# Homework: Entity Framework Code First

This document defines the homework assignments from the ["Database Applications" Course @ Software University](https://softuni.bg/trainings/21/Database-Applications-Mar-2015). Please submit as homework a single zip / rar / 7z archive holding the solutions (source code) of all below described problems.

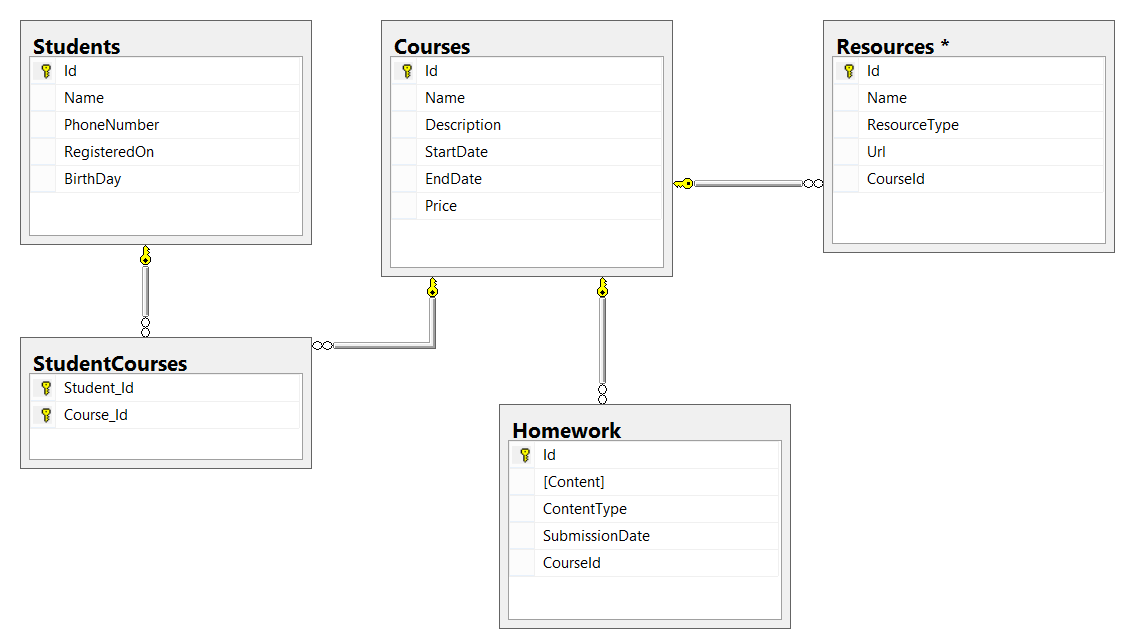
## Code First Student System

Your task is to create a database for the **Student System**, using the **Entity Framework Code First** approach. Model the following tables:

* **Students**: id, name, phone number (optional), registration date, birthday (optional)
* **Courses**: id, name, description (optional), start date, end date, price
* **Resources**: id, name, type of resource (video / presentation / document / other), URL
* **Homework**: id, content, content-type (e.g. application/pdf or application/zip), submission date

Table relations:

* **Students** can be in **many course**s
* **Courses** can have **many students**
* **Courses** can have **many resources**
* **One course** can have **many homework submissions**
* **Homework submissions** have a **student**



Add **navigational properties** in all models to simplify navigation. Annotate the data models with the appropriate **attributes** and validations and **enable code first migrations**.

Separate the **models**, **data** and **client** into **different layers** (projects).

## Seed Some Data in the Database

Write a **seed method** that fills the database with sample data (randomly generated).

Fill a few **students**, **courses**, **resources** and **homework submissions**. Configure Entity Framework to run the **Seed()** method after the database is created for the first time (i.e. only if it's empty).

Run the application several times to ensure that it seeds sample data **only** **once**.

## Working with the Database

Write a console application that works with the EF data layer and performs the following CRUD operations:

1. Lists **all students** and their **homework submissions**. Select only their **names** and for each homework - **content** and **content-type**.
2. List **all courses** with their corresponding **resources**. Select the **course name** and **description** and everything for each **resource**. Order the courses by start date (ascending), then by end date (descending).
3. List **all courses** with **more than 5 resources**. Order them by **resources count** (descending), then by **start date** (descending). Select only the **course name** and the **resource count**.
4. List all **courses** which were active on a **given date** (choose the date depending on the data seeded to ensure there are results), and for each course count the **number of students enrolled**.   
   Select the **course name**, **start** and **end date**, **course duration** (difference between end and start date) and **number of students enrolled**. Order the results by the **number of students** enrolled (in descending order), then by **duration** (descending).
5. For each student, calculate the **number of courses** he/she has enrolled in, the **total price** of these courses and the **average price per course** for the student.  
   Select the **student name**, **number of courses**, **total price** and **average price**. Order the results by **total price** (descending), then by **number of courses** (descending) and then by the **student's name** (ascending).

## Resource Licenses

Resources should now have many **licenses**. A **license** should have an **Id** and **Name**.

Make these changes using Code First Migrations. Make sure no data is lost after the update.