#### https://www.halvorsen.blog

# SQL Server and Visual Studio

Hans-Petter Halvorsen



#### Contents

- 1. Introduction
  - <u>SQL Server</u>
  - <u>Visual Studio</u>
- Code Examples:
- 2. Insert Data into Database
- 3. Get Data from Database

#### https://www.halvorsen.blog

#### SQL Server and Visual Studio

## Introduction

#### Hans-Petter Halvorsen



### Introduction

- We will use **SQL Server**, which is a Database System from Microsoft.
- We will create Applications in **Visual Studio** and **C#** that communicates with the SQL Server Database.
- We will create **Windows Forms** Desktop Applications.
  - Applications that **Writes Data** to the SQL Server.
  - Applications that **Reads Data** from the SQL Server.

### What is a Database?

- A Database is a structured way to store lots of information.
- The information inside the database is stored in different tables.
- - "Everything" today is stored in databases!

Examples:

- Bank/Account systems
- Information in Web pages such as Facebook, Wikipedia, YouTube, etc.
- Online Web Shops
- ... lots of other examples!

### **Database Systems**

We communicate with the Database using a Database Management System (DBMS). We use the **Structured Query Language (SQL)** to communicate with the Database, i.e., Insert Data, Retrieve Data, Update Data and Delete Data from the Database.



SQL – Structured Query Language

#### https://www.halvorsen.blog

#### SQL Server and Visual Studio

# SQL Server

#### Hans-Petter Halvorsen



### Install SQL Server Express

to SQL Server Installation Center			-	пх					
Planning Installation Maintenance Tools Resources	New SQL Server           Launch a wizabet         features to an exis           Image: Server         Launch a downloginternet connection	standalone installation or add fea o install SQL Server 2022 in a non-c sting SQL Server 2022 instance. Reporting Services a 🐒 SQL Server 2022 Setup	atures to an existing i	nstallation or to add			X	Mi	ved Mode is recommended
Options	<ul> <li>Install SQL Server 1</li> <li>Launch a downloa SQL Server Profiler install these tools.</li> <li>Install SQL Server 1</li> <li>Launch a downloa provides Visual Stu SQL Database, the Integration Service</li> <li>Upgrade from a pr Launch a wizard to <u>Click here to fir</u></li> </ul>	Instance Configuration Instance Configuration Specify the name and instance ID Is. Global Rules TP Product Updates a Install Setup Files Stu Install Rules Azure Extension for SQL Server Feature Selection PI Feature Rules to Instance Configuration file Server Configuration Database Engine Configuration Feature Configuration Rules	ID for the instance of S     O Default instance     Named instance: *	QL Server. Instance IE SQLExpress	D becomes part of t	he installation path.	SQL Server 20 Database Specify Data parallelism Licence Terms	-  -  -  -  -  -  -  -  -  -  -  -  -	
			Instance ID: SQL Server directory: Installed instances: Instance Name CITADEL	C:\Program Files\N Instance ID MSSQL11.CITADEL	licrosoft SQL Server Features SQLEngine	2L Server\MSSQL16.SQLEXPRESS es Edition jine Express	Global Rules Product Updates Install Setup File: Install Rules Azure Extension Ver 11.4 Feature Rules Instance Configu	ense terms obal Rules oduct Updates tall Setup Files itall Rules sure Extension for SQL Server ature Selection ature Rules	Server Configuration Data Directories TempDB Memory User Instances FILESTREAM Specify the authentication mode and administrators for the Database Engine. Authentication Mode Windows authentication mode Mixed Mode (SQL Server authentication and Windows authentication) Specify the password for the SQL Server system administrator (sa) account.
Microsoft SQL Server 2022		Installation Progress Complete					Server Configura Database Engine Feature Configur Installation Prog Complete	stion e Configuration ration Rules ress	Enter password: Confirm password: Specify SQL Server administrators SPCI SHPH\hansha (hansha) SQL Server administrators have unrestricted access to the Database Engine.
						< Back Next >			Add Current User Add Remove

#### Install SQL Server Management Studio

#### 1 SOL Server Installation Center

Planning

Tools

Installation

Maintenance

Resources

Options

X

#### New SQL Server standalone installation or add features to an existing installation

Launch a wizard to install SQL Server 2022 in a non-clustered environment or to add features to an existing SQL Server 2022 instance.

#### Install SQL Server Reporting Services

Launch a download page that provides a link to install SQL Server Reporting Services. An internet connection is required to install SSRS.

#### .... Install SQL Server Management Tools

Launch download page that provides a link o SOL Server command-line utilities (SOLCMD n SOL Server Profiler and Database Tuning Advisor install these tools.

#### Install SOL Server Data Tools

Launch a download page that provides a link to provides Visual Studio integration including pro SQL Database, the SQL Server Database Engine, Integration Services, An internet connection is re

#### Upgrade from a previous version of SQL Server

Launch a wizard to upgrade a previous version c Click here to first view Upgrade Documentati

Microsoft SQL Server 2022

#### Download SQL Server Management Studio

(SSMS)

Article • 03/13/2023 • 6 minutes to read • 48 contributors

#### Applies to: SOL Server Azure SOL Database Azure SOL

SQL Server Management Studio (SSMS) is an integrated environment Server to Azure SQL Database, SSMS provides tools to configure, ma and databases. Use SSMS to deploy, monitor, and upgrade the data build queries and scripts.

Use SSMS to query, design, and manage your databases and data w computer or in the cloud.

#### Download SSMS

SSMS 19.0.2 is the latest general availability (GA) version. If you have should uninstall it before installing SSMS 19.0.2. If you have SSMS 1 19.0.2.



Microsoft SQL Server Management Studio with Azure Data Studio

#### Welcome. Click "Install" to begin.

#### Location:

C:\Program Files (x86)\Microsoft SQL Server Management Studio 19

Change

By clicking the "Install" button, I acknowledge that I accept the Privacy Statement and the License Terms for SQL Server Management Studio and Azure Data Studio

SQL Server Management Studio transmits information about your installation experience, as well as other usage and performance data, to Microsoft to help improve the product. To learn more about data processing and privacy controls, and to turn off the collection of this information after installation, see the documentation

Install

Close

#### Login to SQL Server Management Studio

Login Connection Propertie Server Server ype:	SQL Server  s Aways Encrypted Additional Connection Parameters  Database Engine	Login Connection Propertie Server Server type: Server name:	Here the "sa" user is use installation of SQL Serve for the "sa" user. In gene the SQL Server Managen	d. If you select " r Express, you n ral, better to cre nent Studio and	Mixed mode" during eed to specify the Pa ate additional SQL u use that instead of "	ssword sers in sa".
Server name: Authentication: Login: Password:	HANS-PETTER\SQLEXPRESS  SQL Server Authentication GOL Server Authentication Microsoft Entra MFA Microsoft Entra Integrated Microsoft Entra Integrated Microsoft Entra Integrated Identity	Authentication: Login: Password: Connection Security	SQL Server Authentication	교환 Connect to Server	SQL Server	×
Connection Security	Morosoft Entra Default Optional  Trust server certificate Connect Cancel Help Options <<	Host name in certificate:	Trust server certificate	Server Server type: Server name: Authentication: User name:	Database Engine HANS-PETTER\SQLEXPRESS Windows Authentication HANS-PETTER\hansp	~ ~ ~
Choose be • SQL Se • Windo Windo authen	etween: erver Authentication ws Authentication (the o ws account credentials a utication)	current re used for		Password: Connection Security Encryption: Host name in certificate	Remember password       Optional       Trust server certificate       ::       Connect     Cancel	Options <<

### **SQL Server Management Studio**



### Structured Query Language (SQL)

- Structured Query Language (SQL) is used to write, read and update data from the Database System
- You can use SQL inside the "SQL Server Management Studio" or inside your Visual Studio C# App.
- SQL Example: "select \* from SCHOOL"

### **SQL** Examples

Query Examples:

- insert into STUDENT (Name , Number, SchoolId) values ('John Smith', '100005', 1)
- select SchoolId, Name from SCHOOL
- **select** \* from SCHOOL where SchoolId > 100
- update STUDENT set Name='John Wayne' where StudentId=2
- **delete** from STUDENT **where** SchoolId=3

We have 4 different Query Types: INSERT, SELECT, UPDATE and DELETE

CRUD: C – Create or Insert Data, R – Retrieve (Select) Data, U – Update Data, D – Delete Data

#### Database

We will create a Database called "SENSORSYSTEM" and create the following Table:

CREATE TABLE SENSOR
(
 SensorId int NOT NULL IDENTITY (1,1),
 SensorName varchar(50) NOT NULL,
 SensorType varchar(50) NOT NULL
)

Note! This is a very simplified example, typically we create multiple tables.

#### https://www.halvorsen.blog

#### SQL Server and Visual Studio

# Visual Studio

#### Hans-Petter Halvorsen



### Windows Forms Application

#### Create a new project

#### Recent project templates

C#

C#

- SI ASP.NET Core Web App (Razor Pages) C#
- 🗉 Windows Forms App
- 🛎 Console App
- Setup Project
- Windows Forms App Visual Basic
- S MSTest Test Project C#
- <sup>™</sup> Windows Forms App (.NET Framework) C#

- 🗆 ×
Windows Forms × Clear all
All langua · All platfor · All projec ·
Windows Forms       App         A project template for creating a .NET       Windows Forms (WinForms) App.         C#       Windows         Desktop
<ul> <li>Windows Forms App</li> <li>A project template for creating a .NET Windows Forms (WinForms) App.</li> <li>Visual Basic Windows Desktop</li> </ul>
Windows Forms App (.NET Framework) A project for creating an application with a Windows Forms (WinForms) user interface C# Windows Desktop
Windows Forms Control Library (.NET Framework) A project for creating controls to use in Windows Forms (WinForms) applications C# Windows Desktop Library
Windows Forms App (.NET Framework) A project for creating an application with a Windows Forms (WinForms) user interface
Visual Basic Windows Desktop
Back

#### Microsoft.Data.SqlClient

File Edit View Git Project B	uild Debug Test Analyze Tools Extensions Window Help <sup> </sup> ● Search · SensorSys Pl- ▶ SensorSystem · ▶ ♂ ·   ■   ■	tem	🔮 – □ × & GitHub Copilot ≌ 🖗
<ul> <li>File Edit View Git Project B</li> <li>B</li> <li>Colbox</li> <li>General</li> <li>There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.</li> </ul>	uild Debug Test Analyze Tools Extensions Window Help P Search SensorSystem       SensorSystem         PI- SensorSystem       SensorSystem         NuGtem       Formign         Browse       Installed       Updates         Microsoft.Data.       Includ         Microsoft.Data.       Includ         Microsoft.Data.SqlClient * by Microsoft, nugetsqltools, 816M do The current data provider for SQL Server and Azure SQL databases       5.0.1         Microsoft.Data.SqlClient.SNI.runtime * by Microsoft, nuget       6.0.2 ↓         Internal implementation package not meant for direct consumption       9.0.3         Microsoft.Data.Sqlite.Core * by aspnet, dotnetframework, Entity       9.0.3         Microsoft.Data.Sqlite is a lightweight ADO.NET provider for SQLite. T       5.8.5         This package version is deprecated.       Microsoft.Data.Conta * by Microsoft, OData, 177M downloads       5.8.5         Classes to represent, construct, parse, serialize and validate entity d       5.8.5         LINQ-enabled client API for issuing OData queries and consuming O       Microsoft.Data.Sqlite * by aspnet, dotnetframework, EntityFrame * 9.0.3         Microsoft.Data.Sqlite is a lightweight ADO.NET provider for SQLite.       Microsoft.Data.Sqlite * by aspnet, dotnetframework, EntityFrame * 9.0.3         Microsoft.Data.Sqlite * by aspnet, dotnetframework, EntityFrame * 9.0.3       Microsoft.Data.Sqlite * by aspnet, dotnetframework, E	stem          Sensstem * * * *         kage Manager: SensorSystem         Package source: [nuget.org] *         * Microsoft.Data.Sq * nuget.org         * Microsoft.Data.Sq * nuget.org         * Microsoft.Data.Sq * nuget.org         * Configure         • Options         README         Reader Details         * Corosoft SqlClient Data Provider for SQL Server         Microsoft.Data.SqlClient ic opponents which live independently in MFT framework and NFT Core. Going forward, support for rew SQL Server and Azare SQL features will only be implemented in Microsoft.Data.SqlClient.         Supportability	<ul> <li>GitHub Copilot: # R</li> <li>Solution Explorer</li> <li>Search Solution Explorer (Ctrl+" P)</li> <li>Solution 'SensorSystem' (1 of 1 p</li> <li># Dependencies</li> <li># Analyzers</li> <li># Form1.cs</li> <li># Form1.Designer.cs</li> <li># Form1.resx</li> <li># or Program.cs</li> <li>GitHub Copilot Solution Explorer</li> <li>Properties</li> <li>*</li> </ul>
	<ul> <li>Microsoft.Extensions.Configuration.Binder <sup>o</sup> by aspnet, d 9.0.3 Provides the functionality to bind an object to data in configuration p</li> <li>Microsoft.EntityFrameworkCore <sup>o</sup> by aspnet, dotnetframewor 9.0.3 Entity Framework Core is a modern object-database mapper for .NET</li> <li>Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.</li> <li>Don't show this again</li> </ul>	implemented in Microsoft.Data.SqlClient. Supportability The Microsoft.Data.SqlClient package supports the following environments: NET Framework 4.6.2+NET 8.0+ Download	(2) 94   10
	Output Show output from: 이보는 도프 프 = 호 · 이 Developer PowerShell Error List Output	- * x	

📲 Q 🐲 📽 💽 🐂 Ք 💁 🚳 🔞 🖬 🖉 🧏 🧿 🦻 🔺 📾 🖪 兄 🚳 🖛 🛞 🖓 📾 🖾 🦷

https://www.halvorsen.blog

#### SQL Server and Visual Studio

# Insert Data into Database

Hans-Petter Halvorsen



### "Write Sensor Data" App

Sensor System		_		×
Sensor Name:				
Sensor1				
				_
Sensor Type:				
Temperature				
				_
			Save	
	Sensor System Sensor Name: Sensor1  Sensor Type: Temperature	Sensor System Sensor Name: Sensor1  Sensor Type: Temperature	Sensor System — Sensor Name: Sensor1 Sensor Type: Temperature	Sensor System – – – – – – – – – – – – – – – – – – –

The following SQL query will be executed in the C# code:

INSERT INTO SENSOR (SensorName, SensorType) VALUES ('Sensor1', 'Temperature')

#### Database

- SQLQuery2.sql - HANS-PETTER\SQLEXPRESS.SENSORSYSTEM (sa (59))* - Microsoft SQL Server	Management Studio Quick Launch (Ctrl+( P = □ >					
File Edit View Query Project Tools Window Help						
🖸 🗸 ා 🖄 🗸 🖆 🛀 🚆 🚇 New Query 🚇 බ බ බ බ බ 🖓 👗 🗗 බ	- ? - 図 - ♬ - ♬ - □ - □ - □ - □ - □ - □ - □ - □					
F 🐨 SENSORSYSTEM - ▷ Execute = ✔ 🕾 🗐 🗟 🕾 🕮						
Object Explorer	SQLQuery2.sM (sa (59))* * ×					
Connect- # ₩ = T C +	select * from SENSOR					
B HANS-PETTER\SQLEXPRESS (SQL Server 16.0.1135 - sa)						
Databases						
System Databases						
Database Snapshots						
BOOKS						
■ ■ ORDERS						
■ SENSORSYSTEM						
🗉 🛑 Database Diagrams						
🗉 🛑 Tables						
🗉 💻 System Tables						
🗉 🖷 FileTables	CREATE TABLE SENSOR					
🗉 💻 External Tables						
🗉 💻 Graph Tables						
■      ■ dbo.SENSOR	SensorId int NOT NULL IDENTITY (1,1),					
🗉 🖷 Columns	SensorName varchar(50) NOT NULL					
🗄 SensorId (int, not null)						
SensorName (varchar(50), not null)	SensorType varchar(50) NOT NULL					
SensorType (varchar(50), not null)						
🖲 💻 Keys						
🗉 💻 Constraints						
🖲 💻 Triggers						
🗉 💻 Indexes						
🗉 💻 Statistics						
🗉 💻 Views						
🗉 💻 External Resources	150 % - I HANE DETTED SOLEVEDER (50) SENCORVETEM (00000)					
C	To connected (1/1) THANS-PETER/SQLEAPRESS (10					

Ready

#### **Visual Studio**

00	File Edit View Git Project B	uild Debug Format	Test Analyze	Tools Extension	ns Window	Help 🛛 🕫 Search 🛛 🛛 Wr 🔮 🕘 🗆 🛛 🛛
e	- ) 御・ 🛎 🖽 📾 🖻 - 🤉 - 🤇 - 🛛 Deb - Any C	Pl - > SensorSystem -	▷ ∅ •   ➡   ➡   ➡   ■		0.0822	ෂ GitHub Copilot ය 🖉
		For ignl * x				Solution Explorer
ata	Search Toolbox					
So	SalCommandBuilder (	Sensor System				Search Solution Explorer (Ctrl+" P-
urc	SqlConnection (Micro	Sensor Name:				Solution 'WriteSensor' (1 of 1 pro
les	SqlConnection (Micro					🖌 🖻 SensorSystem
	All Windows Forms	Sensor Type:	•			▶ ﷺ Dependencies
	Pointer					▹ I Form1.cs
	BackgroundWorker		Save			▷ c# Program.cs
	BindingSource					
	Button					
	CheckBox					
	E CheckedListBox					
	🛯 ColorDialog					
	🖻 ComboBox					GitHub Conilot Solution Explorer
	ContextMenuStrip					Descetter
	DataGridView					Properties • * ×
	DateTimePicker					Form1 System. Windows.Forms.Fc*
	DomainUpDown					EormBorderStyl Sizable
	ErrorProvider					RightToLeft No
	FileSystemWatcher					RightToLeftLavo False
	FlowLayoutPanel	Output				× Text Sensor System
	FolderBrowserDialog	Show output from:		• <u>€</u> = ≥ ×= ¢	G	UseWaitCursor False
	FontDialog					Text
	🖱 GroupBox					The text associated with the con
	HelpProvider	Developer PowerShe	ell Error List <mark>Ou</mark>	utput		
🖵 F	Ready		15,15	<i>i</i> ≝ 399 x 242	↑ Add to So	urce Control 🕤 🗉 Select Repository 🛀 🔒

### **Connection String**

The Connection String to connect to the Database can be written in many ways. Here is some examples (There are many other ways also):

Windows Authentication (the current Windows account credentials are used for authentication):

string connectionString = "Server=Hans-Petter\\SQLEXPRESS; Database=SENSORSYSTEM; Integrated Security=True;

TrustServerCertificate=True";

#### SQL Server Authentication:

string connectionString = "Server=Hans-Petter\\SQLEXPRESS; Database=SENSORSYSTEM; Uid=sa; Pwd=YourPassword; TrustServerCertificate=True";
Here the "sa" user is used. If you select "Mixed mode" during installation of SQL Server Express, you need to specify the Password for the "sa" user. In general, better to create additional SQL users in the SQL Server Management Studio and use that instead of "sa".



File Edit	/iew Git Project Build Debug Test Analyze Tools Extensions Window Help <sup>  ø Search ·</sup> WriteSensor	€ - □ ×
) • 🛞 👘 • 🗃 🖪	/ ウ・ペ・ <mark>Deb・Any CPI・</mark> ▶ SensorSystem • ▶ ダ・ ▶ 爾 賞 ♥  № 確 国 階 및 母 和 国 <sub>=</sub>	🕫 GitHub Copilot 🖻
Form1.cs	• × Formign]	Solution Expl 👻 🖲 🛛
=SensorS	stem • • SensorSystem.Form1 • • • txtSensorType • •	
1	vusing System;	Search Solution Exp &
2	using System.Windows.Forms;	Solution 'WriteSens
3	using Microsoft.Data.SqlClient;	🖌 🖻 SensorSystem
4		M Dependencies
5	namespace SensorSystem	▹
6	1 Stetemas	Program.cs
<sup>81</sup> 7	public partial class Form1 : Form	
8		
9	<pre>/ public Form1()</pre>	
10		
11	<pre>InitializeComponent();</pre>	
12	}	
13		
14	<pre>private void btnSave Click(object sender, EventArgs e)</pre>	
15		
16	<pre>string sensorName = txtSensorName.Text;</pre>	
17	<pre>string sensorType = txtSensorType.Text;</pre>	CitHub C Colution
18		GILHUD C Solution
19	<pre>string connectionString = "Data Source=Hans-Petter\\SQLEXPRESS;Initial Catalog=SENSORSYSTEM;Integrated Security=True;TrustServerCertificate=True";</pre>	Properties - 🖲 ×
20		
21	string sqlQuery = "INSERI INTO SENSOR (SensorName, SensorType) " +	III 9↓ <i>P</i>
22	VALUES ( + + sensorwame + + , + + sensor ype + + );	
23	SalConnection con = new SalConnection(connectionString);	
25	Squeetien con new squeetien connections in many	
26	con.Open();	
27	SqlCommand sc = new SqlCommand(sqlQuery, con);	
28	<pre>sc.ExecuteNonQuery();</pre>	
29	con.Close();	
30	}	
31	}	
32		
100 % • @ 🛛	issues found ☆ ◆ ↓ ↓ 1::32 Ch:2 SPC CALF	

🗆 Ready

using System; using System.Windows.Forms; using Microsoft.Data.SqlClient;

namespace SensorSystem

public partial class Form1 : Form

#### public Form1()

InitializeComponent();

```
ß
```

```
private void btnSave_Click(object sender, EventArgs e)
```

```
string sensorName = txtSensorName.Text;
string sensorType = txtSensorType.Text;
```

```
string connectionString = "Data Source=Hans-Petter\\SQLEXPRESS;Initial Catalog=SENSORSYSTEM;Integrated
Security=True;TrustServerCertificate=True";
```

```
string sqlQuery = "INSERT INTO SENSOR (SensorName, SensorType) " +
    "VALUES (" + """ + sensorName + """ + "," + """ + sensorType + """ + ")";
```

SqlConnection con = new SqlConnection(connectionString);

```
con.Open();
SqlCommand sc = new SqlCommand(sqlQuery, con);
sc.ExecuteNonQuery();
con.Close();
```

Form1.cs

### Step 2: Create Method

private void btnSave\_Click(object sender, EventArgs e)

SaveData();

Form1.cs: Here is a separate Method "**SaveData()**" is made to improve Code Quality

private void SaveData()

```
string sensorName = txtSensorName.Text;
string sensorType = txtSensorType.Text;
```

string connectionString = "Data Source=Hans-Petter\\SQLEXPRESS;Initial Catalog=SENSORSYSTEM;Integrated Security=True;TrustServerCertificate=True";

```
SqlConnection con = new SqlConnection(connectionString);
con.Open();
SqlCommand sc = new SqlCommand(sqlQuery, con);
sc.ExecuteNonQuery();
con.Close();
```

### Step 3a: Create a Class

using Microsoft.Data.SqlClient;

namespace SensorSystem.Classes

#### class Sensor

Here is a separate **Class "Sensor"** and a Method **"SaveSensorData()"** made to improve Code Quality

```
public void SaveSensorData(string sensorName, string sensorType)
```

```
string connectionString = "Data Source=Hans-Petter\\SQLEXPRESS;Initial Catalog=SENSORSYSTEM;Integrated
Security=True;TrustServerCertificate=True";
```

```
string sqlQuery = "INSERT INTO SENSOR (SensorName, SensorType)" +
    "VALUES (" + """ + sensorName + """ + "," + """ + sensorType + """ + ")";
```

SqlConnection con = new SqlConnection(connectionString); con.Open(); SqlCommand sc = new SqlCommand(sqlQuery, con); sc.ExecuteNonQuery(); con.Close();

Solution Explorer	ф
,ª   ™ → ≪ @   <mark>™</mark> +   <b>⊁</b> ≓	
Search Solution Explorer (Ctrl+")	
Import Solution 'WriteSensor3' (1 of 1 p Import 1 of 1 p Imp	or
🖌 🗉 SensorSystem	
Bependencies	
🖌 🗖 Classes	
▷ ☞ Sensor.cs	
Forml.cs	
C# Program.cs	

### Step 3b: Use the Class

private void btnSave\_Click(object sender, EventArgs e)

SaveData();

}

private void SaveData()

string sensorName = txtSensorName.Text;
string sensorType = txtSensorType.Text;

Sensor sensor = new Sensor();

sensor.SaveSensorData(sensorName, sensorType);

Then we use the Class and Method in "Form1.cs"

Solution Explorer
,∄   <sup>™</sup> ⊙ • ≒ ⊗ 10   <sup>™</sup> ₀ •   ≯ ≓
Search Solution Explorer (Ctrl+")
□ Solution 'WriteSensor3' (1 of 1 pr
🖌 🖻 SensorSystem
P & Dependencies
🖌 🗖 Classes
Sensor.cs
▶ I Form1.cs
C# Program.cs

#### Step 4a: Create Stored Procedure

Create Stored Procedure "SaveSensor" in SQL Server Management Studio:

```
IF EXISTS (SELECT name
    FROM sysobjects
    WHERE name = 'SaveSensor'
    AND type = 'P')
DROP PROCEDURE SaveSensor
GO
```

```
CREATE PROCEDURE SaveSensor
@SensorName varchar(50),
@SensorType varchar(50)
AS
```

INSERT INTO SENSOR (SensorName, SensorType) VALUES (@SensorName, @SensorType)
GO

#### Step 4a: Create Stored Procedure

SaveSensor.sql - HANS-PETTER\SQLEXPRESS.SENSORSYSTEM (sa (68)	) - Microsoft SQL Server Management Studio Quick Launch (Ctrl+1 P - 🗖 🗙
File Edit View Query Project Tools Wi	ndow Help
C · C · C · C · C · C · C · C · C · C ·	a @ @   × @ a   2 · < ·   Ø   ·   # · · · · · · · · · · · · · · · ·
F 🐄 SENSORSYSTEM - ▷ Execute = ✔ 🕾	■ 2 2 2 1 周囲 む ほ 4 4 5 4 4 5
Object Explore: X	SaveSensorEM (sa (68)) • ×
Connect- 🛱 🎁 🗏 🝸 🖒 🚸	PIF EXISTS (SELECT name
🗉 🗟 HANS-PETTER\SQLEXPRESS (SQL Server	FROM sysobjects
🗉 📫 Databases	
🗉 💻 System Databases	
Database Snapshots	AND type = 'P')
	DROP PROCEDURE SaveSensor
	Γ <sup>6</sup> GQ
Database Diagrams	
<ul> <li>Tables</li> </ul>	
🗉 🖷 System Tables	PCREATE PROCEDURE SaveSensor
Image: Bild of the second s	<pre>@SensorName varchar(50),</pre>
🗉 🛑 External Tables	SensorType varchar(50)
🗉 🛑 Graph Tables	
🗉 🎟 dbo.SENSOR	AS
🗉 🔲 Views	
External Resources	INSERT INTO SENSOR (SensorName, SensorType) VALUES (@SensorName, @SensorType)
Brogrammability	
System Stored Procedures	GO
🗉 🖬 dbo.SaveSensor	
- Functions	
🗉 💻 Database Triggers	
🗉 📫 Assemblies	
🗄 📫 Types	
Rules	
Ouery Store	
B = Service Broker	
	Vis Connected(1/1)         HANS-PETTER/SQLEXPRESS (16, 1a (68) SENSORSYSTEM 00:00:00 0 rows
□ Ready	Ln 8 Col 28 Ch 28 INS

### Step 4b: Use Stored Procedure

using System.Data; using Microsoft.Data.SqlClient;

namespace SensorSystem.Classes

```
class Sensor
```

public void SaveSensorData(string sensorName, string sensorType)

```
string connectionString = "Data Source=Hans-Petter\\SQLEXPRESS;Initial
Catalog=SENSORSYSTEM;Integrated Security=True;TrustServerCertificate=True";
```

SqlConnection con = new SqlConnection(connectionString); con.Open();

SqlCommand cmd = new SqlCommand("SaveSensor", con); cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@SensorName", sensorName)); cmd.Parameters.Add(new SqlParameter("@SensorType", sensorType));

cmd.ExecuteNonQuery(); con.Close(); Sensor.cs

### Step 5a: Create App.config

•• File Edit \ ◎ -	View Git Project Build Debug XML Test Analyze Tools Extensions Window Help ■ ♡ - ♡ -   Deb -   Any CPl - ▶ SensorSystem - ▷ ♂ -   ■   ■ _		eSensor5	€ – □ × ® GitHub Copilot <sup>⊮</sup> R
Appfi	<pre>g * * version="1.0" encoding="utf-8" ?&gt; iguration&gt; nnectionStrings&gt; add name="DatabaseConnectionString" connectionString="Data Source=Hans-Petter\SQLE&gt; roviderName="System.Data.SqlClient" /&gt; onnectionStrings&gt; figuration&gt;</pre>	(PRESS;Initi	Solution Explorer Search Solution Ex Solution 'WriteS' Solution 'WriteS' SensorSystem Mappendencie Classes App.config Form1.cs Corrections Corrections	به به x plorer (Ctrl+``) ه ensor5' (1 of 1 project n s
	We will create an "App.config" for and put the Connection String inside that file.		GitHub Copilot Cha Properties XML Document Structure Misc Encoding Output Schemas Stylesheet	t Solution Explorer
Developer	eves found	Ln:1 Ch:1 SPC CRLF → 0 × ↑ Add to S	<b>Encoding</b> Character encodin	g of the document.

### App.config

```
<?xml version="1.0" encoding="utf-8" ?> <configuration>
```

<connectionStrings> <add name="DatabaseConnectionString" connectionString="Server=Hans-Petter\SQLEXPRESS; Database=SENSORSYSTEM;Integrated Security=True;TrustServerCertificate=True" providerName="System.Data.SqlClient" /> </connectionStrings>

</configuration>

### Step 5b: Use App.config

using System.Data; using Microsoft.Data.SqlClient; using System.Configuration;

namespace SensorSystem.Classes

class Sensor

public void SaveSensorData(string sensorName, string sensorType)

string connectionString = ConfigurationManager.ConnectionStrings["DatabaseConnectionString"].ConnectionString;

SqlConnection con = new SqlConnection(connectionString); con.Open();

```
SqlCommand cmd = new SqlCommand("SaveSensor", con);
cmd.CommandType = CommandType.StoredProcedure;
```

cmd.Parameters.Add(new SqlParameter("@SensorName", sensorName)); cmd.Parameters.Add(new SqlParameter("@SensorType", sensorType));

cmd.ExecuteNonQuery(); con.Close();



https://www.halvorsen.blog

#### SQL Server and Visual Studio

# Get Data from Database

Hans-Petter Halvorsen

**Table of Contents** 

### Get Data from Database

- We will create different Applications:
- a) Get Data into TextBoxes
- b) Get Data into into a ListBox and a ComboBox
- c) Get Data into into a DataGridView

### "Read Sensor Data" App

•	Sensor System	_		×	
	Sensor Name: Sensor1				
	Sensor Type:			_	
	Temperature		Deed		When clicking "Read", data fro
		L F	Read	J	shown in the TextBoxes.

The following SQL query will be executed in the C# code:

SELECT SensorName, SensorType FROM SENSOR WHERE SensorId = 1

#### Database

SQLQuery1.sql - HANS-PETTER\SQLEXPRESS.SENSORSYSTEM (sa (59))* - Microsoft SQL Server	Quick Launch (Ctrl+( 🎤 🗕 🗖 🗙			
File Edit View Project Tools Window Help				
💿 - ㅇ   웹 - 🖆 - 🚔 💾 🦉 🔎 New Query 🚇 📾 📾 📾 🛣 🖉 🗇 🌐	9-9-8-	<b>م</b>	- 10	★ 益 □
¥ ₩ SENSORSYSTEM · ▷ Execute = ✓ ಔ @ 🗟 ೫ ಔ 🕮		· ==   *@ <sub>=</sub>		•
Object Explorer 👻 👻 🛪	SQLQuery1.sN	4 (sa (59))*		
Connect• # ¥ = ⊤ ♂ ↔	sele	ct * from	I SENSO	R
🛛 🗟 HANS-PETTER\SQLEXPRESS (SQL Server 16.0.1135 - sa)				
🗉 💻 Databases				CREATE TARLE SENSOR
🛚 💻 System Databases				
🗉 💻 Database Snapshots				(
BOOKS				SensorId int NOT NULL IDENTITY (1,1),
				Sensorname Varchar(50) NOT NULL,
Database Diagrams				SensorType varchar(50) NOT NULL
Iables				
System radies     FileTables				
			-	<b>*</b>
Graph Tables	150 % - 4			
B B do SENSOR		d SensorName	SensorTyp	
	1 1	Sensor1	Temperatu	
E Sensorid (int, not null)	2 2	Sensor2	Temperatu	
I SensorName (varchar(50), not null)	3 3	Sensor3	Pressure	
I SensorType (varchar(50), not null)	4 4	Sensor4	Level	
🗉 💻 Keys	5 5	Sensor5	Temperatu	re
🗉 💻 Constraints	6 6	Sensor6	Pressure	
🖲 📁 Triggers	7 7	Sensor7	Temperatu	re
🗉 📁 Indexes	8 8	Sesnor8	Temperatu	re
🗉 📁 Statistics				
🗉 📁 Views				
🗉 💻 External Resources				
	Query executed succ	esstully.		MAINS-PETTER/SQLEAPRESS (10 58 (39) SENSORSTSTEIN OUROUD TOTOWS

#### **Windows Forms Application**

Create a new project Recent project templates	Windows Forms     ×     Clear all       All langua     ·     All platfor     ·       Cliphic control     ·     ·     ·				
SP.NET Core Web App (Razor Pages)     C#     Windows Forms App     C#	A project template for creating a .NET W C# Windows Desktop	/indows Forms (WinForm	ns) App.		
Cmm Console App Cmm	Visual Basic Windows Desktop	Solution File Edit View Git Project B Solution Stress Solution Control So	uilid Debug Test Analyze Tools Extensions Window Help <sup>™</sup> Search <sup>-</sup> SensorSy PI ► SensorSystem - ▷ ♂ - ■ □ □	stem	€ - □ × ® GitHub Copilot <sup>⊮</sup> ₽
<ul> <li>Windows Forms App Visual Basic</li> <li>MSTest Test Project C#</li> <li>Windows Forms App (.NET Framework)</li> </ul>	<ul> <li>Windows Forms App (.NET Framework) A project for creating an application wit user interface</li> <li>C# Windows Desktop</li> <li>Windows Forms Control Library (.NET F A project for creating controls to use in applications</li> <li>C# Windows Desktop Library</li> <li>Windows Forms App (.NET Framework) A project for creating an application wit user interface</li> <li>Visual Basic Windows Desktop</li> </ul>	Search Toolbox // General // There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.	Browse       Installed       Updates       NuGet Pac         Microsoft.Data.       Includ         Microsoft.Data.SqlClient* by Microsoft, nugetsqltools, 816M do 6.0.1         The current data provider for SQL Server and Azure SQL databases         Microsoft.Data.SqlClient.SNL:runtime* by Microsoft, nuget 6.0.2 +         Internal implementation package not meant for direct consumption         Microsoft.Data.SqlClient.SNL:runtime* by Microsoft, Data.SqlClient.SNL:runtime* by Microsoft, Data.SqlClient.SNL:runtime* by Microsoft, Data.SqlClient.SNL:runtime* by Microsoft.Data.SqlClient.SNL:runtime* by Microsoft, Data.SqlClient*         Microsoft.Data.SqlClient.SNL:runtime* by Microsoft, Data.SqlLet.T         Microsoft.Data.SqlLet by Microsoft, OData, 177M downloads       45.8.5         This package version is deprecated.         Microsoft.Data.SqlLe* by spent, dontefframework, EntityFame       5.8.5         Classes to represent, construct, parse, serialize and validate entity d       5.8.5         Microsoft.Data.SqlLe* by spent, dontefframework, EntityFame       9.0.3         Microsoft.Data.SqlLe* by spent, dontefframework, EntityFame       9.0.3         Microsoft.Extensions.Configuration.Binder* by spent, dontefframework       9.0.3         Provides the functionality to bind an object to data in configuration p       Microsoft.EntityFrameworkCore * a spent, dontefframework       9.0.3         Entity Framework Core is a modern object-database mapper for	Address Address Stander Standers Stande	Solution Explorer Search Solution Explorer (Ctrl+ * /r) * Solution SensorSystem (1 of 1 p * Solution SensorSystem) * Fanalyzers * # Frameworks * # Frameworks * # Form 1.0es * # Form 1.0es * # Form 1.0es * # Program.cs * # Program.cs
					DNG 0.01 (m 153)

### **Connection String**

The Connection String to connect to the Database can be written in many ways. Here is some examples (There are many other ways also):

Windows Authentication (the current Windows account credentials are used for authentication):

string connectionString = "Server=Hans-Petter\\SQLEXPRESS; Database=SENSORSYSTEM; Integrated Security=True;

TrustServerCertificate=True";

#### SQL Server Authentication:

string connectionString = "Server=Hans-Petter\\SQLEXPRESS; Database=SENSORSYSTEM; Uid=sa; Pwd=YourPassword; TrustServerCertificate=True";
Here the "sa" user is used. If you select "Mixed mode" during installation of SQL Server Express, you need to specify the Password for the "sa" user. In general, better to create additional SQL users in the SQL Server Management Studio and use that instead of "sa".

#### Code

using System; using System.Windows.Forms; using Microsoft.Data.SqlClient;
name space SensorSystem {
public partial class Form1 : Form { public Form1()
{     InitializeComponent(); }
private void btnRead_Click(object sender, EventArgs e) {
<pre>string connectionString = "Server=Hans-Petter\\SQLEXPRESS;Initial Catalog=SENSORSYSTEM;Integrated Security=True;TrustServerCertificate=True";</pre>
string sqlQuery = "select SensorName, SensorType from SENSOR WHERE SensorId=1";
con.Open();
SqlCommand cmd = new SqlCommand(sqlQuery, con); SqlDataReader dr = cmd.ExecuteReader();
<pre>if (dr.Read()) {     txtSensorName.Text = dr["SensorName"].ToString();     txtSensorType.Text = dr["SensorType"].ToString();</pre>
} con.Close();
}

}

### "Read Sensors" App

Sensor System		_		×	Here we use a <b>ListBox</b>
Sensors: Sensor1 Sensor2 Sensor3 Sensor4 Sensor5	ListBox				
		F	Read	)	When clicking "Read", data from the SQL Server Database will be shown in the <b>ListBox</b> .

using System; using System.Windows.Forms; using Microsoft.Data.SqlClient;

namespace SensorSystem

public partial class Form1 : Form

public Form1()

InitializeComponent();

private void btnRead\_Click(object sender, EventArgs e)

string connectionString = "Data Source=Hans-Petter\\SQLEXPRESS;Initial Catalog=SENSORSYSTEM;Integrated Security=True;TrustServerCertificate=True";

string sqlQuery = "select SensorName from SENSOR"; SqlConnection con = new SqlConnection(connectionString); con.Open();

```
SqlCommand cmd = new SqlCommand(sqlQuery, con);
SqlDataReader dr = cmd.ExecuteReader();
```

```
lstSensors.Items.Clear();
```

if (dr != null)

```
while (dr.Read())
```

string sensor = dr["SensorName"].ToString();
lstSensors.Items.Add(sensor);

con.Close();

### "Read Sensors" App 2



#### Form1.cs

```
using System;
using System.Windows.Forms;
using Microsoft.Data.SqlClient;
```

namespace SensorSystem

```
public partial class Form1 : Form
```

```
string connectionString = "server=Hans-Petter\\SQLEXPRESS;Initial Catalog=SENSORSYSTEM;Integrated
Security=True;TrustServerCertificate=True";
```

#### public Form1()

```
InitializeComponent();
GetSensorTypes();
```

```
private void cboSensorTypes_SelectedIndexChanged(object sender, EventArgs e)
{
    string sensorTypeSelected = cboSensorTypes.SelectedItem.ToString();
    GetSensors(sensorTypeSelected);
}
```

### GetSensorTypes()

private void GetSensorTypes()

string sqlQuery = "select distinct SensorType from SENSOR order by SensorType";

```
SqlConnection con = new SqlConnection(connectionString);
con.Open();
```

```
SqlCommand cmd = new SqlCommand(sqlQuery, con);
SqlDataReader dr = cmd.ExecuteReader();
```

```
cboSensorTypes.Items.Clear();
if (dr != null)
{
    while (dr.Read())
    {
        string sensorType = dr["SensorType"].ToString();
        cboSensorTypes.Items.Add(sensorType);
    }
```

#### con.Close();

### GetSensors()

private void GetSensors(string sensorTypeSelected)

string sqlQuery = "select SensorName from SENSOR where SensorType = "" + sensorTypeSelected +
"" order by SensorName";

```
SqlConnection con = new SqlConnection(connectionString);
con.Open();
```

```
SqlCommand cmd = new SqlCommand(sqlQuery, con);
SqlDataReader dr = cmd.ExecuteReader();
```

```
IstSensors.Items.Clear();
if (dr != null)
{
    while (dr.Read())
    {
        string sensor = dr["SensorName"].ToString();
        lstSensors.Items.Add(sensor);
    }
}
```

#### con.Close();

### "Read Sensors" App 2b - Class

Sensor System —		×
Sensor Type:		
Terrender		
Temperature	~	
Sensors:		
Sensor1		
Sensor2		
Sensor5		
Sensor/		
Seshoro		

We will improve code structure by creating a separate **Class** called "Sensor" and move most of the code into that class. using Microsoft.Data.SqlClient; using System.Collections.Generic;

namespace SensorSystem.Classes

#### public class Sensor

string connectionString = "Data Source=Hans-Petter\\SQLEXPRESS;Initial Catalog=SENSORSYSTEM;Integrated Security=True;TrustServerCertificate=True";

public List<string> Get SensorTypes()

string sqlQuery = "select distinct SensorType from SENSOR order by SensorType";

SqlConnection con = new SqlConnection(connectionString); con.Open();

SqlCommand cmd = new SqlCommand(sqlQuery, con); SqlDataReader dr = cmd.ExecuteReader();

#### List<string> itemsSensorTypes = new List<string>();

if (dr != null)

```
while (dr.Read())
```

string sensorType = dr["SensorType"].ToString(); itemsSensorTypes.Add(sensorType);

```
}
```

con.Close();

retum itemsSensorTypes;

```
public List<string> GetSensors(string sensorTypeSelected)
```

string sqlQuery = "select SensorName from SENSOR where SensorType = '" + sensorTypeSelected + "' order by SensorName";

```
SqlConnection con = new SqlConnection(connectionString);
con.Open();
```

```
SqlCommand cmd = new SqlCommand(sqlQuery, con);
SqlDataReader dr = cmd.ExecuteReader();
```

List<string> itemsSensors = new List<string>();

```
if (dr != null)
```

```
while (dr.Read())
```

```
string sensor = dr["SensorName"].ToString();
itemsSensors.Add (sensor);
```

```
}
con.Close();
```

retum itemsSensors;



using System; using System.Collections.Generic; using System.Windows.Forms; using SensorSystem.Classes;

#### namespace SensorSystem

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        GetSensorTypes();
    }
```

```
private void cboSensorTypes_SelectedIndexChanged(object sender, EventArgs e)
{
    string sensorTypeSelected = cboSensorTypes.SelectedItem.ToString();
    GetSensors(sensorTypeSelected);
}
private void GetSensorTypes()
{
    //See Next Pages
}
private void GetSensors(string sensorTypeSelected)
{
    //See Next Pages
}
```

Form1.cs

### GetSensorTypes()

private void GetSensorTypes()

```
cboSensorTypes.Items.Clear();
```

```
Sensor sensor = new Sensor();
```

```
List<string> itemsSensorTypes = new List<string>();
itemsSensorTypes = sensor.GetSensorTypes();
```

```
foreach (string sensorType in itemsSensorTypes)
{
    cboSensorTypes.Items.Add(sensorType);
}
```

### GetSensors()

private void GetSensors(string sensorTypeSelected)

```
lstSensors.ltems.Clear();
```

```
Sensor sensor = new Sensor();
```

```
List<string> itemsSensors = new List<string>();
itemsSensors = sensor.GetSensors(sensorTypeSelected);
```

```
foreach (string sensorName in itemsSensors)
{
    lstSensors.ltems.Add(sensorName);
}
```

#### "Read Sensors" App 2c – App.config

S S	ensor System		-		×
	Sensor Type				
	Temperature			~	
	Sensors:				
	Sensor1 Sensor2 Sensor5 Sensor7 Sesnor8				

We will improve the App by putting the Connection String into a Configuration File called **"App.config"**.

That's because it makes it easier to change the Connection string without changing the C# code.

Then we can make an Executable Application and distribute the Application to others that do not have Visual Studio.

### App.config

<?xml version="1.0" encoding="utf-8" ?> <configuration>

<connectionStrings> <add name="DatabaseConnectionString" connectionString="Server=Hans-Petter\SQLEXPRESS;Database=SENSORSYSTEM;Integrated Security=True;TrustServerCertificate=True" providerName="System.Data.SqlClient" /> </connectionStrings>

</configuration>

### Updated "Sensor.cs"

🔍 File Edit View	Git Project Build Debug Test Analyze Tools Extensions Window Help <sup>PSearch</sup> ReadSensors2c	se – 🗆
● • ◎ 10 • ≤ 20 20 5 •	ং · Deb · Any CPI · ▶ SensorSystem · ▷ এ · জ রা ৣ ৬ ৯ ৫৫ জ রা আরু বিরুদ্ধ নির্বাহ	ا GitHub Copilot 🕫
App.config	Sensor.cs • ×	Solution Explorer
SensorSystem	SensorSystem Classes Sensor	
	mg Microsoft_Data_SolClient:	Search Solution Explorer (Ctrl+")
۲ 2 <u>usi</u>	System Collections Generic:	Solution 'ReadSensors2c' (1 of 1 proje
g 3 usi	ng System.Configuration;	A SensorSystem
4		
5 Ynam	espace SensorSystem.Classes	
6 {		
7 4	are entry public class Sensor	App config
8≪		E Form1 cs
9	<pre>string connectionString = ConfigurationManager.ConnectionStrings["DatabaseConnectionString"].ConnectionString;</pre>	
10		, sa Program.cs
11 🗸	nublic listestrings GetSensorTynes()	
12		
13	string sqlQuery = "select distinct SensorType from SENSOR order by SensorType";	
14		
15	<pre>SqlConnection con = new SqlConnection(connectionString);</pre>	
16	<pre>con.Open();</pre>	
17		
18	SqlCommand cmd = new SglCommand(sqlQuery, con);	GitHub Copilot Chat Solution Explorer
19	SqlDataReader dr = cmd.ExecuteReader();	Properties • • ×
20		-
21	List(string) itemsSensoriypes = new List(string)();	11 94 ×
22	if (dp l= pull)	
24		
25	while (dr.Read())	
26	{	
27	<pre>string sensorType = dr["SensorType"].ToString();</pre>	
28	<pre>itemsSensorTypes.Add(sensorType);</pre>	
29	}	
30	}	
31	con.Close();	
32	return itemssensoriypes;	
33	J	
54	1 reference	
35 ¥	<pre>public List<string> GetSensors(string sensorTypeSelected)</string></pre>	
36	{	
37	string sqiQuery = "select SensorName from SENSOR where SensorType = '" + sensorTypeSelected + "' order by SensorName";	
38	Selfernestion con - new Selfernestion(connectionString);	
100 % 🔹 🧟 No issues f	Sqtomection con - new Sqtomection(connectionstring), → Ln:8 Ch:6 SPC CRU	

🕆 Add to Source Control 🕤 🗉 Select Repository - 🔍 🚽

🛡 Ready

#### "Read Sensors" App3 - DataGridView

Sensor S	Bystem			–	Here we will use a DataGridView
	Sensor ID	Sensor Name	Sensor Type		
•	1	Sensor1	Temperature		
	2	Sensor2	Pressure		
	3	Sensor3	Temperature		
	4	Sensor4	Pressure		
	5	Sensor5	Level		
		DataGri	dView		When clicking "Read". data
				Read	from the SQL Server Database will be shown in

#### **Create Class and Method**

00	File Edit \	iew Git Project Build Debug Test Analyze Tools Extensions Window Help 🥍 Search · ReadSensors2	😂 – 🗆 X
Ð	- 💿 🎁 <b>- 🚅 🖪 </b>	Deb · Any CPt · ▶ SensorSystem · ▷ ♂ · ♥ 🖬 🗊 🚽 🖢 🖬 🐨 🦉 🗛 २३ २३	🕫 GitHub Copilot 🖻 🖗
8	Sensor.cs	• •	Solution Explorer
	ConcorSy	SensorSystem.Classes.Sensor	
So	5 .	namespace SensorSystem.Classes	Search Solution Explorer (Ct 🖓
urc	6	(	□ Solution 'ReadSensors2' (1 of
29	7、	viderenes class Sensor	🔺 🖻 SensorSystem
	8	{	▶ # Dependencies
	9	<pre>public int SensorId { get; set; } </pre>	
	10	<pre>public string SensorName { get; set; }</pre>	
	11	public string SensorType { get; set; }	▶ · · · Program.cs
	12		
	13 、	Internet	
	14		
	15	<pre>string connectionString = "Data Source=Hans-Petter\\SOLEXPRESS;Initial Catalog=SENSORSYSTEM;I</pre>	
	16		
	17	<pre>string sqlQuery = "select SensorId, SensorName, SensorType from SENSOR";</pre>	1
	18		
	19	SqlConnection con = new SglConnection(connectionString);	
	20	con.Open();	4
	21	Selfermand and - new Selfermand(selfermand(selfer), sen);	GitHub Copil Solution Expl
	23	SolDataReader dr = cmd_Frequency(Steps);	Properties 🔹 👻 🐇
	24		
	25	<pre>List<sensor> sensorList = new List<sensor>();</sensor></sensor></pre>	111 g+ P
	26 、	if (dr != null)	
	27	{	
	28 `	while (dr.Read())	
	29		
	31	Sensor Sensor - new Sensor(),	
	32	<pre>sensor.SensorId = Convert.ToInt32(dr["SensorId"]):</pre>	
	33	<pre>sensor.SensorName = dr["SensorName"].ToString();</pre>	
	34	<pre>sensor.SensorType = dr["SensorType"].ToString();</pre>	
	35		
	36	sensorList.Add(sensor);	
	37		
	39	son Close():	
	40	return sensorList:	
	41	}	
	42	}	
	43	3	
	100 % - 🧶 🛚 No	issues found 😢 + ∢ トー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	

#### 📕 Q 💁 💞 💽 🐂 🐶 🚇 🖏 📢 刘 🖏 💿 🦞 🚸 🚥 🔽 ங 🖃 🔗 🚸 🐰 📟 📴 🔧 📾 🕿 🌇

using Microsoft.Data.SqlClient; using System; using System.Collections.Generic;

namespace SensorSystem.Classes

class Sensor

public int SensorId { get; set; }
public string SensorName { get; set; }
public string SensorType { get; set; }

#### public List<Sensor> GetSensors()

string connectionString = "Data Source=Hans-Petter\\SQLEXPRESS;Initial Catalog=SENSORSYSTEM;Integrated Security=True;TrustServerCertificate=True";

string sqlQuery = "select SensorId, SensorName, SensorType from SENSOR";

SqlConnection con = new SqlConnection(connectionString); con.Open();

SqlCommand cmd = new SqlCommand(sqlQuery, con); SqlDataReader dr = cmd.ExecuteReader();

List<Sensor> sensorList = new List<Sensor>(); if (dr != null) { while (dr.Read())

Sensor sensor = new Sensor();

sensor.SensorId = Convert.ToInt32(dr["SensorId"]); sensor.SensorName = dr["SensorName"].ToString(); sensor.SensorType = dr["SensorType"].ToString();

sensorList.Add(sensor);

con.Close(); return sensorList;

#### Class "Sensor.cs"

using System; using System.Collections.Generic; using System.Windows.Forms; using SensorSystem.Classes;

namespace SensorSystem

public partial class Form1 : Form

public Form1()

InitializeComponent();

private void btnRead\_Click(object sender, EventArgs e)
{

GetData();

private void GetData()

Sensor sensor = new Sensor(); List<Sensor> sensorList = new List<Sensor>();

sensorList = sensor.GetSensors();

dgwSensors.DataSource = sensorList; FormatDataGridView();

private void FormatDataGridView()

dgwSensors.Columns[0].HeaderText = "Sensor ID"; dgwSensors.Columns[1].HeaderText = "Sensor Name"; dgwSensors.Columns[2].HeaderText = "Sensor Type";

dgwSensors.Columns[0].Width = 100; dgwSensors.Columns[1].Width = 300; dgwSensors.Columns[2].Width = 300; Form1.cs

### Hans-Petter Halvorsen

**University of South-Eastern Norway** 

www.usn.no

E-mail: <u>hans.p.halvorsen@usn.no</u> Web: <u>https://www.halvorsen.blog</u>

