Infrastructure

Dominik Laso + Ondrej Pavlica

PV179

Summary of todays lecture

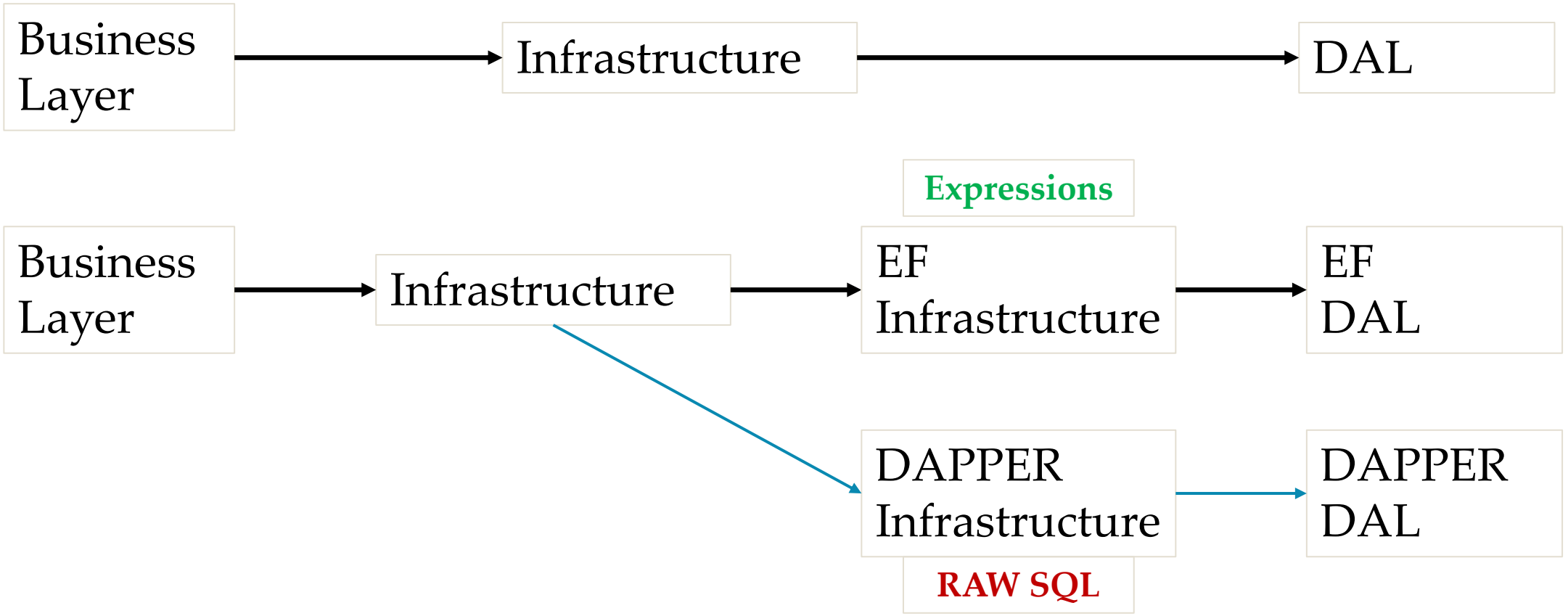
- What is Infrastructure.
- When do we want to have an infrastructure in our project.
- Revision of Repository and Unit Of Work.
  - Why do we want to move it to Infrastructure.
- Query:
  - What is it.
  - How do we implement it.
- Task – Finish the implementation of Query.
- Solution – How it could be finished.

Infrastructure

- What is it?
  - Think about the infrastructure as an abstraction on top of DAL.
  - It allows us to define methods in a generic way so that when we swap out the ORM, we can still call the same methods from BL.
- Why do we want it in our project?
  - Imagine a scenario of you having a system that consists of millions of lines of code.
  - Now, since we need our system to be a bit more efficient we have decided to switch to a different ORM.
  - Its going to be easier to create additional Infrastructure and DAL than rewrite the whole application ☺.

Infrastructure

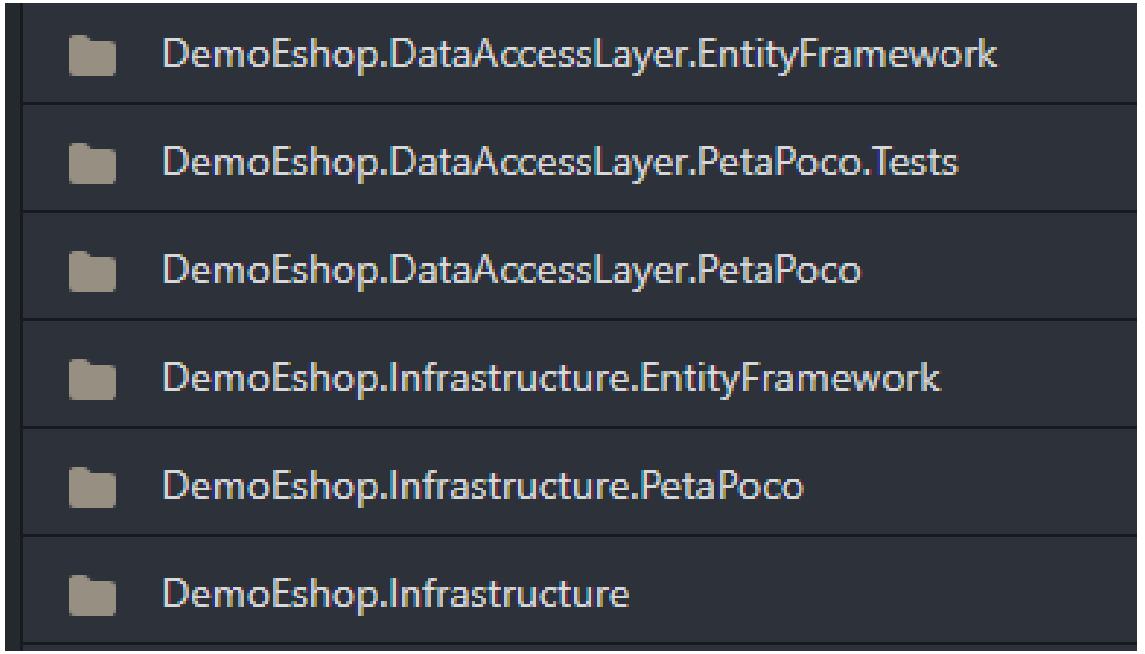
- It needs to be generic so other ORMs can convert it to what they need



Infrastructure

- How should a project with multiple infrastructure types look like?

- What should we have in Infrastructure?
  - Repository
  - Unit Of Work
  - Query



Query

- What is it?
  - Query is an “extension” of Repository. It consists of additional methods that are used for a different database operation than CRUD.
  - Some of these methods are:
    - Where – Filter out the products based on some expression.
    - Order By – Orders the query result.
    - Pagination – Limits the result for a specific range.
- How do we implement such thing?
  - Expression Trees + Reflection (today in task)
  - Build-up of a raw SQL. (in study materials as an example)

Query – example

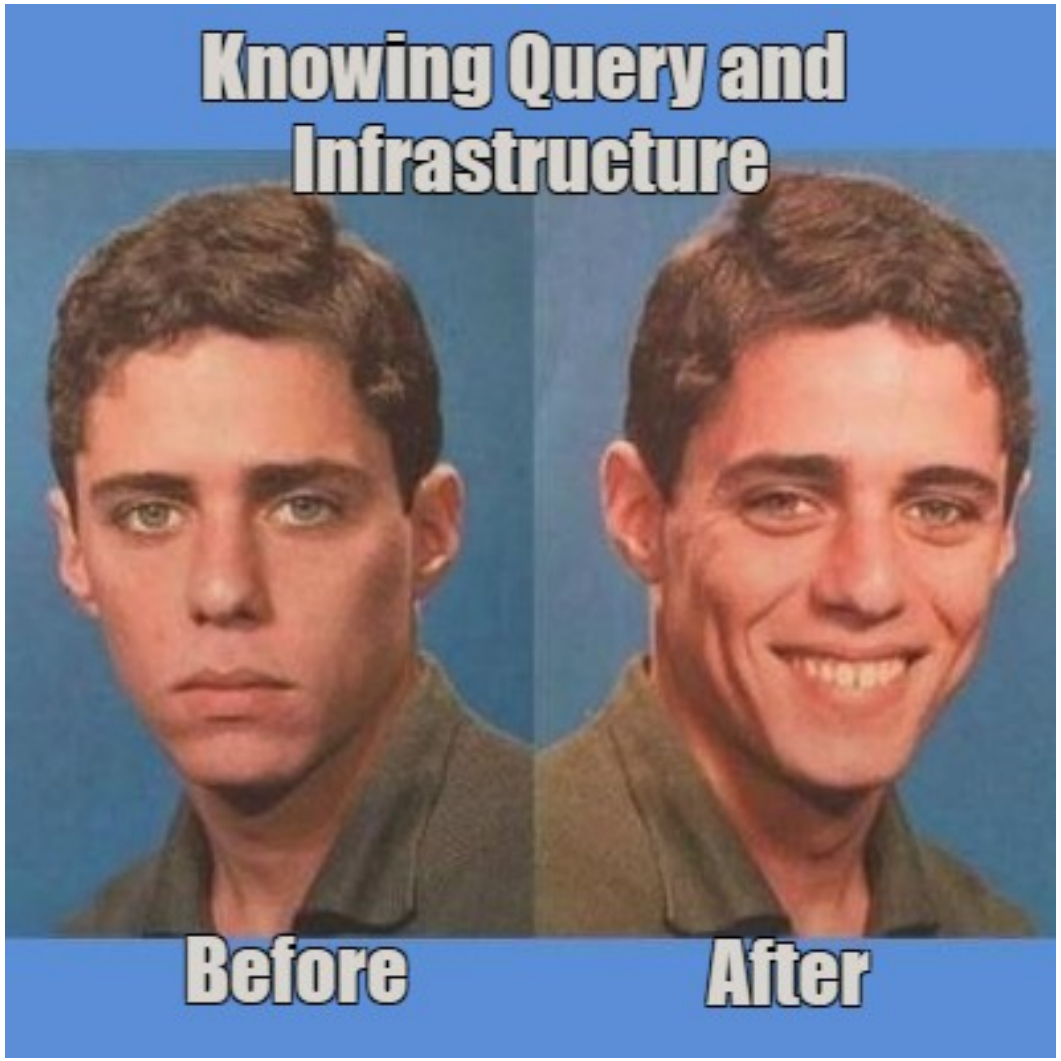
```
7 references
public interface IQuery<TEntity> where TEntity : class, IEntity, new()
{
    /// <summary>
    /// Adds a possiblity to filter the result
    /// </summary>
    5 references
    IQuery<TEntity> Where<T>(Expression<Func<T, bool>> rootPredicate, string columnName) where T : IComparable<T>;

    /// <summary>
    /// Adds a specified sort criteria to the query.
    /// </summary>
    4 references
    IQuery<TEntity> OrderBy<T>(string columnName, bool ascendingOrder = true) where T : IComparable<T>;

    /// <summary>
    /// Adds a posibility to paginate the result
    /// </summary>
    3 references
    IQuery<TEntity> Page(int pageToFetch, int pageSize = 10);

    /// <summary>
    /// Executes the query and returns the results.
    /// </summary>
    9 references
    IEnumerable<TEntity> Execute();
}
```

Questions about the infrastructure or query?



Task

- Implement 2 remaining query methods:
- Add tests for each method
- The full task description can be found in study materials

Thank you for your attention