**Exercise 3: Stored Procedures**

**CODE**

**Create tables**

-- Savings accounts

CREATE TABLE SavingsAccounts (

account\_id INT PRIMARY KEY,

customer\_id INT,

balance DECIMAL(12,2)

);

-- Employees

CREATE TABLE Employees (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR(50),

dept\_id INT,

salary DECIMAL(12,2)

);

-- Generic accounts for transfers

CREATE TABLE BankAccounts (

account\_id INT PRIMARY KEY,

customer\_id INT,

balance DECIMAL(12,2)

);

**Insert Sample Data**

INSERT INTO SavingsAccounts VALUES (101, 1, 1000.00), (102, 2, 2500.00);

INSERT INTO Employees VALUES (1,'Alice', 10, 60000.00),

(2,'Bob', 10, 55000.00),

(3,'Cara', 20, 50000.00);

INSERT INTO BankAccounts VALUES (201, 1, 5000.00), (202, 1, 2000.00);

**Scenario 1 ─ Stored procedure to process monthly interest**

GO

CREATE OR ALTER PROCEDURE ProcessMonthlyInterest

AS

BEGIN

SET NOCOUNT ON;

UPDATE SavingsAccounts

SET balance = balance \* 1.01; -- +1 %

PRINT 'Scenario-1 Monthly interest applied to all savings accounts.';

END;

GO

EXEC ProcessMonthlyInterest;

PRINT '--- Savings account balances after interest ---';

SELECT \* FROM SavingsAccounts;

**Scenario 2 ─ Stored procedure to apply bonus to a department**

GO

CREATE OR ALTER PROCEDURE UpdateEmployeeBonus

@DeptId INT,

@BonusPct DECIMAL(5,4) -- 0.0500 = 5 %

AS

BEGIN

SET NOCOUNT ON;

UPDATE Employees

SET salary = salary \* (1 + @BonusPct)

WHERE dept\_id = @DeptId;

PRINT CONCAT('Scenario - 2 Bonus of ', @BonusPct\*100, '% applied to department ', @DeptId);

END;

GO

EXEC UpdateEmployeeBonus @DeptId = 10, @BonusPct = 0.10; -- 10 % bonus

PRINT '--- Employee salaries after bonus ---';

SELECT \* FROM Employees;

**Scenario 3 ─ Stored procedure for fund transfer**

GO

CREATE OR ALTER PROCEDURE TransferFunds

@SrcAcct INT,

@DestAcct INT,

@Amount DECIMAL(12,2)

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRAN;

DECLARE @SrcBal DECIMAL(12,2);

SELECT @SrcBal = balance FROM BankAccounts WHERE account\_id = @SrcAcct;

IF @SrcBal IS NULL

BEGIN

ROLLBACK;

RAISERROR ('Source account not found.', 16, 1);

RETURN;

END

IF @SrcBal < @Amount

BEGIN

ROLLBACK;

RAISERROR ('Insufficient funds in source account.', 16, 1);

RETURN;

END

UPDATE BankAccounts SET balance = balance - @Amount WHERE account\_id = @SrcAcct;

UPDATE BankAccounts SET balance = balance + @Amount WHERE account\_id = @DestAcct;

COMMIT;

PRINT CONCAT('Scenario-3 Transferred $', @Amount, ' from account ', @SrcAcct, ' to ', @DestAcct);

END;

GO

PRINT '--- Bank account balances after transfer ---';

SELECT \* FROM BankAccounts;

**OUTPUT**

