**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 PROCEDURE ProcessMonthlyInterest

IS

BEGIN

*-- Update the balance for all savings accounts*

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01),

LastModified = SYSDATE

WHERE AccountType = 'Savings';

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 PROCEDURE UpdateEmployeeBonus (

p\_Department IN Employees.Department%TYPE,

p\_BonusPercentage IN NUMBER

)

IS

BEGIN

UPDATE Employees

SET Salary = Salary + (Salary \* p\_BonusPercentage / 100)

WHERE Department = p\_Department;

END UpdateEmployeeBonus;

/

Example: call the procedure to update the salary of ‘Sales’ department.

BEGIN

UpdateEmployeeBonus('Sales', 10);

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 PROCEDURE TransferFunds (

p\_SourceAccountID IN Accounts.AccountID%TYPE,

p\_TargetAccountID IN Accounts.AccountID%TYPE,

p\_Amount IN NUMBER

)

IS

v\_SourceBalance NUMBER;

BEGIN

*-- Check the balance of the source account*

SELECT Balance INTO v\_SourceBalance

FROM Accounts

WHERE AccountID = p\_SourceAccountID;

*-- Check if the source account has sufficient balance*

IF v\_SourceBalance < p\_Amount THEN

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

END IF;

*-- Deduct the amount from the source account*

UPDATE Accounts

SET Balance = Balance - p\_Amount,

LastModified = SYSDATE

WHERE AccountID = p\_SourceAccountID;

*-- Add the amount to the target account*

UPDATE Accounts

SET Balance = Balance + p\_Amount,

LastModified = SYSDATE

WHERE AccountID = p\_TargetAccountID;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

*-- Handle the case where an account does not exist*

RAISE\_APPLICATION\_ERROR(-20002, 'One or both accounts do not exist.');

WHEN OTHERS THEN

*-- Log any other errors*

INSERT INTO ErrorLog (ErrorMessage, ErrorDate)

VALUES (SQLERRM, SYSDATE);

*-- Raise the exception again to propagate it*

RAISE;

END TransferFunds;

/

In this procedure:

* We first check the balance of the source account and ensure it has sufficient funds for the transfer.
* If the source account has sufficient balance, we proceed to deduct the specified amount from the source account and add it to the target account.
* If the source account does not have sufficient balance, an error is raised.
* We handle the case where an account does not exist using the NO\_DATA\_FOUND exception.
* Any other errors are logged in the ErrorLog table and the exception is propagated.

Example:

BEGIN

TransferFunds(101, 202, 500); -- This will transfer 500 Rs from account 101 to account 202.

END;

/