Database Lab

CSE 3104

Lab-02

1 Introduction to SQL

SQL (Structured Query Language) is a database computer language designed for managing data in relational database management systems (RDBMS).

SQL is a standardized computer language that was originally developed by IBM for querying, altering and defining relational databases, using declarative statements.

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**

What can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database

• SQL can set permissions on tables, procedures, and views

Even if SQL is a standard, many of the database systems that exist today implement their own version of the SQL language. In this document we will use the Microsoft SQL Server as an example.

There are lots of different database systems, or DBMS–Database Management Systems, such as:

• Microsoft SQL Server

 o Enterprise, Developer versions, etc

 o Express version is free of charge

• Oracle

• MySQL (Oracle, previously 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

• … lots of other systems

1.1 Data Definition Language (DDL)

The **Data Definition Language (DDL)** manages table and index structure. The most basic items of DDL are the CREATE, ALTER, RENAME and DROP statements:

- **CREATE** creates an object (a table, for example) in the database.
- **DROP** deletes an object in the database, usually irretrievably.
- **ALTER** modifies the structure an existing object in various ways—for example, adding a column to an existing table.

1.2 Data Manipulation Language(DML)

The **Data Manipulation Language (DML)** is the subset of SQL used to add, update and delete data.

The acronym **CRUD** refers to all of the major functions that need to be implemented in a relational database application to consider it complete. Each letter in the acronym can be mapped to a standard SQL statement:

Operation	SQL	Description
Create	INSERT INTO	inserts new data into a database
Read (Retrieve)	SELECT	extracts data from a database
Update	UPDATE	updates data in a database
Delete (Destroy)	DELETE	deletes data from a database

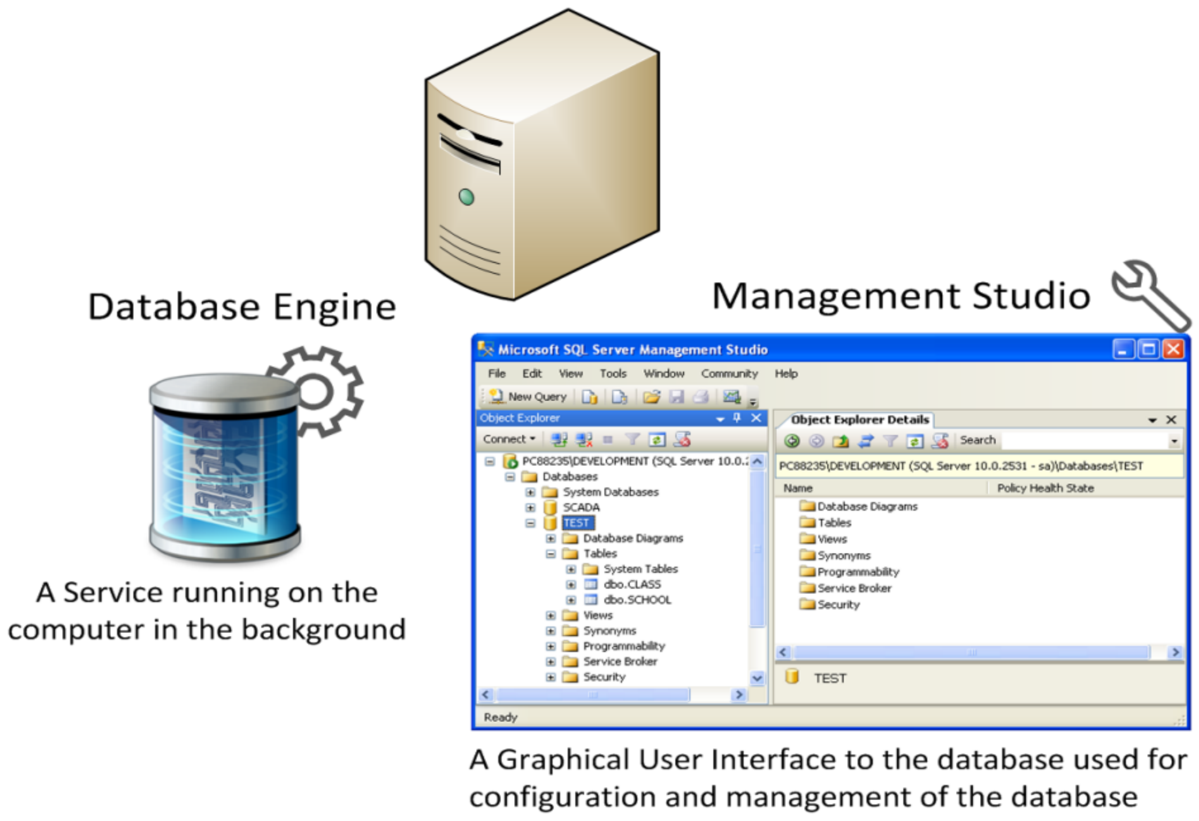
2 Introduction to SQL Server

Microsoft is the vendor of SQL Server. The newest version is “SQL Server 2012”.

We have different editions of SQL Server, where SQL Server Express is free to download and use.

SQL Server uses T-SQL (Transact-SQL). T-SQL is Microsoft's proprietary extension to SQL. T-SQL is very similar to standard SQL, but in addition it supports some extra functionality, built-in functions, etc. T-SQL expands on the SQL standard to include procedural programming, local variables, various support functions for string processing, date processing, mathematics, etc.

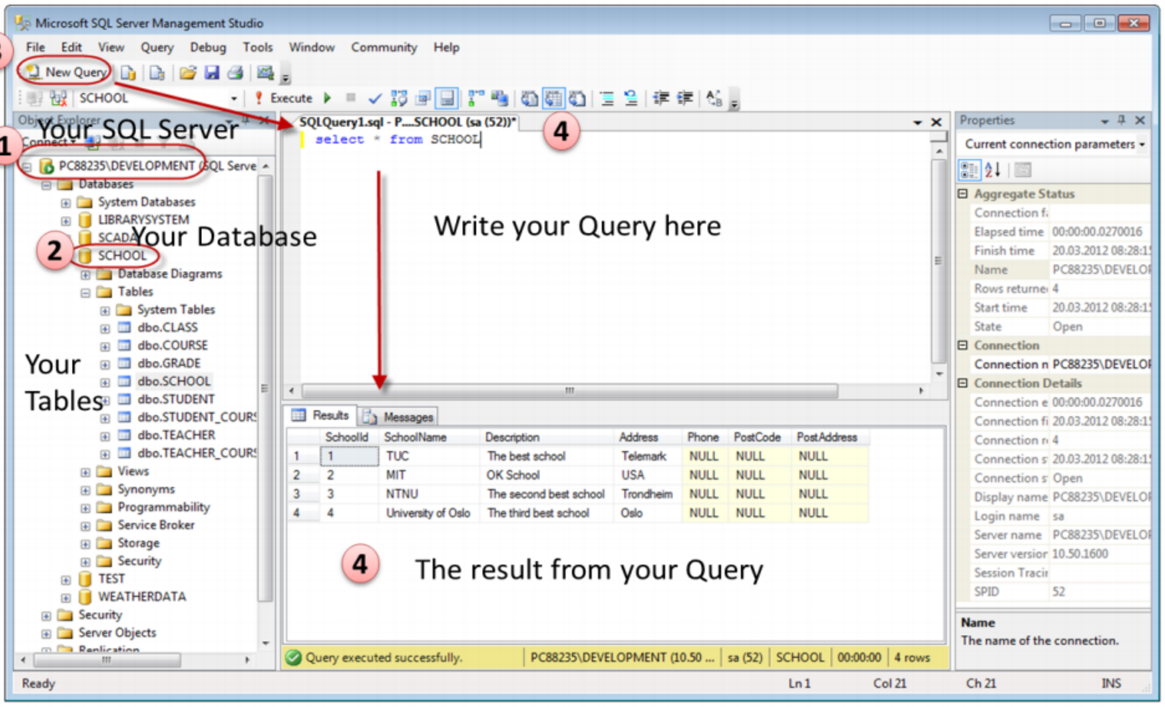
SQL Server consists of a **Database Engine** and a **Management Studio** (and lots of other stuff which we will not mention here). The Database engine has no graphical interface - it is just a service running in the background of your computer (preferable on the server). The Management Studio is graphical tool for configuring and viewing the information in the database. It can be installed on the server or on the client (or both).



2.1 SQL ServerManagement Studio

SQL Server Management Studio is a GUI tool included with SQL Server for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. As mentioned earlier, version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as SQL Server Management Studio Express.

A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.



```
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 f	
Elapsed time	00:00:00.0270016
Finish time	20.03.2012 08:28:1
Name	PC88235\DEVELOP
Rows returne	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

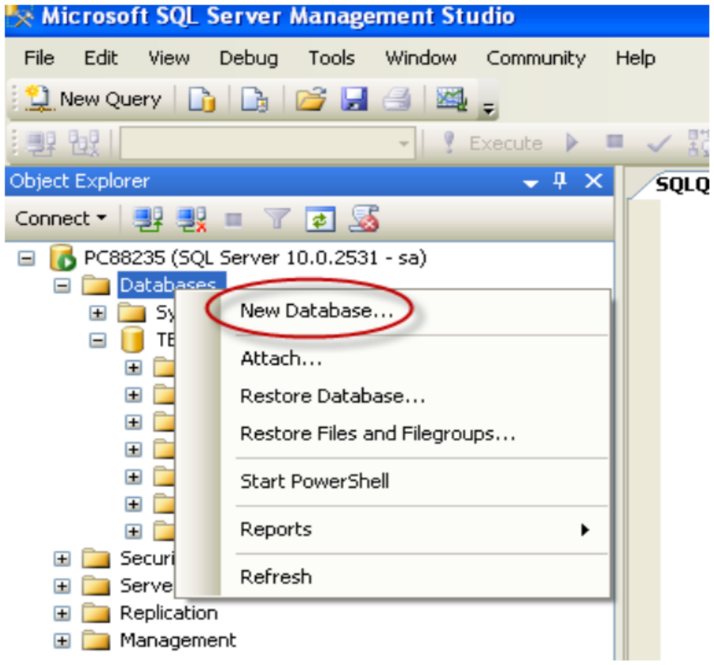
Name
The name of the connection.

When creating SQL commands and queries, the “Query Editor” (select “New Query” from the Toolbar) is used (shown in the figure above).

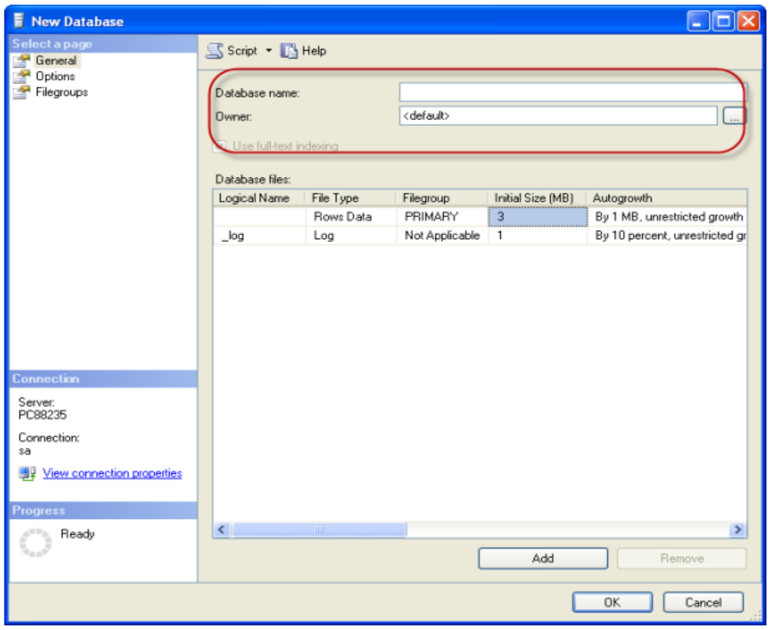
With SQL and the “Query Editor” we can do almost everything with code, but sometimes it is also a good idea to use the different Designer tools in SQL to help us do the work without coding (so much).

2.1.1 Create a new Database

It is quite simple to create a new database in Microsoft SQL Server. Just right-click on the "Databases" node and select "New Database..."



There are lots of settings you may set regarding your database, but the only information you must fill in is the name of your database:



Database name:

Owner: <default>

Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth
	Rows Data	PRIMARY	3	By 1 MB, unrestricted growth
_log	Log	Not Applicable	1	By 10 percent, unrestricted gr

Connection

Server:
PC88235

Connection:
sa

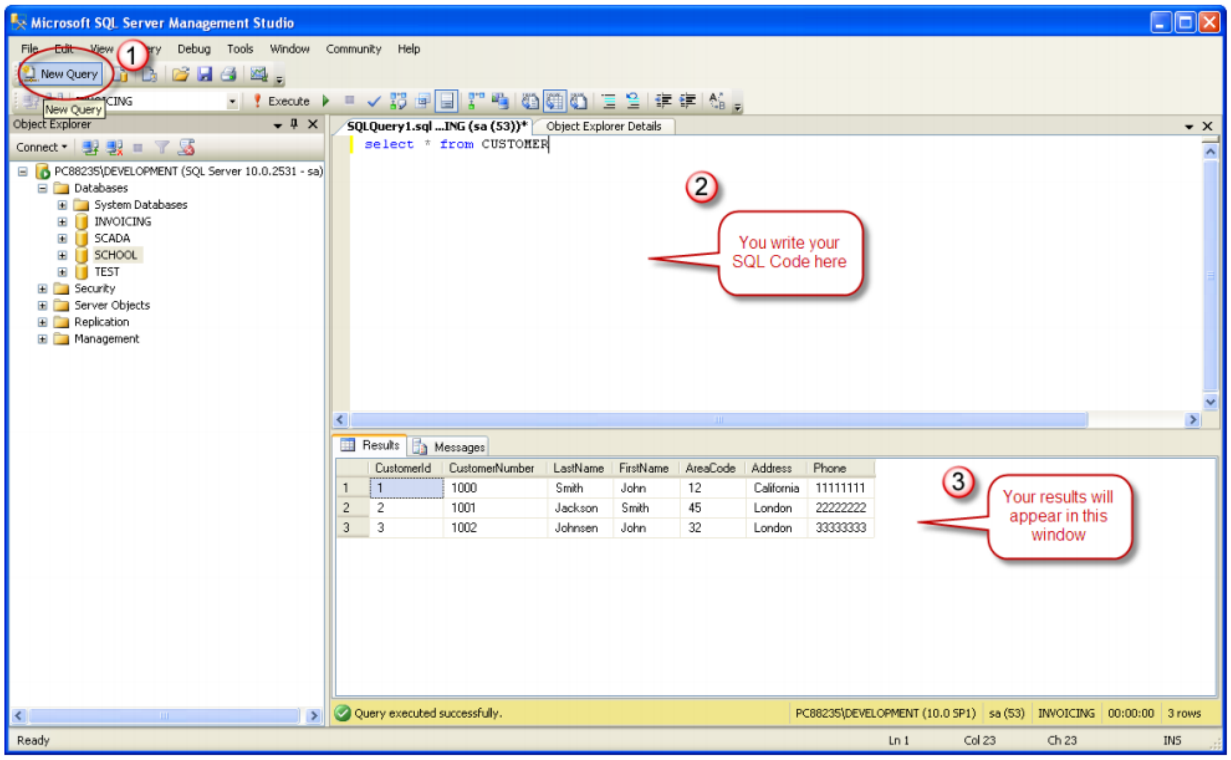
View connection properties

Progress

Ready

2.1.2 Queries

In order to make a new SQL query, select the “New Query” button from the Toolbar.


Microsoft SQL Server Management Studio
New Query
1
2
You write your SQL Code here
3
Your results will appear in this window
Object Explorer
Object Explorer Details
Results
Messages

```
select * from CUSTOMER
```

	CustomerId	CustomerNumber	LastName	FirstName	AreaCode	Address	Phone
1	1	1000	Smith	John	12	California	11111111
2	2	1001	Jackson	Smith	45	London	22222222
3	3	1002	Johnsen	John	32	London	33333333

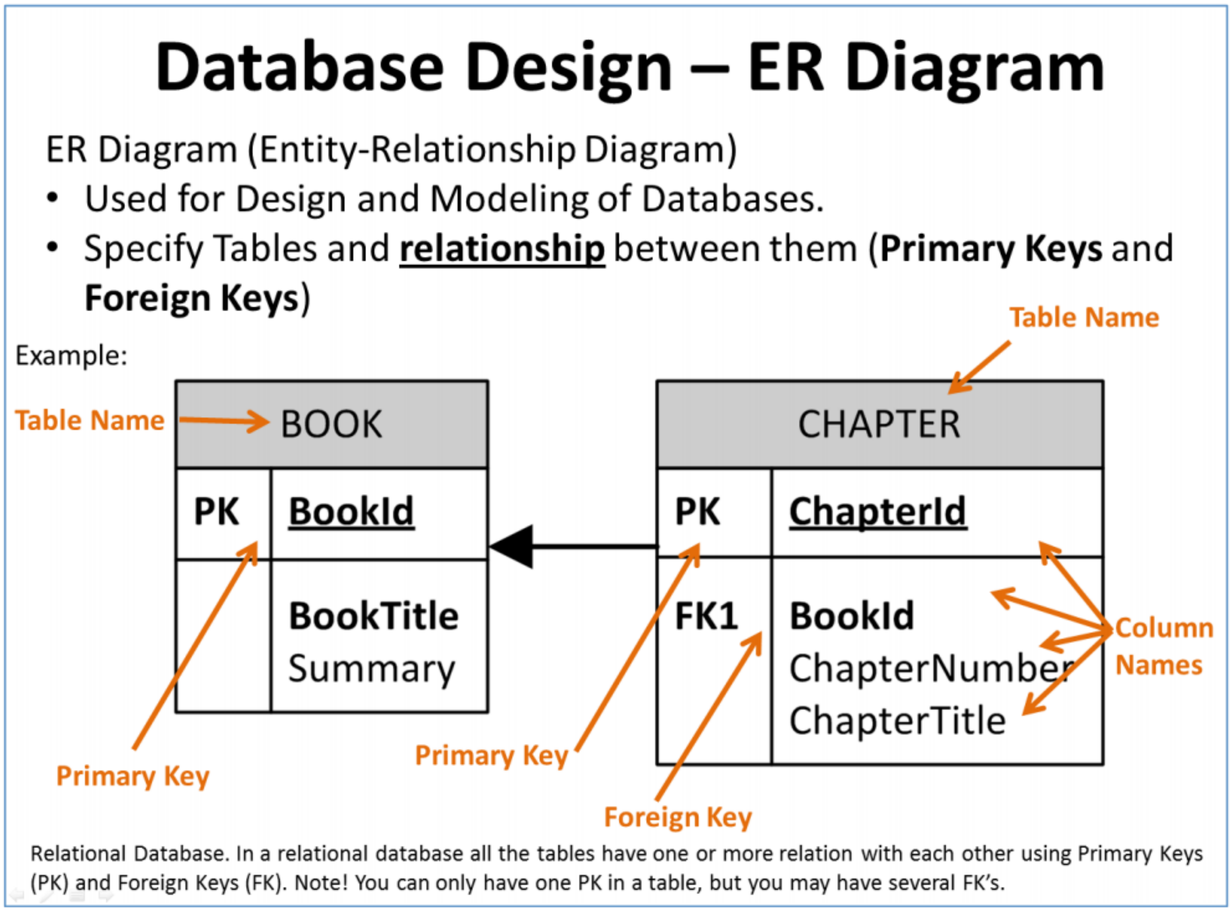
```
Query executed successfully.    PC88235\DEVELOPMENT (10.0 SP1)   sa (53)   INVOICING   00:00:00   3 rows
```

Here we can write any kind of queries that is supported by the SQL language.

Note: You may also use the SQL language to create a new database, but sometimes it is easier to just use the built-in features in the Management Studio.

3 CREATE TABLE

Before you start implementing your tables in the database, you should always spend some time design your tables properly using a design tool like, e.g., ERwin, Toad Data Modeler, PowerDesigner, Visio, etc. This is called Database Modeling.



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.

The **CREATE TABLE** statement is used to create a table in a database.

Syntax:

```
CREATE TABLE table_name
(
column_name1 data_type,
column_name2 data_type,
column_name3 data_type,
....
)
```

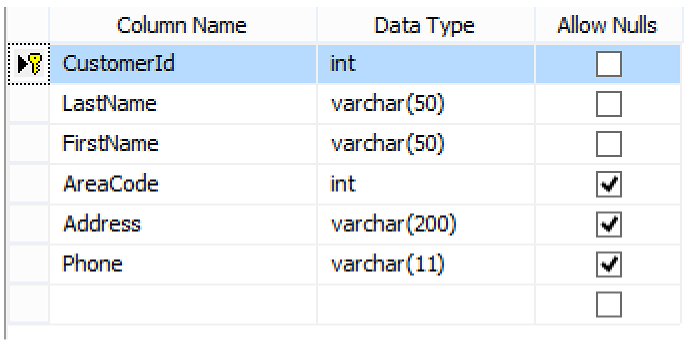
The data type specifies what type of data the column can hold.

You have special data types for numbers, text dates, etc.

Examples:

- Numbers: **int**, **float**
- Text/Stings: **varchar(X)** – where X is the length of the string
- Dates: **datetime**
- etc.

Example: We want to create a table called "CUSTOMER" which has the following columns and data types:



	Column Name	Data Type	Allow Nulls
	CustomerId	int	☐
	LastName	varchar(50)	☐
	FirstName	varchar(50)	☐
	AreaCode	int	✓
	Address	varchar(200)	✓
	Phone	varchar(11)	✓
			☐

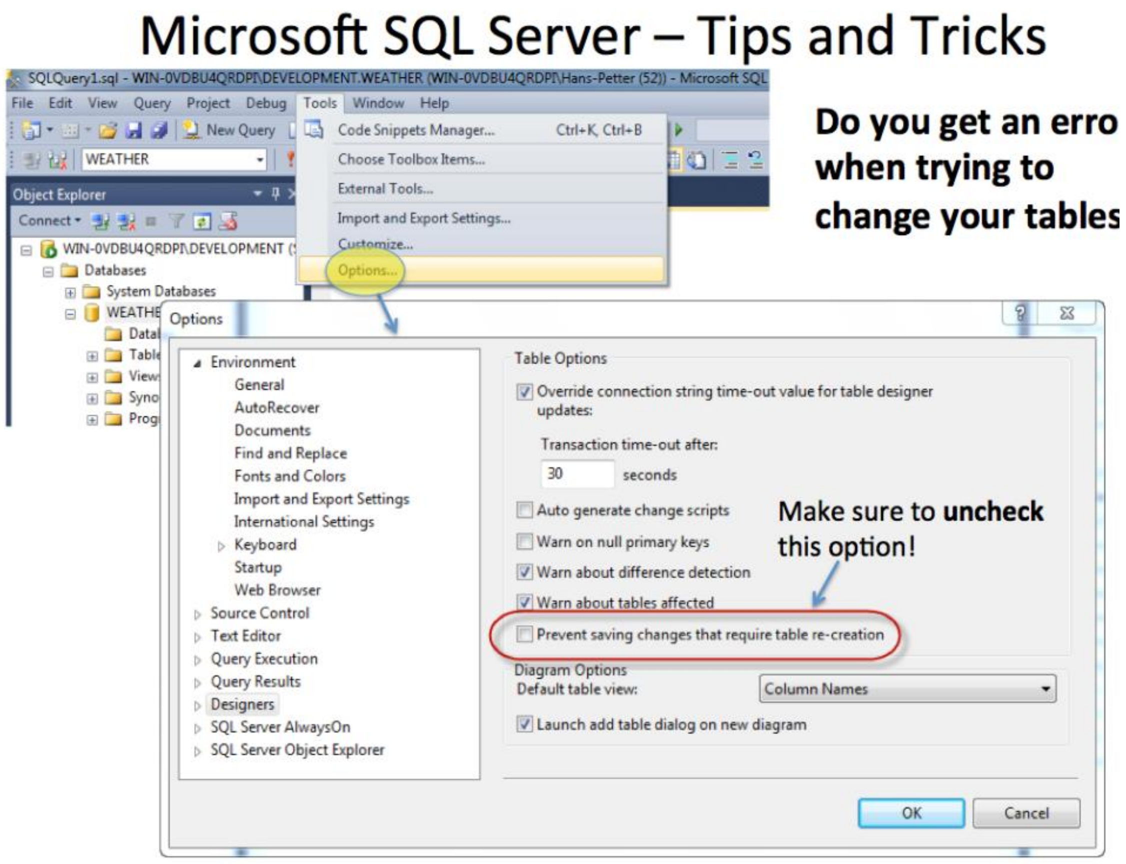
```
CREATE TABLE CUSTOMER
(
        CustomerId int IDENTITY(1,1) PRIMARY KEY,
        LastName varchar(50) NOT NULL,
        FirstName varchar(50) NOT NULL,
        AreaCode int NULL,
        Address varchar(200) NULL,
        Phone varchar(11) NULL,
)
```

Best practice:

When creating tables you should consider following these guidelines:

- 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, ...)


Microsoft SQL Server – Tips and Tricks

Do you get an erro when trying to change your tables

Make sure to uncheck this option!

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

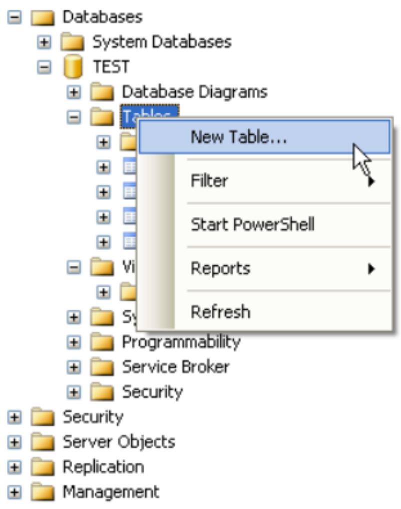
Diagram Options
- Default table view: Column Names
- [✓] Launch add table dialog on new diagram

3.2 Create Tables using the Designer Tools

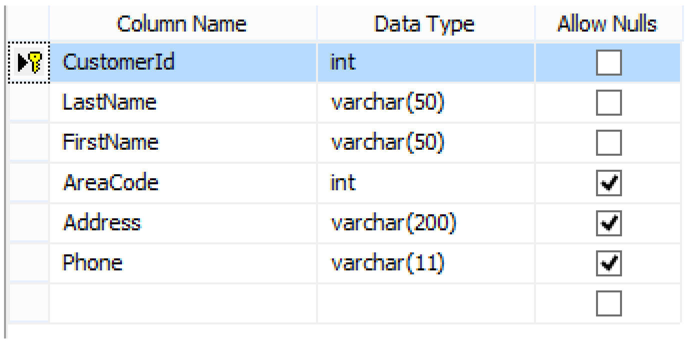
Even if you can do "everything" using the SQL language, it is sometimes easier to do it in the designer tools in the Management Studio in SQL Server.

Instead of creating a script you may as well easily use the designer for creating tables.

Step1: Select "New Table ...":



Step2: Next, the table designer pops up where you can add columns, data types, etc.



Column Name	Data Type	Allow Nulls
CustomerId	int	☐
LastName	varchar(50)	☐
FirstName	varchar(50)	☐
AreaCode	int	✓
Address	varchar(200)	✓
Phone	varchar(11)	✓
		☐

In this designer we may also specify Column Names, Data Types, etc.

Step 3: Save the table by clicking the Save button.

4 INSERT INTO

The INSERT INTO statement is used to insert a new row in a table.

It is possible to write the INSERT INTO statement in two forms.

The first form doesn't specify the column names where the data will be inserted, only their values:

```
INSERT INTO table_name

VALUES (value1, value2, value3,...)
```

Example:

```
INSERT INTO CUSTOMER

VALUES ('Rahman', 'Karim', 1203, 'Dhaka','01912584949')
```

The second form specifies both the column names and the values to be inserted:

```
INSERT INTO table_name (column1, column2, column3,...)

VALUES (value1, value2, value3,...)
```

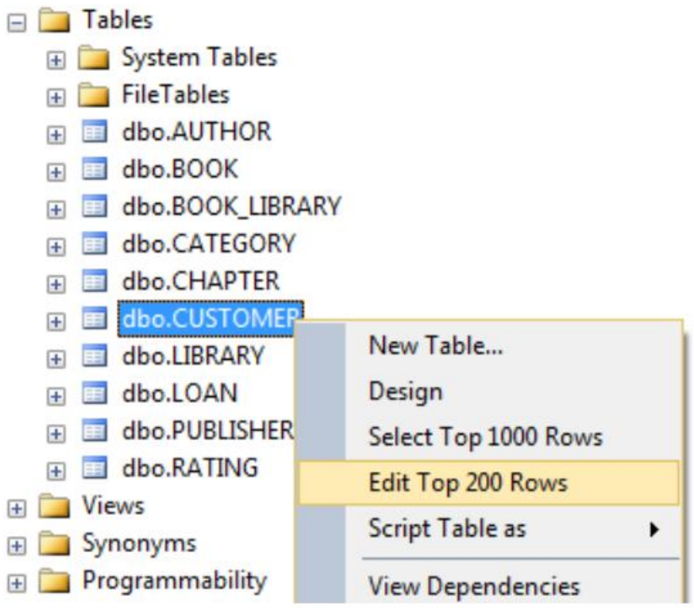
This form is recommended!

Insert Data Only in Specified Columns:

It is also possible to only add data in specific columns.

Insert Data in the Designer Tools:

When you have created the tables you can easily insert data into them using the designer tools. Right-click on the specific table and select "Edit Top 200 Rows":



Then you can enter data in a table format, similar to, e.g., MS Excel:

5 ALTER TABLE

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

To add a column in a table, use the following syntax:

```
ALTER TABLE table_name
ADD column_name datatype
```

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

```
ALTER TABLE table_name
DROP COLUMN column_name
```

To change the data type of a column in a table, use the following syntax:

```
ALTER TABLE table_name
ALTER COLUMN column_name datatype
```

6 SQL Constraints

Constraints can be specified when a table is created (with the CREATE TABLE statement) or after the table is created (with the ALTER TABLE statement).

Here are the most important constraints:

- PRIMARY KEY
- NOT NULL
- UNIQUE
- FOREIGN KEY
- CHECK
- DEFAULT
- IDENTITY

In the sections below we will explain some of these in detail.

6.1 PRIMARY KEY

The PRIMARY KEY constraint uniquely identifies each record in a database table.

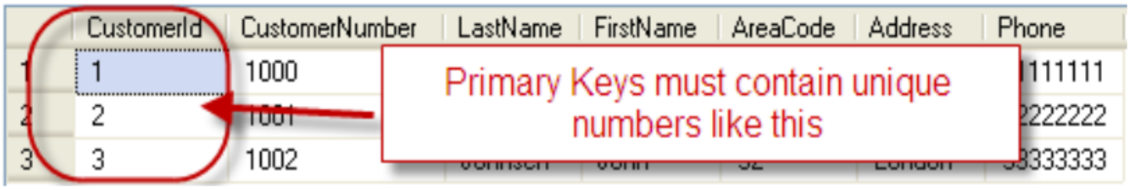
Primary keys must contain unique values. It is normal to just use running numbers, like 1, 2, 3, 4, 5, … as values in Primary Key column. It is a good idea to let the system handle this for you by specifying that the Primary Key should be set to **identity(1,1)**. IDENTITY(1,1) means the first value will be 1 and then it will increment by 1.

Each table should have a primary key, and each table can have only ONE primary key.

If we take a closer look at the CUSTOMER table created earlier:

```
CREATE TABLE CUSTOMER
(
        CustomerId int IDENTITY(1,1) PRIMARY KEY,
        LastName varchar(50) NOT NULL,
        FirstName varchar(50) NOT NULL,
        AreaCode int NULL,
        Address varchar(200) NULL,
        Phone varchar(11) NULL,
)
```

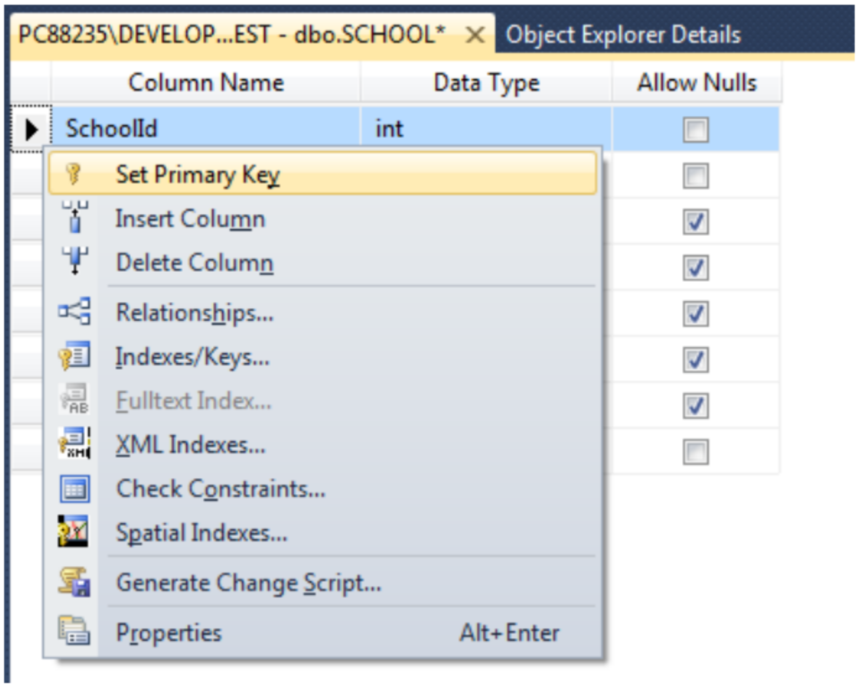
As you see we use the "Primary Key" keyword to specify that a column should be the Primary Key.



	CustomerId	CustomerNumber	LastName	FirstName	AreaCode	Address	Phone
1	1	1000	[illegible]	[illegible]	[illegible]	[illegible]	1111111
2	2	1001	[illegible]	[illegible]	[illegible]	[illegible]	2222222
3	3	1002	Johnsen	John	32	London	33333333

<u>Setting Primary Keys in the Designer Tools:</u>

If you use the Designer tools in SQL Server you can easily set the primary Key in a table just by right-click and select "Set primary Key".



The primary Key column will then have a small key in front to illustrate that this column is a Primary Key.

6.2 AUTO INCREMENT or IDENTITY

Very often we would like the value of the primary key field to be created automatically every time a new record is inserted.

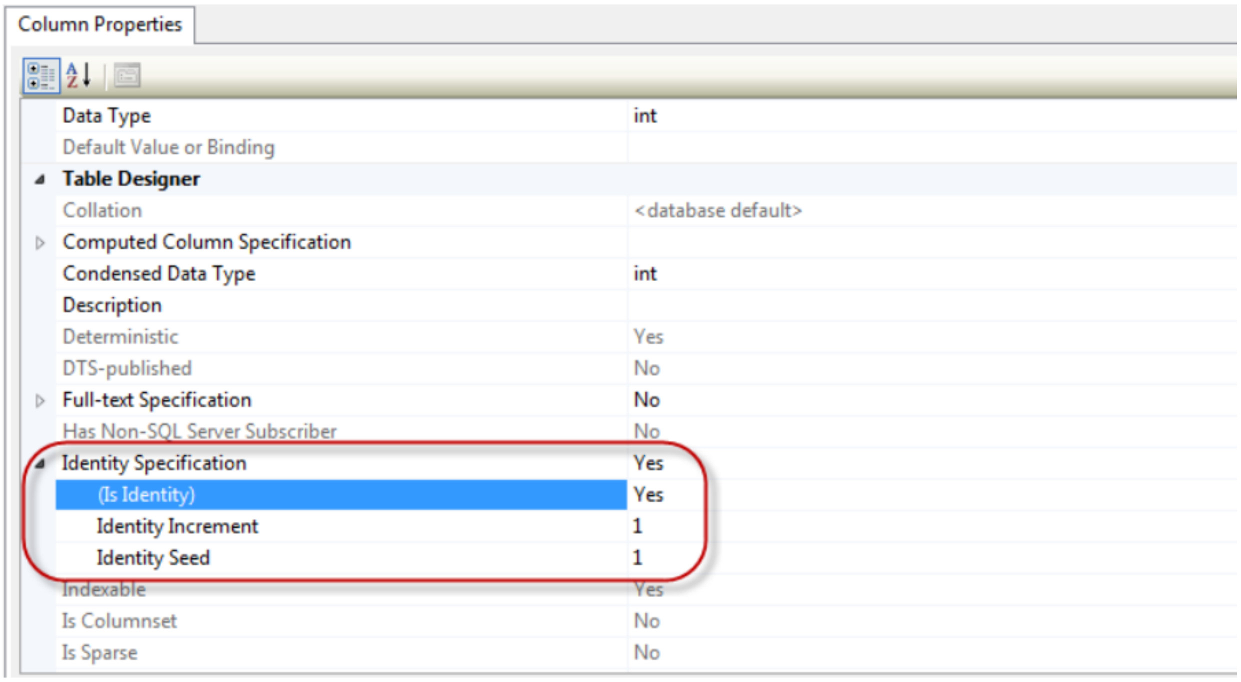
```
CREATE TABLE CUSTOMER
(
        CustomerId int IDENTITY(1,1) PRIMARY KEY,
        LastName varchar(50) NOT NULL,
        FirstName varchar(50) NOT NULL,
        AreaCode int NULL,
        Address varchar(200) NULL,
        Phone varchar(11) NULL,
)
```

As shown below, we use the IDENTITY () for this. IDENTITY (1, 1) means the first value will be 1 and then it will increment by 1.

<u>Setting identity(1,1) in the Designer Tools:</u>

We can use the designer tools to specify that a Primary Key should be an identity colu
that is automatically generated by the system when we insert data in to the table.

Click on the column in the designer and go into the Column Properties window:



Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Is Columnset	No
Is Sparse	No