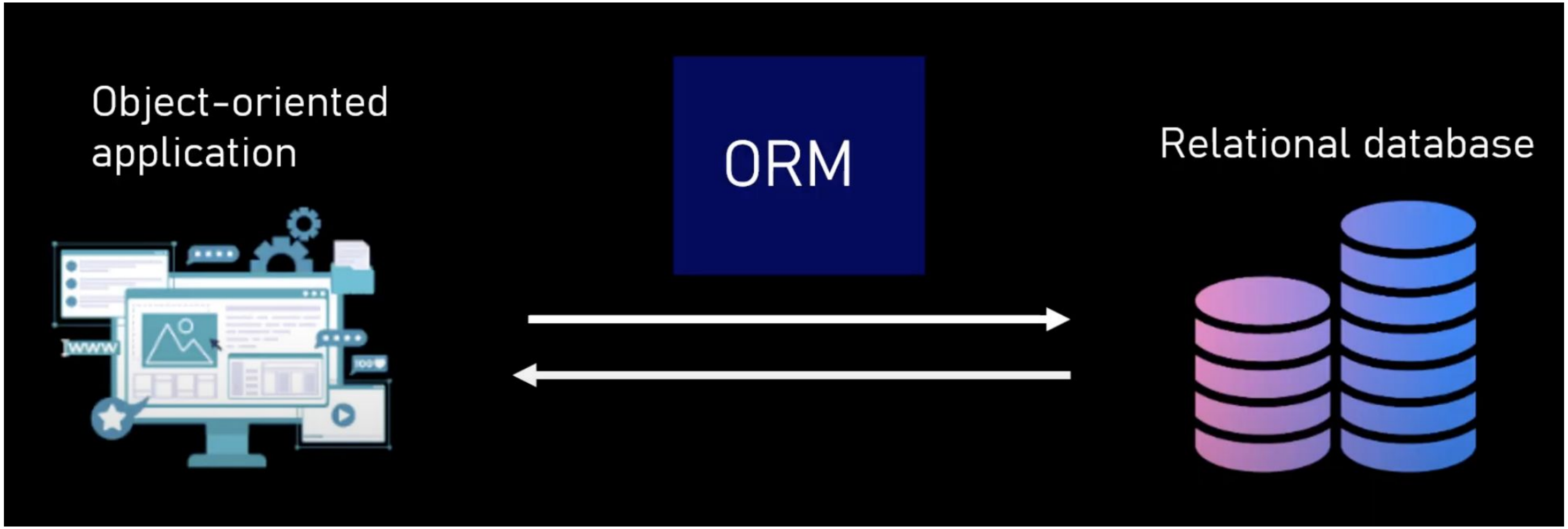
STEP
IT ACADEMY

Entity Framework Core

Instructor: Khouv Tannhuot
Phone: 087 42 47 36 (Telegram)

What is entity framework core?

- EF Core is an Object-Relational Mapper (ORM) framework which allows developers to work with database using C#.



Features of Entity Framework Core

- Lightweight
- Open-source
- Cross-platform

Advantages

It supports many database engines:

- SQL Server
- SQL Lite
- MySQL
- Oracle
- PostgreSQL

How to use EFCore?

- Create application of your choice
- Install the database of your choice
- Install database provider for that database
- Create model and database context class
- Write LINQ Queries to interact with database

Approaches to work with EFCore

- **Code-First Approach**

  We create the domain classes, later we create the database from our code

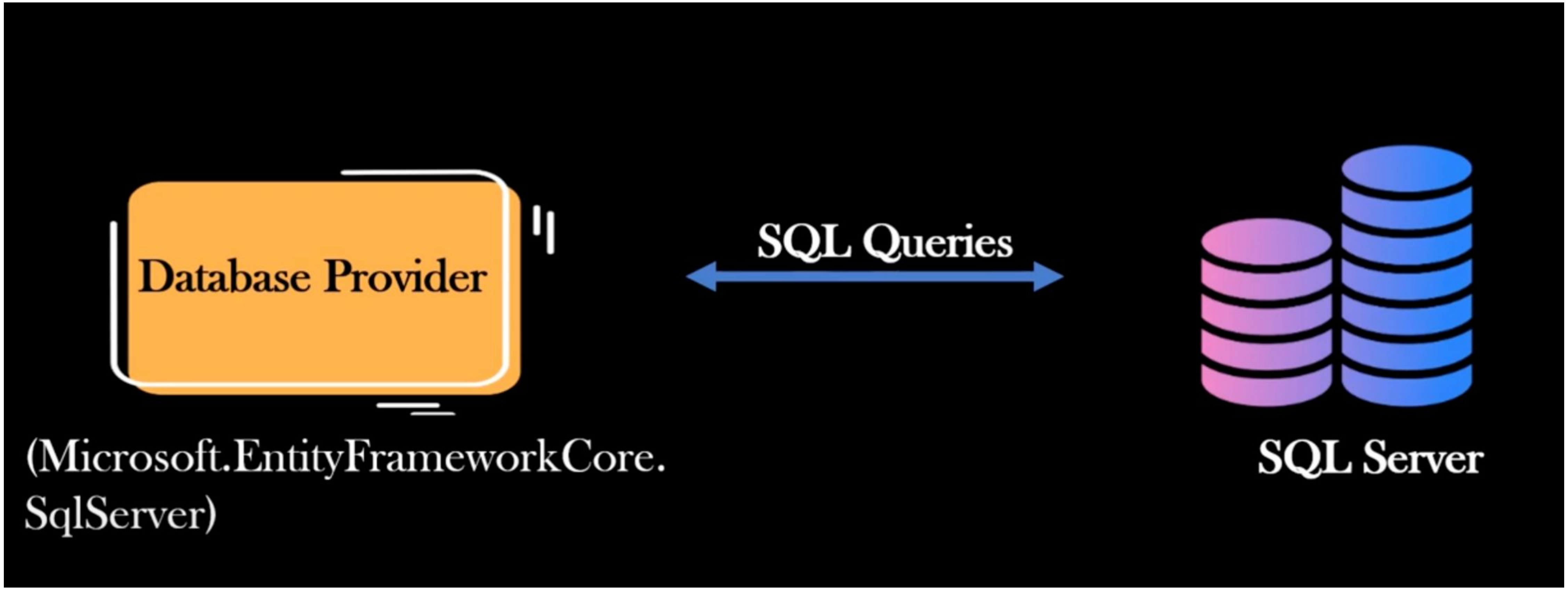
- **Database-First Approach**

  We design the database, later we create classes

Nuget Packages

- Microsoft.EntityFrameworkCore.SqlServer
- Microsoft.EntityFrameworkCore.Tools

What is Database Provider?



Why do we need different database providers?

- To be compatible with different database systems
- For performance optimization
- To support cross-platform compatibility

Note: For each database, database provider will be different.

What is Domain class and DBContext?

- **Domain class:** Model/Class that represents real-world entity
- **DBContext**: is a class that is used to interact with the database

Entity, DBSet in EF Core

- In the context of EF Core, entity is a class that maps to a database table.
- **DbSet** is a class that represent an entity set in a database

DbSet

DbSet allows you to perform various database operations

- Querying
- Inserting
- Updating
- Deleting

What is Migration

- Migration is an EF core feature that provides a way to incrementally update the database schema to keep it in sync with the application’s data model while preserving existing data in the database.

Database creation

- First we create Migration by using **Add-Migration** command.
- Then we create Migration by using **Update-Database** command.

Advantages of Migrations feature

- Migration allows us to version control our database schema changes.
- It allows us to rollback changes if needed.
- It allows us to apply migration without losing existing data.

Primary Key

- By default, EF Core will assume that a property named “Id” or “{className}Id” is the Primary key of an entity.
- You can use data annotations to explicitly specify the PK property using the [Key] attribute.
- You can also configure the primary key using the Fluent API in the OnModelCreating method of your DbContext class.

What is FluentAPI?

- A fluent API is an application programming interface
- It is designed to provide a more expressive and readable way of configuring an API
- It is implemented by using method chaining
- It is not limited to the EF Core
- In addition to EF Core, other frameworks and libraries also use fluent APIs for configuration

Other frameworks using FluentAPI

- NHibernate
- AutoMapper
- Asp.net core
- Asp.net WebApi
- FluentValidation

Method chaining

- It is programming technique that involves invoking multiple methods on an object in a single statement by chaining the method calls together
- it is commonly used in various programming languages and framework including libraries and APIs

EF core and Fluent API

- Creating and configuring model
- Configuring types
- Keys
- Relationships
- Indexes

How to add data to Database?

1. Create instances of your model classes
2. Add them to the appropriate DbSet property in your DbContext class
3. Call the SaveChanges method

How to update data?

1. Retrieve the entity
2. Update the necessary properties of retrieved entity
3. save the change by calling SaveChanges method

How to delete data?

1. Retrieve the entity
2. Delete the entity using the Remove() method
3. Save the change by calling the SaveChanges method

Different ways to load related data

- Eager loading
- Lazy loading
- Explicit loading

What is eager loading?

- Eager loading is a technique used to load related entities along with the main entity being queried in a single database round-trip.
- Eager loading can be implemented in EF core using the Include method or the ThenInclude method.

Include and Theninclude method

- Include method allows you to specify which navigation properties to load, and the ThenInclude method enables you to include further related entities.

Advantages of eager loading

- **Improved performance**
  - It can reduce the number of database round-trips required to retrieve related data.
- **Minimized network traffic**
  - Instead of sending multiple requests to the database for retrieving related entities, you can fetch them all in a single query

Disadvantages of eager loading

- **Increased Data Transfer Size**
  - This may result in a larger amount of data being transferred between the database and the application.
- **Over-fetching of Data**
  - It can result in unnecessary data being fetched from the database, wasting resources and impacting performance.

How to implement explicit loading?

To implement explicit loading, you can use the Entry method of the DbContext class along with the Collection or Reference methods to explicitly load related entities.

Reference and Collection methods

- If you want to explicitly load a related single entity (one-to-one or many-to-one relationship), you can use the Reference method else you can use Collection method.

Advantages of Explicit loading

- Improved performance
  - It allows you to load related entities on demand, reducing the amount of data retrieved from the database.
- Reduced memory usage
  - By selectively loading related entities when needed, you can conserve memory resources

Disadvantages of explicit loading

- Increased complexity
  - We need to explicitly manage the loading of related entities
- Additional queries
  - If not written properly, this may result in increased number of database queries which impacts performance

What is Lazy loading?

- In case of Lazy loading related data is loaded from the database when the navigation property is accessed.

Different ways to implement lazy loading

- Lazy loading with proxies
- Lazy loading without proxies

How to implement Lazy loading?

- Install the Nuget package
  - Microsoft.EntityFrameworkCore.Proxies
  - Call the method UseLazyLoadingProxies()

Note: Entity framework core will enable lazy loading for any navigation property that is Virtual.

Advantages of Lazy loading

- Simplified Development
  - Simplifies our code by automatically loading related entities when they are accessed.
- Reduced Memory Usage
  - Related entities are loaded only when accessed.

Disadvantages of Lazy loading

- Performance overhead
  - Each lazy loading request requires additional round trip to the database.