**Exercise 3: Stored Procedures**

**Scenario 1:** The bank needs to process monthly interest for all savings accounts.

**Question:** Write a stored procedure **ProcessMonthlyInterest** that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

**SOLUTION SCRIPT**

**CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS**

**BEGIN**

**FOR RECORD IN (**

**SELECT CUSTOMERID FROM CUSTOMERS**

**FOR UPDATE**

**) LOOP**

**UPDATE CUSTOMERS**

**SET BALANCE = BALANCE + BALANCE \* 0.01**

**WHERE CUSTOMERID = RECORD.CUSTOMERID;**

**END LOOP;**

**COMMIT;**

**END;**

**/**

**-- or**

**CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS**

**BEGIN**

**UPDATE CUSTOMERS**

**SET BALANCE = BALANCE + BALANCE \* 0.01;**

**COMMIT;**

**END;**

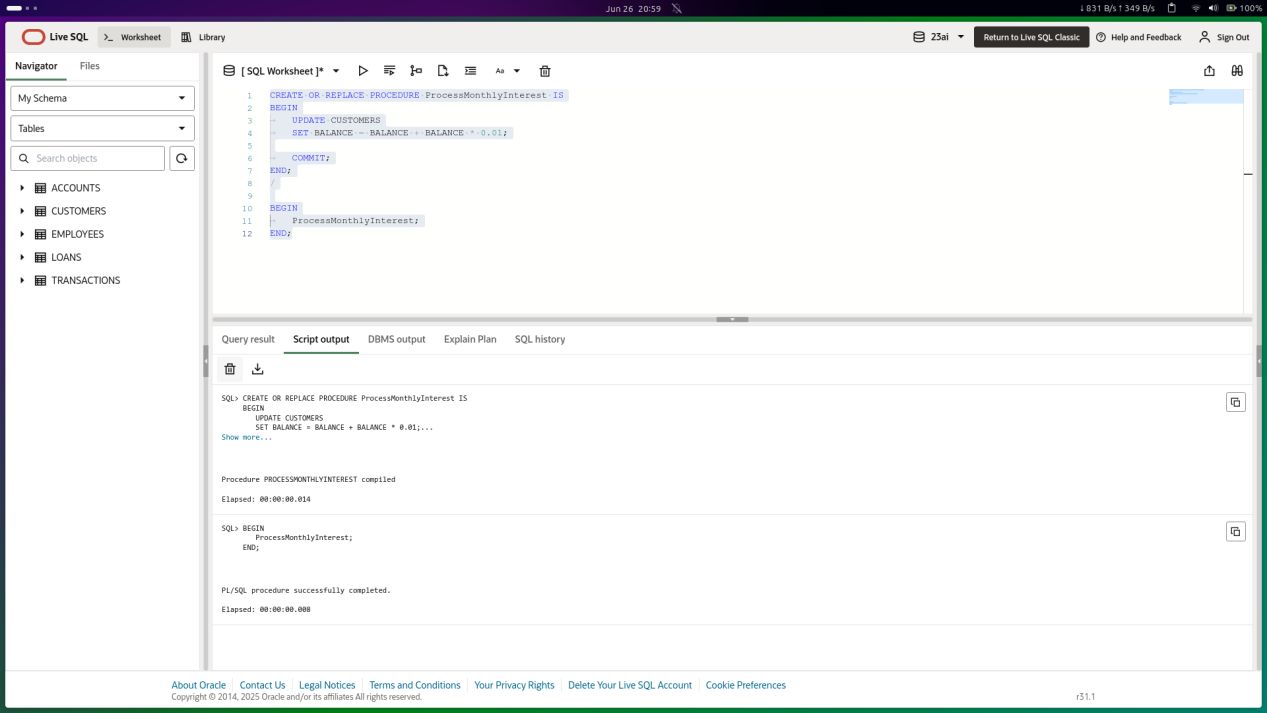
**/**

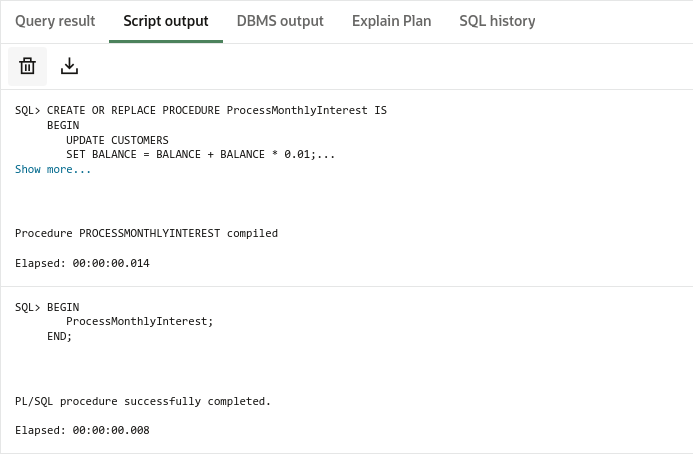
**BEGIN**

**ProcessMonthlyInterest;**

**END;**

**OUTPUT**

****

****

**Scenario 2:** The bank wants to implement a bonus scheme for employees based on their performance.

**Question:** Write a stored procedure **UpdateEmployeeBonus** that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

**SOLUTION SCRIPT**

**CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (**

**dept\_name IN VARCHAR2,**

**bonus\_percent IN NUMBER**

**) IS**

**BEGIN**

**FOR RECORD IN (**

**SELECT EMPLOYEEID FROM EMPLOYEES**

**WHERE DEPARTMENT = dept\_name**

**FOR UPDATE**

**) LOOP**

**UPDATE EMPLOYEES**

**SET SALARY = SALARY + SALARY \* (bonus\_percent/100)**

**WHERE EMPLOYEEID = RECORD.EMPLOYEEID;**

**END LOOP;**

**DBMS\_OUTPUT.PUT\_LINE('Bonus added to ' || dept\_name || ' Employees!');**

**END;**

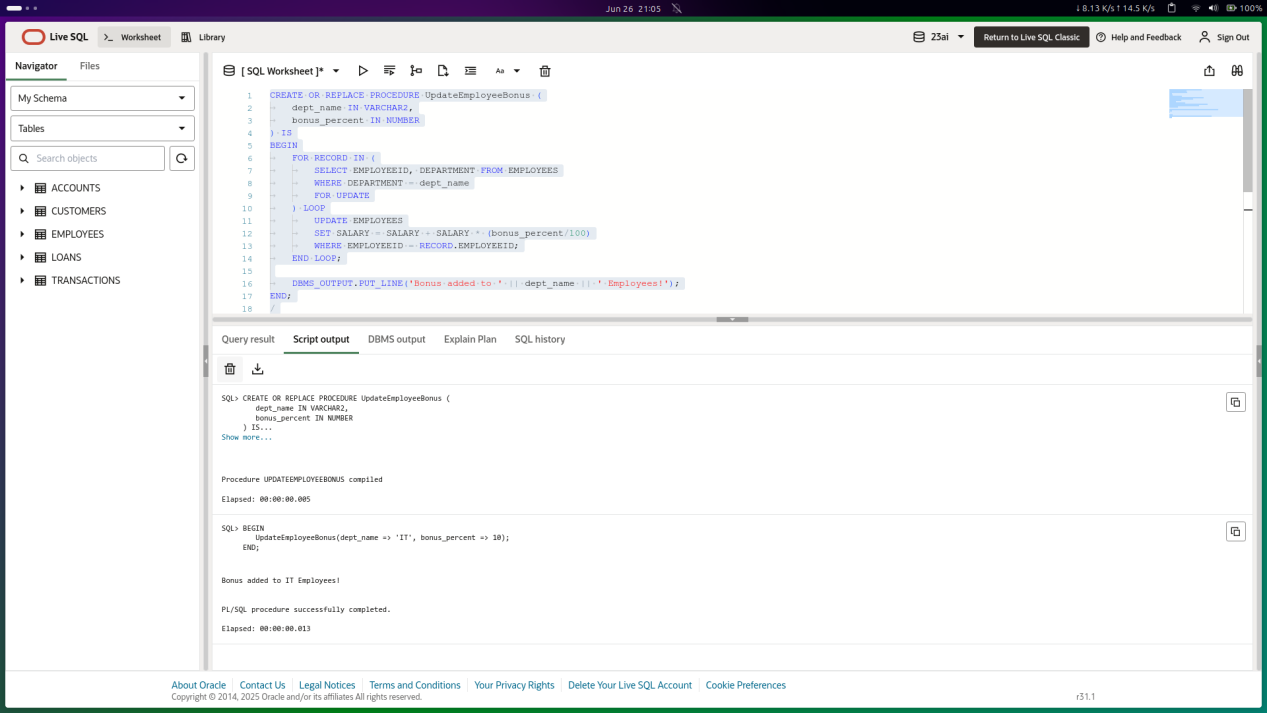
**/**

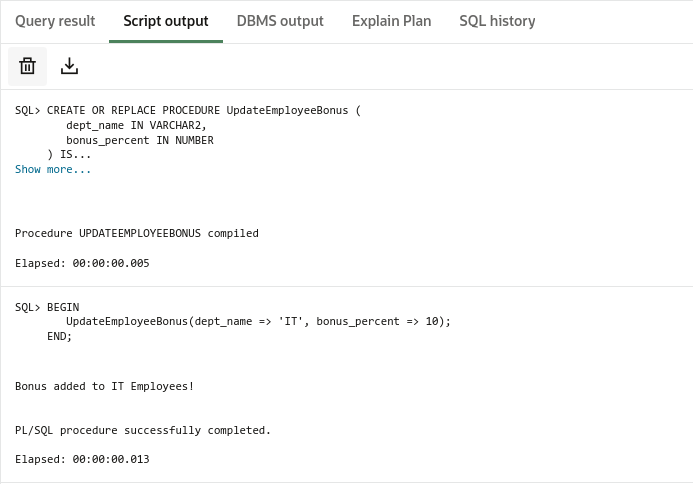
**BEGIN**

**UpdateEmployeeBonus(dept\_name => 'IT', bonus\_percent => 10);**

**END;**

**OUTPUT**

****

****

**Scenario 3:** Customers should be able to transfer funds between their accounts.

**Question:** Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

**SOLUTION SCRIPT**

**CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (**

**dept\_name IN VARCHAR2,**

**bonus\_percent IN NUMBER**

**) IS**

**BEGIN**

**FOR RECORD IN (**

**SELECT EMPLOYEEID FROM EMPLOYEES**

**WHERE DEPARTMENT = dept\_name**

**FOR UPDATE**

**) LOOP**

**UPDATE EMPLOYEES**

**SET SALARY = SALARY + SALARY \* (bonus\_percent/100)**

**WHERE EMPLOYEEID = RECORD.EMPLOYEEID;**

**END LOOP;**

**DBMS\_OUTPUT.PUT\_LINE('Bonus added to ' || dept\_name || ' Employees!');**

**END;**

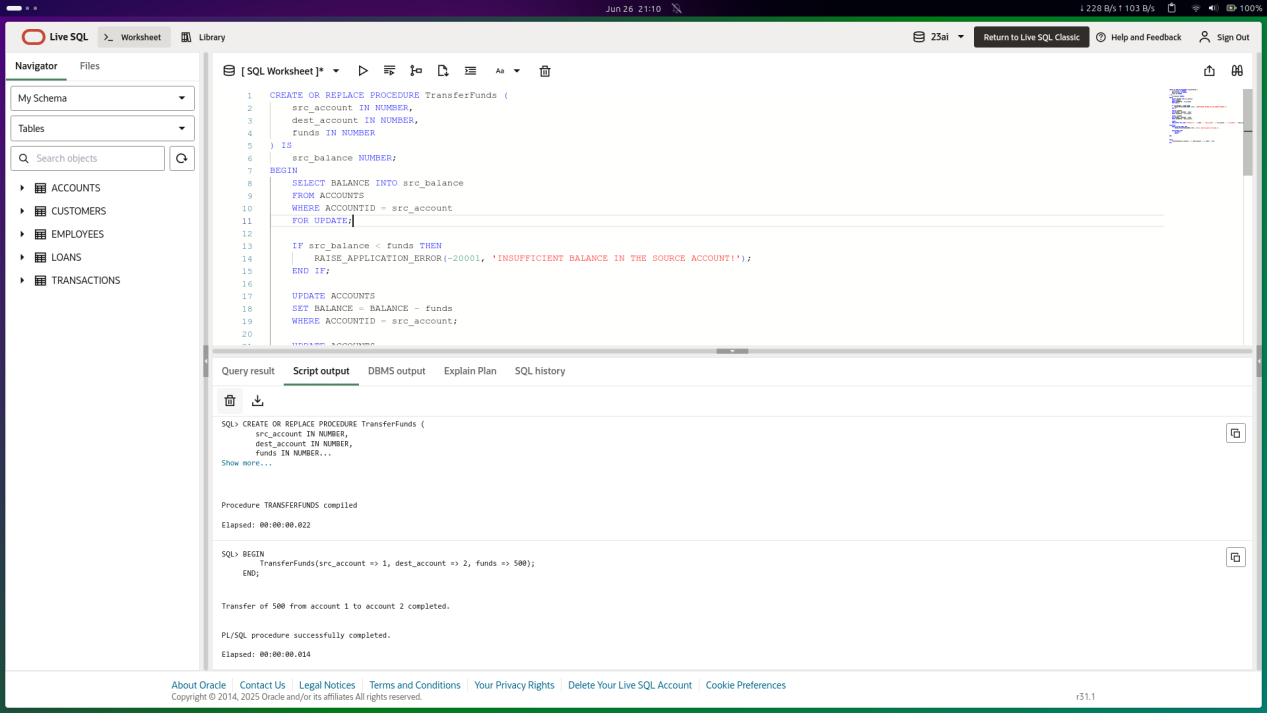
**/**

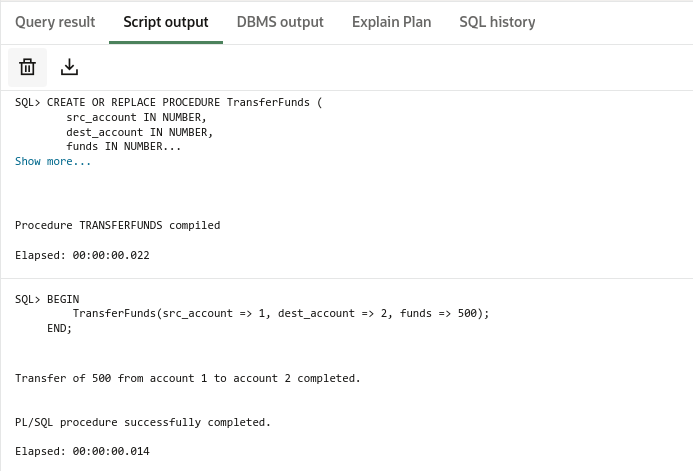
**BEGIN**

**UpdateEmployeeBonus(dept\_name => 'IT', bonus\_percent => 10);**

**END;**

**OUTPUT**

****

****