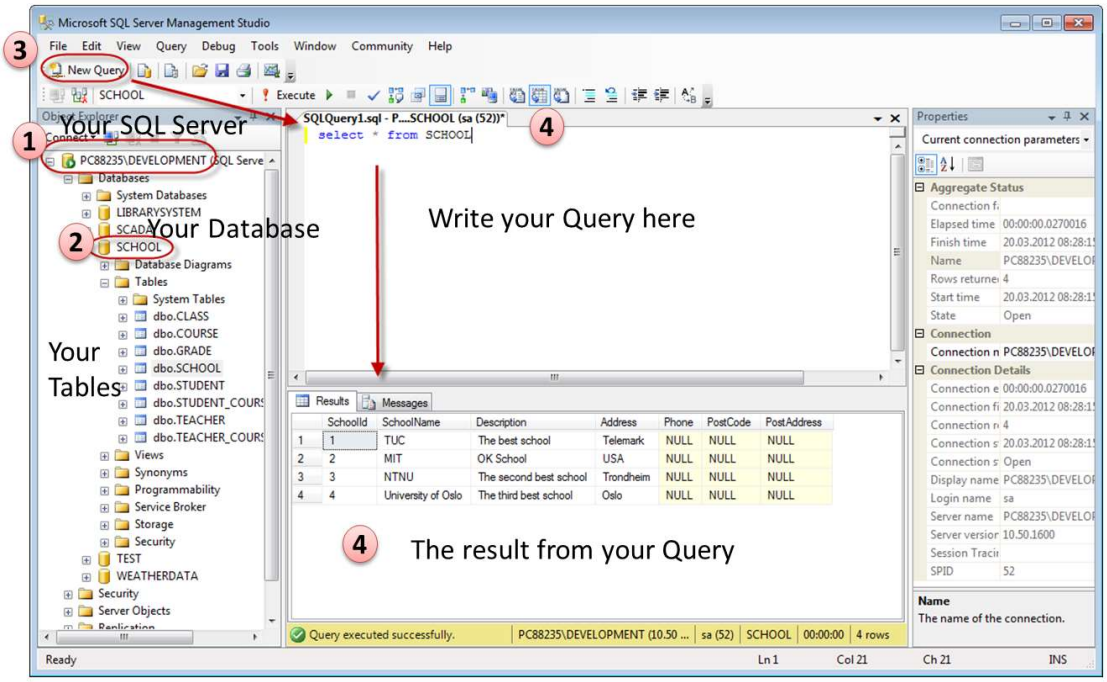
2.1 SQL Server Management Studio

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.

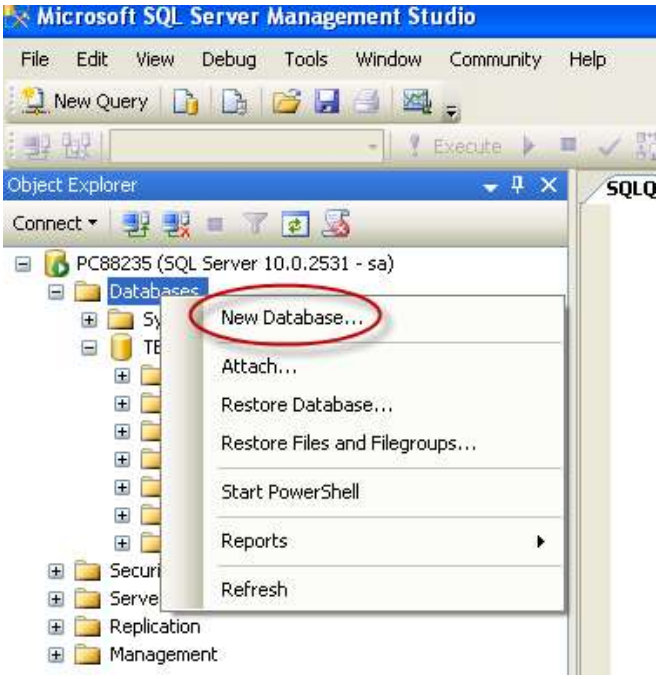


When creating SQL commands and queries, the “Query Editor” (select “New Query” from the Toolbar) is used (shown in the figure above).

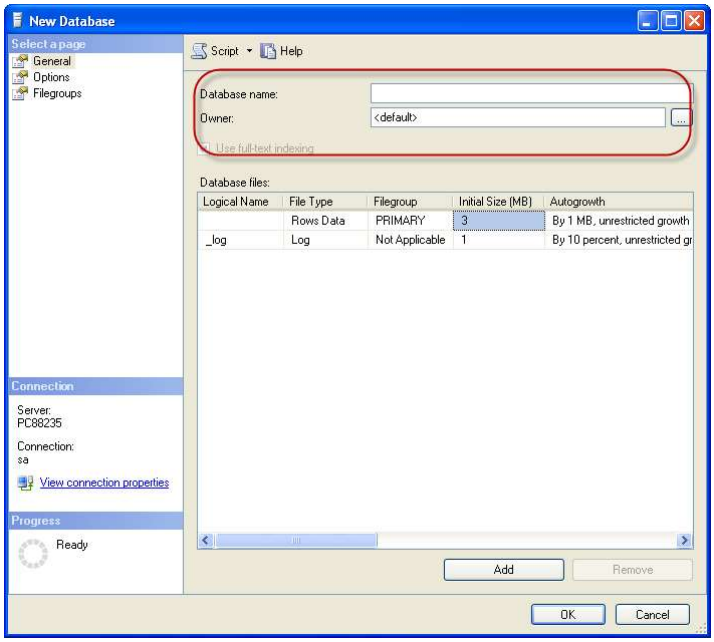
With SQL and the “Query Editor” we can do almost everything with code, but sometimes it is also a good idea to use the different Designer tools in SQL to help us do the work without coding (so much).

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…"



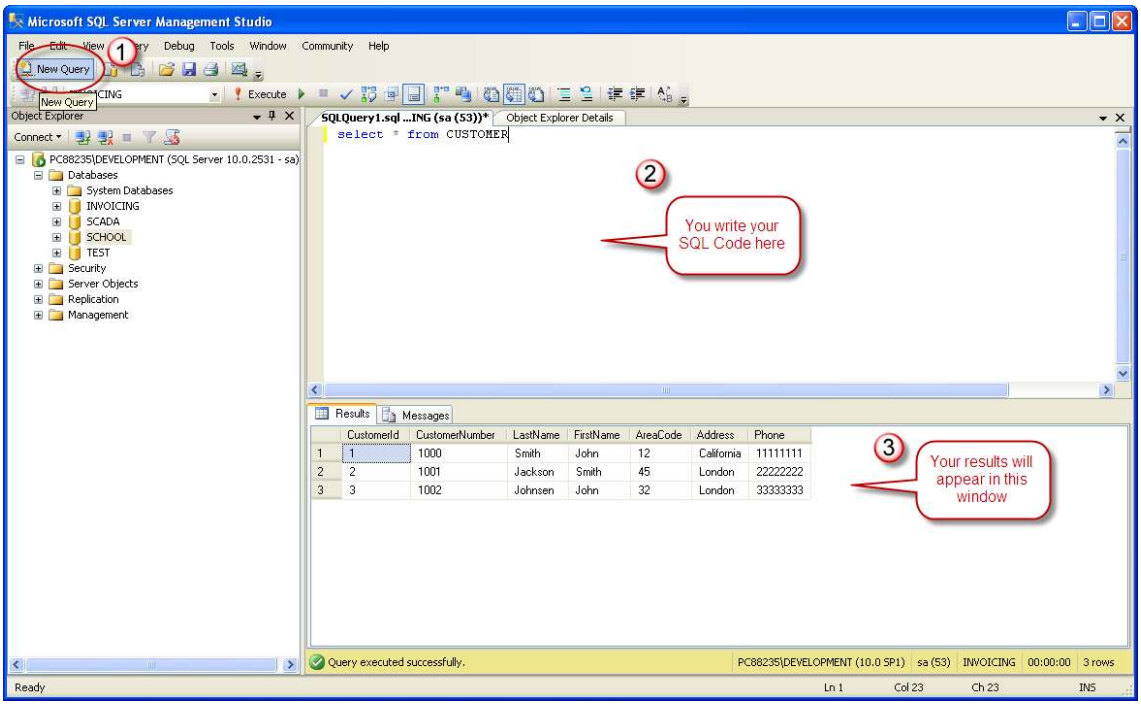
There are lots of settings you may set regarding your database, but the only information you must fill in is the name of your database:



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.

2.1.2 Queries

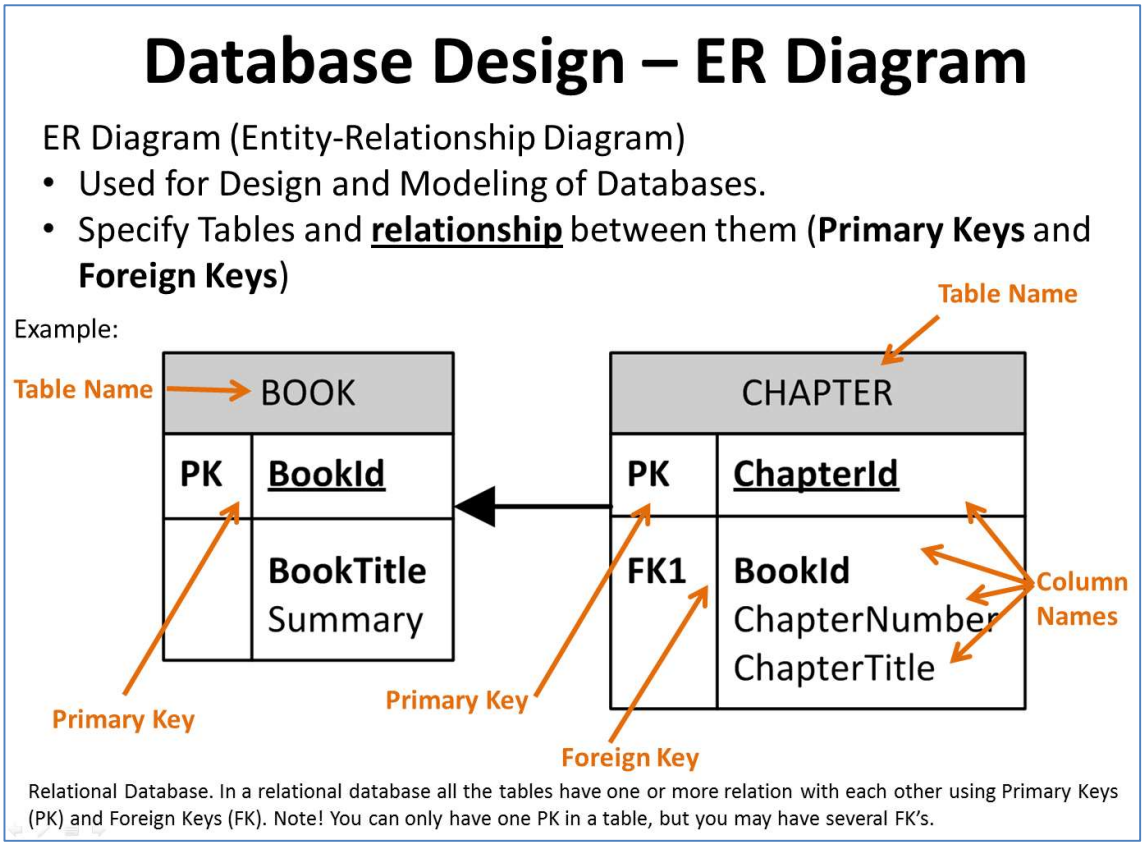
In order to make a new SQL query, select the “New Query” button from the Toolbar.



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

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.



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	☐
CustomerNumber	int	☐
LastName	varchar(50)	☐
FirstName	varchar(50)	☐
AreaCode	int	☑
Address	varchar(50)	☑
Phone	varchar(20)	☑
		☐

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

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.

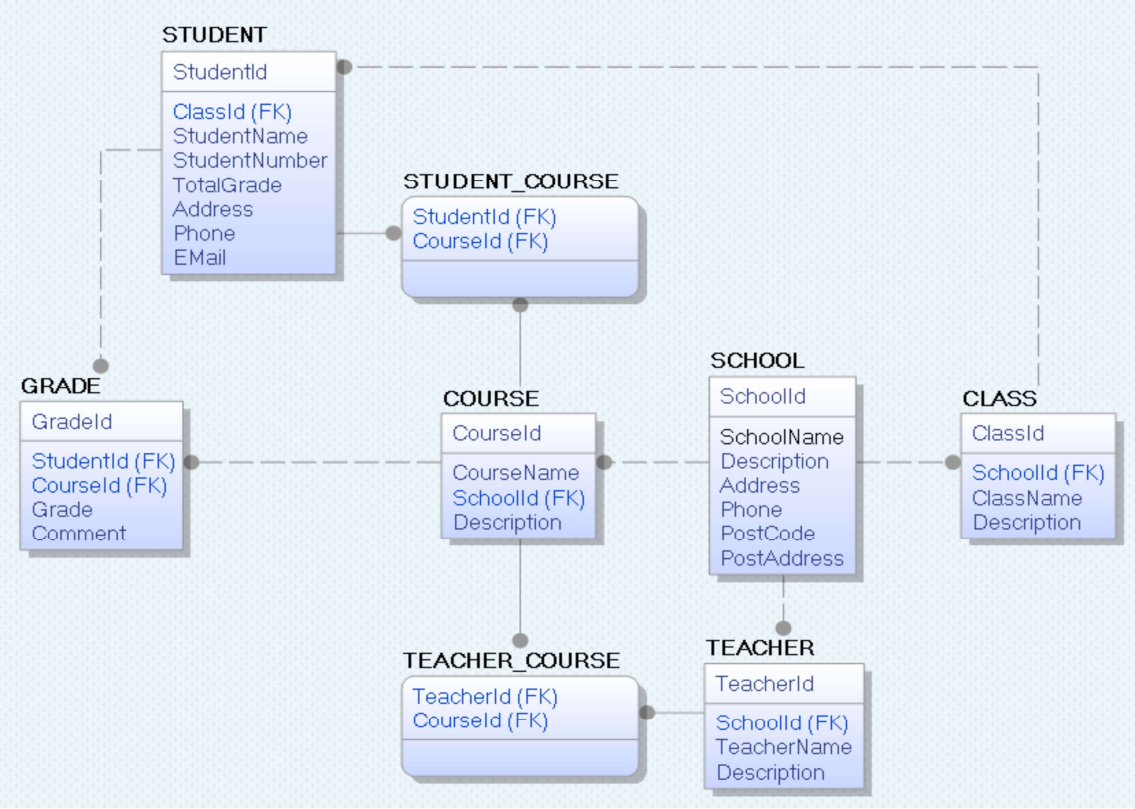
- “Always” use Integer and Identity(1,1) for Primary Keys. Use UNIQUE constraint for other columns that needs to be unique, e.g. RoomNumber

- 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, ...)

3.1 Database Modelling

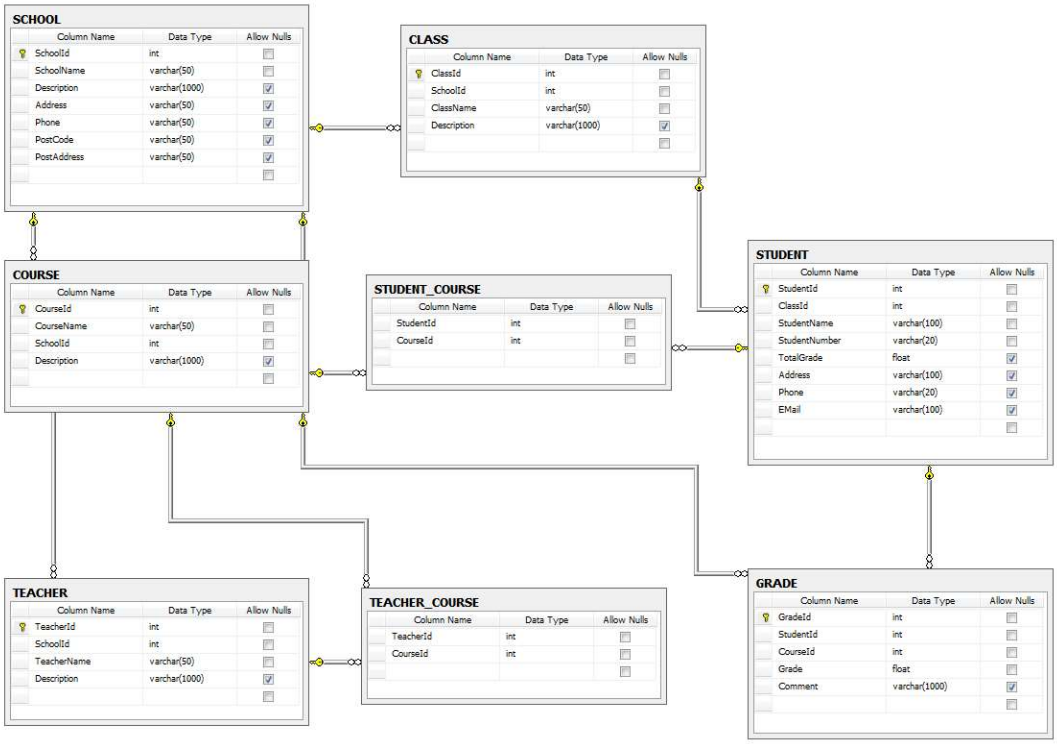
As mention in the beginning of the chapter, you should always start with database modelling before you start implementing the tables in a database system.

Below we see a database model in created with ERwin.



With this tool we can transfer the database model as tables into different database systems, such as e.g., SQL Server. CA ERwin Data Modeler Community Edition is free with a 25 objects limit. It has support for Oracle, SQL Server, MySQL, ODBC and Sybase.

Below we see the same tables inside the design tool in SQL Server.



Microsoft SQL Server – Tips and Tricks