**PL/SQL PROGRAMMING**

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,

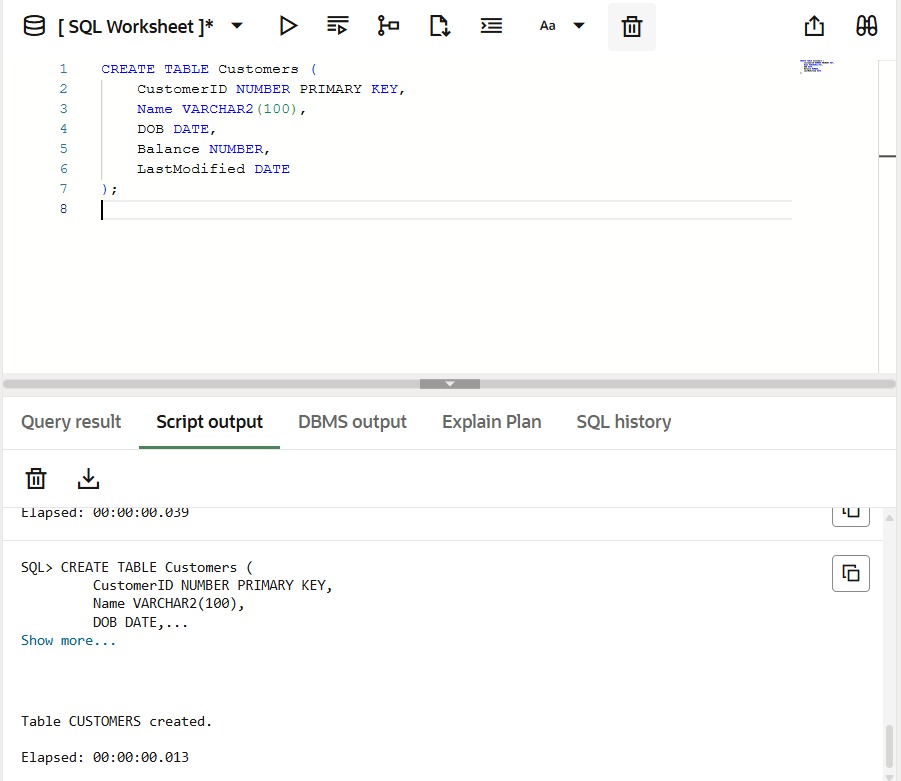
Name VARCHAR2(100),

DOB DATE,

Balance NUMBER,

LastModified DATE

);

**OUTPUT:**

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustomerID NUMBER,

LoanAmount NUMBER,

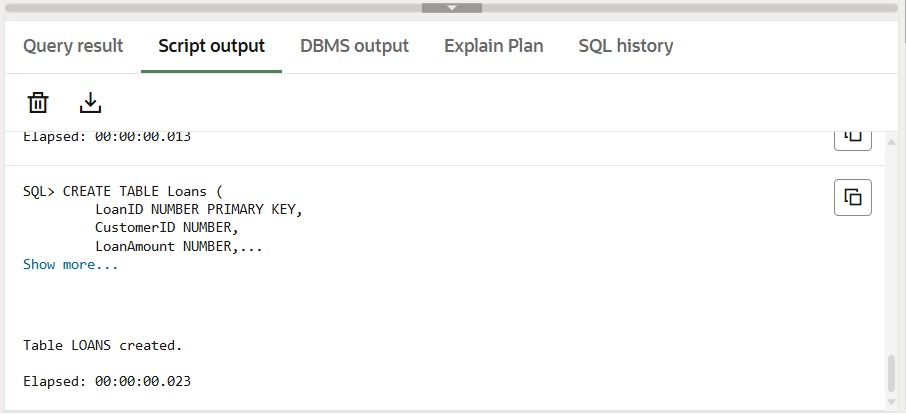
InterestRate NUMBER,

StartDate DATE,

EndDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

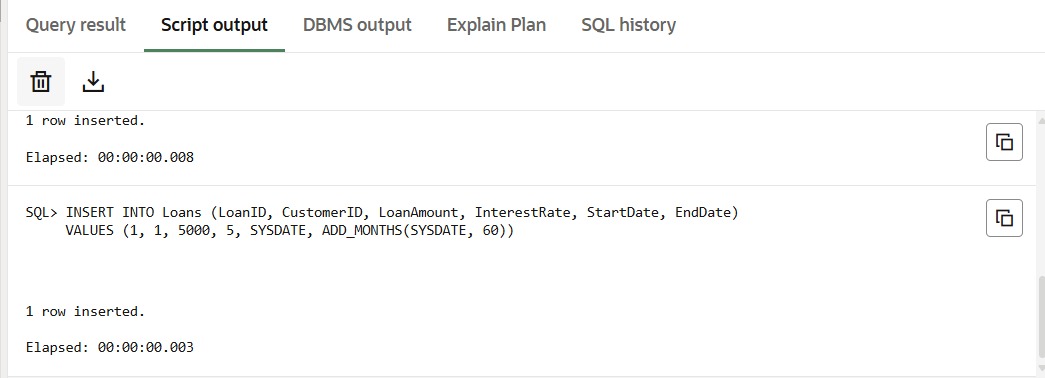
);

**OUTPUT:**

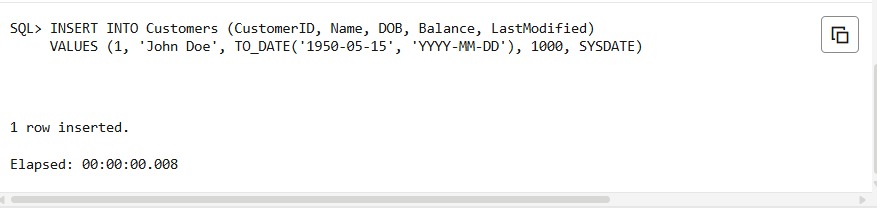
INSERT INTO Customers VALUES (1, 'John Doe', TO\_DATE('1950-05-15', 'YYYY-MM-DD'), 1000, SYSDATE); -- Age > 60

INSERT INTO Customers VALUES (2, 'Jane Smith', TO\_DATE('1990-07-20', 'YYYY-MM-DD'), 1500, SYSDATE); -- Age < 60

INSERT INTO Loans VALUES (1, 1, 5000, 5, SYSDATE, ADD\_MONTHS(SYSDATE, 60)); -- Linked to John

INSERT INTO Loans VALUES (2, 2, 10000, 6, SYSDATE, ADD\_MONTHS(SYSDATE, 48)); -- Linked to Jane

**OUTPUT:**



BEGIN

FOR cust IN (

SELECT l.LoanID, c.DOB

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

) LOOP

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

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE LoanID = cust.LoanID;

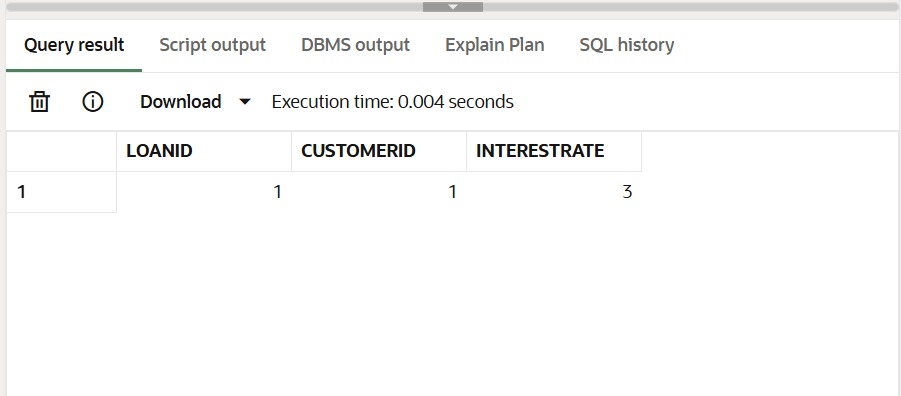
END IF;

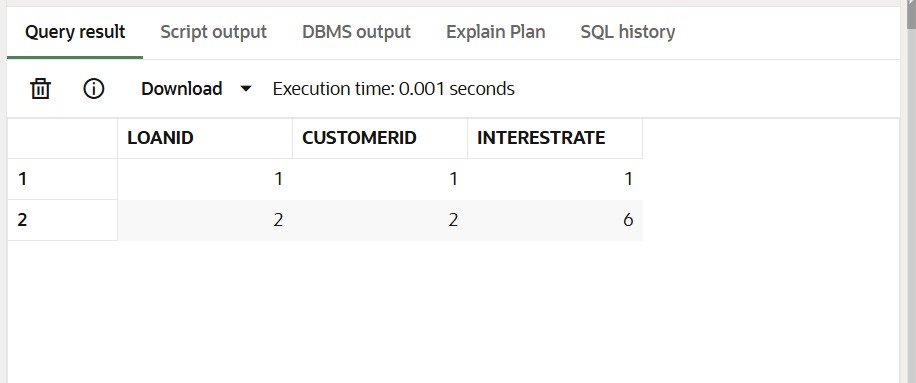
END LOOP;

COMMIT;

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:**

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

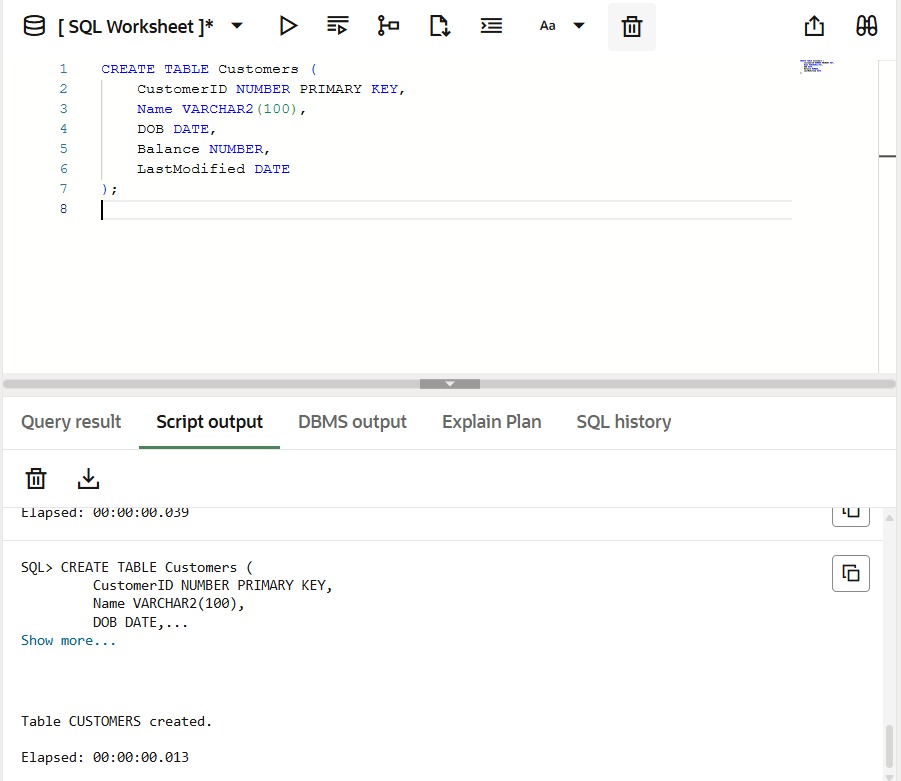
Name VARCHAR2(100),

DOB DATE,

Balance NUMBER,

LastModified DATE

);

**OUTPUT:**

ALTER TABLE Customers ADD IsVIP VARCHAR2(5);

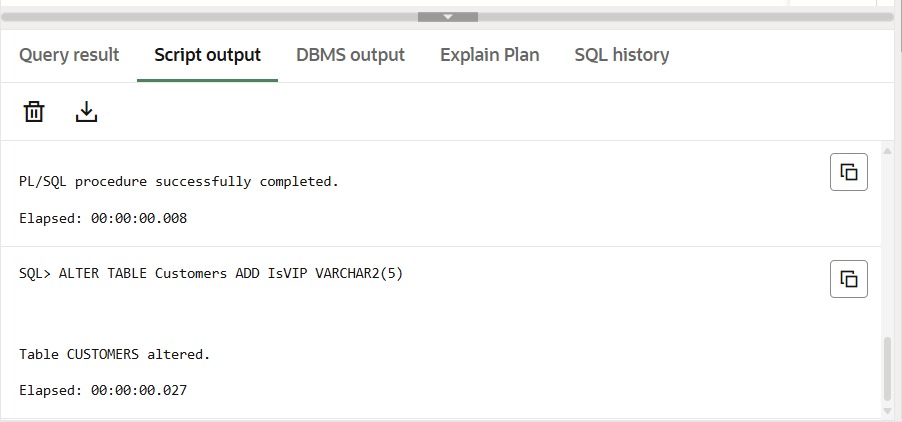
UPDATE Customers SET IsVIP = 'FALSE';

COMMIT;

UPDATE Customers

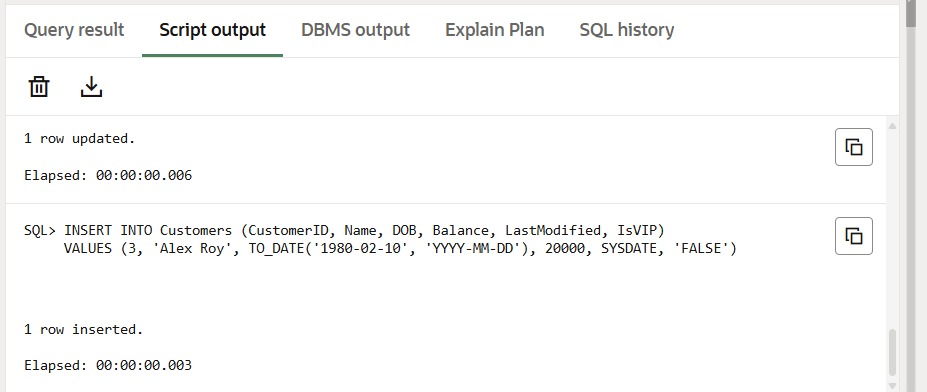
SET Balance = 15000

WHERE CustomerID = 2;

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

VALUES (3, 'Alex Roy', TO\_DATE('1980-02-10', 'YYYY-MM-DD'), 20000, SYSDATE, 'FALSE');

**OUTPUT:**



BEGIN

FOR cust IN (SELECT CustomerID, Balance FROM Customers) LOOP

IF cust.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = cust.CustomerID;

END IF;

END LOOP;

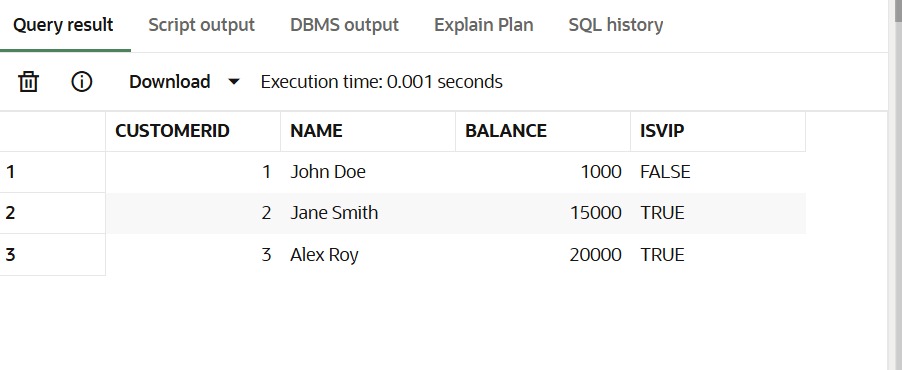
COMMIT;

END;

/

SELECT CustomerID, Name, Balance, IsVIP FROM Customers;

**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:**

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

DOB DATE,

Balance NUMBER,

LastModified DATE

);

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustomerID NUMBER,

LoanAmount NUMBER,

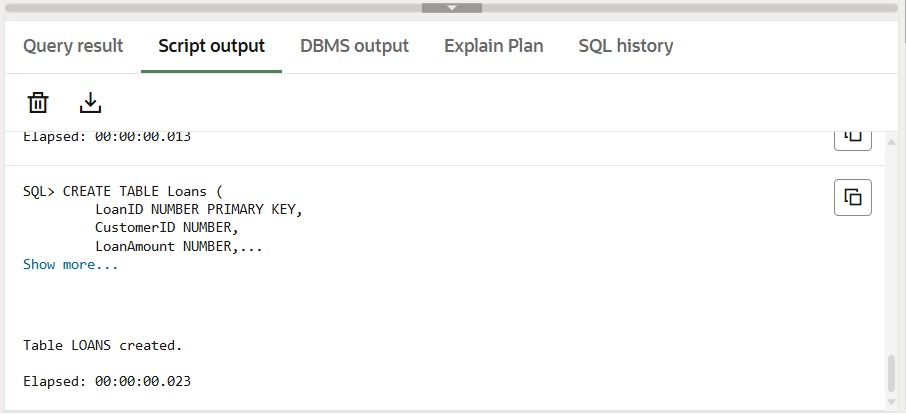
InterestRate NUMBER,

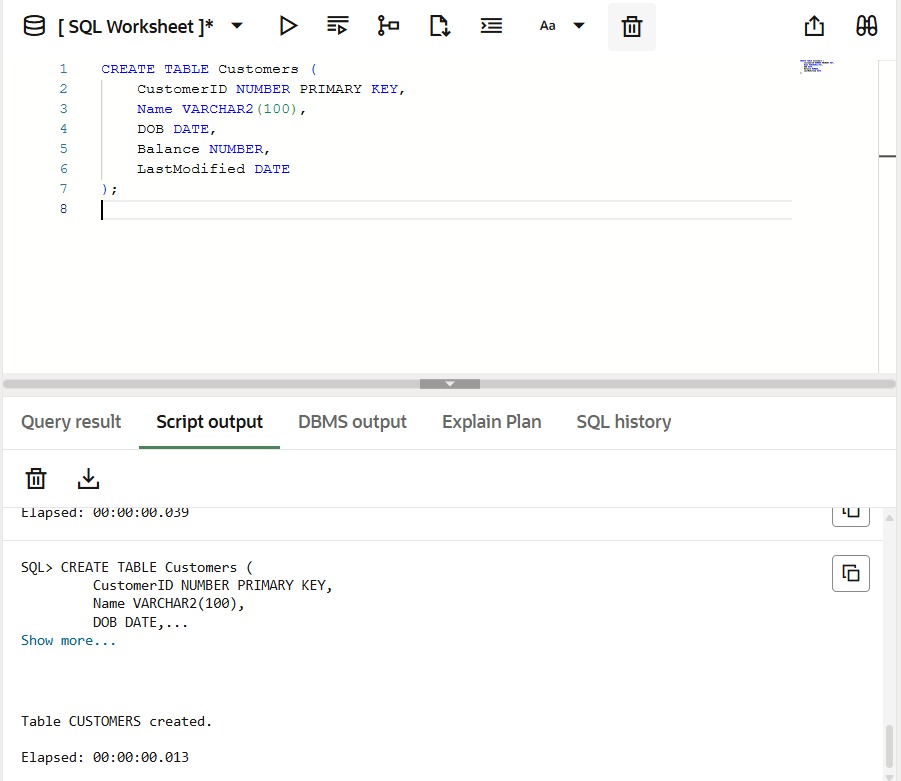
StartDate DATE,

EndDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

**OUTPUT:**



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

VALUES (1, 'John Doe', TO\_DATE('1950-05-15', 'YYYY-MM-DD'), 8000, SYSDATE);

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

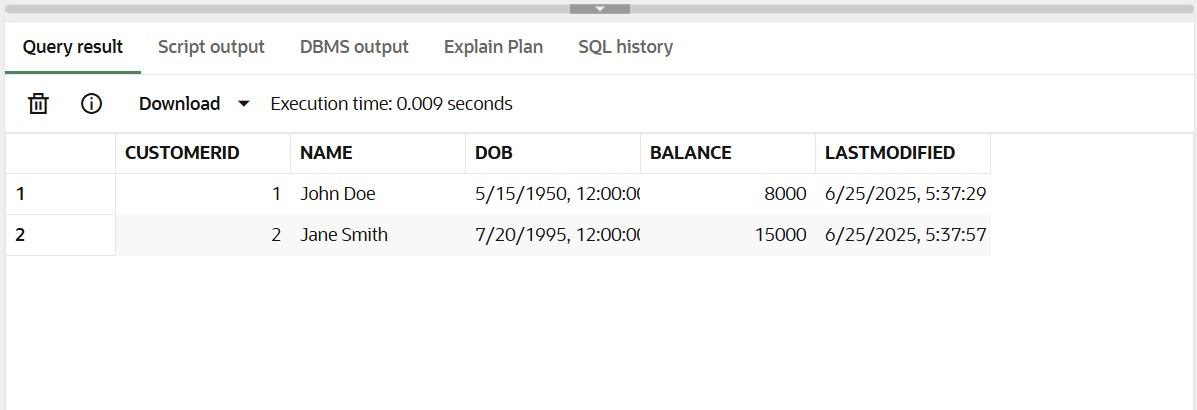
VALUES (2, 'Jane Smith', TO\_DATE('1995-07-20', 'YYYY-MM-DD'), 15000, SYSDATE);

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

VALUES (101, 2, 6000, 5.0, SYSDATE, SYSDATE + 10);

COMMIT;

**OUTPUT:**



BEGIN

FOR loan\_rec IN (

SELECT l.LoanID, 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 ' || loan\_rec.LoanID ||

' for customer ' || loan\_rec.Name ||

' is due on ' || TO\_CHAR(loan\_rec.EndDate, 'DD-MON-YYYY')

);

END LOOP;

END;

/

**OUTPUT:** 