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

Department of Computer Science

FAST-NU, Lahore, Pakistan

**Table of Contents**

[1. Objective 2](#_Toc54289438)

[2. Prerequisites 2](#_Toc54289439)

[3. Task Distribution 2](#_Toc54289440)

[4. Views 3](#_Toc54289441)

[Create a View 3](#_Toc54289442)

[Use a View 4](#_Toc54289443)

[Alter a View 5](#_Toc54289444)

[Insert Update Delete Data Through View 6](#_Toc54289445)

[With Check Option 7](#_Toc54289446)

[5. Stored Procedures 8](#_Toc54289447)

[Benefits of Stored Procedures 8](#_Toc54289448)

[Variables. 8](#_Toc54289449)

[CREATE Stored Procedure 9](#_Toc54289450)

[How to execute Stored Procedure 10](#_Toc54289451)

[Stored Procedures without I/O parameters 10](#_Toc54289452)

[Stored procedure with input parameters 10](#_Toc54289453)

[Store Procedures with output parameters 11](#_Toc54289454)

[IF-ELSE conditions 12](#_Toc54289455)

[Self-exploration 13](#_Toc54289456)

# 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

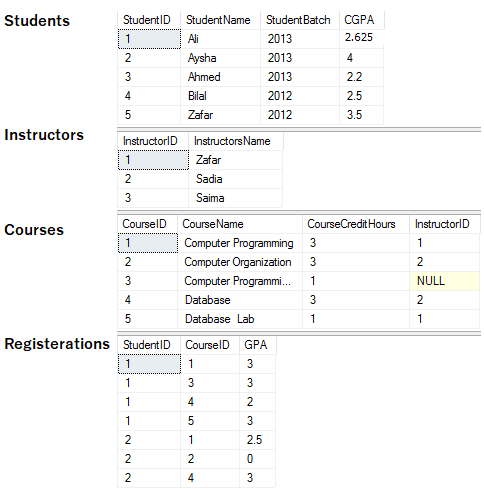
# Task Distribution

|  |  |
| --- | --- |
| Total Time | 120 Minutes |
| Views | 20 Minutes |
| Stored Procedures | 20 Minutes |
| Exercise | 60 Minutes |
| Evaluation | 20 Minutes |

# Views

In previous lab manuals, you have learned how to write select query to retrieve data. While some select queries you write might be used only for one time activity, some select queries are used again and again within your application/environment. Some of these queries that you reuse within your environment contain complex logic, and you would not want to rewrite them every time you use them. SQL server allows you to store a SELECT statement within a database using an object called a view. In this section, you will learn how to CREATE a view, modify data through a view, how to ALTER a view, and how to use a view.

We will use the Student schema for all the examples (given in previous labs)



## Create a View

View is simply a select statement that has been given a name and stored in dataset. View is also called a virtual table, because there is no data in the view itself, it’s just a select query that get data from base tables.

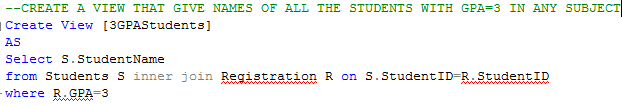
create View <ViewName>

AS

<Select Query>

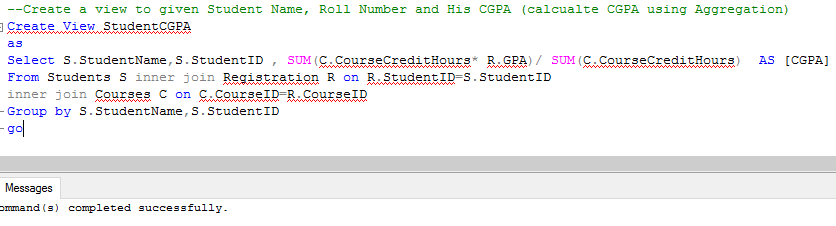
When you excute a create view statement you should get command successful notification, just like when you created a table.

TRY IT



Here the base tables are Student and Registration

TRY THIS



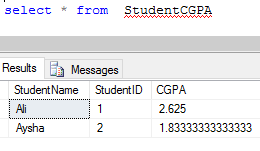
Here the base tables are Students, Registration and Courses.

\*\*NOTE: EVERY COLUMN RETURNED BY SELECT QUERY OF VIEW SHOULD HAVE UNIQUE NAME, DERIVED COLUMNS SHOULD BE GIVEN ALIAS. COLUMNS WITH SAME NAMES SHOULD ALSO BE GIVEN DISTINCT ALIAS

## Use a View

As already told view are virtual tables. You can use them as regular tables in SELECT statement.

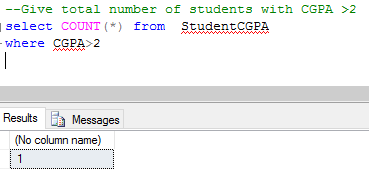
TRY IT



\*\*NOTE: this data was not present in StudentCGPA view, rather when you select a view, the Select query in body of view is executed and result is returned.

Similarly you can join views with tables of views, you can take aggregates of view.

TRY IT



## Alter a View

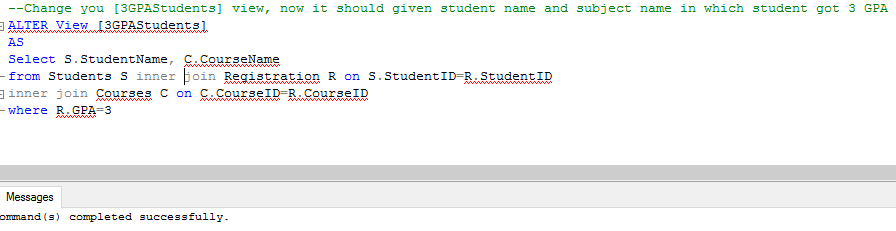
You can change the select query of your view by using following syntax

Alter View <ViewName>

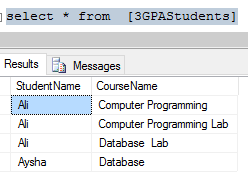
AS

<Select Query>

TRY IT



Now retrived the data from view

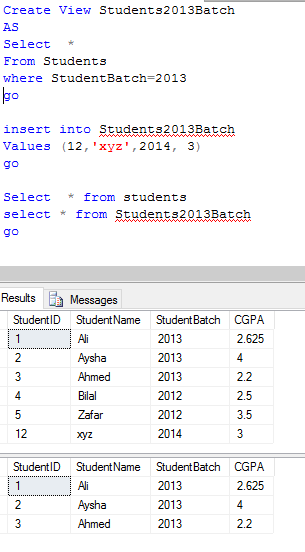


## Insert Update Delete Data Through View

As view is a virtual table and has no data of its own, if you run delete insert or update query on view, the data in base table will change (if the change is feasible and is not violating any constraint). If the select query in View has joins and aggregates then delete insert or update would not work.

Read Elmasri Chapter 5 for more details.

TRY IT

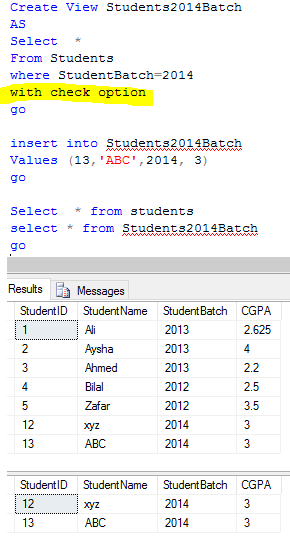


## With Check Option

With Check option ensures that the only data manipulation that can occur through view also must be retrievable though that view.

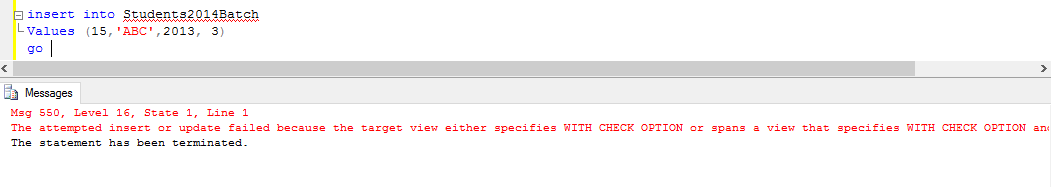
In previous example, the XYZ student we added though the view, was not retrievable thought view. If we add with check option that insertion would not have been possible though view.

TRY IT



Now try adding a row that thought Student2014Batch that will not be retrievable though it

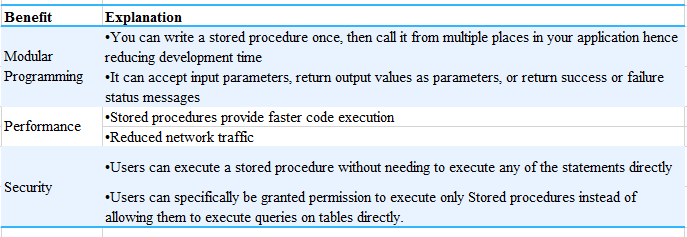
TRY IT



# 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;*

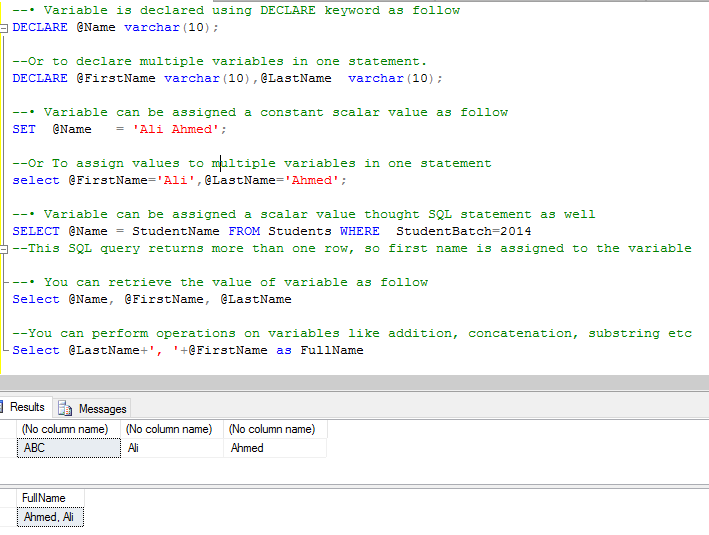
Or To assign values to multiple variables in one statement

* + *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 ,

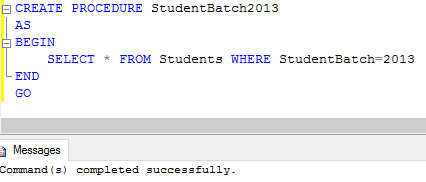
@output\_param2 =@my\_output\_param2 OUTPUT

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

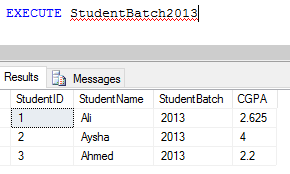
### Stored Procedures without I/O parameters

TRY IT:

Create this procedure to obtain all the students of batch 2013



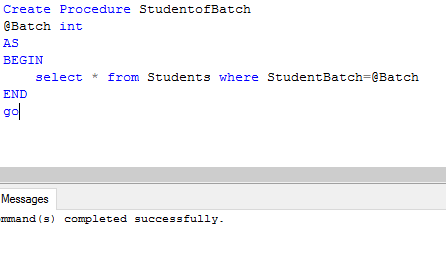
Now execute this procedure



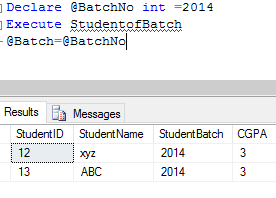
### Stored procedure with input parameters

TRY IT

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



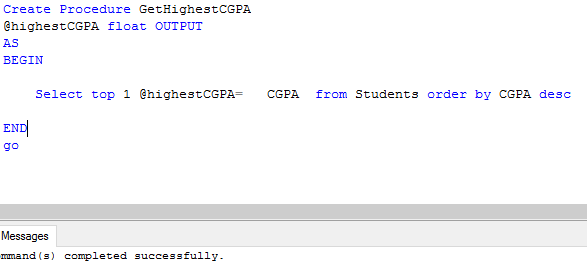
Now execute it



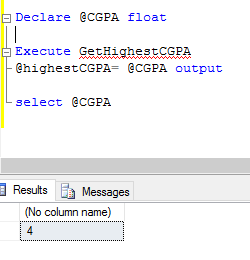
### Store Procedures with output parameters

TRY IT:

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



Execute it

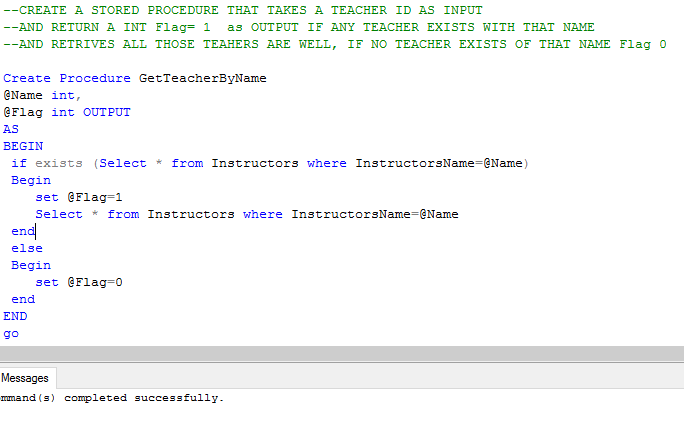


QUESTION: WRITE A SP TO GET AVERAGE CGPA.

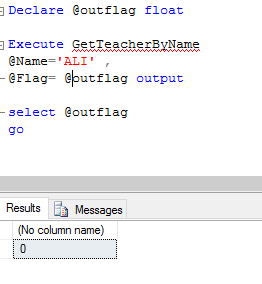
### IF-ELSE conditions

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

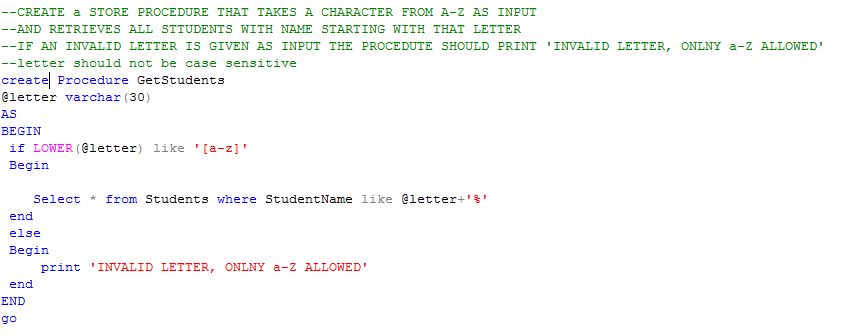
TRY THIS



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)