## Exercise 1 Reverse Engineering

*In this lab you will be using the School database that is installed on your system.*

### Exercise 1.1 Install Packages

1. Create a new Console Application
2. Install nugget packages
   1. Microsoft.EntityFrameworkCore
   2. Microsoft.EntityFrameworkCore.SqlServer
   3. Microsoft.EntityFrameworkCore.Tools
   4. Microsoft.EntityFrameworkCore.SqlServer.Design

### Exercise 1.2 Create Model

1. Use Scaffold-DbContext to create your model from the existing School database. The generated classes must be generated in a DAL-folder. Only include the entities **Course, OnlineCourse, OnsiteCourse, Person and Department**. Use Get-Help Scaffold-DbContext –full for the required command.

## Exercise 2. LINQ

When performing a query check the query that is sent to the database, by using logging to the console.

1. Create and define a LINQuery to select only the first name and last name of all the Students. A student is a person that has an enrollmentdate, but no hiredate assigned.
2. Create and define a LINQuery to select only the first name and last name of all the Students and display only three students at a time. Every group of three students is a new query.
3. Create and define a LINQuery to select all the Students whose last name starts with an “A”;
4. Create and define a LINQuery to select only the first name and last name of all the Students sorted by last name.
5. Create and define a LINQuery to select all the courses. The output should include the department to which the course belongs. For now use the Include extension method to access the related table ( *context.Course.Include(t => t.Department)* )
6. Create and define a LINQuery to select all courses, grouped by department.
7. Create and define a LINQuery to select all departments that offer more than two courses.
8. Create and define a LINQuery to select all on site courses. The output should include the title of the course, the location of the course and the department that the course belongs to.