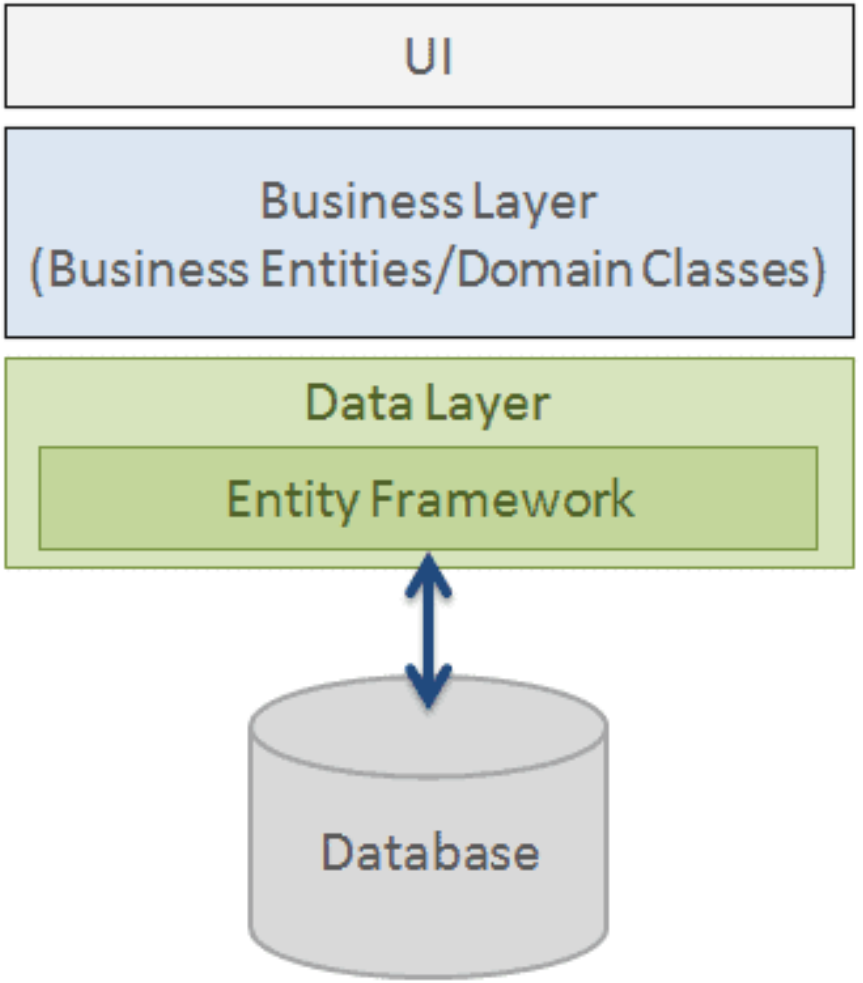
Dell IT Academy

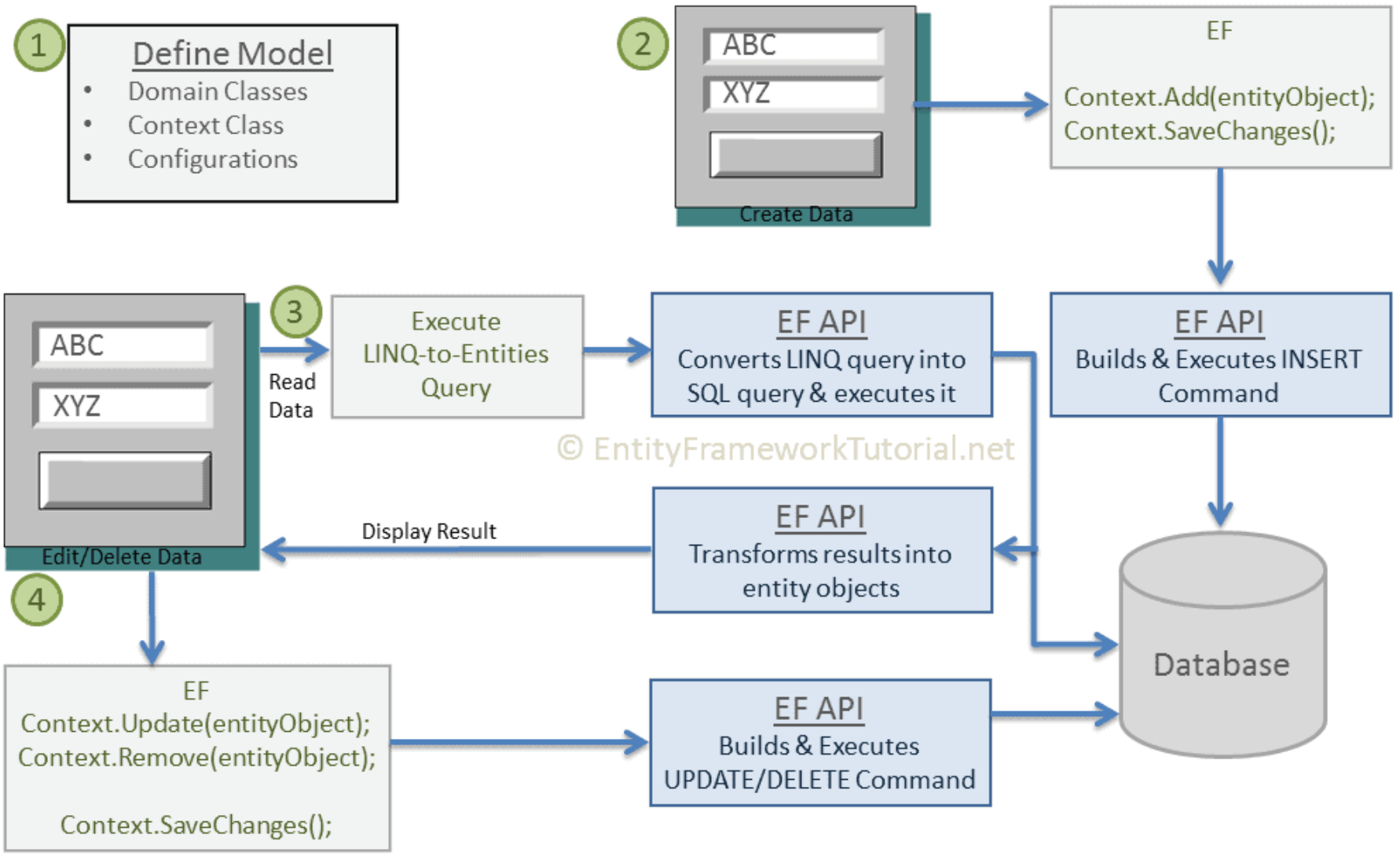
ENTITY FRAMEWORK CORE

Entity Framework Core

- É um framework da família .NET de persistência baseado no padrão *Data Mapper* multiplataforma
  - Windows, Linux e Mac
- Suporta múltiplas fontes de dados
  - https://learn.microsoft.com/en-us/ef/core/providers/
  - Distribuído via pacotes NuGet
- Suporta o uso de LINQ – Language Integrated Query do .NET



Entity Framework Core

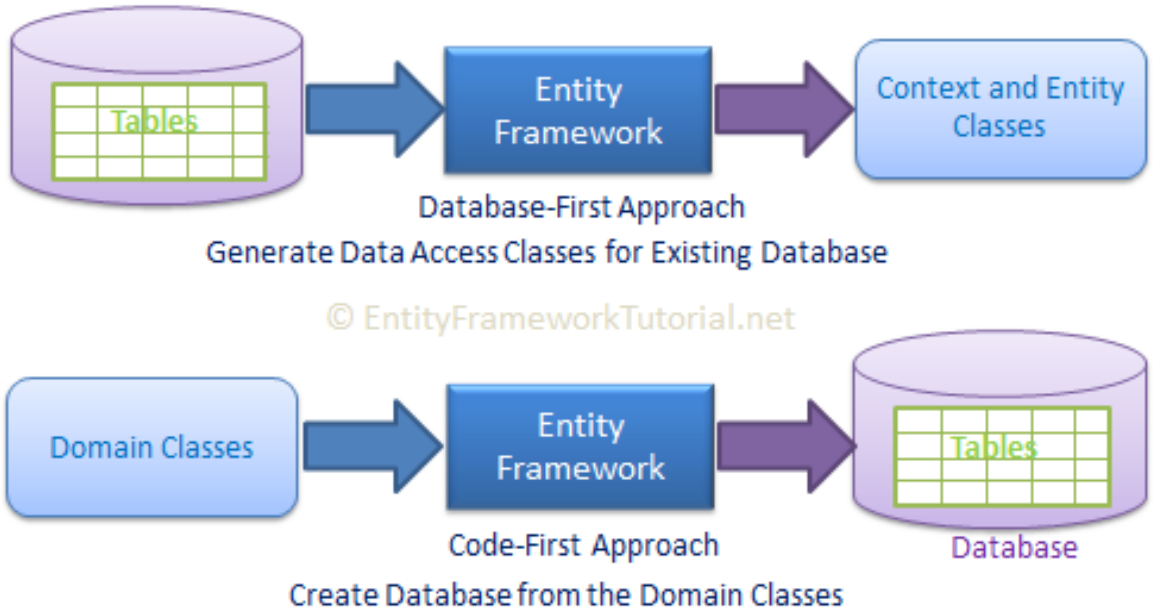


POCO - “*Plain Old C# Objects*”

- Objetos de domínio já existentes
- São objetos que não conhecem o mecanismo de persistência

Models

- Classes de modelo (POCO's) são mapeadas para tabelas, colunas e relacionamentos do banco de dados
- Formas de trabalho:
  - Code-first
  - Database-first



Models

- EFCore permite:
  - Utilizar convenções de mapeamento a fim de diminuir a quantidade de código necessário para mapear os objetos
  - Utilizar anotações sobre os objetos para configurar as regras de mapeamento
  - Utilizar uma API do tipo “fluente” para configurar as regras de mapeamento
    - Permite uma separação total entre um objeto de negócio e as regras de mapeamento
- Documentação:
  - https://learn.microsoft.com/en-us/ef/core/modeling/

Models

Tipo de dados C#	Tipo de dados SQL Server
int	int
string	nvarchar(Max)
decimal	decimal(18,2)
float	real
byte[]	varbinary(Max)
datetime	datetime
bool	bit
byte	tinyint
short	smallint
long	bigint
double	float
char	No mapping
sbyte	No mapping (throws exception)
object	No mapping

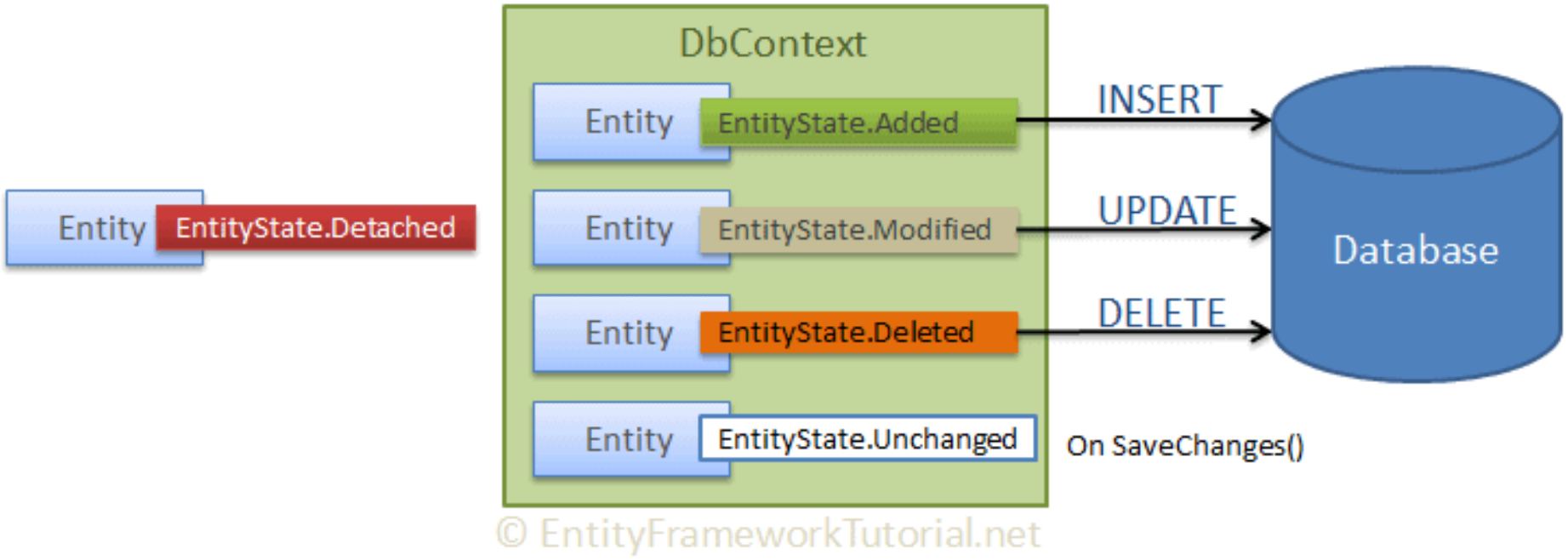
Contexto

- EFCore segue os padrões Data Mapper, Repository e Unit of Work
- Objeto **DbContext** é baseado nesses padrões e possui uma API para
  - Gerenciar objetos em memória (inclusive com cache)
  - Manter a ligação entre o banco de dados e as entidades mapeadas no modelo relacional
  - Gerenciar a conexão com a base de dados
  - Gerenciar o contexto transacional

Contexto

- Objeto **DbSet**
  - Representa uma coleção de entidades em um contexto de persistência
  - É obtida a partir do *DbContext*
  - Provê métodos para operações CRUD sobre um determinado tipo de entidade

Contexto


DbContext
Entity
EntityState.Added
Entity
EntityState.Modified
Entity
EntityState.Deleted
Entity
EntityState.Unchanged
Entity
EntityState.Detached
INSERT
UPDATE
DELETE
Database
On SaveChanges()
© EntityFrameworkTutorial.net

Database-First

- Banco de dados está previamente criado e deve ser acessado via EFCore
- Pode-se utilizar ferramentas de engenharia-reversa via *scaffolding* para automatizar a criação da classes e *DbContext*
- Documentação:
  - https://learn.microsoft.com/en-us/ef/core/managing-schemas/scaffolding

Code-First

- Banco de dados é gerado a partir do código e anotações das classes em projetos
- Um framework de migração (Migrations) atualiza o banco a partir de alterações no modelo
- Documentação:
  - https://learn.microsoft.com/en-us/ef/core/managing-schemas/migrations