**PL/SQL EXERCISES**

**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.

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

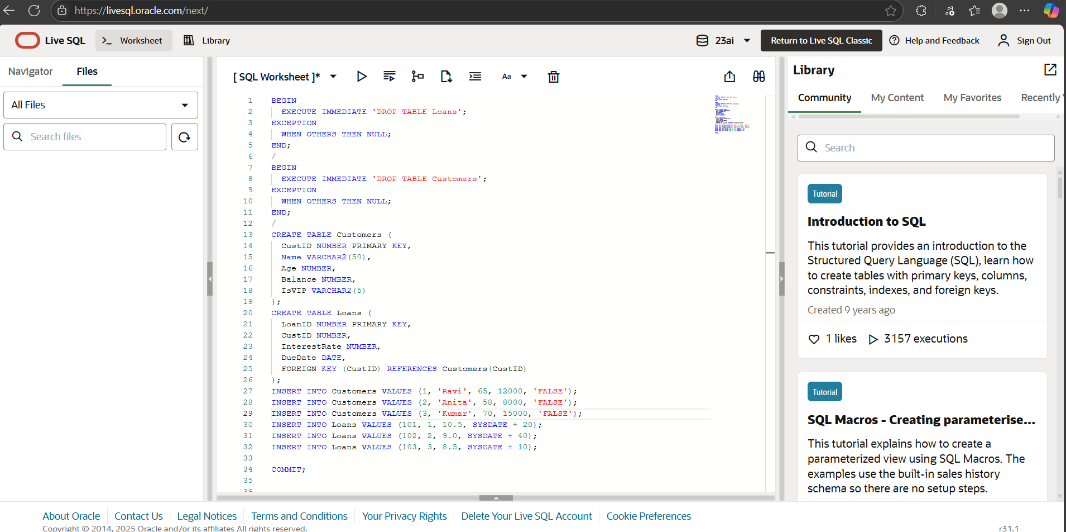
* + **Question:** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.

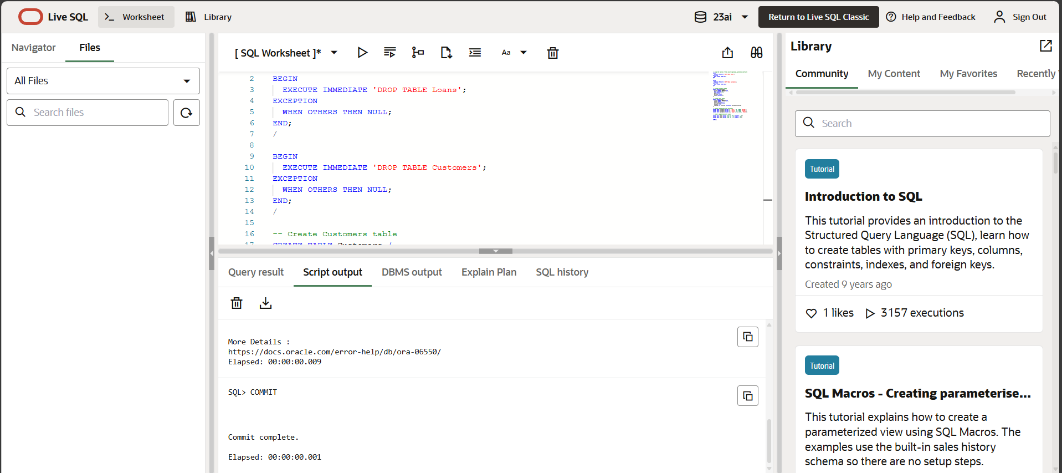
**Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days.

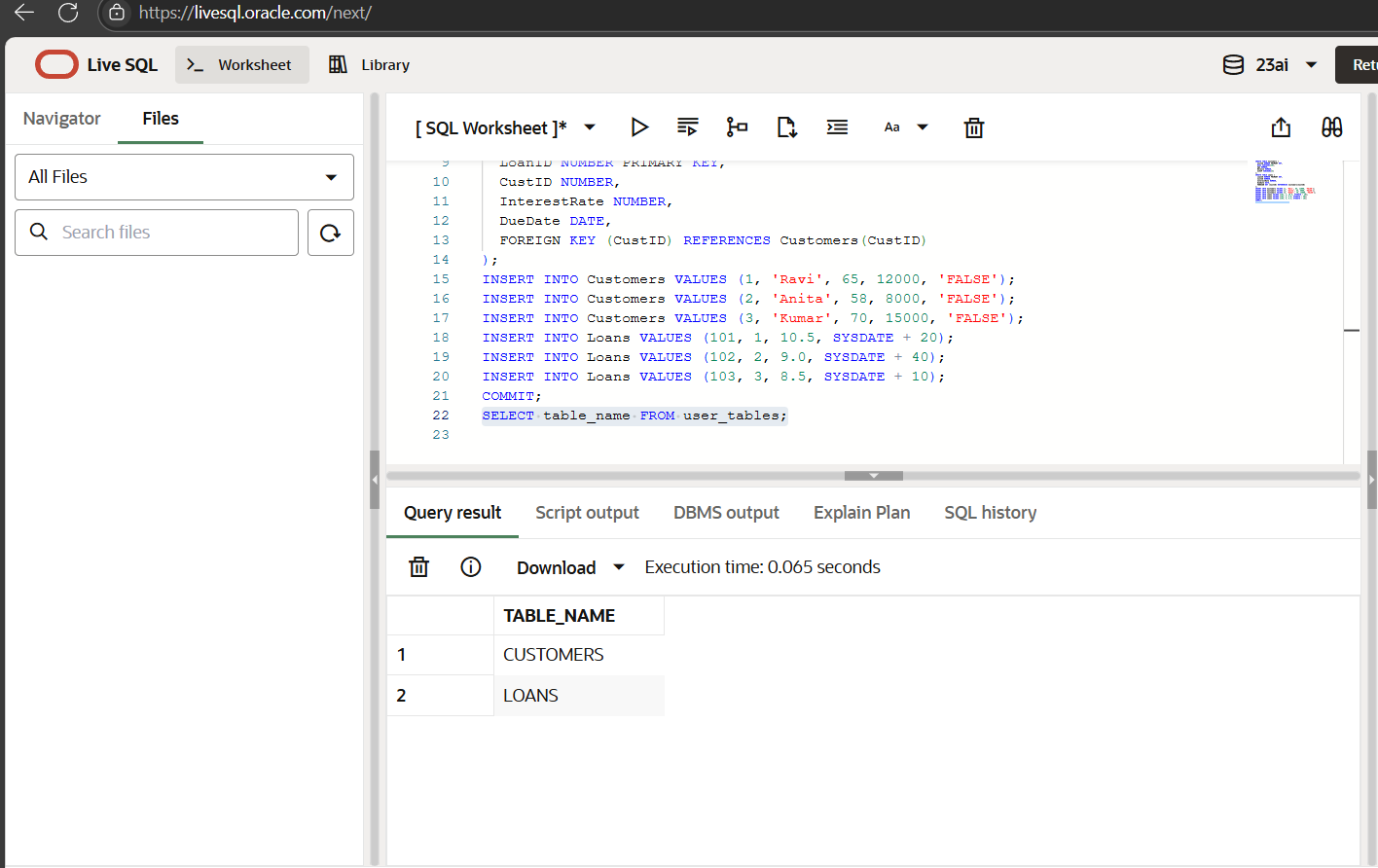
* + **Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

**IMPLEMENTATION IN LIVE SQL :**

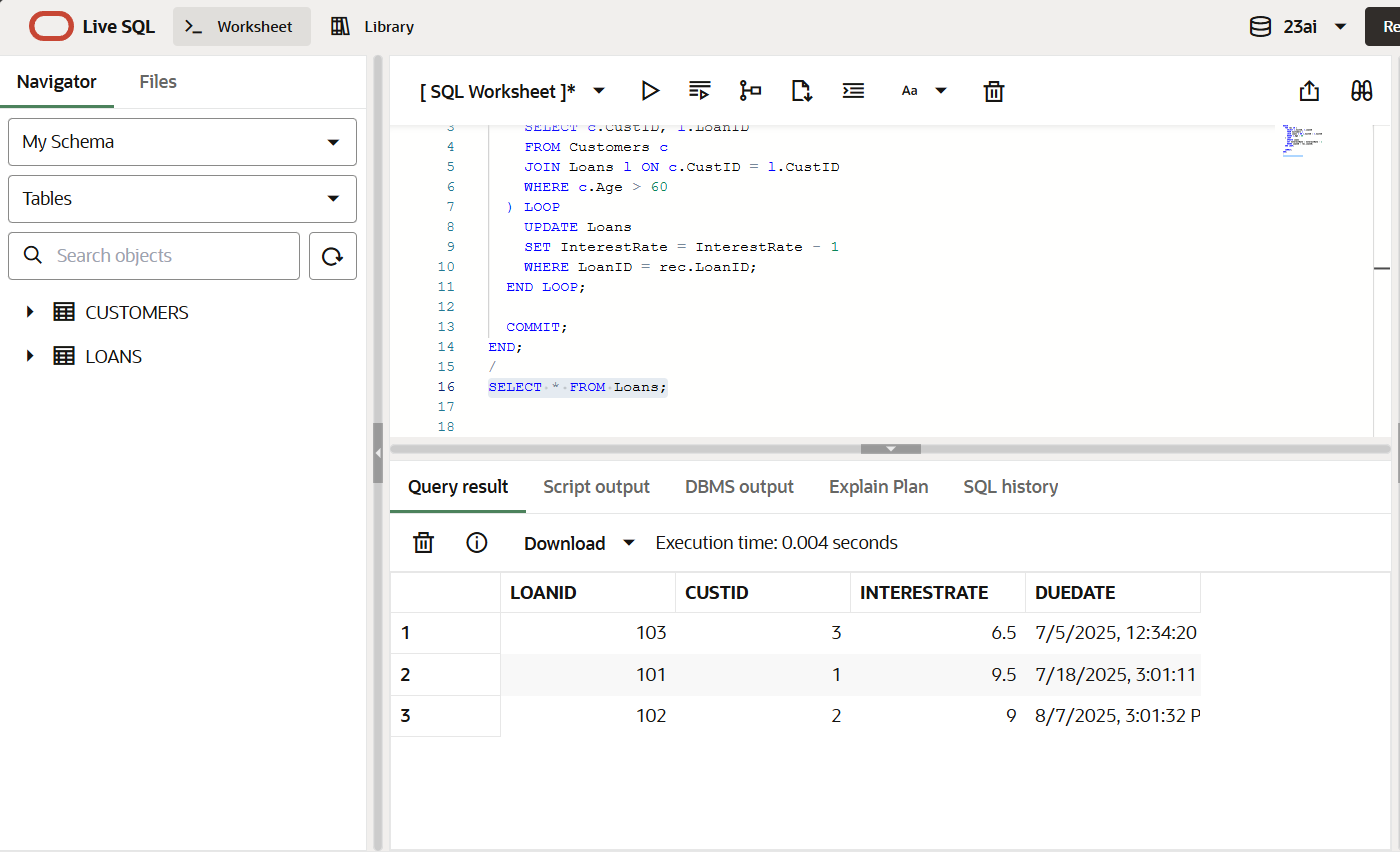
**Creation of Tables and Data stored in Database:**

****

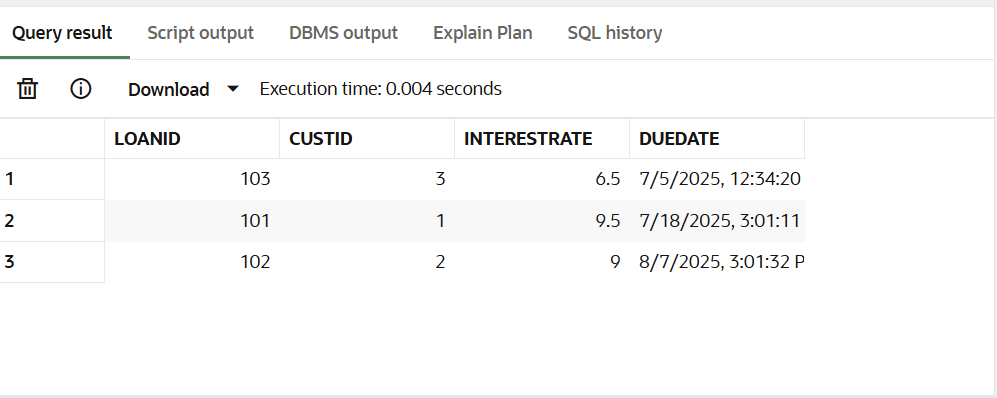
****

****

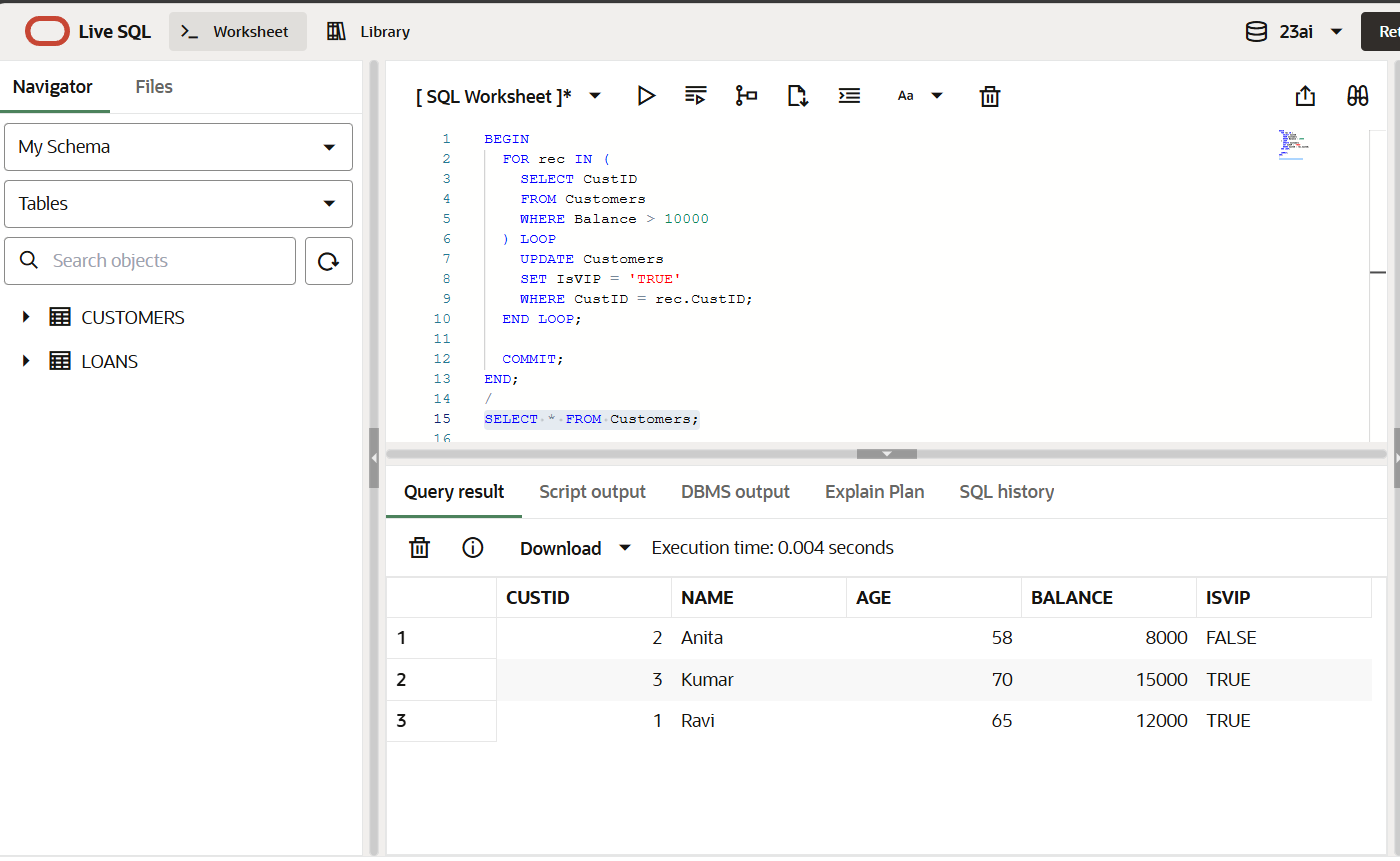
**Scenario 1 Implementation:**

****

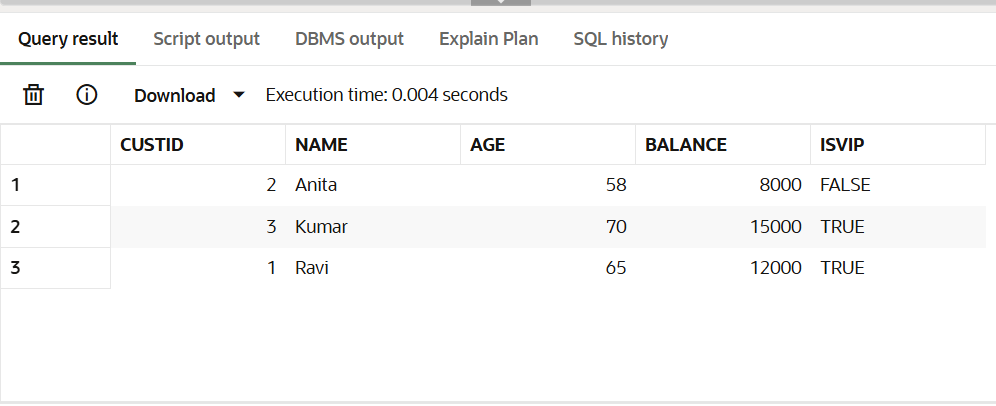
**Scenario 1 Output:**

****

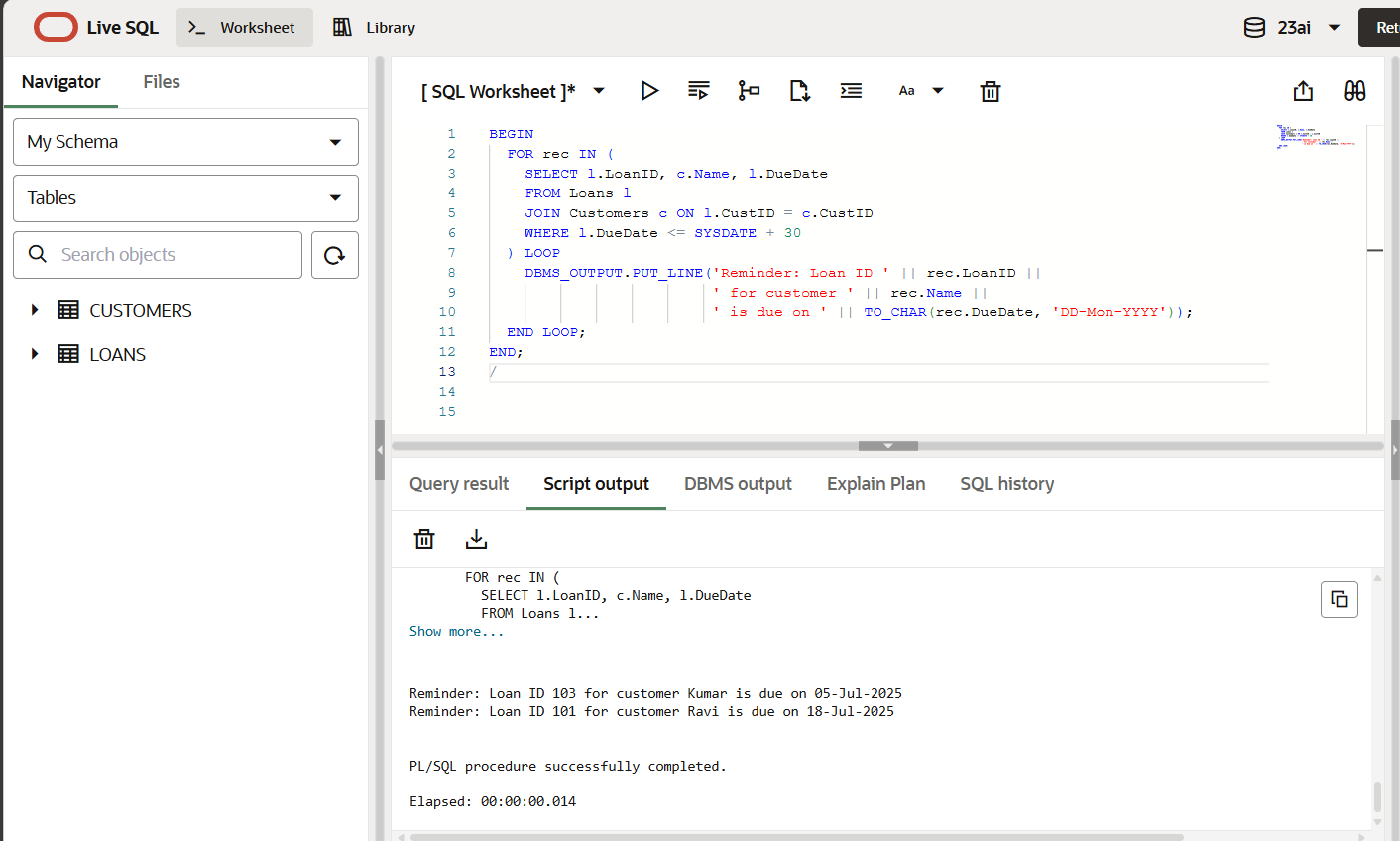
**Scenario 2 Implementation:**

****

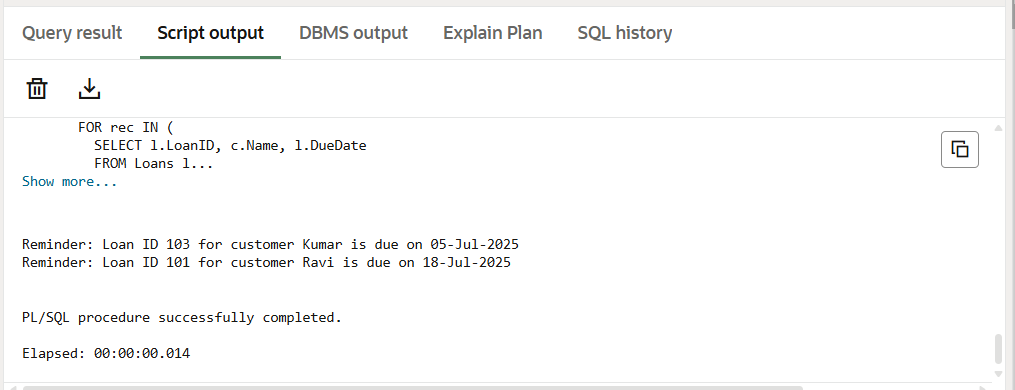
**Scenario 2 Output:**

****

**Scenario 3 Implementation:**

****

**Scenario 3 Output:**

****

**PROGRAM :**

**Creation of tables and sample data :**

CREATE TABLE Customers (

CustID NUMBER PRIMARY KEY,

Name VARCHAR2(50),

Age NUMBER,

Balance NUMBER,

IsVIP VARCHAR2(5)

);

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustID NUMBER,

InterestRate NUMBER,

DueDate DATE,

FOREIGN KEY (CustID) REFERENCES Customers(CustID)

);

INSERT INTO Customers VALUES (1, 'Ravi', 65, 12000, 'FALSE');

INSERT INTO Customers VALUES (2, 'Anita', 58, 8000, 'FALSE');

INSERT INTO Customers VALUES (3, 'Kumar', 70, 15000, 'FALSE');

INSERT INTO Loans VALUES (101, 1, 10.5, SYSDATE + 20);

INSERT INTO Loans VALUES (102, 2, 9.0, SYSDATE + 40);

INSERT INTO Loans VALUES (103, 3, 8.5, SYSDATE + 10);

COMMIT;

/

**Scenario 1 :**

BEGIN

FOR rec IN (

SELECT c.CustID, l.LoanID

FROM Customers c

JOIN Loans l ON c.CustID = l.CustID

WHERE c.Age > 60

) LOOP

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE LoanID = rec.LoanID;

END LOOP;

COMMIT;

END;

/

**Scenario 2 :**

BEGIN

FOR rec IN (

SELECT CustID

FROM Customers

WHERE Balance > 10000

) LOOP

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustID = rec.CustID;

END LOOP;

COMMIT;

END;

/

**Scenario 3 :**

BEGIN

FOR rec IN (

SELECT l.LoanID, c.Name, l.DueDate

FROM Loans l

JOIN Customers c ON l.CustID = c.CustID

WHERE l.DueDate <= SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE(

'Reminder: Loan ID ' || rec.LoanID ||

' for customer ' || rec.Name ||

' is due on ' || TO\_CHAR(rec.DueDate, 'DD-Mon-YYYY')

);

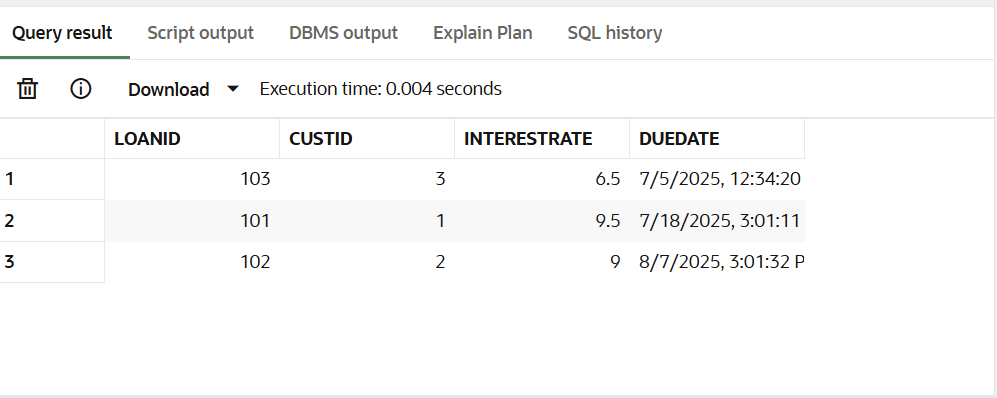
END LOOP;

END;

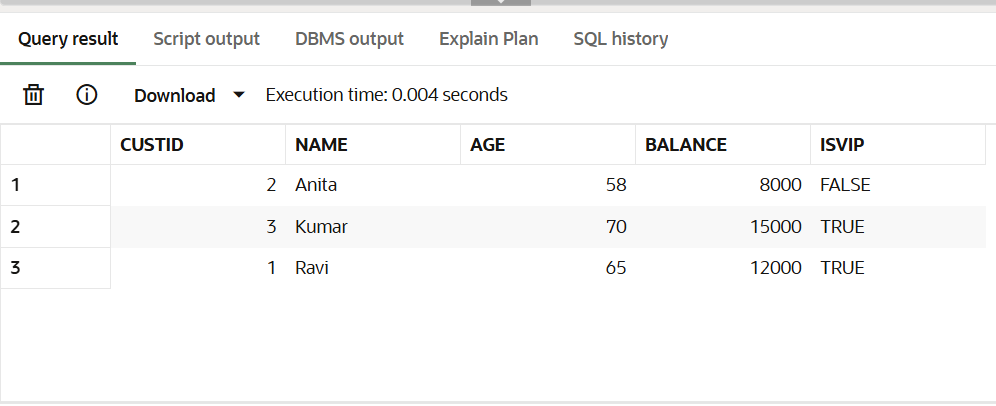
/

**OUTPUT :**

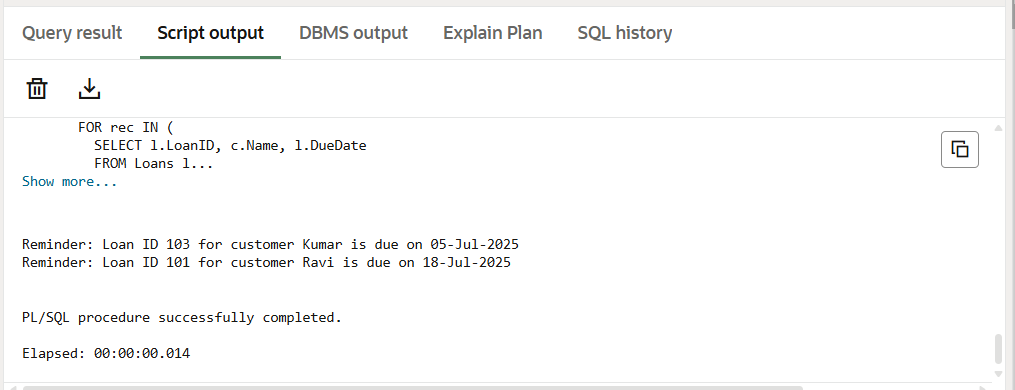
**Scenario 1:**

****

**Scenario 2 :**

****

**Scenario 3 :**

****