**PL/SQL Programming**

**Question 1:**

**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.**

DECLARE

CURSOR c\_customers IS

SELECT customer\_id FROM Customers WHERE age > 60;

BEGIN

FOR rec IN c\_customers LOOP

UPDATE Loans

SET interest\_rate = interest\_rate - 1

WHERE customer\_id = rec.customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Applied 1% discount to customer ID: ' || rec.customer\_id);

END LOOP;

COMMIT;

END;

/

**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.**

BEGIN

FOR rec IN (SELECT customer\_id FROM Customers WHERE balance > 10000) LOOP

UPDATE Customers

SET isVIP = 'TRUE'

WHERE customer\_id = rec.customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Customer ID ' || rec.customer\_id || ' promoted to VIP.');

END LOOP;

COMMIT;

END;

/

**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.**

DECLARE

CURSOR c\_loans IS

SELECT l.loan\_id, l.customer\_id, c.name, l.due\_date

FROM Loans l

JOIN Customers c ON l.customer\_id = c.customer\_id

WHERE l.due\_date BETWEEN SYSDATE AND SYSDATE + 30;

BEGIN

FOR rec IN c\_loans LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || rec.loan\_id ||

' for customer ' || rec.name ||

' is due on ' || TO\_CHAR(rec.due\_date, 'DD-MON-YYYY'));

END LOOP;

END;

/

**Question 2:**

**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.**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

UPDATE Accounts

SET balance = balance \* 1.01

WHERE account\_type = 'SAVINGS';

DBMS\_OUTPUT.PUT\_LINE('Monthly interest of 1% applied to all savings accounts.');

COMMIT;

END;

/

**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.**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_department\_id IN NUMBER,

p\_bonus\_percentage IN NUMBER

) AS

BEGIN

UPDATE Employees

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

WHERE department\_id = p\_department\_id;

DBMS\_OUTPUT.PUT\_LINE('Bonus of ' || p\_bonus\_percentage || '% applied to department ID: ' || p\_department\_id);

COMMIT;

END;

/

**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.**

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_source\_account\_id IN NUMBER,

p\_target\_account\_id IN NUMBER,

p\_amount IN NUMBER

) AS

v\_source\_balance NUMBER;

BEGIN

-- Check source account balance

SELECT balance INTO v\_source\_balance

FROM Accounts

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

FOR UPDATE;

IF v\_source\_balance < p\_amount THEN

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

END IF;

-- Deduct from source

UPDATE Accounts

SET balance = balance - p\_amount

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

-- Credit to target

UPDATE Accounts

SET balance = balance + p\_amount

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

DBMS\_OUTPUT.PUT\_LINE('Transferred ' || p\_amount || ' from account ' || p\_source\_account\_id ||

' to account ' || p\_target\_account\_id);

COMMIT;

END;

/