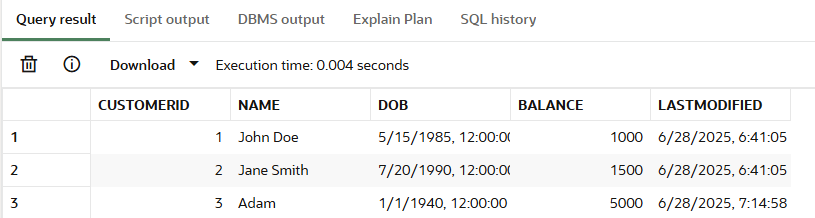
**Week 2 – Hands-on : PL/SQL**

Exercise 1 : Control Structures

**Scenario 1 :** Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates.

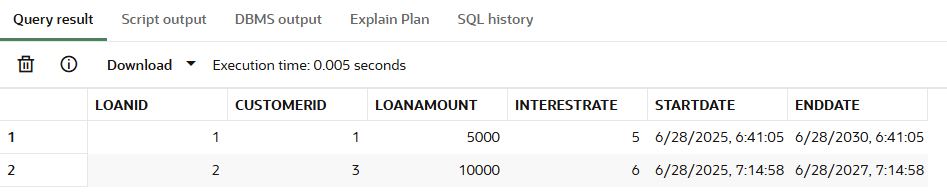
**CUSTOMERS TABLE :**



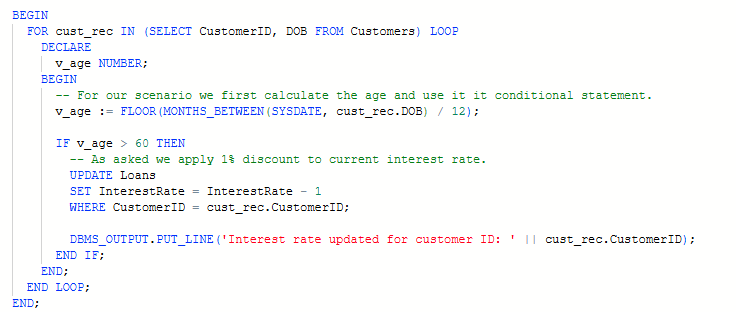


**LOANS TABLE (before executing scenario 1):**

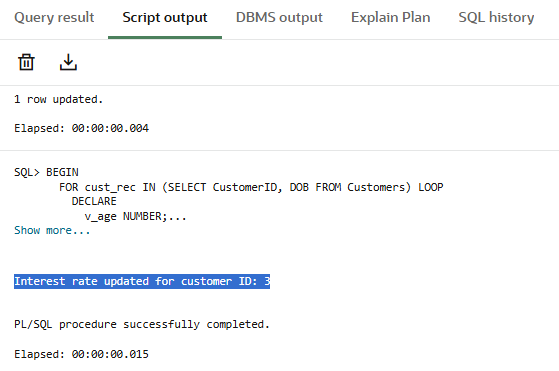
****

****

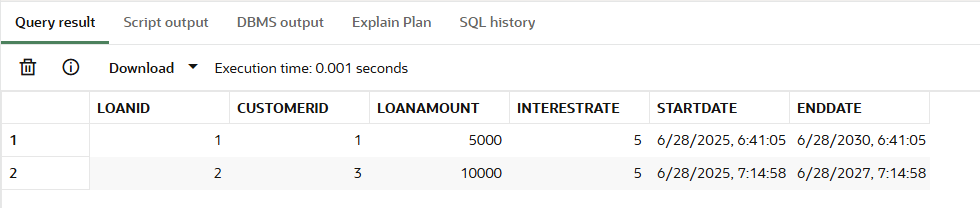
**Solution :**

****

**Script Output :**

****

**LOANS TABLE (after executing scenario 1):**

****

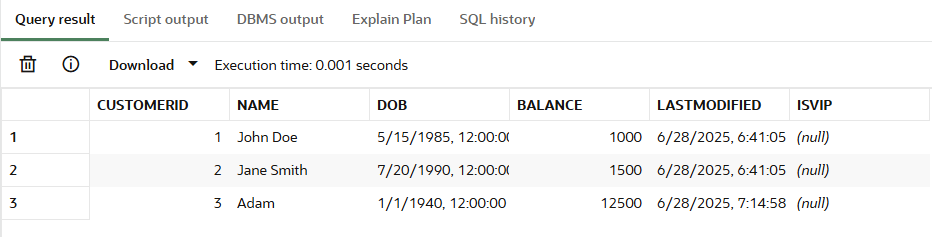
**Scenario 2 :** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.

For this scenario, I have added a new column “IsVIP” and increased balance of a customer to satisfy the constraint balance > 10000.

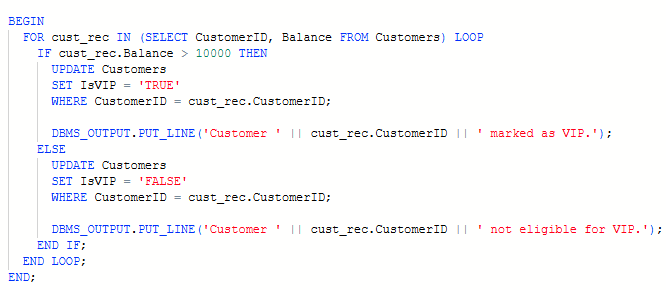




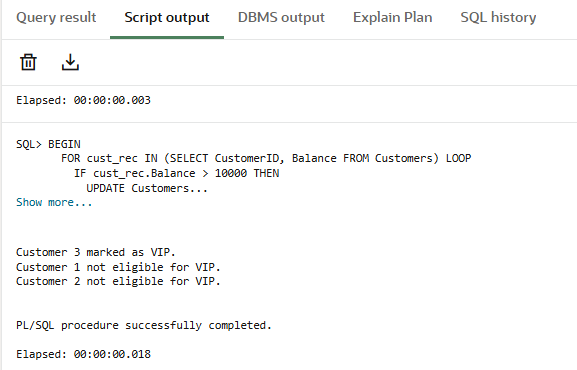
**CUSTOMERS TABLE (before executing scenario):**

****

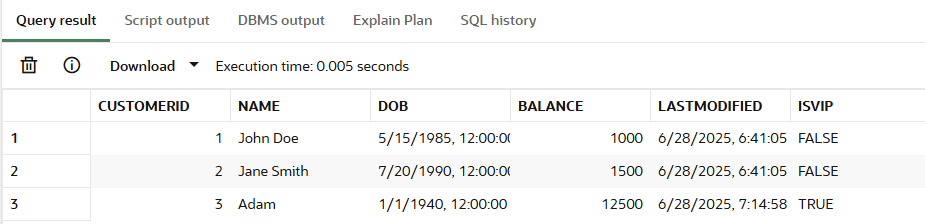
**Solution :**

****

**Script Output :**

****

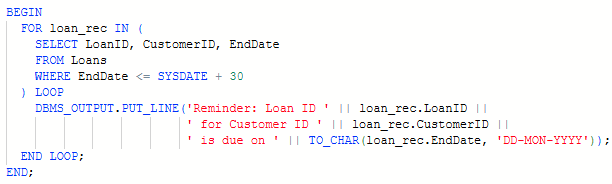
**CUSTOMERS TABLE (after executing scenario):**

****

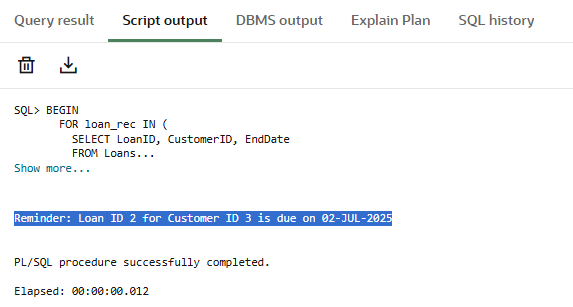
**Scenario 3 :** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

For this scenario I have modified the End date in loans table to fit in for the constraint.

****

****

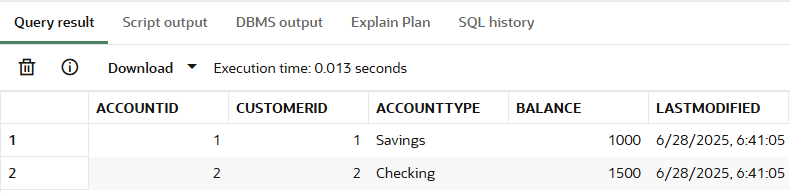
**Script Output :**



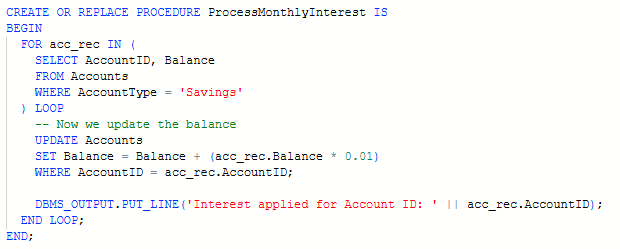
Exercise 3 : Stored Procedures

**Scenario 1 :** 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.

**ACCOUNTS TABLE (before execution of scenario 1) :**

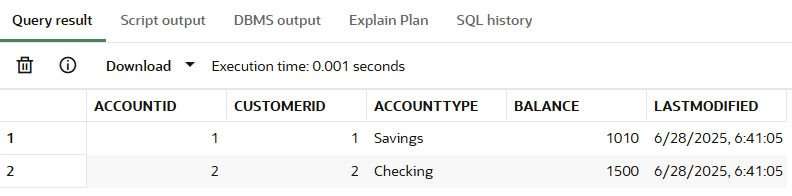
****

**SOLUTION :**

****

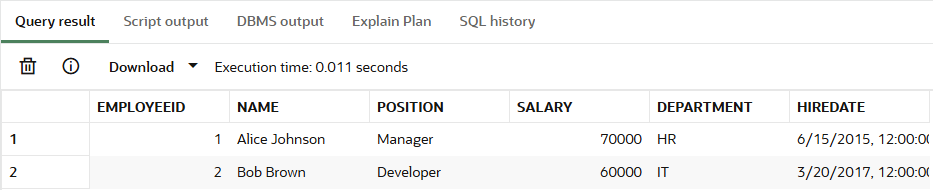
**ACCOUNTS TABLE (after executing scenario 1) :**

****

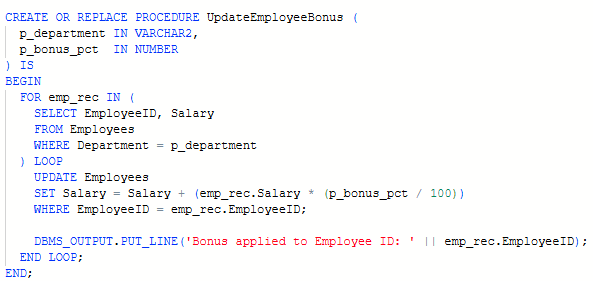
****

**Scenario 2 :** Write a stored procedure **UpdateEmployeeBonus** that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter

**EMPLOYEE TABLE(before executing scenario 2) :**

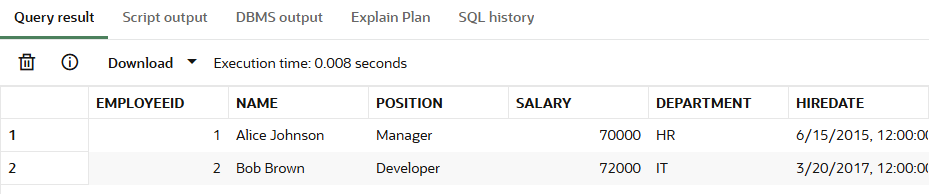


**SOLUTION :**

****

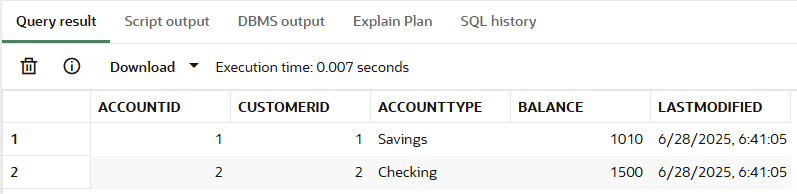
**EMPLOYEE TABLE(after execution of scenario 2) :**



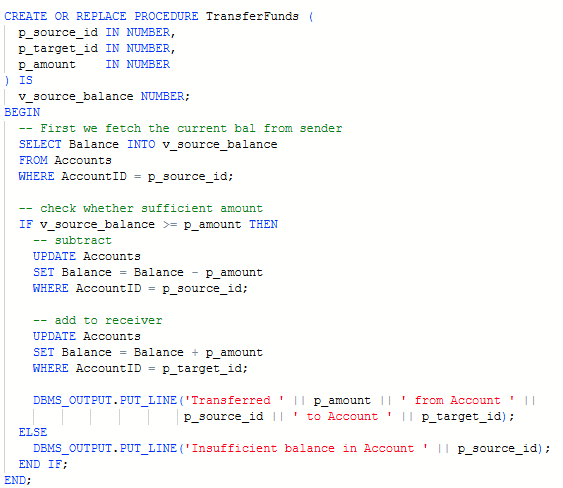


**Scenario 3 :** Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

**ACCOUNTS TABLE (before executing scenario 3) :**

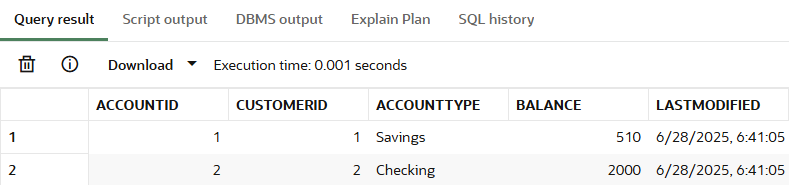


**SOLUTION :**

****

**ACCOUNTS TABLE (after executing scenario 3) :**

****

****