**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;

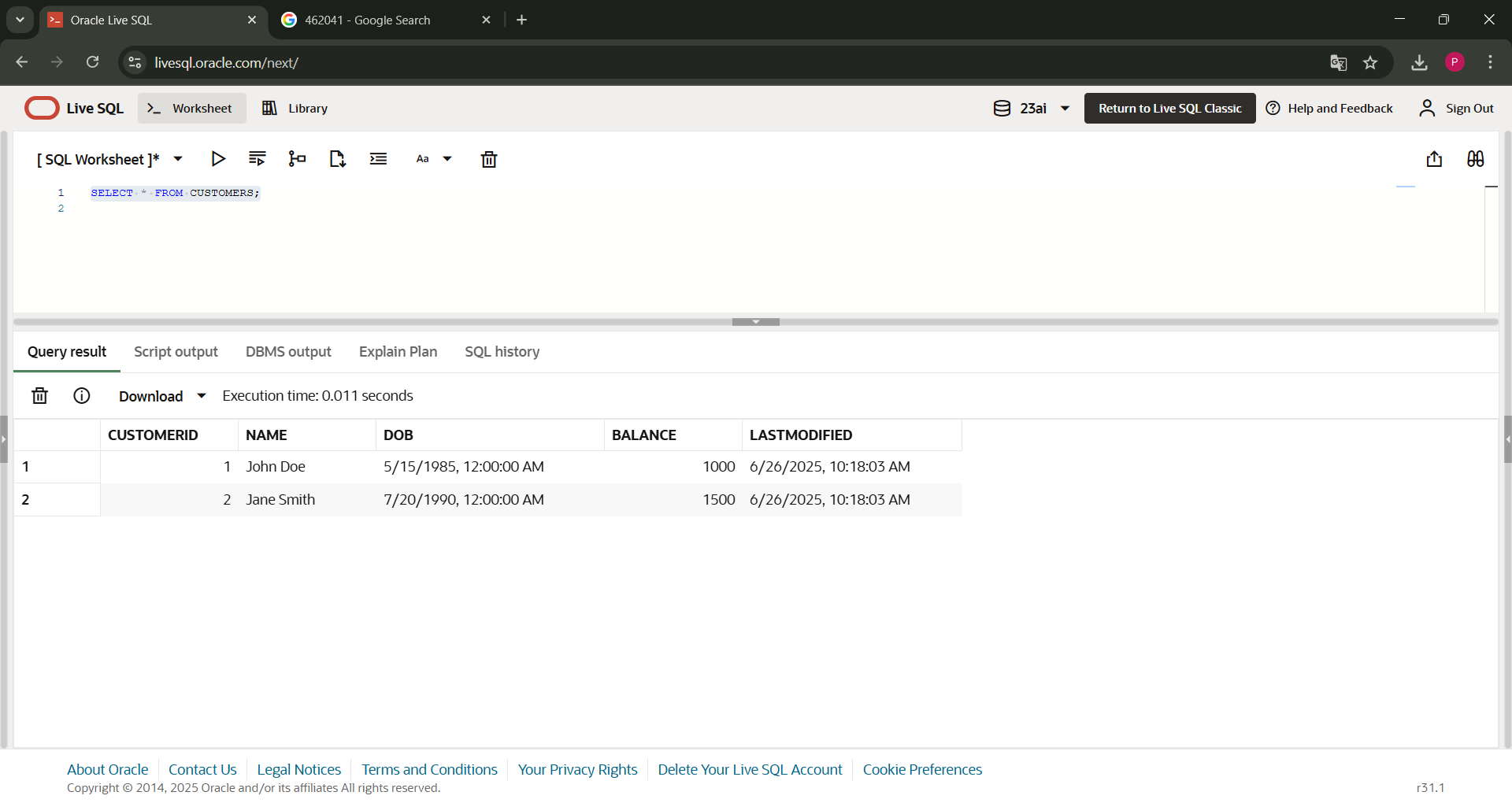
  COMMIT;

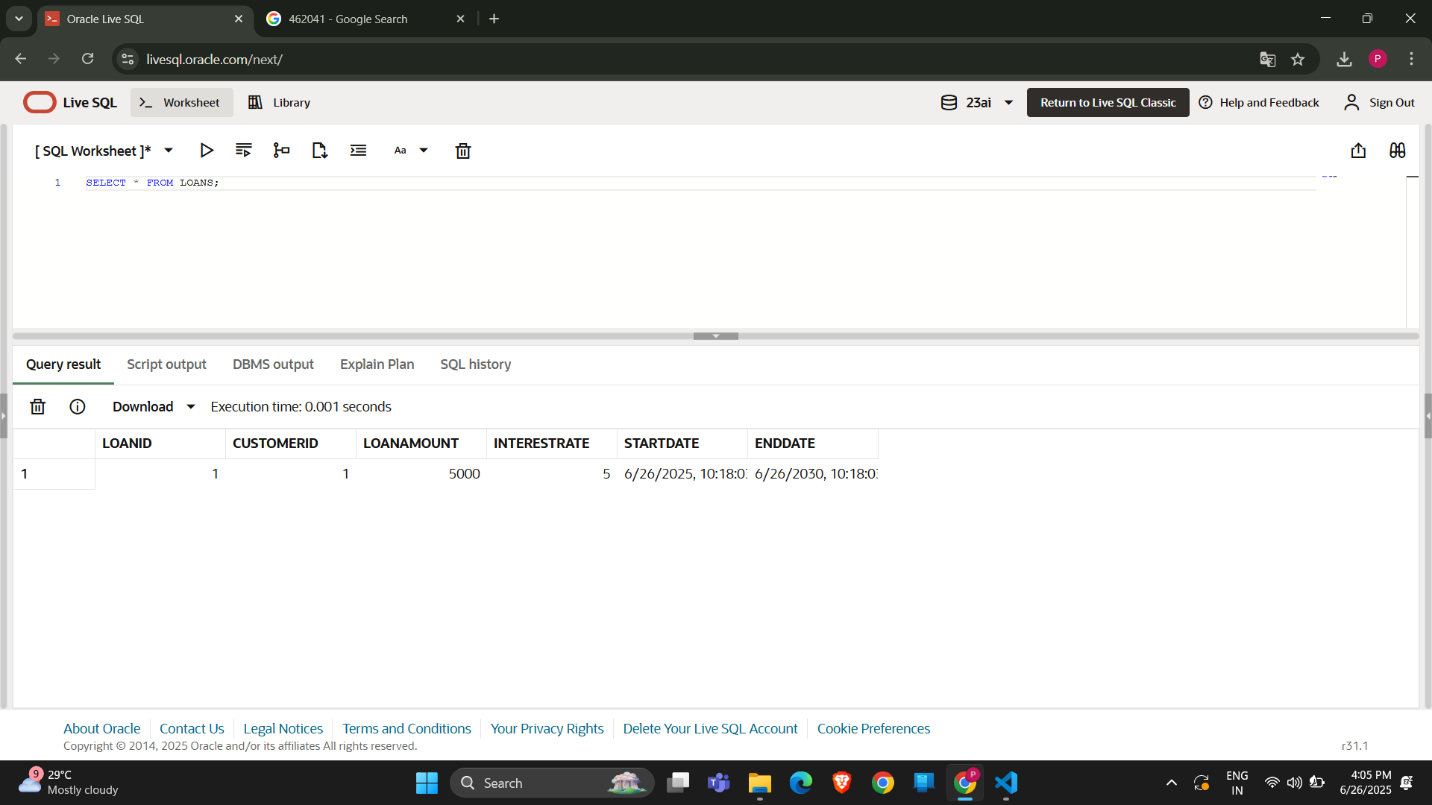
  DBMS\_OUTPUT.PUT\_LINE('Total loans updated: ' || v\_updated\_count);

END;

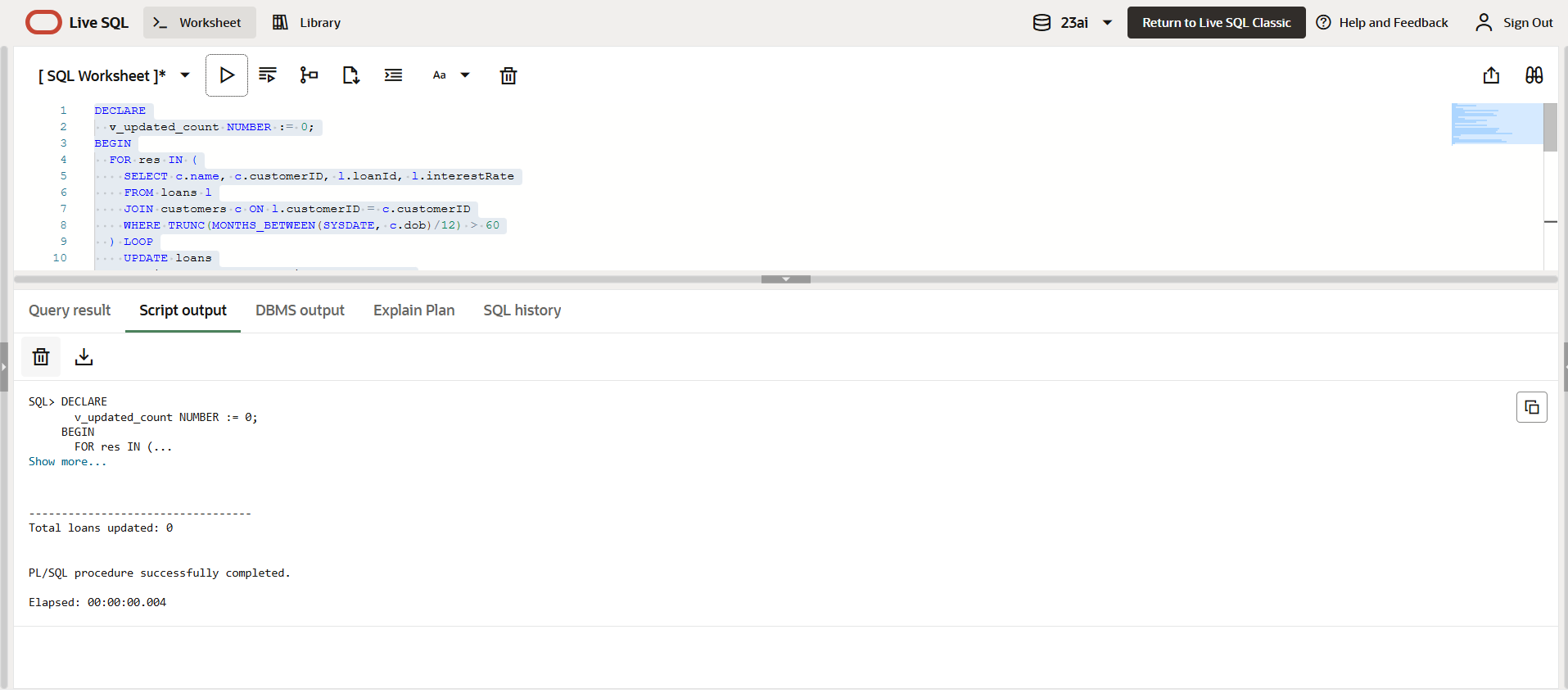
/

**OUTPUT:   
Initial Customers Table:**



**Initial Loans Table:**   


**Query 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:**

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;

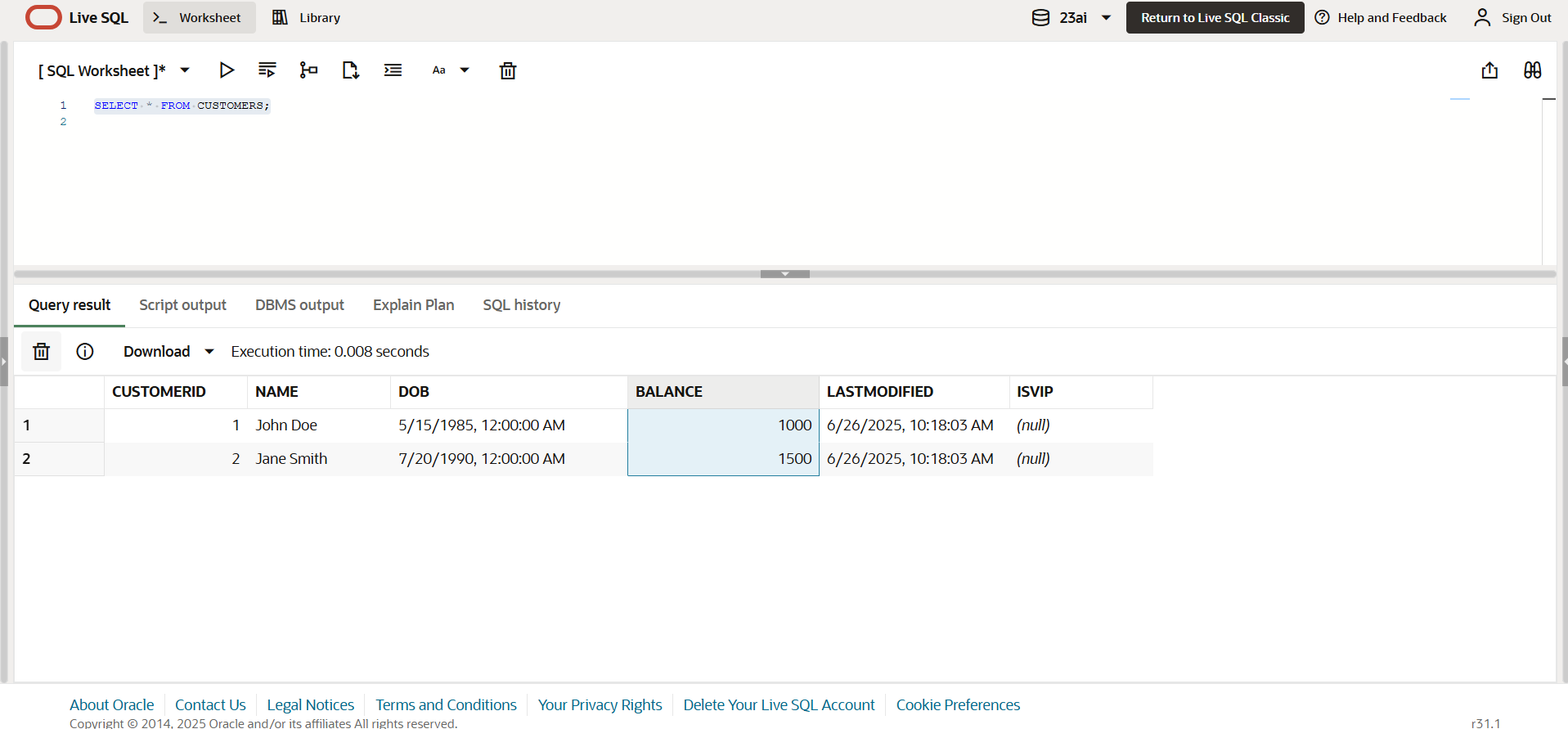
    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE('Total ' || vip\_count || ' VIPs found.');

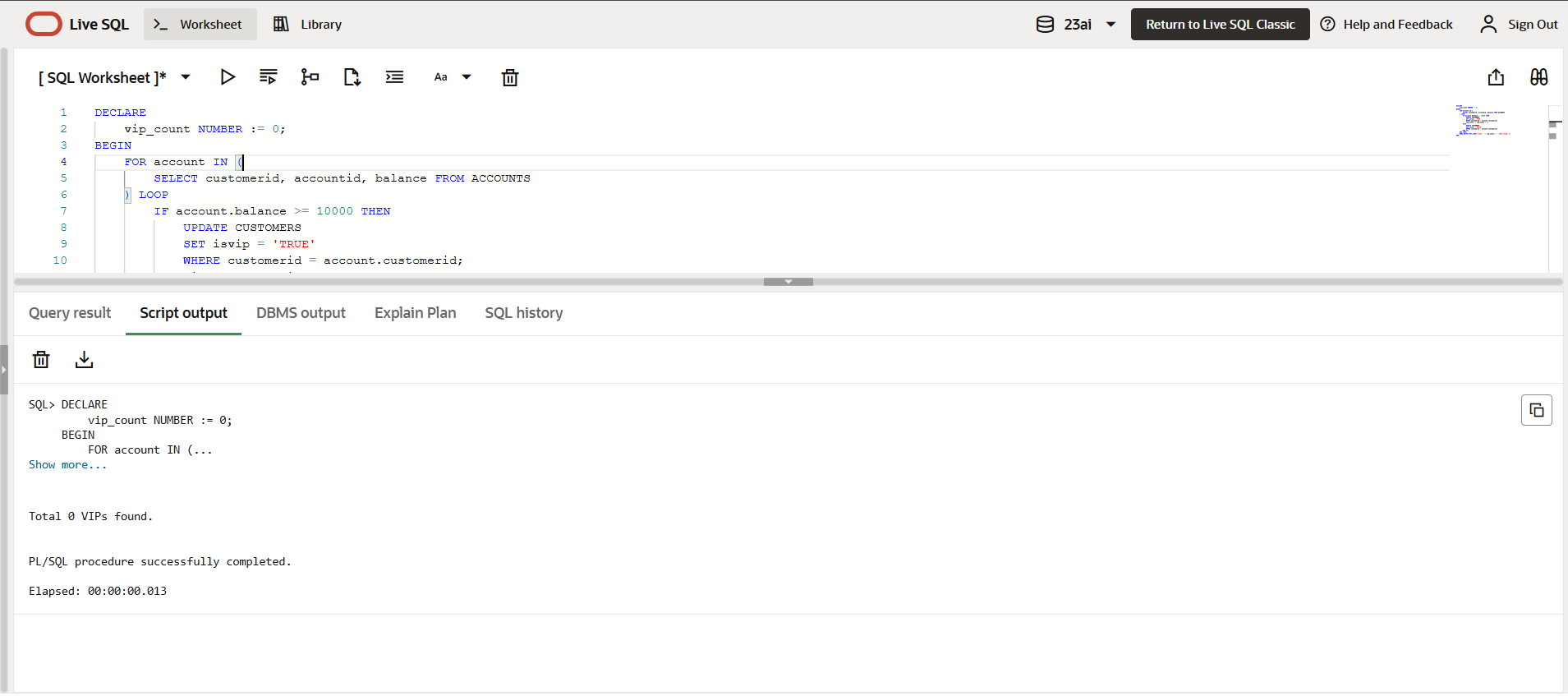
END;

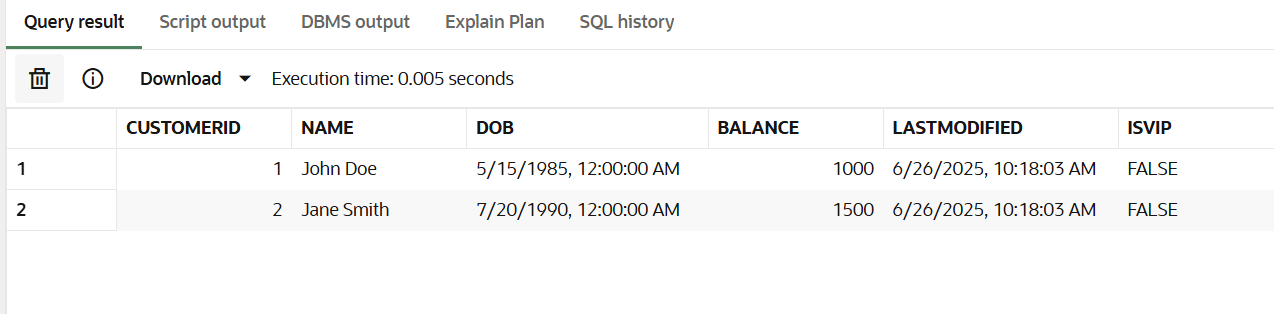
/

**Output:   
Modified Customers Table:   
**

****

**PL/SQL Query Result:**

****

****

**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

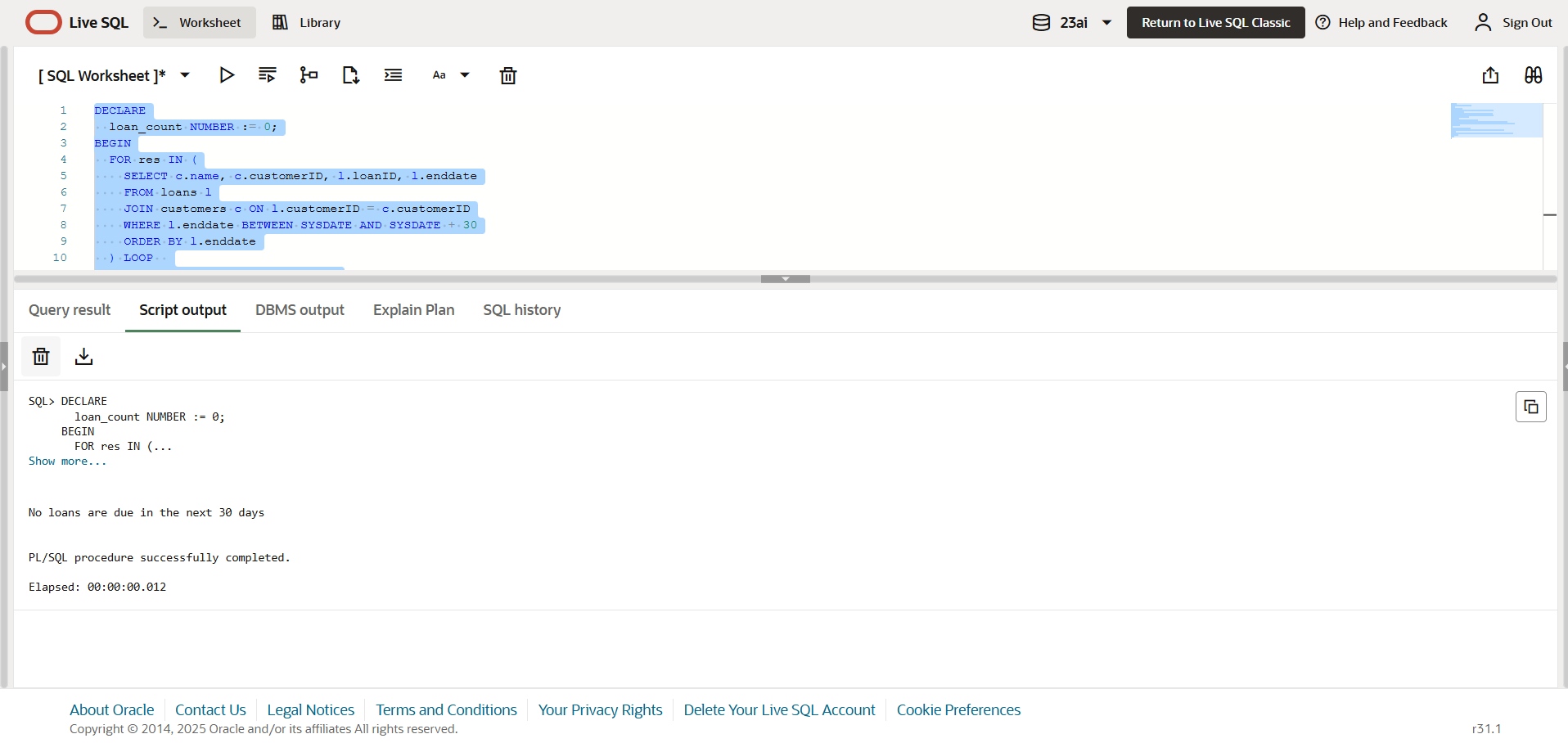
    DBMS\_OUTPUT.PUT\_LINE('Total loans due in next 30 days: ' || loan\_count);

  END IF;

END;

/

**OUTPUT:**

****