**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.

**Solutions:**

**Scenario 1: Process Monthly Interest**

DELIMITER //

CREATE PROCEDURE ProcessMonthlyInterest()

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE account\_id INT;

DECLARE current\_balance DECIMAL(10,2);

DECLARE interest\_rate DECIMAL(3,2) DEFAULT 0.01;

DECLARE account\_cursor CURSOR FOR

SELECT account\_id, balance FROM savings\_accounts;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN account\_cursor;

read\_loop: LOOP

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

IF done THEN

LEAVE read\_loop;

END IF;

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

UPDATE savings\_accounts SET balance = current\_balance WHERE account\_id = account\_id;

END LOOP;

CLOSE account\_cursor;

END //

DELIMITER ;

**Scenario 2: Update Employee Bonus**

DELIMITER //

CREATE PROCEDURE UpdateEmployeeBonus(IN dept\_id INT, IN bonus\_percentage DECIMAL(4,2))

BEGIN

UPDATE employees

SET salary = salary \* (1 + bonus\_percentage / 100)

WHERE department\_id = dept\_id;

END //

DELIMITER ;

**Scenario 3: Transfer Funds**

DELIMITER //

CREATE PROCEDURE TransferFunds(IN from\_account\_id INT, IN to\_account\_id INT, IN transfer\_amount DECIMAL(10,2))

BEGIN

DECLARE source\_balance DECIMAL(10,2);

SELECT balance INTO source\_balance FROM accounts WHERE account\_id = from\_account\_id;

IF source\_balance >= transfer\_amount THEN

UPDATE accounts

SET balance = balance - transfer\_amount

WHERE account\_id = from\_account\_id;

UPDATE accounts

SET balance = balance + transfer\_amount

WHERE account\_id = to\_account\_id;

ELSE

SIGNAL SQLSTATE '45000'

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

END IF;

END //

DELIMITER ;