**Exercise 3: Stored Procedures**

**CODE :**

SET SERVEROUTPUT ON;

Step 1: Drop old tables if they exist

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE Accounts CASCADE CONSTRAINTS';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE Employees CASCADE CONSTRAINTS';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

Step 2: Create tables

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

Balance NUMBER(10,2),

AccountType VARCHAR2(10) CHECK (AccountType IN ('SAVINGS', 'CHECKING'))

);

CREATE TABLE Employees (

EmpID NUMBER PRIMARY KEY,

DeptID NUMBER,

Salary NUMBER(10,2)

);

Step 3: Insert sample data

INSERT INTO Accounts (AccountID, CustomerID, Balance, AccountType) VALUES

(101, 1, 7000, 'SAVINGS'),

(102, 2, 12000, 'CHECKING'),

(103, 3, 8500, 'SAVINGS');

INSERT INTO Employees (EmpID, DeptID, Salary) VALUES

(201, 10, 42000),

(202, 10, 46000),

(203, 20, 51000);

COMMIT;

Step 4: Procedure – Apply interest to savings accounts

CREATE OR REPLACE PROCEDURE ApplyMonthlyInterest IS

BEGIN

FOR acc IN (

SELECT AccountID, Balance

FROM Accounts

WHERE AccountType = 'SAVINGS'

) LOOP

UPDATE Accounts

SET Balance = Balance + (acc.Balance \* 0.01)

WHERE AccountID = acc.AccountID;

DBMS\_OUTPUT.PUT\_LINE('✓ Interest added: Account ' || acc.AccountID);

END LOOP;

COMMIT;

END;

/

Step 5: Procedure – Apply bonus to a department

CREATE OR REPLACE PROCEDURE ApplyDeptBonus (

dept\_id IN NUMBER,

bonus\_rate IN NUMBER

) IS

BEGIN

UPDATE Employees

SET Salary = Salary + (Salary \* bonus\_rate / 100)

WHERE DeptID = dept\_id;

DBMS\_OUTPUT.PUT\_LINE('✓ Bonus applied for Dept: ' || dept\_id || ' at ' || bonus\_rate || '%');

COMMIT;

END;

/

Step 6: Procedure – Transfer funds between accounts securely

CREATE OR REPLACE PROCEDURE SmartTransfer (

from\_acct IN NUMBER,

to\_acct IN NUMBER,

amount IN NUMBER

) IS

available\_balance NUMBER;

BEGIN

SELECT Balance INTO available\_balance

FROM Accounts

WHERE AccountID = from\_acct

FOR UPDATE;

IF available\_balance < amount THEN

RAISE\_APPLICATION\_ERROR(-20001, '⚠ Not enough funds in account ' || from\_acct);

END IF;

UPDATE Accounts SET Balance = Balance - amount WHERE AccountID = from\_acct;

UPDATE Accounts SET Balance = Balance + amount WHERE AccountID = to\_acct;

DBMS\_OUTPUT.PUT\_LINE('₹' || amount || ' transferred from A/C ' || from\_acct || ' ➜ A/C ' || to\_acct);

COMMIT;

END;

/

Step 7: Execute each procedure

BEGIN

ApplyMonthlyInterest;

END;

/

BEGIN

ApplyDeptBonus(10, 7); -- 7% bonus for department 10

END;

/

BEGIN

SmartTransfer(102, 103, 2500); -- transfer ₹2500 from 102 ➜ 103

END;

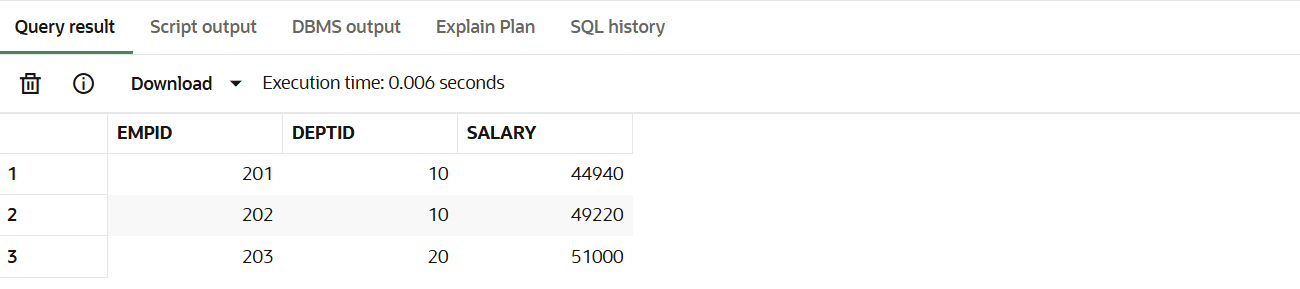
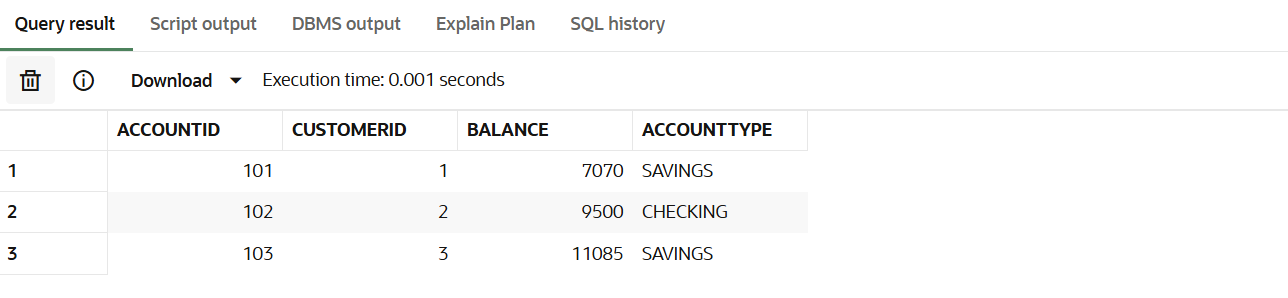
/

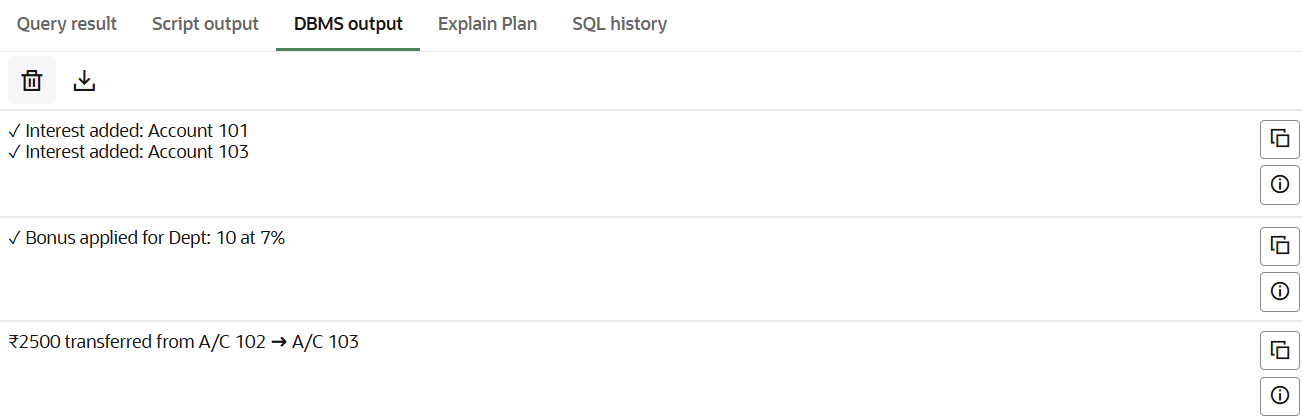
Step 8: View updated results

SELECT \* FROM Accounts;

SELECT \* FROM Employees;

**OUTPUT**:

****

****