# **Skill – PL/SQL Programming**

## **Exercise 1: Control Structures**

### **Scenario 1: Discount for Senior Customers**

The bank wants to apply a 1% discount to loan interest rates for customers above 60 years old.

Code:

BEGIN  
 FOR cust IN (SELECT id, interest\_rate FROM Customers WHERE age > 60) LOOP  
 UPDATE Customers  
 SET interest\_rate = interest\_rate - (interest\_rate \* 0.01)  
 WHERE id = cust.id;  
 END LOOP;  
 COMMIT;  
END;  
/

### **Scenario 2: Promote VIP Customers**

The bank promotes customers to VIP if their balance exceeds $10,000.

Code:

BEGIN  
 FOR cust IN (SELECT id FROM Customers WHERE balance > 10000) LOOP  
 UPDATE Customers  
 SET IsVIP = 'Y'  
 WHERE id = cust.id;  
 END LOOP;  
 COMMIT;  
END;  
/

### **Scenario 3: Loan Due Reminder**

The bank sends reminders to customers whose loans are due in the next 30 days.

Code:

DECLARE  
 v\_name Customers.name%TYPE;  
BEGIN  
 FOR loan\_rec IN (  
 SELECT l.loan\_id, l.customer\_id, c.name  
 FROM Loans l  
 JOIN Customers c ON l.customer\_id = c.id  
 WHERE l.due\_date BETWEEN SYSDATE AND SYSDATE + 30  
 ) LOOP  
 DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || loan\_rec.loan\_id ||  
 ' for customer ' || loan\_rec.name ||  
 ' is due on ' || TO\_CHAR(SYSDATE + 30, 'DD-MON-YYYY'));  
 END LOOP;  
END;  
/

## **Exercise 3: Stored Procedures**

### Scenario 1: Monthly Interest for Savings Accounts

Procedure to calculate and update balance of savings accounts with 1% interest.

Code:

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

### **Scenario 2: Employee Bonus Update**

Procedure to update salary by applying a bonus percentage to a given department.

Code:

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(  
 dept\_id IN NUMBER,  
 bonus\_pct IN NUMBER  
) IS  
BEGIN  
 UPDATE Employees  
 SET salary = salary + (salary \* bonus\_pct / 100)  
 WHERE department\_id = dept\_id;  
 COMMIT;  
END;  
/

Usage:  
EXEC UpdateEmployeeBonus(101, 10);

### **Scenario 3: Transfer Funds Between Accounts**

Procedure to transfer amount between two accounts after verifying sufficient balance.

Code:

CREATE OR REPLACE PROCEDURE TransferFunds(  
 from\_account IN NUMBER,  
 to\_account IN NUMBER,  
 amount IN NUMBER  
) IS  
 from\_balance NUMBER;  
BEGIN  
 SELECT balance INTO from\_balance FROM Accounts  
 WHERE account\_id = from\_account FOR UPDATE;  
  
 IF from\_balance >= amount THEN  
 UPDATE Accounts SET balance = balance - amount WHERE account\_id = from\_account;  
 UPDATE Accounts SET balance = balance + amount WHERE account\_id = to\_account;  
 COMMIT;  
 ELSE  
 RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance in source account.');  
 END IF;  
END;  
/

Usage:  
EXEC TransferFunds(1001, 1002, 500);

#### Output:

Customers above 60 have had their loan interest rates reduced by 1%.  
E.g., if a customer had 10% interest, it is now 9.9%.

#### Output:

All customers with balance greater than $10,000 now have IsVIP = 'Y'.

#### Output:

Reminder: Loan ID 2001 for customer John Doe is due on 23-JUL-2025  
Reminder: Loan ID 2002 for customer Alice Smith is due on 20-JUL-2025

#### Output:

Each savings account balance increased by 1%.  
For example, ₹10,000 becomes ₹10,100.

#### Output:

Employee salaries in department 101 updated by the specified bonus percentage.  
E.g., ₹40,000 becomes ₹44,000 for a 10% bonus.

#### Output:

Funds transferred successfully if sufficient balance existed.  
If not, error: ORA-20001: Insufficient balance in source account.