**Exercise 1: Control Structures**

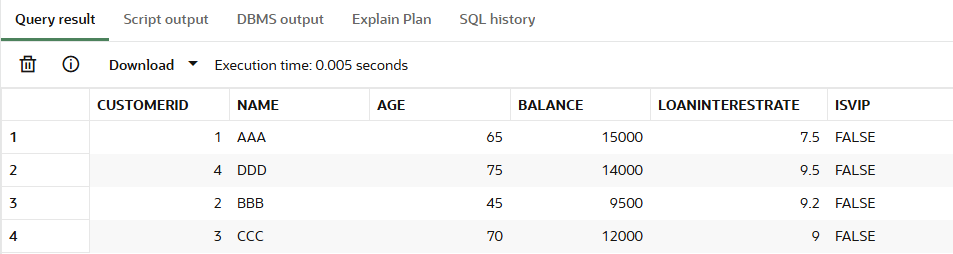
**Scenario Descriptions:**

1. Interest Rate Discount for Senior Citizens:  
   The bank offers a 1% discount on loan interest rates for customers aged above 60. The PL/SQL block loops through the Customers table and applies the discount where applicable.
2. Promotion to VIP Status Based on Balance:  
   Customers maintaining a bank balance of more than $10,000 are eligible for VIP status. This block iterates through all customers and updates the IsVIP flag accordingly.
3. Loan Due Reminders:  
   Customers whose loans are due within the next 30 days should receive a reminder. This block selects relevant loan records and prints a custom message for each customer.

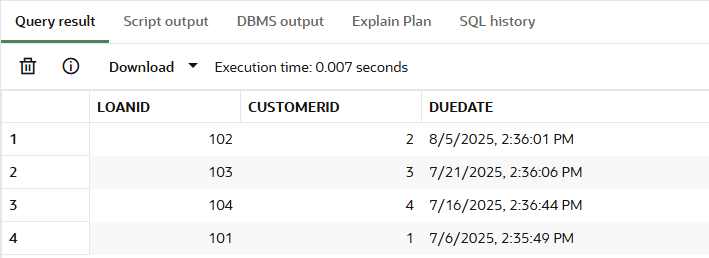
**Table Creation:**

To support the three scenarios given we need to create two main tables:

**Customer Table:**

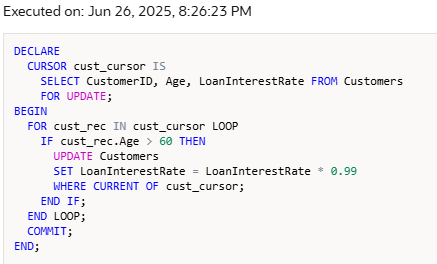


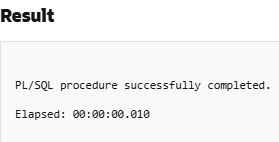
**Loan Table:**



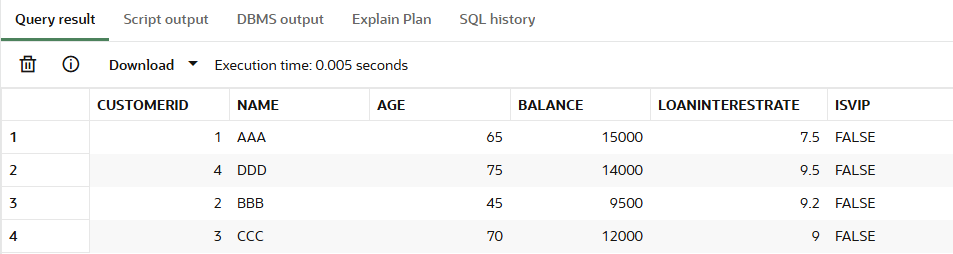
**Scenario 1: Apply 1% Discount to Customers Over 60:**

* 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.

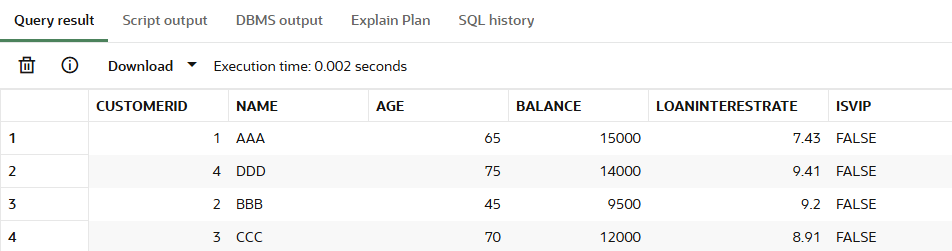
****



**Customer Table Before Applying 1% Discount:**



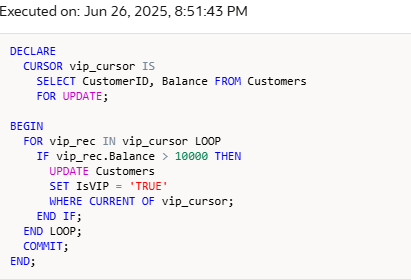
**Customer Table After Applying 1% Discount:**

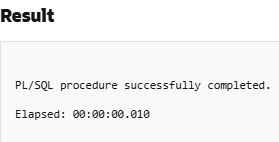
****

Customers named AAA, CCC, and DDD are above 60 years of age, so a 1% discount was applied to their LoanInterestRate by multiplying it with 0.99, effectively reducing the rate by 1% of its current value.

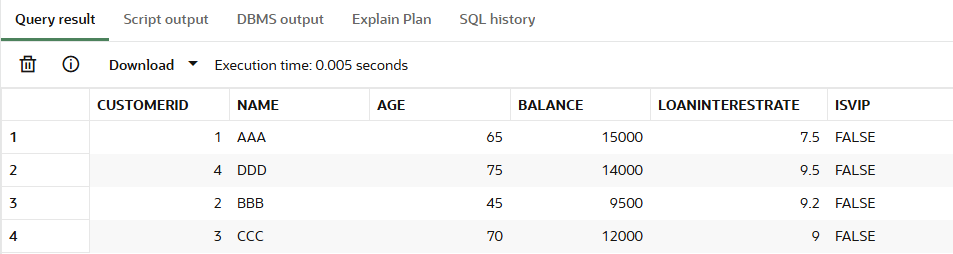
**Scenario 2: A customer can be promoted to VIP status based on their balance.**

o 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.

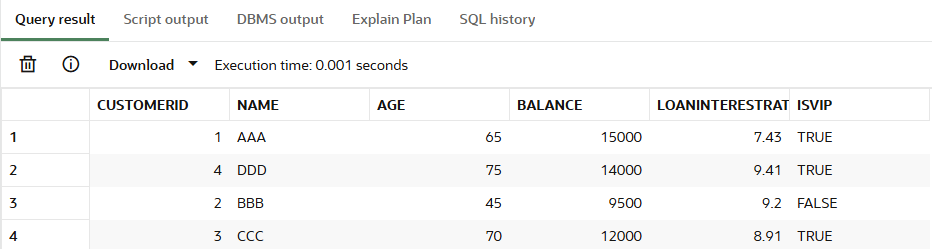




**Customer Table Before Applying VIP Status:**



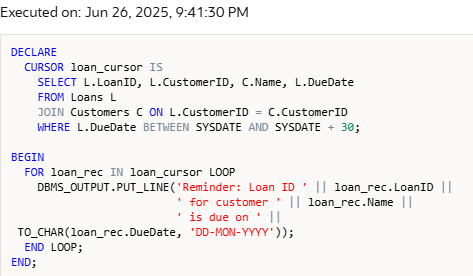
**Customer Table After Applying VIP Status**

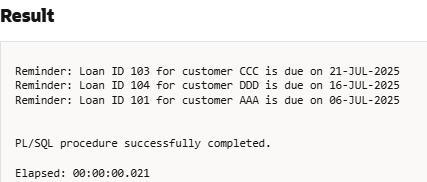
****

Customers named AAA, CCC, and DDD balances are above $10000 so they are granted with VIP.

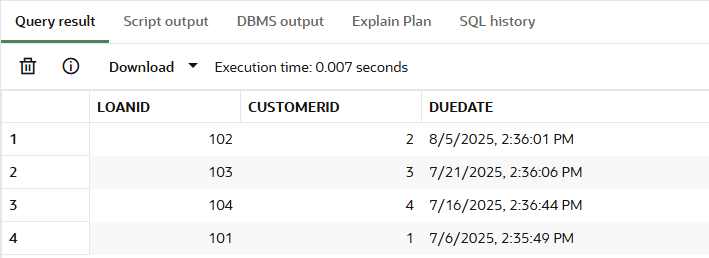
**Scenario 3: Remind Customers with Loans Due in Next 30 Days**

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





Assume today's date is 06/26/2025. From the loan table



LOANID 101,103,104 loan dues are in next 30 days, as they receives the reminder. LOANID 102 is not shown because it is more than 30 days.

**Exercise 3: Stored Procedures**

**Scenario Descriptions**

1.Calculate and update 1% monthly interest for all savings accounts.

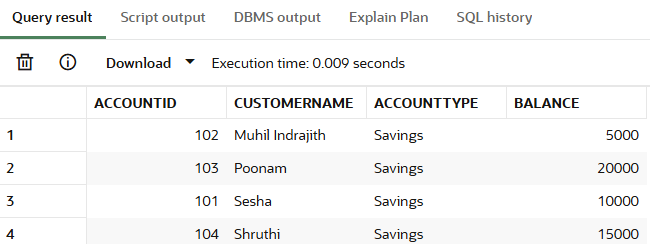
2.Apply a bonus percentage to employee salaries in a given department.

3.Transfer funds between two accounts after checking sufficient balance.

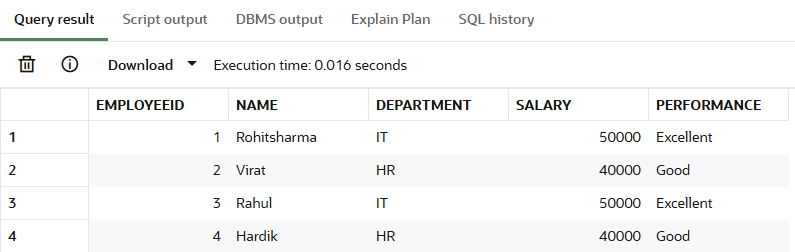
**Table Creation:**

To support the three scenarios given we need to create two main tables:

**Accounts Table:**

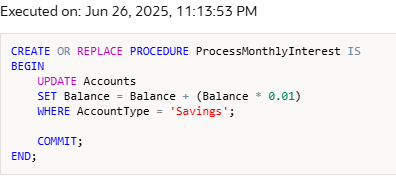
****

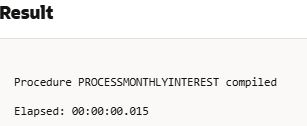
**Employee Table:**

****

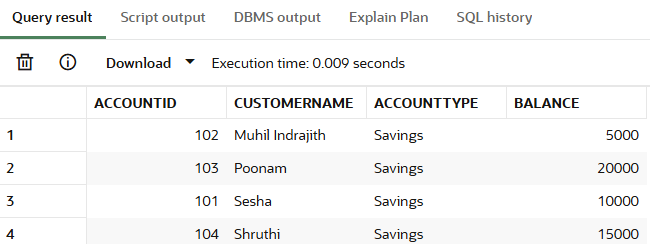
**Scenario 1: Monthly Interest for Savings Accounts**

o 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.

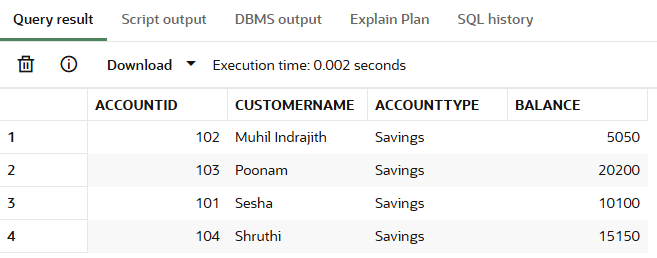




**Accounts Table Before Updating Balance:**

****

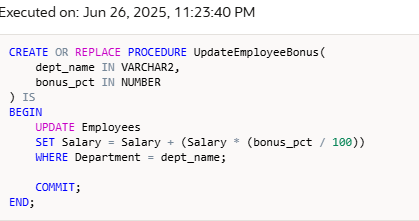
**Accounts Table After Updating Balance:**

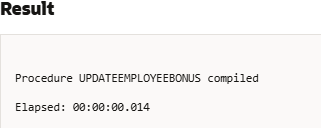


All balance are updated with 1% of interest to their balance.

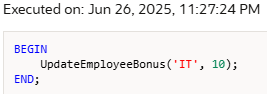
**Scenario 2: Update Employee Bonus by Department**

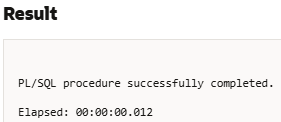
o Write a stored procedure UpdateEmployeeBonus that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.



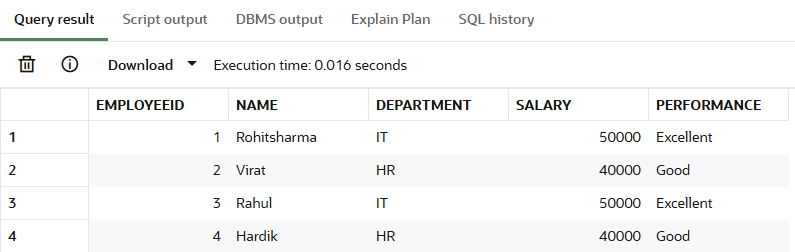


Now we are going to update bonus for the Department IT with bonus percentage 10%

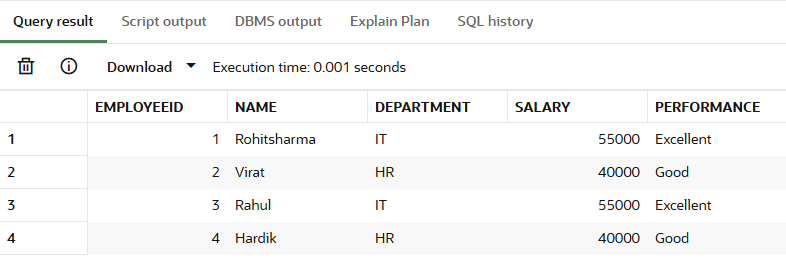




**Employee table before Updating bonus of IT Department:**

****

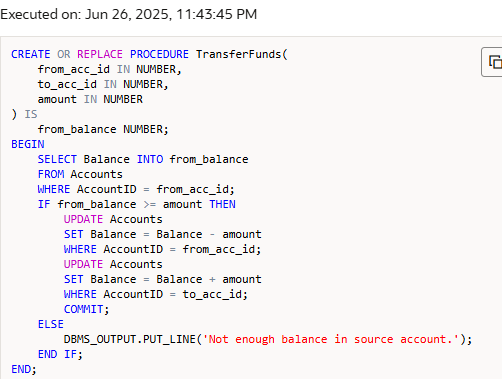
**Employee table after Updating bonus of IT Department:**

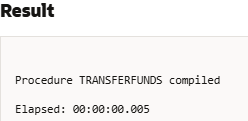
****

IT Department Employees are updated with 10% bonus.

**Scenario 3: Transfer Funds Between Accounts**

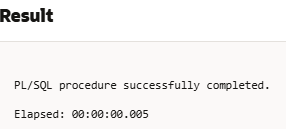
o Write a stored procedure TransferFunds that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.



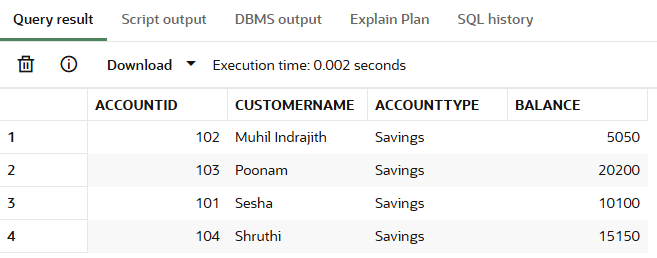


We have created a procedure to transfer money from one account to another. Let check it by transferring 2000 from Account ID 101 to Account ID 103 .





**Account table (Balance updated with 1%) before transferring 2000 from AccountID 101 to AccountID 103. AccountID 101 balance was 10100 and AccountID 103Balance was 5050.**



**Account table after transferring. AccountID 103 balance changes to 22200.**

