**PL/SQL programming**

**Exercise 1: Control Structures**

**Scenario 1: Apply 1% Interest Discount for Age > 60**

**Code:**BEGIN

FOR cust\_rec IN (SELECT CustomerID, Age, LoanInterestRate FROM CUSTOMERS) LOOP

IF cust\_rec.Age > 60 THEN

UPDATE CUSTOMERS

SET LoanInterestRate = LoanInterestRate - 1

WHERE CustomerID = cust\_rec.CustomerID;

DBMS\_OUTPUT.PUT\_LINE('Applied discount for Customer ID: ' || cust\_rec.CustomerID);

END IF;

END LOOP;

COMMIT;

END;

**Scenario 2: Promote Customers to VIP if Balance > $10,000**

**Code:**

BEGIN

FOR cust\_rec IN (SELECT CustomerID, Balance FROM CUSTOMERS) LOOP

IF cust\_rec.Balance > 10000 THEN

UPDATE CUSTOMERS

SET IsVIP = 'TRUE'

WHERE CustomerID = cust\_rec.CustomerID;

DBMS\_OUTPUT.PUT\_LINE('Customer ID ' || cust\_rec.CustomerID || ' promoted to VIP.');

END IF;

END LOOP;

COMMIT;

END;

**Scenario 3: Send Loan Reminders (Due in 30 Days)**

BEGIN

FOR loan\_rec IN (

SELECT LoanID, CustomerID, DueDate

FROM LOANS

WHERE DueDate BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ' || loan\_rec.LoanID ||

' for Customer ' || loan\_rec.CustomerID ||

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

END LOOP;

END;  
 **Exercise 3: Stored Procedures**

**Scenario 1: Process Monthly Interest**

**Code**CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR acc IN (SELECT AccountID, Balance FROM SAVINGS\_ACCOUNTS) LOOP

UPDATE SAVINGS\_ACCOUNTS

SET Balance = Balance + (acc.Balance \* 0.01)

WHERE AccountID = acc.AccountID;

DBMS\_OUTPUT.PUT\_LINE('Interest applied for Account ID: ' || acc.AccountID);

END LOOP;

COMMIT;

END;

**Scenario 2: Update Employee Bonus**

**Code** CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_dept\_id IN NUMBER,

p\_bonus\_percent IN NUMBER

) IS

BEGIN

FOR emp IN (

SELECT EmployeeID, Salary

FROM EMPLOYEES

WHERE DepartmentID = p\_dept\_id

) LOOP

UPDATE EMPLOYEES

SET Salary = Salary + (emp.Salary \* p\_bonus\_percent / 100)

WHERE EmployeeID = emp.EmployeeID;

DBMS\_OUTPUT.PUT\_LINE('Bonus updated for Employee ID: ' || emp.EmployeeID);

END LOOP;

COMMIT;

END;

**Scenario 3: Transfer Funds Between Accounts**

**Code:**CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_account IN NUMBER,

p\_to\_account IN NUMBER,

p\_amount IN NUMBER

) IS

v\_balance NUMBER;

BEGIN

-- Check balance of source account

SELECT Balance INTO v\_balance

FROM ACCOUNTS

WHERE AccountID = p\_from\_account;

IF v\_balance >= p\_amount THEN

UPDATE ACCOUNTS

SET Balance = Balance - p\_amount

WHERE AccountID = p\_from\_account;

UPDATE ACCOUNTS

SET Balance = Balance + p\_amount

WHERE AccountID = p\_to\_account;

DBMS\_OUTPUT.PUT\_LINE('Transferred $' || p\_amount ||

' from Account ' || p\_from\_account ||

' to Account ' || p\_to\_account);

COMMIT;

ELSE

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

END IF;

END;