**EXERCISE 3 : STORED PROCEDURES**

DELIMITER //

-- ==========================================

-- Scenario 1: ProcessMonthlyInterest

-- ==========================================

CREATE PROCEDURE ProcessMonthlyInterest()

BEGIN

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountType = 'Savings';

SELECT 'Monthly interest of 1% applied to all Savings accounts.' AS Message;

END;

//

-- ==========================================

-- Scenario 2: UpdateEmployeeBonus

-- ==========================================

CREATE PROCEDURE UpdateEmployeeBonus(

IN dept\_name VARCHAR(50),

IN bonus\_percent DECIMAL(5,2)

)

BEGIN

UPDATE Employees

SET Salary = Salary + (Salary \* bonus\_percent / 100)

WHERE Department = dept\_name;

SELECT CONCAT('Bonus of ', bonus\_percent, '% applied to ', dept\_name, ' department.') AS Message;

END;

//

-- ==========================================

-- Scenario 3: TransferFunds

-- ==========================================

CREATE PROCEDURE TransferFunds(

IN from\_acc INT,

IN to\_acc INT,

IN amt DECIMAL(10,2)

)

BEGIN

DECLARE from\_bal DECIMAL(10,2);

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

ROLLBACK;

SELECT 'Transfer failed. Transaction rolled back due to an error.' AS ErrorMessage;

END;

START TRANSACTION;

-- Get source account balance

SELECT Balance INTO from\_bal

FROM Accounts

WHERE AccountID = from\_acc

FOR UPDATE;

-- Check if sufficient balance

IF from\_bal < amt THEN

SIGNAL SQLSTATE '45000'

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

END IF;

-- Perform the transfer

UPDATE Accounts SET Balance = Balance - amt WHERE AccountID = from\_acc;

UPDATE Accounts SET Balance = Balance + amt WHERE AccountID = to\_acc;

COMMIT;

SELECT 'Funds transferred successfully.' AS Message;

END;

//

DELIMITER ;

**OUTPUT:**

USE bankdb;

SELECT \* FROM Accounts WHERE AccountType = 'Savings';

SELECT \* FROM Employees;

SELECT \* FROM Accounts;





