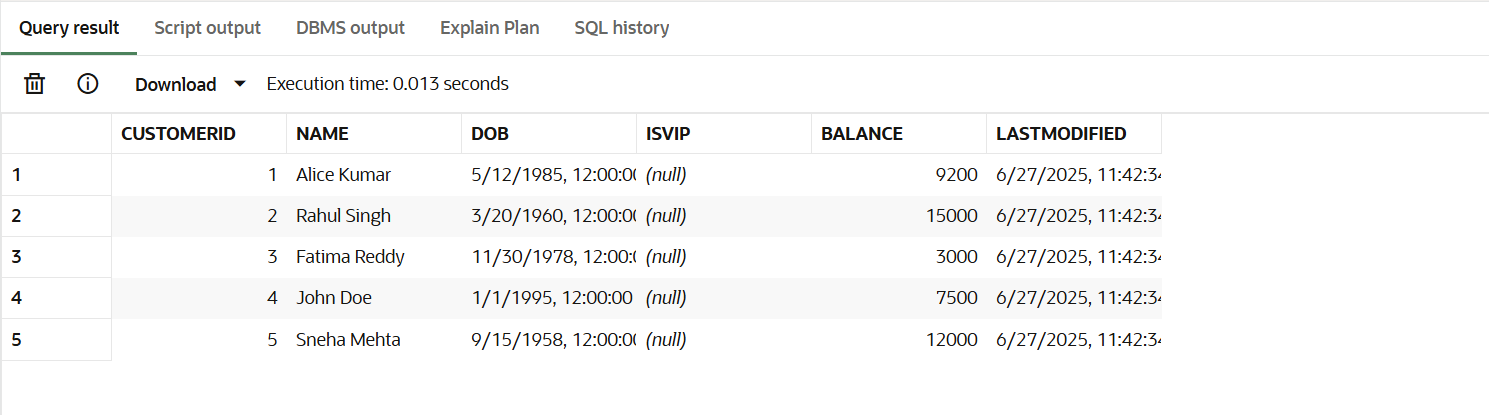
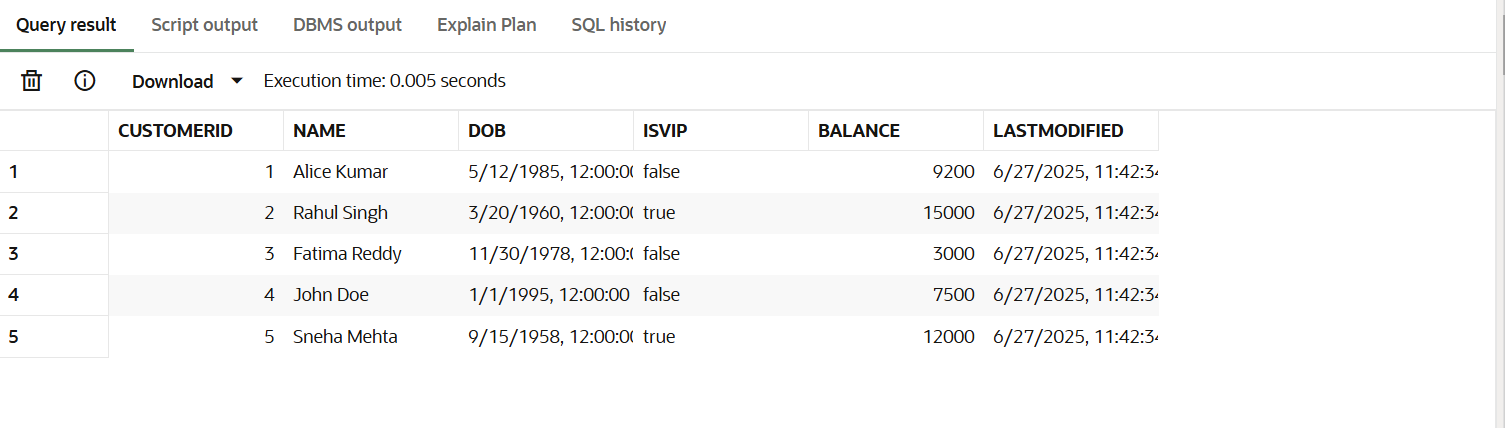
        end if;

    end loop;

end;

/

**OUTPUT:**

**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:**

BEGIN

    for rec in (select c.NAME,c.DOB,l.EndDate from CUSTOMERS c join Loans l on l.CustomerID = c.CustomerID) loop

        IF rec.EndDate > SYSDATE then

            IF TRUNC(MONTHS\_BETWEEN(SYSDATE, rec.DOB) / 12) >= 1 THEN

                DBMS\_OUTPUT.PUT\_LINE(rec.Name || 'your loan due date is near');

            end if;

        end if;

    end loop;

end;

/

**OUTPUT:**

