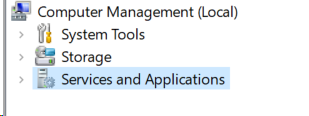
**Connecting SQL server With Golang**

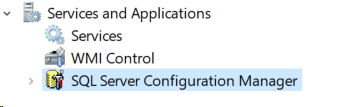
**REQUIRED STEPS:**

In Search bar type << **Computer Management**

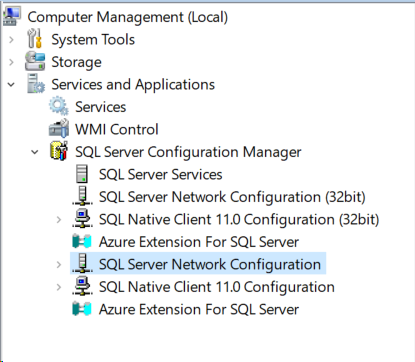
**Go to Services and Applications**



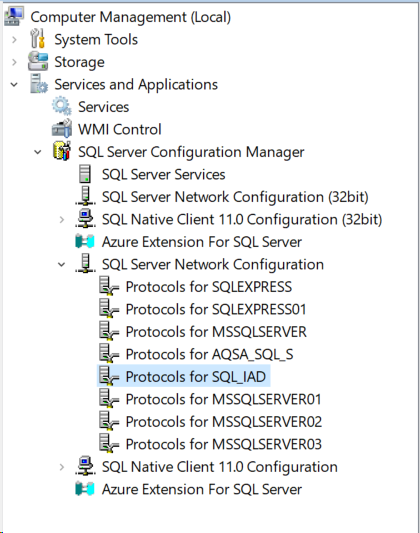
**Go to SQL Server Configuration Manager**



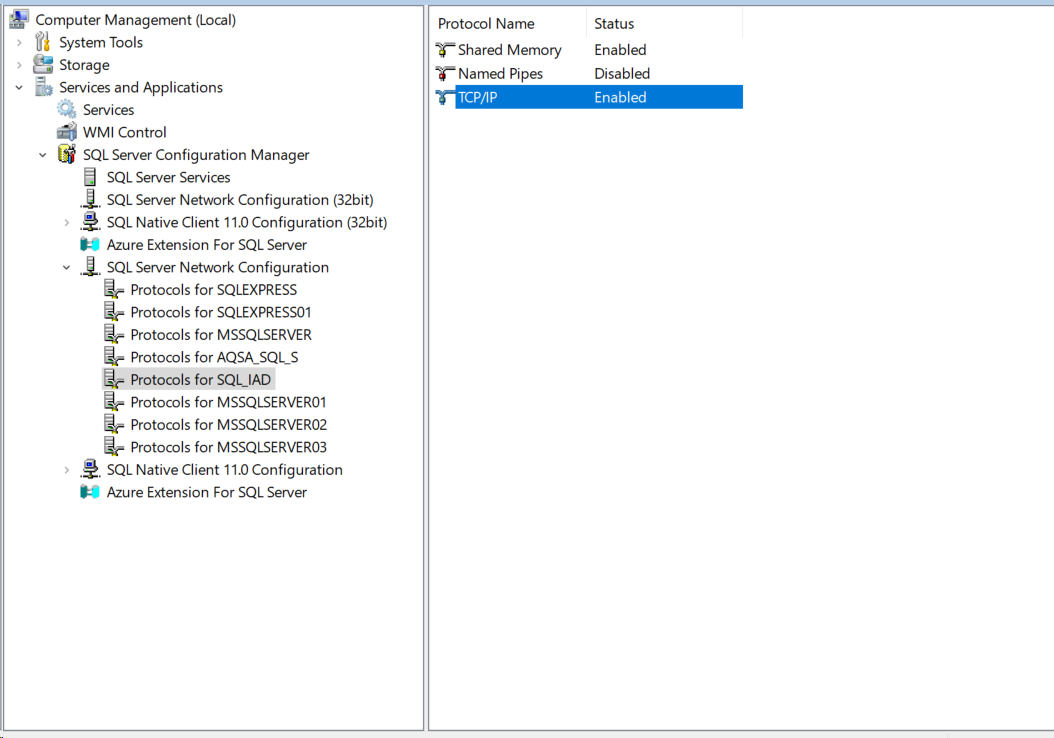
**Go to SQL server Network Configuration**



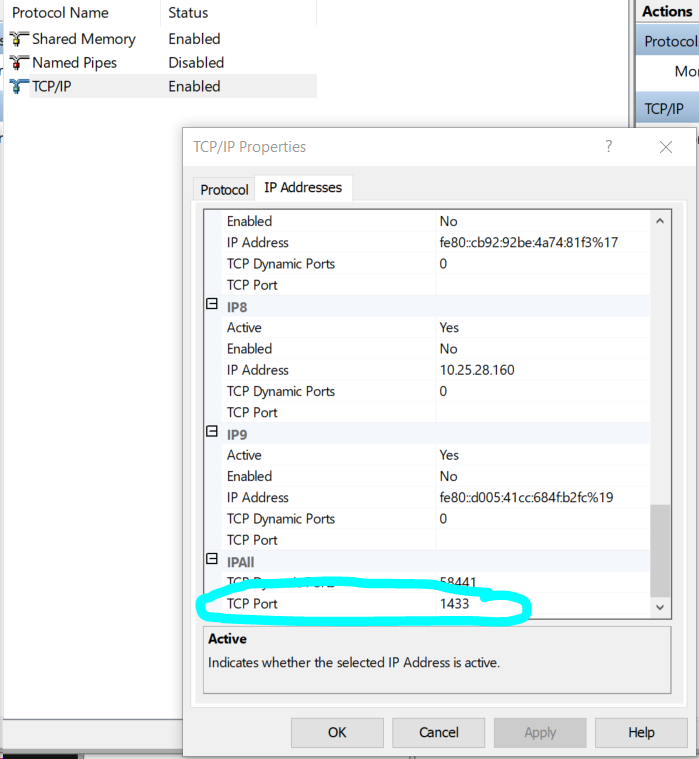
**Go to your Desired Server:**



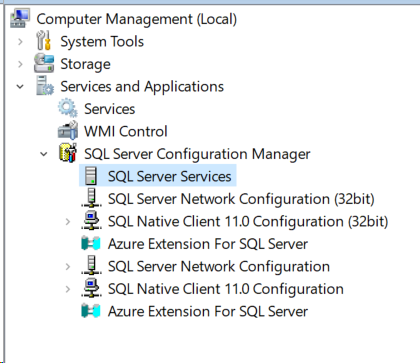
**Enable The TCP/IP from here**



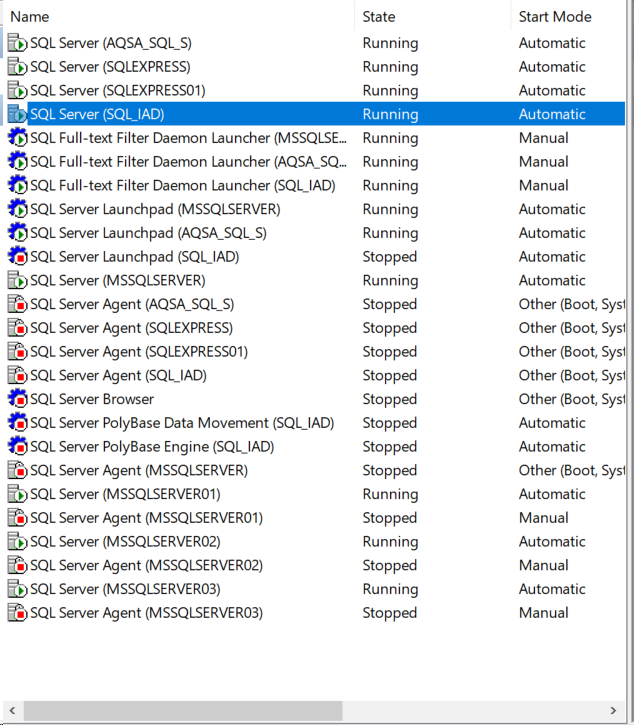
**Right click on TCP/IP Enable it and go to the Properties**

**Go to IP address tab and write the TCP port number:**

**Now go to the SQL SERVER SERVICES**

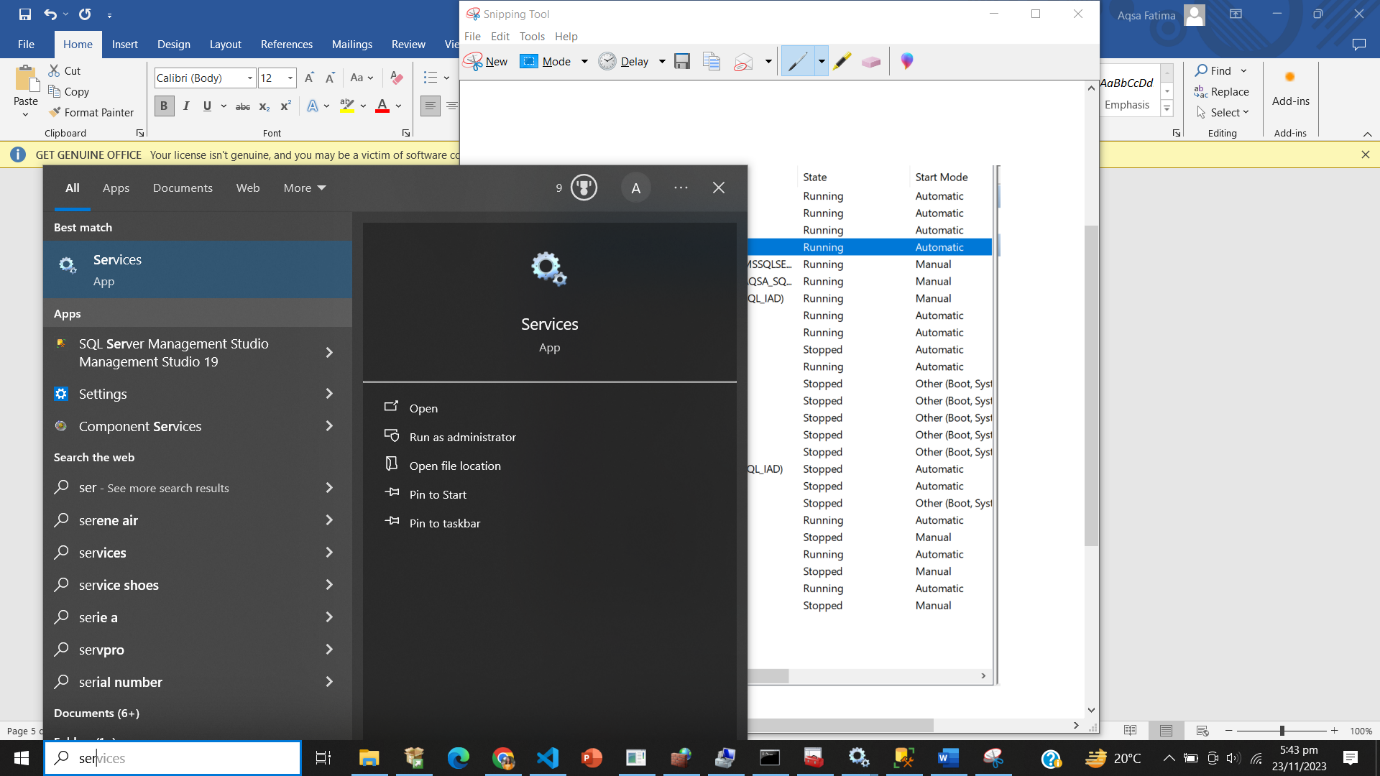


**Right click the server and Restart it. Then the Changed Configuration will be applied**

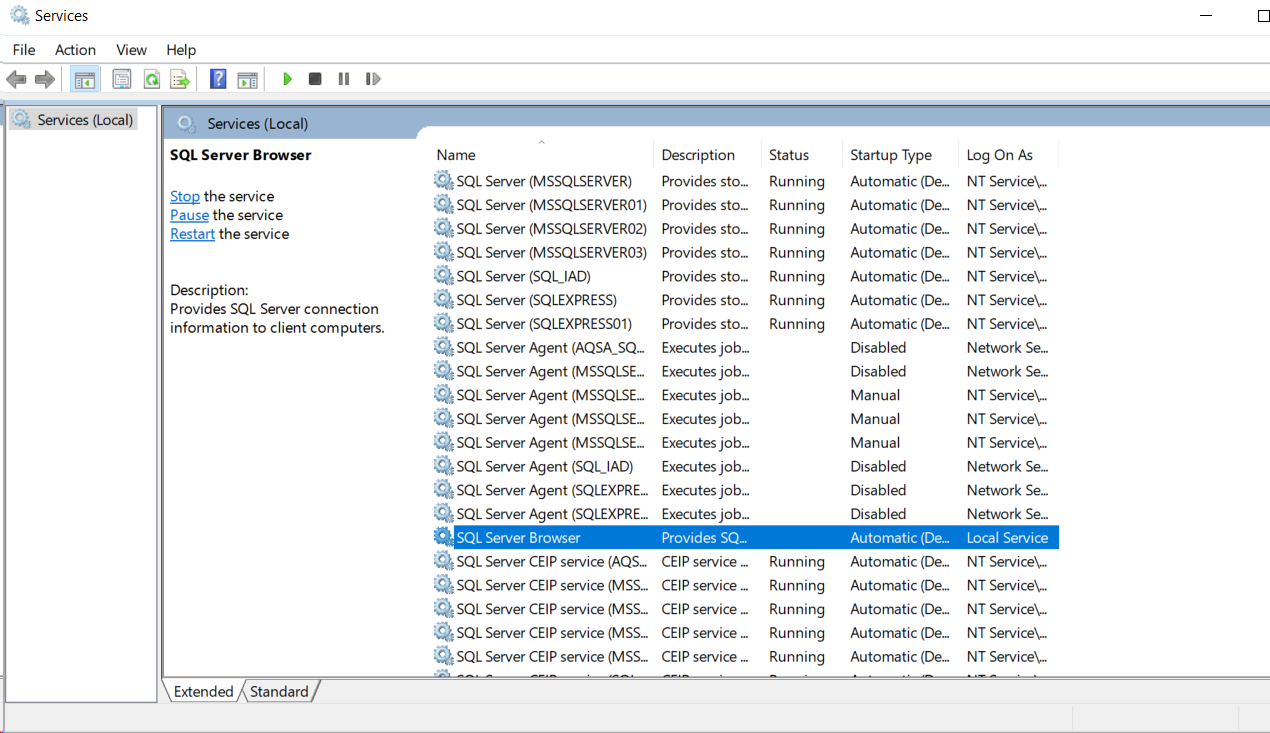


Now we will start the **SQL SERVER BROWSER**

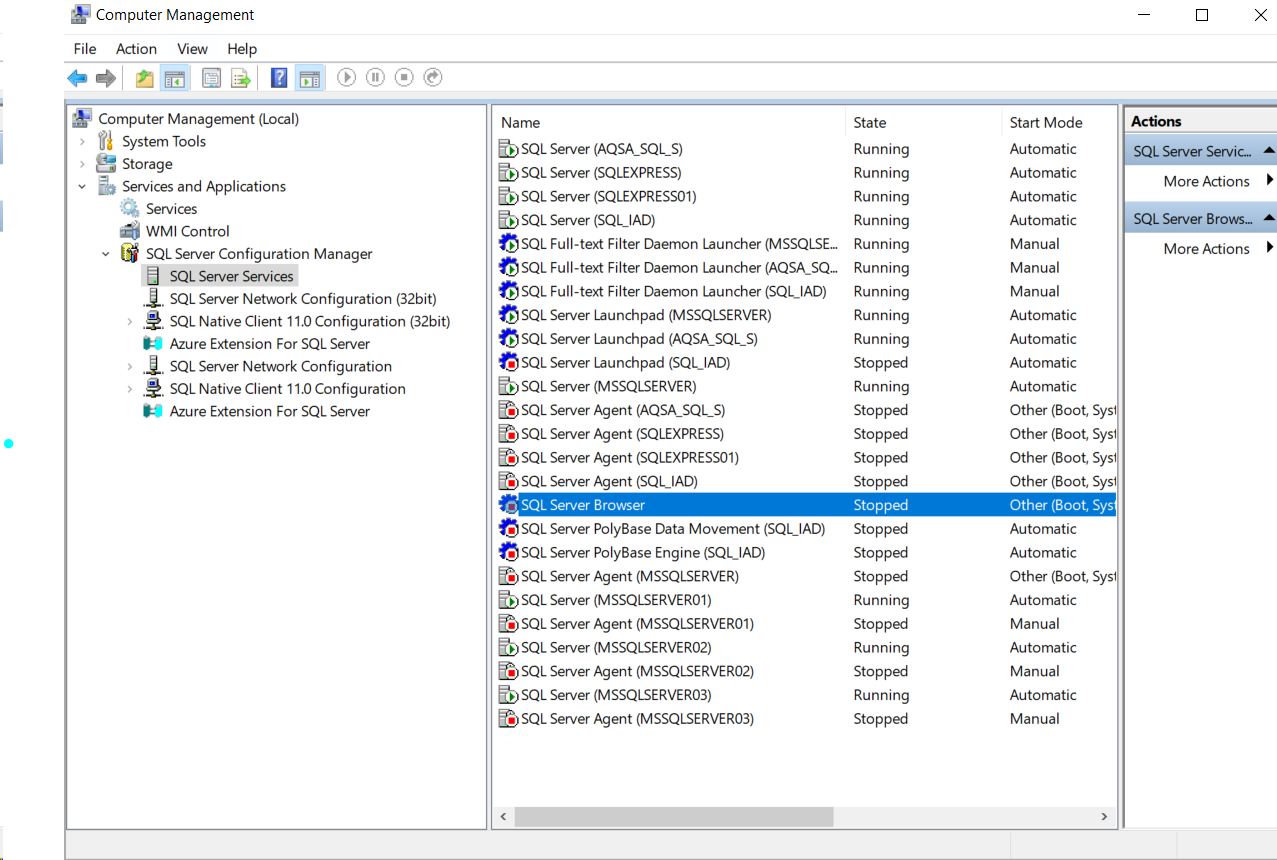
In the Start menu write **“Services”**



From here check the SQL browser is running or not:

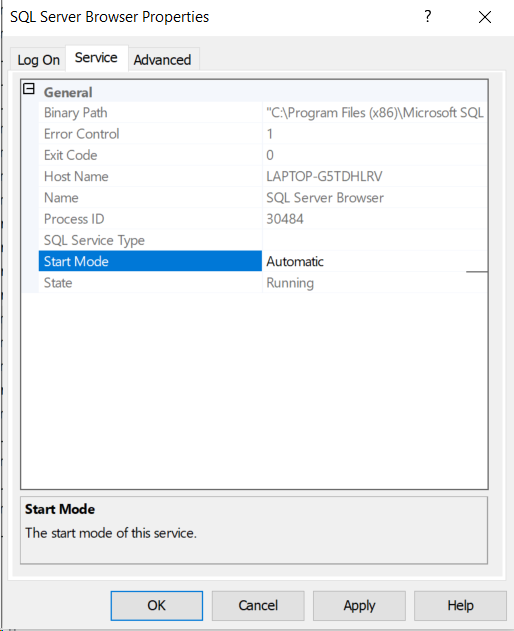


If it not running go to



And Start the SQL Sever Browser. Right Click it and go to the properties. Then go to the services and set the **“Start mode” to “Automatic”**

Now go to the Sql Srver Browser and Restart it to implement the changed Configurations



***Congratulations!!!! You Have Performed All The Required Steps To Setup SQL Server For Connection With Your Golang Code 😊***

***CODING PART (How to connect your SQL server with your Golang Code)***

Create the directories and the files where you want to connect your SQL server with your Golang Code

Open the terminal and go to the required Dirctory:

**Create a Go Module:** Open a terminal and navigate to your project directory. Run the following command to initialize your project as a module:

In the terminal:

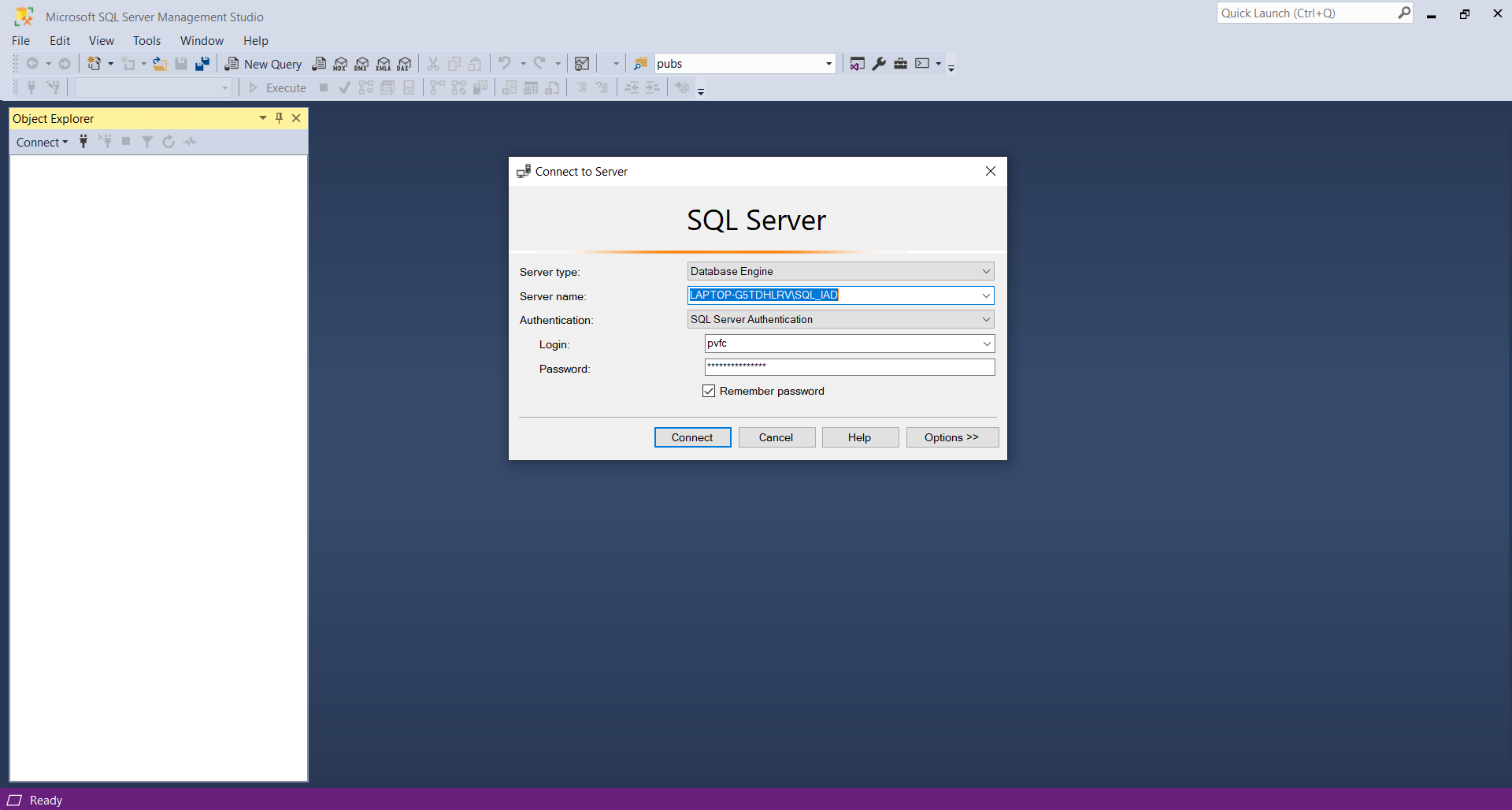
**go mod init <EnterDummyNameForYourServer>**

**Install the Database Driver:** Now, you can install the **go-mssqldb** package using the **go get** command:

In terminal (If you are a student of PIEAS don’t use LAN):

**go get github.com/denisenkom/go-mssqldb**

Go to the ssms (SQL server Management Studio):



**Note:**

Use SQL server Authentication (With username and Password) Otherwise you will face some Issues.

Connection String Pattern:

connString := "server=LAPTOP-G5TDHLRV\\SQL\_IAD;port=1433;database=PVFC;user id=pvfc;password=pvfc;"

In SQL server Managemnet studio. Make the requied tables and insert data into and then connect with golang and try to retrieve the data to test your connection.

USE PVFC; -- Use your database name

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY,

FirstName NVARCHAR(50),

LastName NVARCHAR(50),

Email NVARCHAR(100)

);

INSERT INTO Customer (CustomerID, FirstName, LastName, Email)

VALUES

(1, 'John', 'Doe', 'john.doe@example.com'),

(2, 'Jane', 'Smith', 'jane.smith@example.com');

EXAMPLE CODE TO TEST THE CONNECTION:

package main

import (

    "database/sql"

    "fmt"

    \_ "github.com/denisenkom/go-mssqldb"

)

func main() {

    // Define connection string

    connString := "server=LAPTOP-G5TDHLRV\\SQL\_IAD;port=1433;database=PVFC;user id=pvfc;password=pvfc;"

    // Open a connection to the database

    db, err := sql.Open("sqlserver", connString)

    if err != nil {

        fmt.Println("Error connecting to the database:", err.Error())

        return

    }

    defer db.Close()

    // Test the connection

    err = db.Ping()

    if err != nil {

        fmt.Println("Error pinging database:", err.Error())

        return

    }

    fmt.Println("Connected to the database!")

    // Insert data into the Customer table

    // \_, err = db.Exec(`

    //     USE PVFC;

    //     INSERT INTO Customer (CustomerID, FirstName, LastName, Email)

    //     VALUES (3, 'Alice', 'Johnson', 'alice.johnson@example.com');

    // `)

    // if err != nil {

    //  fmt.Println("Error inserting data:", err.Error())

    //  return

    // }

    // fmt.Println("Data inserted successfully!")

    rows, err := db.Query("SELECT CustomerID, FirstName, LastName, Email FROM Customer")

    if err != nil {

        fmt.Println("Error querying data:", err.Error())

        return

    }

    defer rows.Close()

    fmt.Println("Retrieved data from the Customer table:")

    for rows.Next() {

        var customerID int

        var firstName, lastName, email string

        err := rows.Scan(&customerID, &firstName, &lastName, &email)

        if err != nil {

            fmt.Println("Error scanning row:", err.Error())

            return

        }

        fmt.Printf("%d: %s %s (%s)\n", customerID, firstName, lastName, email)

    }

}

OUTPUT:

