Measurement System

Datalogging and Monitoring

A Practical Example using SQL Server, LabVIEW and Visual Studio/C#

Hans-Petter Halvorsen, M.Sc.

System Overview

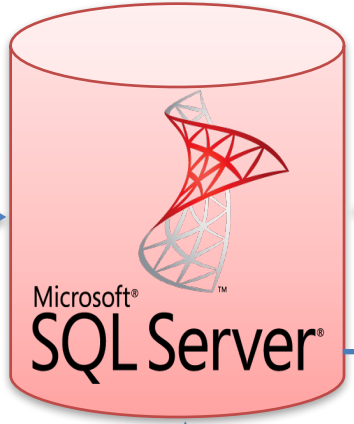
1. Design the Database using ERwin
2. Implement Tables, Views, Stored Procedures and Triggers using SQL Server.
3. Create a Datalogging App using LabVIEW.
4. Create a Data Monitoring App using Visual Studio/C#.

System Overview

Data Logging

Stored Procedure(s)

Database



Views and/or Stored Procedure(s)

Data Monitoring

DAQmx Driver

DAQ

Trigger(s)

Convert Temperature to Fahrenheit

Calculate Average, Max, Min Temperature Data

Necessary Software

- ERwin (CA ERwin Data Modeler Community Edition, free download from Internet)
- SQL Server (Express) Edition (SQL Server xxxx Express with Tools)
- LabVIEW
- DAQmx Driver Software
- LabVIEW SQL Toolkit (© Hans-Petter Halvorsen)
- Visual Studio

ERwin

Database Modelling and Design

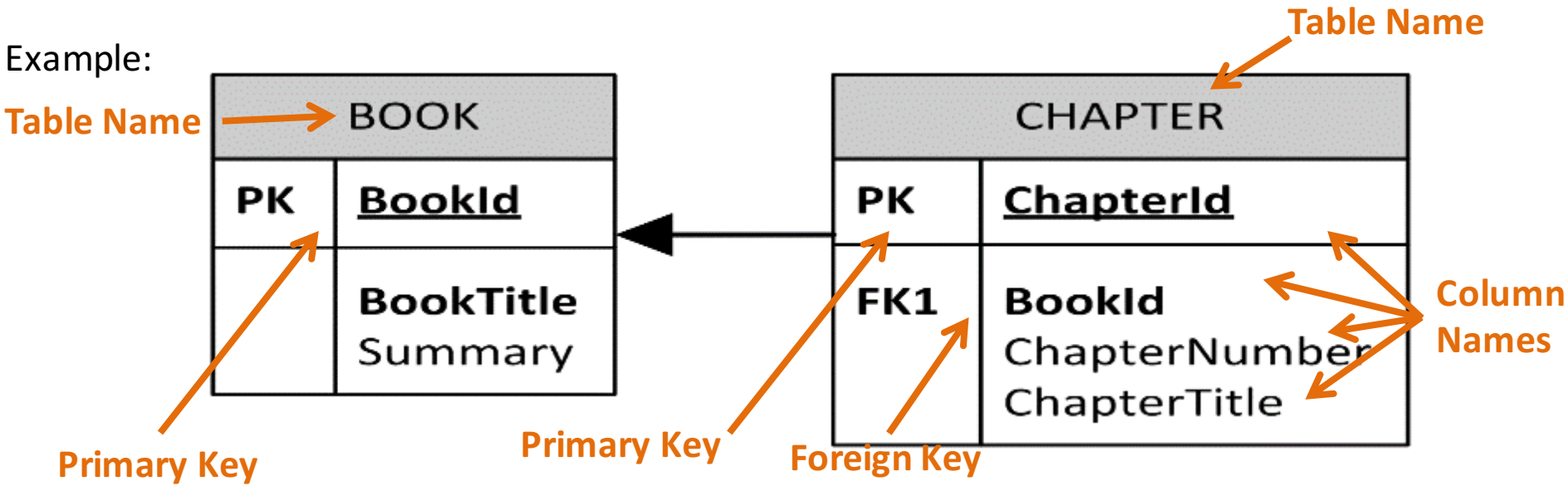
Hans-Petter Halvorsen, M.Sc.

Database Design – ER Diagram

ER Diagram (Entity-Relationship Diagram)

- Used for Design and Modeling of Databases.
- Specify Tables and **relationship** between them (**Primary Keys** and **Foreign Keys**)

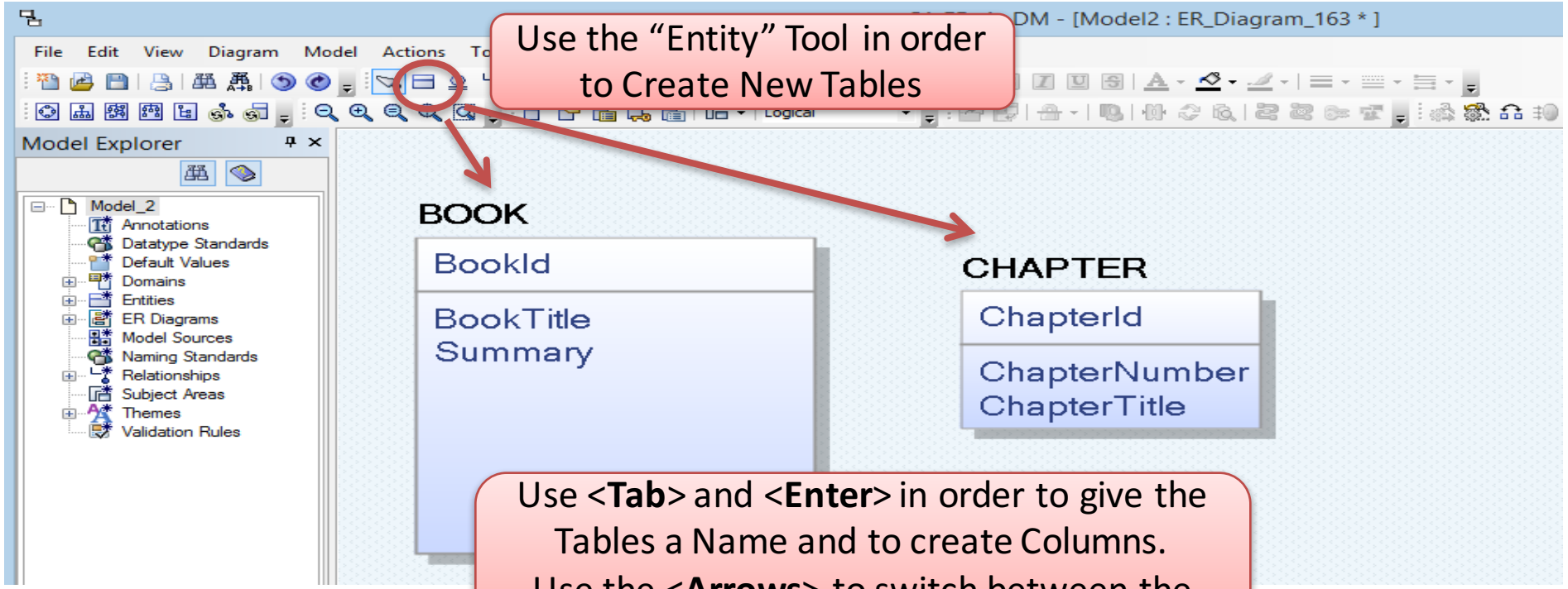
Example:



Relational Database. In a relational database all the tables have one or more relation with each other using Primary Keys (PK) and Foreign Keys (FK). Note! You can only have one PK in a table, but you may have several FK's.

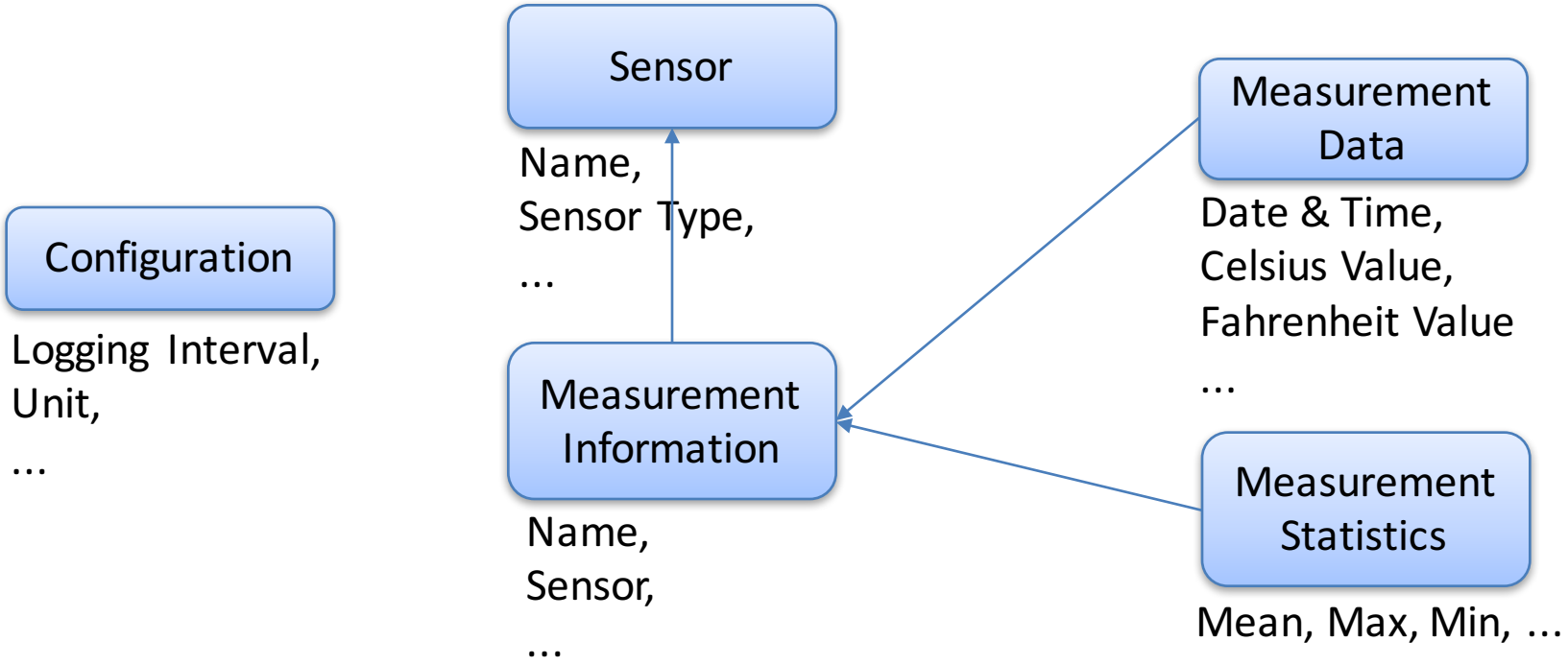
Introduction to ERwin

How-To: Create Tables and Columns

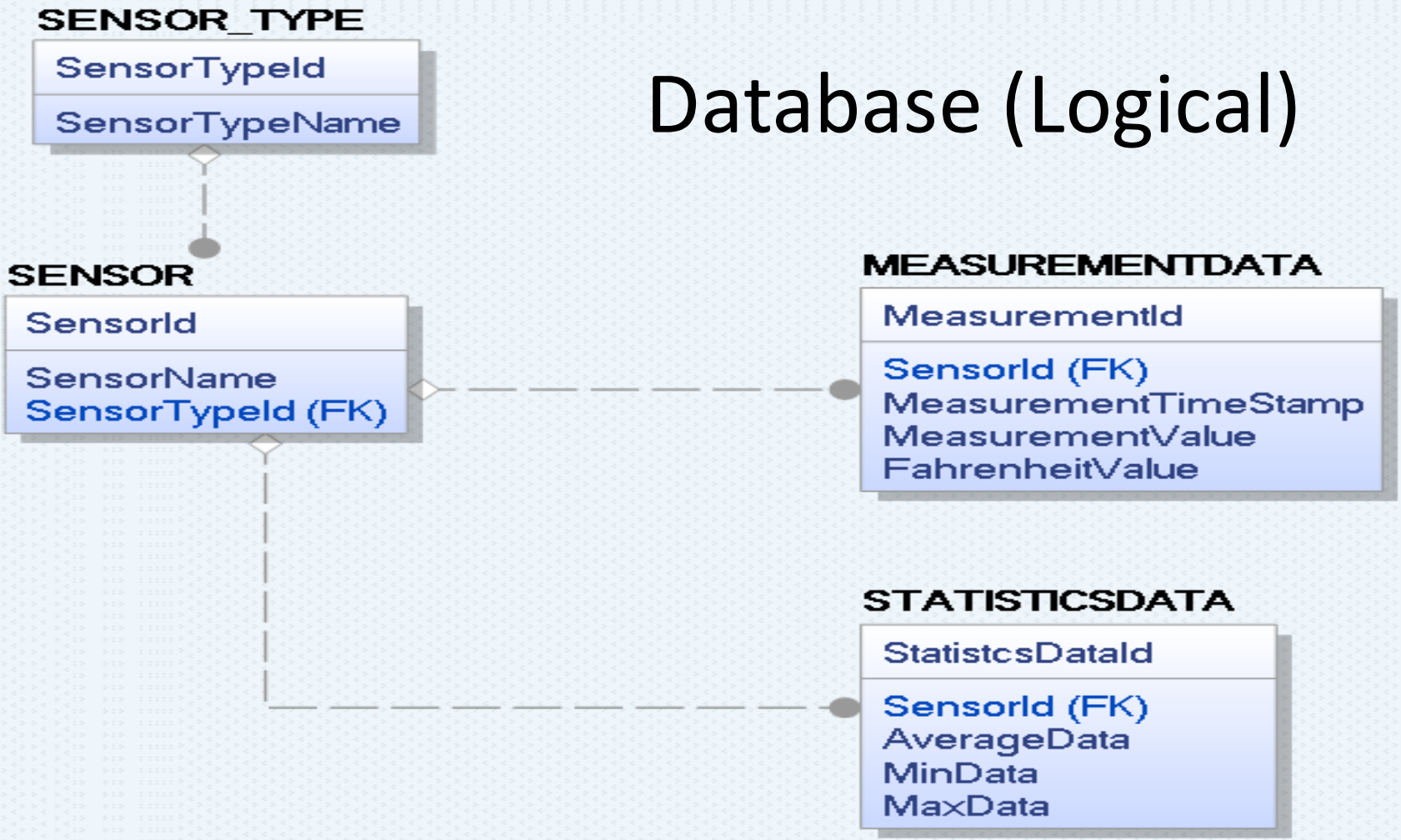


Database Example

Take into considerations that you can add multiple type of sensors later without having to change the database structure

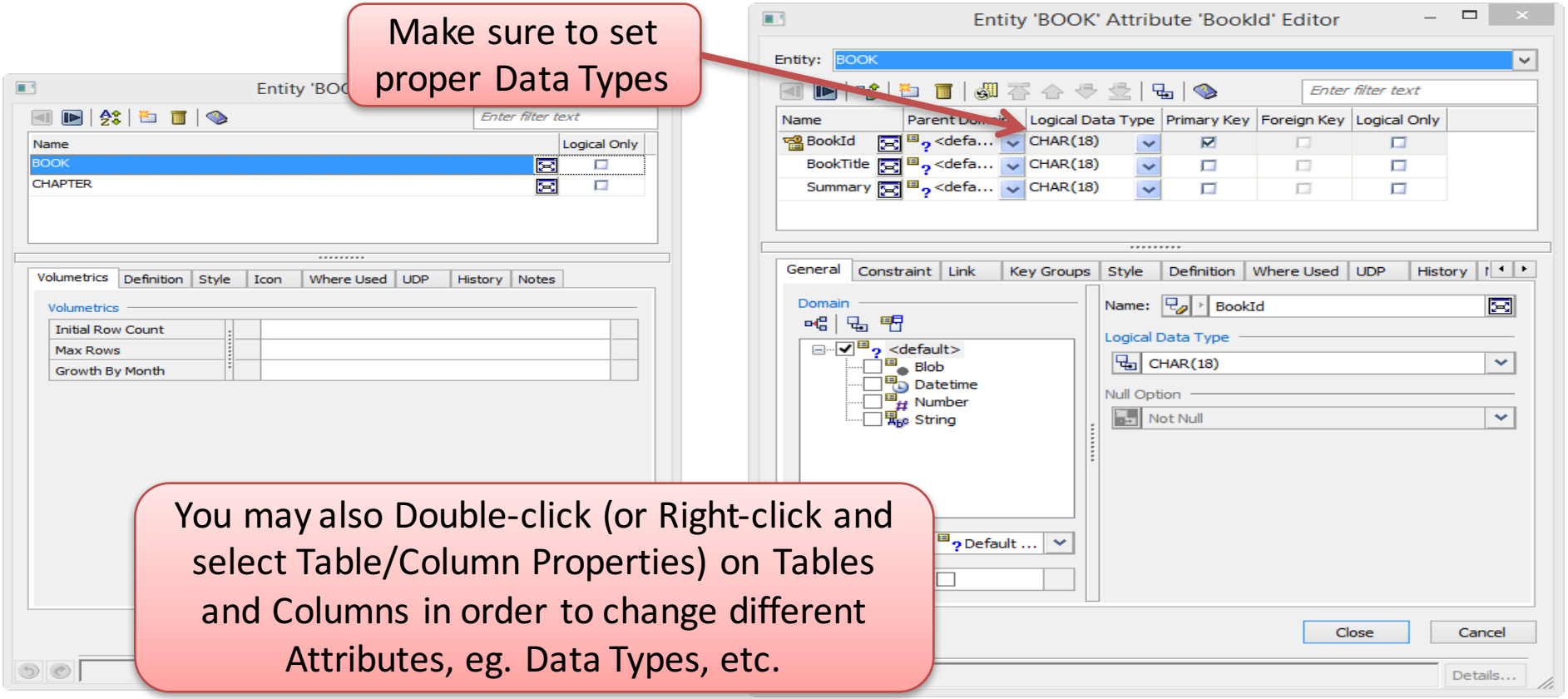


Database (Logical)



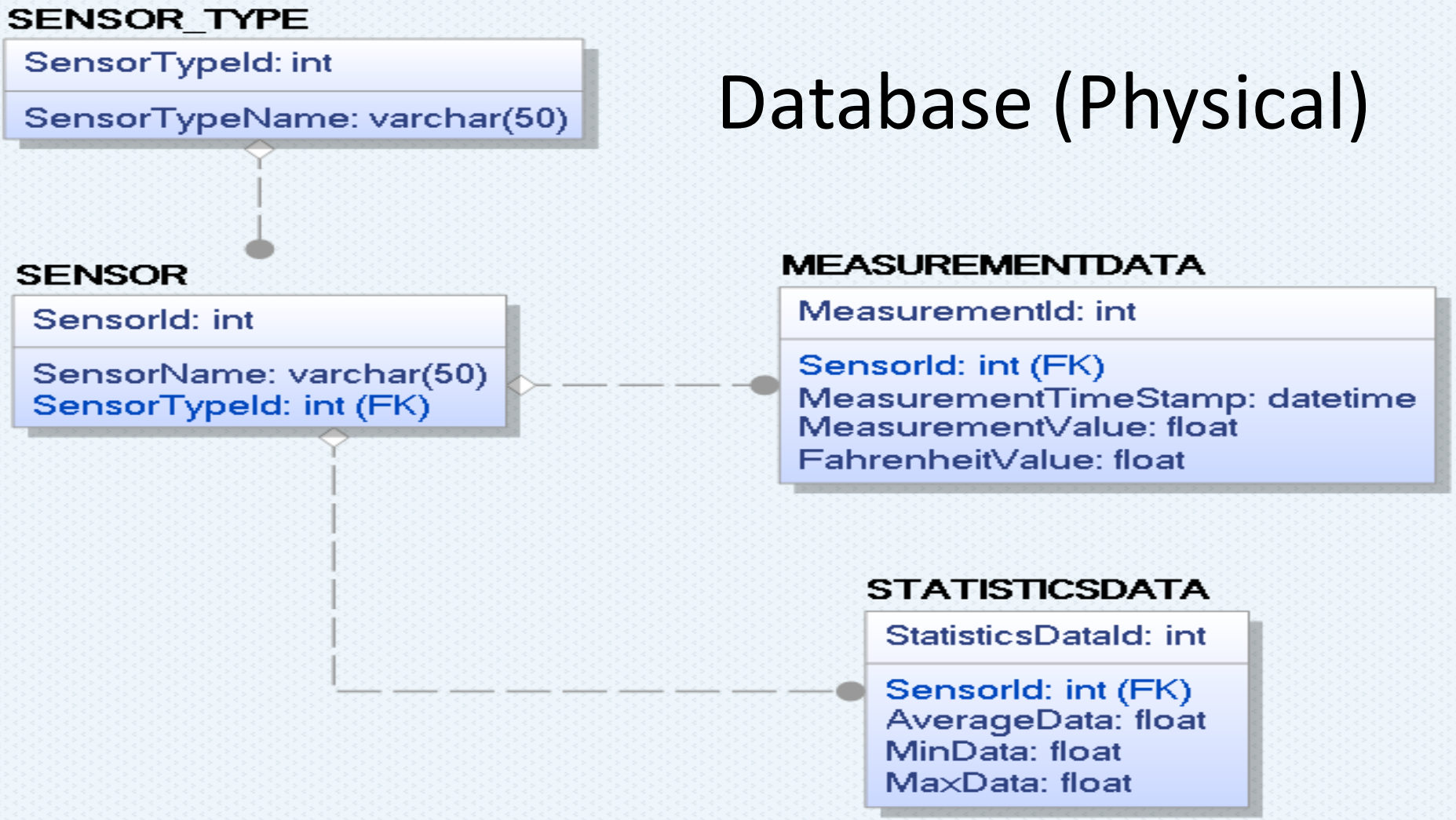
Setting Data Types (Physical Model)

Make sure to set proper Data Types



You may also Double-click (or Right-click and select Table/Column Properties) on Tables and Columns in order to change different Attributes, eg. Data Types, etc.

Database (Physical)

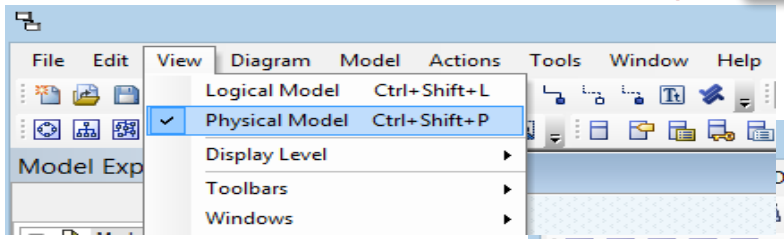



DEMO

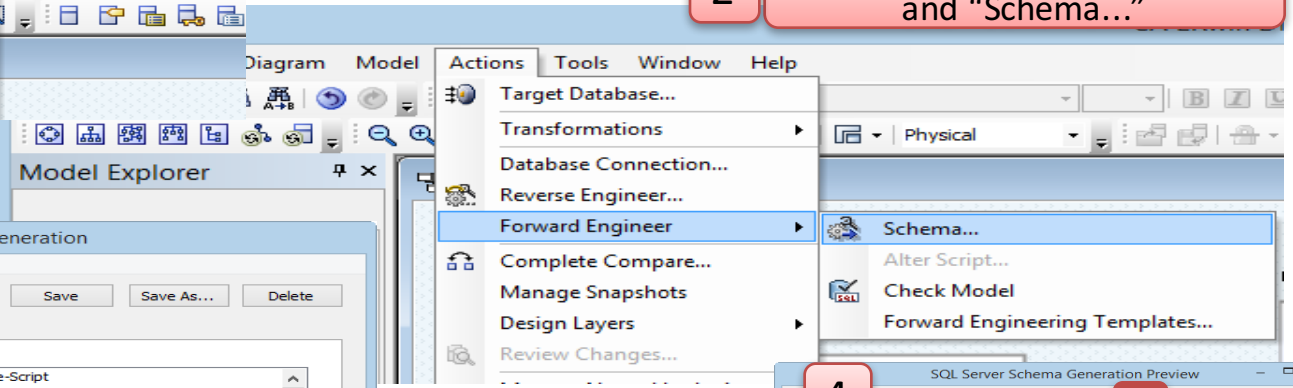
Creating TABLE Script

How-To: Create a SQL Script

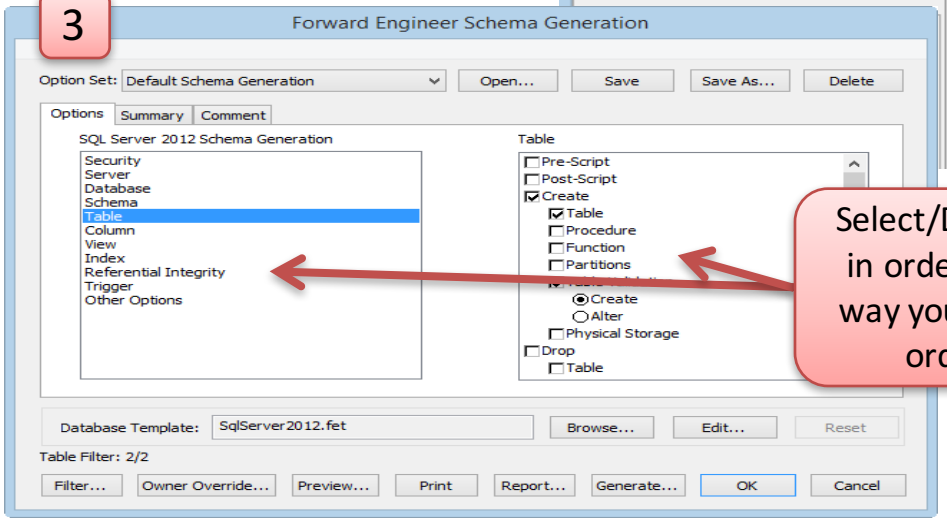
1 Make sure you are using the Physical Model



2 Select "Forward Engineering" and "Schema..."

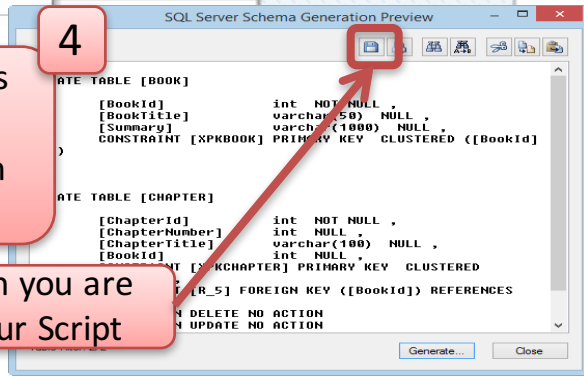


3



Select/Deselect different Options in order to make your script the way you want. Click "Preview" in order to see the results.

4



Click "Save" when you are satisfied with your Script

Table Script

```
CREATE TABLE SENSOR_TYPE
(
SensorTypeId        int  NOT NULL  IDENTITY ( 1,1 ) ,
SensorTypeName      varchar(50)  NULL ,
CONSTRAINT XPKSENSOR_TYPE PRIMARY KEY  CLUSTERED (SensorTypeId ASC)
)
go

CREATE TABLE SENSOR
(
SensorId            int  NOT NULL  IDENTITY ( 1,1 ),
SensorName          varchar(50)  NULL ,
SensorTypeId        int  NULL ,
CONSTRAINT XPKSENSOR PRIMARY KEY  CLUSTERED (SensorId ASC),
CONSTRAINT R_1 FOREIGN KEY (SensorTypeId) REFERENCES SENSOR_TYPE(SensorTypeId)
)
go

CREATE TABLE STATISTICSDATA
(
StatisticsDataId    int  NOT NULL  IDENTITY ( 1,1 ) ,
SensorId            int  NULL ,
AverageData         float  NULL ,
MinData             float  NULL ,
MaxData             float  NULL ,
CONSTRAINT XPKSTATISTICSDATA PRIMARY KEY  CLUSTERED (StatisticsDataId ASC),
CONSTRAINT R_3 FOREIGN KEY (SensorId) REFERENCES SENSOR(SensorId)
)
go

CREATE TABLE MEASUREMENTDATA
(
MeasurementId       int  NOT NULL  IDENTITY ( 1,1 ),
MeasurementTimeStamp  datetime  NULL ,
SensorId            int  NULL ,
MeasurementValue    float  NULL ,
FahrenheitValue     float  NULL ,
CONSTRAINT XPKMEASUREMENTDATA PRIMARY KEY  CLUSTERED (MeasurementId ASC),
CONSTRAINT R_2 FOREIGN KEY (SensorId) REFERENCES SENSOR(SensorId)
)
go
```

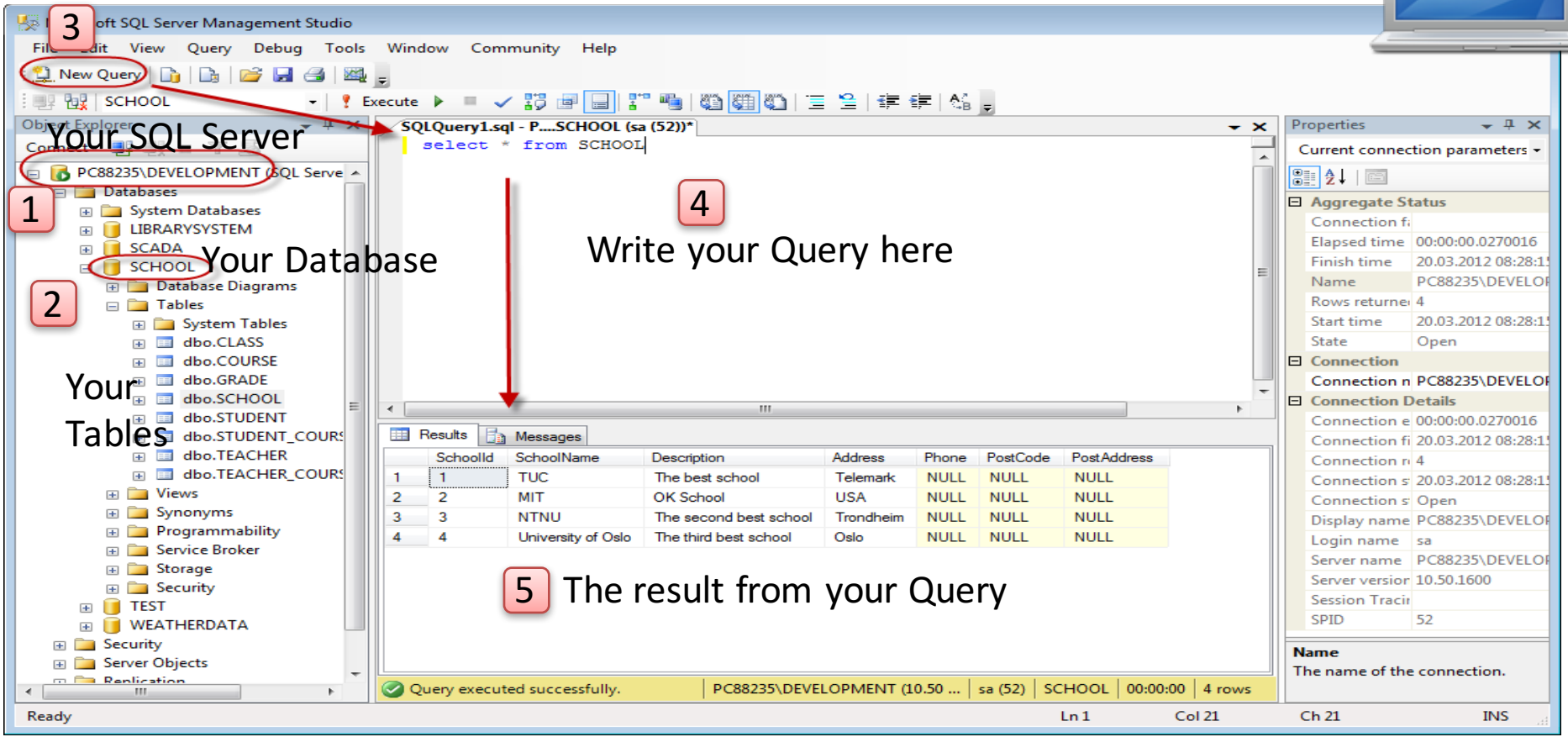

DEMO

SQL Server

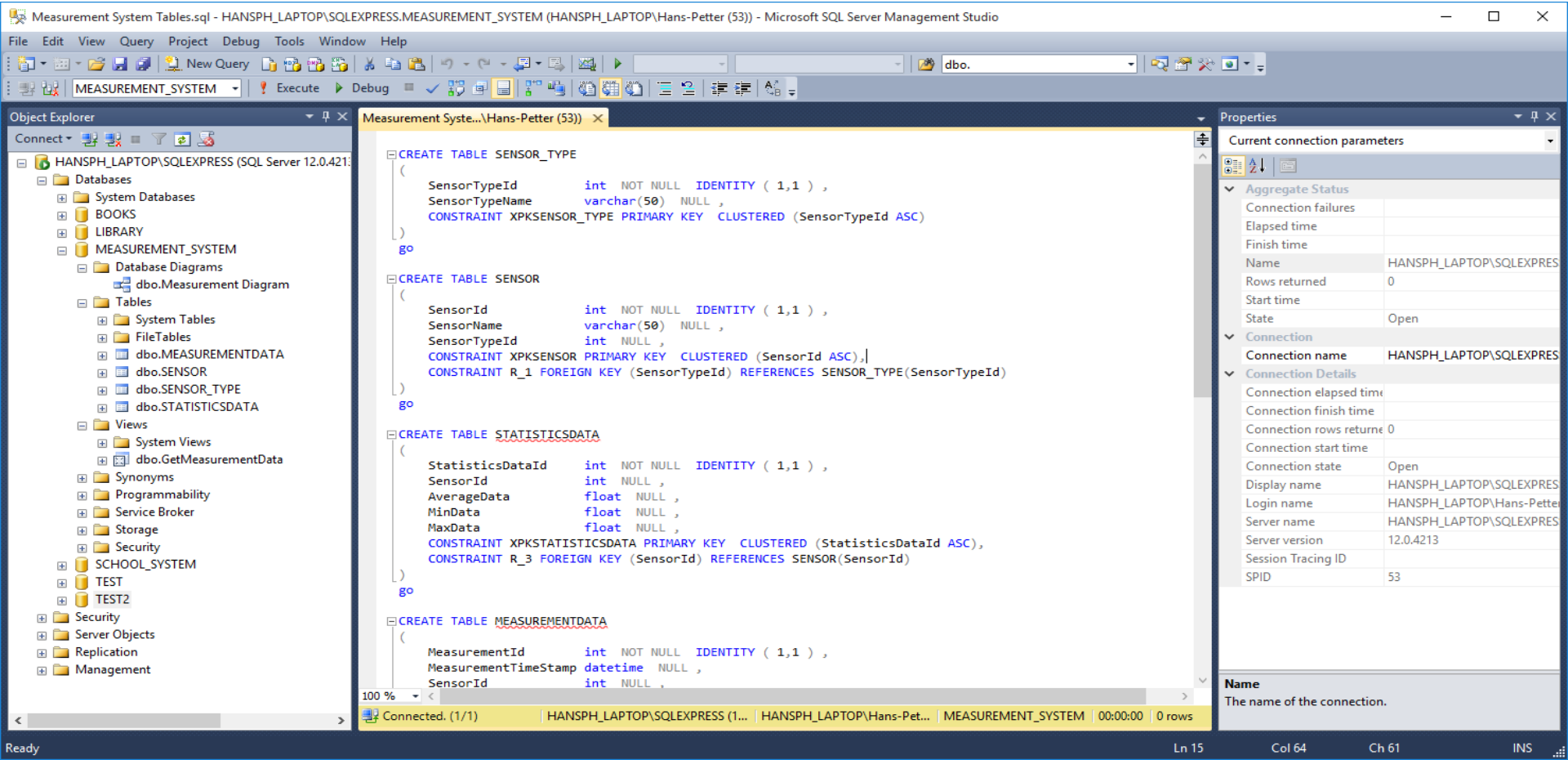
Database Implementation and Structured Query Language (SQL)

Hans-Petter Halvorsen, M.Sc.

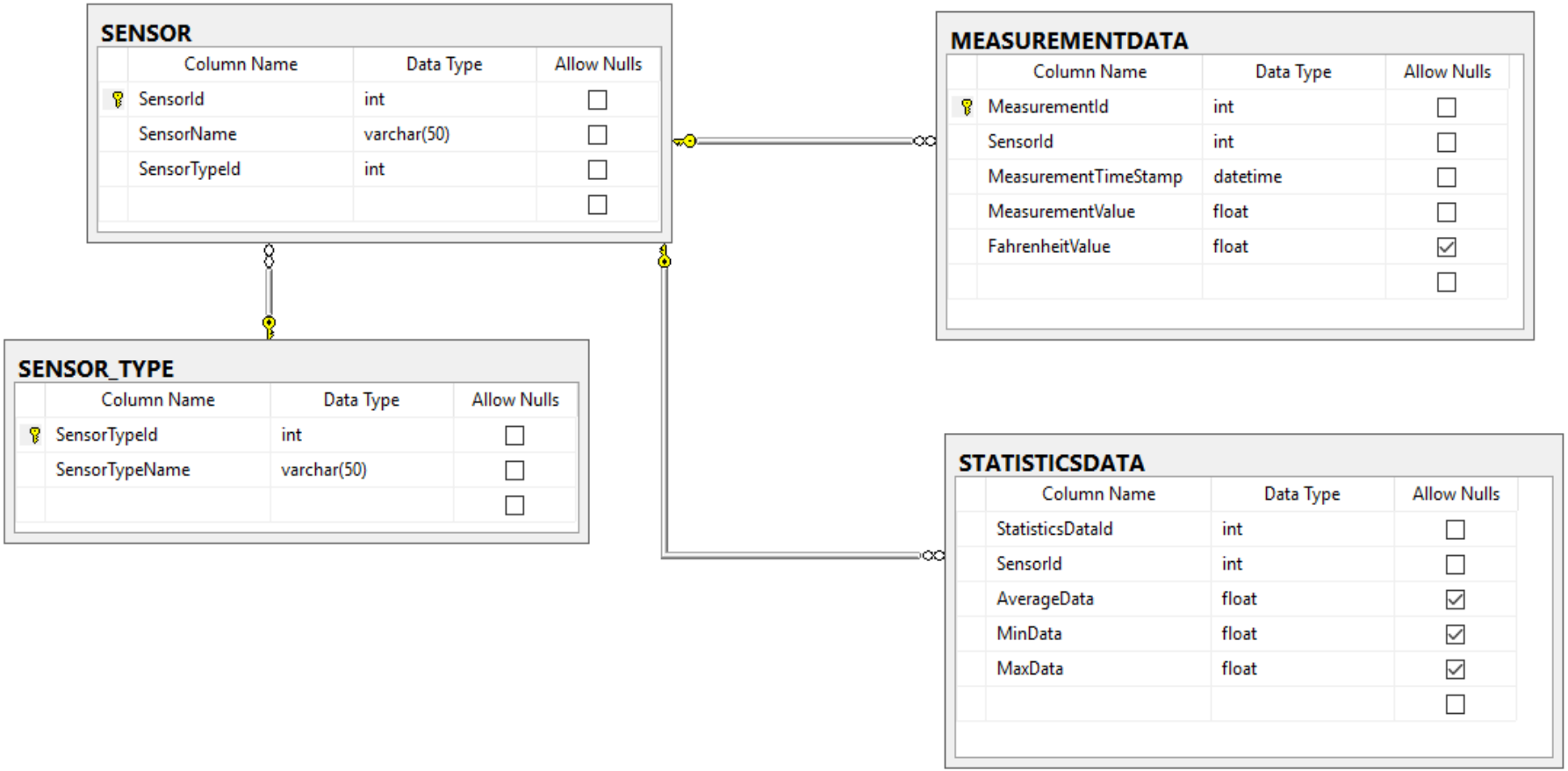
Microsoft SQL Server Management Studio



Insert Tables

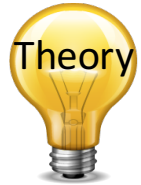


Database


SENSOR
Column Name
Data Type
Allow Nulls
SensorId
int
SensorName
varchar(50)
SensorTypeId
int
MEASUREMENTDATA
Column Name
Data Type
Allow Nulls
MeasurementId
int
SensorId
int
MeasurementTimeStamp
datetime
MeasurementValue
float
FahrenheitValue
float
SENSOR_TYPE
Column Name
Data Type
Allow Nulls
SensorTypeId
int
SensorTypeName
varchar(50)
STATISTICSDATA
Column Name
Data Type
Allow Nulls
StatisticsDataId
int
SensorId
int
AverageData
float
MinData
float
MaxData
float


DEMO

SQL – Structured Query Language



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**

Default Data

```
declare
@SensorTypeId int,
@SensorId int

insert into SENSOR_TYPE (SensorTypeName) values ('TC-01 Thermocouple')

select @SensorTypeId = SensorTypeId from SENSOR_TYPE where SensorTypeName='TC-01 Thermocouple'


insert into SENSOR (SensorName, SensorTypeId) values ('TC01-1', @SensorTypeId)

select @SensorId = SensorId from SENSOR where SensorName='TC01-1'


insert into STATISTICSDATA (SensorId, AverageData, MinData, maxData) values(@SensorId, 0, 0, 0)
```


DEMO

Views, Stored Procedures and Triggers

- **Views**: Views are virtual tables for easier access to data stored in multiple tables.
- **Stored Procedures**: A Stored Procedure is a precompiled collection of SQL statements. In a stored procedure you can use if sentence, declare variables, etc.
- **Triggers**: A database trigger is code that is automatically executed in response to certain events on a particular table in a database.

Stored Procedures

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

CREATE PROCEDURE SaveMeasurementData
@SensorName varchar(50),
@MeasurementValue float
AS

DECLARE
@SensorId int

select @SensorId = SensorId from SENSOR where SensorName = @SensorName

insert into MEASUREMENTDATA (SensorId, MeasurementValue, MeasurementTimeStamp)
values (@SensorId, @MeasurementValue, getdate())

GO
```

Views

```
IF EXISTS (SELECT name
   FROM   sysobjects
   WHERE  name = 'GetMeasurementData'
   AND   type = 'V')
DROP VIEW GetMeasurementData
GO

CREATE VIEW GetMeasurementData
AS

SELECT
SENSOR.SensorId,
SENSOR.SensorName,
MEASUREMENTDATA.MeasurementId,
MEASUREMENTDATA.MeasurementTimeStamp,
MEASUREMENTDATA.MeasurementValue,
MEASUREMENTDATA.FahrenheitValue

FROM MEASUREMENTDATA
INNER JOIN SENSOR ON MEASUREMENTDATA.SensorId = SENSOR.SensorId

GO
```

Triggers

```
IF EXISTS (SELECT name
  FROM   sysobjects
  WHERE  name = 'ConvertFahrenheit'
  AND   type = 'TR')
DROP TRIGGER ConvertFahrenheit
GO

CREATE TRIGGER ConvertFahrenheit ON MEASUREMENTDATA
FOR UPDATE, INSERT
AS

DECLARE
@MeasurementId int,
@MeasurementValue float,
@FahrenheitValue float

select @MeasurementId = MeasurementId from INSERTED
select @MeasurementValue = MeasurementValue from INSERTED

set @FahrenheitValue = (@MeasurementValue*9)/5 + 32;

update MEASUREMENTDATA set FahrenheitValue= @FahrenheitValue where MeasurementId= @MeasurementId

GO
```

Triggers

```
IF EXISTS (SELECT name
  FROM   sysobjects
  WHERE  name = 'CalculateStatistics'
  AND   type = 'TR')
DROP TRIGGER CalculateStatistics
GO

CREATE TRIGGER CalculateStatistics   ON MEASUREMENTDATA
FOR UPDATE, INSERT, DELETE
AS

DECLARE
@SensorId int,
@AverageData float,
@MinData float,
@MaxData float

select @SensorId = SensorId from INSERTED

select @AverageData = AVG(MeasurementValue) from MEASUREMENTDATA where SensorId = @SensorId
select @MinData = MIN(MeasurementValue) from MEASUREMENTDATA where SensorId = @SensorId
select @MaxData = MAX(MeasurementValue) from MEASUREMENTDATA where SensorId = @SensorId

update STATISTICSDATA set
AverageData = @AverageData,
MinData = @MinData,
MaxData = @MaxData
where SensorId = @SensorId

GO
```


DEMO

Datalogging using LabVIEW

Hans-Petter Halvorsen, M.Sc.

Datalogging using LabVIEW

Start by Design and Implement the Database Tables using ERwin

Database Design & Modelling

Create Stored Procedure(s) and Triggers in SQL Server

TC-01 Thermocouple

DAQ

Data

Stored Procedure(s)

Microsoft SQL Server Database

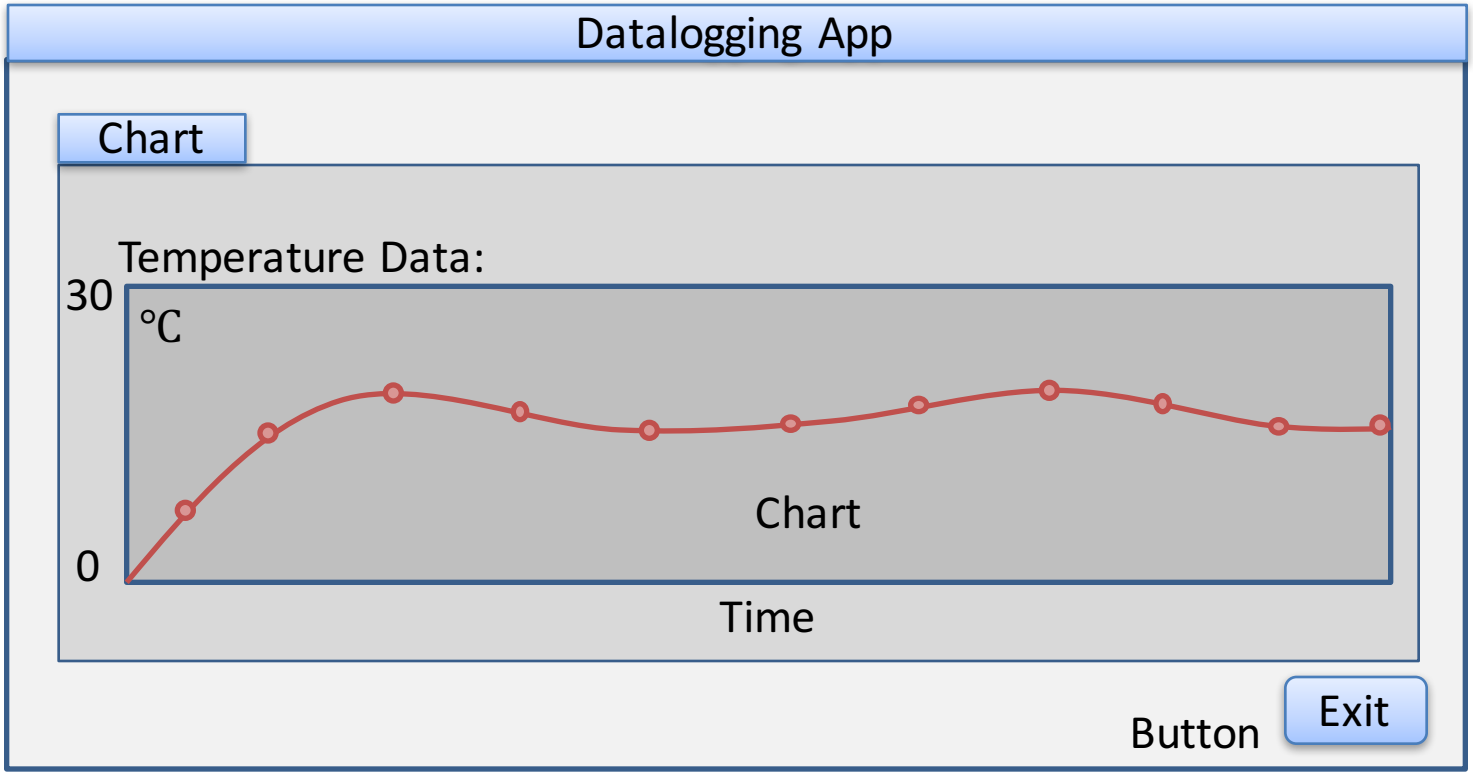
Triggers

Convert Temperature to Celsius/Fahrenheit

Calculate Average, Max, Min Temperature Data

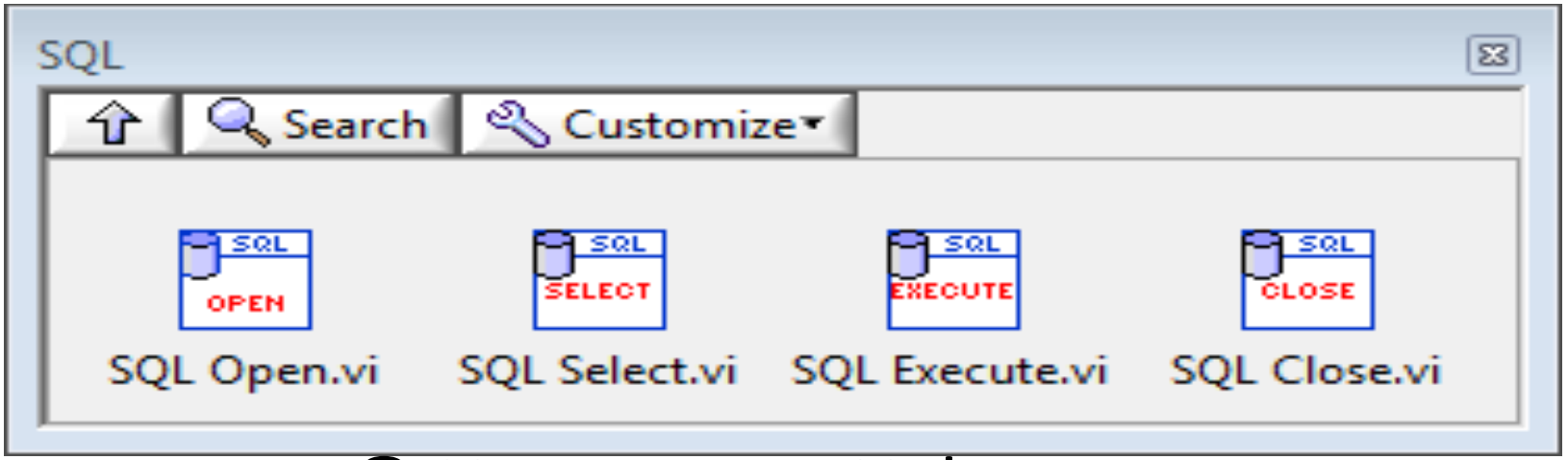
LabVIEW HMI Example

The Temperature Data from the TC-01 DAQ device should be stored in the Database.



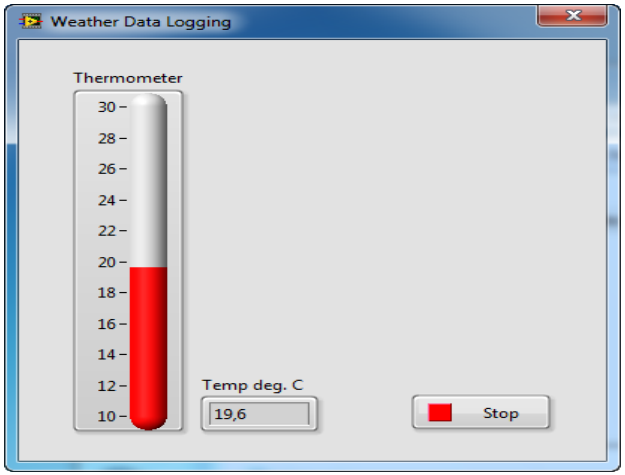
LabVIEW SQL Toolkit

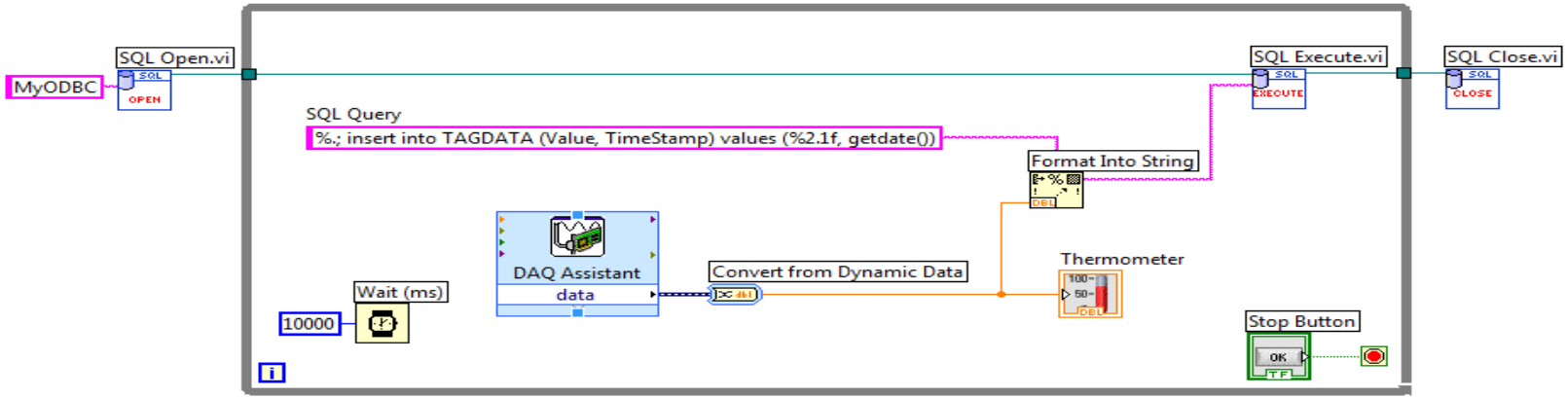
For Easy Database Communication with LabVIEW



Download for free here:
http://home.hit.no/~hansha/documents/labview/code/SQLToolkit.zip

LabVIEW SQL Toolkit Example





Logging App.vi Front Panel

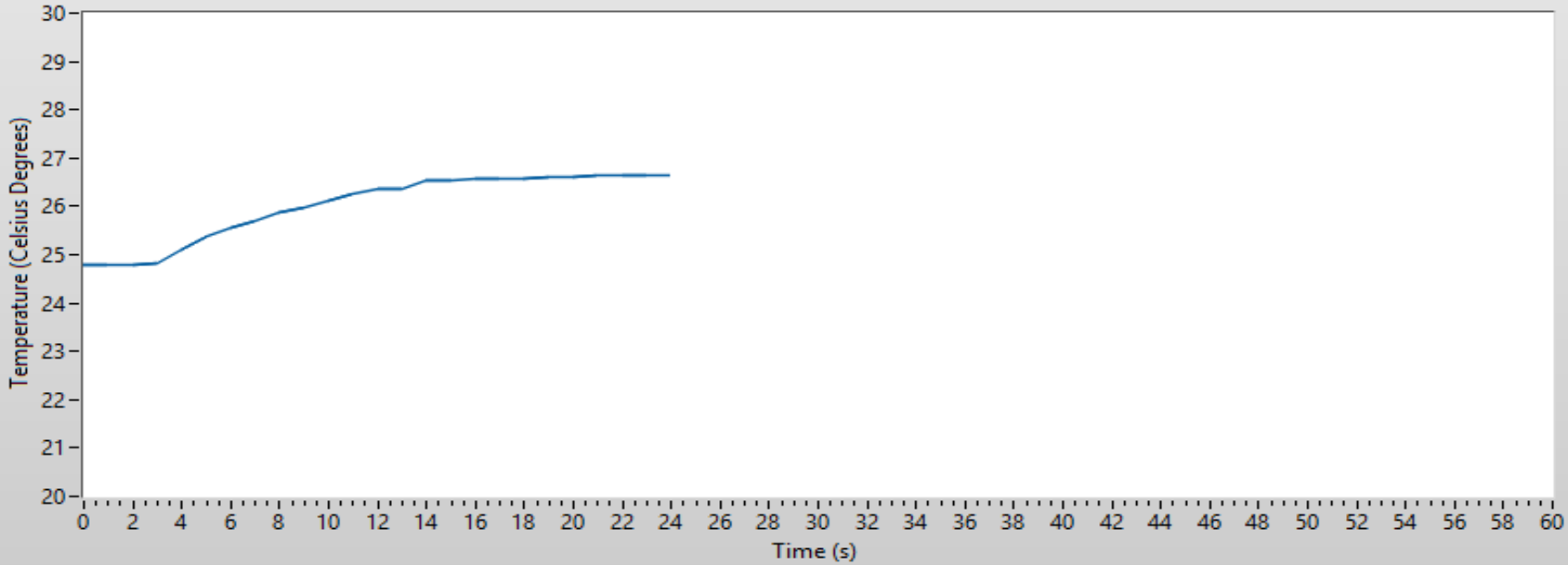
File Edit View Project Operate Tools Window Help

15pt Application Font

Search

Plot 0

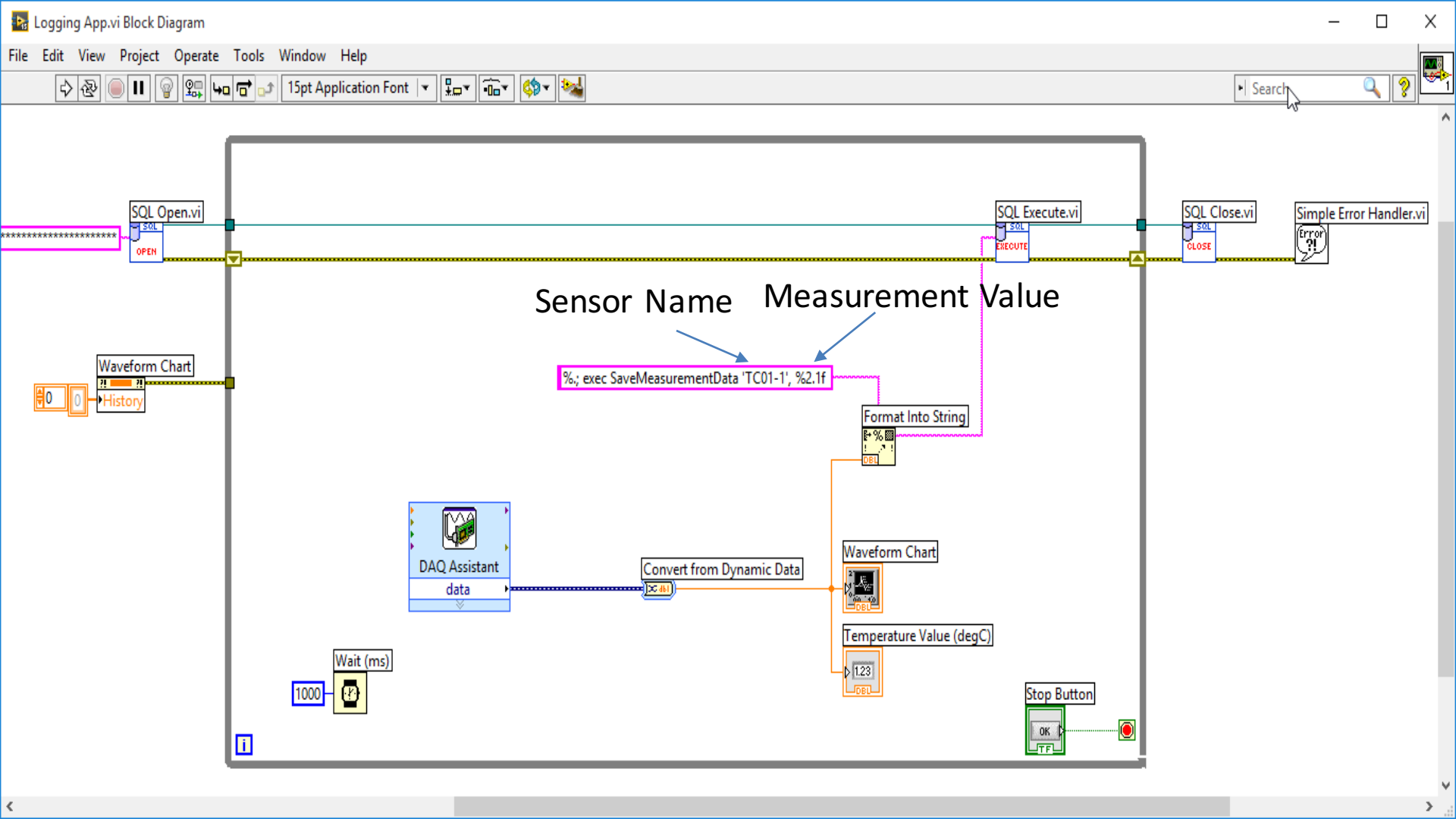
Waveform Chart



Temperature Value (degC)

26,7

Stop


Logging App.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
SQL Open.vi
SQL Execute.vi
SQL Close.vi
Simple Error Handler.vi
Sensor Name
Measurement Value
%.; exec SaveMeasurementData 'TC01-1', %2.1f
Format Into String
Waveform Chart
History
DAQ Assistant
data
Convert from Dynamic Data
Waveform Chart
Temperature Value (degC)
Wait (ms)
1000
Stop Button
OK

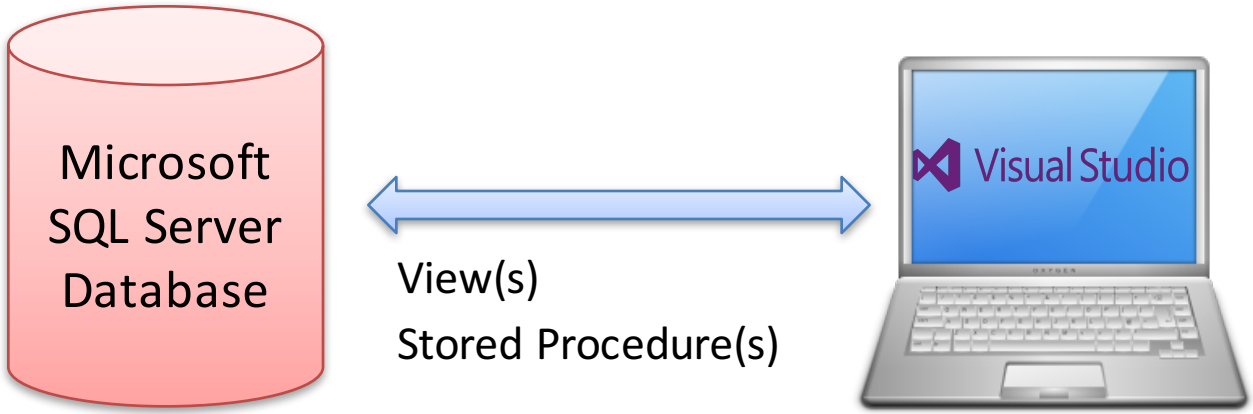

DEMO

Visual Studio

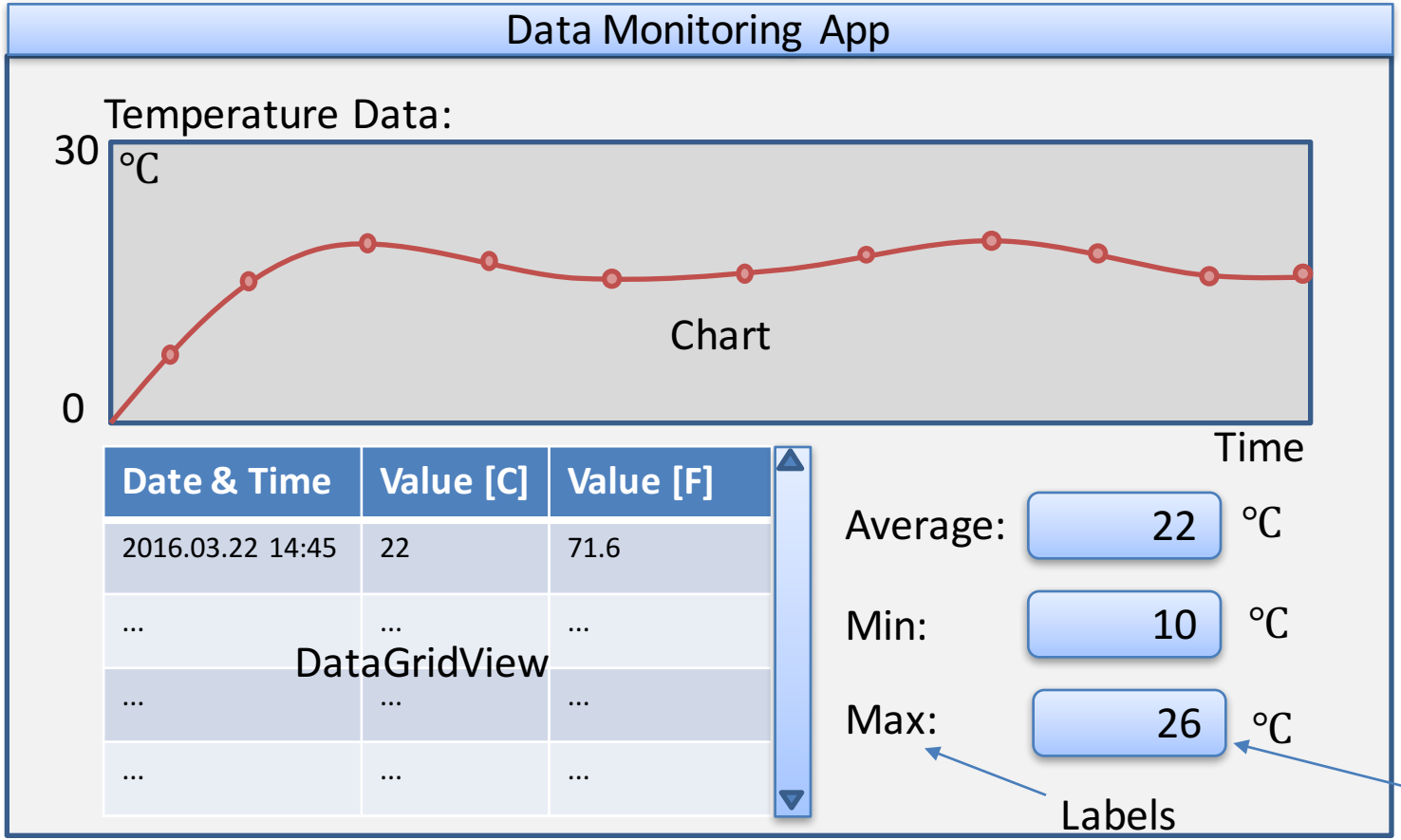
Data Monitoring using Visual Studio/C#

Hans-Petter Halvorsen, M.Sc.

Data Monitoring using Visual Studio/C#

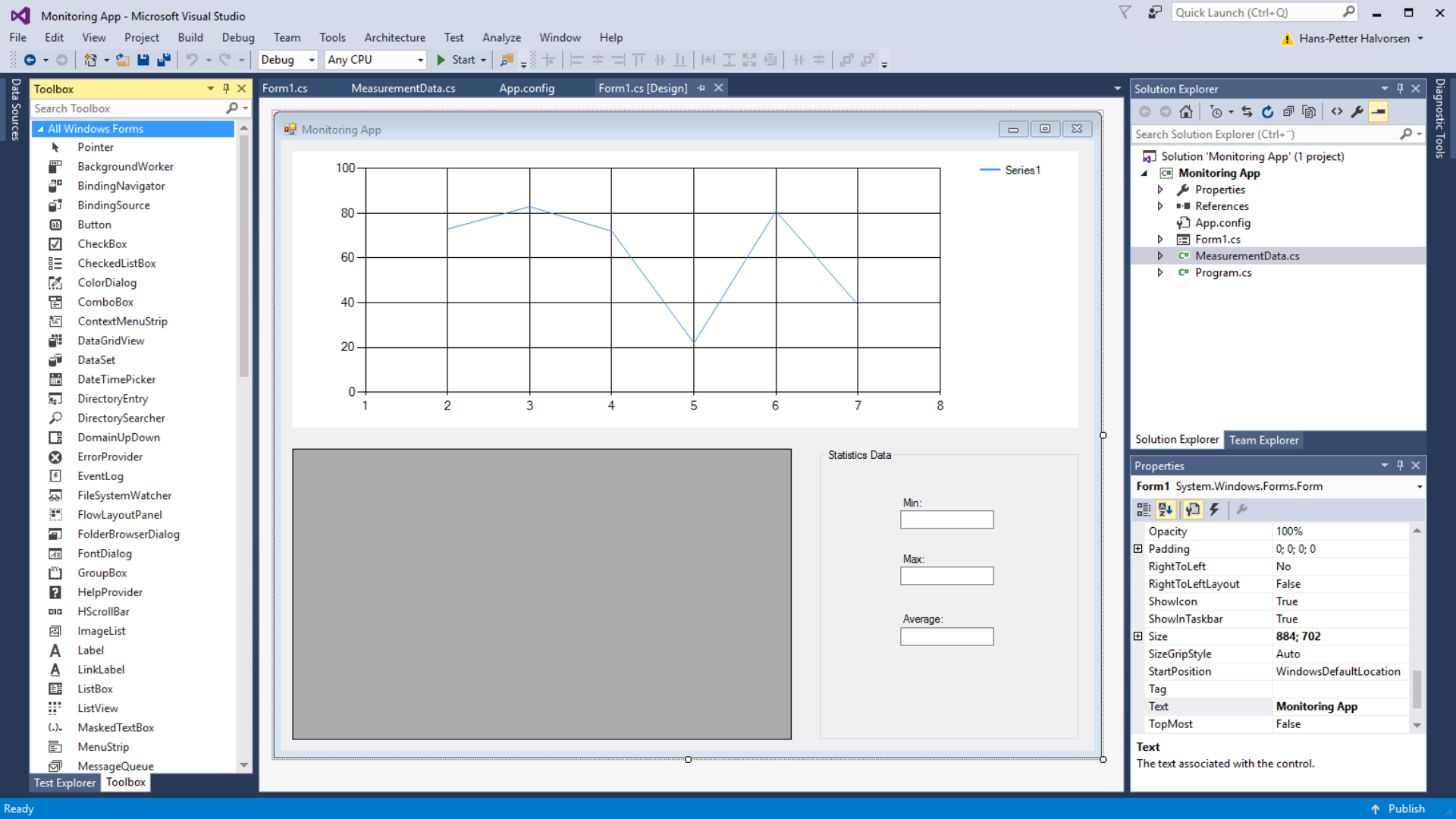


Visual Studio HMI Example

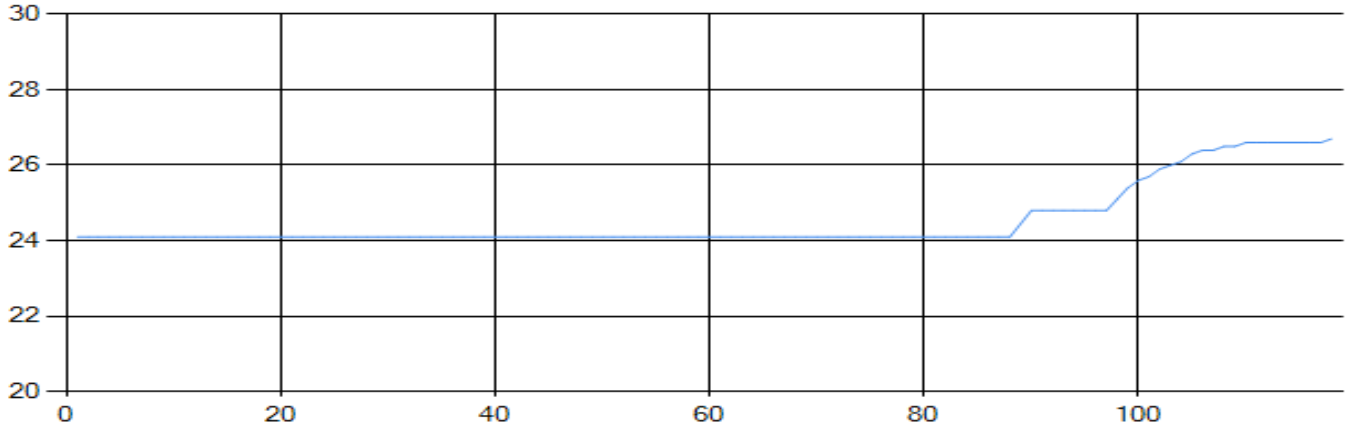


You should get the Data from the Database

Typically you get Data from the Database using Views and/or Stored Procedures


Monitoring App - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Architecture Test Analyze Window Help
Debug
Any CPU
Start
Toolbox
Search Toolbox
All Windows Forms
Pointer
BackgroundWorker
BindingNavigator
BindingSource
Button
CheckBox
CheckedListBox
ColorDialog
ComboBox
ContextMenuStrip
DataGridView
DataSet
DateTimePicker
DirectoryEntry
DirectorySearcher
DomainUpDown
ErrorProvider
EventLog
FileSystemWatcher
FlowLayoutPanel
FolderBrowserDialog
FontDialog
GroupBox
HelpProvider
HScrollBar
ImageList
Label
LinkLabel
ListBox
ListView
MaskedTextBox
MenuStrip
MessageQueue
Test Explorer
Toolbox
Form1.cs
MeasurementData.cs
App.config
Form1.cs [Design]
Monitoring App
100
80
60
40
20
0
1
2
3
4
5
6
7
8
Series1
Statistics Data
Min:
Max:
Average:
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Monitoring App' (1 project)
Monitoring App
Properties
References
App.config
Form1.cs
MeasurementData.cs
Program.cs
Solution Explorer
Team Explorer
Properties
Form1 System.Windows.Forms.Form
Opacity 100%
Padding 0; 0; 0; 0
RightToLeft No
RightToLeftLayout False
ShowIcon True
ShowInTaskbar True
Size 884; 702
SizeGripStyle Auto
StartPosition WindowsDefaultLocation
Tag
Text Monitoring App
TopMost False
Text
The text associated with the control.
Ready
Publish

Monitoring App



MeasurementId	MeasurementTimeStamp	MeasurementValue	FahrenheitValue
1	07.04.2016 10.44	24,1	75,38
2	07.04.2016 10.44	24,1	75,38
3	07.04.2016 10.44	24,1	75,38
4	07.04.2016 10.44	24,1	75,38
5	07.04.2016 10.44	24,1	75,38
6	07.04.2016 10.44	24,1	75,38
7	07.04.2016 10.44	24,1	75,38
8	07.04.2016 10.44	24,1	75,38
9	07.04.2016 10.44	24,1	75,38
10	07.04.2016 10.44	24,1	75,38
11	07.04.2016 10.44	24,1	75,38
12	07.04.2016 10.44	24,1	75,38
13	07.04.2016 10.44	24,1	75,38

Statistics Data

Min:
24,1

Max:
26,7

Average:
24,5350427350427

Get Data from Database

Connection String stored in App.config

```
public List<MeasurementData> GetMeasurementData()
{

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

    List<MeasurementData> measurementDataList = new List<MeasurementData>();

    SqlConnection con = new SqlConnection(connectionString);

    string selectSQL = "select MeasurementId, MeasurementTimeStamp, MeasurementValue, FahrenheitValue from GetMeasurementData where SensorName ='TC01-1'";

    con.Open();

    SqlCommand cmd = new SqlCommand(selectSQL, con);

    SqlDataReader dr = cmd.ExecuteReader();

    if (dr != null)
    {
       while (dr.Read())
       {
          MeasurementData measurementData = new MeasurementData();

          measurementData.MeasurementId = Convert.ToInt32(dr["MeasurementId"]);
          measurementData.MeasurementTimeStamp = Convert.ToDateTime(dr["MeasurementTimeStamp"]);
          measurementData.MeasurementValue = Convert.ToDouble(dr["MeasurementValue"]);
          measurementData.FahrenheitValue = Convert.ToDouble(dr["FahrenheitValue"]);

          measurementDataList.Add(measurementData);
       }
    }

    con.Close();

    return measurementDataList;
}
```

```
public int MeasurementId { get; set; }
public DateTime MeasurementTimeStamp { get; set; }
public double MeasurementValue { get; set; }
public double FahrenheitValue { get; set; }
```

DataGridView

```
…

List<MeasurementData> measurementList = new List<MeasurementData>();

measurementList = measurementData.GetMeasurementData();

gridMeasurementData.DataSource = measurementList;
```

Charting in Visual Studio

Visual Studio has a Chart control that you can use in Windows Forms or Web application (ASP.NET)

https://msdn.microsoft.com/en-us/library/dd489237.aspx

http://www.i-programmer.info/programming/uiux/2756-getting-started-with-net-charts.html

```
using System.Windows.Forms.DataVisualization.Charting;
...
chart1.Series.Clear();
chart1.Series.Add("My Data");
chart1.Series["My Data"].ChartType=SeriesChartType.Line;
...
int[] x = {1, 2, 3, 4, 5, 6, 7, 8};
int[] y = {20, 22, 25, 24, 28, 27, 24, 26};
for (int i = 0; i < x.Length; i++)
{
   chart1.Series["My Data"].Points.AddXY(x[i],y[i]);
}
```

Creating a Web App? Use the following Namespace instead:
System.Web.UI.DataVisualization.Charting

Chart Data from Database

```
private void FillChart()
{

   chartMeasurementData.Series.Clear();
   chartMeasurementData.Series.Add("MeasurementData");
   chartMeasurementData.Series["MeasurementData"].ChartType = SeriesChartType.Line;

   ChartArea area = chartMeasurementData.ChartAreas[0];
   area.AxisY.Minimum = 20;
   area.AxisY.Maximum = 30;

   List<MeasurementData> measurementList = new List<MeasurementData>();
   MeasurementData measurementData = new MeasurementData();
   measurementList = measurementData.GetMeasurementData();

   foreach (MeasurementData data in measurementList)
   {
     chartMeasurementData.Series["MeasurementData"].Points.AddXY(data.MeasurementId, data.MeasurementValue);
   }

}
```

Timer

In Visual Studio you may want to use a Timer instead of a While Loop in order to read values at specific intervals.

1

Select the "Timer" component in the Toolbox

2 **Initialization:**

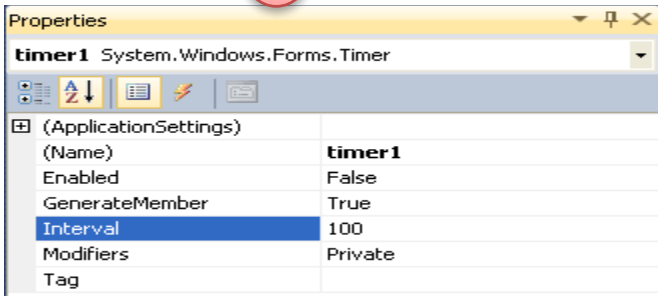
```
public Form1()
    {
        InitializeComponent();

        timer1.Start();
    }
```

Double-click on the Timer object in order to create the Event

4 **Timer Event:**

```
private void timer1_Tick(object sender, EventArgs e)
    {
      … //Read from DB
      … //Formatting
      … //Plot Data
    }
```

Properties: 3



You may specify the Timer Interval in the Properties Window

Structure your Code properly!!
Define Classes and Methods which you can use here


DEMO

Summary

We have done the following:

1. Designed the Database using ERwin
2. Implemented Tables, Views, Stored Procedures and Triggers using SQL Server
3. Created a Datalogging App using LabVIEW that saves Measurement Data into the Database
4. Created a Data Monitoring App using Visual Studio/C# where we retrieve Data from the SQL Server Database

Improvements

The Examples shown is a simple and straightforward solution, but it is a minimal solution where many improvements can be done, some examples are:

- The Database Design and structure can be further improved
- Monitoring App: When updating: Get only the latest value
- Use TimeStamp values on the x-axis instead of 1, 2, 3, ..
- Number of Decimals
- General improvements in GUI
- General improvements in LabVIEW and C# Code
- Web Services could be used instead of direct access to the Database
- etc.

Recommended Litterature

- Tutorial: Introduction to LabVIEW
  http://home.hit.no/~hansha/?page=labview
- Tutorial: Introduction to Database Systems
  http://home.hit.no/~hansha/?tutorial=database
- Tutorial: Structured Query Language (SQL)
  http://home.hit.no/~hansha/?tutorial=sql
- Tutorial: Database Communication in LabVIEW
  http://home.hit.no/~hansha/?tutorial=database_labview
- Tutorial: Using SQL Server in C#
- Tutorial: Introduction to Visual Studio and C#
  http://home.hit.no/~hansha/?tutorial=csharp
- Tutorial: Data Acquisition in LabVIEW
  http://home.hit.no/~hansha/?tutorial=daq

Hans-Petter Halvorsen, M.Sc.

University College of Southeast Norway

www.usn.no

E-mail: hans.p.halvorsen@hit.no

Blog: http://home.hit.no/~hansha/