**PL/SQL PROGRAMMING -Hands On**

**Exercise 1: Control Structures**

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

* + **Question:** 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.

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

* + **Question:** 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.

**Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days.

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

**SOLUTION:**  
**SCENARIO 1: APPLY 1% DISCOUNT FOR CUSTOMERS ABOVE 60**

BEGIN

FOR customer\_rec IN (SELECT CustomerID, Age, InterestRate FROM Customers) LOOP

IF customer\_rec.Age > 60 THEN

UPDATE Customers

SET InterestRate = InterestRate - (InterestRate \* 0.01)

WHERE CustomerID = customer\_rec.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

/  
**SCENARIO 2: PROMOTE CUSTOMERS TO VIP BASED ON BALANCE**

BEGIN

FOR customer\_rec IN (SELECT CustomerID, Balance FROM Customers) LOOP

IF customer\_rec.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = customer\_rec.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

/  
**SCENARIO 3: REMIND CUSTOMERS WHOSE LOANS ARE DUE WITHIN 30 DAYS**

BEGIN

FOR loan\_rec IN (

SELECT LoanID, CustomerID, DueDate

FROM Loans

WHERE DueDate <= SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Customer ' || loan\_rec.CustomerID ||

' has a loan (Loan ID: ' || loan\_rec.LoanID ||

') due on ' || TO\_CHAR(loan\_rec.DueDate, 'DD-Mon-YYYY'));

END LOOP;

END;

/

**EXERCISE 3: STORED PROCEDURES**

**Scenario 1:** The bank needs to process monthly interest for all savings accounts.

* + **Question:** 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.

**Scenario 2:** The bank wants to implement a bonus scheme for employees based on their performance.

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

**Scenario 3:** Customers should be able to transfer funds between their accounts.

* + **Question:** 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.

SOLUTION:  
**SCENARIO 1: PROCESS MONTHLY INTEREST**

Apply 1% monthly interest to all savings accounts.

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR acc IN (SELECT AccountID, Balance FROM Accounts WHERE AccountType = 'SAVINGS') LOOP

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountID = acc.AccountID;

END LOOP;

COMMIT;

END;

/  
**SCENARIO 2: UPDATE EMPLOYEE BONUS**

Add bonus to salaries of employees in a specific department.

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 DepartmentID = dept\_id;

COMMIT;

END;

/  
**SCENARIO 3: TRANSFER FUNDS**

Transfer money from one account to another after checking balance.

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 AccountID = from\_account

FOR UPDATE;

IF from\_balance >= amount THEN

UPDATE Accounts

SET Balance = Balance - amount

WHERE AccountID = from\_account;

UPDATE Accounts

SET Balance = Balance + amount

WHERE AccountID = to\_account;

COMMIT;

ELSE

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in source account.');

END IF;

END;

/  
  
EXEC ProcessMonthlyInterest;

EXEC UpdateEmployeeBonus(10, 15); -- 15% bonus to department 10

EXEC TransferFunds(101, 202, 5000);