**STORED PROCEDURE**

**SCENARIO 1:**

CREATE TABLE Accounts (

    AccountID NUMBER PRIMARY KEY,

    Customer\_ID NUMBER,

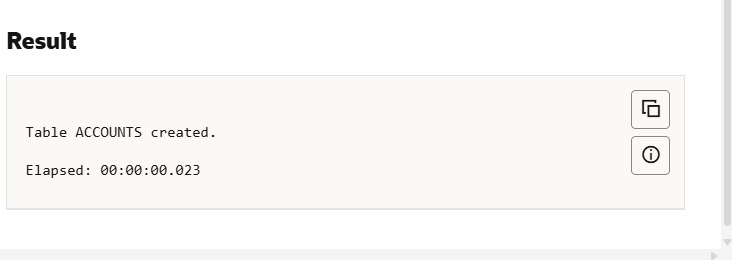
    AccountType VARCHAR2(20),

    Balance NUMBER,

    LastModified DATE

);

**OUTPUT:**



SELECT object\_name, object\_type

FROM user\_objects

WHERE object\_name = 'ACCOUNTS';

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

    UPDATE Accounts

    SET Balance = Balance + (Balance \* 0.01)

    WHERE UPPER(AccountType) = 'SAVINGS';

DBMS\_OUTPUT.PUT\_LINE('Monthly interest applied to savings accounts.');

END;

/

BEGIN

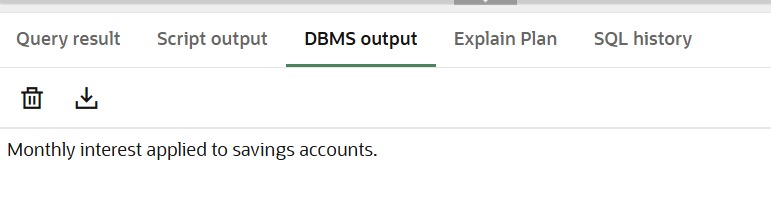
    ProcessMonthlyInterest;

END;

/

**OUTPUT:**





INSERT INTO Accounts (AccountID, Customer\_ID, AccountType, Balance, LastModified)

VALUES (101, 1, 'SAVINGS', 15000, SYSDATE);

INSERT INTO Accounts (AccountID, Customer\_ID, AccountType, Balance, LastModified)

VALUES (102, 2, 'SAVINGS', 23000, SYSDATE);

INSERT INTO Accounts (AccountID, Customer\_ID, AccountType, Balance, LastModified)

VALUES (103, 3, 'SAVINGS', 1000, SYSDATE);

COMMIT;

BEGIN

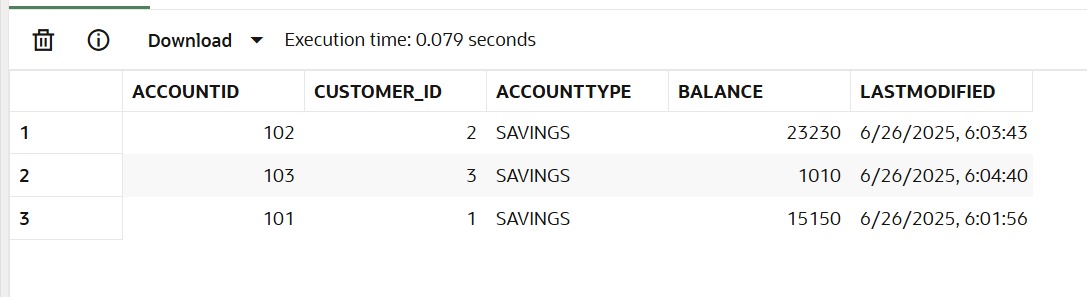
    ProcessMonthlyInterest;

END;

/

SELECT \* FROM Accounts;

**OUTPUT:**

****

**SCENARIO 2:**

CREATE TABLE Employees (

    EmployeeID NUMBER PRIMARY KEY,

    Name VARCHAR2(100),

    Position VARCHAR2(50),

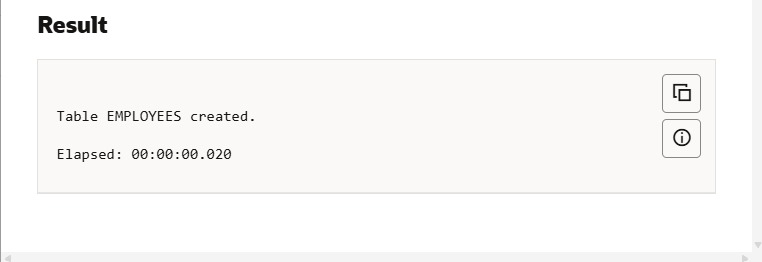
    Salary NUMBER,

    Department VARCHAR2(50),

    HireDate DATE

);

**OUTPUT:**



INSERT INTO Employees VALUES (1, 'Alice Johnson', 'Manager', 70000, 'HR', TO\_DATE('2015-06-15', 'YYYY-MM-DD'));

INSERT INTO Employees VALUES (2, 'Bob Brown', 'Developer', 60000, 'IT', TO\_DATE('2017-03-20', 'YYYY-MM-DD'));

INSERT INTO Employees VALUES (3, 'Charlie Ray', 'Tester', 50000, 'IT', TO\_DATE('2019-08-01', 'YYYY-MM-DD'));

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

    p\_department IN VARCHAR2,

    p\_bonus\_percent IN NUMBER

)

AS

BEGIN

    UPDATE Employees

    SET Salary = Salary + (Salary \* p\_bonus\_percent / 100)

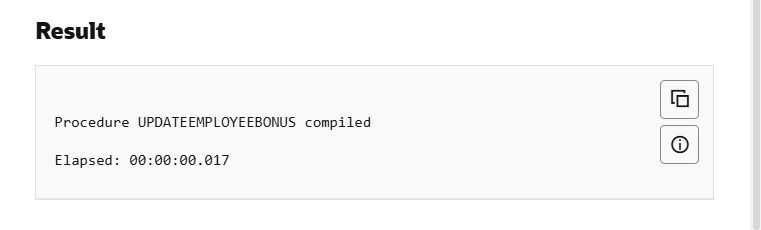
    WHERE Department = p\_department;

    DBMS\_OUTPUT.PUT\_LINE('Bonus of ' || p\_bonus\_percent || '% applied to department: ' || p\_department);

END;

/

**OUTPUT:**



BEGIN

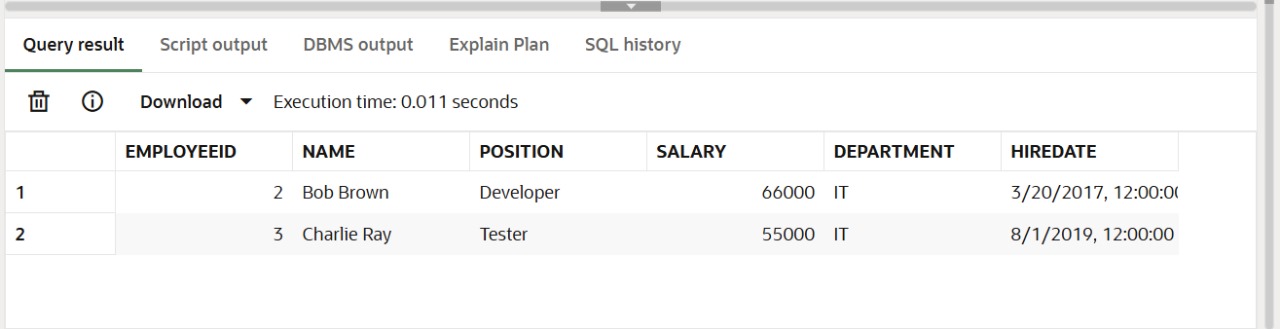
    UpdateEmployeeBonus('IT', 10);

END;

/

SELECT \* FROM Employees WHERE Department = 'IT';

**OUTPUT:**

****

**SCENARIO 3:**

CREATE TABLE Accounts (

    AccountID NUMBER PRIMARY KEY,

    CustomerID NUMBER,

    AccountType VARCHAR2(20),

    Balance NUMBER,

    LastModified DATE,

    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

CREATE OR REPLACE PROCEDURE TransferFunds(

    p\_from\_account IN NUMBER,

    p\_to\_account   IN NUMBER,

    p\_amount       IN NUMBER

)

IS

    v\_from\_balance NUMBER;

BEGIN

    IF p\_amount <= 0 THEN

        RAISE\_APPLICATION\_ERROR(-20001, 'Transfer amount must be greater than zero.');

    END IF;

    -- Get current balance of the source account

    SELECT Balance INTO v\_from\_balance

    FROM Accounts

    WHERE AccountID = p\_from\_account;

    IF v\_from\_balance < p\_amount THEN

        RAISE\_APPLICATION\_ERROR(-20002, 'Insufficient balance in the source account.');

    END IF;

    UPDATE Accounts

    SET Balance = Balance - p\_amount,

        LastModified = SYSDATE

    WHERE AccountID = p\_from\_account;

    UPDATE Accounts

    SET Balance = Balance + p\_amount,

        LastModified = SYSDATE

    WHERE AccountID = p\_to\_account;

    DBMS\_OUTPUT.PUT\_LINE('₹' || p\_amount || ' transferred from Account ID ' || p\_from\_account ||

                         ' to Account ID ' || p\_to\_account);

EXCEPTION

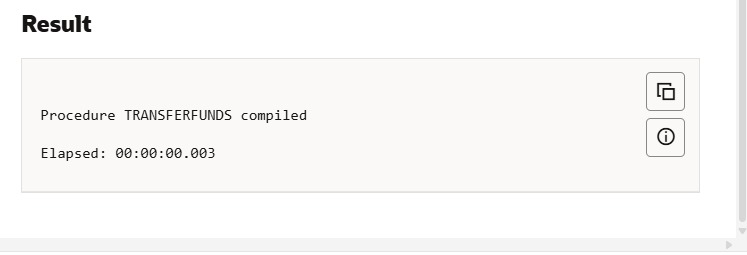
    WHEN NO\_DATA\_FOUND THEN

        RAISE\_APPLICATION\_ERROR(-20003, 'One of the accounts does not exist.');

END;

/

**OUTPUT:**



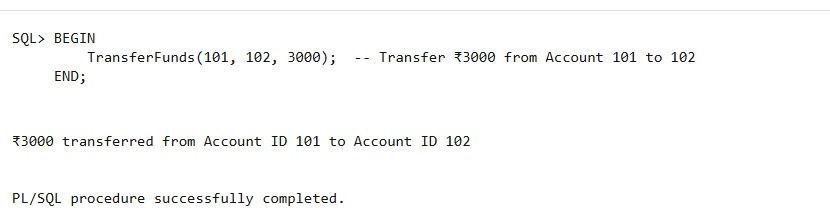
BEGIN

    TransferFunds(101, 102, 3000);  -- Transfer ₹3000 from Account 101 to 102

END;

/

**OUTPUT:**

****