Database Overview

Introduction to Database Systems, ERwin, SQL Server, SQL, etc.

Hans-Petter Halvorsen, M.Sc.

Contents

- Database Modelling/Design using ERwin
- Generate SQL Table Script using ERwin
- Generate Tables in SQL Server using the SQL Script generated by ERwin
- Use Structured Query Language (SQL)
- Create Stored Procedures, View, Triggers
- Database Communication in LabVIEW
- Database Communication in C#

Necessary Software

- **ERwin** (CA ERwin Data Modeler Community Edition, free download from Internet)
- **SQL Server** (Express) Edition (Download for free from Internet or DreamSpark: “SQL Server xxxx Express with Tools”)
- LabVIEW
- DAQmx Driver Software
- LabVIEW SQL Toolkit (© Hans-Petter Halvorsen)
- Visual Studio

Make sure to install the necessary Software before you go to the laboratory!

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

https://www.youtube.com/watch?v=n75iPNrzN-o

Database Systems

Hans-Petter Halvorsen, M.Sc.

Old fashion Database (Data-storage) Systems

Not too long ago, this was the only data-storage device most companies needed. Those days are over.

Theory

Database Systems

A Database is a structured way to store lots of information.

The information 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.
- Fronter, TimeEdit, etc.
- … lots of other examples!

Database Management Systems (DBMS)

Theory

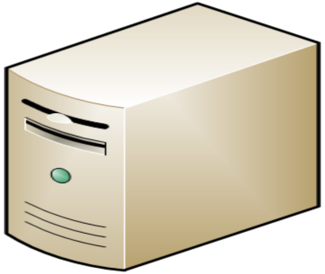
- **Microsoft SQL Server**
  - Enterprise, Developer versions, etc. (Professional use)
  - Express version is free of charge
- **Oracle**
- **MySQL** (owned by Oracle, but previously owned by Sun Microsystems) - MySQL can be used free of charge (open source license), Web sites that use MySQL: YouTube, Wikipedia, Facebook
- Microsoft Access
- IBM DB2
- Sybase, etc. (we have hundreds different DBMS)

We will use **SQL server** because it is very popular in the industry today, and we can use it for free via the **Microsoft DreamSpark Premium Subscription** – which is available for the students and staff at Telemark University College, or use the Express version which is available for free for everybody.

Microsoft SQL Server

Software

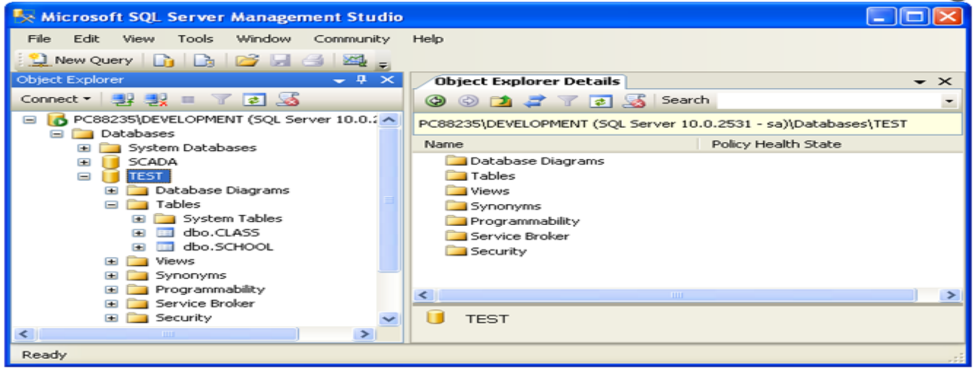
SQL Server consists of a **Database Engine** and a **Management Studio**. The **Database Engine** has no graphical interface - it is just a service running in the background of your computer (preferable on the server). The **Management Studio** is graphical tool for configuring and viewing the information in the database. It can be installed on the server or on the client (or both).



Database Engine

A Service running on the computer in the background

Management Studio



```
PC88235\DEVELOPMENT (SQL Server 10.0.2
  Databases
    System Databases
    SCADA
    TEST
      Database Diagrams
      Tables
        System Tables
        dbo.CLASS
        dbo.SCHOOL
      Views
      Synonyms
      Programmability
      Service Broker
      Security
```

PC88235\DEVELOPMENT (SQL Server 10.0.2531 - sa)\Databases\TEST

Name	Policy Health State
Database Diagrams	
Tables	
Views	
Synonyms	
Programmability	
Service Broker	
Security	

TEST

A Graphical User Interface to the database used for configuration and management of the database https://www.youtube.com/watch?v=P3n6hRNup8Y

ERwin

Database Design & Modelling

Hans-Petter Halvorsen, M.Sc.

Software

CA ERwin Data Modeler Community Edition

- Free!
- Max 25 Tables (good enough for our purpose)
- Download here:

http://erwin.com/products/data-modeler/community-edition

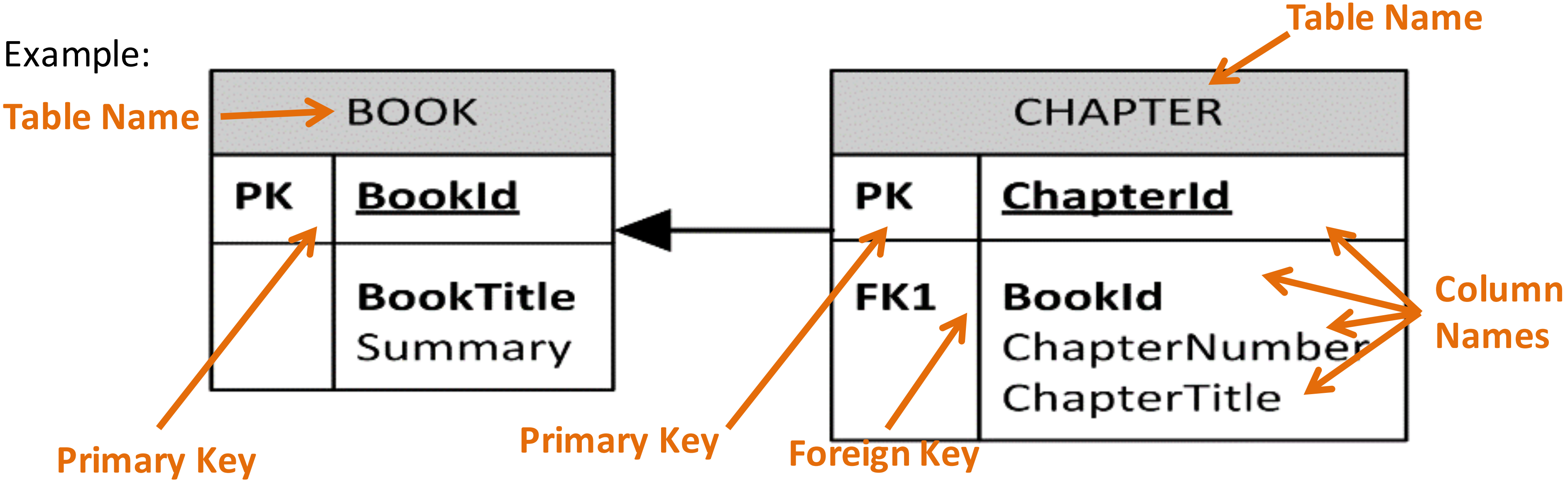
Database Design – ER Diagram

Theory

ER Diagram (Entity-Relationship Diagram)

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

Example:



BOOK	
PK	<u>BookId</u>
	BookTitle Summary

CHAPTER	
PK	<u>ChapterId</u>
FK1	BookId ChapterNumber ChapterTitle

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.

Database - “Best Practice”

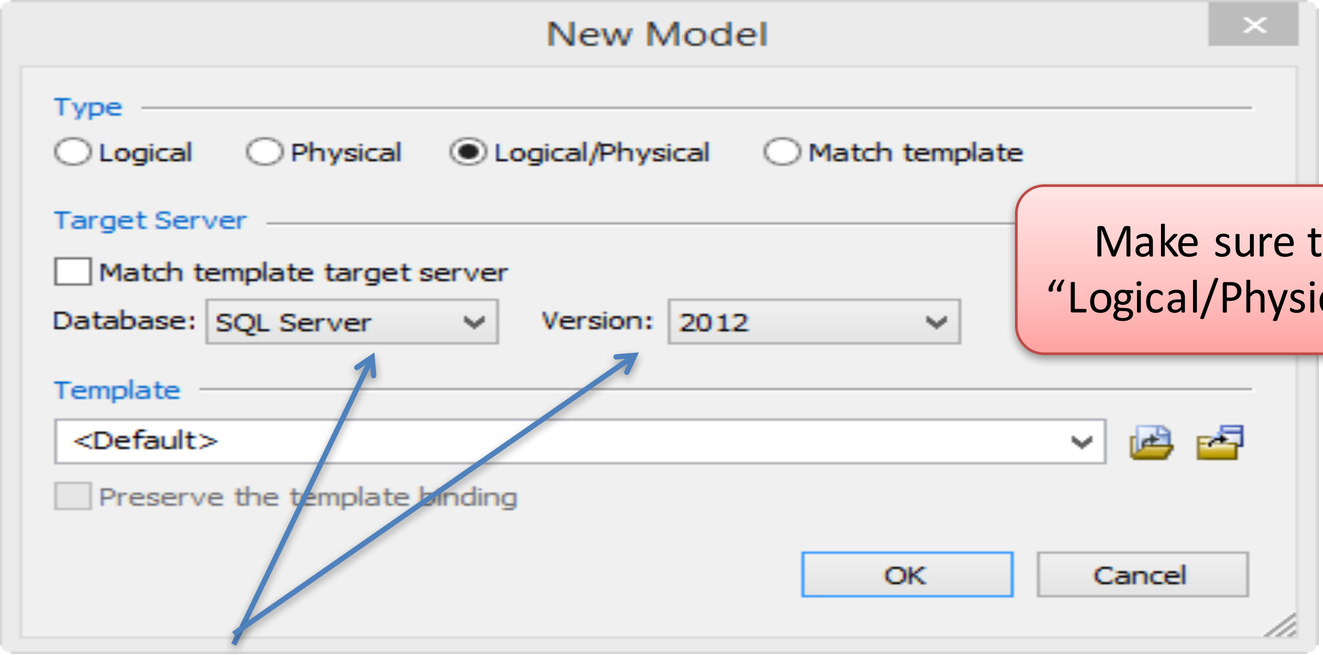
- **Tables**: Use upper case and singular form in table names – not plural, e.g., “STUDENT” (not “students”)
- **Columns**: Use Pascal notation, e.g., “StudentId”
- **Primary Key**:
  - If the table name is “COURSE”, name the Primary Key column “CourseId”, etc.
  - “Always” use Integer and Identity(1,1) for Primary Keys. Use UNIQUE constraint for other columns that needs to be unique, e.g. “RoomNumber”
- Specify **Required** Columns (NOT NULL) – i.e., which columns that need to have data or not
- Standardize on few/these **Data Types**: *int*, *float*, *varchar(x), datetime*, *bit*
- Use English for table and column names
- Avoid abbreviations! (Use “RoomNumber” – not “RoomNo”, “RoomNr”, ...)

It is recommended that you follow these guidelines!

Introduction to ERwin

Open ERwin and select File->New...

The following window appears (New Model):



New Model

Type

○ Logical ○ Physical ● Logical/Physical ○ Match template

Target Server

□ Match template target server

Database: SQL Server

Version: 2012

Template

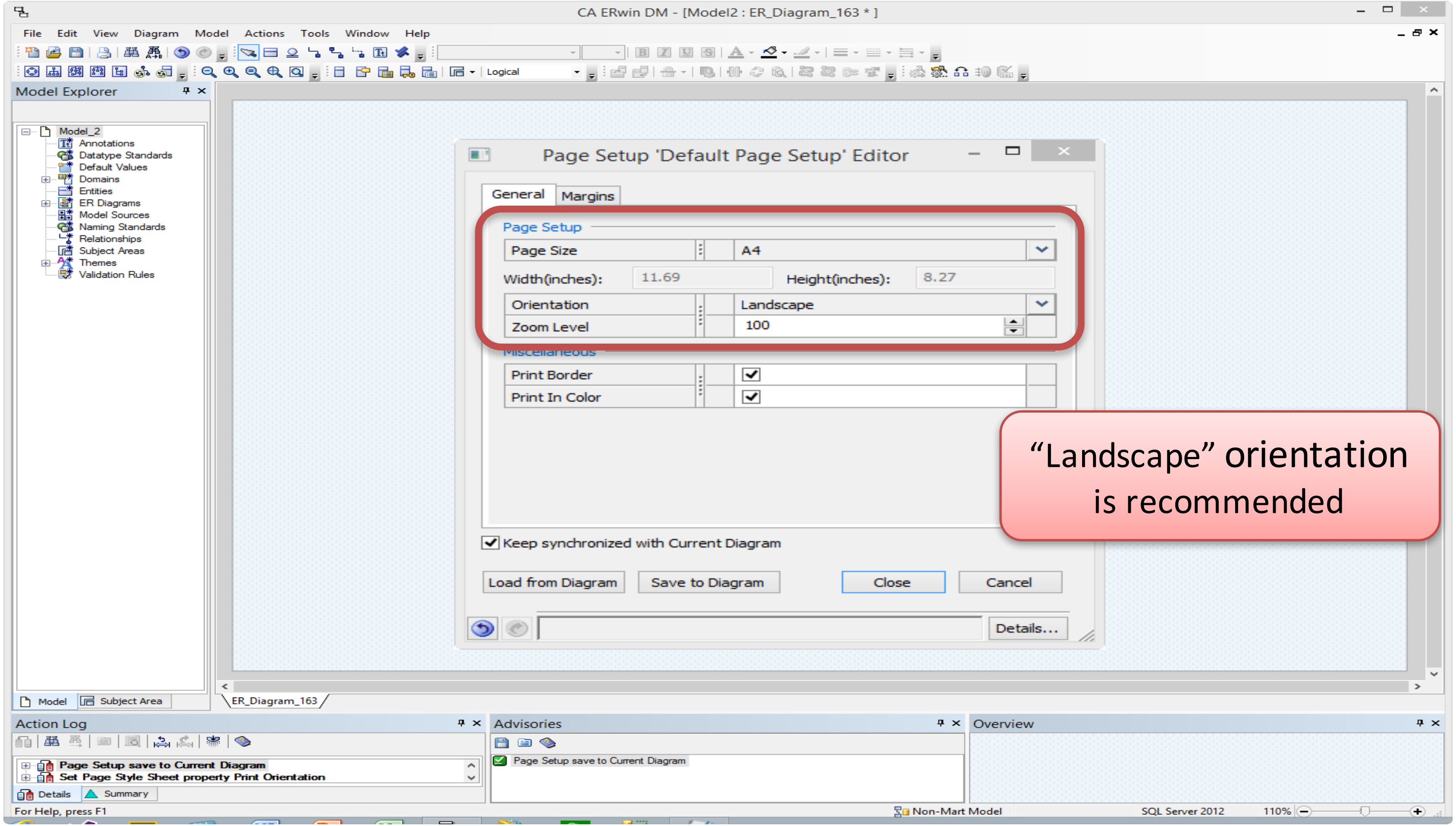
<Default>

□ Preserve the template binding

Make sure to select “Logical/Physical” Model

Select the Database Type and Version you shall use

Introduction to ERwin



CA ERwin DM - [Model2 : ER_Diagram_163 *]

Page Setup 'Default Page Setup' Editor

General | Margins

Page Setup

Page Size	A4
Width(inches): 11.69	Height(inches): 8.27
Orientation	Landscape
Zoom Level	100

Miscellaneous

Print Border	☑
Print In Color	☑

☑ Keep synchronized with Current Diagram

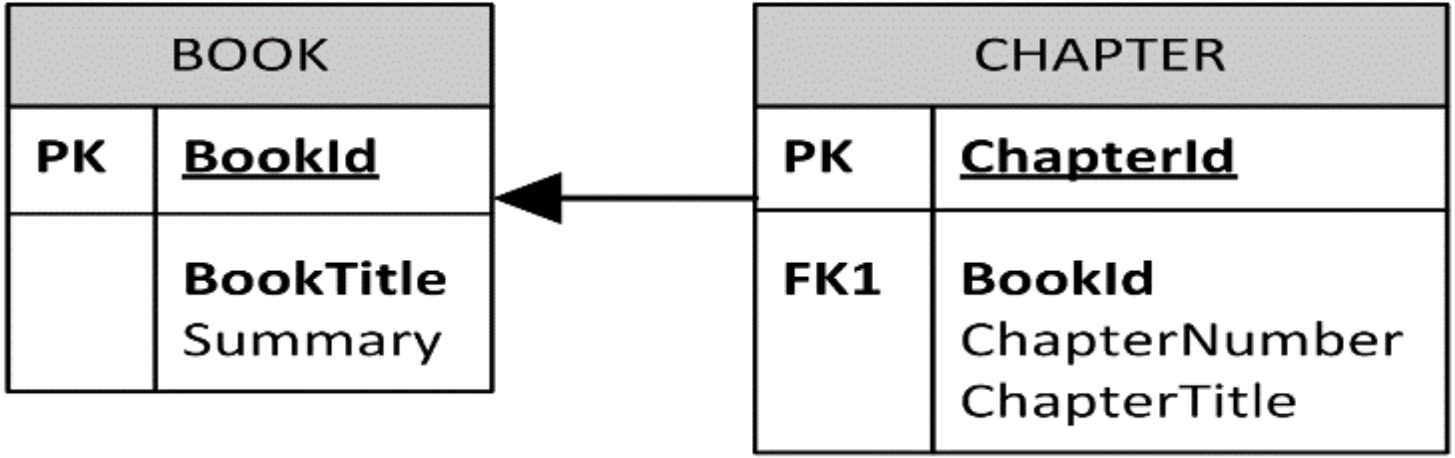
Action Log: Page Setup save to Current Diagram; Set Page Style Sheet property Print Orientation

Advisories: Page Setup save to Current Diagram

“Landscape” orientation is recommended

Introduction to ERwin

Try to create the following Tables, Columns, Primary Keys and Foreign Keys using ERwin:

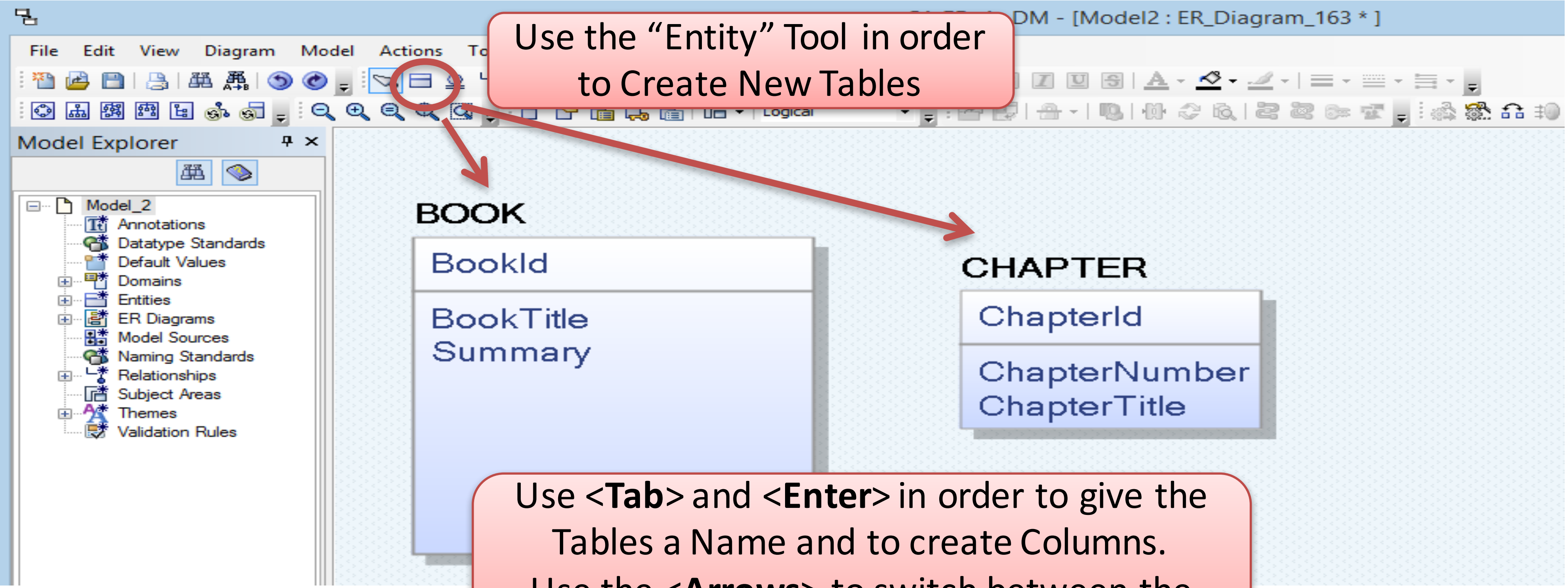


BOOK	
PK	**BookId**
	BookTitle Summary

CHAPTER	
PK	**ChapterId**
FK1	**BookId** ChapterNumber ChapterTitle

Introduction to ERwin

How-To: Create Tables and Columns

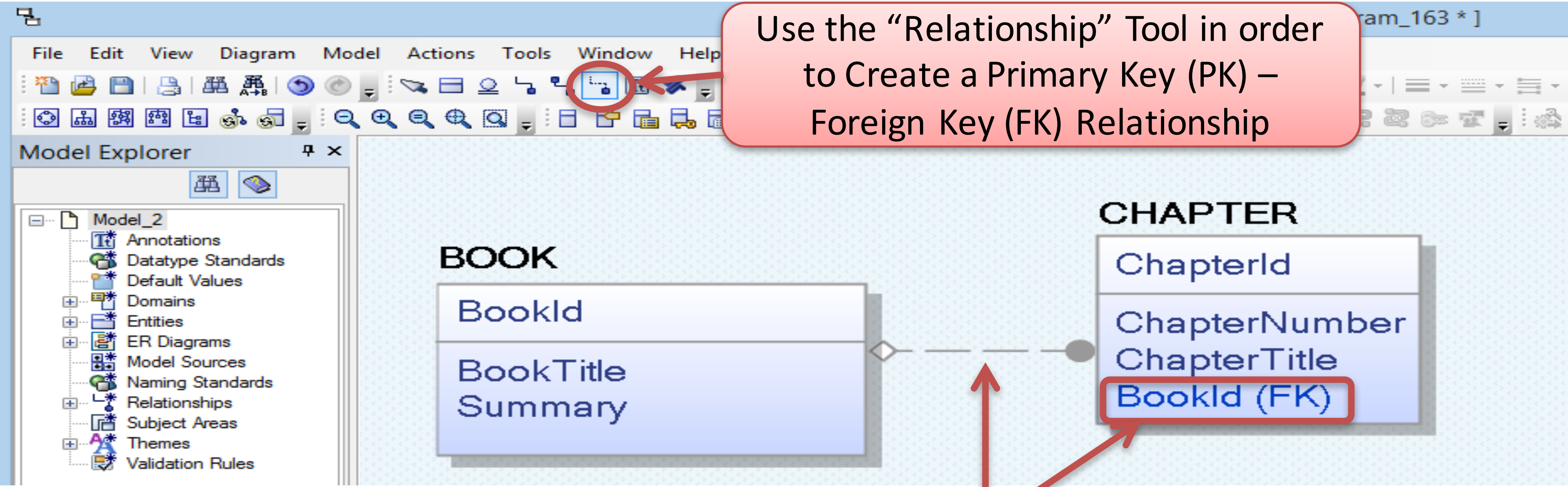


Use the “Entity” Tool in order to Create New Tables

Use <**Tab**> and <**Enter**> in order to give the Tables a Name and to create Columns. Use the <**Arrows**> to switch between the Columns inside a Table

Introduction to ERwin

How-To: Create Primary Key (PK) – Foreign Key (FK) Relationships:

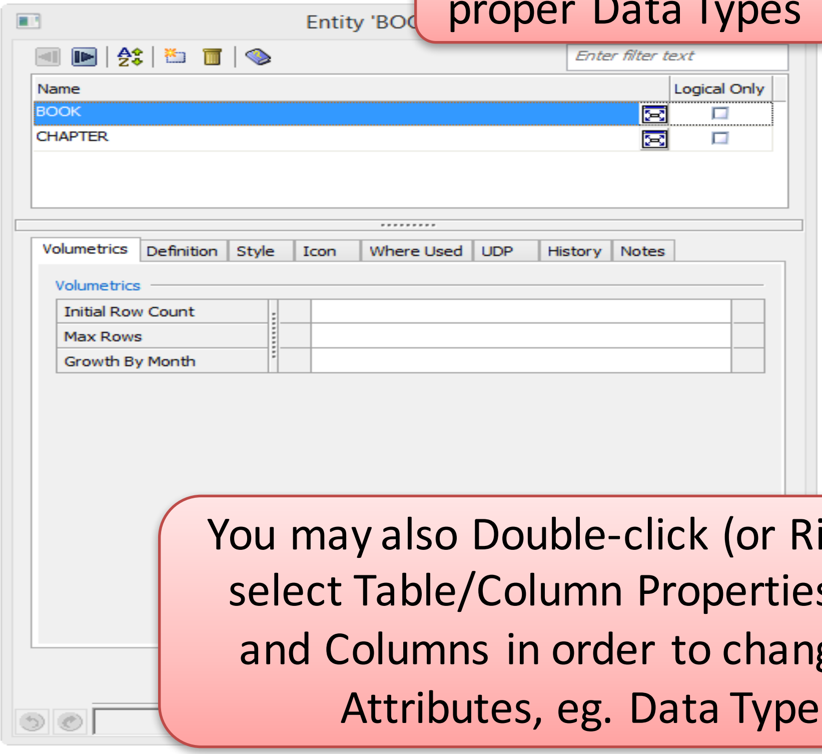


Use the “Relationship” Tool in order to Create a Primary Key (PK) – Foreign Key (FK) Relationship

Click first on the PK table and then on the FK table using the “Relationship” Tool. The Relationship Connection and the FK column itself is then Created Automatically

Setting Data Types (**Physical Model**)

Make sure to set proper Data Types



Entity 'BOO

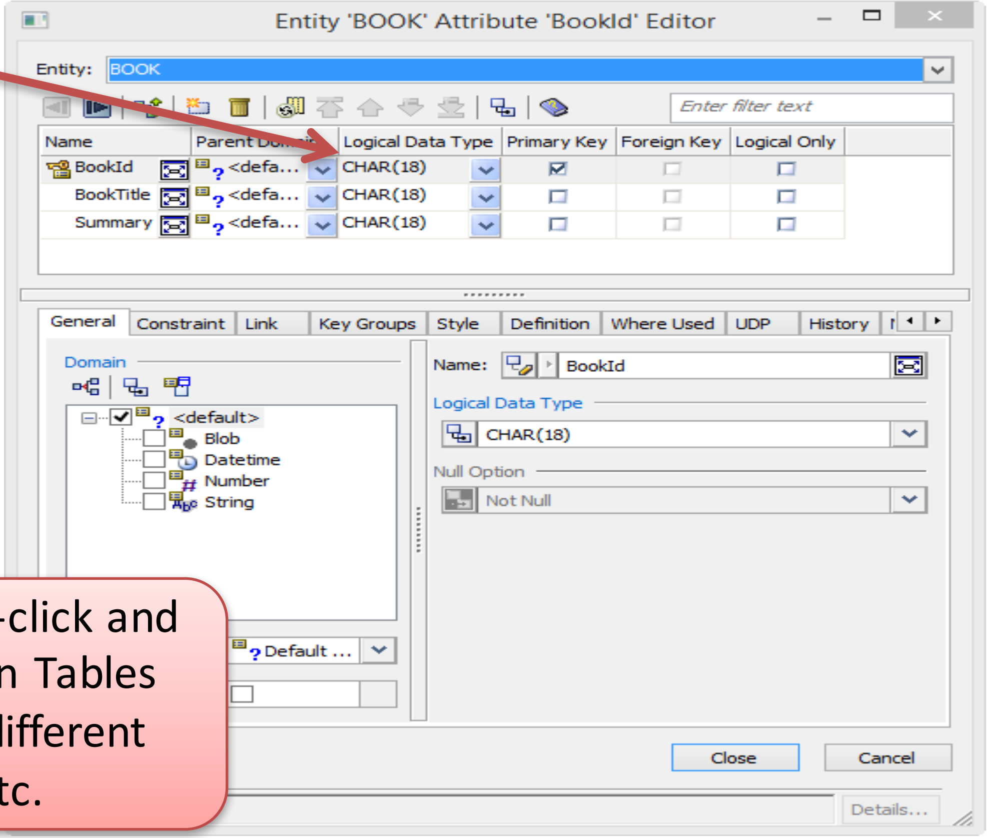
Enter filter text

Name	Logical Only
BOOK	□
CHAPTER	□

Volumetrics | Definition | Style | Icon | Where Used | UDP | History | Notes

Volumetrics

Initial Row Count	
Max Rows	
Growth By Month	



Entity 'BOOK' Attribute 'BookId' Editor

Entity: BOOK

Enter filter text

Name	Parent Domain	Logical Data Type	Primary Key	Foreign Key	Logical Only
BookId	<defa...	CHAR(18)	☑	□	□
BookTitle	<defa...	CHAR(18)	□	□	□
Summary	<defa...	CHAR(18)	□	□	□

General | Constraint | Link | Key Groups | Style | Definition | Where Used | UDP | History

Domain

- ☑ <default>
  - □ Blob
  - □ Datetime
  - □ Number
  - □ String

Default ...

Name: BookId

Logical Data Type

CHAR(18)

Null Option

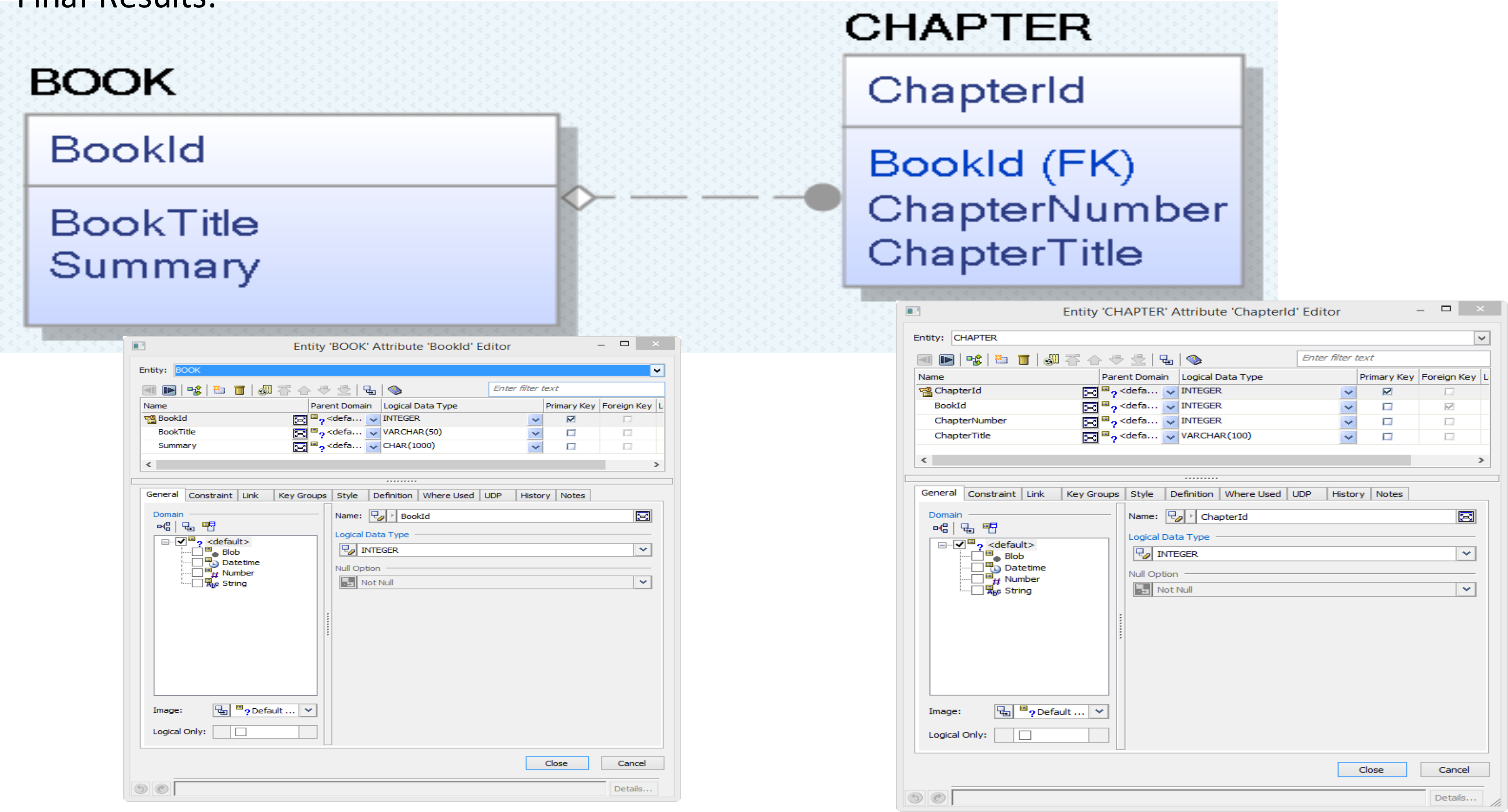
Not Null

Close | Cancel

Details...

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.

Final Results:



BOOK

BookId

BookTitle
Summary

CHAPTER

ChapterId

BookId (FK)
ChapterNumber
ChapterTitle

Entity 'BOOK' Attribute 'BookId' Editor

Entity: BOOK

Name	Parent Domain	Logical Data Type	Primary Key	Foreign Key
BookId	<defa...	INTEGER	☑	□
BookTitle	<defa...	VARCHAR(50)	□	□
Summary	<defa...	CHAR(1000)	□	□

General | Constraint | Link | Key Groups | Style | Definition | Where Used | UDP | History | Notes

Domain: ☑ <default> □ Blob □ Datetime □ Number □ String

Name: BookId

Logical Data Type: INTEGER

Null Option: Not Null

Image: Default ...

Logical Only: □

Close | Cancel | Details...

Entity 'CHAPTER' Attribute 'ChapterId' Editor

Entity: CHAPTER

Name	Parent Domain	Logical Data Type	Primary Key	Foreign Key
ChapterId	<defa...	INTEGER	☑	□
BookId	<defa...	INTEGER	□	☑
ChapterNumber	<defa...	INTEGER	□	□
ChapterTitle	<defa...	VARCHAR(100)	□	□

General | Constraint | Link | Key Groups | Style | Definition | Where Used | UDP | History | Notes

Domain: ☑ <default> □ Blob □ Datetime □ Number □ String

Name: ChapterId

Logical Data Type: INTEGER

Null Option: Not Null

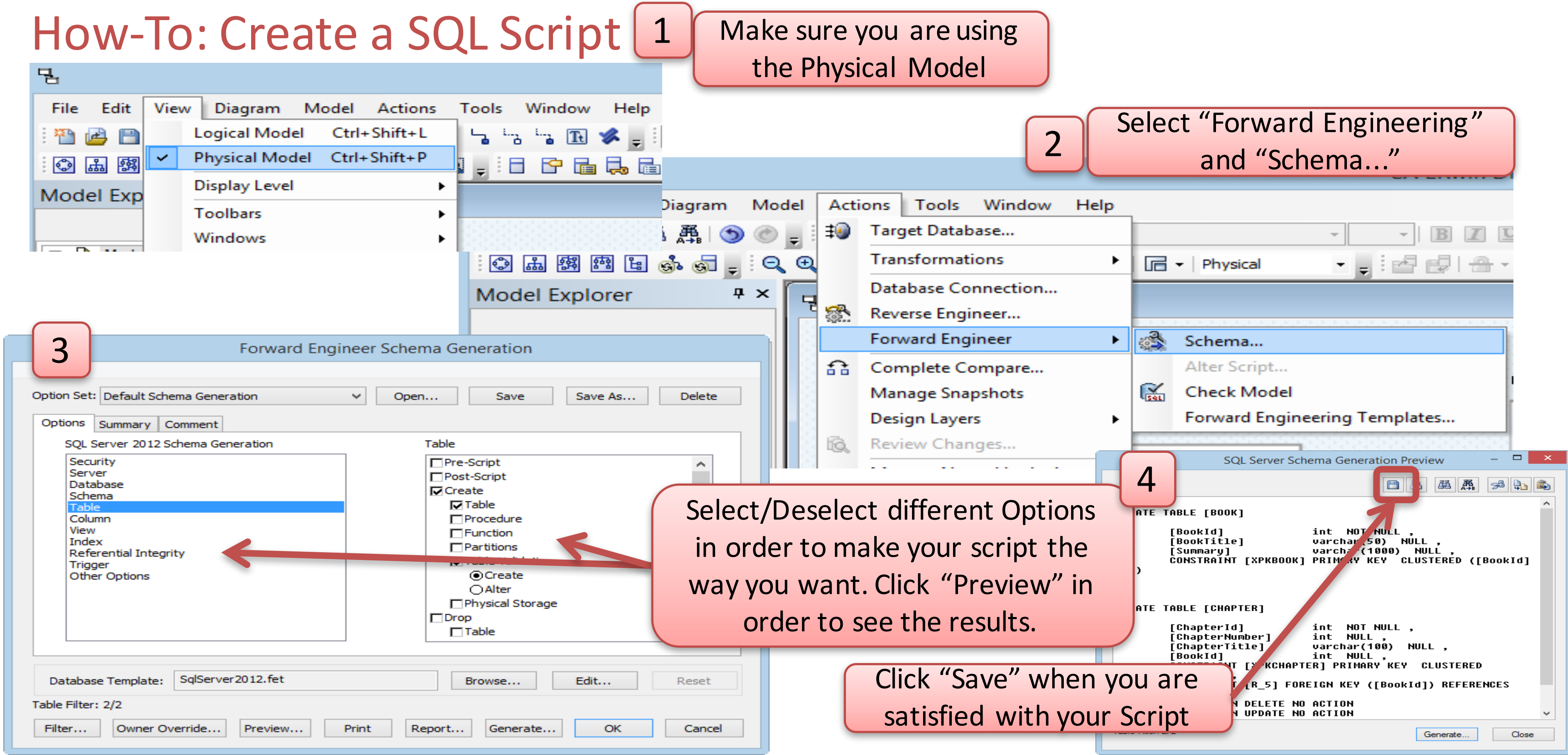
Image: Default ...

Logical Only: □

Close | Cancel | Details...

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

Forward Engineer Schema Generation

Option Set: Default Schema Generation

SQL Server 2012 Schema Generation

Table

☐ Pre-Script
☐ Post-Script
☑ Create
☑ Table
☐ Procedure
☐ Function
☐ Partitions
● Create
○ Alter
☐ Physical Storage
☐ Drop
☐ Table

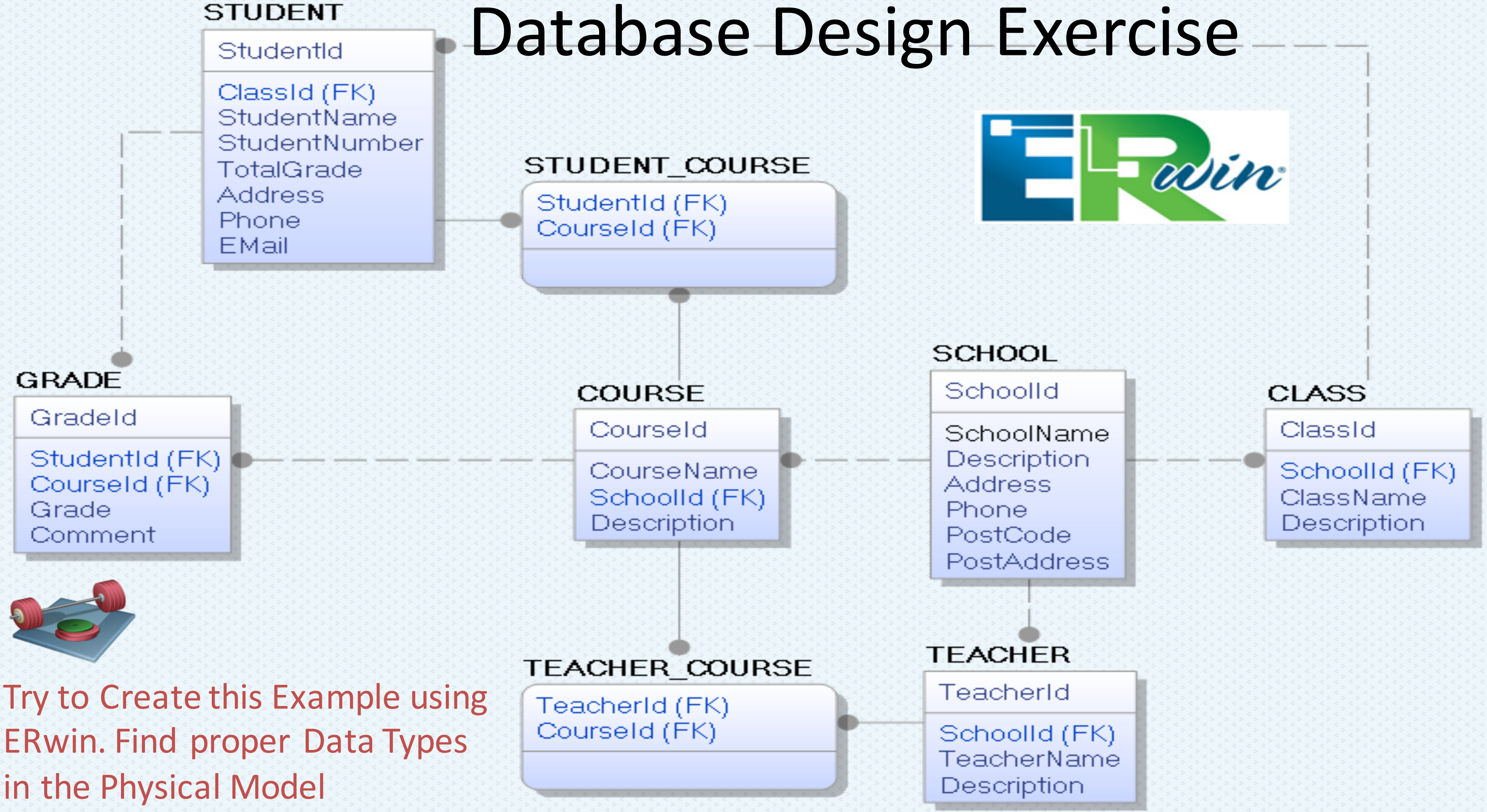
Database Template: SqlServer2012.fet

Table Filter: 2/2

SQL Server Schema Generation Preview

```
ATE TABLE [BOOK]

      [BookId]                int  NOT NULL ,
      [BookTitle]             varchar(50)  NULL ,
      [Summary]               varchar(1000)  NULL ,
      CONSTRAINT [XPKBOOK] PRIMARY KEY  CLUSTERED ([BookId]
)


ATE TABLE [CHAPTER]

      [ChapterId]             int  NOT NULL ,
      [ChapterNumber]         int  NULL ,
      [ChapterTitle]          varchar(100)  NULL ,
      [BookId]                int  NULL ,
      [XPKCHAPTER] PRIMARY KEY  CLUSTERED

      [R_5] FOREIGN KEY ([BookId]) REFERENCES

      N DELETE NO ACTION
      N UPDATE NO ACTION
```

Database Design Exercise



STUDENT
StudentId
ClassId (FK)
StudentName
StudentNumber
TotalGrade
Address
Phone
EMail

STUDENT_COURSE
StudentId (FK)
CourseId (FK)

GRADE
GradeId
StudentId (FK)
CourseId (FK)
Grade
Comment

COURSE
CourseId
CourseName
SchoolId (FK)
Description

SCHOOL
SchoolId
SchoolName
Description
Address
Phone
PostCode
PostAddress

CLASS
ClassId
SchoolId (FK)
ClassName
Description

TEACHER_COURSE
TeacherId (FK)
CourseId (FK)

TEACHER
TeacherId
SchoolId (FK)
TeacherName
Description

Try to Create this Example using ERwin. Find proper Data Types in the Physical Model

Congratulations! - You are finished with the Example

https://www.youtube.com/watch?v=SlR4KOhAG1U

SQL Server

Database Implementation

Hans-Petter Halvorsen, M.Sc.

SQL Server 2014 Installation

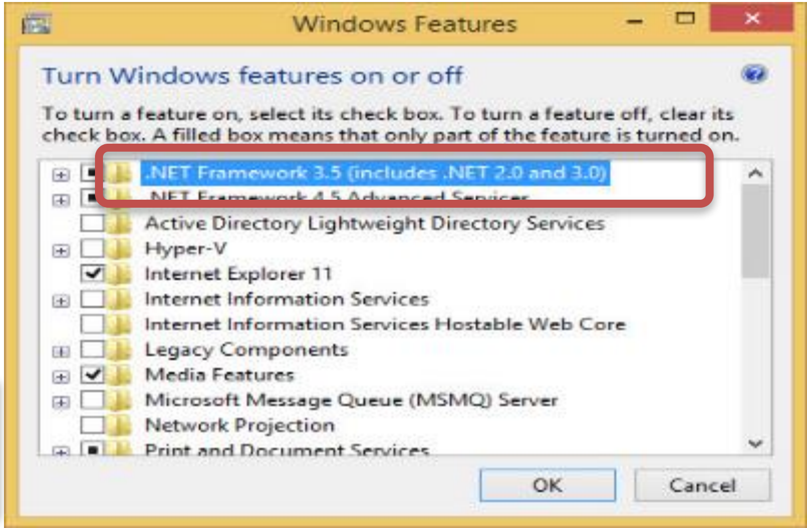
SQL Server has different Editions and Installation Packages. Here we will go through the installation of **SQL Server 2014 Express with Tools**.

Preparation 1: Download Installation Package (**SQL Server 2014 Express with Tools**) from Internet or DreamSparks.

Preparation 2: Are you using Windows 8/10? – You may need to install the .NET Framework 3.5 in advance:

- Go to Settings. Choose Control Panel then choose Programs.
- Click Turn Windows features on or off, and the user will see Windows feature window.
- You can enable this feature by click on .NET Framework 3.5 (include .NET 2.0 and 3.0) select it and click OK. After this step, it will download the entire package from internet and install the .NET Framework 3.5 feature.

Note! It takes some time to download, so make sure to do it before the Lab Work in Class!!



Windows Features

Turn Windows features on or off

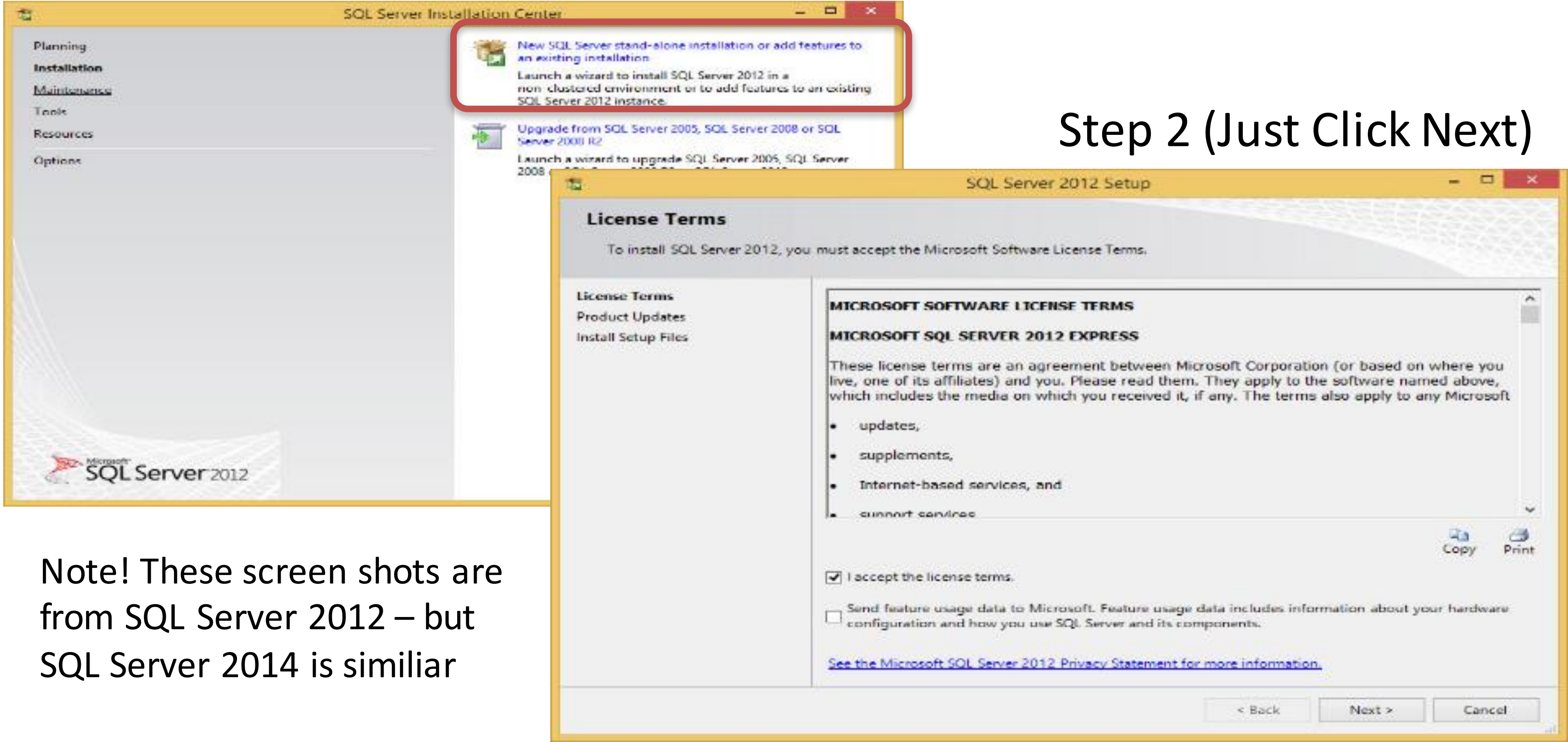
To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ⊞ ■ .NET Framework 3.5 (includes .NET 2.0 and 3.0)
- ⊞ ■ .NET Framework 4.5 Advanced Services
- □ Active Directory Lightweight Directory Services
- ⊞ □ Hyper-V
- ☑ Internet Explorer 11
- ⊞ □ Internet Information Services
- □ Internet Information Services Hostable Web Core
- ⊞ □ Legacy Components
- ⊞ ☑ Media Features
- ⊞ □ Microsoft Message Queue (MSMQ) Server
- □ Network Projection
- ⊞ ■ Print and Document Services

Start Installing SQL Server 2014 Express with Tools

Step 1

Step 2 (Just Click Next)



SQL Server Installation Center

New SQL Server stand-alone installation or add features to an existing installation

Launch a wizard to install SQL Server 2012 in a non-clustered environment or to add features to an existing SQL Server 2012 instance.

Upgrade from SQL Server 2005, SQL Server 2008 or SQL Server 2008 R2

Launch a wizard to upgrade SQL Server 2005, SQL Server 2008 ...

SQL Server 2012 Setup

License Terms

To install SQL Server 2012, you must accept the Microsoft Software License Terms.

MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT SQL SERVER 2012 EXPRESS

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft

- updates,
- supplements,
- Internet-based services, and
- support services

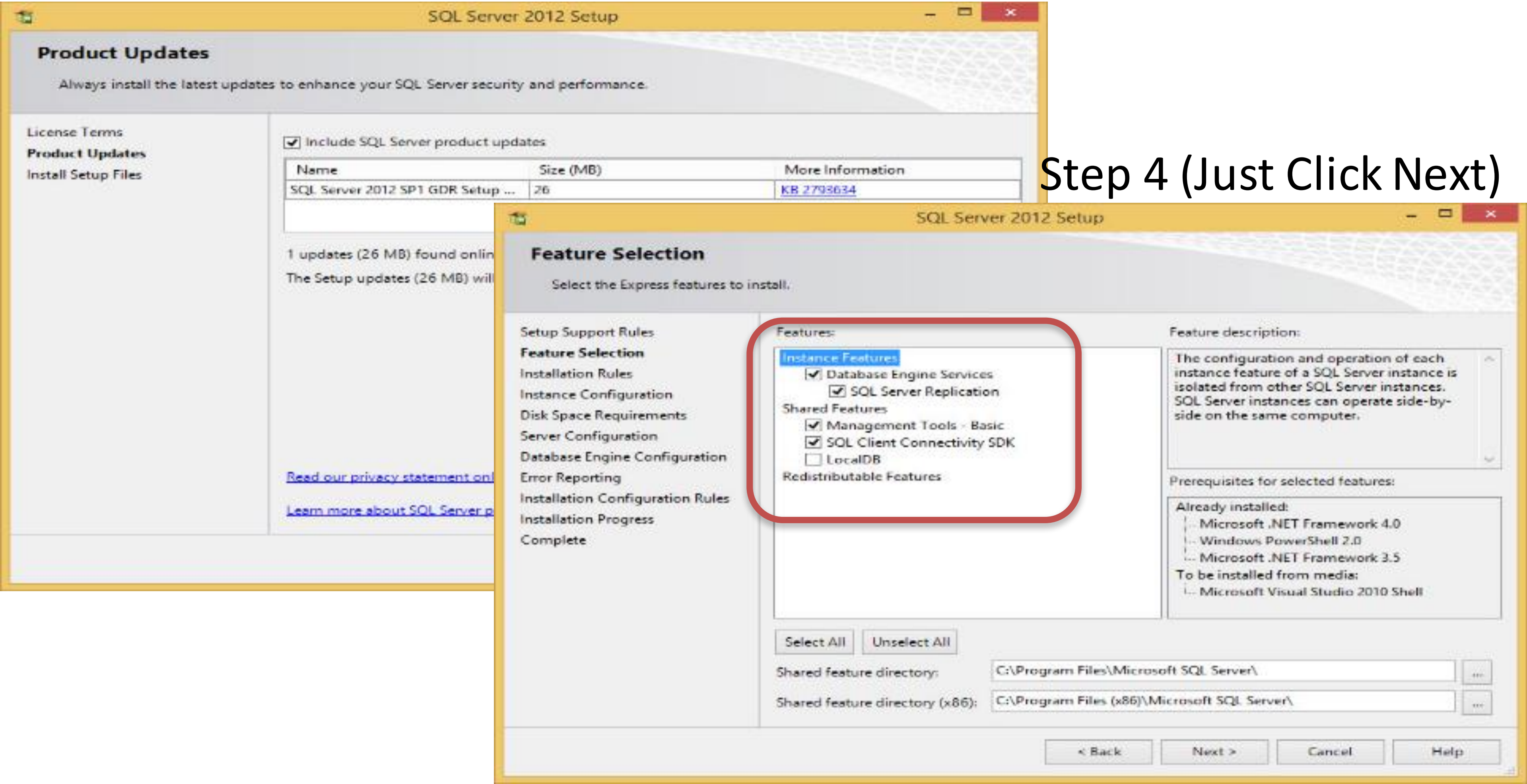
☑ I accept the license terms.

□ Send feature usage data to Microsoft. Feature usage data includes information about your hardware configuration and how you use SQL Server and its components.

See the Microsoft SQL Server 2012 Privacy Statement for more information.

Note! These screen shots are from SQL Server 2012 – but SQL Server 2014 is similiar

Step 3 (Just Click Next)



SQL Server 2012 Setup

Product Updates

Always install the latest updates to enhance your SQL Server security and performance.

☑ Include SQL Server product updates

Name	Size (MB)	More Information
SQL Server 2012 SP1 GDR Setup ...	26	KB 2793634

1 updates (26 MB) found onlin

The Setup updates (26 MB) will

Read our privacy statement onl

Learn more about SQL Server p

Step 4 (Just Click Next)

SQL Server 2012 Setup

Feature Selection

Select the Express features to install.

Features:

Instance Features
- ☑ Database Engine Services
  - ☑ SQL Server Replication

Shared Features
- ☑ Management Tools - Basic
- ☑ SQL Client Connectivity SDK
- □ LocalDB

Redistributable Features

Feature description:

The configuration and operation of each instance feature of a SQL Server instance is isolated from other SQL Server instances. SQL Server instances can operate side-by-side on the same computer.

Prerequisites for selected features:

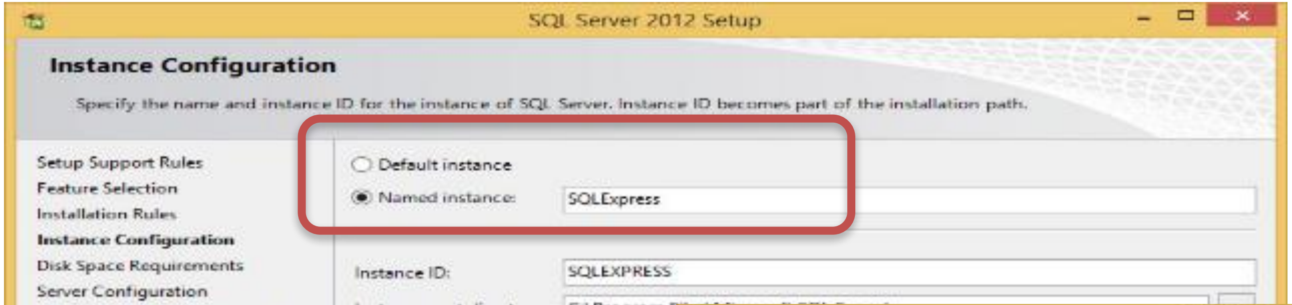
Already installed:
- Microsoft .NET Framework 4.0
- Windows PowerShell 2.0
- Microsoft .NET Framework 3.5

To be installed from media:
- Microsoft Visual Studio 2010 Shell

Shared feature directory: C:\Program Files\Microsoft SQL Server\

Shared feature directory (x86): C:\Program Files (x86)\Microsoft SQL Server\

Step 5(Use Default or change the Name if you want to)



SQL Server 2012 Setup

Instance Configuration

Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.

Setup Support Rules
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Installation Progress
Complete

○ Default instance

● Named instance: SQLExpress

Instance ID: SQLEXPRESS

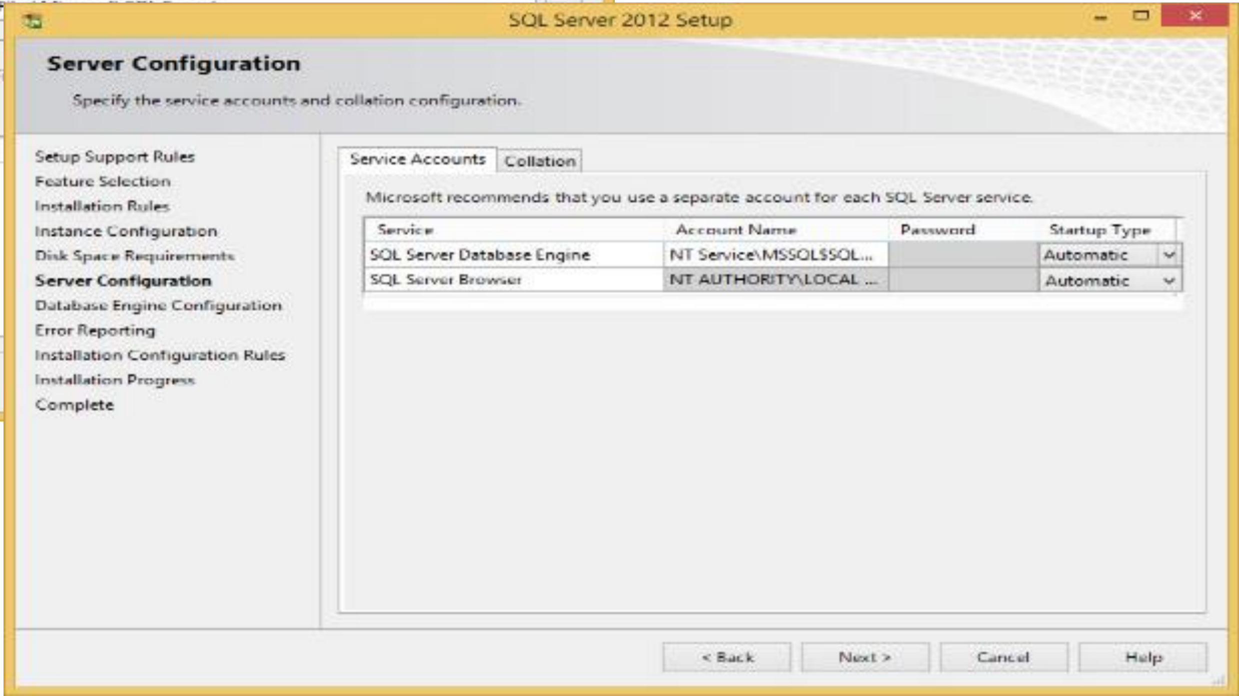
Instance root directory: C:\Program F

SQL Server directory: C:\Program F

Installed instances:

Instance Name	Instance ID

Step 6 (Just Click Next)



SQL Server 2012 Setup

Server Configuration

Specify the service accounts and collation configuration.

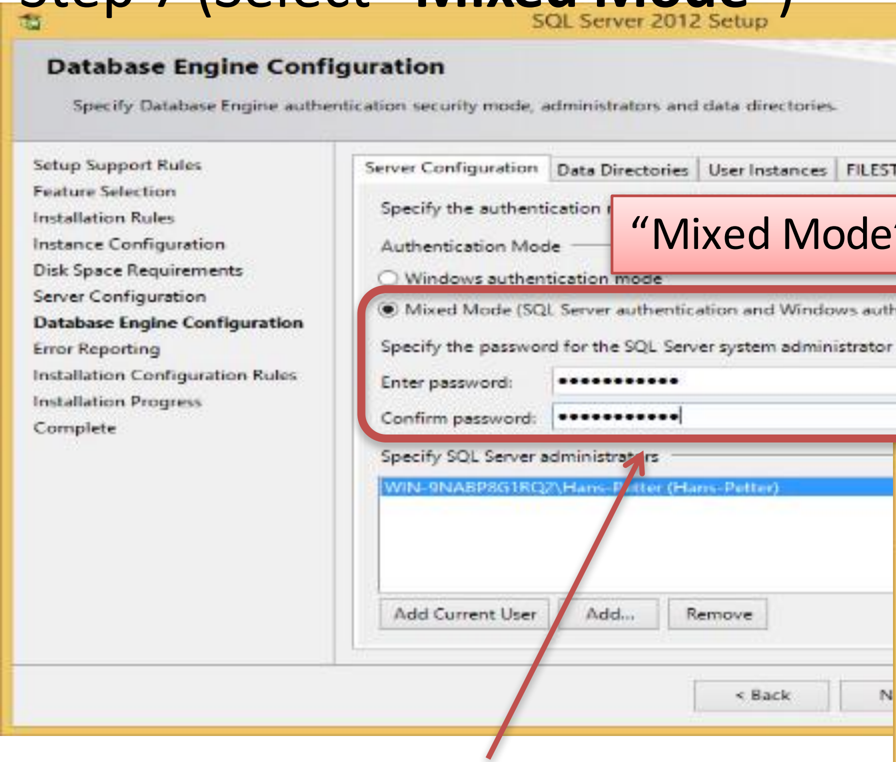
Setup Support Rules
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Installation Progress
Complete

Service Accounts | Collation

Microsoft recommends that you use a separate account for each SQL Server service.

Service	Account Name	Password	Startup Type
SQL Server Database Engine	NT Service\MSSQL$SQL...		Automatic
SQL Server Browser	NT AUTHORITY\LOCAL ...		Automatic

Step 7 (Select “**Mixed Mode**”)



SQL Server 2012 Setup

Database Engine Configuration

Specify Database Engine authentication security mode, administrators and data directories.

Setup Support Rules
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Installation Progress
Complete

Server Configuration | Data Directories | 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.

Enter password: ●●●●●●●●●●●●

Confirm password: ●●●●●●●●●●●●

Specify SQL Server administrators

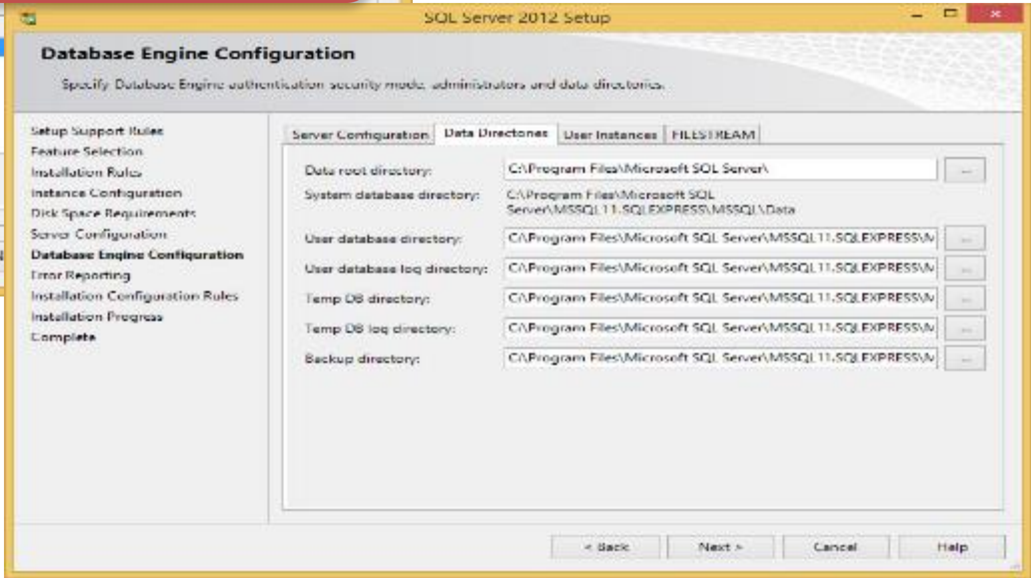
WIN-9NABP8G1RQ7\Hans-Petter (Hans-Petter)

“Mixed Mode” is recommended

Enter Password for “sa” user and make sure to remember it!!

Step 7b

(use Default location or change folder for the Database Files)



SQL Server 2012 Setup

Database Engine Configuration

Specify Database Engine authentication security mode, administrators and data directories.

Setup Support Rules
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Installation Progress
Complete

Server Configuration | Data Directories | User Instances | FILESTREAM

Data root directory: C:\Program Files\Microsoft SQL Server\

System database directory: C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\Data

User database directory: C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\M

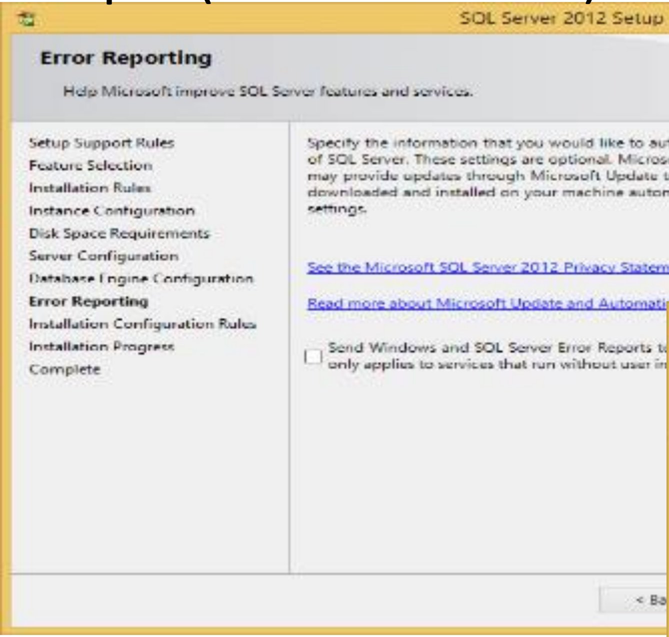
User database log directory: C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\M

Temp DB directory: C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\M

Temp DB log directory: C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\M

Backup directory: C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\M

Step 8 (Just Click Next)



Error Reporting

Help Microsoft improve SQL Server features and services.

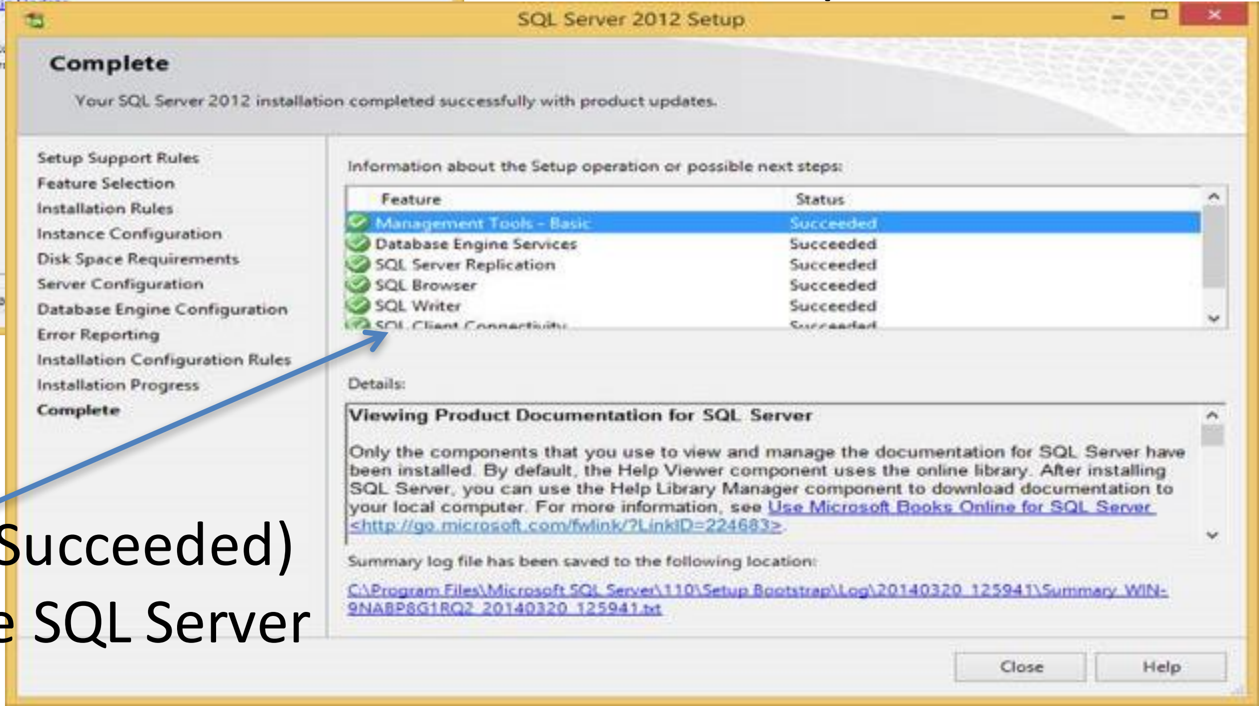
Specify the information that you would like to automatically send to Microsoft to improve future releases of SQL Server. These settings are optional. Microsoft treats this information as confidential. Microsoft may provide updates through Microsoft Update to modify feature usage data. These updates might be downloaded and installed on your machine automatically, depending on your Automatic Update settings.

See the Microsoft SQL Server 2012 Privacy Statement for more information.

Read more about Microsoft Update and Automati

□ Send Windows and SQL Server Error Reports t
only applies to services that run without user in

Step 9 – Finished!



Complete

Your SQL Server 2012 installation completed successfully with product updates.

Information about the Setup operation or possible next steps:

Feature	Status
Management Tools - Basic	Succeeded
Database Engine Services	Succeeded
SQL Server Replication	Succeeded
SQL Browser	Succeeded
SQL Writer	Succeeded
SQL Client Connectivity	Succeeded

Details:

Viewing Product Documentation for SQL Server

Only the components that you use to view and manage the documentation for SQL Server have been installed. By default, the Help Viewer component uses the online library. After installing SQL Server, you can use the Help Library Manager component to download documentation to your local computer. For more information, see Use Microsoft Books Online for SQL Server <http://go.microsoft.com/fwlink/?LinkID=224683>.

Summary log file has been saved to the following location:

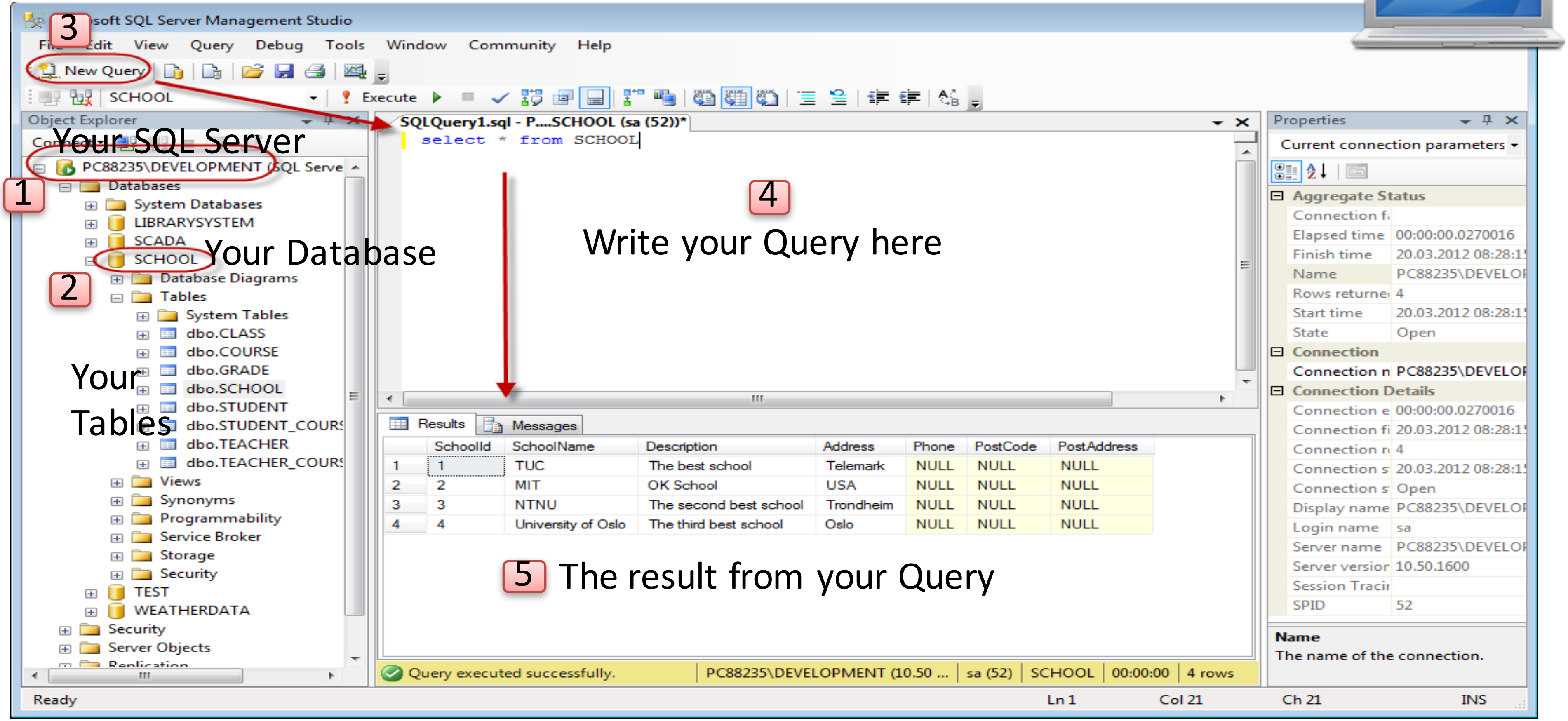
C:\Program Files\Microsoft SQL Server\110\Setup Bootstrap\Log\20140320_125941\Summary_WIN-9NABP8G1RQ2_20140320_125941.txt

Hopefully are all Green (Succeeded)

You are now ready to use SQL Server

Microsoft SQL Server Management Studio

Software



```
select * from SCHOOL
```

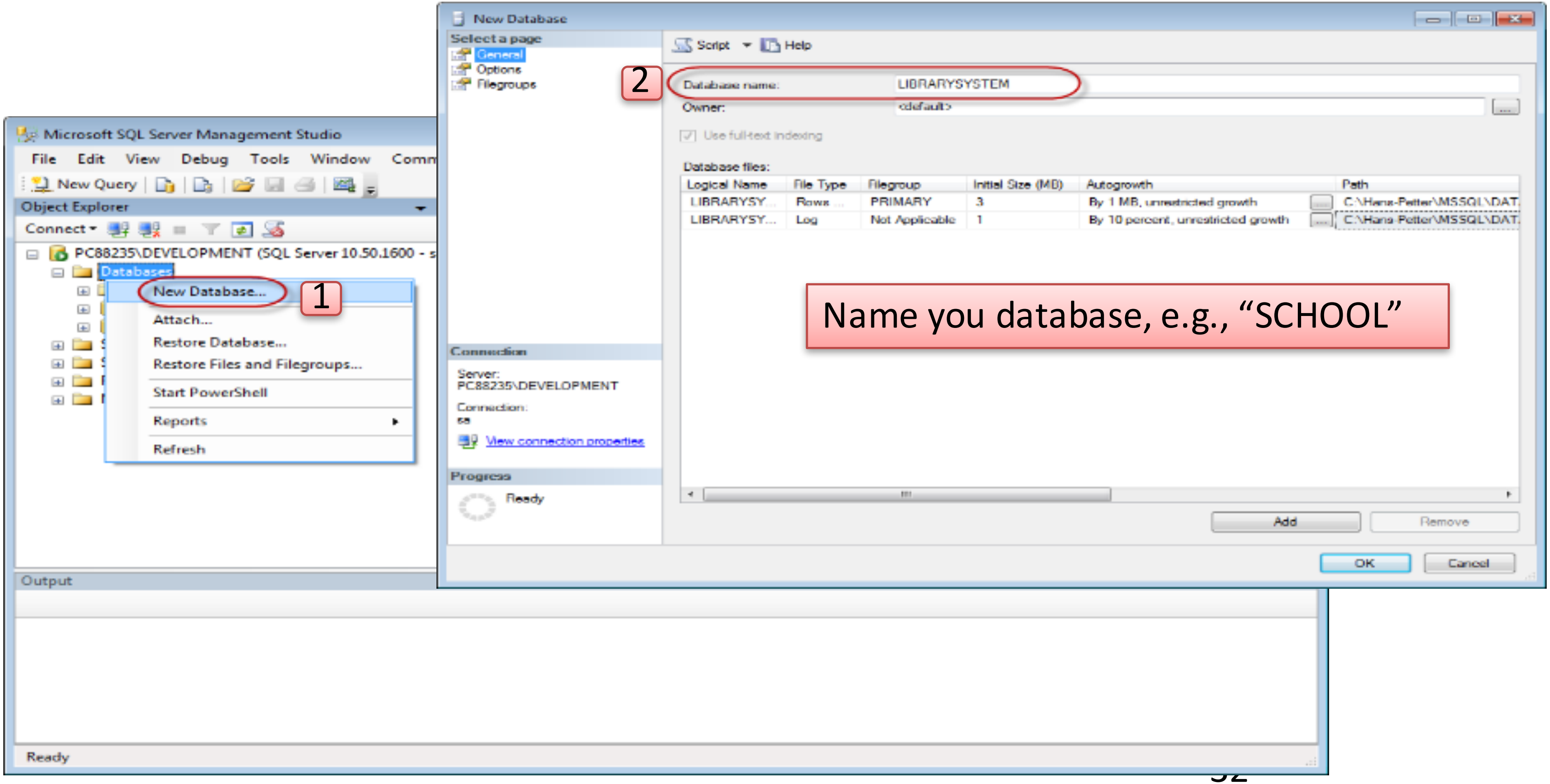
	SchoolId	SchoolName	Description	Address	Phone	PostCode	PostAddress
1	1	TUC	The best school	Telemark	NULL	NULL	NULL
2	2	MIT	OK School	USA	NULL	NULL	NULL
3	3	NTNU	The second best school	Trondheim	NULL	NULL	NULL
4	4	University of Oslo	The third best school	Oslo	NULL	NULL	NULL

Query executed successfully. | PC88235\DEVELOPMENT (10.50 ... | sa (52) | SCHOOL | 00:00:00 | 4 rows

Aggregate Status	
Connection fa	
Elapsed time	00:00:00.0270016
Finish time	20.03.2012 08:28:1
Name	PC88235\DEVELOP
Rows returned	4
Start time	20.03.2012 08:28:1
State	Open
Connection	
Connection n	PC88235\DEVELOP
Connection Details	
Connection e	00:00:00.0270016
Connection fi	20.03.2012 08:28:1
Connection r	4
Connection s	20.03.2012 08:28:1
Connection s	Open
Display name	PC88235\DEVELOP
Login name	sa
Server name	PC88235\DEVELOP
Server versior	10.50.1600
Session Tracir	
SPID	52

Name
The name of the connection.

Microsoft SQL Server – Create a New Database



1 New Database...

2 Database name: LIBRARYSYSTEM

Owner: <default>

☑ Use full-text indexing

Server: PC88235\DEVELOPMENT

Connection: sa

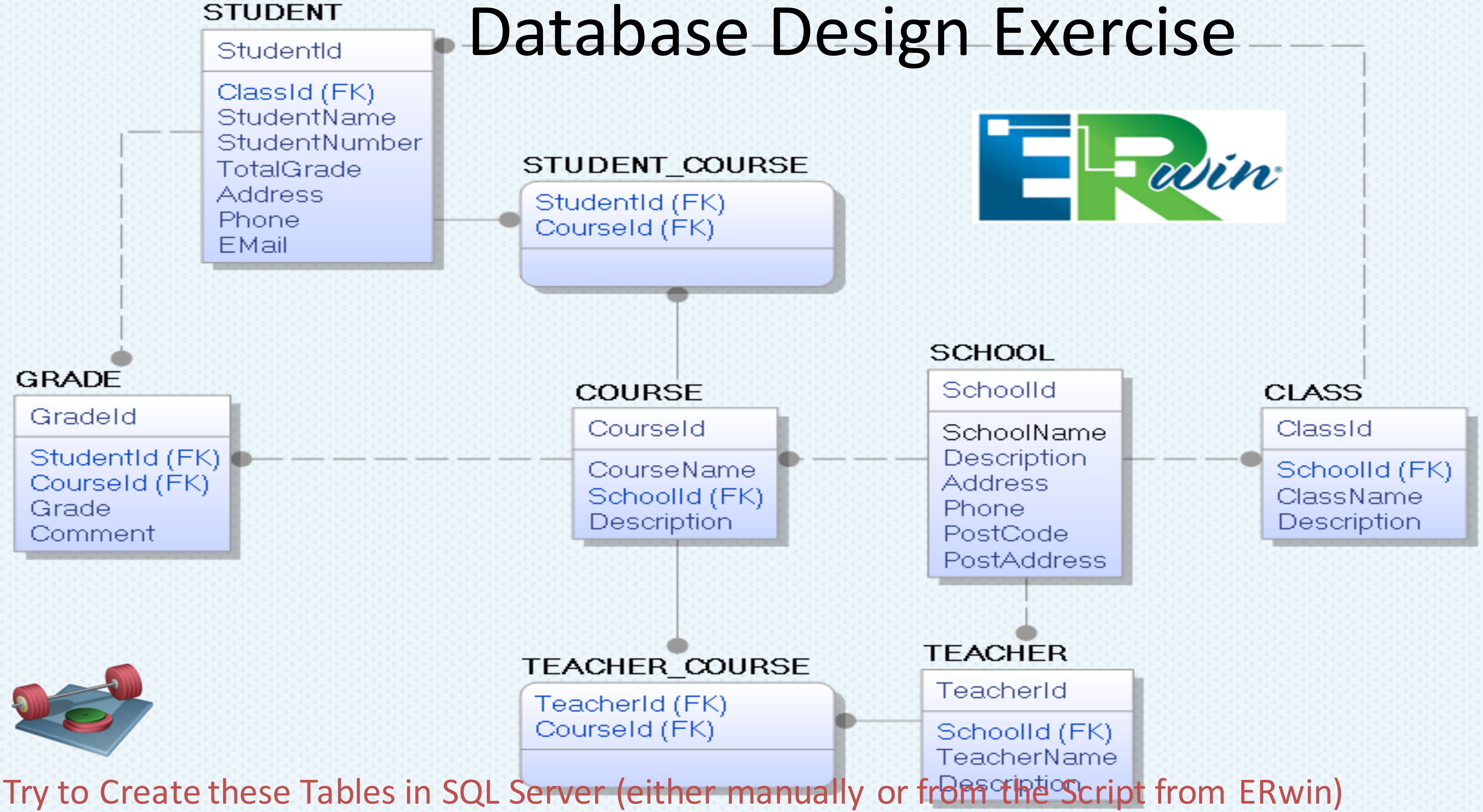
View connection properties

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth	Path
LIBRARYSY...	Rows ...	PRIMARY	3	By 1 MB, unrestricted growth	C:\Hans-Petter\MSSQL\DAT
LIBRARYSY...	Log	Not Applicable	1	By 10 percent, unrestricted growth	C:\Hans-Petter\MSSQL\DAT

Name you database, e.g., “SCHOOL”

Database Design Exercise



STUDENT
- StudentId
- ClassId (FK)
- StudentName
- StudentNumber
- TotalGrade
- Address
- Phone
- EMail

STUDENT_COURSE
- StudentId (FK)
- CourseId (FK)

GRADE
- GradeId
- StudentId (FK)
- CourseId (FK)
- Grade
- Comment

COURSE
- CourseId
- CourseName
- SchoolId (FK)
- Description

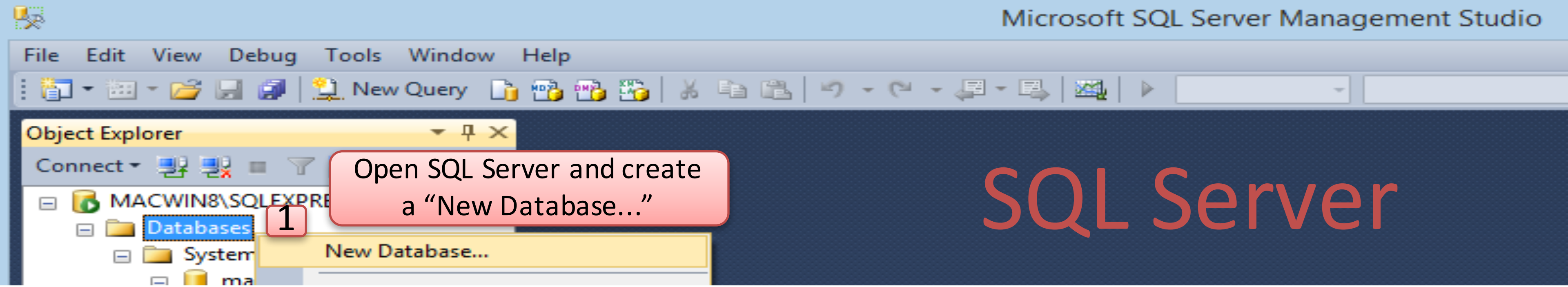
SCHOOL
- SchoolId
- SchoolName
- Description
- Address
- Phone
- PostCode
- PostAddress

CLASS
- ClassId
- SchoolId (FK)
- ClassName
- Description

TEACHER_COURSE
- TeacherId (FK)
- CourseId (FK)

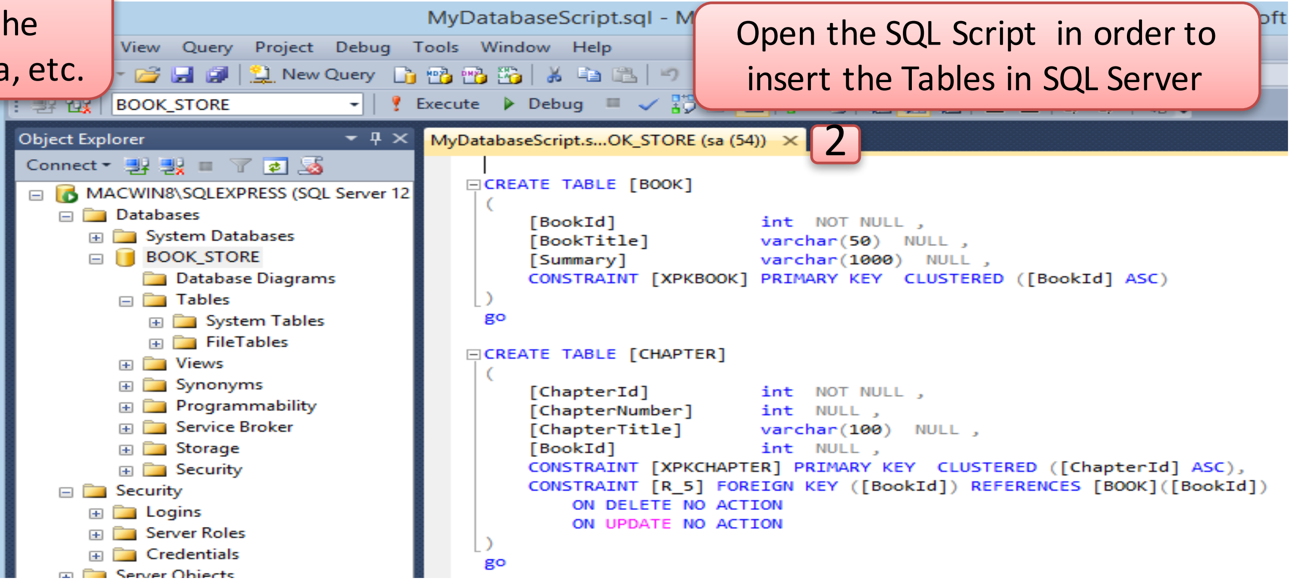
TEACHER
- TeacherId
- SchoolId (FK)
- TeacherName
- Description

Try to Create these Tables in SQL Server (either manually or from the Script from ERwin)



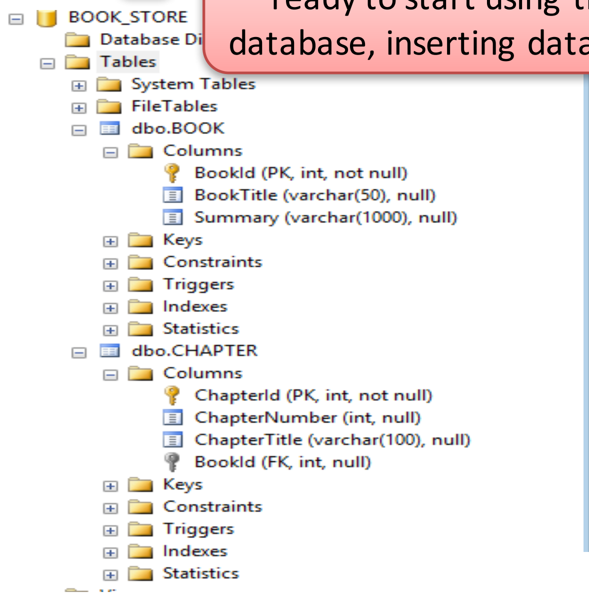
1 Open SQL Server and create a "New Database..."

SQL Server



2 Open the SQL Script in order to insert the Tables in SQL Server

```
CREATE TABLE [BOOK]
(
    [BookId]              int  NOT NULL ,
    [BookTitle]           varchar(50)  NULL ,
    [Summary]             varchar(1000)  NULL ,
    CONSTRAINT [XPKBOOK] PRIMARY KEY  CLUSTERED ([BookId] ASC)
)
go

CREATE TABLE [CHAPTER]
(
    [ChapterId]           int  NOT NULL ,
    [ChapterNumber]       int  NULL ,
    [ChapterTitle]        varchar(100)  NULL ,
    [BookId]              int  NULL ,
    CONSTRAINT [XPKCHAPTER] PRIMARY KEY  CLUSTERED ([ChapterId] ASC),
    CONSTRAINT [R_5] FOREIGN KEY ([BookId]) REFERENCES [BOOK]([BookId])
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
go
```



3 You are Finished. You are ready to start using the database, inserting data, etc.

Microsoft SQL Server

Do you get an error when trying to change your tables?

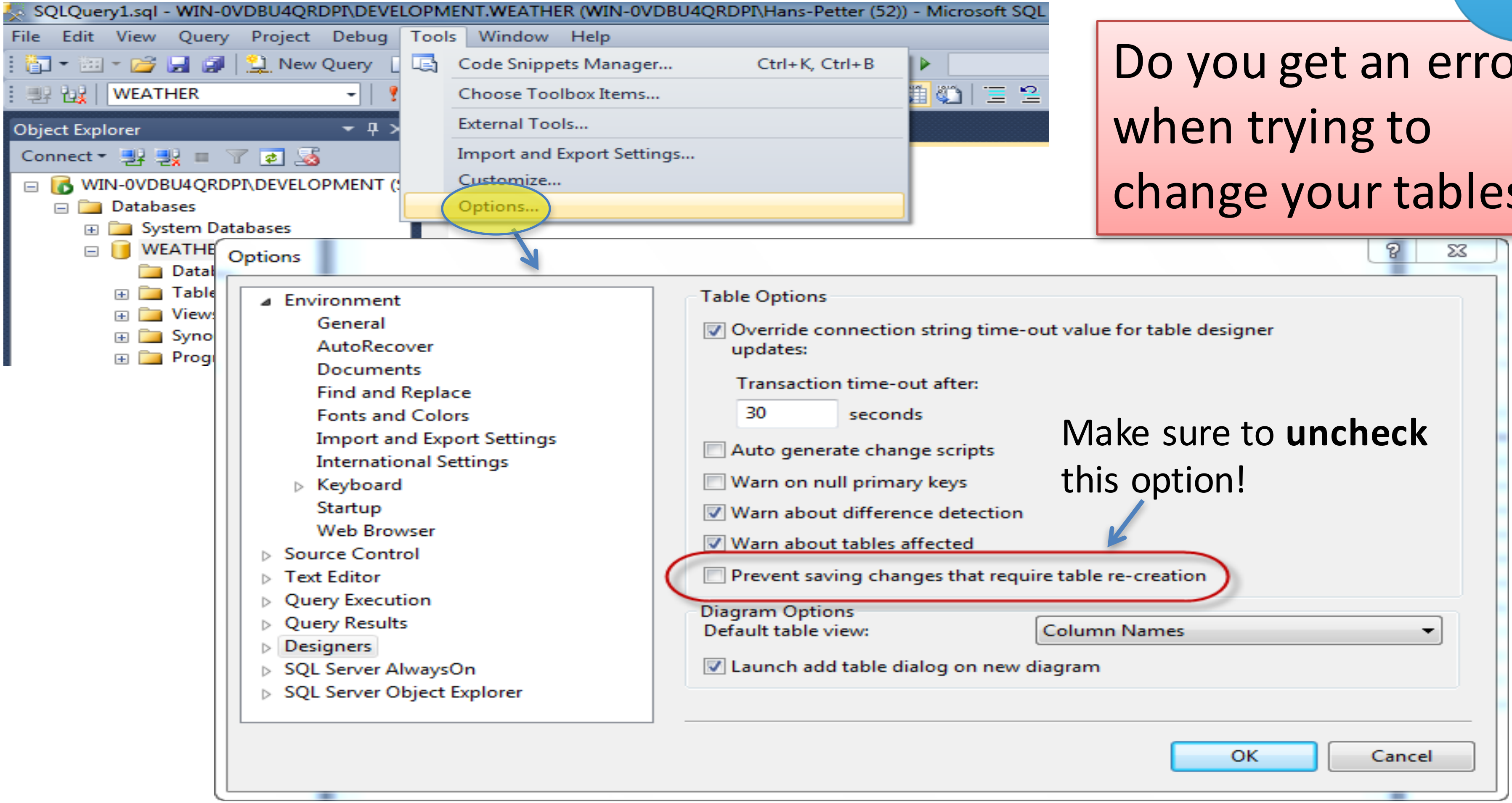


Table Options

☑ Override connection string time-out value for table designer updates:

Transaction time-out after: 30 seconds

☐ Auto generate change scripts

☐ Warn on null primary keys

☑ Warn about difference detection

☑ Warn about tables affected

☐ Prevent saving changes that require table re-creation

Diagram Options

Default table view: Column Names

☑ Launch add table dialog on new diagram

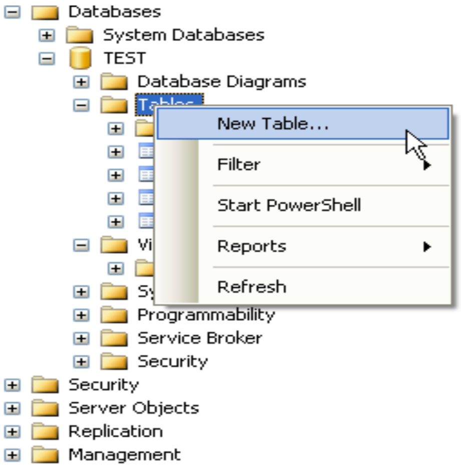
Make sure to **uncheck** this option!

Create Tables using the Designer Tools in SQL Server

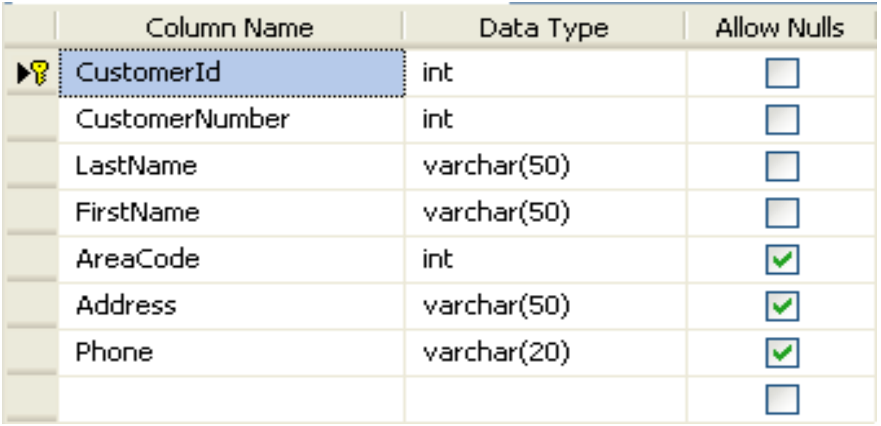
Even if you can do "everything" using the SQL language, it is sometimes easier to do something in the designer tools in the Management Studio in SQL Server.

Instead of creating a script you may as well easily use the designer for creating tables, constraints, inserting data, etc.

1 Select "New Table …":



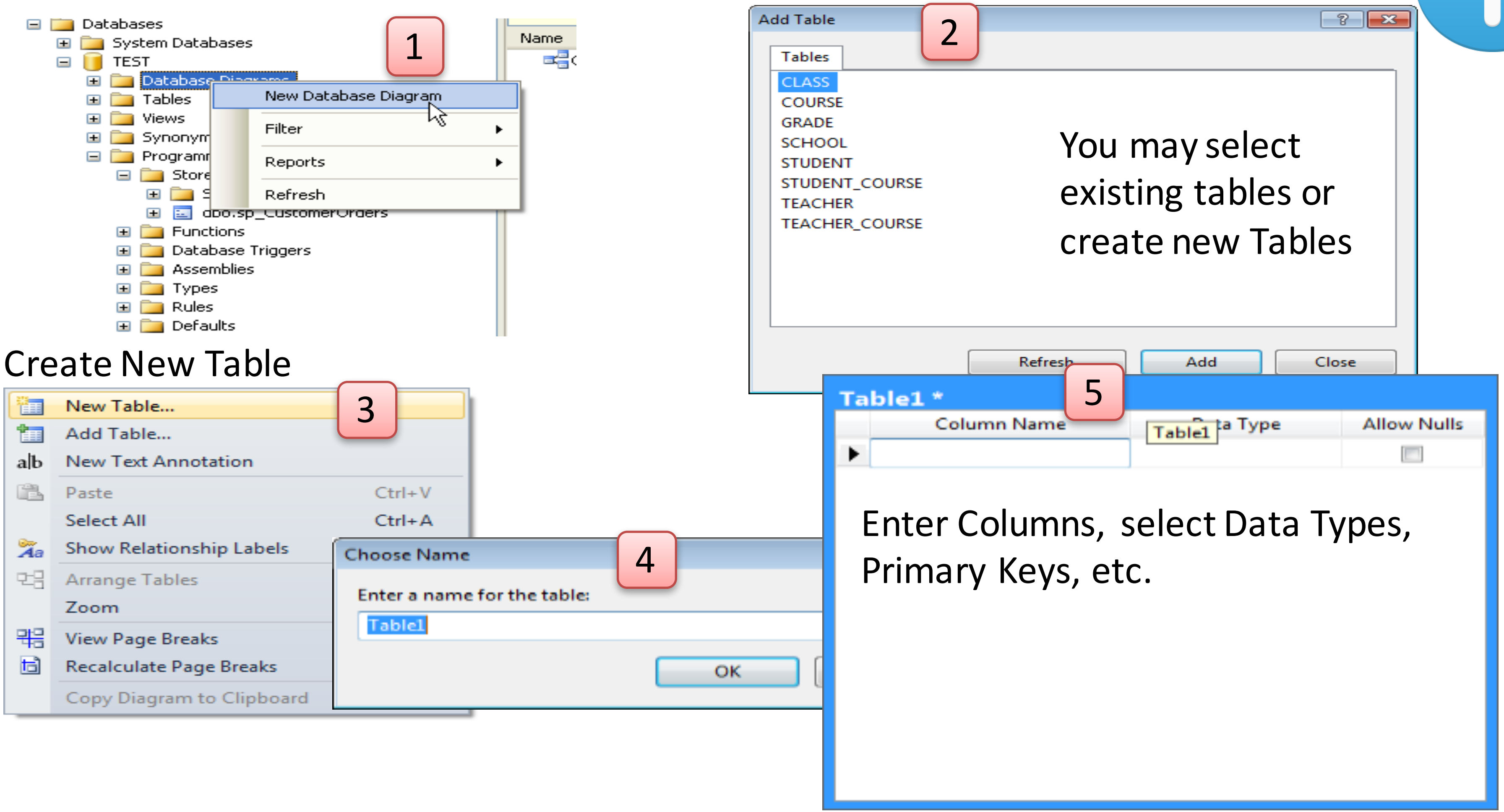
2 Next, the table designer pops up where you can add columns, data types, etc.



Column Name	Data Type	Allow Nulls
CustomerId	int	☐
CustomerNumber	int	☐
LastName	varchar(50)	☐
FirstName	varchar(50)	☐
AreaCode	int	☑
Address	varchar(50)	☑
Phone	varchar(20)	☑
		☐

In this designer we may also specify constraints, such as primary keys, unique, foreign keys, etc.

Create Tables with the “Database Diagram”



You may select existing tables or create new Tables

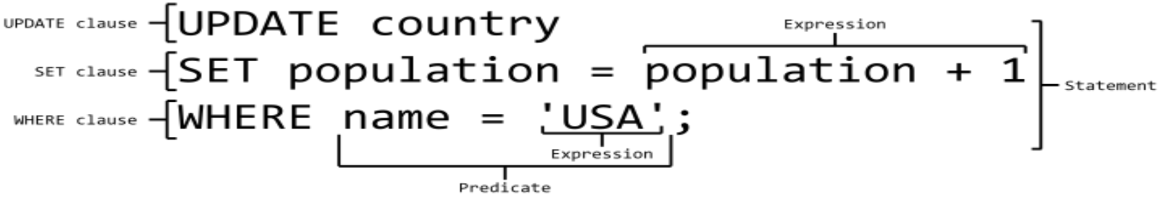
Create New Table

Enter Columns, select Data Types, Primary Keys, etc.

Congratulations! - You are finished with the Example

SQL

Structured Query Language



```
UPDATE country
SET population = population + 1
WHERE name = 'USA';
```

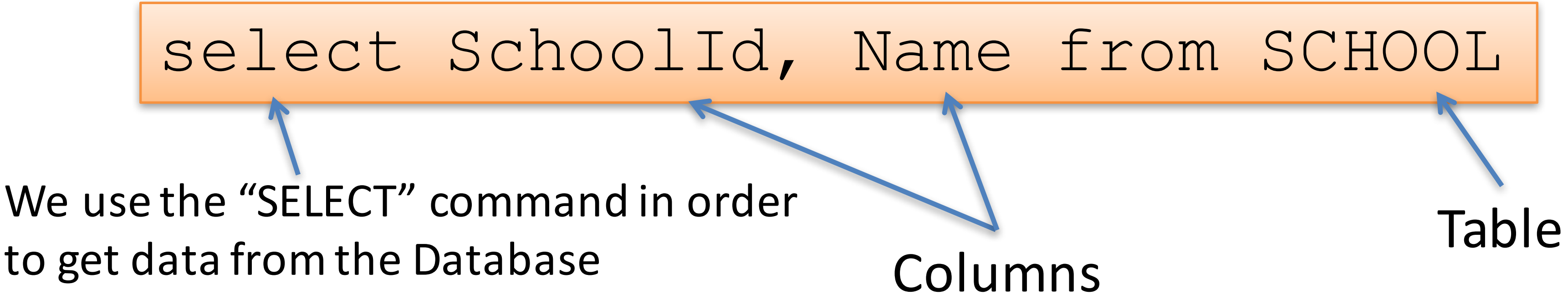
Hans-Petter Halvorsen, M.Sc.

What is SQL?

Theory

- SQL – Structured Query Language
- SQL is a standard language for accessing databases – and manipulate data
- SQL is not case sensitive

Example:



```
select SchoolId, Name from SCHOOL
```

We use the "SELECT" command in order to get data from the Database

Columns

Table

Theory

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**

Theory

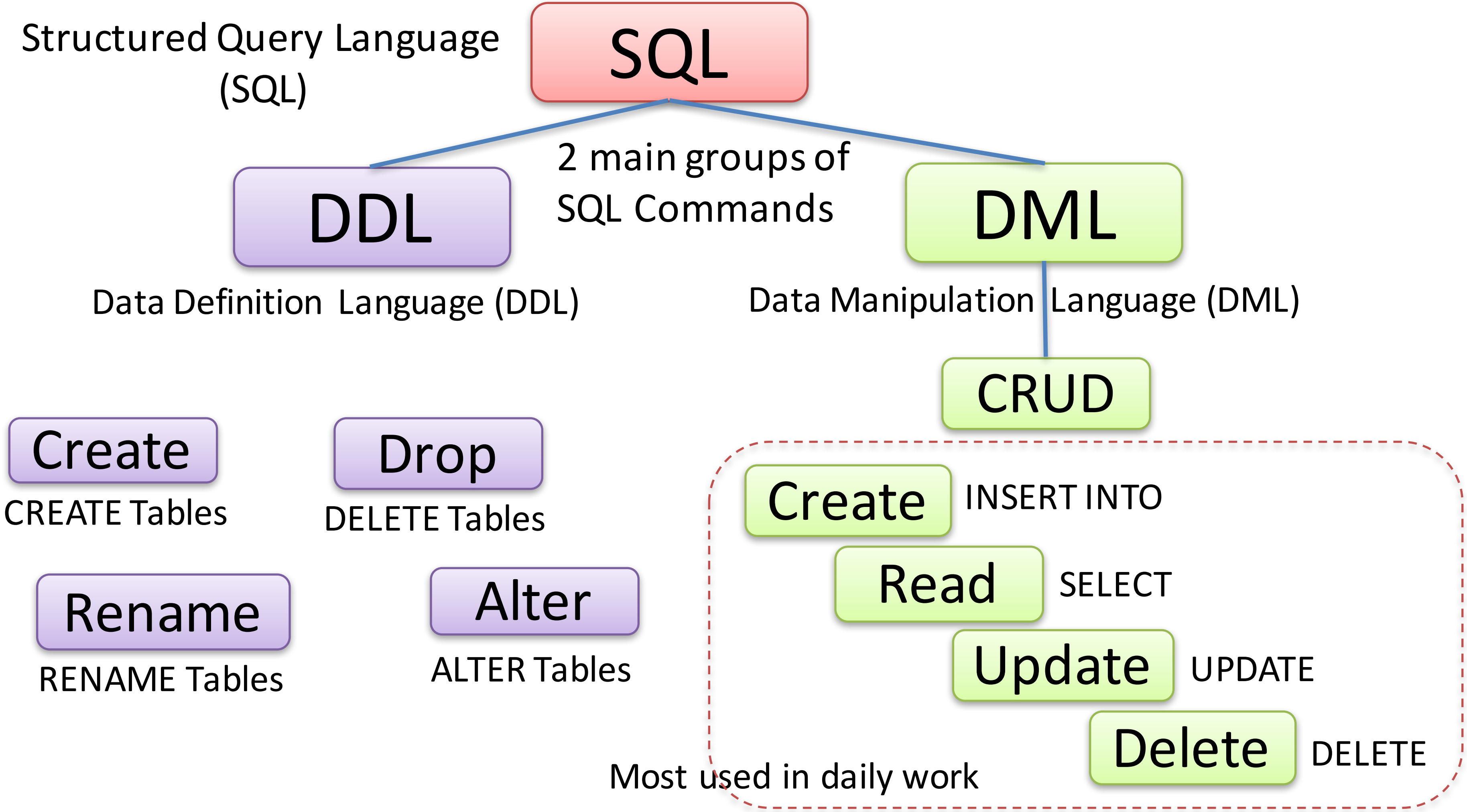
Important SQL Commands

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database

These are most used in your daily work. These commands are used to insert or modify data

- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

These are used when creating or modifying existing Tables



Structured Query Language (SQL)

SQL

2 main groups of SQL Commands

DDL

Data Definition Language (DDL)

DML

Data Manipulation Language (DML)

Create

CREATE Tables

Drop

DELETE Tables

Rename

RENAME Tables

Alter

ALTER Tables

CRUD

Create INSERT INTO

Read SELECT

Update UPDATE

Delete DELETE

Most used in daily work

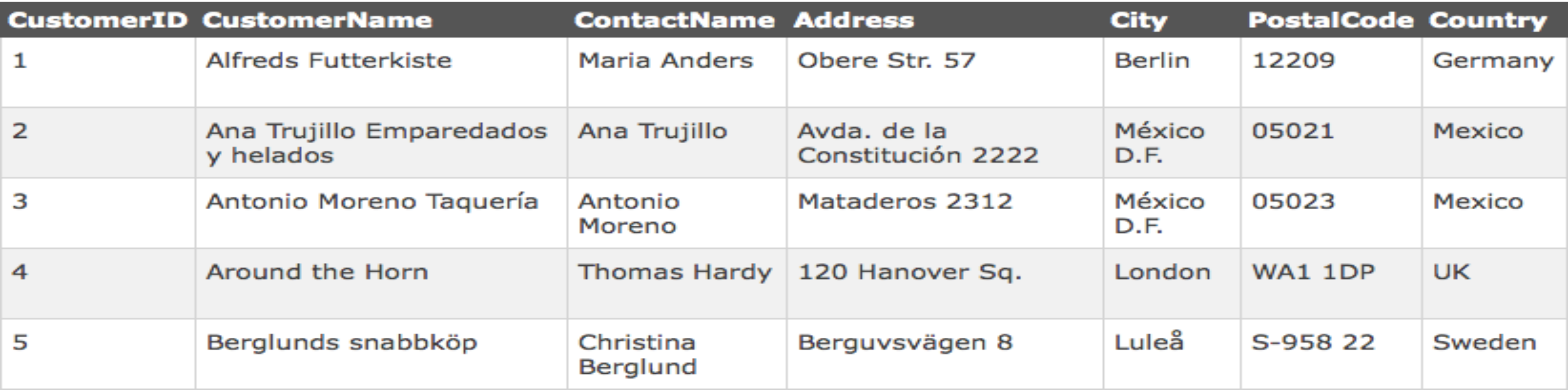
Create Tables using SQL

Example:

```
CREATE TABLE SCHOOL
(
        SchoolId int IDENTITY(1, 1) NOT NULL PRIMARY KEY,
        SchoolName varchar(50) NOT NULL UNIQUE,
        Description varchar(1000) NULL,
        Address varchar50) NULL,
        Phone varchar(50) NULL,
        PostCode varchar(50) NULL,
        PostAddress varchar(50) NULL,
)
GO
...
...
```

SQL Queries

Table Name: CUSTOMER



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Try to Create the following Table and Data using SQL

INSERT

Example

```
INSERT INTO CUSTOMER (CustomerName, ContactName, Address, City, PostalCode, Country)
VALUES ('Cardinal','Tom B. Erichsen','Skagen 21','Stavanger','4006','Norway');
```

SELECT

Try to Write and Execute the following Queries

```
SELECT * FROM CUSTOMER
```

Note! SQL is NOT case sensitive: “select” is the same as “SELECT”

```
SELECT CustomerName, City FROM CUSTOMER
```

```
SELECT DISTINCT City FROM CUSTOMER
```

```
SELECT * FROM CUSTOMER WHERE Country='Mexico'
```

```
SELECT * FROM CUSTOMER WHERE CustomerID=1
```

```
SELECT * FROM CUSTOMER WHERE Country='Germany’  AND City='Berlin'
```

```
SELECT * FROM CUSTOMER WHERE City='Berlin’ OR City=’Luleå'
```

```
SELECT * FROM CUSTOMER ORDER BY Country
```

```
SELECT * FROM CUSTOMER ORDER BY Country  DESC
```

UPDATE

Try to Write and Execute the following Query

```
UPDATE CUSTOMER
SET ContactName='Alfred Schmidt', City='Hamburg'
WHERE CustomerName='Alfreds Futterkiste'
```

Update Warning!

Be careful when updating records. What happens if we had omitted the WHERE clause, in the example above, like this:

~~UPDATE CUSTOMER~~
~~SET ContactName='Alfred Schmidt', City='Hamburg';~~

DELETE

Try to Write and Execute the following Query

```
DELETE FROM CUSTOMER
WHERE CustomerName='Alfreds Futterkiste'
AND ContactName='Maria Anders'
```

Note! It is possible to delete all rows in a table without deleting the table

```
DELETE * FROM CUSTOMER
```

Note! Be careful! All Data will be lost!!

Congratulations! - You are finished with the Example

https://www.youtube.com/watch?v=SHELF_iQUeU

Advanced SQL Features

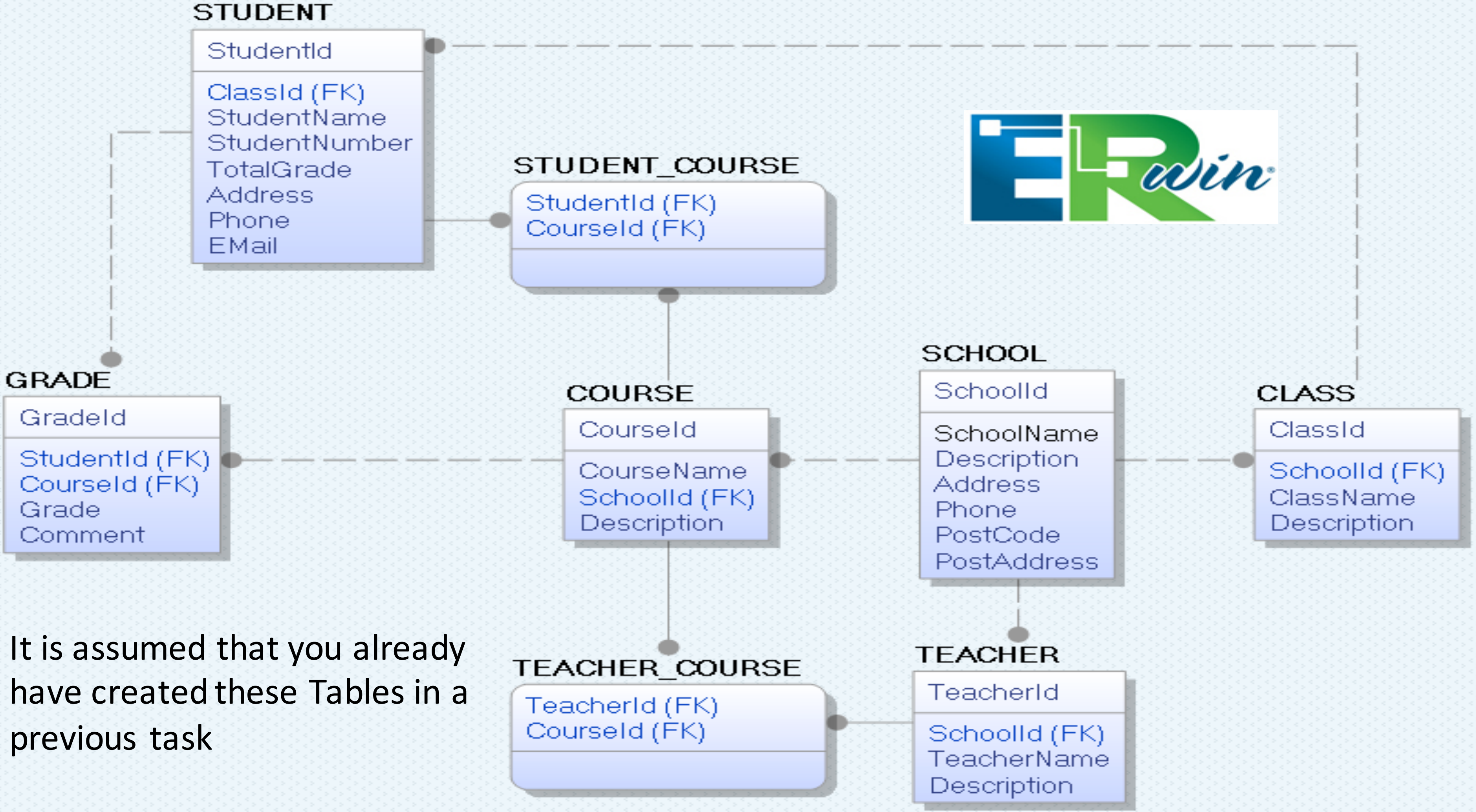
Views, Stored Procedures, Triggers and Functions

Hans-Petter Halvorsen, M.Sc.

Theory

Advanced SQL Features

- **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.
- **Functions**: With SQL and SQL Server you can use lots of built-in functions or you may create your own functions



STUDENT
StudentId
ClassId (FK)
StudentName
StudentNumber
TotalGrade
Address
Phone
EMail

STUDENT_COURSE
StudentId (FK)
CourseId (FK)

GRADE
GradeId
StudentId (FK)
CourseId (FK)
Grade
Comment

COURSE
CourseId
CourseName
SchoolId (FK)
Description

SCHOOL
SchoolId
SchoolName
Description
Address
Phone
PostCode
PostAddress

CLASS
ClassId
SchoolId (FK)
ClassName
Description

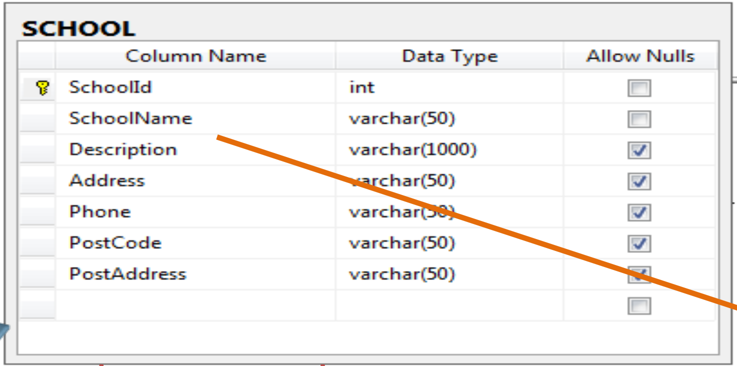
TEACHER_COURSE
TeacherId (FK)
CourseId (FK)

TEACHER
TeacherId
SchoolId (FK)
TeacherName
Description

It is assumed that you already have created these Tables in a previous task

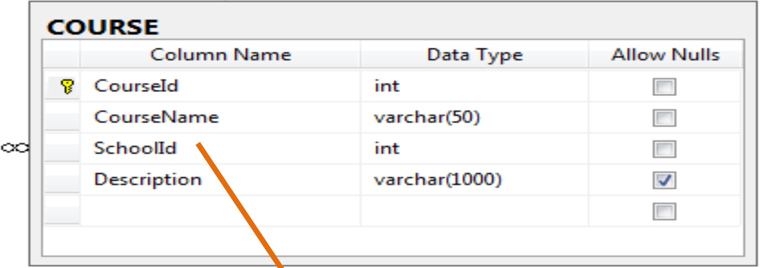
Get Data from <u>multiple</u> tables in a single Query using <u>Joins</u>

Example:



SCHOOL

	Column Name	Data Type	Allow Nulls
🔑	SchoolId	int	□
	SchoolName	varchar(50)	□
	Description	varchar(1000)	☑
	Address	varchar(50)	☑
	Phone	varchar(50)	☑
	PostCode	varchar(50)	☑
	PostAddress	varchar(50)	☑
			□

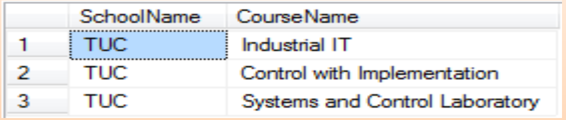


COURSE

	Column Name	Data Type	Allow Nulls
🔑	CourseId	int	□
	CourseName	varchar(50)	□
	SchoolId	int	□
	Description	varchar(1000)	☑
			□

Try this Example

```
select
SchoolName,
CourseName
from
SCHOOL
inner join COURSE on SCHOOL.SchoolId = COURSE.SchoolId
```



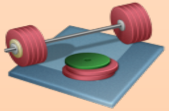
	SchoolName	CourseName
1	TUC	Industrial IT
2	TUC	Control with Implementation
3	TUC	Systems and Control Laboratory

You link Primary Keys and Foreign Keys together

Creating Views using SQL code

1 Create View:

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

CREATE VIEW CourseData
AS

SELECT
SCHOOL.SchoolId,
SCHOOL.SchoolName,
COURSE.CourseId,
COURSE.CourseName,
COURSE.Description

FROM
SCHOOL
INNER JOIN COURSE ON SCHOOL.SchoolId = COURSE.SchoolId
GO
```



Try to Create this View and make sure it works

A View is a “virtual” table that can contain data from <u>multiple</u> tables

This part is not necessary – but if you make any changes, you need to delete the old version before you can update it

The Name of the View

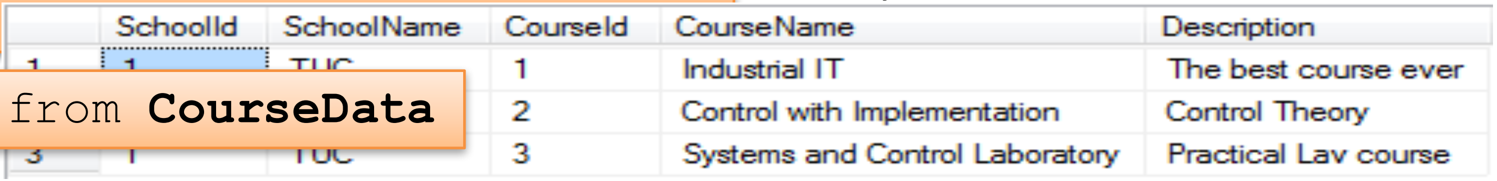
Inside the View you join the different tables together using the **JOIN** operator

You can Use the View as an ordinary table in Queries:

Using the View:

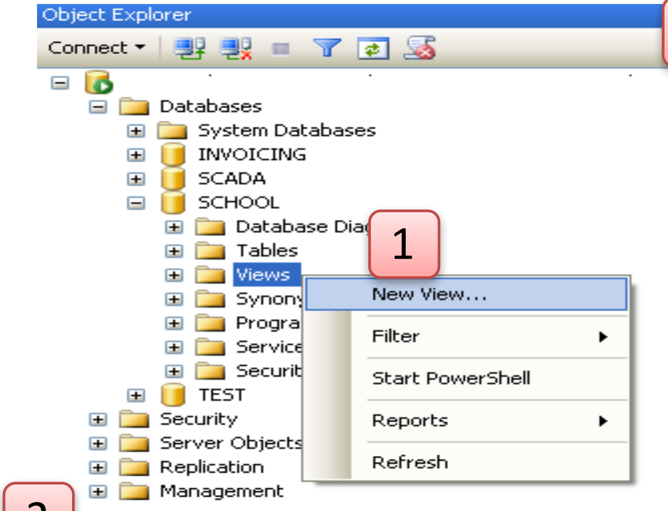
2

```
select * from CourseData
```



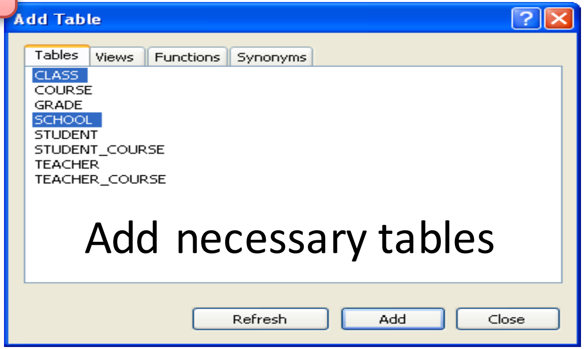
	SchoolId	SchoolName	CourseId	CourseName	Description
1	1	TUC	1	Industrial IT	The best course ever
			2	Control with Implementation	Control Theory
3	1	TUC	3	Systems and Control Laboratory	Practical Lav course

Creating Views using the Editor



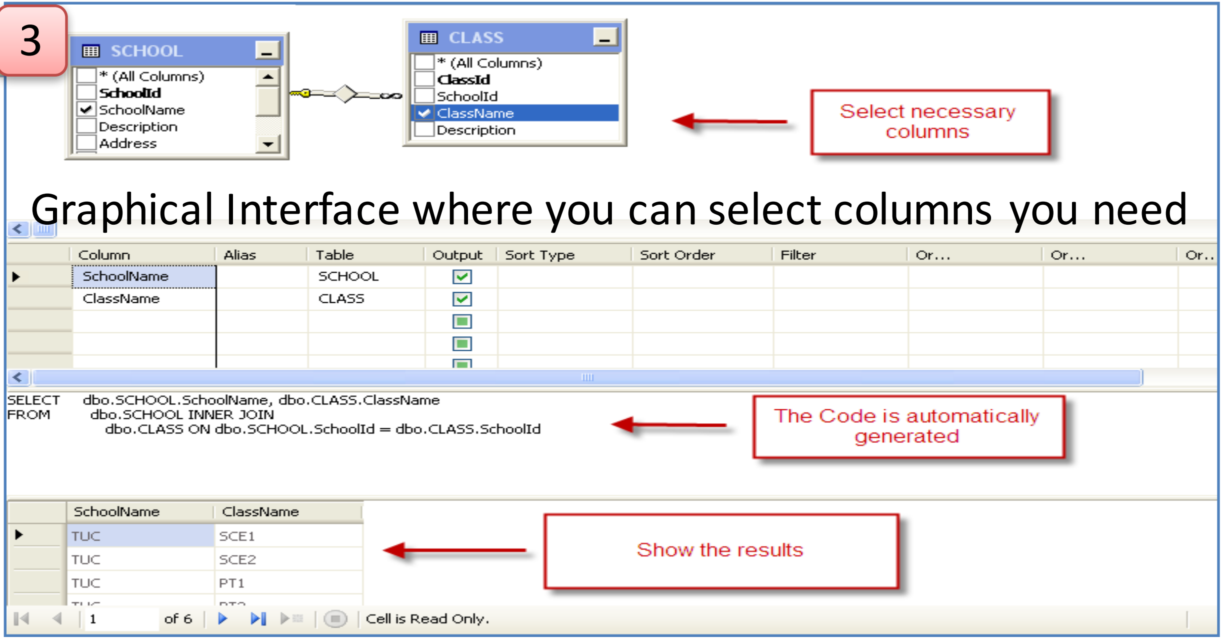
1

2



Add necessary tables

3



Select necessary columns

Graphical Interface where you can select columns you need

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...
SchoolName		SCHOOL	☑						
ClassName		CLASS	☑						

```
SELECT    dbo.SCHOOL.SchoolName, dbo.CLASS.ClassName
FROM        dbo.SCHOOL INNER JOIN
               dbo.CLASS ON dbo.SCHOOL.SchoolId = dbo.CLASS.SchoolId
```

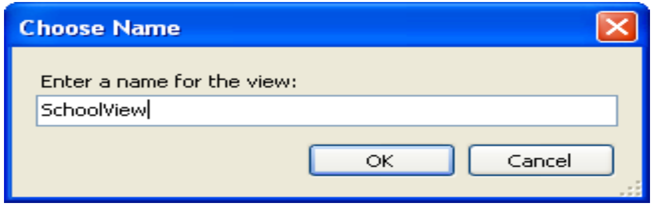
The Code is automatically generated

SchoolName	ClassName
TUC	SCE1
TUC	SCE2
TUC	PT1

1 of 6 Cell is Read Only.

Show the results

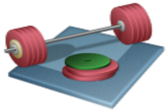
4



Enter a name for the view:

SchoolView

Save the View



Try this Example

Stored Procedure

1 Create Stored Procedure:

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

CREATE PROCEDURE StudentGrade
@Student varchar(50),
@Course varchar(10),
@Grade varchar(1)

AS

DECLARE
@StudentId int,
@CourseId int

select @StudentId = StudentId from STUDENT where StudentName = @Student

select @CourseId = CourseId from COURSE where CourseName = @Course

insert into GRADE (StudentId, CourseId, Grade)
values (@StudentId, @CourseId, @Grade)
GO
```

A Stored Procedure is like a Method in C# - it is a piece of code with SQL commands that do a specific task – and you reuse it

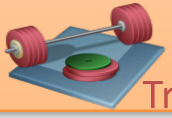
This part is not necessary – but if you make any changes, you need to delete the old version before you can update it

Procedure Name

Input Arguments

Internal/Local Variables
Note! Each variable starts with @

SQL Code (the “body” of the Stored Procedure)



Try to Create this Stored Procedure and make sure it works

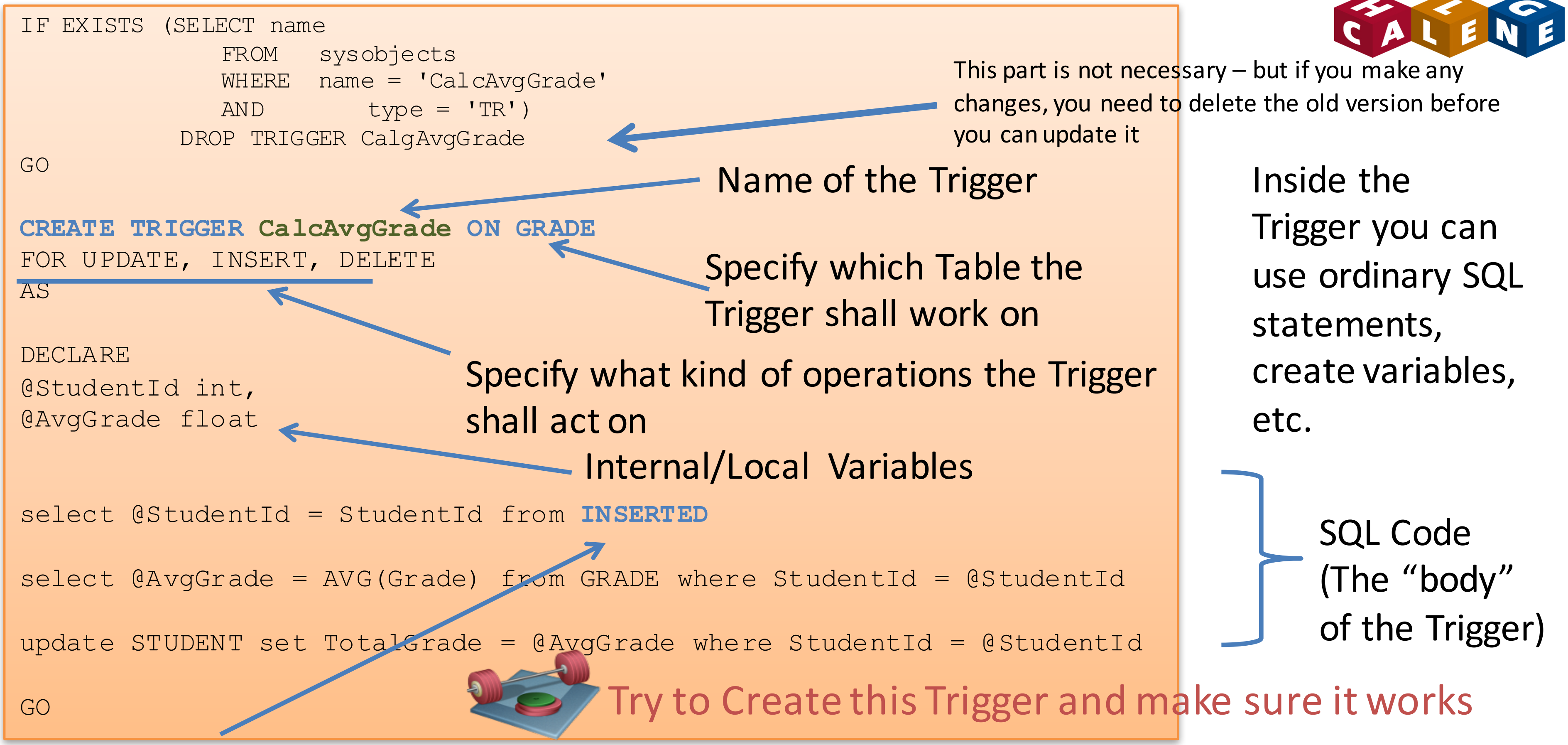
2 Using the Stored Procedure:

```
execute StudentGrade 'John Wayne', 'SCE2006', 'B'
```

Trigger

A Trigger is executed when you insert, update or delete data in a Table specified in the Trigger.

Create the Trigger:



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

CREATE TRIGGER CalcAvgGrade ON GRADE
FOR UPDATE, INSERT, DELETE
AS

DECLARE
@StudentId int,
@AvgGrade float

select @StudentId = StudentId from INSERTED

select @AvgGrade = AVG(Grade) from GRADE where StudentId = @StudentId

update STUDENT set TotalGrade = @AvgGrade where StudentId = @StudentId

GO
```

This part is not necessary – but if you make any changes, you need to delete the old version before you can update it

Name of the Trigger

Specify which Table the Trigger shall work on

Specify what kind of operations the Trigger shall act on

Internal/Local Variables

Inside the Trigger you can use ordinary SQL statements, create variables, etc.

SQL Code (The "body" of the Trigger)

Try to Create this Trigger and make sure it works

Note! "INSERTED" is a temporarily table containing the latest inserted data, and it is very handy to use inside a trigger

Quiz

Test your skills with this Multiple Choice Test

http://www.w3schools.com/quiztest/quiztest.asp?qtest=SQL

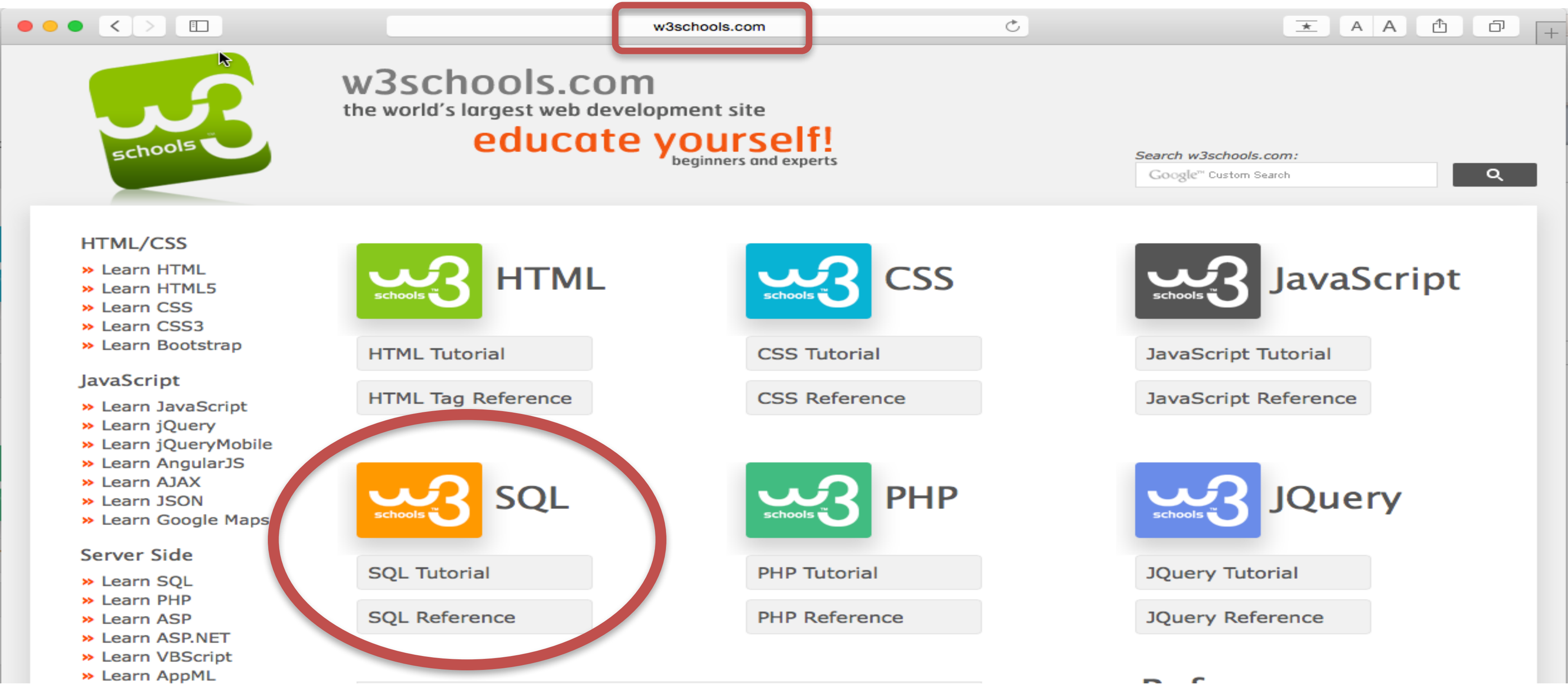
Result:

20 of 20

100%

Perfect!!!

Want to learn more SQL?



w3schools.com

the world's largest web development site

educate yourself!

beginners and experts

HTML/CSS

- » Learn HTML
- » Learn HTML5
- » Learn CSS
- » Learn CSS3
- » Learn Bootstrap

JavaScript

- » Learn JavaScript
- » Learn jQuery
- » Learn jQueryMobile
- » Learn AngularJS
- » Learn AJAX
- » Learn JSON
- » Learn Google Maps

Server Side

- » Learn SQL
- » Learn PHP
- » Learn ASP
- » Learn ASP.NET
- » Learn VBScript
- » Learn AppML

HTML

HTML Tutorial

HTML Tag Reference

CSS

CSS Tutorial

CSS Reference

JavaScript

JavaScript Tutorial

JavaScript Reference

SQL

SQL Tutorial

SQL Reference

PHP

PHP Tutorial

PHP Reference

JQuery

JQuery Tutorial

JQuery Reference

Congratulations! - You are finished with the Example

https://www.youtube.com/watch?v=q0_2zPtBbeE

LabVIEW

Database Communication in LabVIEW

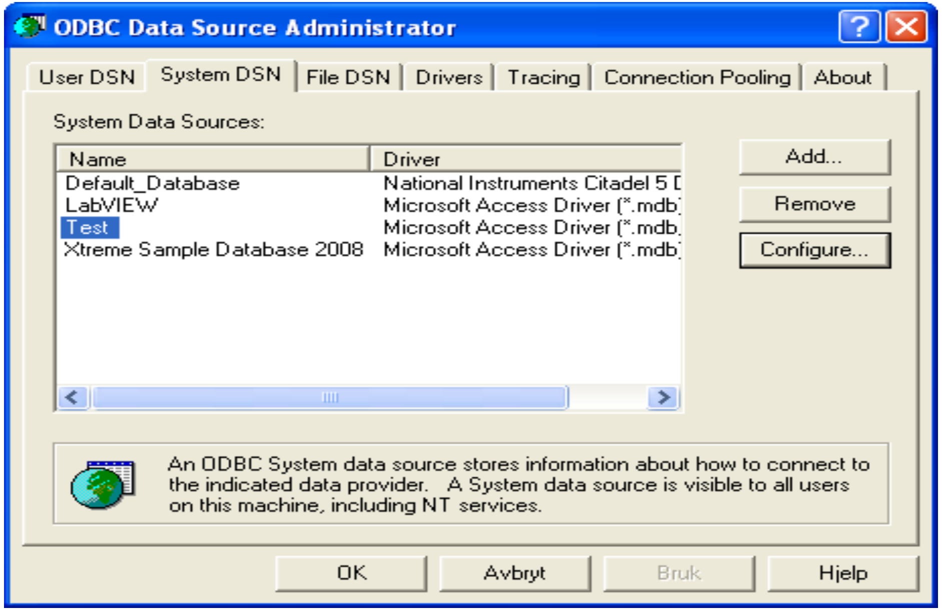
Hans-Petter Halvorsen, M.Sc.

ODBC

Theory

ODBC (Open Database Connectivity) is a standardized interface (API) for accessing the database from a client. You can use this standard to communicate with databases from different vendors, such as Oracle, SQL Server, etc. The designers of ODBC aimed to make it independent of programming languages, database systems, and operating systems.

Control Panel → Administrative Tools → Data Sources (ODBC)



System Data Sources:

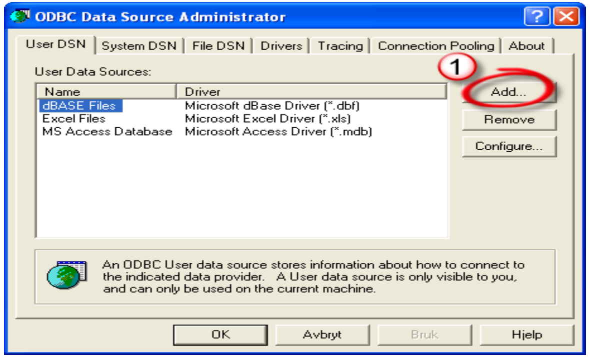
Name	Driver
Default_Database	National Instruments Citadel 5 D
LabVIEW	Microsoft Access Driver (*.mdb)
Test	Microsoft Access Driver (*.mdb
Xtreme Sample Database 2008	Microsoft Access Driver (*.mdb)

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

We will use this ODBC Connection later in LabVIEW in order to open the Database Connection from LabVIEW

Note! Make sure to use the **32 bit** version of the ODBC Tool!

ODBC – Step by Step Instructions



ODBC Data Source Administrator

User DSN | System DSN | File DSN | Drivers | Tracing | Connection Pooling | About

User Data Sources:

Name	Driver
dBASE Files	Microsoft dBase Driver (*.dbf)
Excel Files	Microsoft Excel Driver (*.xls)
MS Access Database	Microsoft Access Driver (*.mdb)

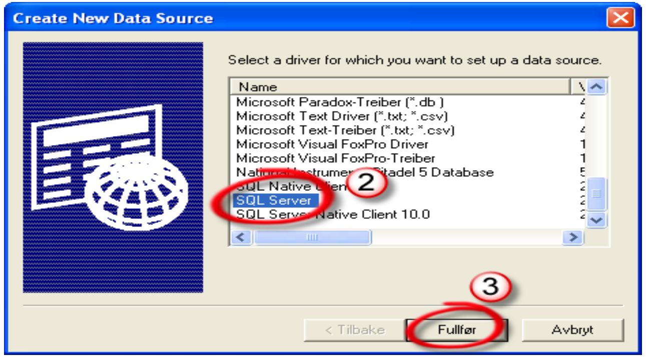
1 Add...

Remove

Configure...

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

OK | Avbryt | Bruk | Hjelp

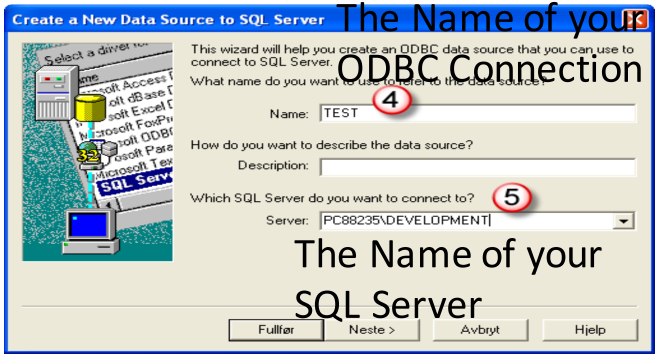


Create New Data Source

Select a driver for which you want to set up a data source.

Name
Microsoft Paradox-Treiber (*.db)
Microsoft Text Driver (*.txt; *.csv)
Microsoft Text-Treiber (*.txt; *.csv)
Microsoft Visual FoxPro Driver
Microsoft Visual FoxPro-Treiber
National Instruments Citadel 5 Database
SQL Native Client
2 SQL Server
SQL Server Native Client 10.0

< Tilbake | 3 Fullfør | Avbryt



Create a New Data Source to SQL Server

This wizard will help you create an ODBC data source that you can use to connect to SQL Server.

What name do you want to use to refer to the data source?

4 Name: TEST

How do you want to describe the data source?

Description:

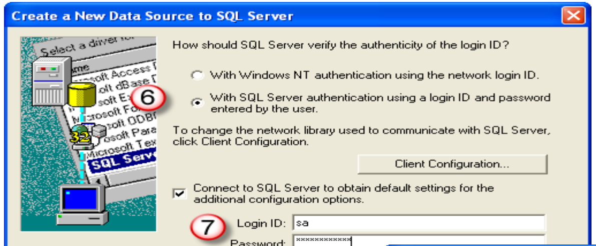
Which SQL Server do you want to connect to? 5

Server: PC88235\DEVELOPMENT

Fullfør | Neste > | Avbryt | Hjelp

The Name of your ODBC Connection

The Name of your SQL Server



Create a New Data Source to SQL Server

How should SQL Server verify the authenticity of the login ID?

○ With Windows NT authentication using the network login ID.

6 ● With SQL Server authentication using a login ID and password entered by the user.

To change the network library used to communicate with SQL Server, click Client Configuration.

Client Configuration...

☑ Connect to SQL Server to obtain default settings for the additional configuration options.

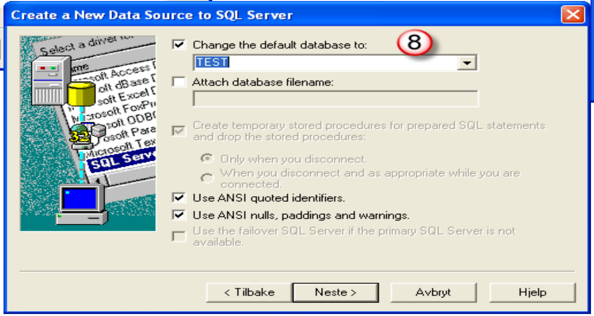
7 Login ID: sa

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

< Tilbake | Neste >

Use either Windows or SQL Server authentication (Windows is simplest to use!)

Select the Database you are using



Create a New Data Source to SQL Server

☑ Change the default database to: 8

TEST

□ Attach database filename:

☑ Create temporary stored procedures for prepared SQL statements and drop the stored procedures:

● Only when you disconnect.

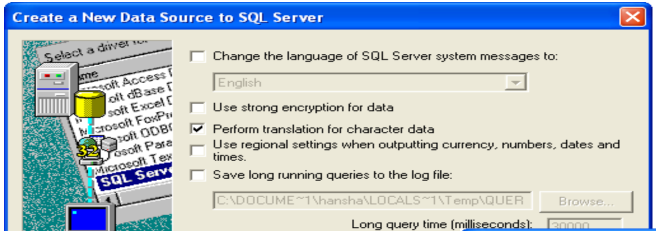
○ When you disconnect and as appropriate while you are connected.

☑ Use ANSI quoted identifiers.

☑ Use ANSI nulls, paddings and warnings.

□ Use the failover SQL Server if the primary SQL Server is not available.

< Tilbake | Neste > | Avbryt | Hjelp



Create a New Data Source to SQL Server

□ Change the language of SQL Server system messages to:

English

□ Use strong encryption for data

☑ Perform translation for character data

□ Use regional settings when outputting currency, numbers, dates and times.

□ Save long running queries to the log file:

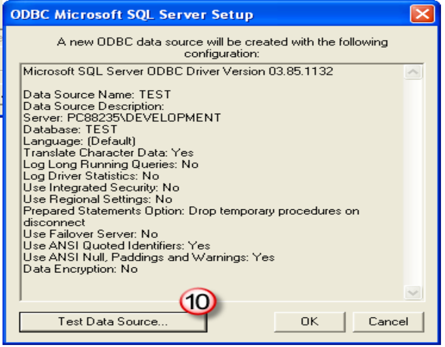
C:\DOCUME~1\hansha\LOCALS~1\Temp\QUER | Browse...

Long query time (milliseconds): 30000

□ Log ODBC driver statistics to the log file:

C:\DOCUME~1\hansha\LOCALS~1\Te

< Tilbake | 9 Fullfør



ODBC Microsoft SQL Server Setup

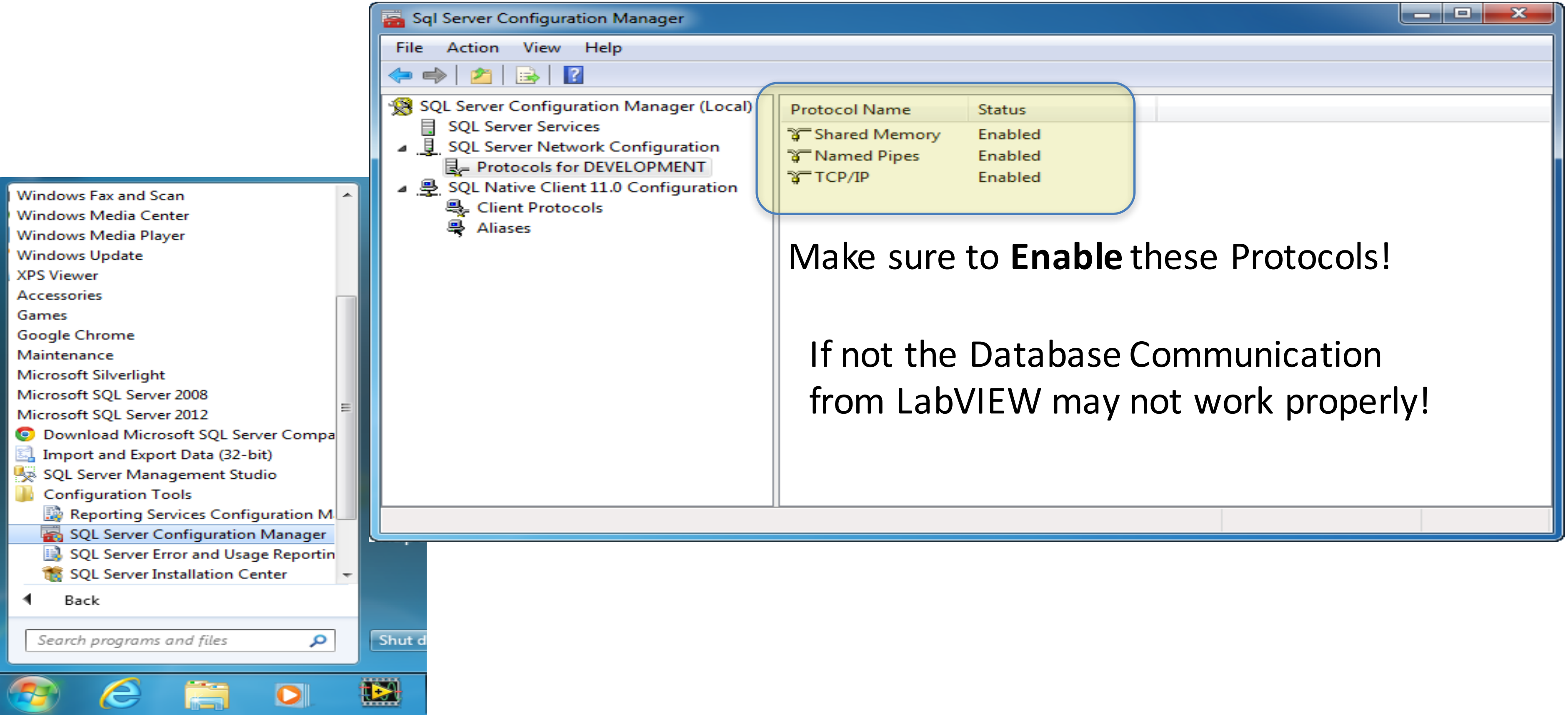
A new ODBC data source will be created with the following configuration:

```
Microsoft SQL Server ODBC Driver Version 03.85.1132

Data Source Name: TEST
Data Source Description:
Server: PC88235\DEVELOPMENT
Database: TEST
Language: (Default)
Translate Character Data: Yes
Log Long Running Queries: No
Log Driver Statistics: No
Use Integrated Security: No
Use Regional Settings: No
Prepared Statements Option: Drop temporary procedures on
disconnect
Use Failover Server: No
Use ANSI Quoted Identifiers: Yes
Use ANSI Null, Paddings and Warnings: Yes
Data Encryption: No
```

10 Test Data Source... | OK | Cancel

Test your connection to see if its works

SQL Server Configuration Manager


Sql Server Configuration Manager
File
Action
View
Help
SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration
Protocols for DEVELOPMENT
SQL Native Client 11.0 Configuration
Client Protocols
Aliases
Windows Fax and Scan
Windows Media Center
Windows Media Player
Windows Update
XPS Viewer
Accessories
Games
Google Chrome
Maintenance
Microsoft Silverlight
Microsoft SQL Server 2008
Microsoft SQL Server 2012
Download Microsoft SQL Server Compa
Import and Export Data (32-bit)
SQL Server Management Studio
Configuration Tools
Reporting Services Configuration M
SQL Server Configuration Manager
SQL Server Error and Usage Reportin
SQL Server Installation Center
Back
Search programs and files
Shut d

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

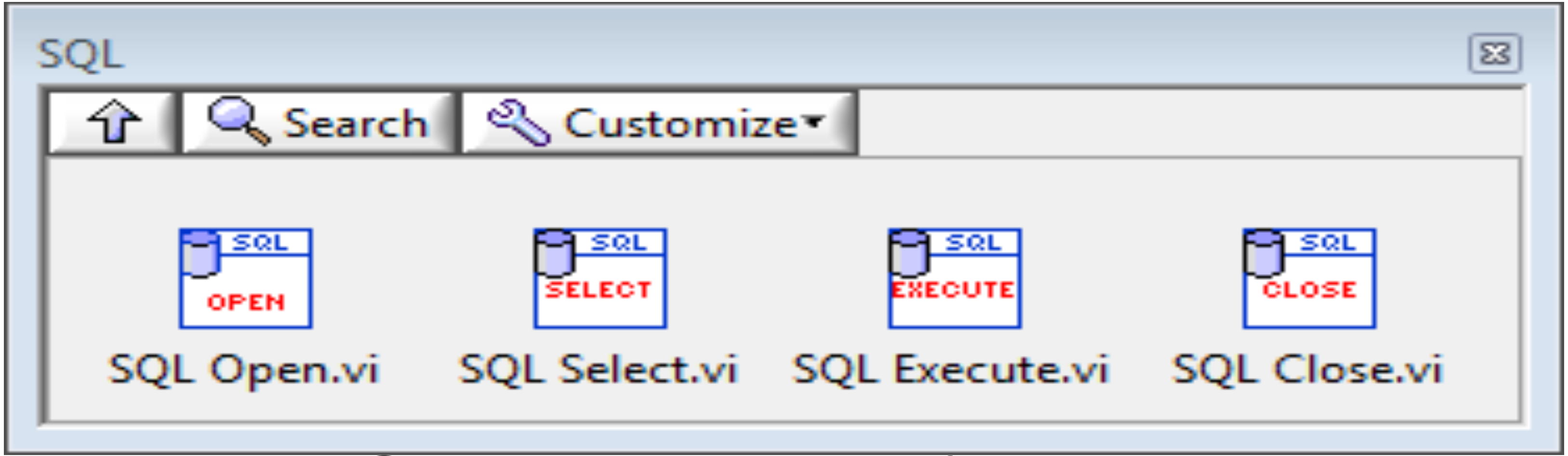
Make sure to **Enable** these Protocols!

If not the Database Communication from LabVIEW may not work properly!

LabVIEW SQL Toolkit

Software

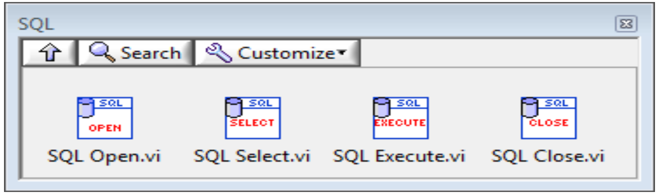
For Easy Database Communication with LabVIEW



© Hans-Petter Halvorsen

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

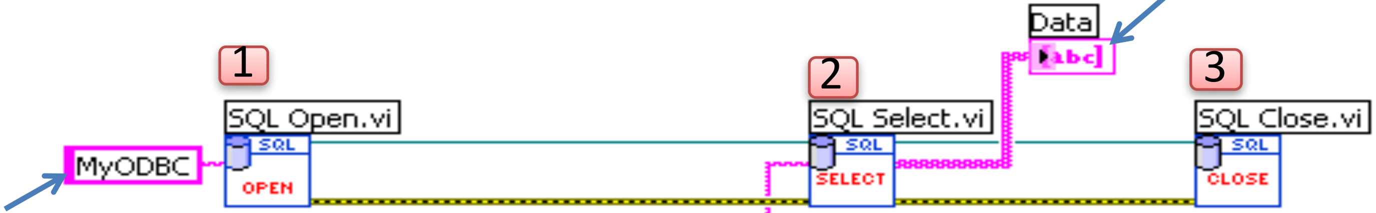
LabVIEW SQL Toolkit



Easy Access to Database Systems from LabVIEW

Example 1: Get Data from Database into LabVIEW:

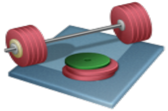
2D Table with Data



Your ODBC Connection

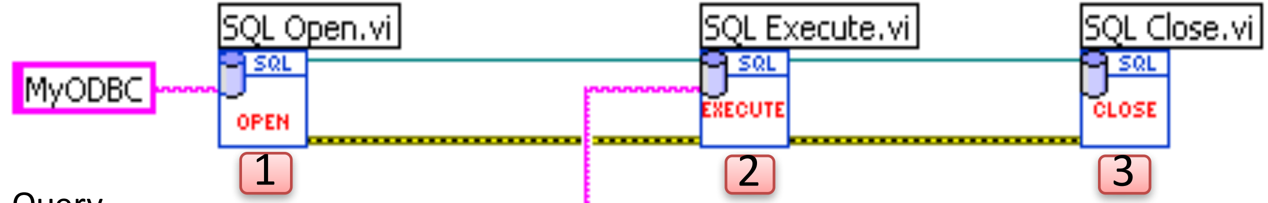
Query

```
select * from SCHOOL order by Name
```



Try these Examples

Example 2: Write Data to Database from LabVIEW:



Query

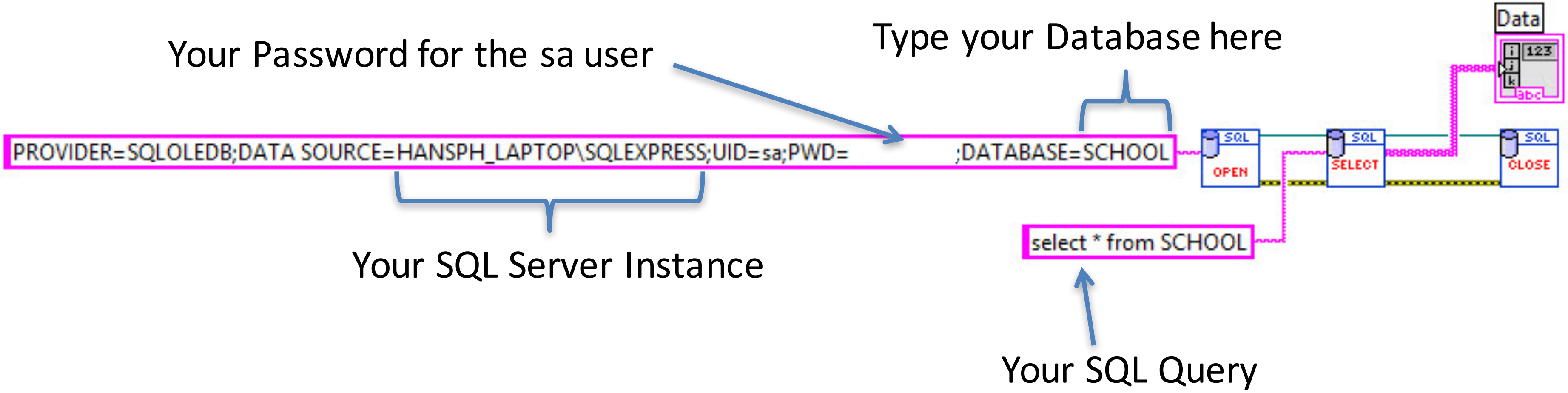
```
insert into SCHOOL (Name, Description) values ('TUC', 'The Best School in the World')
```

LabVIEW SQL Toolkit

Easy Access to Database Systems from LabVIEW

Example

Alternative Solution: Type in the **Connection String** for your Database



Your Password for the sa user

Type your Database here

```
PROVIDER=SQLOLEDB;DATA SOURCE=HANSPH_LAPTOP\SQLEXPRESS;UID=sa;PWD=                ;DATABASE=SCHOOL
```

Your SQL Server Instance

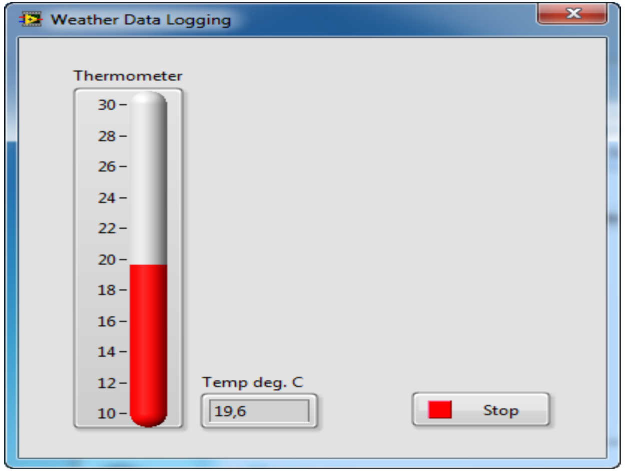
```
select * from SCHOOL
```

Your SQL Query

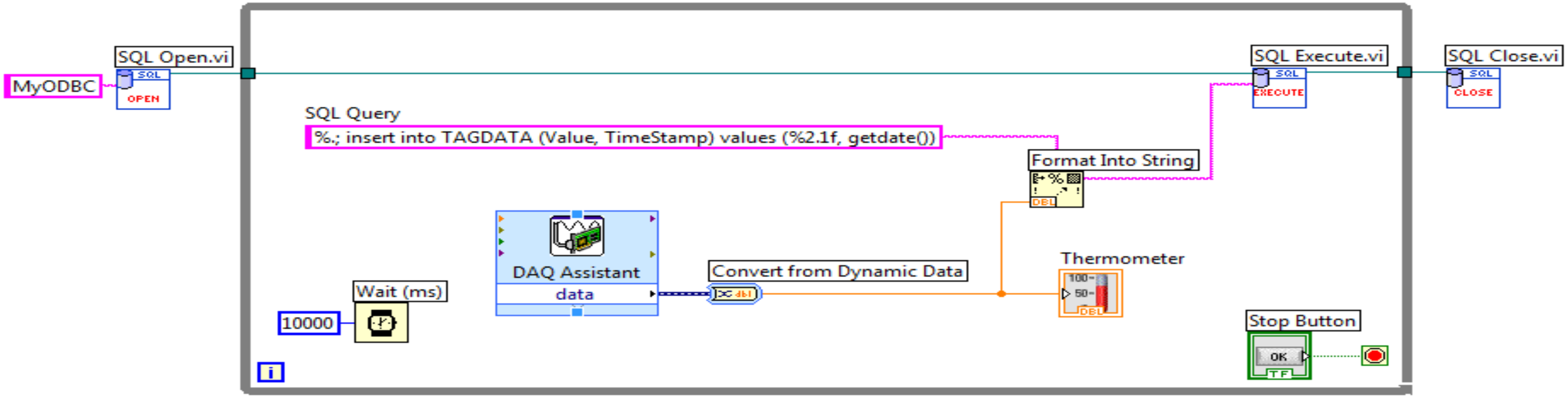
Note! When using this method, you dont need to create an ODBC Connection first!

Database Communication in LabVIEW

Example

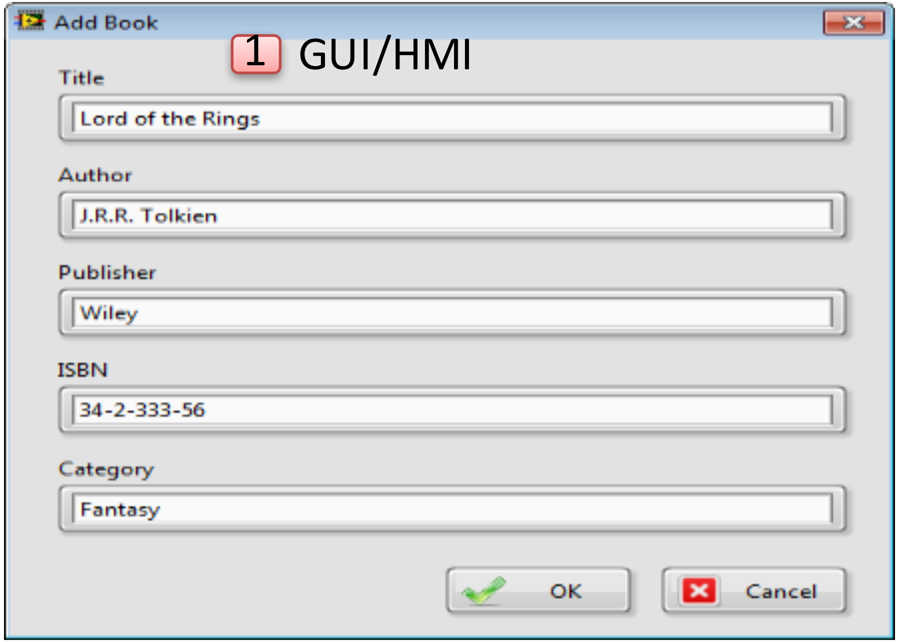


In this Example we use the NI TC-01 Thermocouple device in order to log Temperature data to a SQL Server Database from LabVIEW.



```
%.; insert into TAGDATA (Value, TimeStamp) values (%2.1f, getdate())
```

LabVIEW SQL Toolkit Example



1 GUI/HMI

Title: Lord of the Rings

Author: J.R.R. Tolkien

Publisher: Wiley

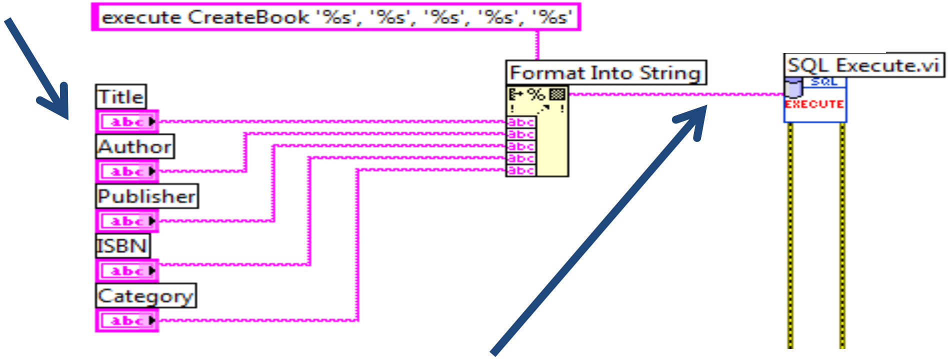
ISBN: 34-2-333-56

Category: Fantasy

If we want to save input data from the user we can use the “**Format Into String**” function

The **%s** operator will be replaced by the text from the TextBox on the Front Panel. For Numbers we can use **%d** (Integer) or **%f** for Floating-point Number.

2 Code:



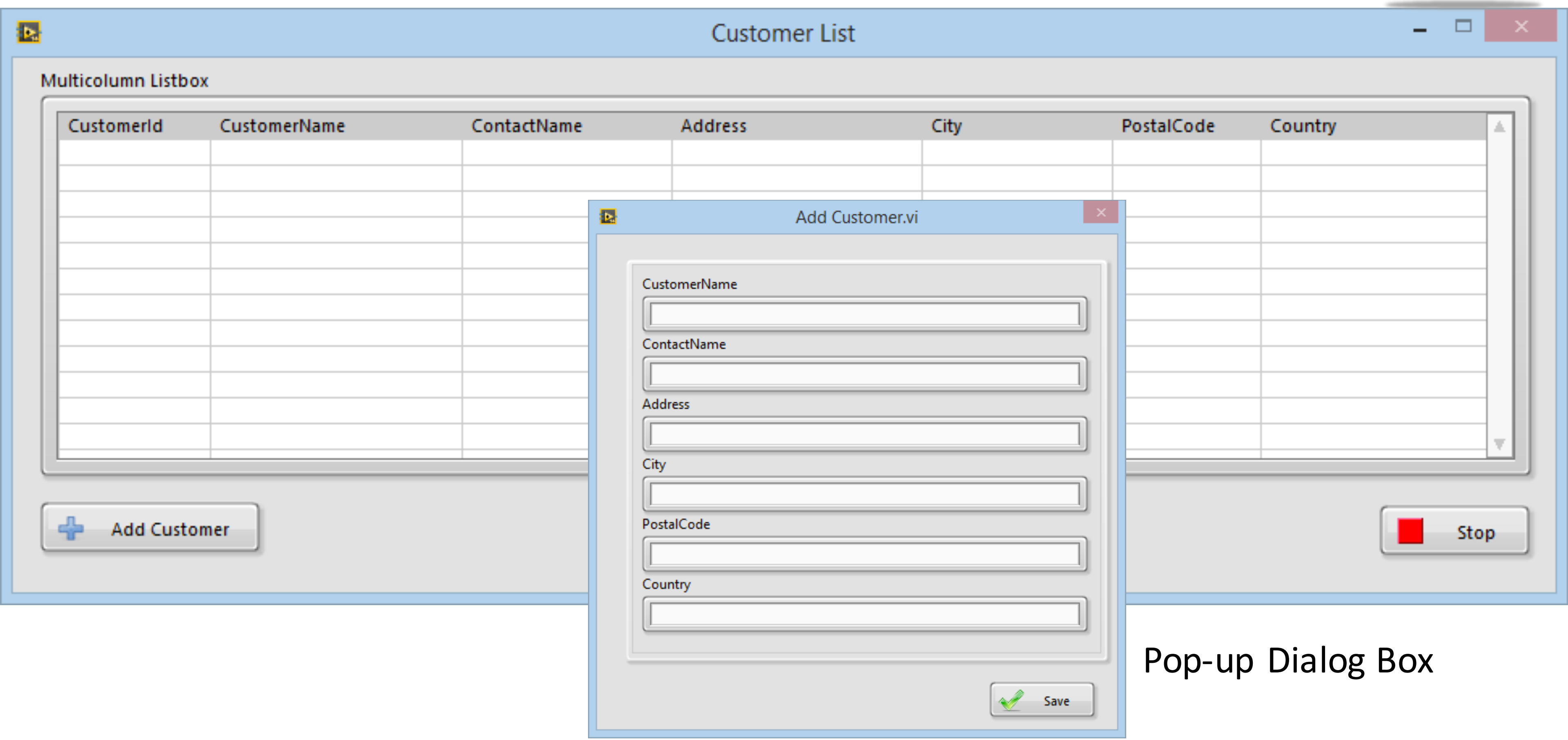
```
execute CreateBook '%s', '%s', '%s', '%s', '%s'
```

3 Resulting SQL Query:

execute **CreateBook** ‘Lord of the Rings', ‘J.R.R. Tolkien', Wiley', ‘32-2-333-56', Fantasy'

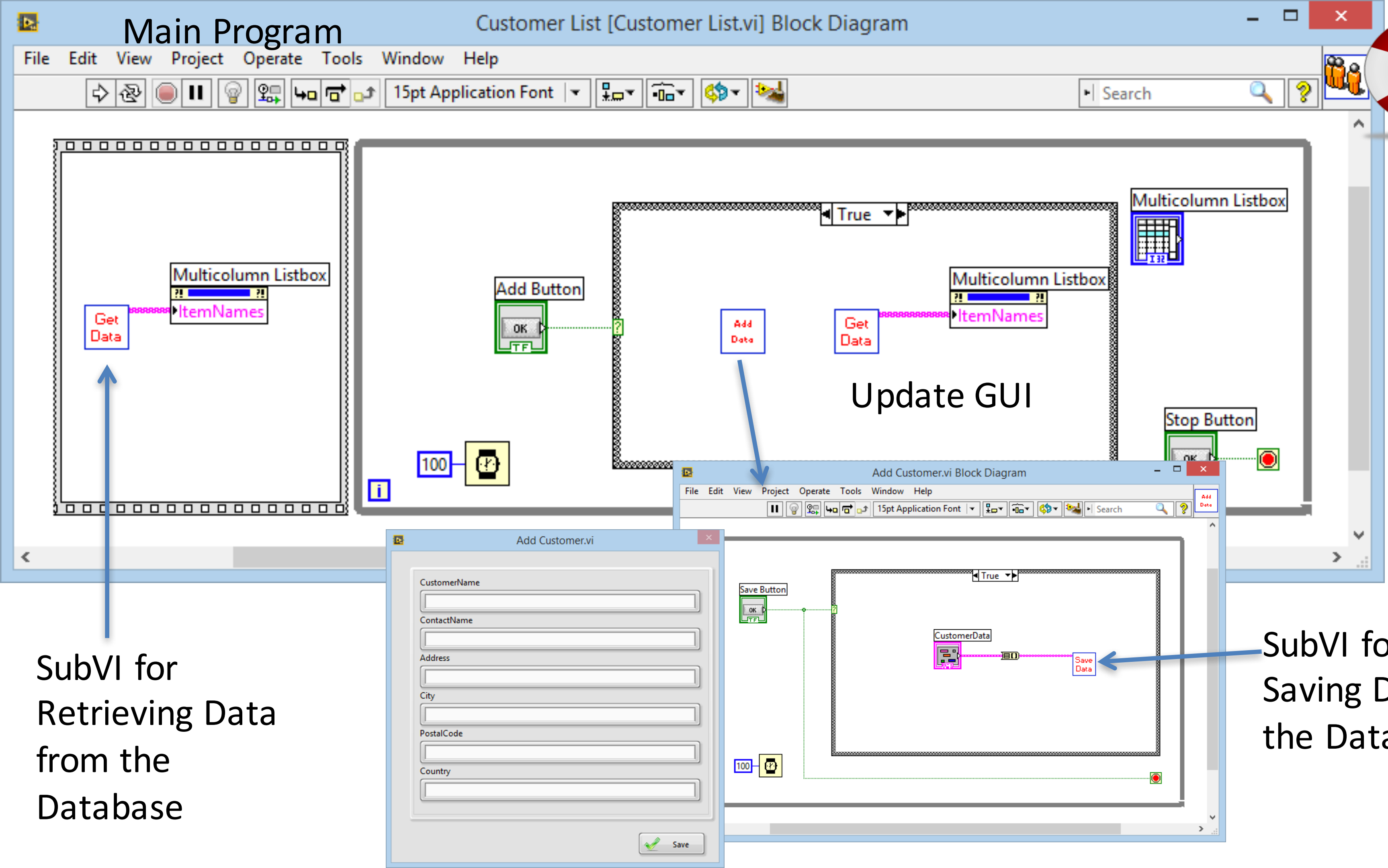
LabVIEW Example

Main Program



Pop-up Dialog Box

Main Program



Update GUI

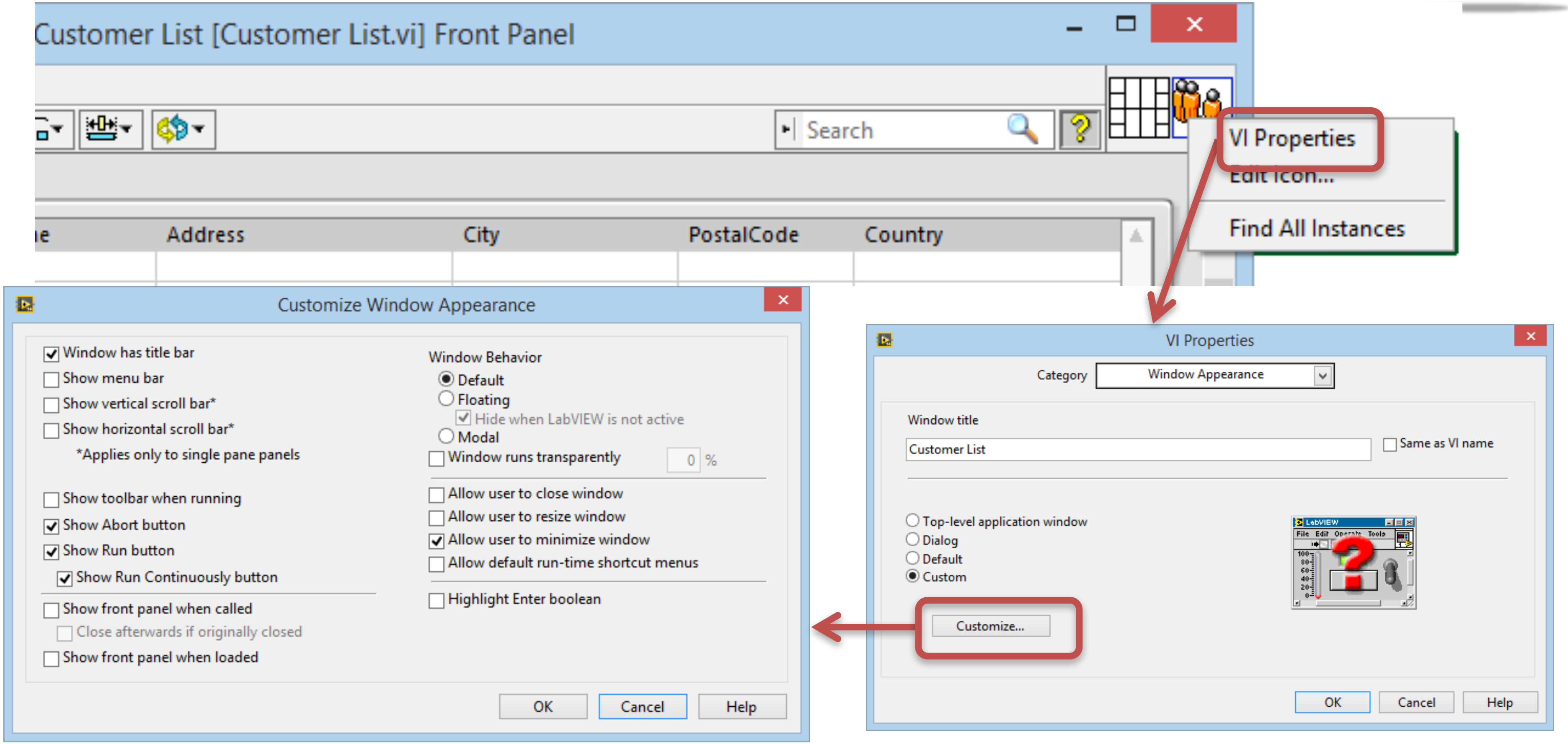
SubVI for Retrieving Data from the Database

SubVI for Saving Data to the Database

Example

VI Properties Settings

Example



VI Properties

Category: Window Appearance

Window title: Customer List

□ Same as VI name

○ Top-level application window
○ Dialog
○ Default
● Custom

Customize Window Appearance

☑ Window has title bar
□ Show menu bar
□ Show vertical scroll bar*
□ Show horizontal scroll bar*
*Applies only to single pane panels

□ Show toolbar when running
☑ Show Abort button
☑ Show Run button
☑ Show Run Continuously button

□ Show front panel when called
□ Close afterwards if originally closed
□ Show front panel when loaded

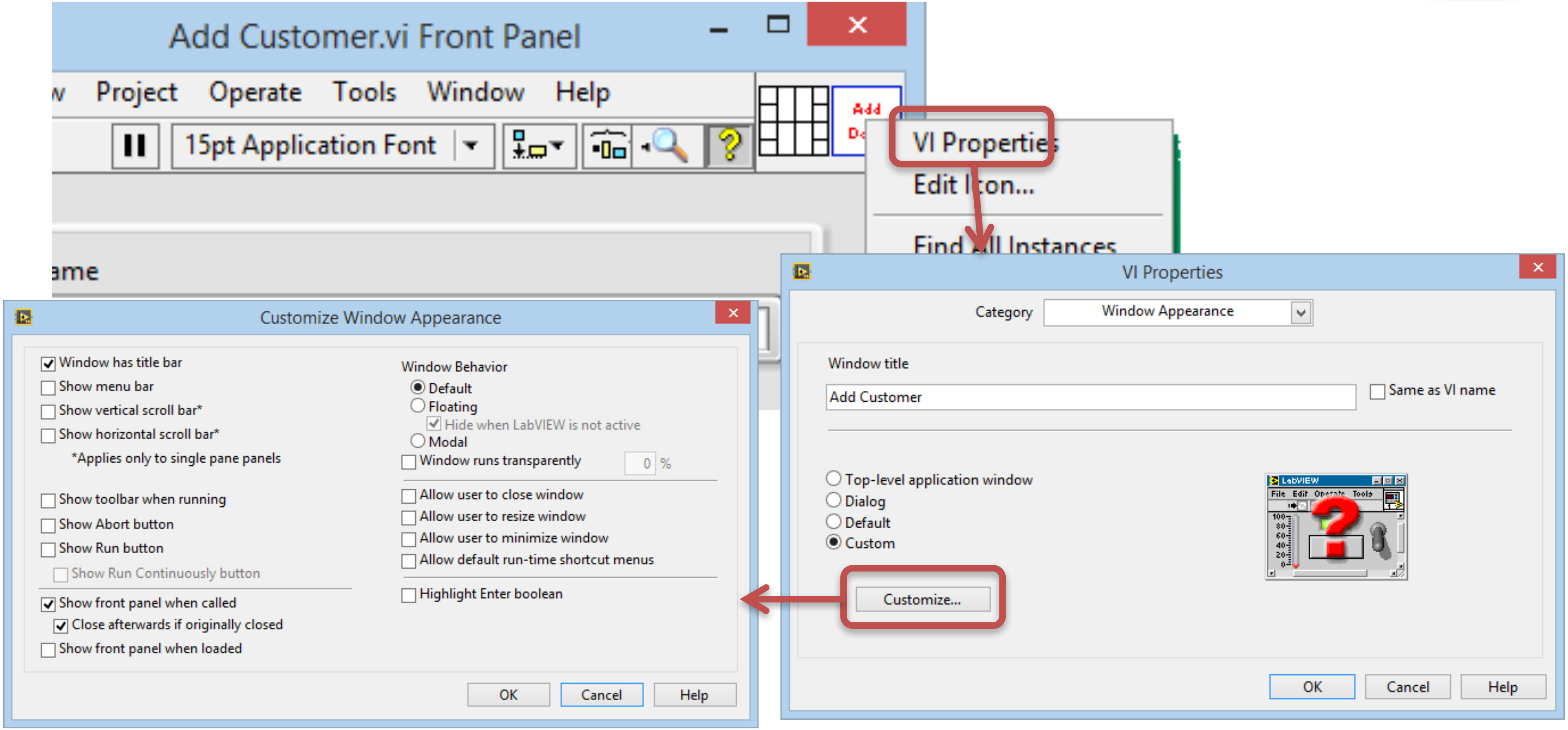
Window Behavior
● Default
○ Floating
☑ Hide when LabVIEW is not active
○ Modal
□ Window runs transparently 0 %

□ Allow user to close window
□ Allow user to resize window
☑ Allow user to minimize window
□ Allow default run-time shortcut menus

□ Highlight Enter boolean

VI Properties Settings

Example



VI Properties

Category: Window Appearance

Window title: Add Customer

□ Same as VI name

○ Top-level application window
○ Dialog
○ Default
● Custom

Customize...

OK Cancel Help

Customize Window Appearance

☑ Window has title bar
□ Show menu bar
□ Show vertical scroll bar*
□ Show horizontal scroll bar*
*Applies only to single pane panels

□ Show toolbar when running
□ Show Abort button
□ Show Run button
□ Show Run Continuously button

☑ Show front panel when called
☑ Close afterwards if originally closed
□ Show front panel when loaded

Window Behavior
● Default
○ Floating
☑ Hide when LabVIEW is not active
○ Modal
□ Window runs transparently 0 %

□ Allow user to close window
□ Allow user to resize window
□ Allow user to minimize window
□ Allow default run-time shortcut menus

□ Highlight Enter boolean

OK Cancel Help

Congratulations! - You are finished with the Example

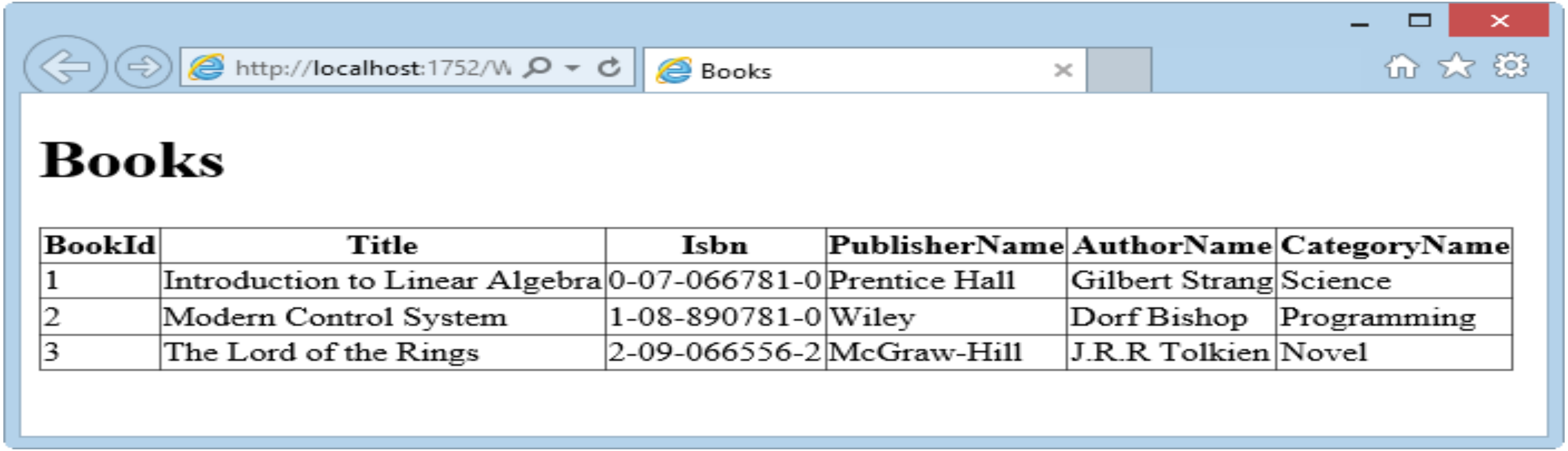
Visual Studio

Database Communication in Visual Studio/C#

Visual Studio

Hans-Petter Halvorsen, M.Sc.

ASP.NET WebForm App Example



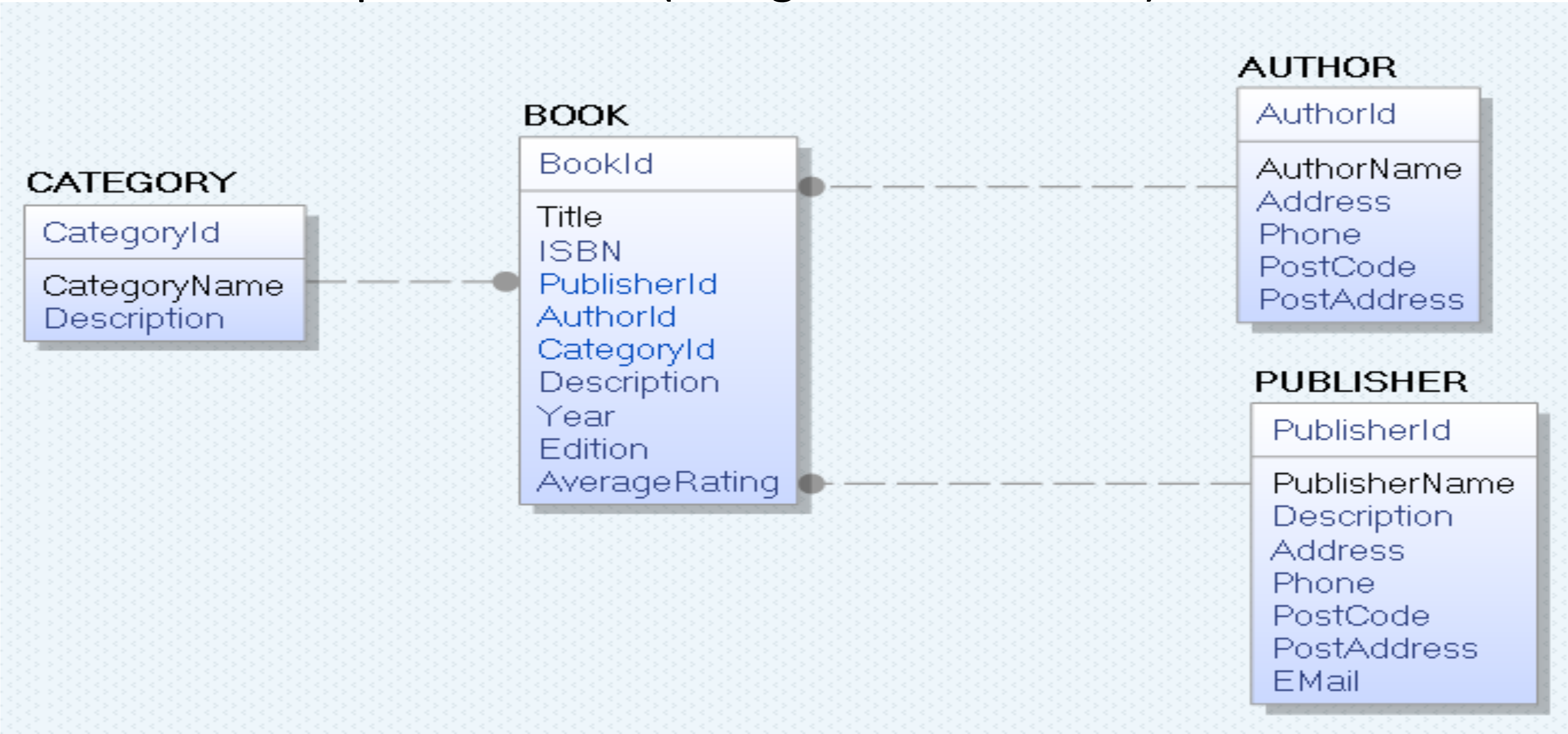
Books

BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbert Strang	Science
2	Modern Control System	1-08-890781-0	Wiley	Dorf Bishop	Programming
3	The Lord of the Rings	2-09-066556-2	McGraw-Hill	J.R.R Tolkien	Novel

ASP.NET is a Web Framework available from Visual Studio. Easily explained, it is just a “Template” for creating Web Pages using C#

Database

This is our Example Database (Designed with ERwin)



CATEGORY

CategoryId

CategoryName
Description

BOOK

BookId

Title
ISBN
PublisherId
AuthorId
CategoryId
Description
Year
Edition
AverageRating

AUTHOR

AuthorId

AuthorName
Address
Phone
PostCode
PostAddress

PUBLISHER

PublisherId

PublisherName
Description
Address
Phone
PostCode
PostAddress
EMail

SQL Script - Tables

```
if not exists (select * from dbo.sysobjects  where id = object_id(N'[AUTHOR]')  and OBJECTPROPERTY(id,  N'IsUserTable')  = 1)
CREATE  TABLE [AUTHOR]
(
                       [AuthorId]  [int] IDENTITY(1,  1) NOT NULL PRIMARY  KEY,
                       [AuthorName]  [varchar](50)  NOT NULL UNIQUE,
                       [Address]  [varchar](50)  NULL,
                       [Phone]  [varchar](50)  NULL,
                       [PostCode]  [varchar](50)  NULL,
                       [PostAddress]  [varchar](50)  NULL,
)
GO

if not exists (select * from dbo.sysobjects  where id = object_id(N'[PUBLISHER]')  and OBJECTPROPERTY(id,  N'IsUserTable')  = 1)
CREATE  TABLE [PUBLISHER]
(
                       [PublisherId]  [int] IDENTITY(1,  1) NOT NULL PRIMARY  KEY,
                       [PublisherName]  [varchar](50)  NOT NULL UNIQUE,
                       [Description]  [varchar](1000)  NULL,
                       [Address]  [varchar](50)  NULL,
                       [Phone]  [varchar](50)  NULL,
                       [PostCode]  [varchar](50)  NULL,
                       [PostAddress]  [varchar](50)  NULL,
                       [EMail]  [varchar](50)  NULL,
)
GO

if not exists (select * from dbo.sysobjects  where id = object_id(N'[CATEGORY]')  and OBJECTPROPERTY(id,  N'IsUserTable')  = 1)
CREATE  TABLE [CATEGORY]
(
                       [CategoryId]  [int] IDENTITY(1,  1) NOT NULL PRIMARY  KEY,
                       [CategoryName]  [varchar](50)  NOT NULL UNIQUE,
                       [Description]  [varchar](1000)  NULL,
)
GO

if not exists (select * from dbo.sysobjects  where id = object_id(N'[BOOK]')  and OBJECTPROPERTY(id,  N'IsUserTable')  = 1)
CREATE  TABLE [BOOK]
(
                       [BookId]  [int] IDENTITY(1,  1) NOT NULL PRIMARY  KEY,
                       [Title]  [varchar](50)  NOT NULL UNIQUE,
                       [ISBN]  [varchar](20)  NOT NULL,
                       [PublisherId]  [int] NOT NULL FOREIGN  KEY REFERENCES  [PUBLISHER]  ([PublisherId]),
                       [AuthorId]  [int] NOT NULL  FOREIGN  KEY REFERENCES  [AUTHOR]  ([AuthorId]),
                       [CategoryId]  [int] NOT NULL FOREIGN  KEY REFERENCES  [CATEGORY]  ([CategoryId]),
                       [Description]  [varchar](1000)  NULL,
                       [Year]  [date]  NULL,
                       [Edition]  [int] NULL,
                       [AverageRating]  [float]  NULL,
)
GO
```

SQL Script – Insert some Data into the Tables

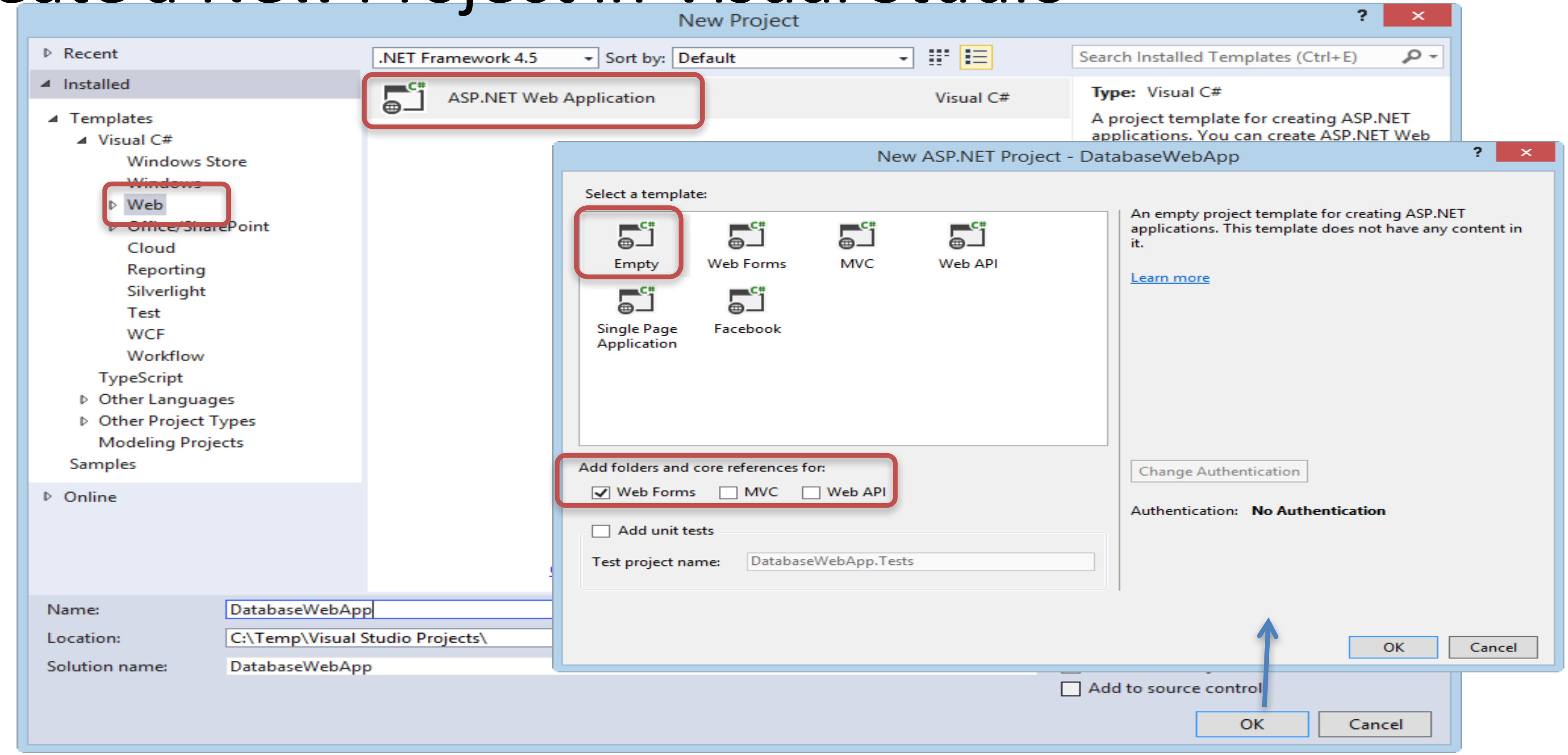
```sql
--CATEGORY ----------------------------------
INSERT INTO CATEGORY (CategoryName) VALUES ('Science')
GO
INSERT INTO CATEGORY (CategoryName) VALUES ('Programming')
GO
INSERT INTO CATEGORY (CategoryName) VALUES ('Novel')
GO

--AUTHOR ----------------------------------
INSERT INTO AUTHOR (AuthorName) VALUES ('Knut Hamsun')
GO
INSERT INTO AUTHOR (AuthorName) VALUES ('Gilbert Strang')
GO
INSERT INTO AUTHOR (AuthorName) VALUES ('J.R.R Tolkien')
GO
INSERT INTO AUTHOR (AuthorName) VALUES ('Dorf Bishop')
GO

--PUBLISHER ----------------------------------
INSERT INTO PUBLISHER (PublisherName) VALUES ('Prentice Hall')
GO
INSERT INTO PUBLISHER (PublisherName) VALUES ('Wiley')
GO
INSERT INTO PUBLISHER (PublisherName) VALUES ('McGraw-Hill')
GO
```

SQL Script – Insert some Data into the Tables

```
--BOOK ----------------------------------
INSERT INTO BOOK (Title, ISBN, PublisherId, AuthorId, CategoryId) VALUES
(
'Introduction to Linear Algebra',
'0-07-066781-0',
(select PublisherId from PUBLISHER where PublisherName='Prentice Hall'),
(select AuthorId from AUTHOR where AuthorName='Gilbert Strang'),
(select CategoryId from CATEGORY where CategoryName='Science')
)
GO

INSERT INTO BOOK (Title, ISBN, PublisherId, AuthorId, CategoryId) VALUES
(
'Modern Control System',
'1-08-890781-0',
(select PublisherId from PUBLISHER where PublisherName='Wiley'),
(select AuthorId from AUTHOR where AuthorName='Dorf Bishop'),
(select CategoryId from CATEGORY where CategoryName='Programming')
)
GO

INSERT INTO BOOK (Title, ISBN, PublisherId, AuthorId, CategoryId) VALUES
(
'The Lord of the Rings',
'2-09-066556-2',
(select PublisherId from PUBLISHER where PublisherName='McGraw-Hill'),
(select AuthorId from AUTHOR where AuthorName='J.R.R Tolkien'),
(select CategoryId from CATEGORY where CategoryName='Novel')
)
GO
```

ASP.NET Web Form

Create a New Project in Visual Studio



New Project

.NET Framework 4.5 Sort by: Default

Search Installed Templates (Ctrl+E)

ASP.NET Web Application Visual C#

Type: Visual C#

A project template for creating ASP.NET applications. You can create ASP.NET Web

Name: DatabaseWebApp

Location: C:\Temp\Visual Studio Projects\

Solution name: DatabaseWebApp

□ Add to source control

New ASP.NET Project - DatabaseWebApp

Select a template:

An empty project template for creating ASP.NET applications. This template does not have any content in it.

Add folders and core references for:

☑ Web Forms □ MVC □ Web API

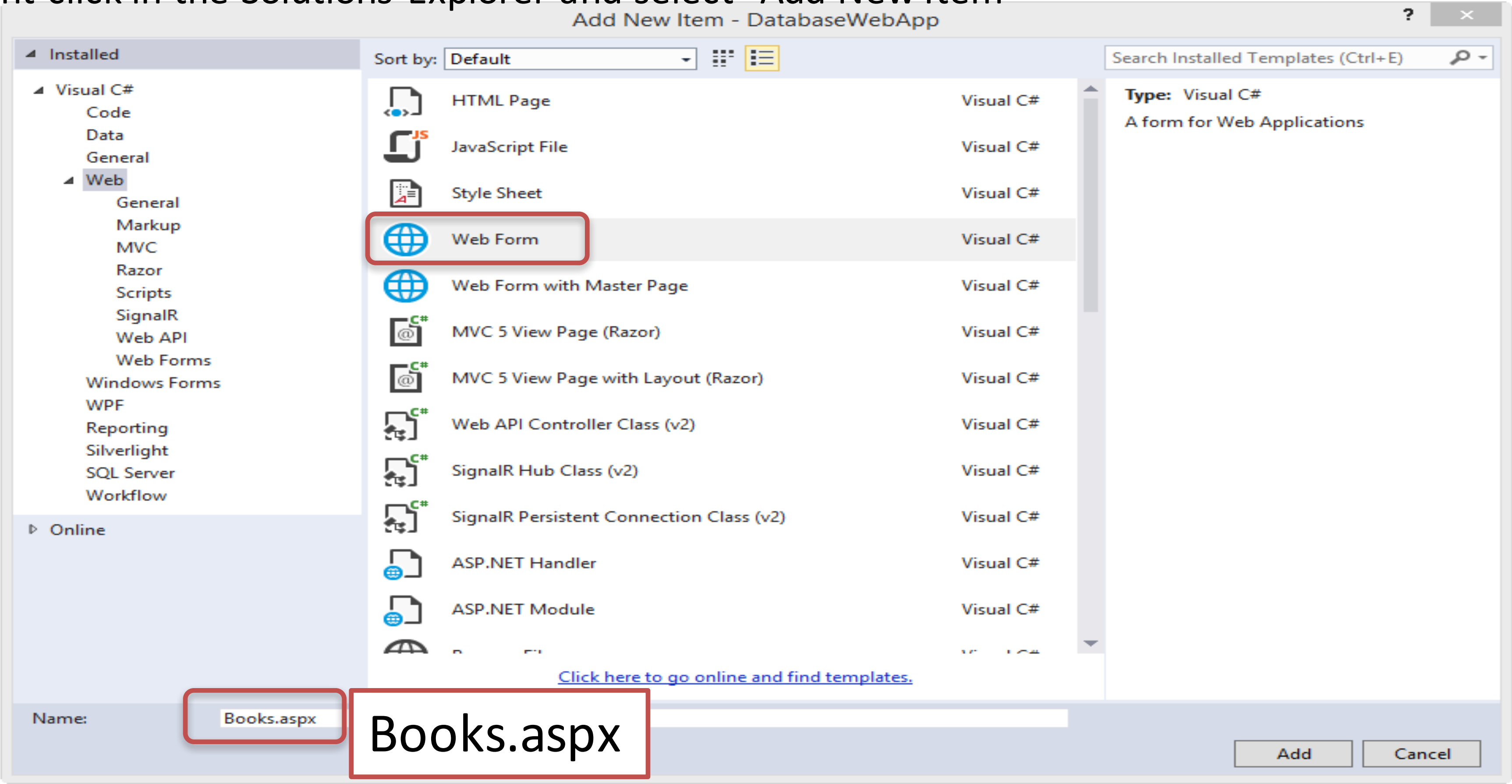
□ Add unit tests

Test project name: DatabaseWebApp.Tests

Authentication: **No Authentication**

Add a New Web Form ("Books.aspx")

Right-click in the Solutions Explorer and select "Add New Item"



Add New Item - DatabaseWebApp

Sort by: Default

Type: Visual C#

A form for Web Applications

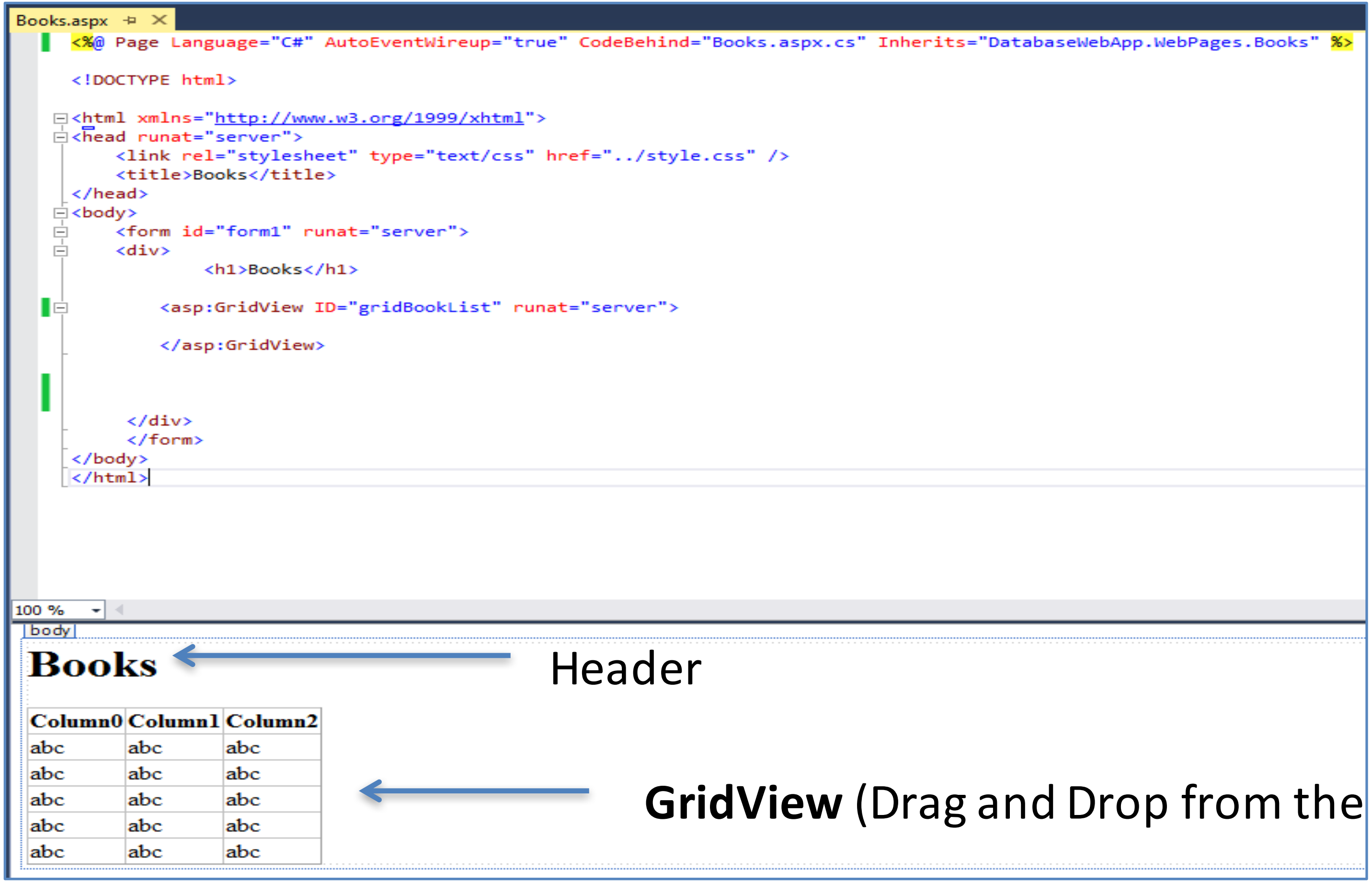
Web Form — Visual C#

Name: Books.aspx

?
Installed
Visual C#
Code
Data
General
Web
General
Markup
MVC
Razor
Scripts
SignalR
Web API
Web Forms
Windows Forms
WPF
Reporting
Silverlight
SQL Server
Workflow
Online
Search Installed Templates (Ctrl+E)
HTML Page Visual C#
JavaScript File Visual C#
Style Sheet Visual C#
Web Form with Master Page Visual C#
MVC 5 View Page (Razor) Visual C#
MVC 5 View Page with Layout (Razor) Visual C#
Web API Controller Class (v2) Visual C#
SignalR Hub Class (v2) Visual C#
SignalR Persistent Connection Class (v2) Visual C#
ASP.NET Handler Visual C#
ASP.NET Module Visual C#
Click here to go online and find templates.
Add
Cancel

Books.aspx

Create the following GUI (“Books.aspx”)



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Books.aspx.cs" Inherits="DatabaseWebApp.WebPages.Books" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <link rel="stylesheet" type="text/css" href="../style.css" />
    <title>Books</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
            <h1>Books</h1>

        <asp:GridView ID="gridBookList" runat="server">

        </asp:GridView>



     </div>
     </form>
</body>
</html>
```

Books

Header

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

GridView (Drag and Drop from the Toolbox)

Create the following Code ("Books.aspx.cs")

```
using System.Web.Configuration;
using DatabaseWebApp.Data;

...

public partial class BookList : System.Web.UI.Page
{
 private string connectionString = WebConfigurationManager.ConnectionStrings["LibraryDBConnectionString"].ConnectionString;

        void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                FillBookGrid();
            }
        }

        private void FillBookGrid()
        {
            List<Book> bookList = new List<Book>();
            Book book = new Book();

            bookList = book.GetBooks(connectionString);

            gridBookList.DataSource = bookList;
            gridBookList.DataBind();
        }
}
```

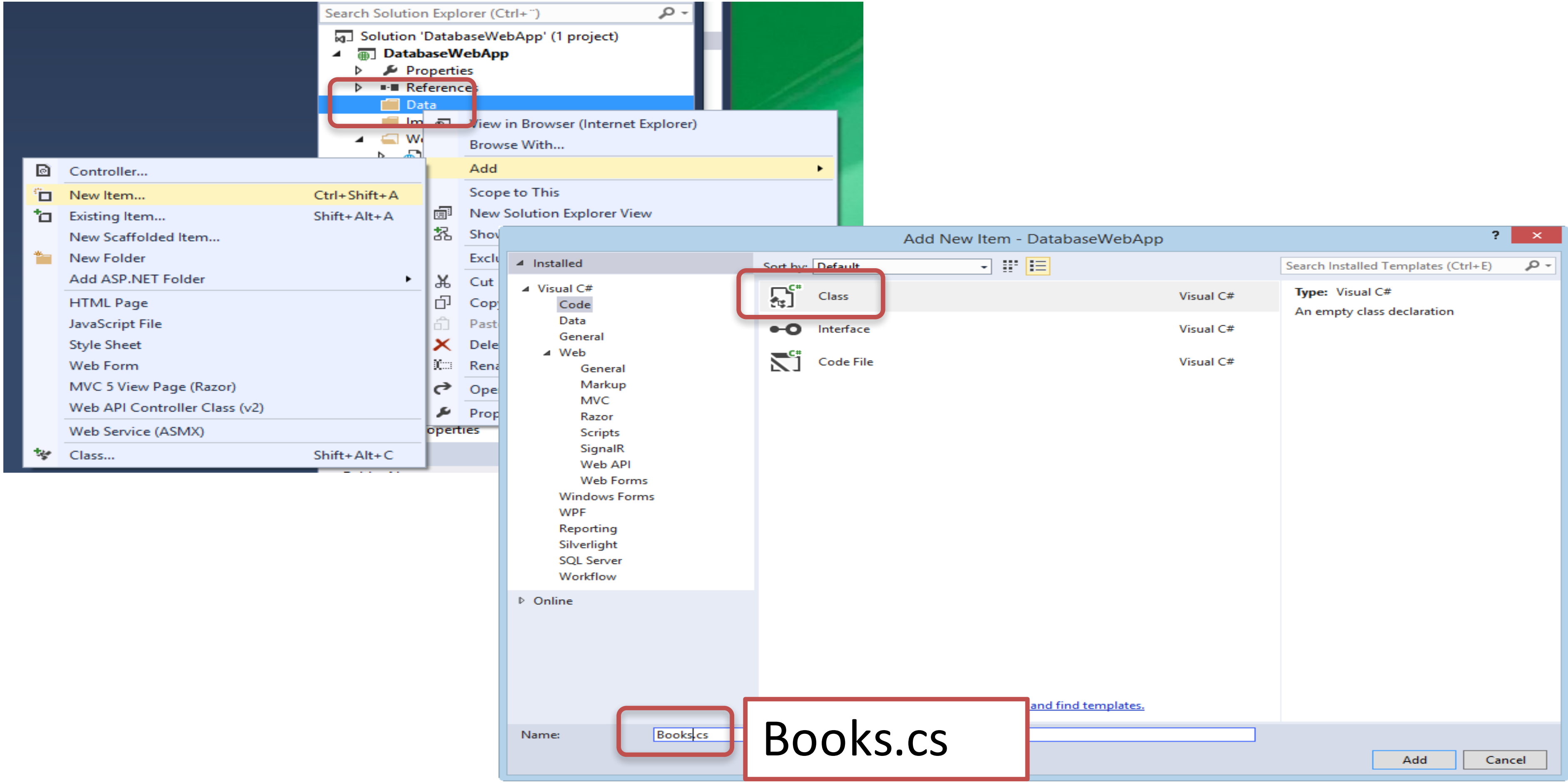
Reference to our Class that communicates with the Database

Note!!

We shall create the Connection String to the Database in the "Web.config page"

See next slides for implementation of the Book Class

Create Database Code – Create a new Class (“Books.cs”)



Type: Visual C#

An empty class declaration

Name: Books.cs

Books.cs

Create the Following Class in “Books.cs”

```
using System.Data.SqlClient;
using System.Data.SqlTypes;
using System.Data;

public class Book
 {
        public int BookId { get; set; }
        public string Title { get; set; }
        public string Isbn { get; set; }
        public string PublisherName { get; set; }
        public string AuthorName { get; set; }
        public string CategoryName { get; set; }

        public List<Book> GetBooks(string connectionString)
        {

            List<Book> bookList = new List<Book>();

            SqlConnection con = new SqlConnection(connectionString);

            string selectSQL = "select BookId, Title, Isbn, PublisherName, AuthorName, CategoryName from GetBookData";

            con.Open();

            SqlCommand cmd = new SqlCommand(selectSQL, con);

            SqlDataReader dr = cmd.ExecuteReader();

            if (dr != null)
            {
                while (dr.Read())
                {
                    Book book = new Book();

                    book.BookId = Convert.ToInt32(dr["BookId"]);
                    book.Title = dr["Title"].ToString();
                    book.Isbn = dr["ISBN"].ToString();
                    book.PublisherName = dr["PublisherName"].ToString();
                    book.AuthorName = dr["AuthorName"].ToString();
                    book.CategoryName = dr["CategoryName"].ToString();

                    bookList.Add(book);
                }
            }

            return bookList;
        }
}
```

GetBookData is a View (see next slide)

SQL Script – Views – “GetBookData”

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

CREATE VIEW GetBookData
AS

SELECT
BOOK.BookId,
BOOK.Title,
BOOK.ISBN,
PUBLISHER.PublisherName,
AUTHOR.AuthorName,
CATEGORY.CategoryName

FROM BOOK
INNER JOIN AUTHOR ON BOOK.AuthorId = AUTHOR.AuthorId
INNER JOIN PUBLISHER ON BOOK.PublisherId = PUBLISHER.PublisherId
INNER JOIN CATEGORY ON BOOK.CategoryId = CATEGORY.CategoryId

GO
```

Create Database Connection String in “Web.config”

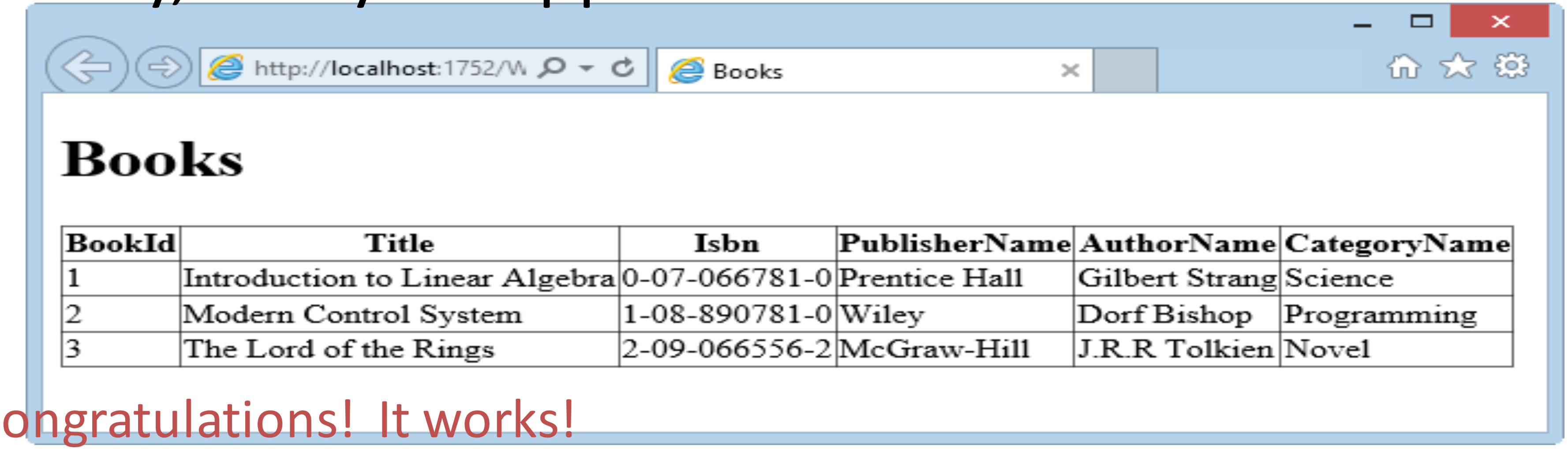
Your Database

```
<connectionStrings>
   <add name="LibraryDBConnectionString"
             connectionString="Data Source=macwin8;Initial Catalog=BOOKS;Persist Security Info=True;User ID=sa;Password=xxx"
             providerName="System.Data.SqlClient" />
 </connectionStrings>
```

Where “xxx” is your SQL Server Database Password

UserName and Password
for your SQL Server

Finally, Run your application:



Books

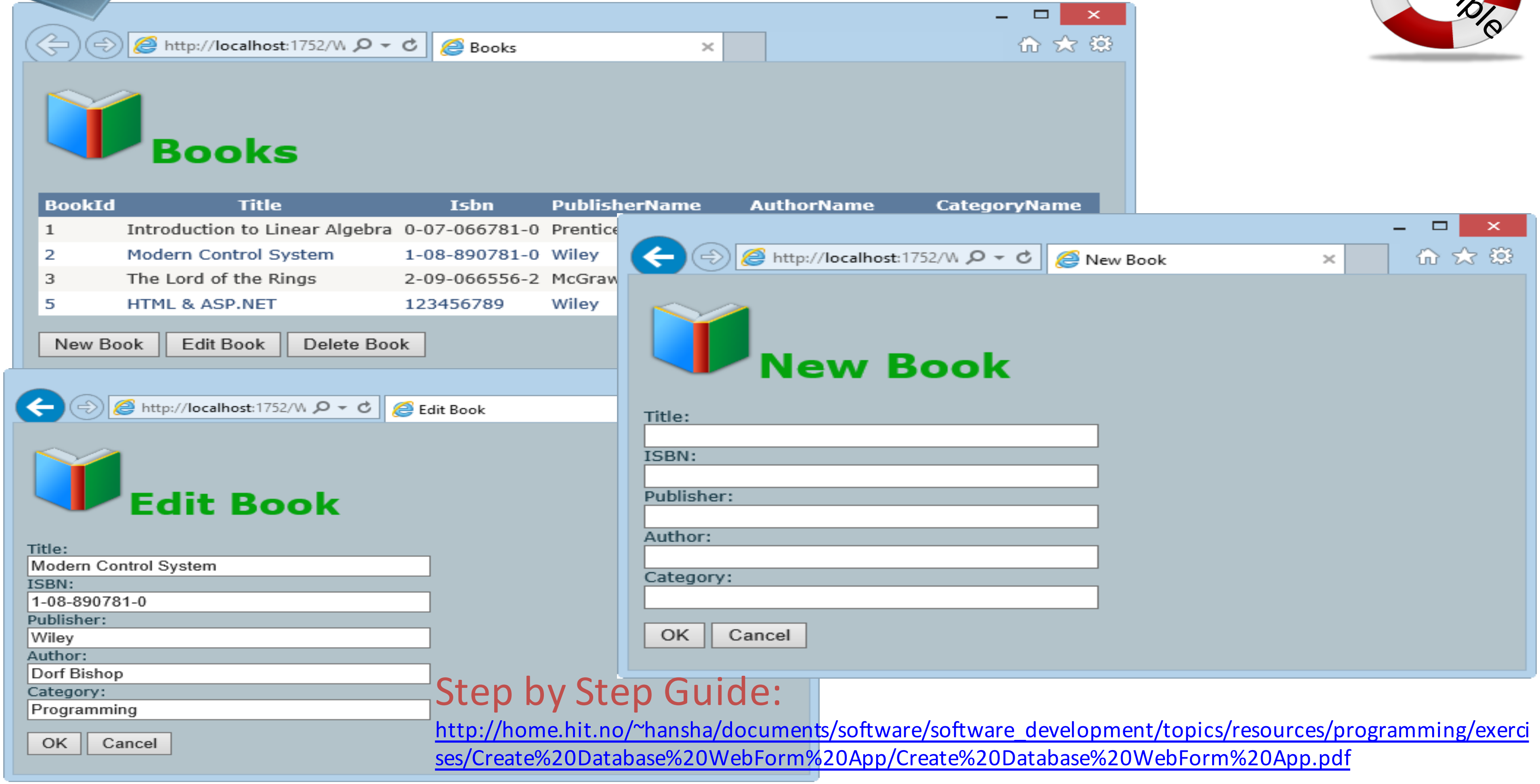
BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentice Hall	Gilbert Strang	Science
2	Modern Control System	1-08-890781-0	Wiley	Dorf Bishop	Programming
3	The Lord of the Rings	2-09-066556-2	McGraw-Hill	J.R.R Tolkien	Novel

Congratulations! It works!

Congratulations! - You are finished with the Example

Try to Create Add, Edit and Delete as well

Example



http://localhost:1752/W

Books

Books

BookId	Title	Isbn	PublisherName	AuthorName	CategoryName
1	Introduction to Linear Algebra	0-07-066781-0	Prentic		
2	Modern Control System	1-08-890781-0	Wiley		
3	The Lord of the Rings	2-09-066556-2	McGraw		
5	HTML & ASP.NET	123456789	Wiley		

http://localhost:1752/W

Edit Book

Edit Book

Title: Modern Control System

ISBN: 1-08-890781-0

Publisher: Wiley

Author: Dorf Bishop

Category: Programming

http://localhost:1752/W

New Book

New Book

Title:

ISBN:

Publisher:

Author:

Category:

Step by Step Guide:

http://home.hit.no/~hansha/documents/software/software_development/topics/resources/programming/exercises/Create%20Database%20WebForm%20App/Create%20Database%20WebForm%20App.pdf

Congratulations! - You are finished with the Example

Congratulations! - You are finished with all the Examples in this Tutorial

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/