**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.

**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.

**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.

**SCENARIO 1:**

DELIMITER $$

CREATE PROCEDURE CalculateMonthlyInterest()

BEGIN

DECLARE finished INT DEFAULT 0;

DECLARE acc\_id INT;

DECLARE acc\_balance DECIMAL(10, 2);

DECLARE interest\_rate CONSTANT DECIMAL(10, 2) DEFAULT 0.01;

-- Cursor to fetch all savings accounts

DECLARE account\_cursor CURSOR FOR

SELECT AccountID, Balance FROM Accounts WHERE AccountType = 'Savings';

-- Handler for cursor

DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished = 1;

OPEN account\_cursor;

process\_loop: LOOP

FETCH account\_cursor INTO acc\_id, acc\_balance;

IF finished THEN

LEAVE process\_loop;

END IF;

-- Apply interest to the balance

SET acc\_balance = acc\_balance \* (1 + interest\_rate);

-- Update the balance in the account

UPDATE Accounts

SET Balance = acc\_balance

WHERE AccountID = acc\_id;

END LOOP;

CLOSE account\_cursor;

END$$

DELIMITER ;

**Scenario 2:**

DELIMITER $$

CREATE PROCEDURE ApplyEmployeeBonus(

IN department\_id INT,

IN bonus\_pct DECIMAL(5, 2)

)

BEGIN

-- Add bonus to employee salaries in the specified department

UPDATE Employees

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

WHERE Department = department\_id;

END$$

DELIMITER ;

**Scenario 3:**

DELIMITER $$

CREATE PROCEDURE TransferAccountFunds(

IN from\_account\_id INT,

IN to\_account\_id INT,

IN transfer\_amount DECIMAL(10, 2)

)

BEGIN

DECLARE from\_balance DECIMAL(10, 2);

-- Retrieve the balance of the source account

SELECT Balance INTO from\_balance FROM Accounts WHERE AccountID = from\_account\_id;

-- Ensure the source account has sufficient funds

IF from\_balance >= transfer\_amount THEN

-- Deduct the transfer amount from the source account

UPDATE Accounts

SET Balance = Balance - transfer\_amount

WHERE AccountID = from\_account\_id;

-- Credit the transfer amount to the target account

UPDATE Accounts

SET Balance = Balance + transfer\_amount

WHERE AccountID = to\_account\_id;

ELSE

-- Raise an error for insufficient funds

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Insufficient funds in the source account';

END IF;

END$$

DELIMITER ;