**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.**

CREATE OR REPLACE PROCEDURE ApplyMonthlyInterest IS

BEGIN

-- Apply 1% interest to all savings accounts

UPDATE Accounts

SET Balance = Balance \* 1.01

WHERE AccountType = 'Savings';

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Interest successfully applied to all savings accounts.');

END ApplyMonthlyInterest;

**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 ApplyDepartmentBonus (

dept\_name IN VARCHAR2,

bonus\_percent IN NUMBER

) IS

BEGIN

-- Update salaries by the given bonus percentage for a specific department

UPDATE Employees

SET Salary = Salary \* (1 + bonus\_percent / 100)

WHERE Department = dept\_name;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Bonus of ' || bonus\_percent || '% applied to department: ' || dept\_name);

END ApplyDepartmentBonus;

**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 ExecuteFundTransfer (

sender\_account\_id IN NUMBER,

receiver\_account\_id IN NUMBER,

transfer\_amount IN NUMBER

) IS

sender\_balance NUMBER;

BEGIN

-- Retrieve the balance of the sender's account

SELECT Balance INTO sender\_balance

FROM Accounts

WHERE AccountID = sender\_account\_id;

-- Check for sufficient funds

IF sender\_balance < transfer\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Transfer failed: Insufficient funds in account ' || sender\_account\_id);

END IF;

-- Perform the transfer

BEGIN

UPDATE Accounts

SET Balance = Balance - transfer\_amount

WHERE AccountID = sender\_account\_id;

UPDATE Accounts

SET Balance = Balance + transfer\_amount

WHERE AccountID = receiver\_account\_id;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Successfully transferred ' || transfer\_amount ||

' from Account ' || sender\_account\_id ||

' to Account ' || receiver\_account\_id);

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Transfer failed. Reason: ' || SQLERRM);

END;

END ExecuteFundTransfer;