**WEEK – 2**

**PLSQL\_Exercises**

**Exercise 3: Stored Procedures**

**CODE:**

CREATE TABLE accounts (

    account\_id NUMBER PRIMARY KEY,

    customer\_name VARCHAR2(100),

    balance NUMBER,

    account\_type VARCHAR2(20) -- e.g., 'savings' or 'current'

);

INSERT INTO accounts VALUES (1, 'Pooja', 10000, 'savings');

INSERT INTO accounts VALUES (2, 'Raj', 5000, 'current');

INSERT INTO accounts VALUES (3, 'Anu', 20000, 'savings');

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

    UPDATE accounts

    SET balance = balance + (balance \* 0.01)

    WHERE account\_type = 'savings';

END;

EXEC ProcessMonthlyInterest;

-- Check results

SELECT \* FROM accounts;

BEGIN

   EXECUTE IMMEDIATE 'DROP TABLE employees';

EXCEPTION

   WHEN OTHERS THEN

      IF SQLCODE != -942 THEN

         RAISE;

      END IF;

END;

CREATE TABLE employees (

    emp\_id NUMBER PRIMARY KEY,

    name VARCHAR2(100),

    department VARCHAR2(50),

    salary NUMBER

);

INSERT INTO employees VALUES (1, 'Arjun', 'IT', 50000);

INSERT INTO employees VALUES (2, 'Meera', 'HR', 45000);

INSERT INTO employees VALUES (3, 'Sara', 'IT', 60000);

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

    dept IN VARCHAR2,

    bonus\_percent IN NUMBER

) IS

BEGIN

    UPDATE employees

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

    WHERE department = dept;

END;

/

EXEC UpdateEmployeeBonus('IT', 10);

-- Check updated salaries

SELECT \* FROM employees;

CREATE OR REPLACE PROCEDURE TransferFunds (

    from\_account IN NUMBER,

    to\_account IN NUMBER,

    amount IN NUMBER

) IS

    insufficient\_balance EXCEPTION;

    from\_balance NUMBER;

BEGIN

    -- Check balance of source

    SELECT balance INTO from\_balance FROM accounts WHERE account\_id = from\_account;

    IF from\_balance < amount THEN

        RAISE insufficient\_balance;

    END IF;

    -- Deduct from source

    UPDATE accounts SET balance = balance - amount WHERE account\_id = from\_account;

    -- Add to destination

    UPDATE accounts SET balance = balance + amount WHERE account\_id = to\_account;

EXCEPTION

    WHEN insufficient\_balance THEN

        DBMS\_OUTPUT.PUT\_LINE('❌ Insufficient balance in source account.');

    WHEN OTHERS THEN

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

END;

/

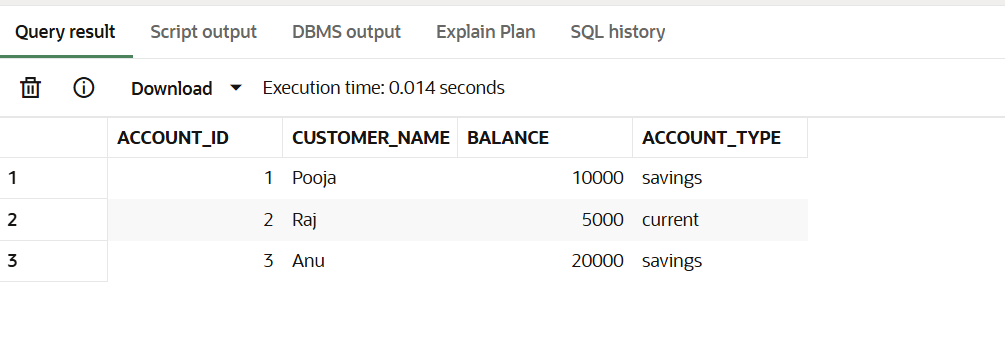
EXEC TransferFunds(1, 2, 500);

-- Check updated balances

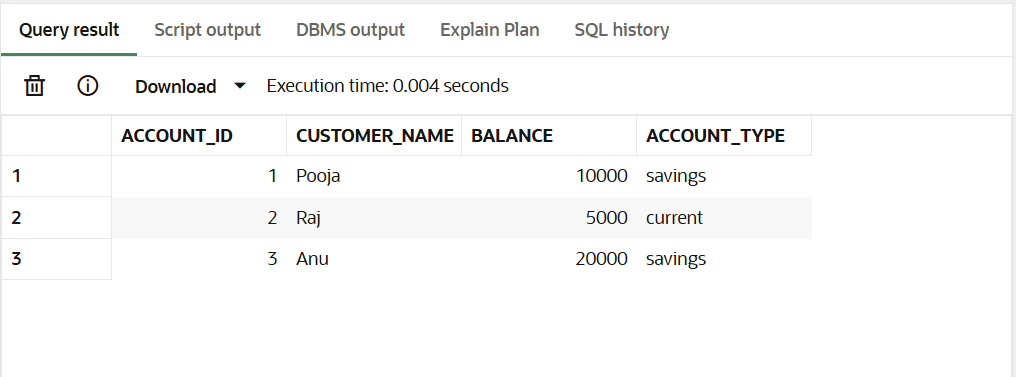
SELECT \* FROM accounts;

**OUTPUT:**

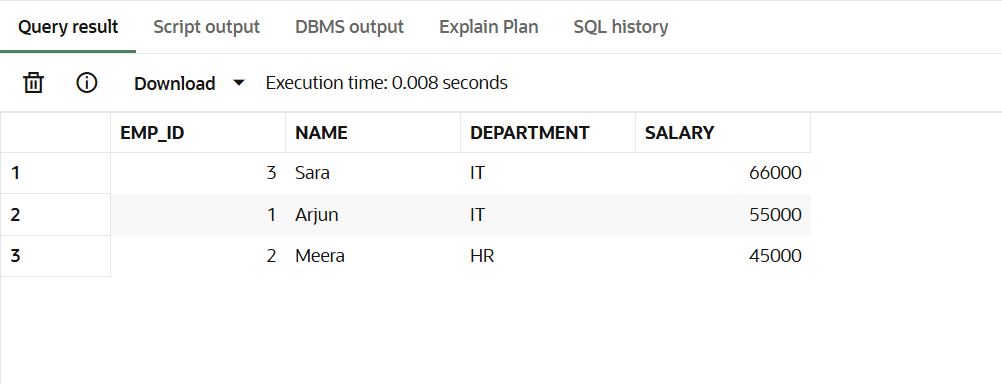
**Table:**



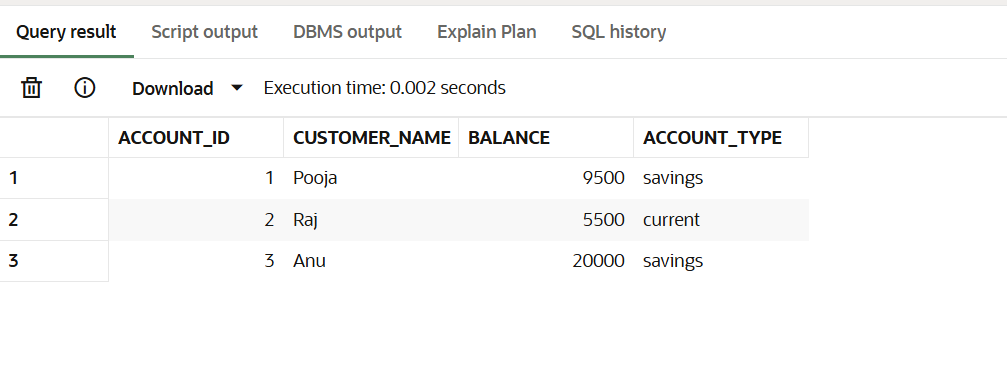
**Scenario – 1:**



**Scenario – 2:**



**Scenario – 3:**

****