**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.

->**Solution:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest

IS

BEGIN

FOR account\_rec IN (SELECT AccountID, Balance FROM Accounts WHERE AccountType = 'Savings')

LOOP

UPDATE Accounts

SET Balance = Balance \* 1.01

WHERE AccountID = account\_rec.AccountID;

END LOOP;

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.

->**Solution:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_Department IN VARCHAR2,

p\_BonusPercentage IN NUMBER

)

IS

BEGIN

UPDATE Employees

SET Salary = Salary \* (1 + p\_BonusPercentage / 100)

WHERE Department = p\_Department;

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.

->**Solution:**

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_FromAccountID IN NUMBER,

p\_ToAccountID IN NUMBER,

p\_Amount IN NUMBER

)

IS

InsufficientFunds EXCEPTION;

PRAGMA EXCEPTION\_INIT(InsufficientFunds, -20001);

v\_FromAccountBalance NUMBER;

BEGIN

SELECT Balance INTO v\_FromAccountBalance

FROM Accounts

WHERE AccountID = p\_FromAccountID

FOR UPDATE;

IF v\_FromAccountBalance < p\_Amount THEN

RAISE InsufficientFunds;

END IF;

UPDATE Accounts

SET Balance = Balance - p\_Amount

WHERE AccountID = p\_FromAccountID;

UPDATE Accounts

SET Balance = Balance + p\_Amount

WHERE AccountID = p\_ToAccountID;

COMMIT;

EXCEPTION

WHEN InsufficientFunds THEN

ROLLBACK;

INSERT INTO ErrorLog (ErrorMessage, ErrorDate)

VALUES ('Insufficient funds for transfer from AccountID: ' || p\_FromAccountID, SYSDATE);

WHEN OTHERS THEN

ROLLBACK;

INSERT INTO ErrorLog (ErrorMessage, ErrorDate)

VALUES ('Error during transfer from AccountID: ' || p\_FromAccountID || ' to AccountID: ' || p\_ToAccountID || '. ' || SQLERRM, SYSDATE);

END TransferFunds;

/