**WEEK – 2**

**HANDS ON**

**PL/SQL PROGRAMMING**

**Exercise 3: Stored Procedures**

Scenario – 1

CREATE TABLE SAVINGS\_ACCOUNT (

    ACCOUNT\_ID NUMBER PRIMARY KEY,

    CUSTOMER\_ID NUMBER,

    BALANCE NUMBER(12,2),

    ACCOUNT\_TYPE VARCHAR2(20)

);

INSERT INTO SAVINGS\_ACCOUNT VALUES (201, 1, 10000.00, 'SAVINGS');

INSERT INTO SAVINGS\_ACCOUNT VALUES (202, 2, 15000.00, 'SAVINGS');

INSERT INTO SAVINGS\_ACCOUNT VALUES (203, 3, 20000.00, 'CURRENT');

INSERT INTO SAVINGS\_ACCOUNT VALUES (204, 4, 100000.00, 'SAVINGS');

INSERT INTO SAVINGS\_ACCOUNT VALUES (205, 5, 35000.00, 'CURRENT');

INSERT INTO SAVINGS\_ACCOUNT VALUES (206, 6, 20000.00, 'SAVINGS');

COMMIT;

SELECT \* FROM SAVINGS\_ACCOUNT;

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

    UPDATE SAVINGS\_ACCOUNT

    SET BALANCE = BALANCE + (BALANCE \* 0.01)

    WHERE ACCOUNT\_TYPE = 'SAVINGS';

    COMMIT;

END;

/

BEGIN

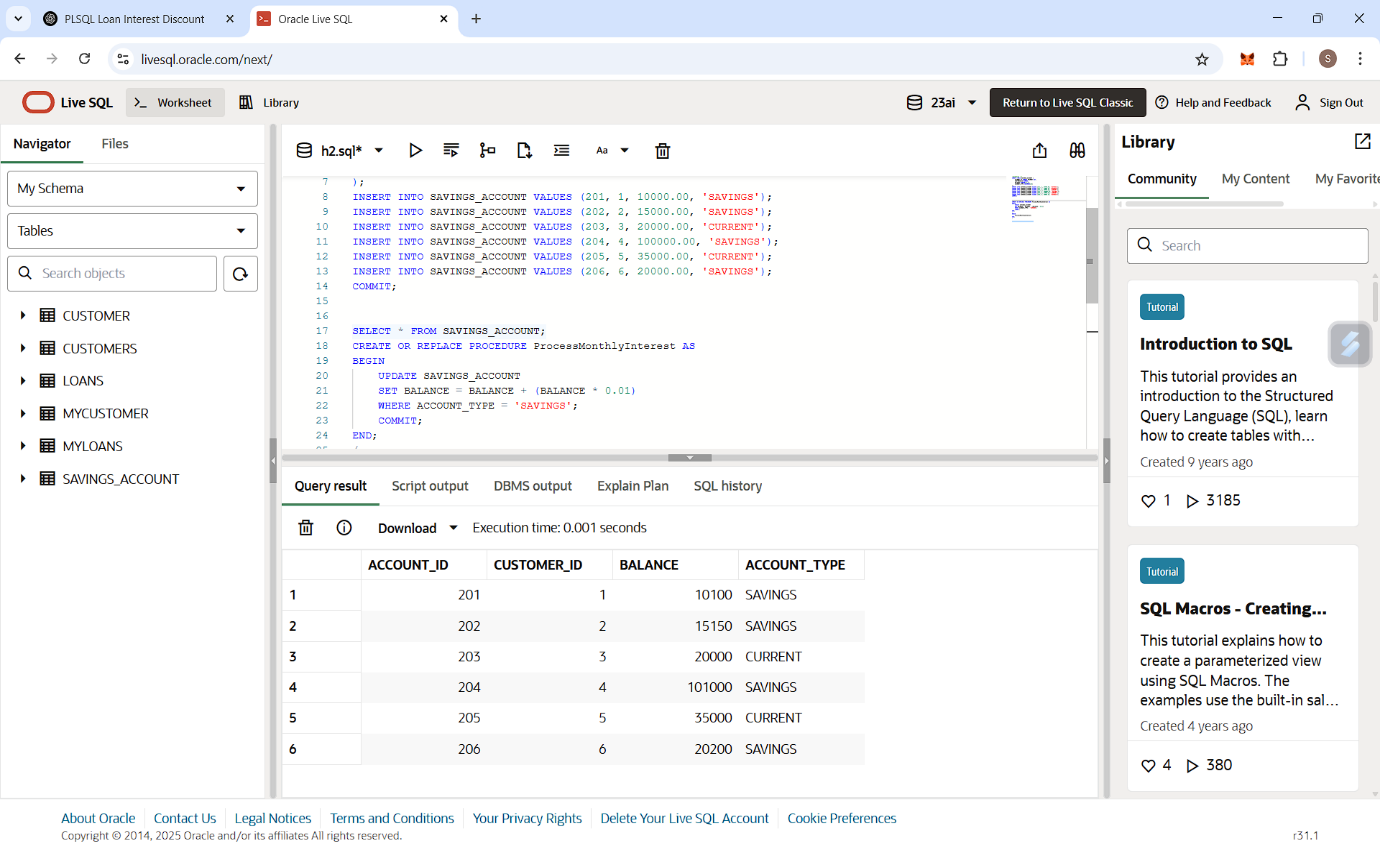
    ProcessMonthlyInterest;

END;

/

SELECT \* FROM SAVINGS\_ACCOUNT;

**OUTPUT**

****

SCENARIO – 2

CREATE TABLE EMPLOYEE (

    EMP\_ID NUMBER PRIMARY KEY,

    NAME VARCHAR2(50),

    DEPARTMENT VARCHAR2(50),

    SALARY NUMBER(10, 2)

);

INSERT INTO EMPLOYEE VALUES (1, 'Alice', 'Finance', 50000);

INSERT INTO EMPLOYEE VALUES (2, 'Bob', 'IT', 60000);

INSERT INTO EMPLOYEE VALUES (3, 'Charlie', 'Finance', 55000);

INSERT INTO EMPLOYEE VALUES (4, 'Diana', 'HR', 48000);

COMMIT;

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

    dept\_name IN VARCHAR2,

    bonus\_percent IN NUMBER

) AS

    rows\_updated NUMBER;

BEGIN

    UPDATE EMPLOYEE

    SET SALARY = SALARY + (SALARY \* bonus\_percent / 100)

    WHERE UPPER(DEPARTMENT) = UPPER(dept\_name);

rows\_updated := SQL%ROWCOUNT;

    DBMS\_OUTPUT.PUT\_LINE(rows\_updated || ' employee(s) updated.');

COMMIT;

END;

/

BEGIN

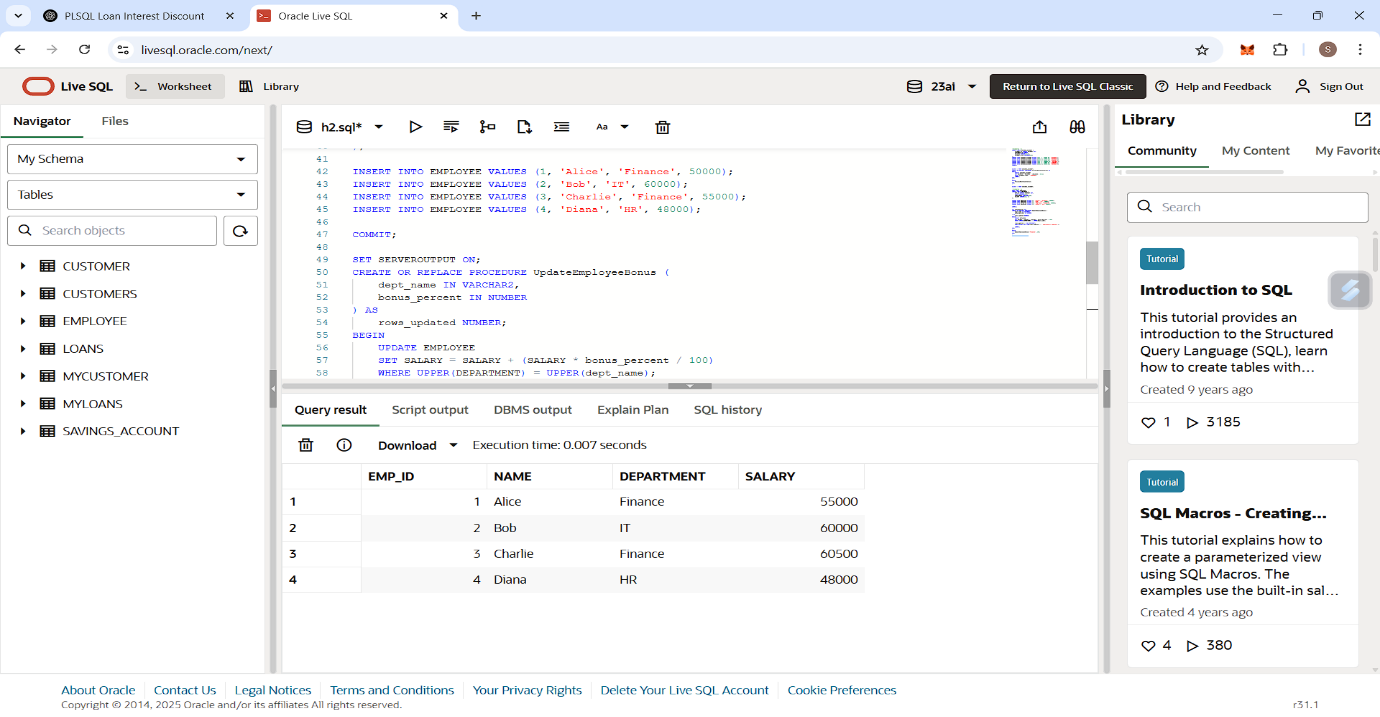
    UpdateEmployeeBonus('Finance', 10);

END;

/

SELECT \* FROM EMPLOYEE;

**OUTPUT**

****

SCENARIO – 3

CREATE TABLE BANK\_ACCOUNT (

    ACCOUNT\_ID NUMBER PRIMARY KEY,

    CUSTOMER\_ID NUMBER,

    BALANCE NUMBER(12, 2)

);

INSERT INTO BANK\_ACCOUNT VALUES (301, 1, 20000.00); -- Source

INSERT INTO BANK\_ACCOUNT VALUES (302, 2, 10000.00); -- Destination

COMMIT;

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE TransferFunds (

    from\_account\_id IN NUMBER,

    to\_account\_id IN NUMBER,

    amount IN NUMBER

) AS

    source\_balance NUMBER;

BEGIN

    SELECT BALANCE INTO source\_balance

    FROM BANK\_ACCOUNT

    WHERE ACCOUNT\_ID = from\_account\_id;

    IF source\_balance < amount THEN

        DBMS\_OUTPUT.PUT\_LINE('Insufficient balance. Transfer aborted.');

    ELSE

        UPDATE BANK\_ACCOUNT

        SET BALANCE = BALANCE - amount

        WHERE ACCOUNT\_ID = from\_account\_id;

        UPDATE BANK\_ACCOUNT

        SET BALANCE = BALANCE + amount

        WHERE ACCOUNT\_ID = to\_account\_id;

DBMS\_OUTPUT.PUT\_LINE('Transfer of ₹' || amount ||

                             ' from Account ' || from\_account\_id ||

                             ' to Account ' || to\_account\_id || ' completed.');

COMMIT;

    END IF;

EXCEPTION

    WHEN NO\_DATA\_FOUND THEN

        DBMS\_OUTPUT.PUT\_LINE('One or both account IDs are invalid.');

    WHEN OTHERS THEN

        DBMS\_OUTPUT.PUT\_LINE('Unexpected error: ' || SQLERRM);

END;

/

BEGIN

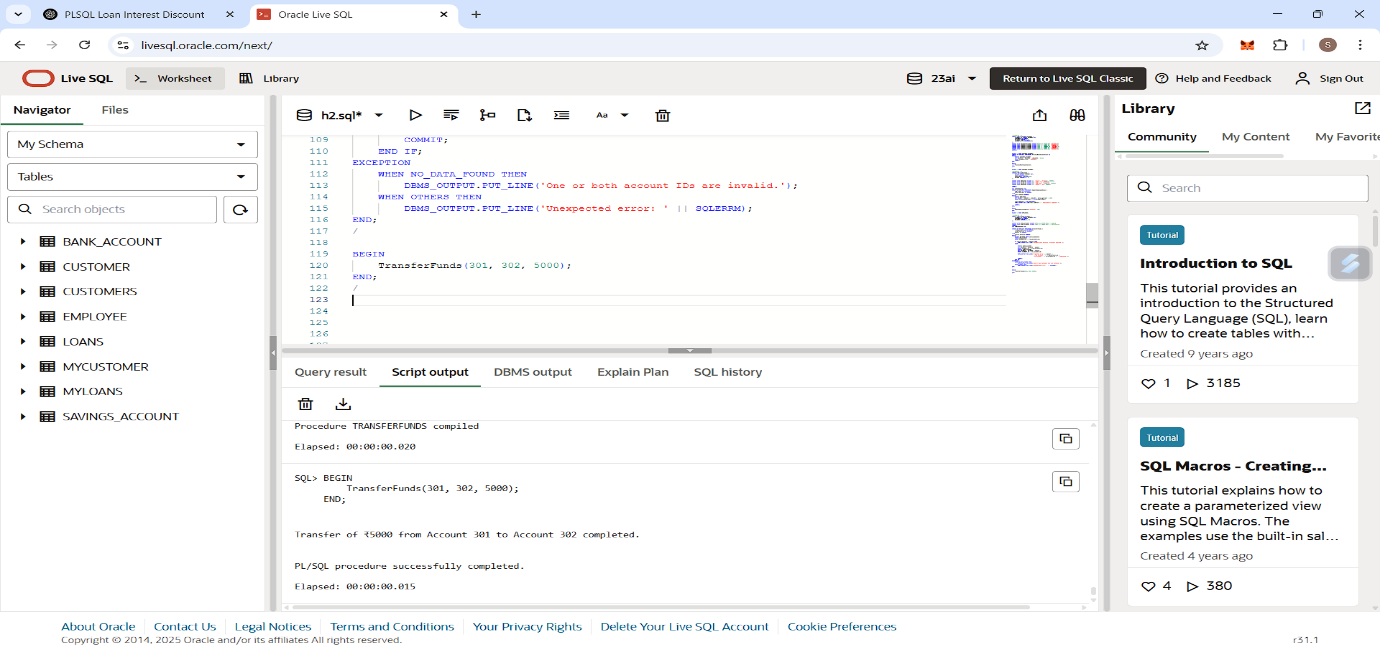
    TransferFunds(301, 302, 5000);

END;

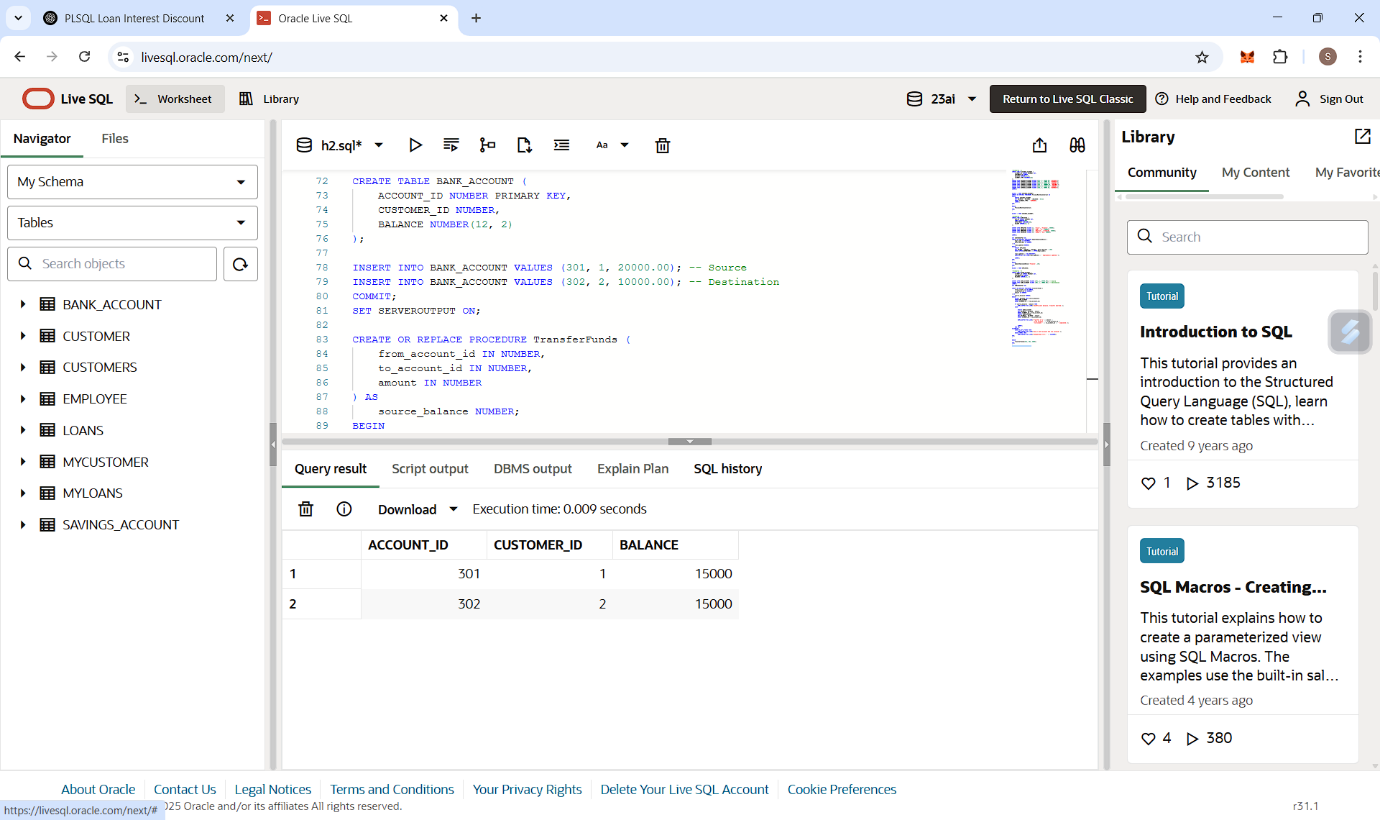
/

SELECT \* FROM BANK\_ACCOUNT;

**OUTPUT**

****

**BALANCE OF BOTH ACCOUNTS AFTER TRANSACTION IS DONE**

****