Lesson 8, Questions

In some of the questions there are references to Tom Dykstra and Rick Anderson: *Getting Started with Entity Framework 6 Code First using MVC 5*. You'll find the book at Fronter, or can download it from http://www.asp.net/mvc/tutorials/getting-started-with-ef-using-mvc/creating-an-entity-framework-data-model-for-an-asp-net-mvc-application.

1.

When you use the code-first approach to create entities inside the *Entity Framework* (EF) there are naming conventions you can follow to make life easier.

- What is the naming convention for tables in the database?
- What is the naming convention for columns?
- How should you name a property to make it *primary key*?
- What are the naming conventions for *foreign key* properties?

2.

Navigation properties

- What is a navigation property?
- Explain the navigation properties at the entity diagram displayed at Dykstra and Anderson, p. 18.
- Navigation properties are typically defined as `virtual`. Please explain why.
- A navigation property that holds multiple entities must be a list. Which *type* must that list be?

3.

Override conventions

- How do you specify a property that do not follow the conventions as *primary key*?
- How do you specify a property that do not follow the conventions as *foreign key*?
- Normally, the database automatically generates primary key values as you insert new rows. How can you instruct EF to avoid this for specific primary key attributes?

4.

Please read the code below and explain it in details:

```
using ContosoUniversity.Models;
using System.Data.Entity;
using System.Data.Entity.ModelConfiguration.Conventions;
```

```
namespace ContosoUniversity.DAL {

   public class SchoolContext : DbContext {
      public SchoolContext() : base("SchoolContext") {
      }
      public DbSet<Student> Students { get; set; }
      public DbSet<Enrollment> Enrollments { get; set; }
      public DbSet<Course> Courses { get; set; }

      protected override void OnModelCreating(DbModelBuilder modelBuilder) {
         modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();
      }
   }
}
```

5. The `Seed` method in Initializer classes

- What is the purpose of the `Seed` method of the `SchoolInitializer` class (Dykstra and Anderson, p. 23-24)?
- How do you tell the EF to use the initializer class?
- When is the `Seed` method inside the initializer class called?
- What happens when the `Seed` method is called?
- Where should the seed method me located

6.

In the Contoso University project, you use LocalDB as database engine

- What is LocalDB?
- How do you specify the name and location of you LocalDB database file?
- What is the preferred location for LocalDB database files for web development?

7. Please explain this piece of xml:

```
<connectionStrings>
    <add name="SchoolContext"
        connectionString="Data Source=(localdb)\MSSQLLocalDB;
        Integrated Security=True;
        MultipleActiveResultSets=True;
        AttachDbFilename=|DataDirectory|ContosoUniversity.mdf“
        providerName="System.Data.SqlClient" />
</connectionStrings>
```

8.

On p. 27 in Dykstra and Anderson a new StudentController is generated based on a MVC scaffolding item called **MVC 5 Controller with views, using Entity Framework**

- Which files and methods does the scaffolding code generate
- Please explain the action methods inside the `StudentController`
- Explain the code inside the `Student/Index.cshtml` view

9.

- Look at the code in Dykstra and Anderson p. 161.
- What does *lazy loading* mean?
- What is *eager loading*?
- When do you think it is appropriate to use *lazy loading*?

10.

- Read the `Create` method code in Dykstra and Anderson p. 42 carefully and make sure you understand each line.
- Which precautions are taken to make the code more secure?
- How does the `Edit` method p. 47 differ from the `Create` method?