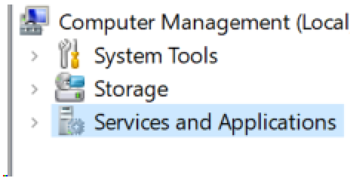
Connecting SQL server With Golang

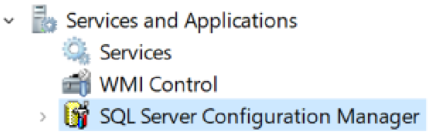
REQUIRED STEPS:

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

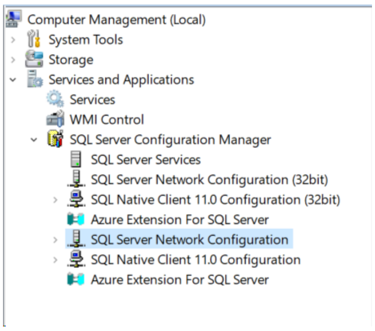
Go to Services and Applications



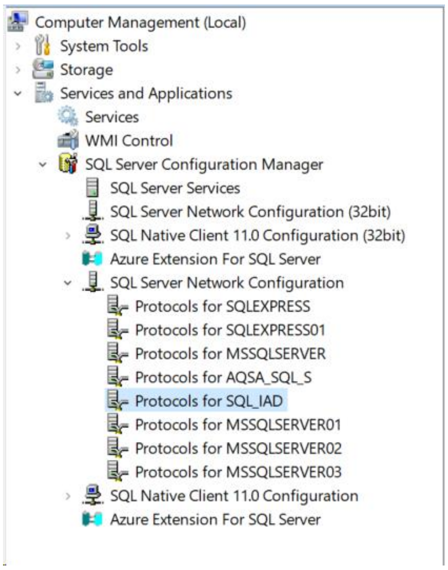
Go to SQL Server Configuration Manager



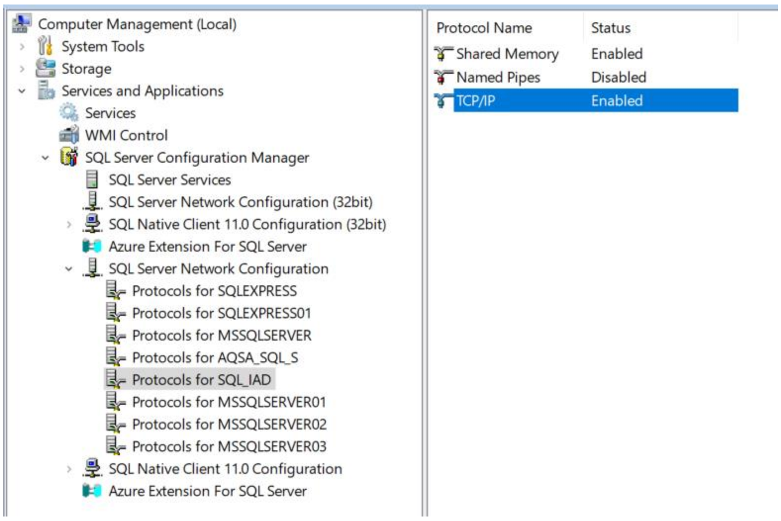
Go to SQL server Network Configuration


Computer Management (Local)
System Tools
Storage
Services and Applications
Services
WMI Control
SQL Server Configuration Manager
SQL Server Services
SQL Server Network Configuration (32bit)
SQL Native Client 11.0 Configuration (32bit)
Azure Extension For SQL Server
SQL Server Network Configuration
SQL Native Client 11.0 Configuration
Azure Extension For SQL Server

Go to your Desired Server:


Computer Management (Local)
System Tools
Storage
Services and Applications
Services
WMI Control
SQL Server Configuration Manager
SQL Server Services
SQL Server Network Configuration (32bit)
SQL Native Client 11.0 Configuration (32bit)
Azure Extension For SQL Server
SQL Server Network Configuration
Protocols for SQLEXPRESS
Protocols for SQLEXPRESS01
Protocols for MSSQLSERVER
Protocols for AQSA_SQL_S
Protocols for SQL_IAD
Protocols for MSSQLSERVER01
Protocols for MSSQLSERVER02
Protocols for MSSQLSERVER03
SQL Native Client 11.0 Configuration
Azure Extension For SQL Server

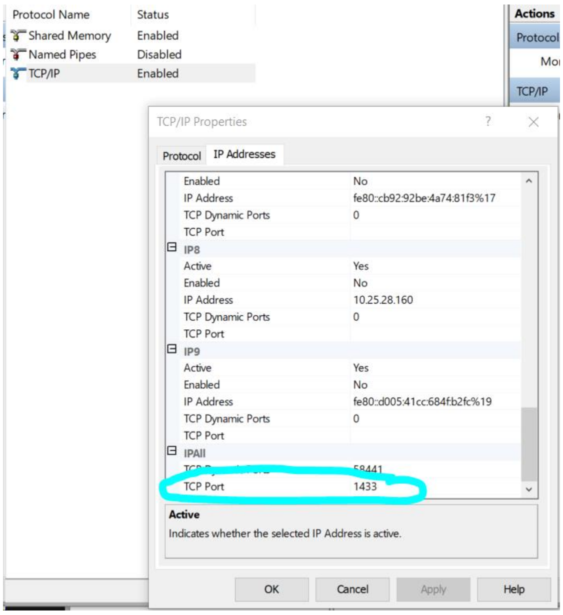
Enable The TCP/IP from here


Computer Management (Local)
System Tools
Storage
Services and Applications
Services
WMI Control
SQL Server Configuration Manager
SQL Server Services
SQL Server Network Configuration (32bit)
SQL Native Client 11.0 Configuration (32bit)
Azure Extension For SQL Server
SQL Server Network Configuration
Protocols for SQLEXPRESS
Protocols for SQLEXPRESS01
Protocols for MSSQLSERVER
Protocols for AQSA_SQL_S
Protocols for SQL_IAD
Protocols for MSSQLSERVER01
Protocols for MSSQLSERVER02
Protocols for MSSQLSERVER03
SQL Native Client 11.0 Configuration
Azure Extension For SQL Server

Protocol Name	Status
Shared Memory	Enabled
Named Pipes	Disabled
TCP/IP	Enabled

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

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



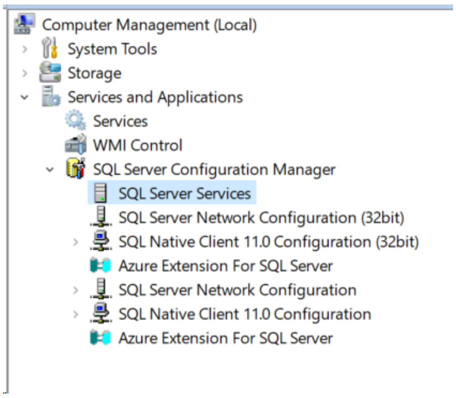
Protocol Name	Status
Shared Memory	Enabled
Named Pipes	Disabled
TCP/IP	Enabled

Enabled	No
IP Address	fe80::cb92:92be:4a74:81f3%17
TCP Dynamic Ports	0
TCP Port	
⊟ IP8	
Active	Yes
Enabled	No
IP Address	10.25.28.160
TCP Dynamic Ports	0
TCP Port	
⊟ IP9	
Active	Yes
Enabled	No
IP Address	fe80::d005:41cc:684f:b2fc%19
TCP Dynamic Ports	0
TCP Port	
⊟ IPAll	
TCP Dynamic Ports	58441
TCP Port	1433

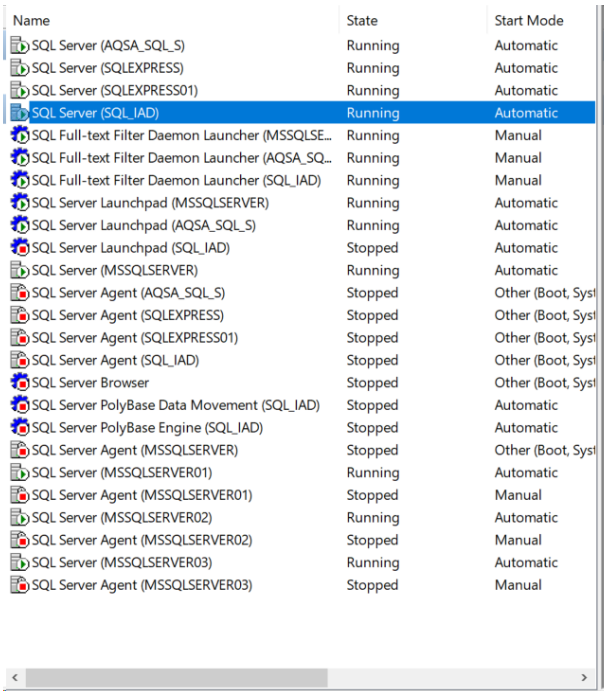
Active

Indicates whether the selected IP Address is active.

Now go to the SQL SERVER SERVICES


Computer Management (Local)
System Tools
Storage
Services and Applications
Services
WMI Control
SQL Server Configuration Manager
SQL Server Services
SQL Server Network Configuration (32bit)
SQL Native Client 11.0 Configuration (32bit)
Azure Extension For SQL Server
SQL Server Network Configuration
SQL Native Client 11.0 Configuration
Azure Extension For SQL Server

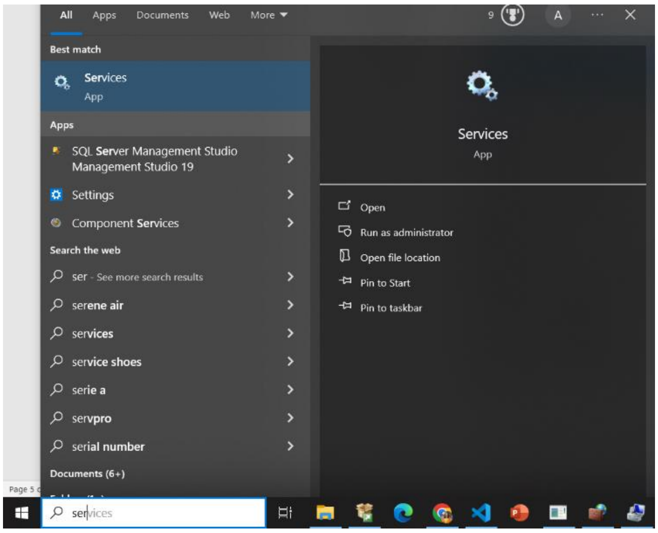
Right click the server and Restart it. Then the Changed Configuration will be applied



Name	State	Start Mode
SQL Server (AQSA_SQL_S)	Running	Automatic
SQL Server (SQLEXPRESS)	Running	Automatic
SQL Server (SQLEXPRESS01)	Running	Automatic
SQL Server (SQL_IAD)	Running	Automatic
SQL Full-text Filter Daemon Launcher (MSSQLSE...	Running	Manual
SQL Full-text Filter Daemon Launcher (AQSA_SQ...	Running	Manual
SQL Full-text Filter Daemon Launcher (SQL_IAD)	Running	Manual
SQL Server Launchpad (MSSQLSERVER)	Running	Automatic
SQL Server Launchpad (AQSA_SQL_S)	Running	Automatic
SQL Server Launchpad (SQL_IAD)	Stopped	Automatic
SQL Server (MSSQLSERVER)	Running	Automatic
SQL Server Agent (AQSA_SQL_S)	Stopped	Other (Boot, Syst
SQL Server Agent (SQLEXPRESS)	Stopped	Other (Boot, Syst
SQL Server Agent (SQLEXPRESS01)	Stopped	Other (Boot, Syst
SQL Server Agent (SQL_IAD)	Stopped	Other (Boot, Syst
SQL Server Browser	Stopped	Other (Boot, Syst
SQL Server PolyBase Data Movement (SQL_IAD)	Stopped	Automatic
SQL Server PolyBase Engine (SQL_IAD)	Stopped	Automatic
SQL Server Agent (MSSQLSERVER)	Stopped	Other (Boot, Syst
SQL Server (MSSQLSERVER01)	Running	Automatic
SQL Server Agent (MSSQLSERVER01)	Stopped	Manual
SQL Server (MSSQLSERVER02)	Running	Automatic
SQL Server Agent (MSSQLSERVER02)	Stopped	Manual
SQL Server (MSSQLSERVER03)	Running	Automatic
SQL Server Agent (MSSQLSERVER03)	Stopped	Manual

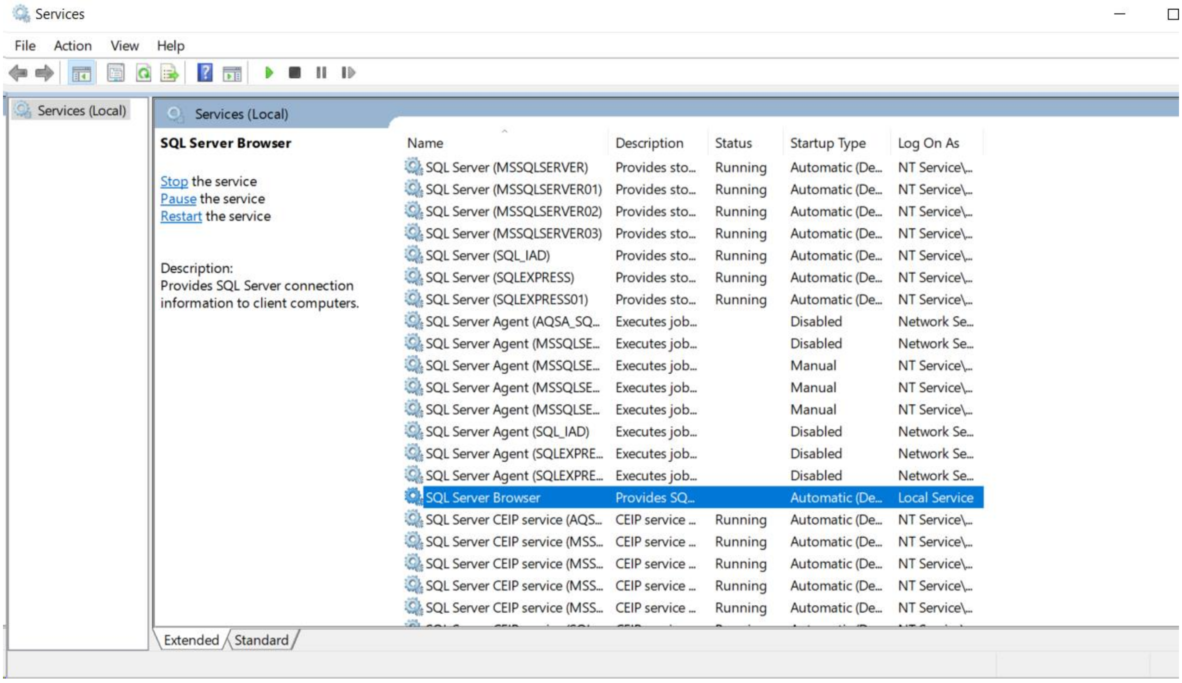
Now we will start the **SQL SERVER BROWSER**

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



Connecting SQL server with Golang

From here check the SQL browser is running or not:



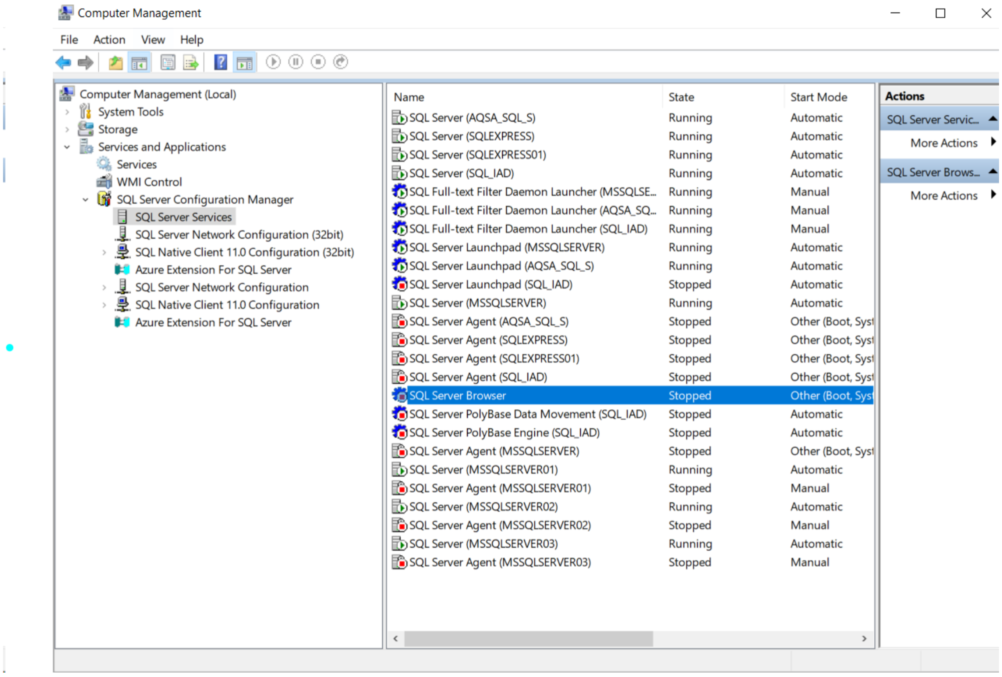
SQL Server Browser

Stop the service
Pause the service
Restart the service

Description:
Provides SQL Server connection information to client computers.

Name	Description	Status	Startup Type	Log On As
SQL Server (MSSQLSERVER)	Provides sto...	Running	Automatic (De...	NT Service\...
SQL Server (MSSQLSERVER01)	Provides sto...	Running	Automatic (De...	NT Service\...
SQL Server (MSSQLSERVER02)	Provides sto...	Running	Automatic (De...	NT Service\...
SQL Server (MSSQLSERVER03)	Provides sto...	Running	Automatic (De...	NT Service\...
SQL Server (SQL_IAD)	Provides sto...	Running	Automatic (De...	NT Service\...
SQL Server (SQLEXPRESS)	Provides sto...	Running	Automatic (De...	NT Service\...
SQL Server (SQLEXPRESS01)	Provides sto...	Running	Automatic (De...	NT Service\...
SQL Server Agent (AQSA_SQ...	Executes job...		Disabled	Network Se...
SQL Server Agent (MSSQLSE...	Executes job...		Disabled	Network Se...
SQL Server Agent (MSSQLSE...	Executes job...		Manual	NT Service\...
SQL Server Agent (MSSQLSE...	Executes job...		Manual	NT Service\...
SQL Server Agent (MSSQLSE...	Executes job...		Manual	NT Service\...
SQL Server Agent (SQL_IAD)	Executes job...		Disabled	Network Se...
SQL Server Agent (SQLEXPRE...	Executes job...		Disabled	Network Se...
SQL Server Agent (SQLEXPRE...	Executes job...		Disabled	Network Se...
SQL Server Browser	Provides SQ...		Automatic (De...	Local Service
SQL Server CEIP service (AQS...	CEIP service ...	Running	Automatic (De...	NT Service\...
SQL Server CEIP service (MSS...	CEIP service ...	Running	Automatic (De...	NT Service\...
SQL Server CEIP service (MSS...	CEIP service ...	Running	Automatic (De...	NT Service\...
SQL Server CEIP service (MSS...	CEIP service ...	Running	Automatic (De...	NT Service\...
SQL Server CEIP service (MSS...	CEIP service ...	Running	Automatic (De...	NT Service\...

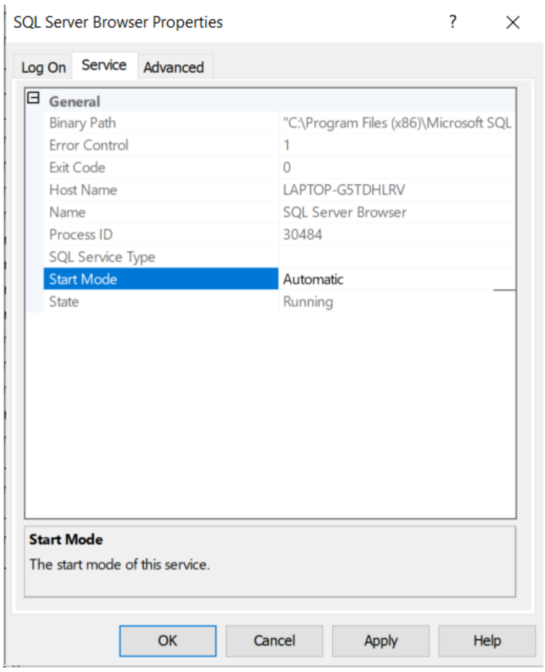
If it not running go to


Computer Management
File
Action
View
Help
Computer Management (Local)
System Tools
Storage
Services and Applications
Services
WMI Control
SQL Server Configuration Manager
SQL Server Services
SQL Server Network Configuration (32bit)
SQL Native Client 11.0 Configuration (32bit)
Azure Extension For SQL Server
SQL Server Network Configuration
SQL Native Client 11.0 Configuration
Azure Extension For SQL Server
Actions
SQL Server Servic...
More Actions
SQL Server Brows...
More Actions

Name	State	Start Mode
SQL Server (AQSA_SQL_S)	Running	Automatic
SQL Server (SQLEXPRESS)	Running	Automatic
SQL Server (SQLEXPRESS01)	Running	Automatic
SQL Server (SQL_IAD)	Running	Automatic
SQL Full-text Filter Daemon Launcher (MSSQLSE...	Running	Manual
SQL Full-text Filter Daemon Launcher (AQSA_SQ...	Running	Manual
SQL Full-text Filter Daemon Launcher (SQL_IAD)	Running	Manual
SQL Server Launchpad (MSSQLSERVER)	Running	Automatic
SQL Server Launchpad (AQSA_SQL_S)	Running	Automatic
SQL Server Launchpad (SQL_IAD)	Stopped	Automatic
SQL Server (MSSQLSERVER)	Running	Automatic
SQL Server Agent (AQSA_SQL_S)	Stopped	Other (Boot, Syst
SQL Server Agent (SQLEXPRESS)	Stopped	Other (Boot, Syst
SQL Server Agent (SQLEXPRESS01)	Stopped	Other (Boot, Syst
SQL Server Agent (SQL_IAD)	Stopped	Other (Boot, Syst
SQL Server Browser	Stopped	Other (Boot, Syst
SQL Server PolyBase Data Movement (SQL_IAD)	Stopped	Automatic
SQL Server PolyBase Engine (SQL_IAD)	Stopped	Automatic
SQL Server Agent (MSSQLSERVER)	Stopped	Other (Boot, Syst
SQL Server (MSSQLSERVER01)	Running	Automatic
SQL Server Agent (MSSQLSERVER01)	Stopped	Manual
SQL Server (MSSQLSERVER02)	Running	Automatic
SQL Server Agent (MSSQLSERVER02)	Stopped	Manual
SQL Server (MSSQLSERVER03)	Running	Automatic
SQL Server Agent (MSSQLSERVER03)	Stopped	Manual

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



SQL Server Browser Properties

Log On | Service | Advanced

General	
Binary Path	"C:\Program Files (x86)\Microsoft SQL
Error Control	1
Exit Code	0
Host Name	LAPTOP-G5TDHLRV
Name	SQL Server Browser
Process ID	30484
SQL Service Type	
Start Mode	Automatic
State	Running

Start Mode

The start mode of this service.

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:

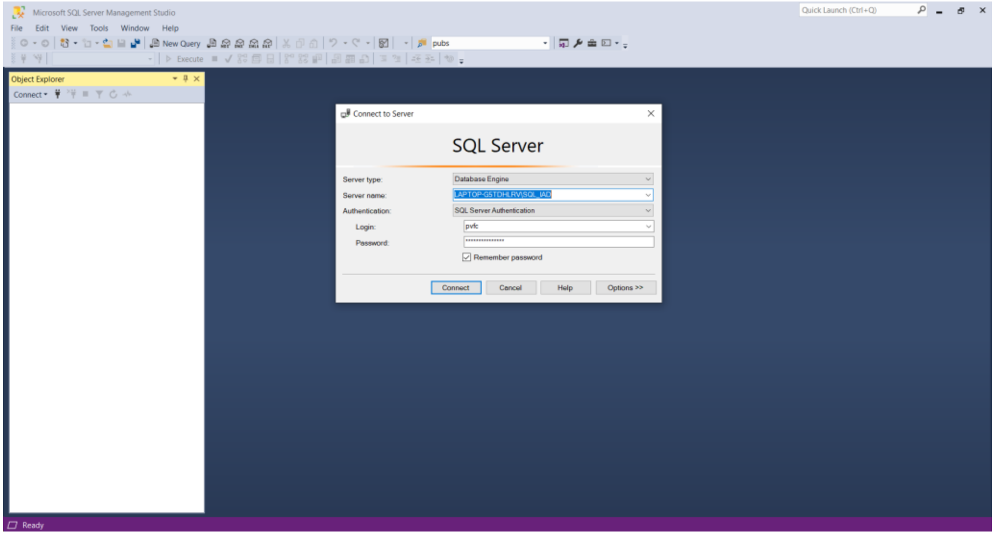
```
go mod init <EnterDummyNameForYourServer>
```

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):


Microsoft SQL Server Management Studio
File Edit View Tools Window Help
New Query
Execute
pubs
Quick Launch (Ctrl+Q)
Object Explorer
Connect
Ready

Connect to Server

SQL Server

Server type: Database Engine

Server name: LAPTOP-G5TDHLRV\SQL_IAD

Authentication: SQL Server Authentication

Login: pvfc

Password: ***************

☑ Remember password

Note:

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

Connecting SQL server with Golang

Connection String Pattern:

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

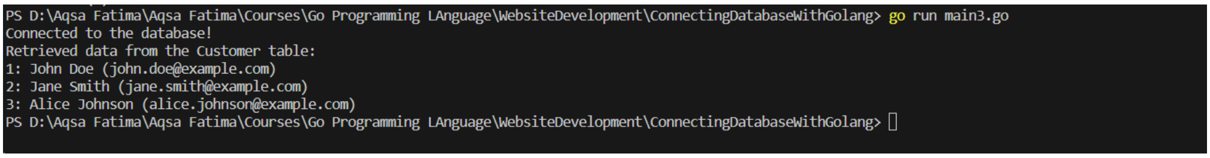
In SQL server Managemnet studio. Make the requied tables and insert data into and then connect with golang and try to retrieve the data to test your connection.

```
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:



```
PS D:\Aqsa Fatima\Aqsa Fatima\Courses\Go Programming LAnguage\WebsiteDevelopment\ConnectingDatabaseWithGolang> go run main3.go
Connected to the database!
Retrieved data from the Customer table:
1: John Doe (john.doe@example.com)
2: Jane Smith (jane.smith@example.com)
3: Alice Johnson (alice.johnson@example.com)
PS D:\Aqsa Fatima\Aqsa Fatima\Courses\Go Programming LAnguage\WebsiteDevelopment\ConnectingDatabaseWithGolang> 
```