Stored Procedure

# 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 IS

v\_AccountID Accounts.AccountID%TYPE;

v\_Balance Accounts.Balance%TYPE;

v\_Interest NUMBER(10, 2);

c\_InterestRate CONSTANT NUMBER := 0.01;

BEGIN

FOR r IN (SELECT AccountID, Balance FROM Accounts WHERE AccountType = 'Savings') LOOP

v\_AccountID := r.AccountID;

v\_Balance := r.Balance;

v\_Interest := v\_Balance \* c\_InterestRate;

UPDATE Accounts

SET Balance = Balance + v\_Interest,

LastModified = SYSDATE

WHERE AccountID = v\_AccountID;

END LOOP;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

RAISE;

END ProcessMonthlyInterest;

/

# 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(bonusPercentage number, empDept varchar2) IS

BEGIN

UPDATE employees

SET salary = salary + (salary \* (bonusPercentage / 100))

WHERE department = empDept;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

RAISE;

END UpdateEmployeeBonus;

/

# 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\_SourceAccountID NUMBER,

p\_DestinationAccountID NUMBER,

p\_Amount NUMBER

) IS

v\_SourceBalance Accounts.Balance%TYPE;

v\_DestinationBalance Accounts.Balance%TYPE;

insufficient\_balance EXCEPTION;

BEGIN

SELECT Balance INTO v\_SourceBalance

FROM Accounts

WHERE AccountID = p\_SourceAccountID;

IF v\_SourceBalance < p\_Amount THEN

RAISE insufficient\_balance;

END IF;

UPDATE Accounts

SET Balance = Balance - p\_Amount,

LastModified = SYSDATE

WHERE AccountID = p\_SourceAccountID;

SELECT Balance INTO v\_DestinationBalance

FROM Accounts

WHERE AccountID = p\_DestinationAccountID;

UPDATE Accounts

SET Balance = Balance + p\_Amount,

LastModified = SYSDATE

WHERE AccountID = p\_DestinationAccountID;

COMMIT;

EXCEPTION

WHEN insufficient\_balance THEN

ROLLBACK;

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

WHEN NO\_DATA\_FOUND THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: One or both of the accounts do not exist.');

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: An unexpected error occurred.');

END TransferFunds;

/