# 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.

# Answer:

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS BEGIN

-- Update the balance of all savings accounts by applying 1% interest UPDATE Accounts

SET Balance = Balance \* 1.01 WHERE AccountType = 'Savings';

-- Commit the transaction COMMIT;

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.

# Answer:

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus( p\_department IN VARCHAR2,

p\_bonus\_percentage IN NUMBER

## ) IS BEGIN

-- Update the salary of employees in the specified department UPDATE Employees

SET Salary = Salary \* (1 + p\_bonus\_percentage / 100) WHERE Department = p\_department;

-- Commit the transaction COMMIT;

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.

# Answer:

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_from\_account IN NUMBER, p\_to\_account IN NUMBER, p\_amount IN NUMBER

## ) IS

v\_from\_balance NUMBER;

BEGIN

*-- Check current balance of the from account* SELECT Balance INTO v\_from\_balance FROM Accounts

WHERE AccountID = p\_from\_account;

*-- Ensure sufficient funds*

*IF* v\_from\_balance < p\_amount *THEN*

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

*END IF*;

*-- Perform the transfer*

UPDATE Accounts

SET Balance = Balance - p\_amount WHERE AccountID = p\_from\_account;

UPDATE Accounts

SET Balance = Balance + p\_amount WHERE AccountID = p\_to\_account;

*-- Commit the transaction*

## COMMIT;

EXCEPTION

*WHEN* NO\_DATA\_FOUND *THEN*

DBMS\_OUTPUT.PUT\_LINE('One of the account IDs does not exist.'); ROLLBACK;

*WHEN* OTHERS *THEN* DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM); ROLLBACK;

END TransferFunds;

/