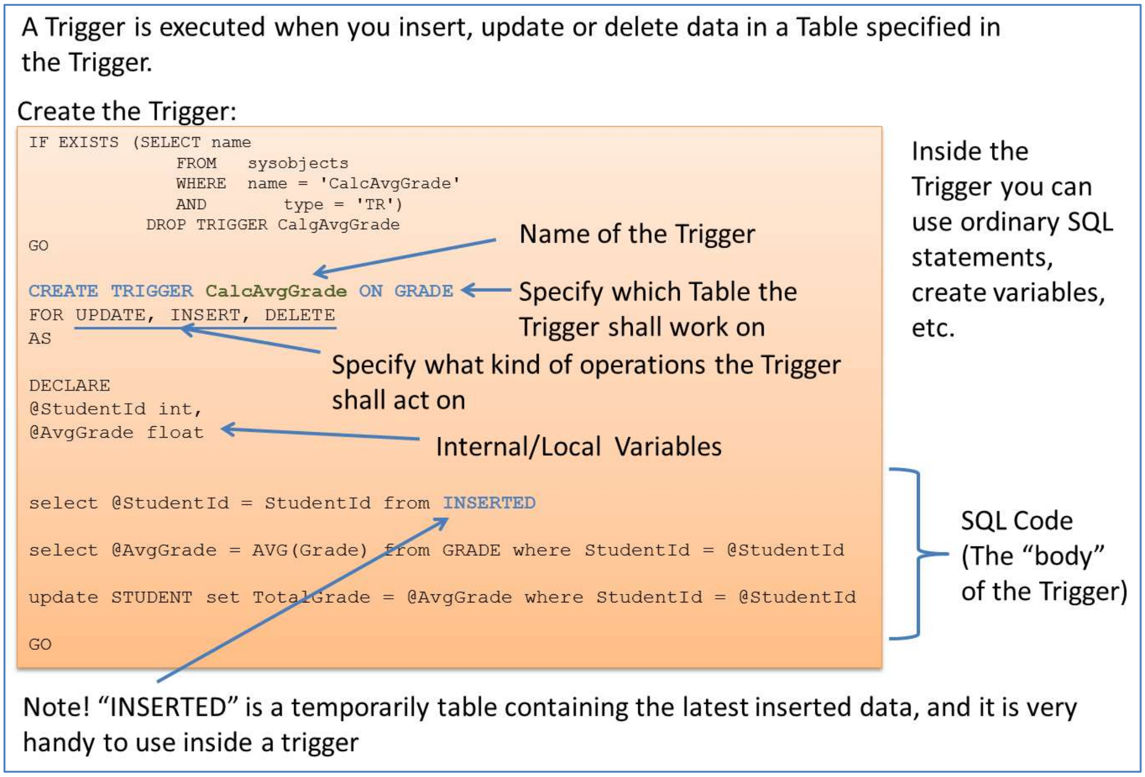
12 Triggers

A database trigger is code that is automatically executed in response to certain events on a particular table in a database.



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
```

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)

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

Syntax for creating a Trigger:

```
CREATE TRIGGER <TriggerName> on <TableName>
FOR INSERT, UPDATE, DELETE
AS
… Create your Code here
GO
```

The Trigger will automatically be executed when data is inserted, updated or deleted in the table as specified in the Trigger header.

INSERTED and DELETED:

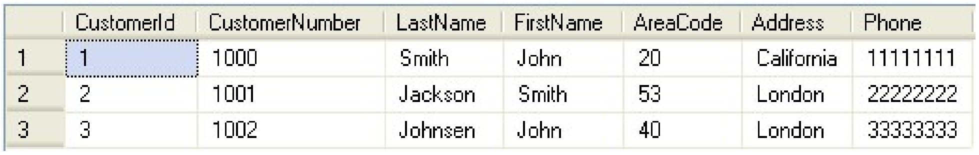
Inside triggers we can use two special tables: the DELETED table and the INSERTED tables. SQL Server automatically creates and manages these tables. You can use these temporary, memory-resident tables to test the effects of certain data modifications. You cannot modify the data in these tables.

The DELETED table stores copies of the affected rows during DELETE and UPDATE statements. During the execution of a DELETE or UPDATE statement, rows are deleted from the trigger table and transferred to the DELETED table.

The INSERTED table stores copies of the affected rows during INSERT and UPDATE statements. During an insert or update transaction, new rows are added to both the INSERTED table and the trigger table. The rows in the INSERTED table are copies of the new rows in the trigger table.

Example:

We will use the CUSTOMER table as an example:



	CustomerId	CustomerNumber	LastName	FirstName	AreaCode	Address	Phone
1	1	1000	Smith	John	20	California	11111111
2	2	1001	Jackson	Smith	53	London	22222222
3	3	1002	Johnsen	John	40	London	33333333

We will create a TRIGGER that will check if the Phone number is valid when we insert or update data in the CUSTOMER table. The validation check will be very simple, i.e., we will check if the Phone number is less than 8 digits (which is normal length in Norway). If the Phone number is less than 8 digits, the following message “Phone Number is not valid” be written in place of the wrong number in the Phone column.

The TRIGGER becomes something like this:

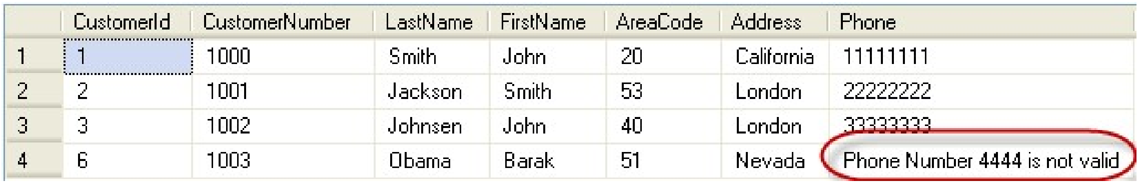
```
IF EXISTS (SELECT name
        FROM   sysobjects
        WHERE  name = 'CheckPhoneNumber'
        AND        type = 'TR')
     DROP TRIGGER CheckPhoneNumber
GO

CREATE TRIGGER CheckPhoneNumber ON CUSTOMER
FOR UPDATE, INSERT
AS

DECLARE
@CustomerId int,
@Phone varchar(50),
@Message varchar(50)

set nocount on

select @CustomerId = CustomerId from INSERTED

select @Phone = Phone from INSERTED
```

```
set @Message = 'Phone Number ' + @Phone + ' is not valid'


if len(@Phone) < 8 --Check if Phone Number have less than 8 digits
      update CUSTOMER set Phone = @Message where CustomerId =
@CustomerId

set nocount off

GO
```

We test the TRIGGER with the following INSERT INTO statement:

```
INSERT INTO CUSTOMER
(CustomerNumber, LastName, FirstName, AreaCode, Address, Phone)

VALUES
('1003', 'Obama', 'Barak', 51, 'Nevada', '4444')
```

The results become:



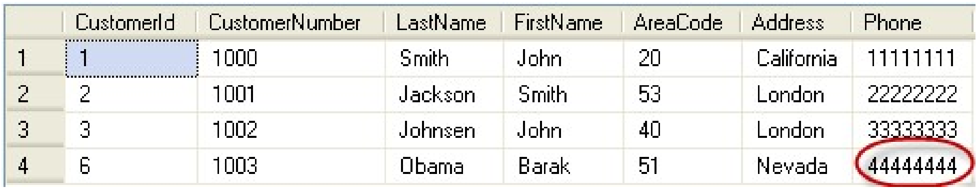
	CustomerId	CustomerNumber	LastName	FirstName	AreaCode	Address	Phone
1	1	1000	Smith	John	20	California	11111111
2	2	1001	Jackson	Smith	53	London	22222222
3	3	1002	Johnsen	John	40	London	33333333
4	6	1003	Obama	Barak	51	Nevada	Phone Number 4444 is not valid

As you can see, the TRIGGER works as expected.

We try to update the Phone number to a valid number:

```
update CUSTOMER set Phone = '44444444' where CustomerNumber = '1003'
```

The results become:



	CustomerId	CustomerNumber	LastName	FirstName	AreaCode	Address	Phone
1	1	1000	Smith	John	20	California	11111111
2	2	1001	Jackson	Smith	53	London	22222222
3	3	1002	Johnsen	John	40	London	33333333
4	6	1003	Obama	Barak	51	Nevada	44444444

13 Communication from other Applications

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:

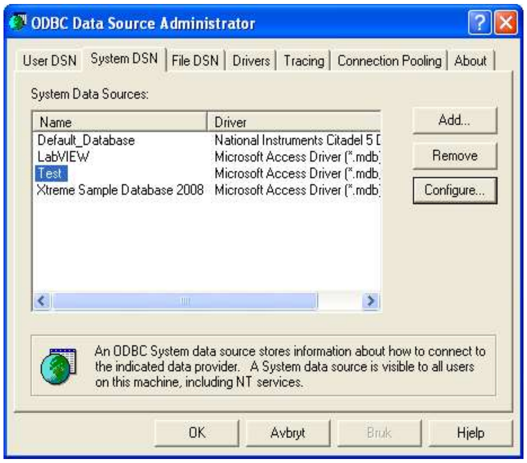
- Bank/Account systems
- Information in Web pages such as Facebook, Wikipedia, YouTube
- … lots of other examples

This means we need to be able to communicate with the database from other applications and programming languages in order to insert, update or retrieve data from the database.

13.1 ODBC

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.

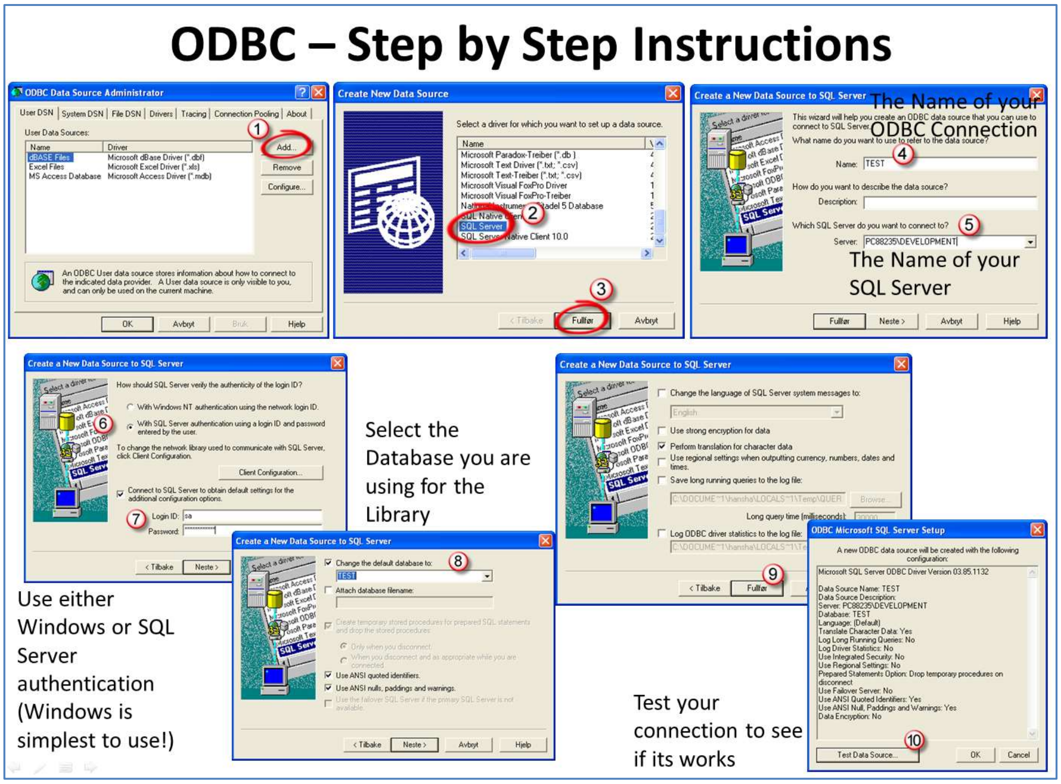
We will use the ODBC Data Source Administrator:



System Data Sources:

Name	Driver
Default_Database	National Instruments Citadel 5 [
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.



ODBC – Step by Step Instructions

ODBC Data Source Administrator

User Data Sources:

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

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.

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
SQL Server
SQL Server Native Client 10.0

Create a New Data Source to SQL Server

The Name of your ODBC Connection

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?

Name: TEST

How do you want to describe the data source?

Description:

Which SQL Server do you want to connect to?

Server: PC88235\DEVELOPMENT

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.

● 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.

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

Login ID: sa
Password: ************

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

Select the Database you are using for the Library

Create a New Data Source to SQL Server

☑ Change the default database to:

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.

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

Long query time (milliseconds): 30000

☐ Log ODBC driver statistics to the log file:

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

ODBC Microsoft SQL Server Setup

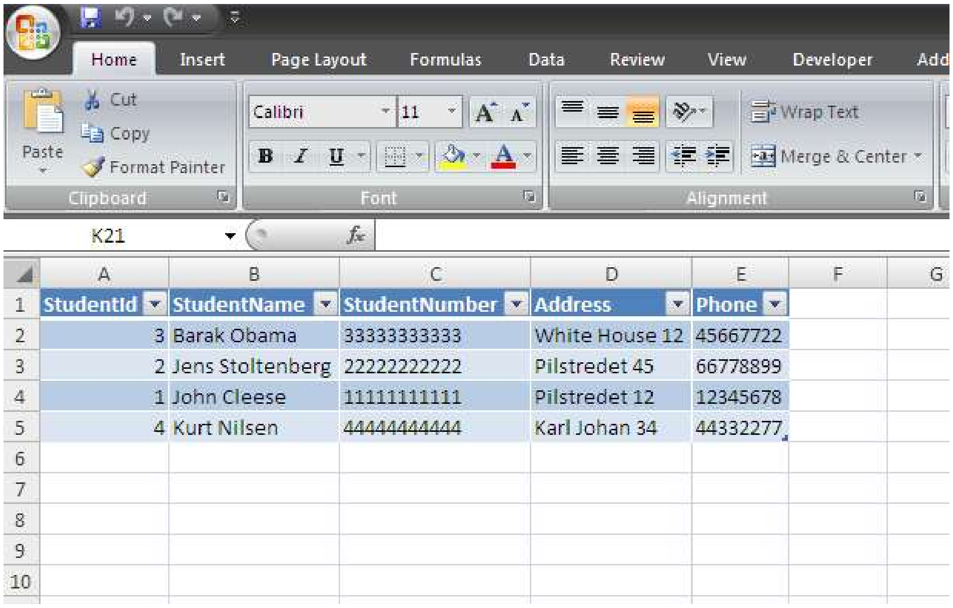
A new ODBC data source will be created with the following configuration:

```
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
```

Test your connection to see if its works

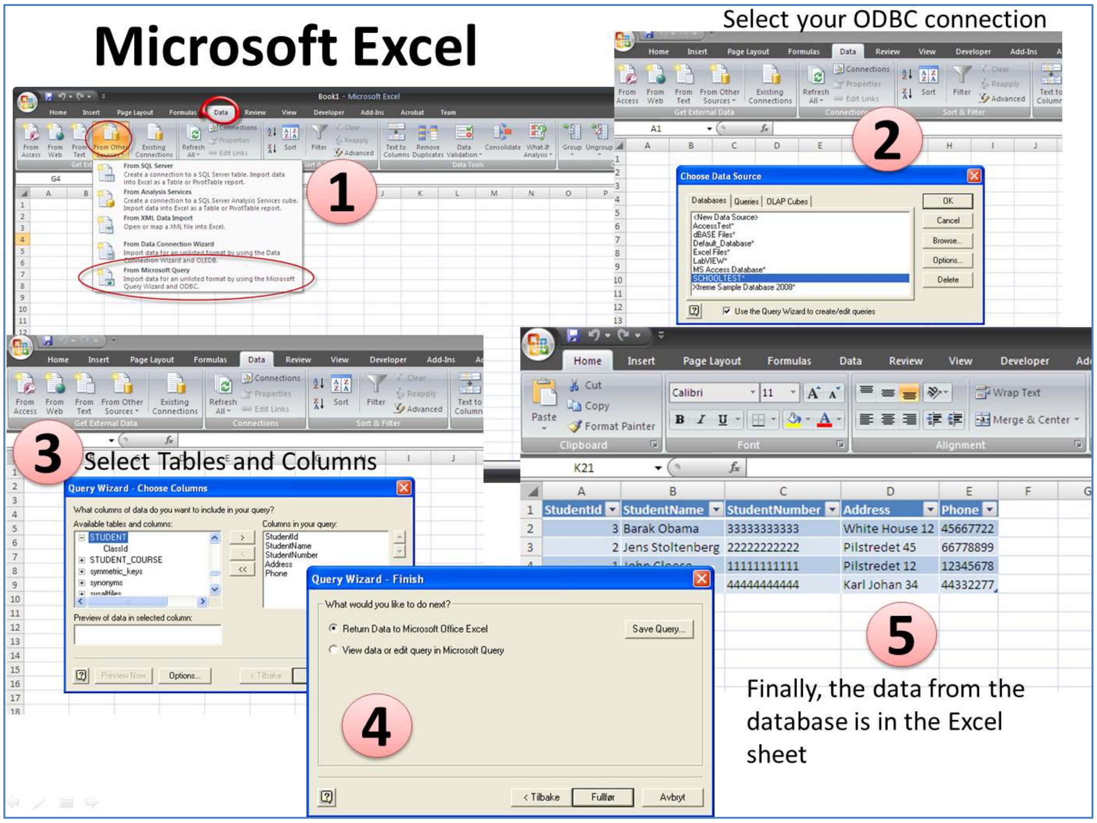
13.2 Microsoft Excel

Microsoft Excel has the ability to retrieve data from different data sources, including different database systems. It is very simple to retrieve data from SQL Server into Excel since Excel and SQL Server has the same vendor (Microsoft).



	A	B	C	D	E	F	G
1	StudentId	StudentName	StudentNumber	Address	Phone		
2	3	Barak Obama	33333333333	White House 12	45667722		
3	2	Jens Stoltenberg	22222222222	Pilstredet 45	66778899		
4	1	John Cleese	11111111111	Pilstredet 12	12345678		
5	4	Kurt Nilsen	44444444444	Karl Johan 34	44332277		
6							
7							
8							
9							
10							

Microsoft Excel



1. From SQL Server — Create a connection to a SQL Server table. Import data into Excel as a Table or PivotTable report.
From Analysis Services — Create a connection to a SQL Server Analysis Services cube. Import data into Excel as a Table or PivotTable report.
From XML Data Import — Open or map a XML file into Excel.
From Data Connection Wizard — Import data for an unlisted format by using the Data Connection Wizard and OLEDB.
From Microsoft Query — Import data for an unlisted format by using the Microsoft Query Wizard and ODBC.

2. Select your ODBC connection

Choose Data Source

Databases | Queries | OLAP Cubes

- <New Data Source>
- AccessTest*
- dBASE Files*
- Default_Database*
- Excel Files*
- LabVIEW*
- MS Access Database*
- SCHOOLTEST*
- Xtreme Sample Database 2008*

OK | Cancel | Browse... | Options... | Delete

☑ Use the Query Wizard to create/edit queries

3. Select Tables and Columns

Query Wizard - Choose Columns

What columns of data do you want to include in your query?

Available tables and columns:

- STUDENT
  - ClassId
- STUDENT_COURSE
- symmetric_keys
- synonyms
- sysaltfiles

Columns in your query:

- StudentId
- StudentName
- StudentNumber
- Address
- Phone

Preview of data in selected column:

Preview Now | Options... | < Tilbake

4. Query Wizard - Finish

What would you like to do next?

● Return Data to Microsoft Office Excel

○ View data or edit query in Microsoft Query

Save Query... | < Tilbake | Fullfør | Avbryt

5. K21

	A	B	C	D	E
1	StudentId	StudentName	StudentNumber	Address	Phone
2	3	Barak Obama	33333333333	White House 12	45667722
3	2	Jens Stoltenberg	22222222222	Pilstredet 45	66778899
4	1	John Cleese	11111111111	Pilstredet 12	12345678
			44444444444	Karl Johan 34	44332277

Finally, the data from the database is in the Excel sheet

14 References

My Blog: https://www.halvorsen.blog

Microsoft official SQL Server Web site - http://www.microsoft.com/sqlserver

SQL Server Books Online - http://msdn.microsoft.com/en-us/library/ms166020.aspx

SQL Server Help

w3shools.com - http://www.w3schools.com/sql

Wikipedia – Microsoft SQL Server - http://en.wikipedia.org/wiki/Microsoft_SQL_Server

Wikipedia - SQL - http://en.wikipedia.org/wiki/SQL

Wikipedia – Transact SQL - http://en.wikipedia.org/wiki/T-SQL