# **Instructions for setting up the environment to work with JDBC and SQL SERVER**

## **Installing SQL Server (Recommend MS SQL Server 2019)**

SQL Server is software that runs on the server. It is highly stable software, operating almost 24/24 without errors. Usually, the servers also install Window Server. It is stable, and the software execution and installation rights management mechanism is much more stringent than our personal computers (PC). However, since we are developers before the application is launched, it is normal to use a PC. This also entails a lot of environmental problems that we must overcome. In general, the errors are diverse, with different Error codes, even the same Error code, but the solutions are also different.

So, if SQL Server is SERVER software, what is its client? That answer is the Web/Mobile/Desktop application that programmers will build. Here we use Java technology to make these applications. However, creating an application is too time-consuming, and in the process, we need to check if our SQL statements are working correctly or not. Therefore, we usually install SQL Server Management Studio (SSMS). SSMS is just a client software like the application we intend to develop. Still, the difference is that it is a product of Microsoft, and the function is only to execute commands to SQL Server on the graphical interface. Many students mistakenly believe that SSMS and SQL Server are the same. 😊 Now you see that it is not; without SSMS, SQL Server is still running, and without SQL Server, SSMS is useless.

### **SQL Server as Window Service**

Going back to SQL Server, this is a software that runs without an interface, it is installed on your computer as a Window Service. To manipulate it we just need to go to the RUN window > type services.msc

Graphical user interface, text, application, email

Description automatically generated

SQL Server is run under the name SQL Server (\*\*\*\*\*), \*\*\*\*\* is the instance name, we can have multiple installations of the same SQL Server instance as long as their instance names are different. together. The instance name does not have to be present. As shown above, there are 2 instances of SQL Server installed in the system. SQL Server service (MSSQLSERVER) is not START yet and therefore I cannot use SSMS or any client software to connect to it. However, the Express version is ok, can work with it already.

Graphical user interface, text, application, email

Description automatically generated

I can open SSMS to do a few things before working with JDBC. SQL Server allows 2 Authentication mechanisms: Window Authentication (Use the Windows user's certificate always), and SQL Server Authentication (use Username and password managed by SQL Server). JDBC version 4 only allows working with SQL Server Authentication. If you already have a sa (system admin) account when installing, skip to the next part (1.b), otherwise, you can follow to create an account with the same permissions as sa.

### **Create Account to be authenticated via SQL Server Authentication Mode**

First, you log into SQL Server from SSMS using Window authentication. You will almost certainly be able to log in; if the system reports an error, one likely cause is: Wrong Server name, and the not started SQL Server Service operation. Other errors are based on the error code.

Graphical user interface, text, application, email

Description automatically generated

Next step, on SSMS, check the Object Explorer pane, navigate to Security > Logins > right click New Login >

Graphical user interface, application

Description automatically generated

Remember to choose SQL Server Authentication mode. Just set a simple password, for example, 12345, because the system will change it; no need to put anything complicated. Don't click OK in haste.

Select Server Roles > check on the left-hand panel to add Sysadmin rights or whatever roles you want to assign. Then click ok to finish.

Graphical user interface, text, application

Description automatically generated

Continue, right-click the connection root item on the Object Explorer bar > select Properties > open the dialog select Security > Make sure that SQL Server and Window Authentication mode is selected.

Graphical user interface, text, application, email

Description automatically generated

Reopen windows services. Restart the service, then try to connect again with SQL Server Authentication mode. Change the password.

### **SQL Server Configuration** **Manager**

An application that uses JDBC library to connect to SQL Server only works via TCP/IP protocol and SQL Server Authentication mode, you have done a part of having an account to login to the system. Now you need to check if the SQL Server software allows connections via TCP/IP. To do this we need to open software called SQL Server Configuration Manager.

First check your SQL Server version is 2012, 2014, 17, or 19 to open the corresponding path of SQL Server Manager.

| **Version** | **Path** |
| --- | --- |
| SQL Server 2019 | C:\Windows\SysWOW64\SQLServerManager15.msc |
| SQL Server 2017 | C:\Windows\SysWOW64\SQLServerManager14.msc |
| SQL Server 2016 | C:\Windows\SysWOW64\SQLServerManager13.msc |
| SQL Server 2014 (12.x) | C:\Windows\SysWOW64\SQLServerManager12.msc |
| SQL Server 2012 (11.x) | C:\Windows\SysWOW64\SQLServerManager11.msc |

More detail at:

https://docs.microsoft.com/en-us/sql/relational-databases/sql-server-configuration-manager?view=sql-server-ver15

For example, if I install SQL Server 2019,

open RUN > C:\Windows\SysWOW64\SQLServerManager15.msc

and I can open the configuration for SQL Server on my machine. SQLServerManager15 looks like the below:

Graphical user interface, text, application

Description automatically generated

Find SQL Server Network Configuration > Protocols for [Instance name you're using – SQLExpress for me].

It can be seen in the figure that my TCP/IP is disabled; I just need to enable it. Then switch to the IP Addresses tab to set the port for SQL Server. Find the IP All group > TCP Port > set the port number; people usually prefer to use the default port of 1433. Select **OK, then Restart the SQL Server service**.

|  |  |
| --- | --- |
|  |  |

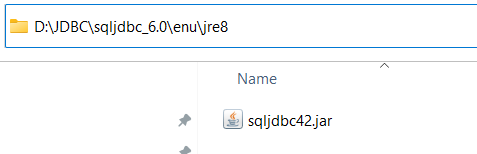
## **Install JDBC Driver and Check Connection.**

JDBC Driver has many versions, and each version is compatible with a different version of java. Which version to use? The instructor will guide you. Here I take an example using JDBC Driver for SQL Server 6.0.

Download: <https://www.microsoft.com/en-us/download/details.aspx?id=11774>

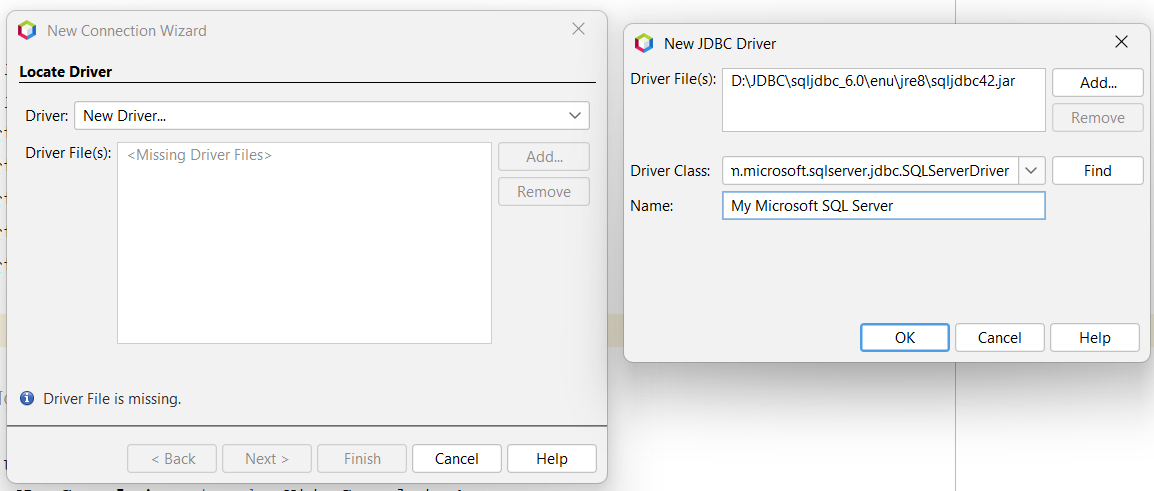
Remember to unzip and put it in an address where it's easy to find.

Since I'm using Java 8, I’m only interested in JRE 8; find the corresponding JAR file:



Your instructor will guide you through working with JDBC in detail, this document is for environmental testing purposes only. For a quick test, to avoid writing code, you can directly open NetBeans. On the menu bar select >Window > Service > Databases > right-click New Connection

On the Driver bar, select New Driver > Add the corresponding JDBC jar file.



Click Ok > Next > fill in the information as you logged into SSMS. Ignore the JDBC Url part, it's self-generated.

Graphical user interface, text, application, email

Description automatically generated

When finished, click Test Connection 🡪 if you see Connection Succeeded, the environment is ready. Otherwise, you can check with your instructor.