|  |
| --- |
| **National University of Computer and Emerging Sciences** |
| Lab Manual 6  “Stored Procedures” |
|  |
| Database Systems Lab |
| Fall 2023 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

**Table of Contents**

[1. Objective 2](#_Toc148054329)

[2. Prerequisites 2](#_Toc148054330)

[3. Task Distribution 2](#_Toc148054331)

[4. Stored Procedures 3](#_Toc148054332)

[Benefits of Stored Procedures 3](#_Toc148054333)

[Variables. 3](#_Toc148054334)

[CREATE Stored Procedure 4](#_Toc148054335)

[How to execute Stored Procedure 5](#_Toc148054336)

[Stored Procedures without I/O parameters 5](#_Toc148054337)

[Stored procedure with input parameters 5](#_Toc148054338)

[Store Procedures with output parameters 6](#_Toc148054339)

[IF-ELSE conditions 7](#_Toc148054340)

[Self-exploration 8](#_Toc148054341)

# Objective

The purpose of this lab manual is to introduce views and stored procedures and how to create them and use them.

# Prerequisites

* SQL Server 2014 Database Development.
* Chapter 5 Elmasri

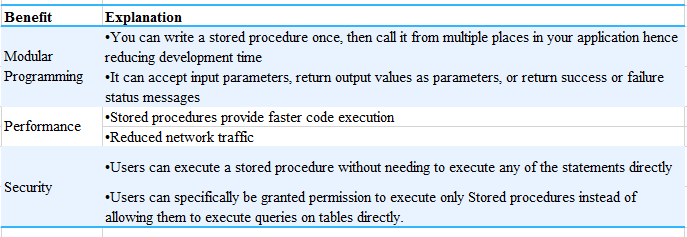
# Task Distribution

|  |  |
| --- | --- |
| Total Time | 150 Minutes |
| Stored Procedures | 30 Minutes |
| Exercise | 90 Minutes |
| Evaluation | 30 Minutes |

# Stored Procedures

Stored Procedure in SQL server can be defined as the set of logical group of SQL statements which are grouped to perform a specific task. A stored procedure is a prepared SQL code that you save so that you can reuse the code over and over again.

## Benefits of Stored Procedures



Every time you execute simple SQL statements, syntax checking and compilation are done before execution and data return. However, syntax check and compilation is done while creating a procedure, and not on every execution which makes it faster than simple SQL statements.

## Variables.

Before we start with stored procedures, we should get to know the variables. Like in any other programing language SQL also provides scalar variables, which are very useful when creating stored procedures.

* Variable in SQL start with @ symbol
* Variable is declared using DECLARE keyword as follow
  + *DECLARE @variableName datatype;*

Or to declare multiple variables in one statement.

* + *DECLARE @variable1Name Datatype,@variable2Name  datatype;*
* Variable can be assigned a constant scalar value as follow
  + *SET  @ variableName  = value;*

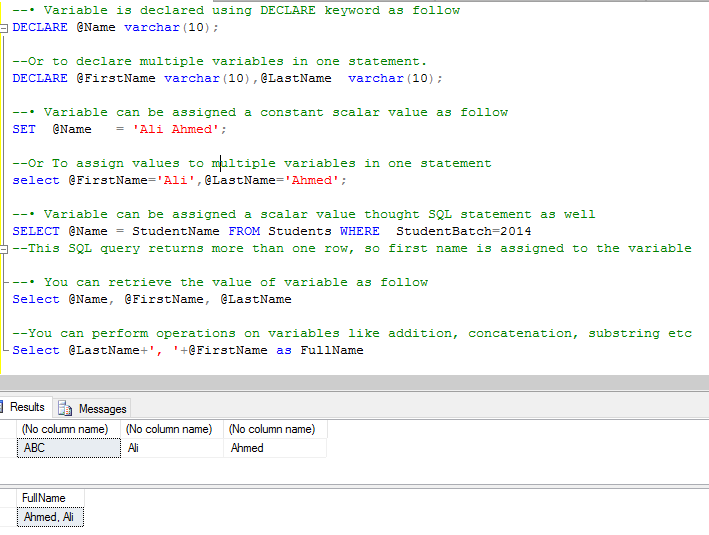
Or To assign values to multiple variables in one statement

* + *select @ variable1Name   = value, @variable2Name  =value;*
* Variable can be assigned a scalar value thought SQL statement as well
  + *SELECT @vairableName = columnName FROM Table WHERE  <condition>*

If SQL query returns more than one row, 1st value will be assigned to variable

* You can retrieve the value of variable as follow
  + *Select @variableName*
* You can perform operations on variables like addition, concatenation, substring etc.

TRY IT



NOTE: USE AND DECLARE VARIABLE IN SAME BATCH OF STATEMENTS, IF DECLARE STATEMENT IS NOT IN SAME BATCH, YOU WILL GET ERROR WHILE USING A VARIABLE.

## CREATE Stored Procedure

Following is the syntax to create stored procedure: Input and output parameter a uses as required.

CREATE PROCEDURE [procedureName]

@input\_param1 datatype,

@input\_param2 datatype,

@output\_param1 datatype OUTPUT,

@output \_param2 datatype OUTPUT

AS

BEGIN

(SQL Queries)

END

go

## How to execute Stored Procedure

declare @my\_output\_param1 int,

@my\_output\_param2 varchar(10) --these are the variables in which output variables of procedure will return values

Exec dbo.procedure\_name

@input\_param1=value,

@input\_param2 =value,

@output\_param1=@my\_output\_param1 OUTPUT ,

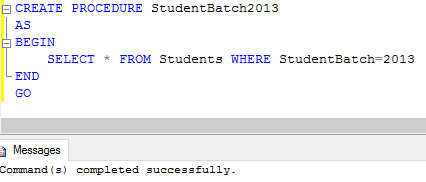
@output\_param2 =@my\_output\_param2 OUTPUT

select @my\_output\_param1 ,@my\_output\_param2 – you will then have to use select statements to retrieve data from parameters

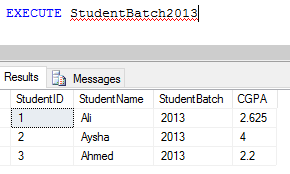
### Stored Procedures without I/O parameters

TRY IT:

Create this procedure to obtain all the students of batch 2013



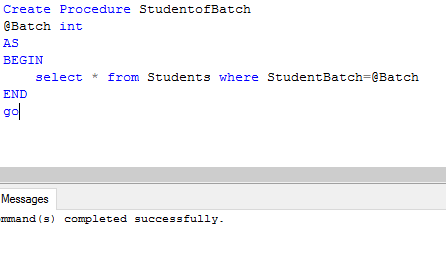
Now execute this procedure



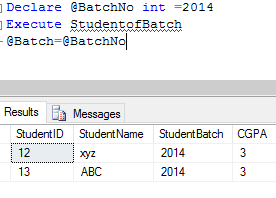
### Stored procedure with input parameters

TRY IT

Create a SP which takes batchNo as input and returns all students of that batch.



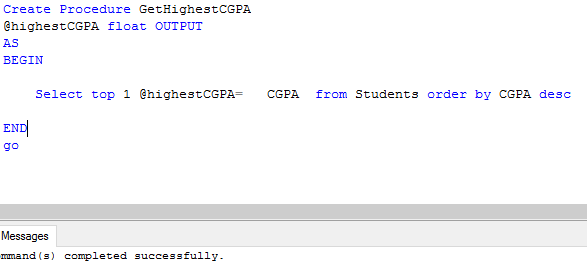
Now execute it



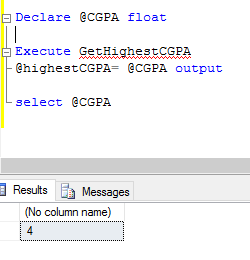
### Store Procedures with output parameters

TRY IT:

Create a stored procedure that will return max CGPA in an output parameter



Execute it

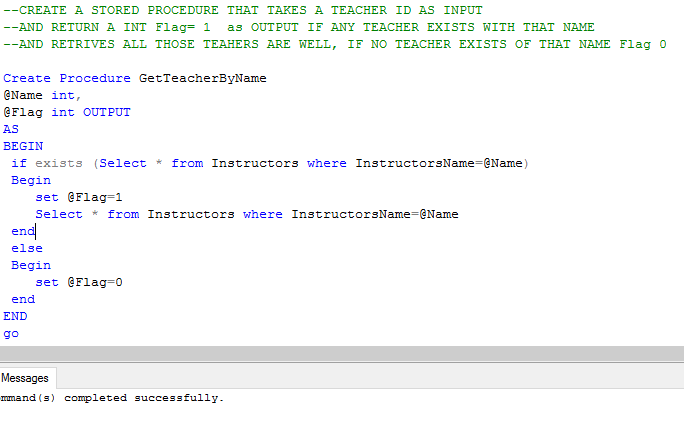


QUESTION: WRITE A SP TO GET AVERAGE CGPA.

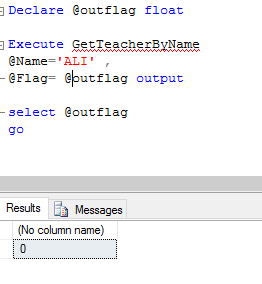
### IF-ELSE conditions

Like in any programing language IF—ELSE in SQL provide ability to conditionally execute a code.

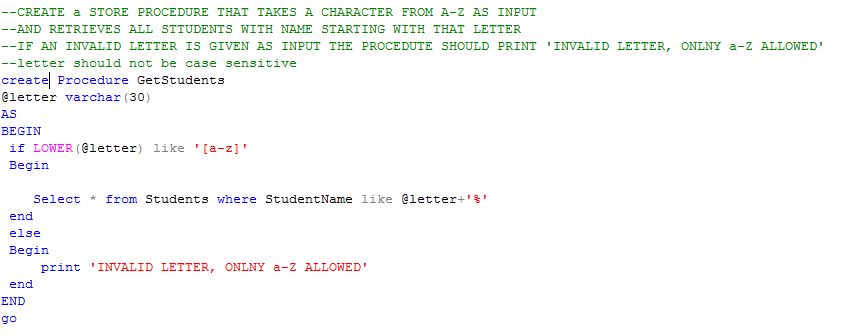
TRY THIS



Execute it



TRY ANOTHER



TRY EXECUTING THESE

execute GetStudents @letter= 'B'

execute GetStudents @letter= '1'

## Self-exploration

* What are default values? How can you set default values of parameters of Stored Procedures?
* How can you alter your procedure? (Hint same as View)