**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.

**Ans:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

UPDATE Accounts

SET Balance = Balance \* 1.01

WHERE AccountType = 'Savings';

COMMIT;

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

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.

Ans:

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

emp\_department IN Employees.Department%TYPE,

emp\_bonusPercentage IN NUMBER

) AS

BEGIN

UPDATE Employees

SET Salary = Salary \* (1 + emp\_bonusPercentage / 100)

WHERE Department = emp\_department;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Employee bonuses updated for the department of ' || emp\_department);

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.

**Ans:**

CREATE OR REPLACE PROCEDURE TransferFunds (

fromAccountID IN Accounts.AccountID%TYPE,

toAccountID IN Accounts.AccountID%TYPE,

amount IN NUMBER

) AS

fromBalance Accounts.Balance%TYPE;

toBalance Accounts.Balance%TYPE;

BEGIN

SELECT Balance INTO fromBalance

FROM Accounts

WHERE AccountID = fromAccountID

FOR UPDATE;

IF fromBalance < amount THEN

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

END IF;

SELECT Balance INTO toBalance

FROM Accounts

WHERE AccountID = toAccountID

FOR UPDATE;

UPDATE Accounts

SET Balance = fromBalance - amount

WHERE AccountID = fromAccountID;

UPDATE Accounts

SET Balance = toBalance + amount

WHERE AccountID = toAccountID;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Funds transferred successfully from the account no ' || fromAccountID || ' to the account no ' || toAccountID);

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

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

END TransferFunds;

/