Server side Web Development

Lecture 9

Db Queries and Mapping (2)

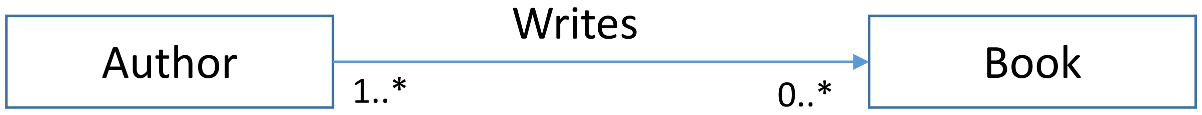
Outline

- Querying the database
  - EF and Many to many relationships
  - Loading schemes
- Mapping of Entity classes to the View model classes
  - Automapper and specifying mappings

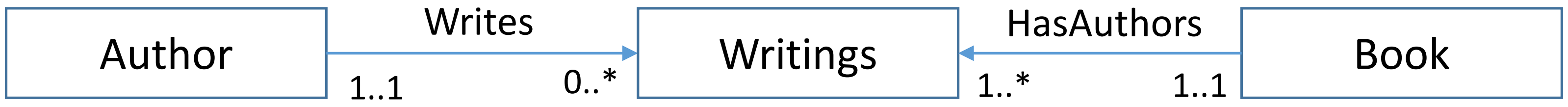
Many to many relationships and Database

- Many to many relationships between two relations
  - Is implemented in database with an intermediate relation

Konceptual model



Logical model



- It is the logical model which is implemented in the database

Many to many relationships and EF

- Using a database first approach
- EF interprets the logical schema in the database
- Logical model, without any other attributes than the foreign keys, then there is no need to create a class for the intermediate relation
  - EF manages querying the references
- Logical model, with additional attributes, then the intermediate relation need a class
  - References need to be queried manually

Loading schemes (referenced data)

- Basically three types
  - Lazy loading
  - Eager loading
  - Explicit loading

Lazy loading

- Lets EF automatically retrieve referenced data when it is needed
- DbContext.Configuration.LazyLoading is set to true default
- Requires that the entity class is
  - Public and not sealed
  - The navigation property must be set to *virtual*
- The downside is that if it might load a lot of data when it is not necessary

Eager loading

- The programmer tells EF when to load data
- Do not use virtual as navigation property of the entity class
- Uses the Include method to tell which referenbced data to load
- Downside: Often gives more complex queries and slower peformance

Explicit loading

- Similar to lazy loading that referenced data is loaded later when needed
- Do not use virtual as navigation property of the entity class
- Loaded when programmer wants it by using
  - Entry method
- Together with
  - Collection method
    - To find a collection of objects
  - Reference method
    - To find property data

Running queries using raw SQL

- Var db is the DbContext object

  ```
  db.Database.ExecuteSqlCommand([query as string])
  ```

Automapper – Prevent additional loading

- There are several options that can be added to a mapping
  - This is setup in the AutoMapperCongig.cs file for the profile of a mapping
- To mention some
  - ForMember
    - Can be used to map one properties that automapper will not automatically map, e.g. CreateMap<AUTHOR, Author>(). ForMember(m => m.p1, opt => opt.MapFrom(src => src.p2))
    - Can be used to prevent further mapping of a property, e.g. CreateMap<BOOK, Book>().ForMember(m => m.Authors, opt => opt.Ignore());
  - AfterMap [or BeforeMap]
    - Can be used to perform an operation either directly specified or by calling a method, e.g. CreateMap<AUTHOR, Author>().AfterMap((src, dest) => dest.noOfBooks = src.BOOKs.ToList().Count);
      (In this case there is a propery noOfBooks in the Author class, for which automapper preforms the calculation)

Reading

- Books:
  - Lerman, Julia “Programming Entity Framework: DbContext”, Fulltext biblioteket
- Links:
  - https://github.com/AutoMapper/AutoMapper/wiki