**Exercise 1: Control Structures**

**Scenario 1:**

The bank wants to apply a discount to loan interest rates for customers above 60 years old.

**PL/SQL Query:**

BEGIN

  EXECUTE IMMEDIATE 'DROP TABLE Loans';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

BEGIN

  EXECUTE IMMEDIATE 'DROP TABLE Customers';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

CREATE TABLE Customers (

    customer\_id NUMBER PRIMARY KEY,

    name VARCHAR2(50),

    age NUMBER,

    balance NUMBER,

    isVIP VARCHAR2(5),

    loan\_interest\_rate NUMBER

);

CREATE TABLE Loans (

    loan\_id NUMBER PRIMARY KEY,

    customer\_id NUMBER,

    due\_date DATE,

    FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

INSERT INTO Customers VALUES (1, 'Ravi', 65, 15000, 'FALSE', 9.5);

INSERT INTO Customers VALUES (2, 'Meena', 45, 7000, 'FALSE', 10.0);

INSERT INTO Customers VALUES (3, 'Arun', 70, 12000, 'FALSE', 11.0);

INSERT INTO Loans VALUES (101, 1, SYSDATE + 10);

INSERT INTO Loans VALUES (102, 2, SYSDATE + 35);

INSERT INTO Loans VALUES (103, 3, SYSDATE + 5);

COMMIT;

BEGIN

  FOR rec IN (SELECT customer\_id, age FROM Customers) LOOP

    IF rec.age > 60 THEN

      UPDATE Customers

      SET loan\_interest\_rate = loan\_interest\_rate - 1

      WHERE customer\_id = rec.customer\_id;

    END IF;

  END LOOP;

  COMMIT;

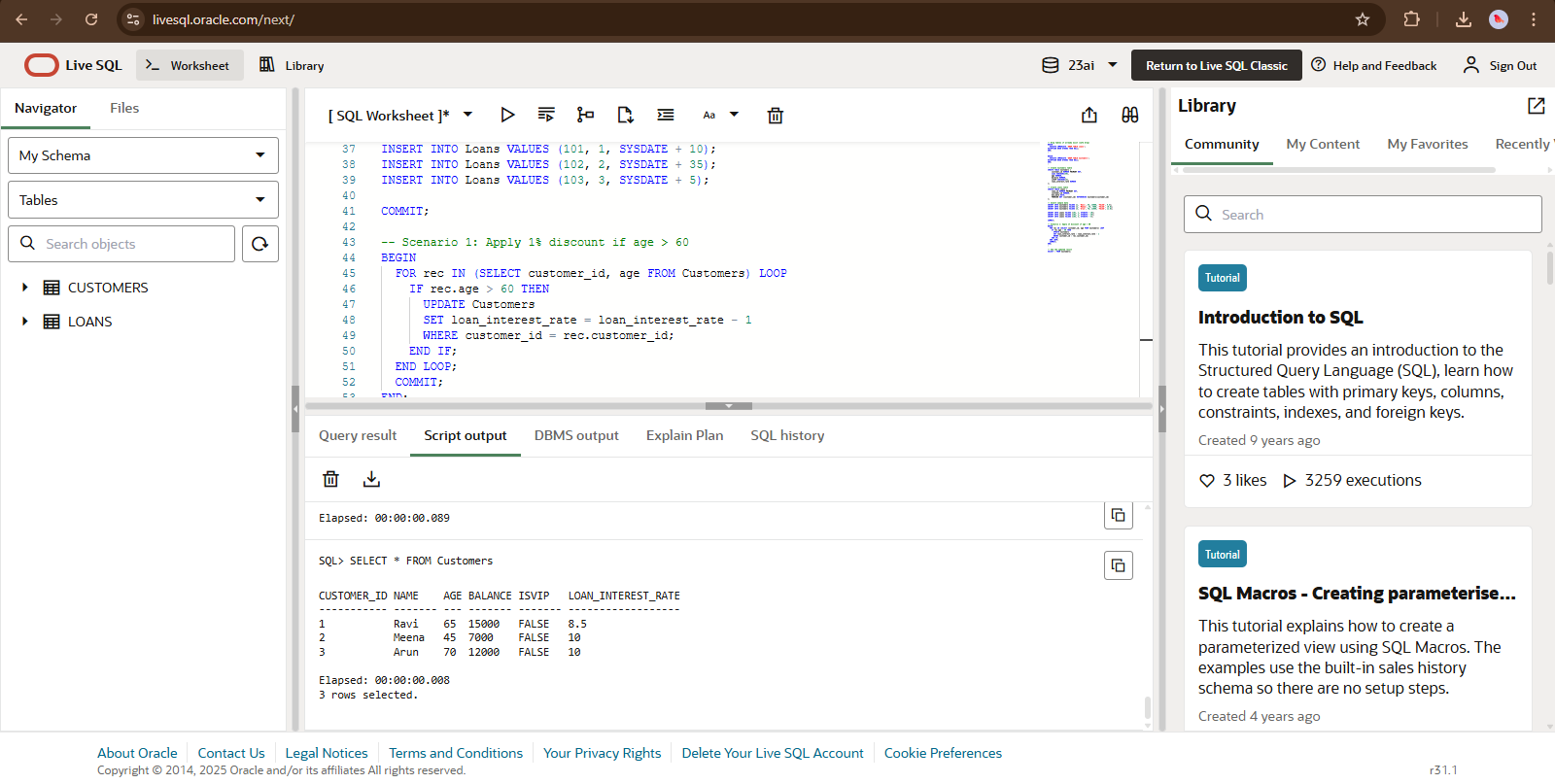
END;

/

-- See the updated result

SELECT \* FROM Customers;

**Script Output:**

****

**Scenario 2:**

A customer can be promoted to VIP status based on their balance.

**PL/SQL Query:**

-- Optional: Drop and re-create Customers table with fresh data

BEGIN

  EXECUTE IMMEDIATE 'DROP TABLE Customers';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

-- Create Customers table again

CREATE TABLE Customers (

    customer\_id NUMBER PRIMARY KEY,

    name VARCHAR2(50),

    age NUMBER,

    balance NUMBER,

    isVIP VARCHAR2(5),

    loan\_interest\_rate NUMBER

);

-- Insert sample data

INSERT INTO Customers VALUES (1, 'Ravi', 65, 15000, 'FALSE', 8.5);

INSERT INTO Customers VALUES (2, 'Meena', 45, 7000, 'FALSE', 10.0);

INSERT INTO Customers VALUES (3, 'Arun', 70, 12000, 'FALSE', 10.0);

COMMIT;

/

-- Scenario 2: Promote customer to VIP if balance > 10000

BEGIN

  FOR rec IN (SELECT customer\_id, balance FROM Customers) LOOP

    IF rec.balance > 10000 THEN

      UPDATE Customers

      SET isVIP = 'TRUE'

      WHERE customer\_id = rec.customer\_id;

    END IF;

  END LOOP;

  COMMIT;

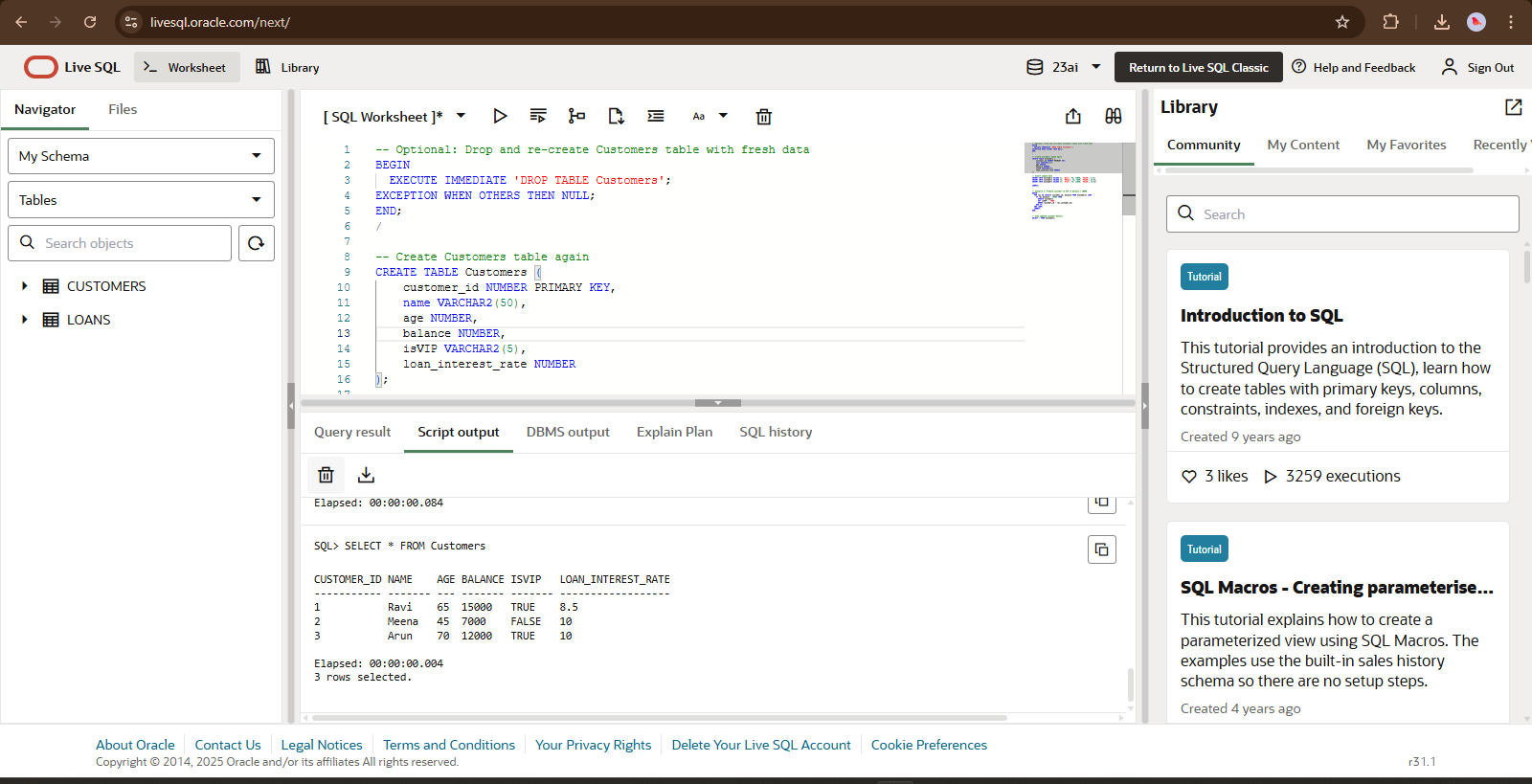
END;

/

-- View updated customer details

SELECT \* FROM Customers;

**Script Output:**



**Scenario 3:**

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

**PL/SQL Query:**

-- Drop and re-create Loans and Customers tables for clean setup

BEGIN

  EXECUTE IMMEDIATE 'DROP TABLE Loans';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

BEGIN

  EXECUTE IMMEDIATE 'DROP TABLE Customers';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

-- Create Customers table

CREATE TABLE Customers (

    customer\_id NUMBER PRIMARY KEY,

    name VARCHAR2(50),

    age NUMBER,

    balance NUMBER,

    isVIP VARCHAR2(5),

    loan\_interest\_rate NUMBER

);

-- Create Loans table

CREATE TABLE Loans (

    loan\_id NUMBER PRIMARY KEY,

    customer\_id NUMBER,

    due\_date DATE,

    FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

-- Insert sample customers

INSERT INTO Customers VALUES (1, 'Ravi', 65, 15000, 'TRUE', 8.5);

INSERT INTO Customers VALUES (2, 'Meena', 45, 7000, 'FALSE', 10.0);

INSERT INTO Customers VALUES (3, 'Arun', 70, 12000, 'TRUE', 10.0);

-- Insert sample loans (some due within 30 days, some not)

INSERT INTO Loans VALUES (101, 1, SYSDATE + 10); -- within 30 days

INSERT INTO Loans VALUES (102, 2, SYSDATE + 45); -- beyond 30 days

INSERT INTO Loans VALUES (103, 3, SYSDATE + 5);  -- within 30 days

COMMIT;

/

-- Scenario 3: Print reminders for loans due within next 30 days

BEGIN

  FOR loan\_rec IN (

    SELECT l.loan\_id, l.customer\_id, l.due\_date, c.name

    FROM Loans l

    JOIN Customers c ON l.customer\_id = c.customer\_id

    WHERE l.due\_date <= SYSDATE + 30

  ) LOOP

    DBMS\_OUTPUT.PUT\_LINE('Reminder: Customer ' || loan\_rec.name ||

                         ' (Loan ID: ' || loan\_rec.loan\_id ||

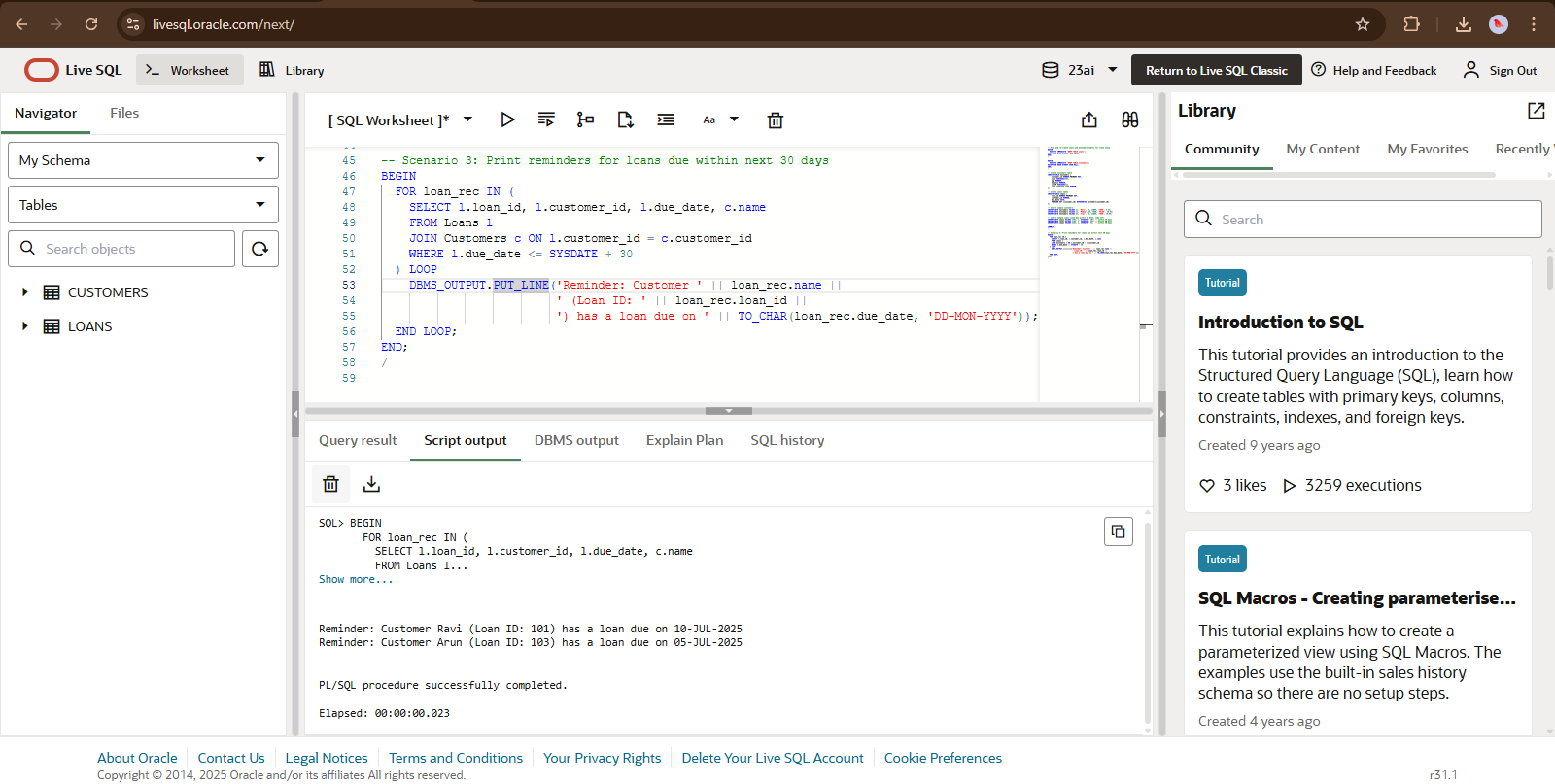
                         ') has a loan due on ' || TO\_CHAR(loan\_rec.due\_date, 'DD-MON-YYYY'));

  END LOOP;

END;

/

**Script Output:**

****