**Module 2: PL/SQL Programming**

**Exercise 1: Control Structures**

**Scenario 1: Discount for Customers above 60 Years old**

**Code:**

DECLARE

v\_customer\_id NUMBER;

v\_age NUMBER;

v\_loan\_interest\_rate NUMBER;

BEGIN

FOR cust\_rec IN (SELECT customer\_id, age, loan\_interest\_rate FROM Customers) LOOP

v\_customer\_id := cust\_rec.customer\_id;

v\_age := cust\_rec.age;

v\_loan\_interest\_rate := cust\_rec.loan\_interest\_rate;

IF v\_age > 60 THEN

-- Apply a 1% discount to the loan interest rate

UPDATE Customers

SET loan\_interest\_rate = v\_loan\_interest\_rate \* 0.99

WHERE customer\_id = v\_customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Discount applied to customer ID: ' || v\_customer\_id);

END IF;

END LOOP;

END;

/

**OUTPUT:**

Discount applied to customer ID: 123

Discount applied to customer ID: 456

**Scenario 2: Promote Customers to VIP Status**

**Code:**

DECLARE

v\_customer\_id NUMBER;

v\_balance NUMBER;

BEGIN

FOR cust\_rec IN (SELECT customer\_id, balance FROM Customers) LOOP

v\_customer\_id := cust\_rec.customer\_id;

v\_balance := cust\_rec.balance;

IF v\_balance > 10000 THEN

-- Set the IsVIP flag to TRUE

UPDATE Customers

SET IsVIP = TRUE

WHERE customer\_id = v\_customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Customer ID ' || v\_customer\_id || ' has been promoted to VIP status.');

END IF;

END LOOP;

END;

/

**OUTPUT:**

Customer ID 123 has been promoted to VIP status.

Customer ID 789 has been promoted to VIP status.

**Scenario 3: Send Reminders for Loans Due Within 30 Days**

**Code:**

DECLARE

v\_customer\_id NUMBER;

v\_loan\_id NUMBER;

v\_due\_date DATE;

BEGIN

FOR loan\_rec IN (SELECT customer\_id, loan\_id, due\_date FROM Loans WHERE due\_date BETWEEN SYSDATE AND ADD\_MONTHS(SYSDATE, 1)) LOOP

v\_customer\_id := loan\_rec.customer\_id;

v\_loan\_id := loan\_rec.loan\_id;

v\_due\_date := loan\_rec.due\_date;

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || v\_loan\_id || ' for customer ID ' || v\_customer\_id || ' is due on ' || v\_due\_date);

END LOOP;

END;

/

**OUTPUT:**

Reminder: Loan ID 101 for customer ID 123 is due on 2023-12-15

Reminder: Loan ID 102 for customer ID 456 is due on 2023-12-20

**Exercise 3: Stored Procedures**

**Scenario 1: Process Monthly Interest for Savings Accounts**

**Code:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

-- Update the balance of all savings accounts by applying a 1% interest rate

UPDATE SavingsAccounts

SET balance = balance \* 1.01;

DBMS\_OUTPUT.PUT\_LINE('Monthly interest processed for all savings accounts.');

END ProcessMonthlyInterest;

/

**OUTPUT:**

Monthly interest processed for all savings accounts.

**Scenario 2: Update Employee Bonus**

**Code:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_department\_id IN NUMBER,

p\_bonus\_percentage IN NUMBER

) AS

BEGIN

-- Update the salary of employees in the given department by adding the bonus percentage

UPDATE Employees

SET salary = salary \* (1 + p\_bonus\_percentage / 100)

WHERE department\_id = p\_department\_id;

DBMS\_OUTPUT.PUT\_LINE('Bonus applied to employees in department ' || p\_department\_id || '.');

END UpdateEmployeeBonus;

/

**OUTPUT:**

Bonus applied to employees in department 10.

**Scenario 3: Transfer Funds Between Accounts**

**Code:**

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_source\_account\_id IN NUMBER,

p\_destination\_account\_id IN NUMBER,

p\_amount IN NUMBER

) AS

v\_source\_balance NUMBER;

v\_destination\_balance NUMBER;

BEGIN

-- Check if the source account has sufficient balance

SELECT balance INTO v\_source\_balance

FROM Accounts

WHERE account\_id = p\_source\_account\_id;

IF v\_source\_balance < p\_amount THEN

DBMS\_OUTPUT.PUT\_LINE('Insufficient balance in source account.');

RETURN;

END IF;

-- Deduct the amount from the source account

UPDATE Accounts

SET balance = balance - p\_amount

WHERE account\_id = p\_source\_account\_id;

-- Add the amount to the destination account

UPDATE Accounts

SET balance = balance + p\_amount

WHERE account\_id = p\_destination\_account\_id;

DBMS\_OUTPUT.PUT\_LINE('Funds transferred successfully.');

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error during fund transfer: ' || SQLERRM);

END TransferFunds;

/

**OUTPUT:**

Funds transferred successfully.