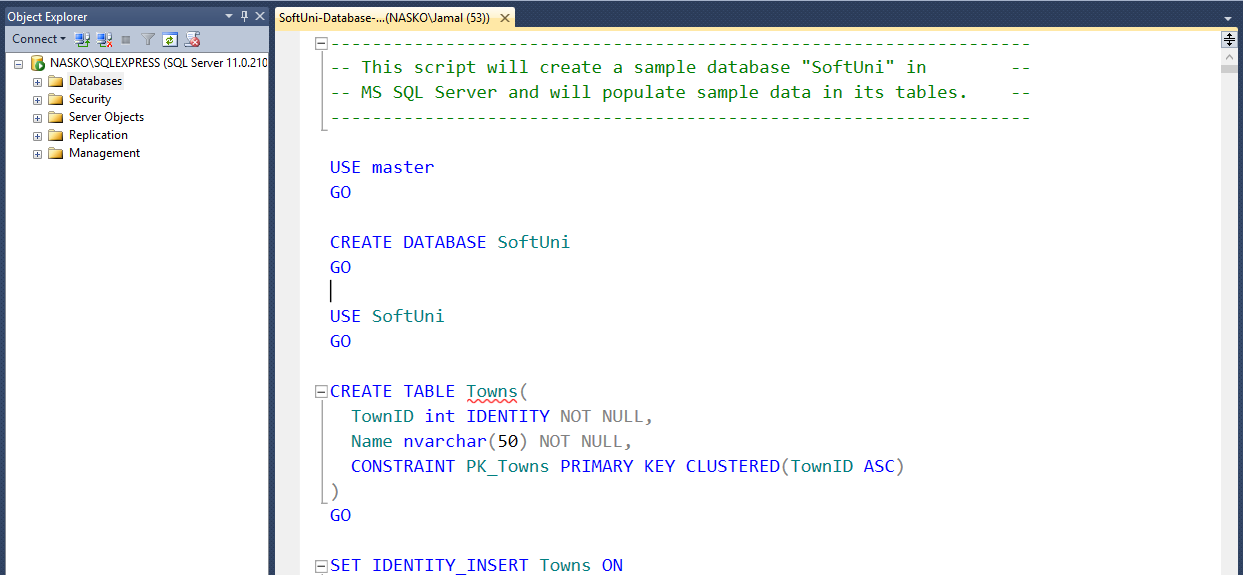
# Exercise: Entity Framework - Employees

This document defines an in-class exercise from the ["Database Applications" Course @ Software University](https://softuni.bg/trainings/21/Database-Applications-Mar-2015).

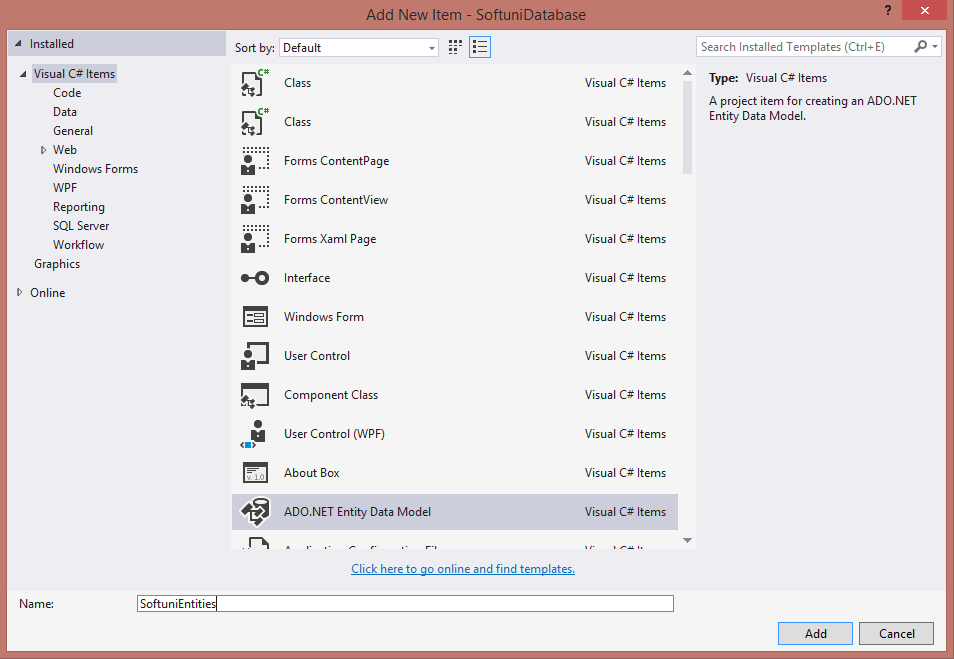
## Import the SoftUni database

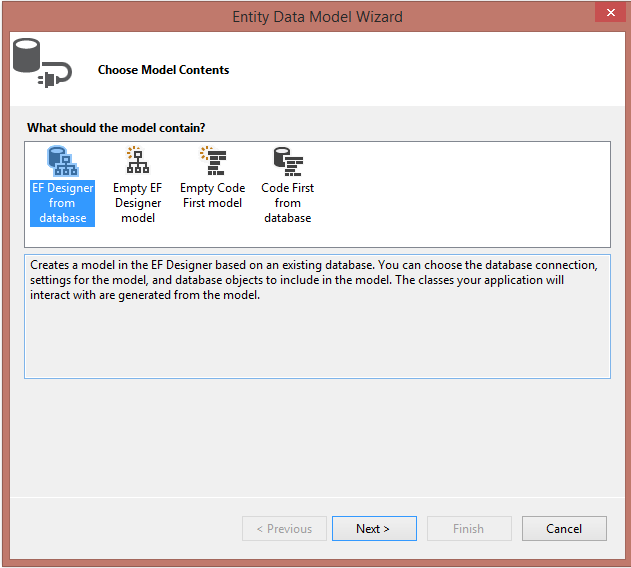
Import the SoftUni database into SQL Management Studio by **executing** the provided **.sql** script.



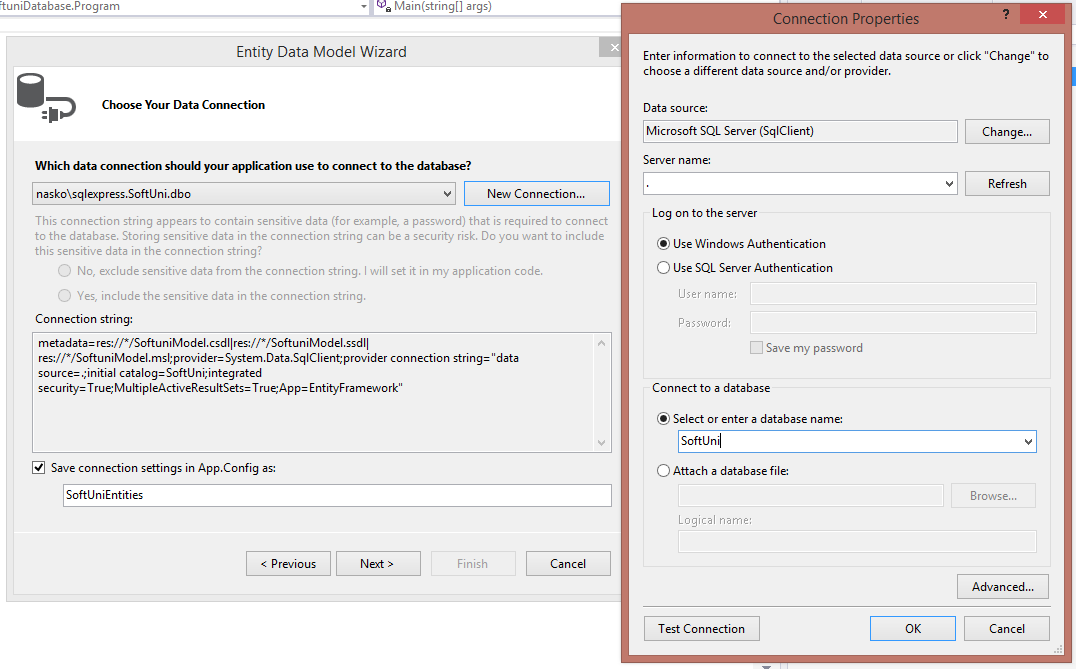
## Database First

Model the existing database by using Database First.

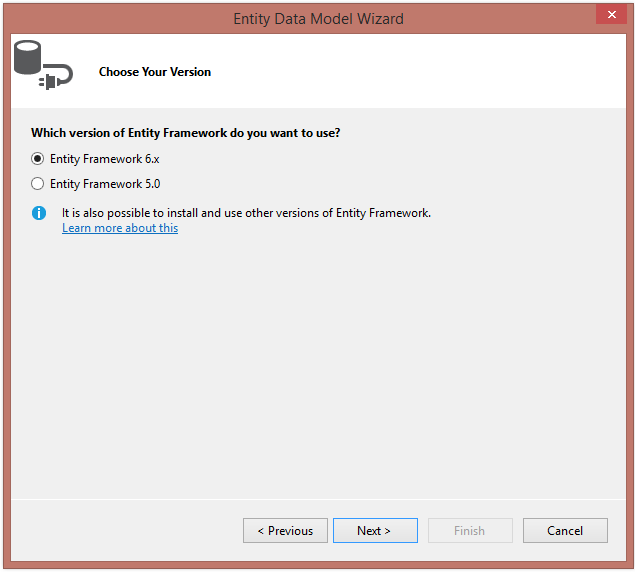




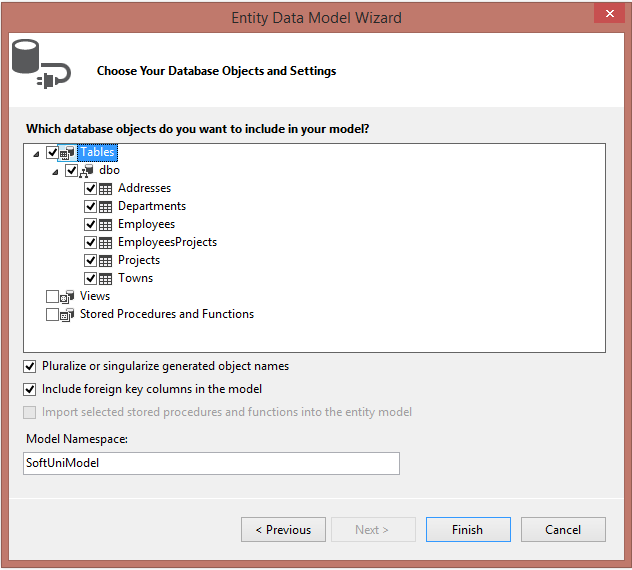
Choose your **server name** (SQL management studio connection) and the **database** you wish to model.



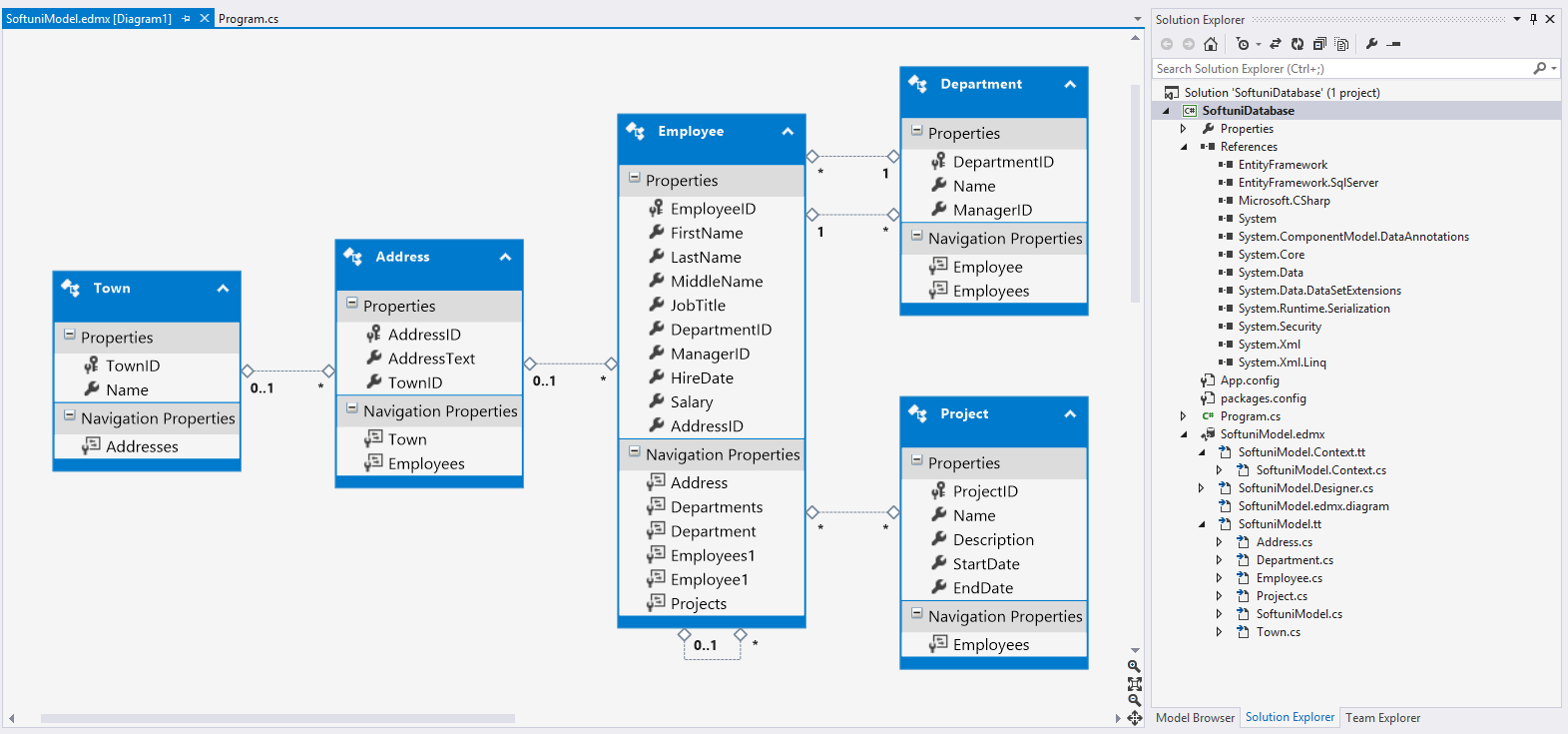
Select the latest stable Entitiy Framework version.

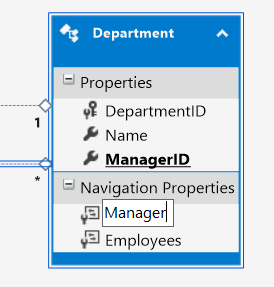


Select the tables the desired tables from the target database. Exclude any views and stored procedures.



Click finish. The result should be a generated **model diagram**, which resembles the **database schema**.

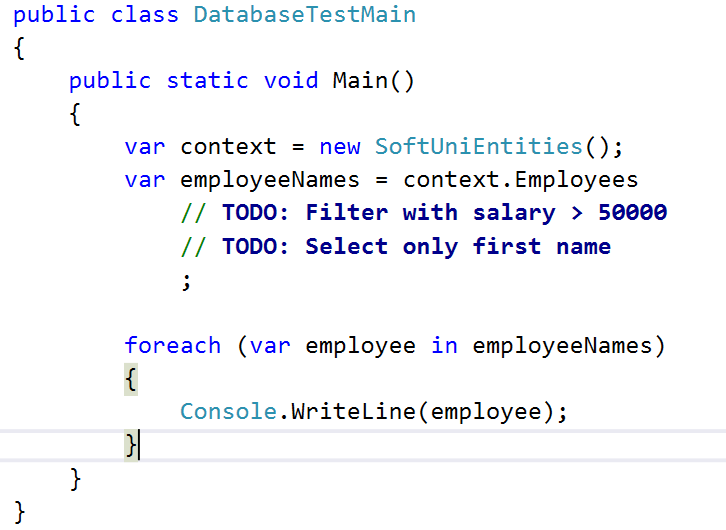


Entity Framework has successfully **mapped the database schema to C# classes**. However, it isn't quite good with names - e.g. **Employees1**, **Employee1**. Edit the diagram and give the properties more **proper names**.

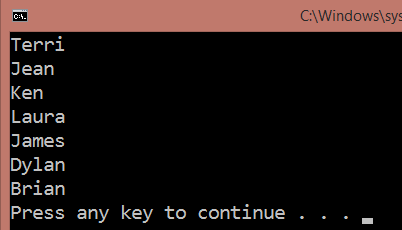
## Employee Queries

### Step 1 - Employees with Salary Over 50 000

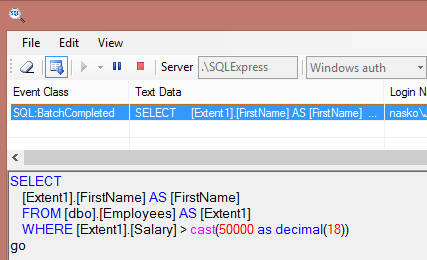
Let's start writing C# queries! Your first task is to extract **all employees** with **salary** over **50000**. Create a new context and use lambdas to build your query. Make sure the query returns **only the** **first names** of those employees.



Result on console:

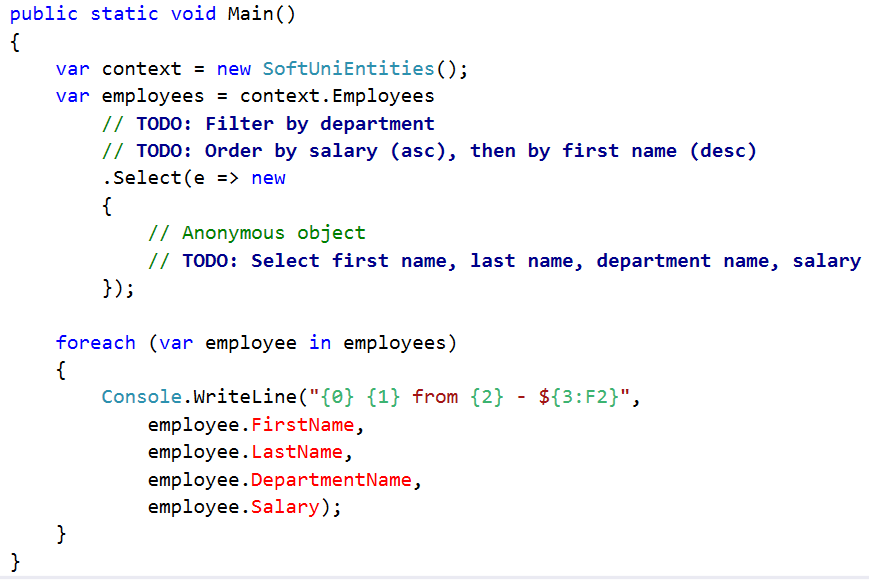


Use Express Profiler and check if the made query by Entity Framework is correct (there is only one query).

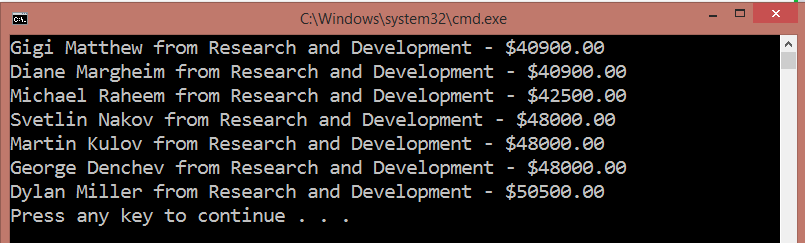


### Step 2 - Employees from Seattle

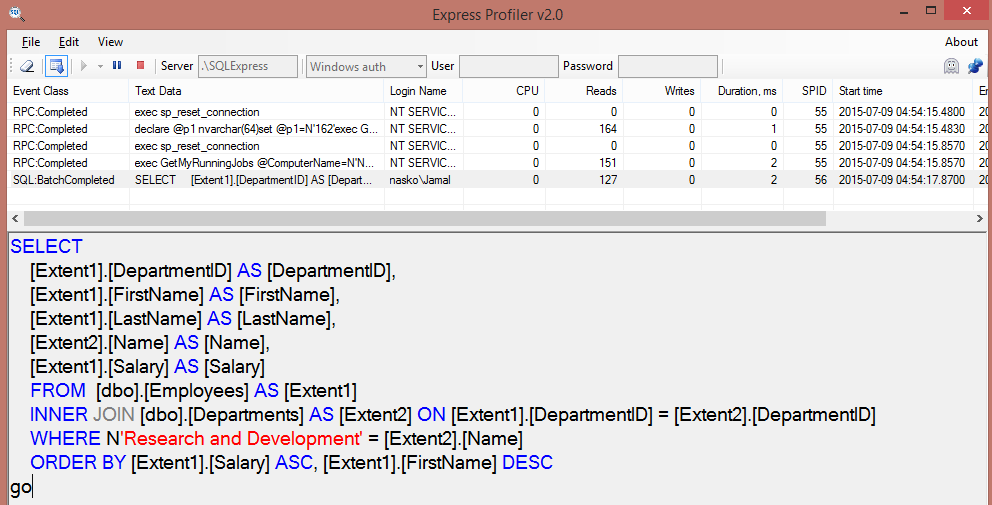
Extract all employees from the **Research and Development** department. Order them by **salary** (in ascending order), then by **first name** (in descending order). Select only their **first name**, **last name**, **department name** and **salary**.



Result on console:

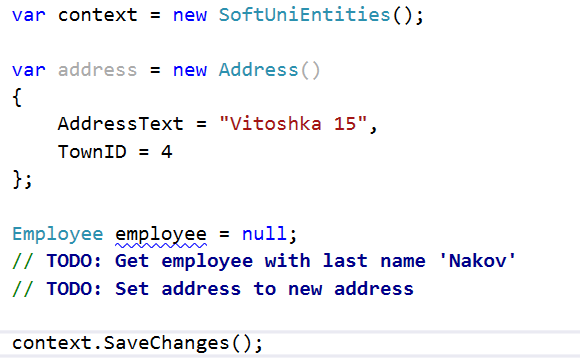


Use Express Profiler and check if the made query by Entity Framework is correct (there is only one query).

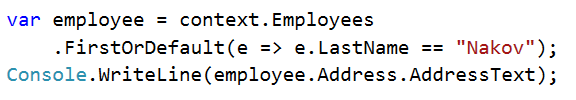


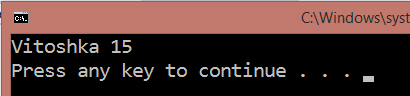
## Adding a New Address and Updating Employee

Create a new address with **text** "**Vitoshka 15**" and **TownId** **4**. Add the town to the



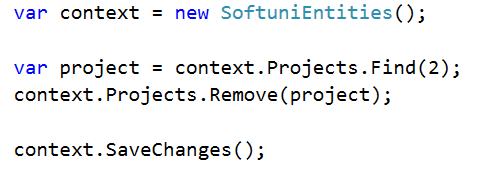
The above code should successfully **insert a new address** in the database and **set it as Nakov's new address**.





## Deleting Project by Id

Let's delete a project by id. Sounds simple:



However…



The project is **referenced** by the junction (many-to-many) table **EmployeesProjects**. Therefore we cannot safely delete it. First, we need to remove any references to that row in the **Projects** table.

This is done by removing the project from all employees who reference it. **Tip**: Get those employees from the **Employee** navigation property.

