**The University Of Azad Jammu & Kashmir,**

**Muzaffarabad**

**Department of Software Engineering**

**LAB TASK 12**

**Database Systems**

**Course Code**: **CS-2204**

**Submitted To:**

Engr. Tahir Jahangir

**Submitted By:**

Ahmed Ali

**Roll No:**

2023-SE-41

**Session**: 2023-2027

Contents

[Lab Task – 12: SQL Server – Functions & Stored Procedures 5](#_Toc207376198)

[🎯 Objective: 5](#_Toc207376199)

[📘 Task 01: Database and Table Setup 5](#_Toc207376200)

[Instructions: 5](#_Toc207376201)

[📘 Task 02: Scalar Function with One Parameter 6](#_Toc207376202)

[Instructions: 6](#_Toc207376203)

[📘 Task 03: Scalar Function with Multiple Parameters 8](#_Toc207376204)

[Instructions: 8](#_Toc207376205)

[📘 Task 04: Table-Valued Function 10](#_Toc207376206)

[Instructions: 10](#_Toc207376207)

[📘 Task 05: Stored Procedure for Insertion and Retrieval 11](#_Toc207376208)

[Instructions: 11](#_Toc207376209)

[Part A: 11](#_Toc207376210)

[Part B: 11](#_Toc207376211)

# Lab Task – 12: SQL Server – Functions & Stored Procedures

# 🎯 Objective:

In this lab, students will explore the creation and usage of **scalar functions**, **table-valued functions**, and **stored procedures** in SQL Server. They will reinforce their understanding by writing reusable logic blocks for data retrieval and insertion using user-defined functions and procedures.

# 📘 Task 01: Database and Table Setup

🎯 **Objective**: Prepare a clean working environment with structured data.

## Instructions:

1. Create a new database with your name in the format:

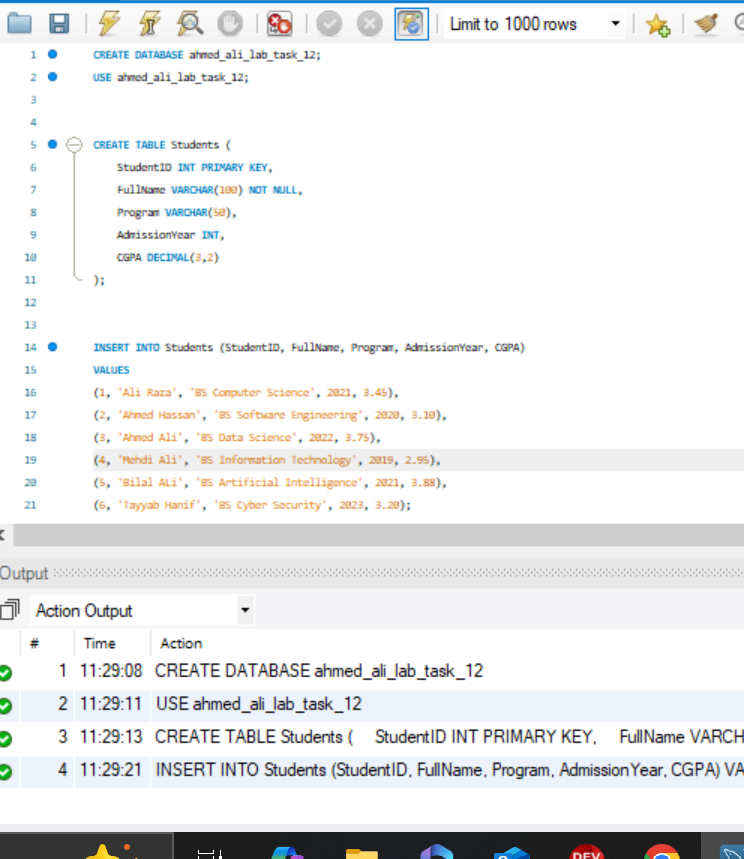
your\_full\_name\_lab\_task\_12

*Example: hania\_kanwal\_lab\_task\_12*

1. Create a table named Students with the following structure:

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| StudentID | INT | PRIMARY KEY |
| FullName | VARCHAR(100) | NOT NULL |
| Program | VARCHAR(50) | — |
| AdmissionYear | INT | — |
| CGPA | DECIMAL(3,2) | — |

1. Insert at least **6 sample records** into the Students table with varied programs and CGPA values.

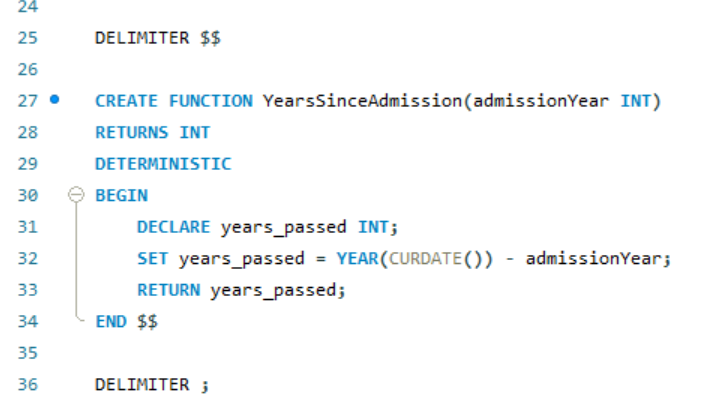


# 📘 Task 02: Scalar Function with One Parameter

🎯 **Objective**: Create a user-defined function with a single input.

## Instructions:

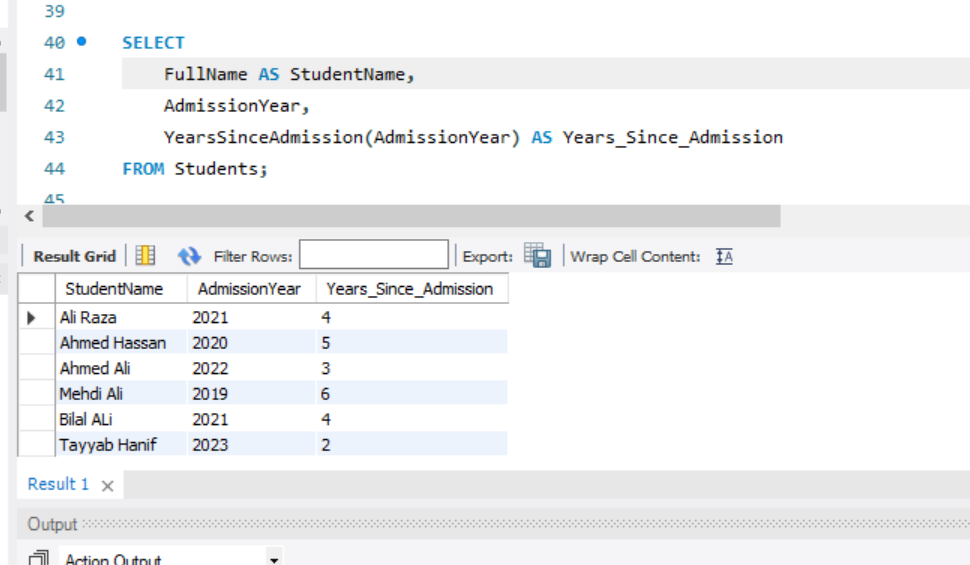
1. Create a scalar function that takes a student’s admission year as input and returns how many years have passed since their admission.



1. Write a SELECT query using this function to display:
   * Student name

o Admission year

* + Years since admission(calculated using your function)

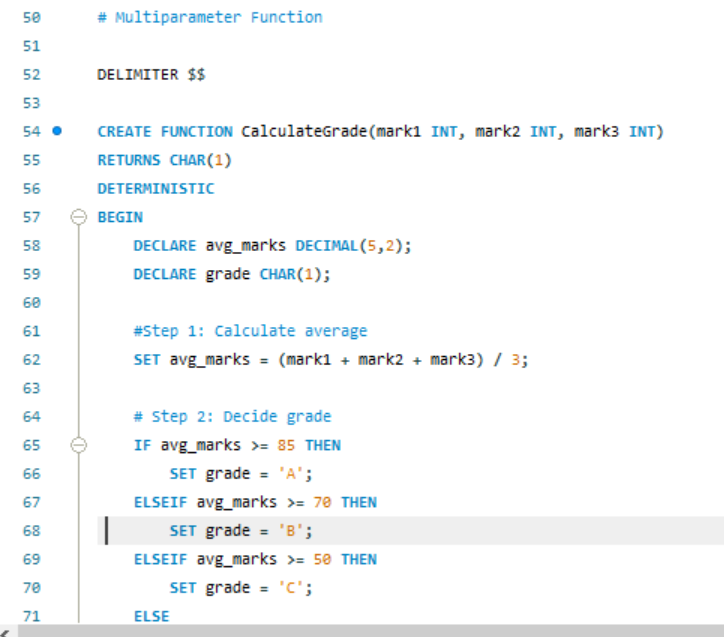


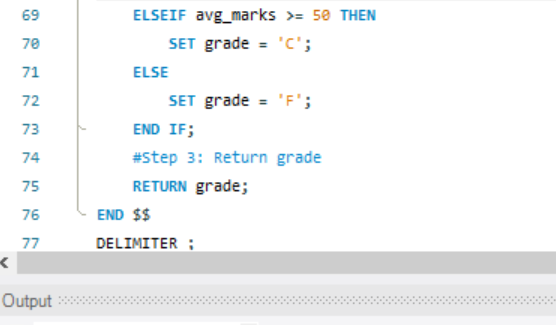
# 📘 Task 03: Scalar Function with Multiple Parameters

🎯 **Objective**: Understand parameterized logic with scalar UDFs.

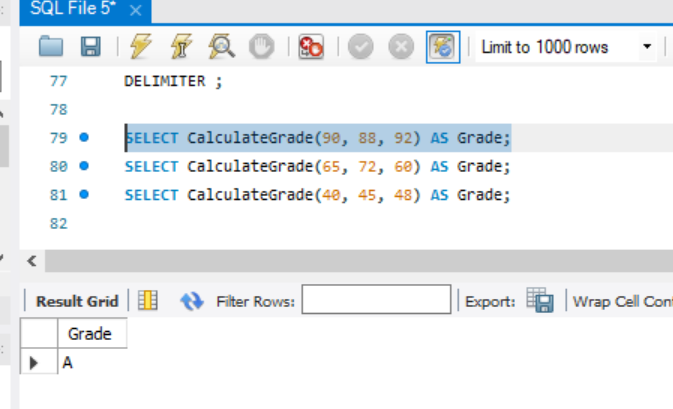
## Instructions:

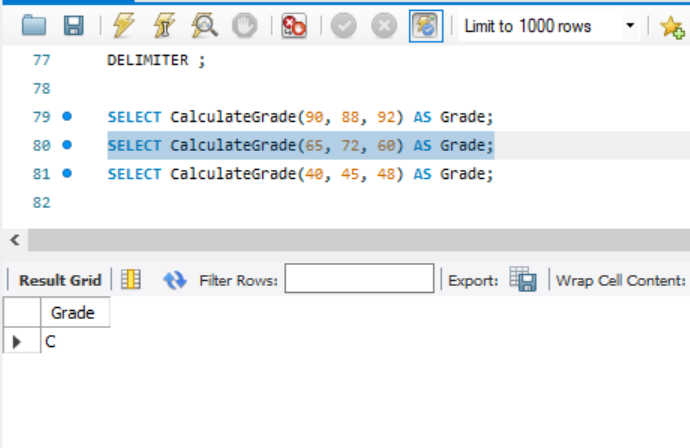
1. Create a function that takes three marks (out of 100) as input, calculates the average, and returns a **grade** as per the logic below: o A: 85 and above o B: 70–84 o C: 50–69 o F: Below 50





1. Use the function in a SELECT statement to test at least two sample input values.



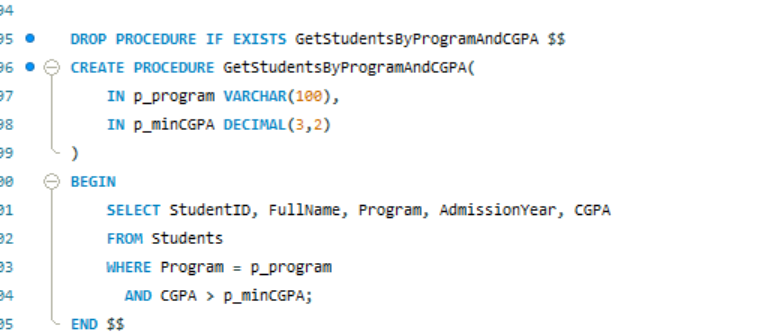


# 📘 Task 04: Table-Valued Function

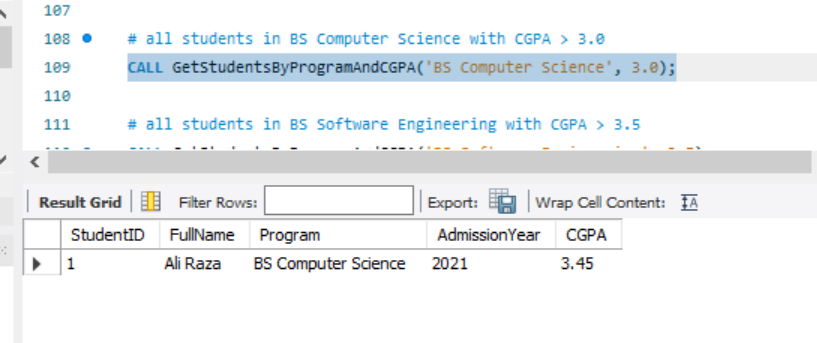
🎯 **Objective**: Use UDFs to return result sets dynamically.

## Instructions:

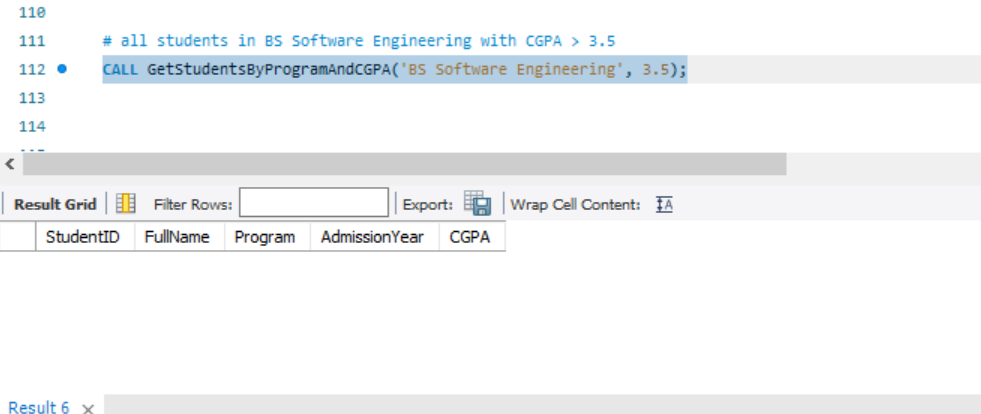
1. Create a table-valued function that:
   * Takes two parameters: program name and minimum CGPA.
   * Returns students from that program with CGPA above the threshold.



1. Use the function in a SELECT query to return:
   * All students in **BSCS** with CGPA above 3.0



* All students in **BSSE** with CGPA above 3.5



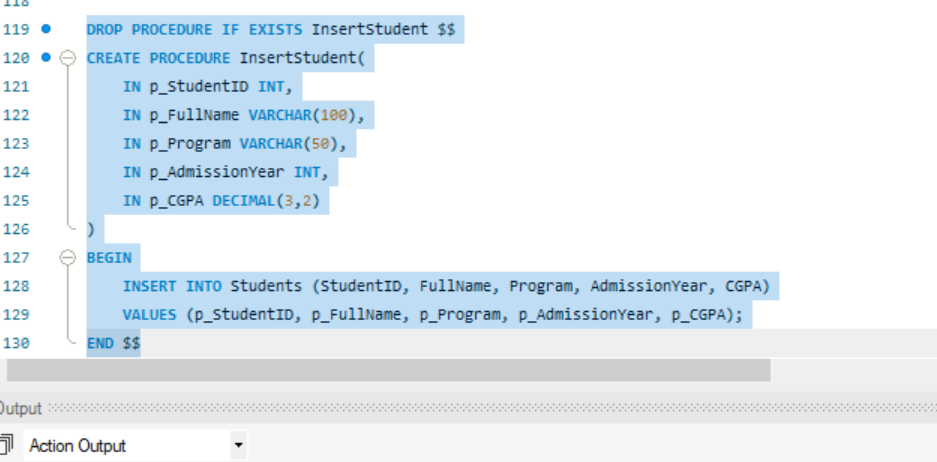
# 📘 Task 05: Stored Procedure for Insertion and Retrieval

🎯 **Objective**: Use stored procedures for real-world-like actions.

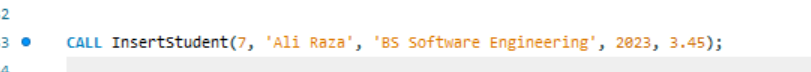
## Instructions:

## Part A:

 Create a stored procedure that inserts a new student into the Students table..



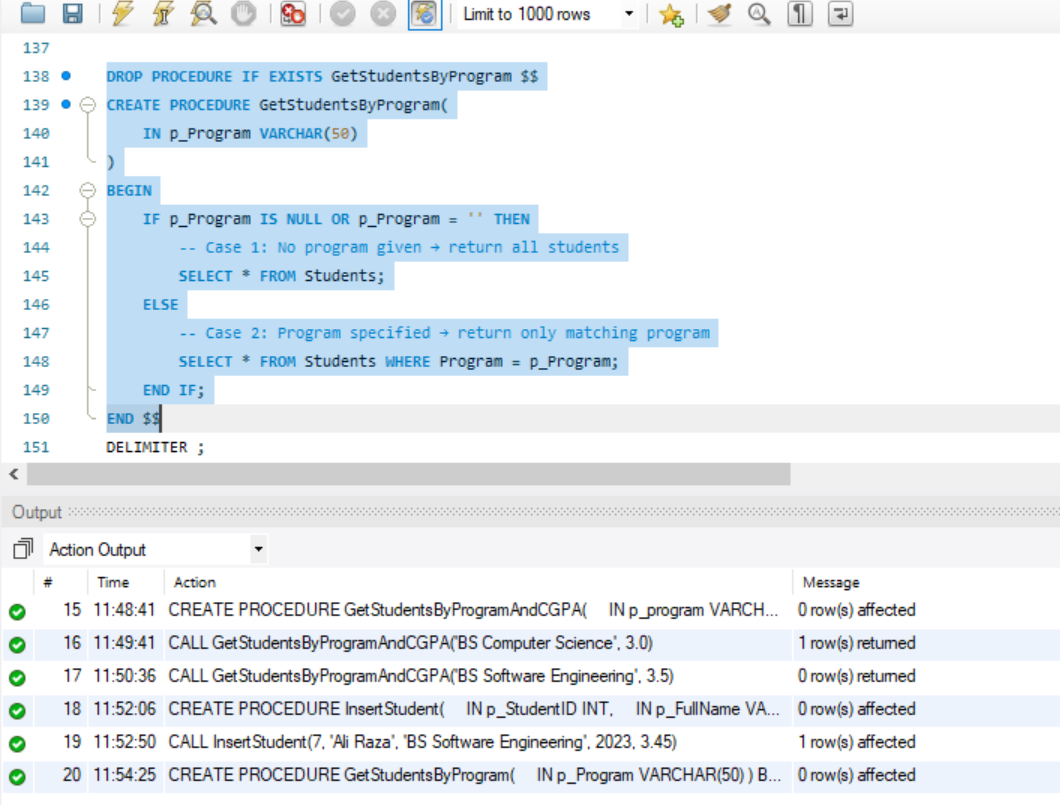
 Use this procedure to add at least one new record



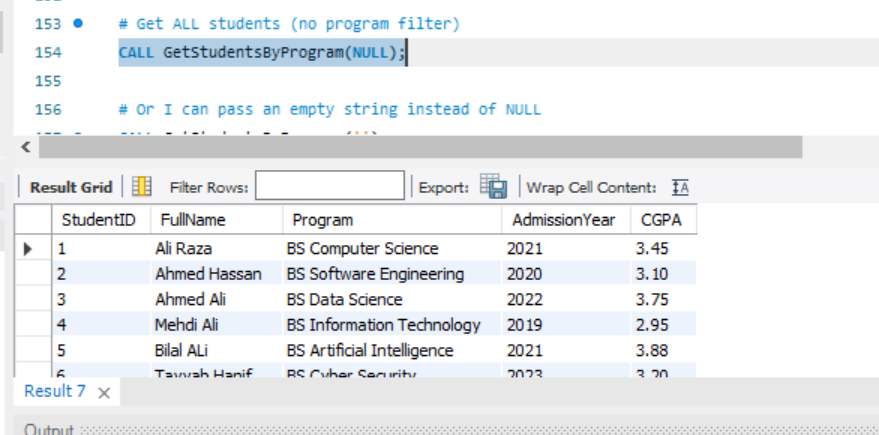
## Part B:

 Create another stored procedure that:

* Accepts an optional input parameter for the Program.

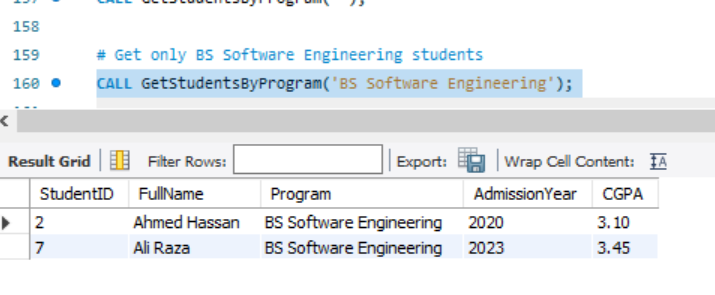


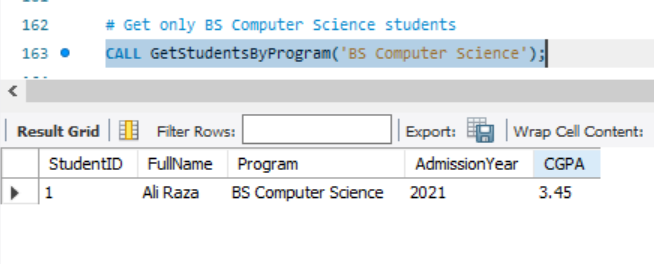
* If no input is given, return all students.



* If a program is specified, return students from that program only.

 Execute both forms of the procedure to test its behavior.





**Reflection and Learnings:**

In this lab, I learned how to create and use stored procedures for inserting and retrieving data efficiently. I understood how parameters make procedures flexible, especially when filtering results conditionally. A key challenge was managing optional parameters and syntax errors, but resolving them improved my confidence in writing reusable SQL code. Overall, this task highlighted how stored procedures improve security, consistency, and ease of database operations.