**Exercise 3: Stored Procedures**

For this task we need a pre-defined data for which we use file named as ‘Data.sql’.

**Output-**

Acc. ID Cus. ID Account\_type Balance

1 101 savings 10000

2 102 savings 15000

3 103 current 20000

EMP ID Name Department Salary

1 Rahul HR 30000

2 Anita IT 45000

3 Ravi IT 50000

**Scenario 1: Monthly Interest Processing for Savings Accounts**

**Problem Statement:**  
The bank needs to **apply a 1% interest** to all savings account balances on a monthly basis.

**Procedure Name:** ProcessMonthlyInterest

**Approach:**

* Use a cursor to iterate over all savings accounts.
* Update each account balance by adding 1% of the current balance.

**Output-**

Acc. ID Cus. ID Account\_type Balance

1 101 savings 10100

2 102 savings 15150

3 103 current 20000

**Scenario 2: Employee Bonus Based on Department Performance**

**Problem Statement:**  
Update the **salary** of all employees in a specific department by a **bonus percentage** passed as a parameter.

**Procedure Name:** UpdateEmployeeBonus

**Approach:**

* Accept department ID and bonus percentage as input.
* Update employee salaries accordingly.

**Example Call:**

EXEC UpdateEmployeeBonus(101, 10);

**Output-**

EMP ID Name Department Salary

1 Rahul HR 30000

2 Anita IT 49500

3 Ravi IT 55000

**Scenario 3: Fund Transfer Between Customer Accounts**

**Problem Statement:**  
Implement a fund transfer mechanism ensuring the **source account has enough balance** before debiting.

**Procedure Name:** TransferFunds

**Approach:**

* Check the balance of the source account.
* Deduct the amount from the source account if sufficient.
* Credit the amount to the target account.
* Handle insufficient balance error.

**Example Call:**

EXEC TransferFunds(1001, 1002, 500);

**Output-**

Acc. ID Cus. ID Account\_type Balance

1 101 savings 9500

2 102 savings 15500

3 103 current 20000