**Exercise 3 : Stored Procedures**

**TABLES :**

**CODE :**

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

AccountType VARCHAR2(20), -- 'Savings', 'Checking', etc.

Balance NUMBER(10, 2)

);

CREATE TABLE Employees (

EmployeeID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Department VARCHAR2(50),

Salary NUMBER(10, 2)

);

CREATE TABLE Transfers (

TransferID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

FromAccount NUMBER,

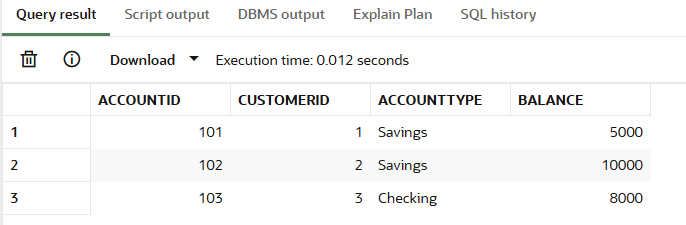
ToAccount NUMBER,

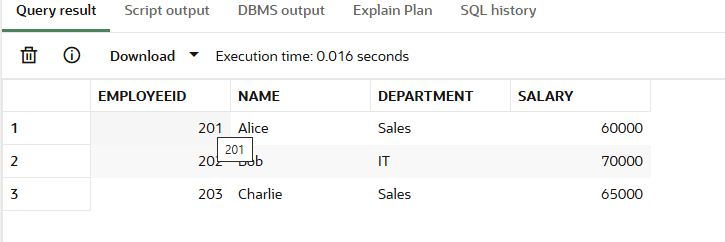
Amount NUMBER(10, 2),

TransferDate DATE DEFAULT SYSDATE

);

**TABLES OUTPUT :**

****

****

**Scenario 1: Monthly Interest Processing**

**CODE :**

**CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS**

**BEGIN**

**UPDATE Accounts**

**SET Balance = Balance \* 1.01**

**WHERE AccountType = 'Savings';**

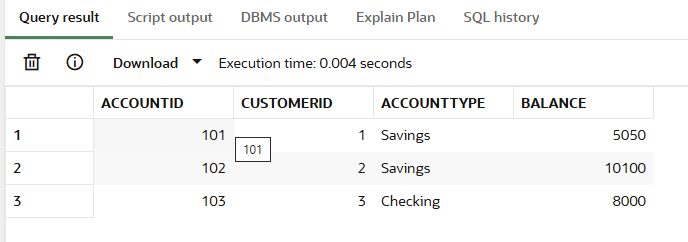
**COMMIT;**

**END;**

**EXEC ProcessMonthlyInterest;**

**SELECT \* FROM Accounts;**

**OUTPUT :**

****

**Scenario 2: Employee Bonus Update**

**CODE :**

**CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (**

**p\_Department IN VARCHAR2,**

**p\_BonusPercent IN NUMBER -- e.g. 10 for 10%**

**) IS**

**BEGIN**

**UPDATE Employees**

**SET Salary = Salary + (Salary \* p\_BonusPercent / 100)**

**WHERE Department = p\_Department;**

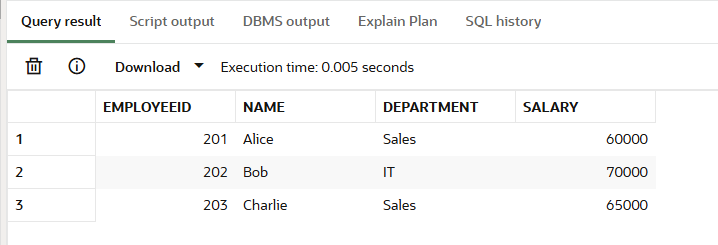
**COMMIT;**

**END;**

**EXEC UpdateEmployeeBonus('Sales', 8);**

**SELECT \* FROM Employees;**

**OUTPUT :**

****

**Scenario 3: Transfer Funds Between Accounts**

**CODE :**

**CREATE OR REPLACE PROCEDURE TransferFunds (**

**p\_FromAccount IN NUMBER,**

**p\_ToAccount IN NUMBER,**

**p\_Amount IN NUMBER**

**) IS**

**v\_FromBalance NUMBER;**

**BEGIN**

**-- Check balance of source account**

**SELECT Balance INTO v\_FromBalance**

**FROM Accounts**

**WHERE AccountID = p\_FromAccount**

**FOR UPDATE;**

**IF v\_FromBalance < p\_Amount THEN**

**RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in source account.');**

**END IF;**

**-- Deduct from source**

**UPDATE Accounts**

**SET Balance = Balance - p\_Amount**

**WHERE AccountID = p\_FromAccount;**

**-- Add to destination**

**UPDATE Accounts**

**SET Balance = Balance + p\_Amount**

**WHERE AccountID = p\_ToAccount;**

**-- Log transfer (optional)**

**INSERT INTO Transfers (FromAccount, ToAccount, Amount)**

**VALUES (p\_FromAccount, p\_ToAccount, p\_Amount);**

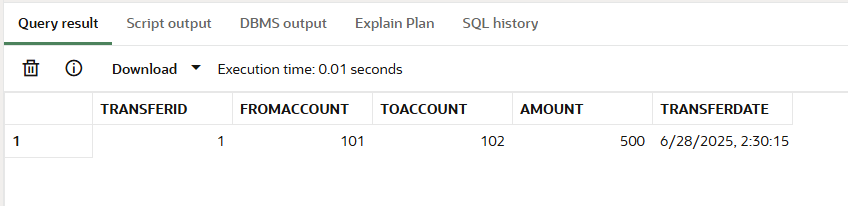
**COMMIT;**

**END;**

**EXEC TransferFunds(101, 102, 500);**

**SELECT \* FROM Transfers;**

**OUTPUT :**

****