# PL/SQL Exercises – Bank Scenarios

Date: 29 June 2025

## Exercise 1: Control Structures

### Scenario 1: Apply 1% Interest Discount to Customers Above 60

PL/SQL Block:  
BEGIN  
 FOR cust IN (SELECT CustomerID, Age FROM Customers) LOOP  
 IF cust.Age > 60 THEN  
 UPDATE Loans  
 SET InterestRate = InterestRate - 1  
 WHERE CustomerID = cust.CustomerID;  
 END IF;  
 END LOOP;  
 COMMIT;  
END;

✅ Output: Interest rates for all customers above age 60 are decreased by 1%.

### Scenario 2: Set VIP Flag for Customers with Balance > $10,000

PL/SQL Block:  
BEGIN  
 FOR cust IN (SELECT CustomerID, Balance FROM Customers) LOOP  
 IF cust.Balance > 10000 THEN  
 UPDATE Customers  
 SET IsVIP = 'TRUE'  
 WHERE CustomerID = cust.CustomerID;  
 END IF;  
 END LOOP;  
 COMMIT;  
END;

✅ Output: IsVIP flag set to TRUE for customers with balance over $10,000.

### Scenario 3: Print Reminders for Loans Due in Next 30 Days

PL/SQL Block:  
BEGIN  
 FOR loan IN (  
 SELECT LoanID, CustomerID, DueDate  
 FROM Loans  
 WHERE DueDate BETWEEN SYSDATE AND SYSDATE + 30  
 ) LOOP  
 DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || loan.LoanID ||   
 ' for Customer ' || loan.CustomerID ||   
 ' is due on ' || TO\_CHAR(loan.DueDate, 'DD-MON-YYYY'));  
 END LOOP;  
END;

✅ Output: Reminder messages are printed for loans due in the next 30 days.

## Exercise 3: Stored Procedures

### Scenario 1: ProcessMonthlyInterest – Apply 1% Interest to Savings Accounts

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS  
BEGIN  
 UPDATE SavingsAccounts  
 SET Balance = Balance + (Balance \* 0.01);  
 COMMIT;  
END;

✅ Output: Balances of all savings accounts updated with 1% interest.

### Scenario 2: UpdateEmployeeBonus – Add Bonus to Employee Salaries in a Department

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (  
 p\_DepartmentID IN NUMBER,  
 p\_BonusPercent IN NUMBER  
) IS  
BEGIN  
 UPDATE Employees  
 SET Salary = Salary + (Salary \* p\_BonusPercent / 100)  
 WHERE DepartmentID = p\_DepartmentID;  
 COMMIT;  
END;

✅ Output: Salaries of employees in the specified department increased by the bonus percentage.

### Scenario 3: TransferFunds – Transfer Between Accounts Safely

CREATE OR REPLACE PROCEDURE TransferFunds (  
 p\_SourceAccountID IN NUMBER,  
 p\_TargetAccountID IN NUMBER,  
 p\_Amount IN NUMBER  
) IS  
 v\_Balance NUMBER;  
BEGIN  
 SELECT Balance INTO v\_Balance FROM Accounts WHERE AccountID = p\_SourceAccountID FOR UPDATE;  
  
 IF v\_Balance < p\_Amount THEN  
 RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance in source account.');  
 END IF;  
  
 UPDATE Accounts  
 SET Balance = Balance - p\_Amount  
 WHERE AccountID = p\_SourceAccountID;  
  
 UPDATE Accounts  
 SET Balance = Balance + p\_Amount  
 WHERE AccountID = p\_TargetAccountID;  
  
 COMMIT;  
END;

✅ Output: Transfers funds after checking for sufficient balance. Raises an error if balance is insufficient.