PL/SQL Stored Procedures - Exercise 3

# Scenario 1: ProcessMonthlyInterest

Calculates and updates the balance of all savings accounts by applying a 1% interest rate.

## Table: SavingsAccounts

CREATE TABLE SavingsAccounts (  
 AccountID NUMBER PRIMARY KEY,  
 AccountHolder VARCHAR2(100),  
 Balance NUMBER(10, 2)  
);

## Insert Sample Data

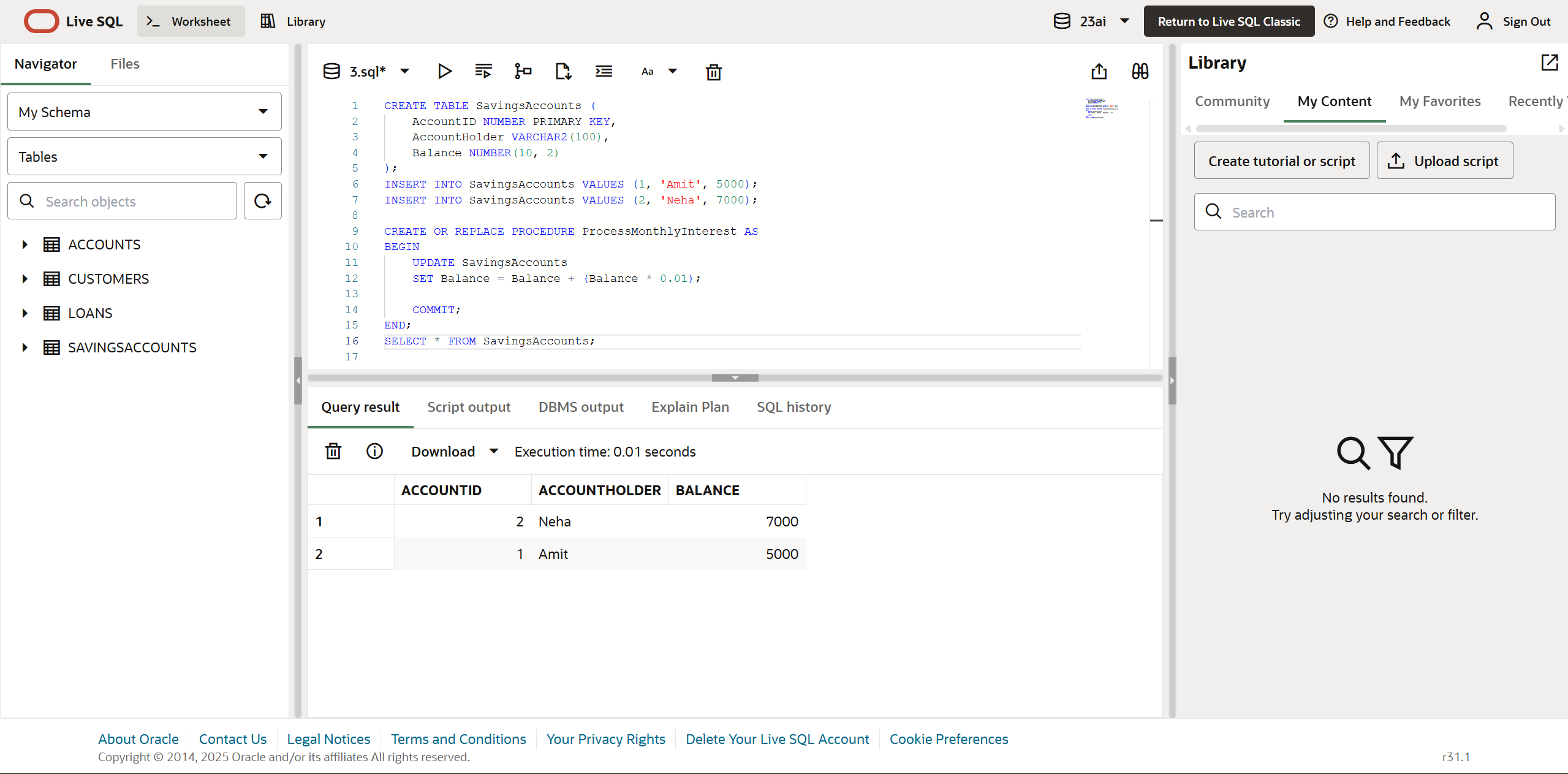
INSERT INTO SavingsAccounts VALUES (1, 'Amit', 5000);  
INSERT INTO SavingsAccounts VALUES (2, 'Neha', 7000);

## Procedure: ProcessMonthlyInterest

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS  
BEGIN  
 UPDATE SavingsAccounts  
 SET Balance = Balance + (Balance \* 0.01);  
  
 COMMIT;  
END;

## Test Output

EXEC ProcessMonthlyInterest;  
SELECT \* FROM SavingsAccounts;



# Scenario 2: UpdateEmployeeBonus

Updates salary of employees in a given department by adding a bonus percentage.

## Table: Employees

CREATE TABLE Employees (  
 EmpID NUMBER PRIMARY KEY,  
 Name VARCHAR2(100),  
 Department VARCHAR2(50),  
 Salary NUMBER(10, 2)  
);

## Insert Sample Data

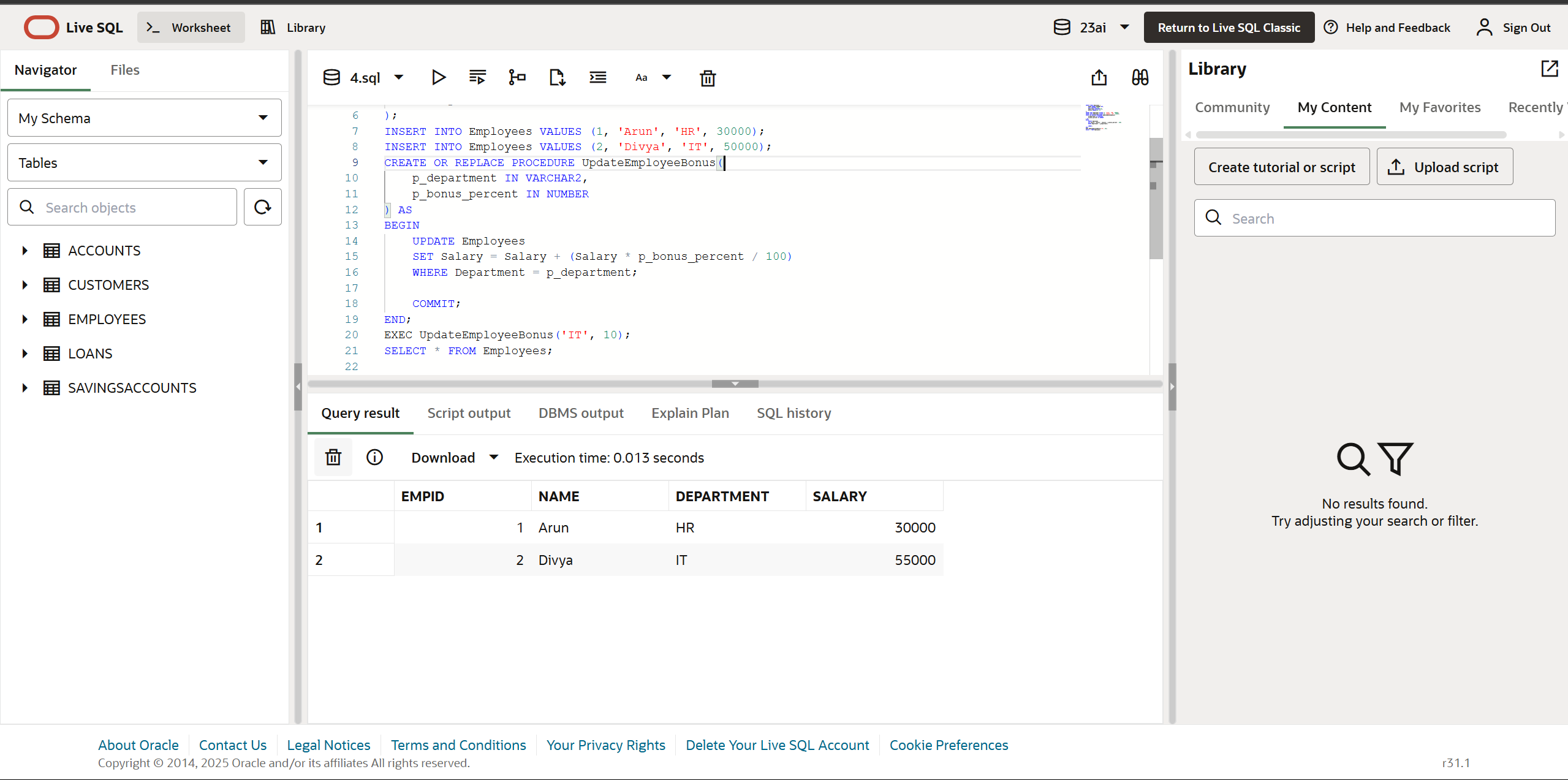
INSERT INTO Employees VALUES (1, 'Arun', 'HR', 30000);  
INSERT INTO Employees VALUES (2, 'Divya', 'IT', 50000);

## Procedure: UpdateEmployeeBonus

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;  
  
 COMMIT;  
END;

## Test Output

EXEC UpdateEmployeeBonus('IT', 10);  
SELECT \* FROM Employees;



# Scenario 3: TransferFunds

Transfers funds from one account to another after checking sufficient balance.

## Table: BankAccounts

CREATE TABLE BankAccounts (  
 AccountID NUMBER PRIMARY KEY,  
 HolderName VARCHAR2(100),  
 Balance NUMBER(10, 2)  
);

## Insert Sample Data

INSERT INTO BankAccounts VALUES (101, 'Sameer', 8000);  
INSERT INTO BankAccounts VALUES (102, 'Pooja', 3000);

## Procedure: TransferFunds

CREATE OR REPLACE PROCEDURE TransferFunds(  
 p\_from\_account IN NUMBER,  
 p\_to\_account IN NUMBER,  
 p\_amount IN NUMBER  
) AS  
 v\_balance NUMBER;  
BEGIN  
 SELECT Balance INTO v\_balance FROM BankAccounts WHERE AccountID = p\_from\_account;  
  
 IF v\_balance < p\_amount THEN  
 RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance.');  
 END IF;  
  
 UPDATE BankAccounts SET Balance = Balance - p\_amount WHERE AccountID = p\_from\_account;  
 UPDATE BankAccounts SET Balance = Balance + p\_amount WHERE AccountID = p\_to\_account;  
  
 COMMIT;  
END;

## Test Output

EXEC TransferFunds(101, 102, 2000);  
SELECT \* FROM BankAccounts;

