Stores\_Procedures

--Scenario 1

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR acc IN (

SELECT AccountID, Balance

FROM Accounts

WHERE AccountType = 'Savings'

) LOOP

UPDATE Accounts

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

LastModified = SYSDATE

WHERE AccountID = acc.AccountID;

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

END LOOP;

COMMIT;

END;

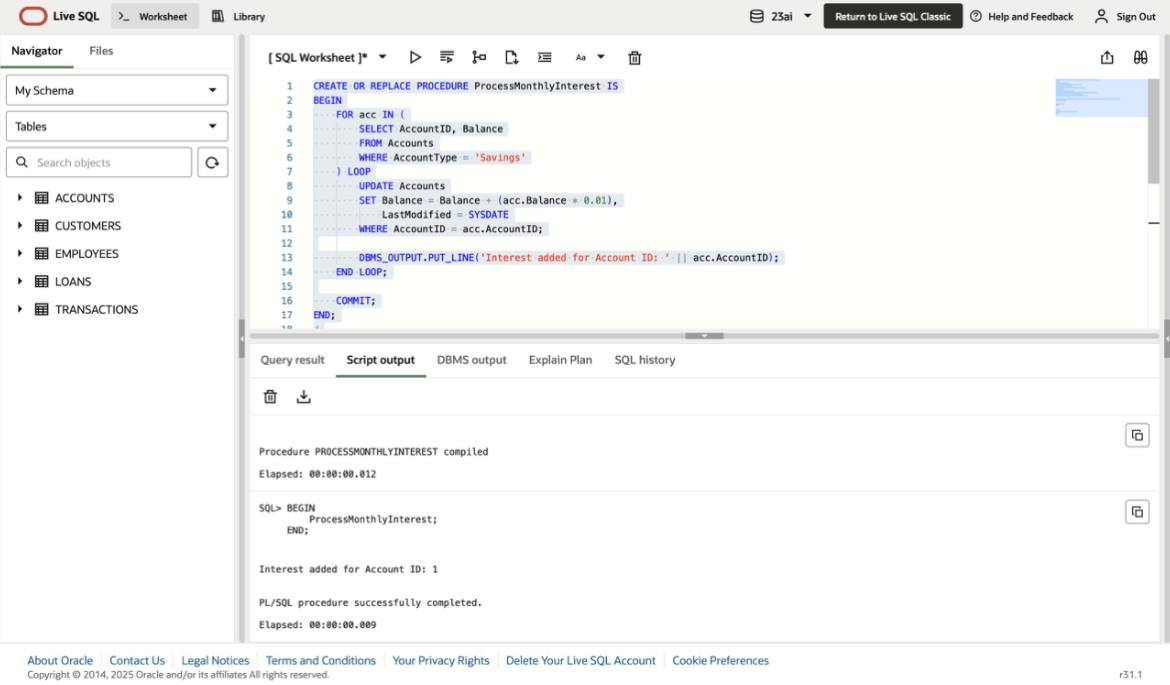
/

BEGIN

ProcessMonthlyInterest;

END;

/



--- Scenario 2

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_dept IN VARCHAR2,

p\_bonus\_percent IN NUMBER

) IS

BEGIN

FOR emp IN (

SELECT EmployeeID, Salary FROM Employees WHERE Department = p\_dept

) LOOP

UPDATE Employees

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

WHERE EmployeeID = emp.EmployeeID;

DBMS\_OUTPUT.PUT\_LINE('Bonus applied to Employee ID: ' || emp.EmployeeID);

END LOOP;

COMMIT;

END;

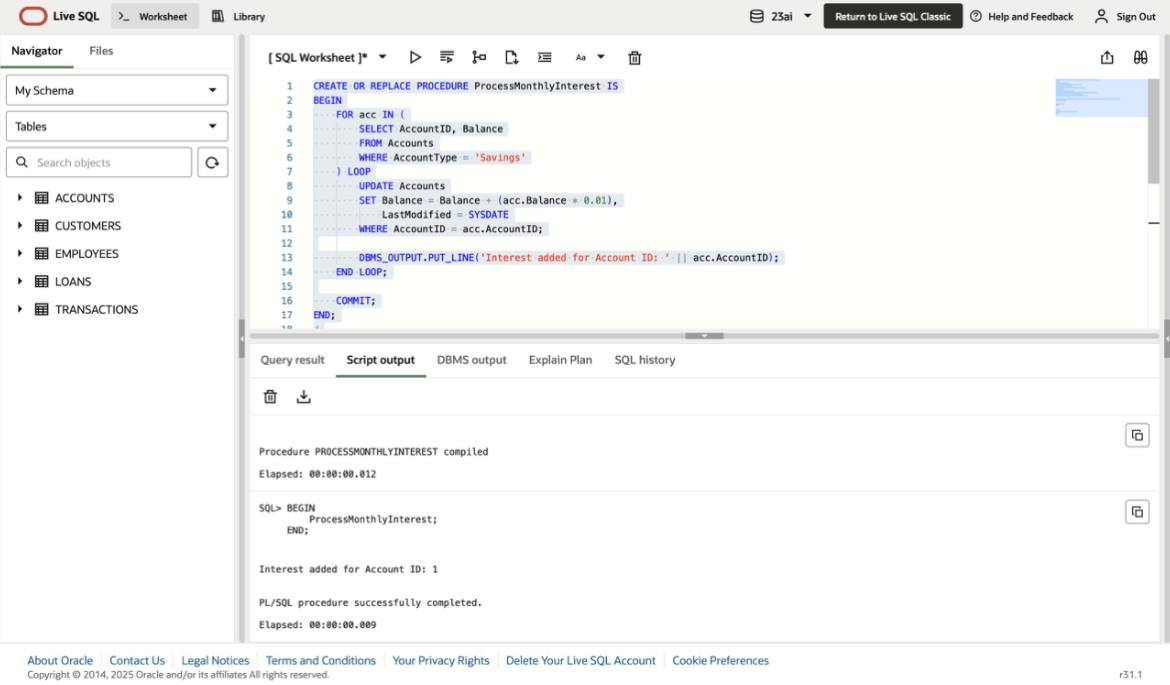
/

BEGIN

UpdateEmployeeBonus('IT', 10);

END;

/



--- Scenario 3

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_acc IN NUMBER,

p\_to\_acc IN NUMBER,

p\_amt IN NUMBER

) IS

v\_bal NUMBER;

BEGIN

-- Check balance

SELECT Balance INTO v\_bal FROM Accounts WHERE AccountID = p\_from\_acc;

IF v\_bal < p\_amt THEN

RAISE\_APPLICATION\_ERROR(-20010, 'Insufficient funds in source account.');

END IF;

-- Debit and Credit

UPDATE Accounts SET Balance = Balance - p\_amt WHERE AccountID = p\_from\_acc;

UPDATE Accounts SET Balance = Balance + p\_amt WHERE AccountID = p\_to\_acc;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Transferred ' || p\_amt || ' from ' || p\_from\_acc || ' to ' || p\_to\_acc);

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/

BEGIN

TransferFunds(2, 3, 500);

END;

/

