PL-SQL Exercise Results

Exercise-1: Control Structures

For code visit

Scenario-1:

Apply *1%* loan discount on interest rates for users whose *age > 60*

Results

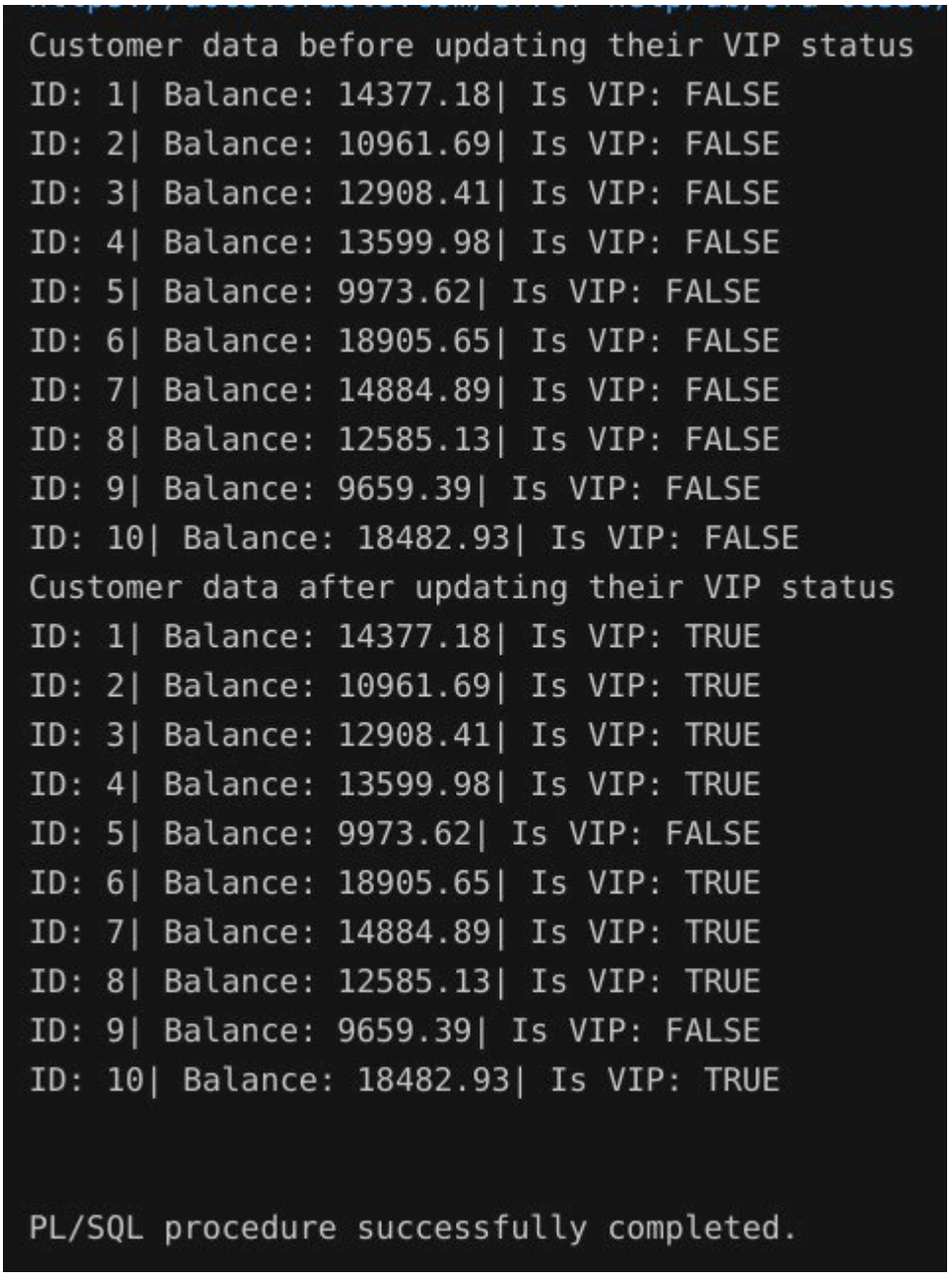
```
Customer data whose age is greater than 60 before applying 1% discount on interest rate
User_ID: 2 | Age: 79 | Loan_ID: 1 | Interest: 7.77
User_ID: 8 | Age: 69 | Loan_ID: 7 | Interest: 9.14
User_ID: 9 | Age: 72 | Loan_ID: 8 | Interest: 6.07
User_ID: 1 | Age: 74 | Loan_ID: 10 | Interest: 9.11
Customer data after applying 1% discount on interest rate
User_ID: 2 | Age: 79 | Loan_ID: 1 | Interest: 6.77
User_ID: 8 | Age: 69 | Loan_ID: 7 | Interest: 8.14
User_ID: 9 | Age: 72 | Loan_ID: 8 | Interest: 5.07
User_ID: 1 | Age: 74 | Loan_ID: 10 | Interest: 8.11


PL/SQL procedure successfully completed.
```

Scenario-2:

Update customer status as ***VIP*** if their balance is over 10,000

Results



Scenario-3:

Get data of users whose loan due date is less than next *30 days* with appropriate message

Results

```
Loan data with due date message:
ID: 2 | Loan ID: 1 | Due Date: 2025-06-26 | Message: Loan is due in 1 day(s).
ID: 3 | Loan ID: 2 | Due Date: 2025-06-27 | Message: Loan is due in 2 day(s).
ID: 4 | Loan ID: 3 | Due Date: 2025-06-28 | Message: Loan is due in 3 day(s).
ID: 5 | Loan ID: 4 | Due Date: 2025-06-29 | Message: Loan is due in 4 day(s).
ID: 6 | Loan ID: 5 | Due Date: 2025-06-30 | Message: Loan is due in 5 day(s).
ID: 7 | Loan ID: 6 | Due Date: 2025-07-01 | Message: Loan is due in 6 day(s).
ID: 8 | Loan ID: 7 | Due Date: 2025-07-02 | Message: Loan is due in 7 day(s).
ID: 9 | Loan ID: 8 | Due Date: 2025-07-03 | Message: Loan is due in 8 day(s).
ID: 10 | Loan ID: 9 | Due Date: 2025-07-04 | Message: Loan is due in 9 day(s).
ID: 1 | Loan ID: 10 | Due Date: 2025-07-05 | Message: Loan is due in 10 day(s).



PL/SQL procedure successfully completed.
```

Exercise-3: Stored Procedures

For code visit

Scenario-1:

Creating procedure that process monthly interest for all savings accounts.

Results

```
User data before applying 1% interest on SAVINGS account
ID: 1 | Balance: 14377.18 | Account Type: CURRENT
ID: 2 | Balance: 10961.69 | Account Type: CURRENT
ID: 3 | Balance: 12908.41 | Account Type: CURRENT
ID: 4 | Balance: 13735.98 | Account Type: SAVINGS
ID: 5 | Balance: 9973.62 | Account Type: CURRENT
ID: 6 | Balance: 19094.71 | Account Type: SAVINGS
ID: 7 | Balance: 14884.89 | Account Type: CURRENT
ID: 8 | Balance: 12710.98 | Account Type: SAVINGS
ID: 9 | Balance: 9755.98 | Account Type: SAVINGS
ID: 10 | Balance: 18667.76 | Account Type: SAVINGS

Number of records updated are 5
User data after applying 1% interest on SAVINGS account
ID: 1 | Balance: 14377.18 | Account Type: CURRENT
ID: 2 | Balance: 10961.69 | Account Type: CURRENT
ID: 3 | Balance: 12908.41 | Account Type: CURRENT
ID: 4 | Balance: 13873.34 | Account Type: SAVINGS
ID: 5 | Balance: 9973.62 | Account Type: CURRENT
ID: 6 | Balance: 19285.66 | Account Type: SAVINGS
ID: 7 | Balance: 14884.89 | Account Type: CURRENT
ID: 8 | Balance: 12838.09 | Account Type: SAVINGS
ID: 9 | Balance: 9853.54 | Account Type: SAVINGS
ID: 10 | Balance: 18854.44 | Account Type: SAVINGS


PL/SQL procedure successfully completed.
```

Scenario-2:

Create a procedure to add bonus to employees salary which takes bonus percentage and employee department as input parameters

Results

```
Employees data before adding to bonus
ID: 2 | Dept: Dept-1 | Salary: 15413.96
ID: 4 | Dept: Dept-1 | Salary: 38111.37
ID: 6 | Dept: Dept-1 | Salary: 17390.19
ID: 8 | Dept: Dept-1 | Salary: 21059.96
ID: 10 | Dept: Dept-1 | Salary: 36206.52

Bonus applied successfully to department: Dept-1
Employees data after adding to bonus
ID: 2 | Dept: Dept-1 | Salary: 16955.36
ID: 4 | Dept: Dept-1 | Salary: 41922.51
ID: 6 | Dept: Dept-1 | Salary: 19129.21
ID: 8 | Dept: Dept-1 | Salary: 23165.96
ID: 10 | Dept: Dept-1 | Salary: 39827.17



PL/SQL procedure successfully completed.
```

Scenario-3:

Transfer funds from one customer to another, by checking sender and receiver existence and sender balance limit.

Results

```
Balance before transaction
Sender -> ID: 1 | Balance: 15377.18
Receiver -> ID: 2 | Balance: 9961.69
Transfer of 1000 from user 1 to user 2 successful.
Balance after transaction
Sender -> ID: 1 | Balance: 14377.18
Receiver -> ID: 2 | Balance: 10961.69


PL/SQL procedure successfully completed.
```