**Exercise 3: Stored Procedures**

**Scenario 1:**

The bank needs to process monthly interest for all savings accounts.

**PL/SQL Query:**

-- Drop Accounts table if exists

BEGIN

  EXECUTE IMMEDIATE 'DROP TABLE Accounts';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

-- Create Accounts table

CREATE TABLE Accounts (

    account\_id NUMBER PRIMARY KEY,

    account\_type VARCHAR2(20),

    balance NUMBER

);

-- Insert sample data

INSERT INTO Accounts VALUES (101, 'Savings', 5000);

INSERT INTO Accounts VALUES (102, 'Current', 3000);

INSERT INTO Accounts VALUES (103, 'Savings', 8000);

COMMIT;

/

-- Procedure to apply 1% interest to savings accounts

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

  UPDATE Accounts

  SET balance = balance + (balance \* 0.01)

  WHERE account\_type = 'Savings';

  COMMIT;

END;

/

-- Call the procedure

BEGIN

  ProcessMonthlyInterest;

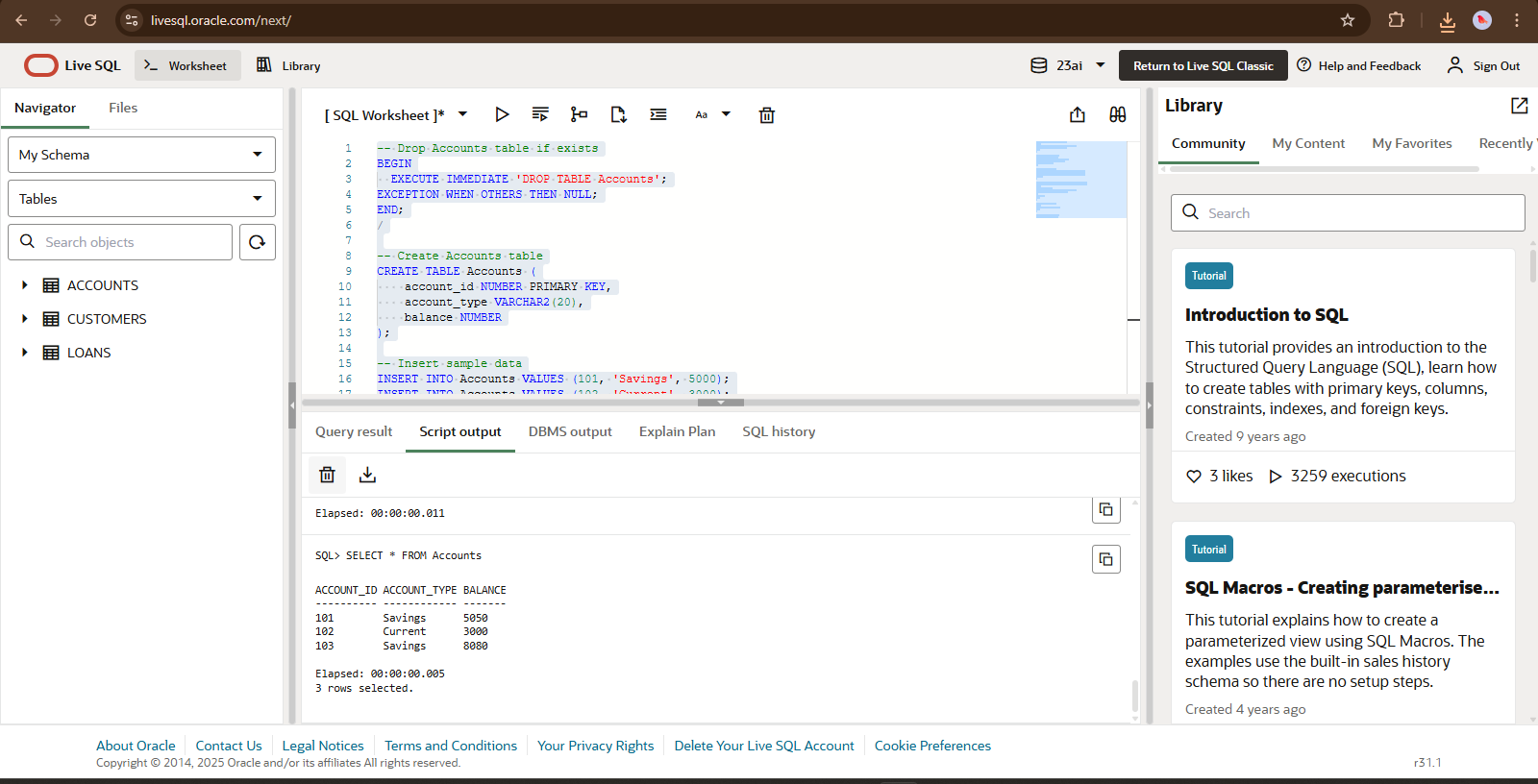
END;

/

-- View updated balances

SELECT \* FROM Accounts;

**Script Output:**

****

**Scenario 2:**

The bank wants to implement a bonus scheme for employees based on their performance**.**

**PL/SQL Query:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

dept\_id IN NUMBER,

bonus\_percent IN NUMBER

) IS

BEGIN

FOR rec IN (

SELECT employee\_id, salary

FROM employees

WHERE department\_id = dept\_id

) LOOP

UPDATE employees

SET salary = salary + (rec.salary \* bonus\_percent / 100)

WHERE employee\_id = rec.employee\_id;

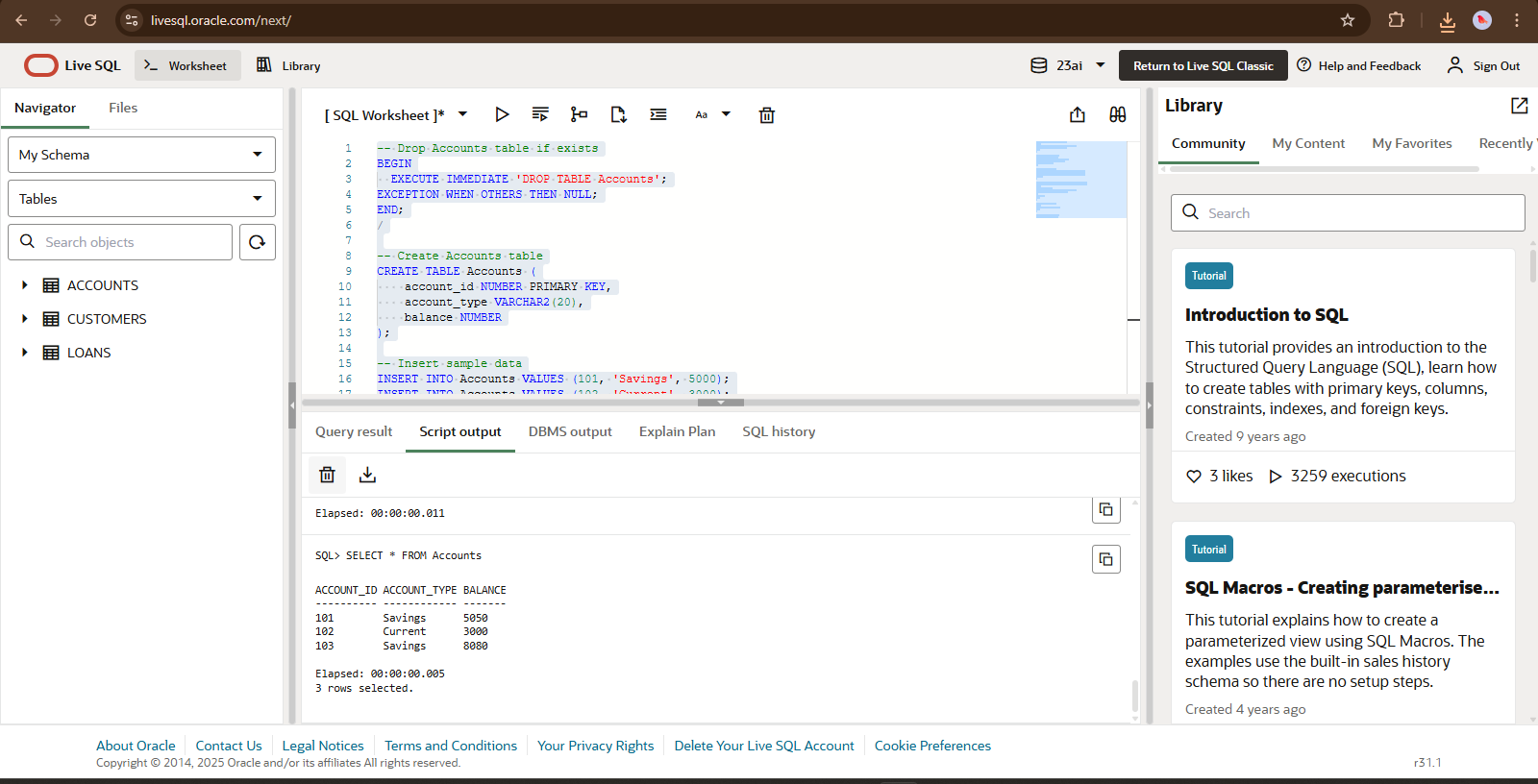
DBMS\_OUTPUT.PUT\_LINE('Bonus applied to Employee ID: ' || rec.employee\_id);

END LOOP;

END;

/

**Script Output:**

****

**Scenario 3:**

Customers should be able to transfer funds between their accounts.

**PL/SQL Query:**

-- Drop Accounts table if needed

BEGIN

  EXECUTE IMMEDIATE 'DROP TABLE Accounts';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

-- Create fresh Accounts table

CREATE TABLE Accounts (

    account\_id NUMBER PRIMARY KEY,

    balance NUMBER

);

-- Insert sample accounts

INSERT INTO Accounts VALUES (201, 10000);

INSERT INTO Accounts VALUES (202, 5000);

COMMIT;

/

-- Procedure to transfer funds

CREATE OR REPLACE PROCEDURE TransferFunds(

  from\_acc IN NUMBER,

  to\_acc IN NUMBER,

  amount IN NUMBER

) AS

  from\_balance NUMBER;

BEGIN

  -- Check balance of source account

  SELECT balance INTO from\_balance FROM Accounts WHERE account\_id = from\_acc;

  IF from\_balance >= amount THEN

    UPDATE Accounts SET balance = balance - amount WHERE account\_id = from\_acc;

    UPDATE Accounts SET balance = balance + amount WHERE account\_id = to\_acc;

    COMMIT;

  ELSE

    DBMS\_OUTPUT.PUT\_LINE('Insufficient balance in account ' || from\_acc);

  END IF;

END;

/

-- Call the procedure to transfer ₹2000 from 201 to 202

BEGIN

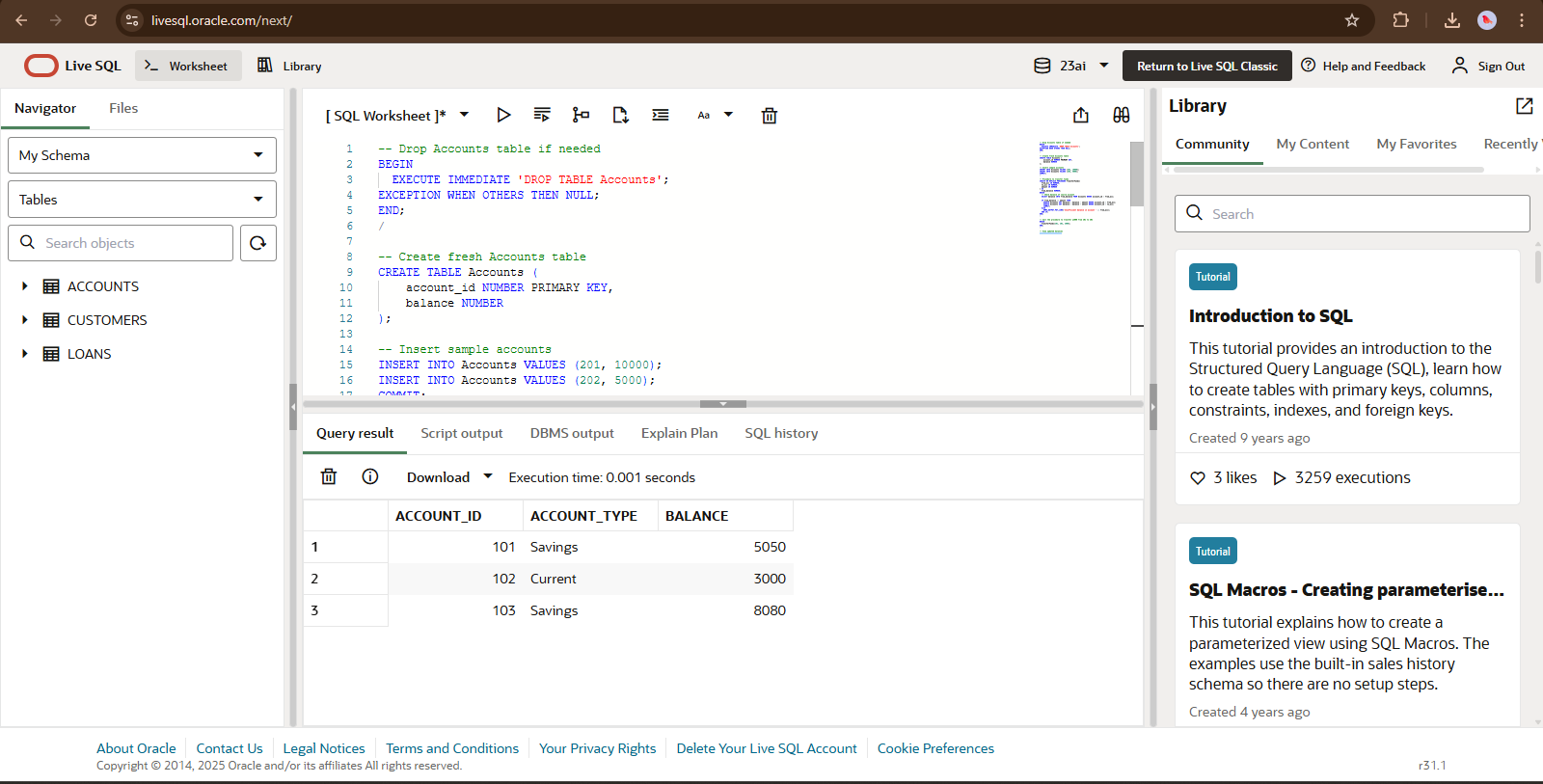
  TransferFunds(201, 202, 2000);

END;

/

-- View updated balances

SELECT \* FROM Accounts;

**Script Output:  
**