Exercise 1 – Control Structures (PL/SQL)

# 1. Objective

To develop PL/SQL blocks that use control structures (like loops and conditionals) to perform customer-related banking operations.

# 2. Problem Statement / Scenario

You are developing backend logic for a banking system. Three main tasks are required:  
- Apply interest rate discounts for senior citizens.  
- Set VIP status for high-balance customers.  
- Send loan due reminders to eligible customers.

# 3. Approach / Steps

## 3.1 Understanding Control Structures in PL/SQL

PL/SQL supports control flow via IF, LOOP, and FOR constructs, which are ideal for processing sets of data from queries.

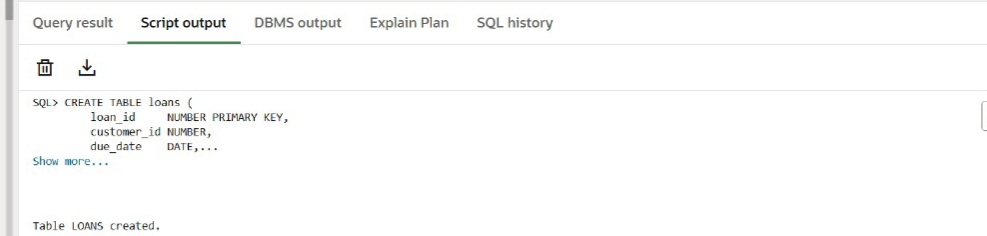
## 3.2 Setup

Create customers and loans tables and insert test data.

Customer Table Creation:



Loans Table Creation:



## 3.3 Implementation

We use FOR loops and conditionals to perform the required operations in PL/SQL.

# 4. Code

Scenario 1: Apply 1% Discount to Loan Interest Rates for Age > 60

BEGIN  
 FOR rec IN (SELECT customer\_id FROM customers WHERE age > 60) LOOP  
 UPDATE customers  
 SET loan\_interest\_rate = loan\_interest\_rate - 1  
 WHERE customer\_id = rec.customer\_id;  
 END LOOP;  
 COMMIT;  
END;

Scenario 2: Set VIP Flag for Customers with Balance > 10000

BEGIN  
 FOR rec IN (SELECT customer\_id FROM customers WHERE balance > 10000) LOOP  
 UPDATE customers  
 SET isvip = 'TRUE'  
 WHERE customer\_id = rec.customer\_id;  
 END LOOP;  
 COMMIT;  
END;

Scenario 3: Loan Due Reminders Within 30 Days

BEGIN  
 FOR rec IN (  
 SELECT customer\_id, loan\_id, due\_date  
 FROM loans  
 WHERE due\_date BETWEEN SYSDATE AND SYSDATE + 30  
 ) LOOP  
 DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || rec.loan\_id ||   
 ' for Customer ID ' || rec.customer\_id ||   
 ' is due on ' || TO\_CHAR(rec.due\_date, 'DD-MON-YYYY'));  
 END LOOP;  
END;

# 5. Output Verification

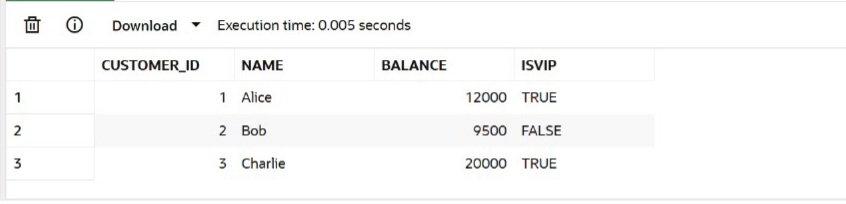
PL/SQL Block Execution Message:



Scenario 1 Result – Updated Interest Rates:



Scenario 2 Result – VIP Flag Updated:



Scenario 3 Result – Reminder Messages:



# 6. Conclusion

PL/SQL control structures allow efficient processing of business logic such as conditional updates and looping through records. In this exercise, we implemented real banking scenarios using FOR loops and conditionals with DBMS output for dynamic messages.