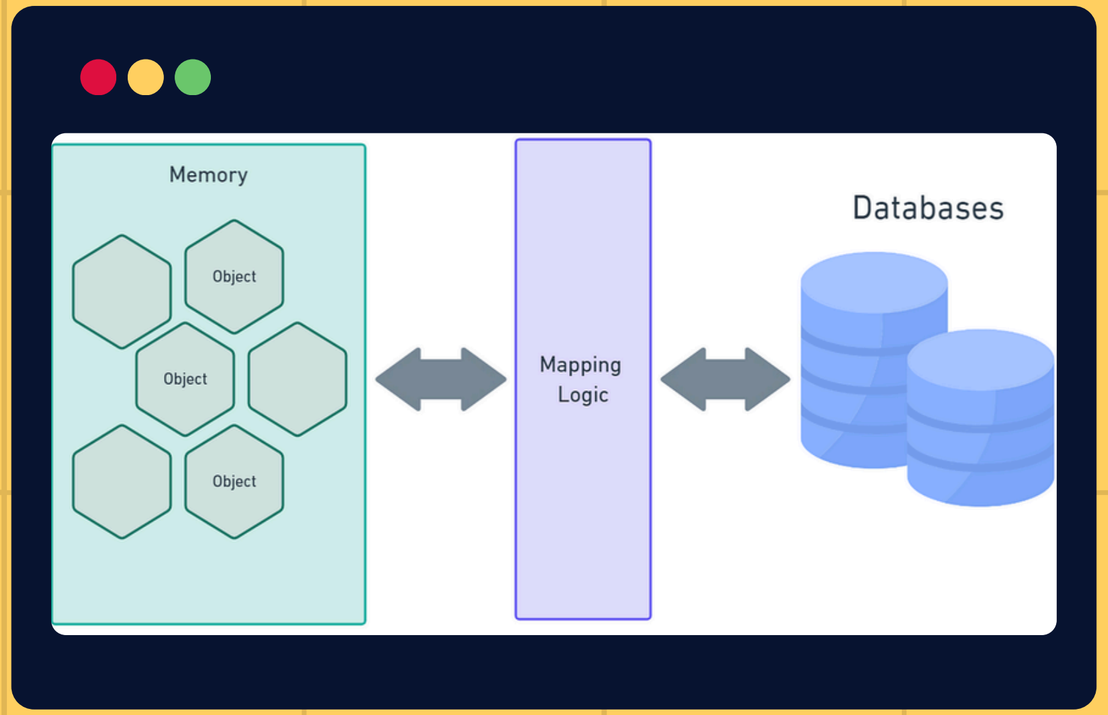
ORM & Entity Framework

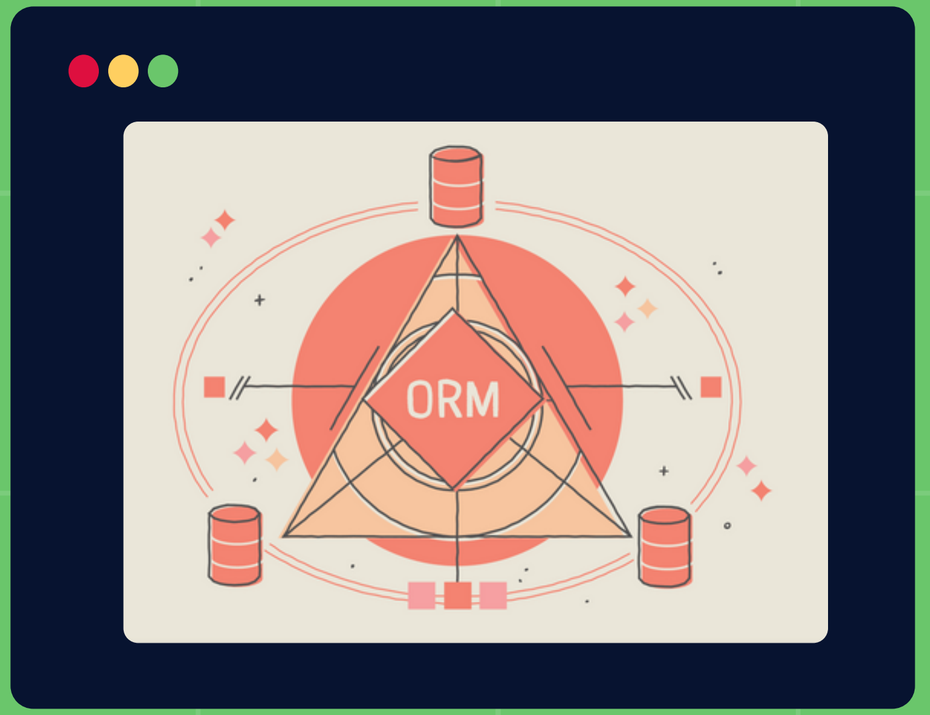
Poklop
Kotek
Junková

What is ORM?



- ORM (Object-Relational Mapping) is a technology that connects object-oriented programming with relational databases.
- It allows programmers to work with database tables using objects instead of writing SQL queries directly.
- ORM acts as a layer between the application and the database, automatically converting objects to tables and vice versa.

ORM enables mapping between objects in code and tables in a relational database. When an application needs to store an object in the database, the ORM generates the corresponding SQL statement.



How does ORM work?

When retrieving data from a database, an ORM converts tabular data back into objects.
An ORM also tracks changes made to objects and automatically synchronizes them with the database.

EF

```csharp
using var context = new LibraryContext();

// CREATE
context.Books.Add(new Book { Title = "The Hobbit", Author = "J.R.R. Tolkien", Year = 1937 });
context.SaveChanges();

// READ & UPDATE
var book = context.Books.FirstOrDefault(b => b.Title == "The Hobbit");
if (book != null) { book.Year = 1951; context.SaveChanges(); }

// DELETE
context.Books.Remove(book);
context.SaveChanges();
```

WITHOUT ORM

```csharp
using var conn = new SqlConnection("Server=...;Database=library;");
conn.Open();

// CREATE
var cmd = new SqlCommand("INSERT INTO Books (Title, Author, Year) VALUES (@t, @a, @y)", conn);
cmd.Parameters.AddWithValue("@t", "The Hobbit");
cmd.Parameters.AddWithValue("@a", "J.R.R. Tolkien");
cmd.Parameters.AddWithValue("@y", 1937);
cmd.ExecuteNonQuery();

// READ & UPDATE
cmd = new SqlCommand("UPDATE Books SET Year=1951 WHERE Title='The Hobbit'", conn);
cmd.ExecuteNonQuery();

// DELETE
cmd = new SqlCommand("DELETE FROM Books WHERE Title='The Hobbit'", conn);
cmd.ExecuteNonQuery();
```

Advantages & disadvantages of ORM

Advantages

- Simplify database work
- Automatic mapping
- Security
- Portability
- Increase productivity

Disadvantages

- Performance Management
- More Complex Questions
- Learning New Technology

Popular ORM frameworks

- Python: SQLAlchemy, Django ORM
- C#: **Entity Framework**
- Java: Hibernate
- PHP: Doctrine

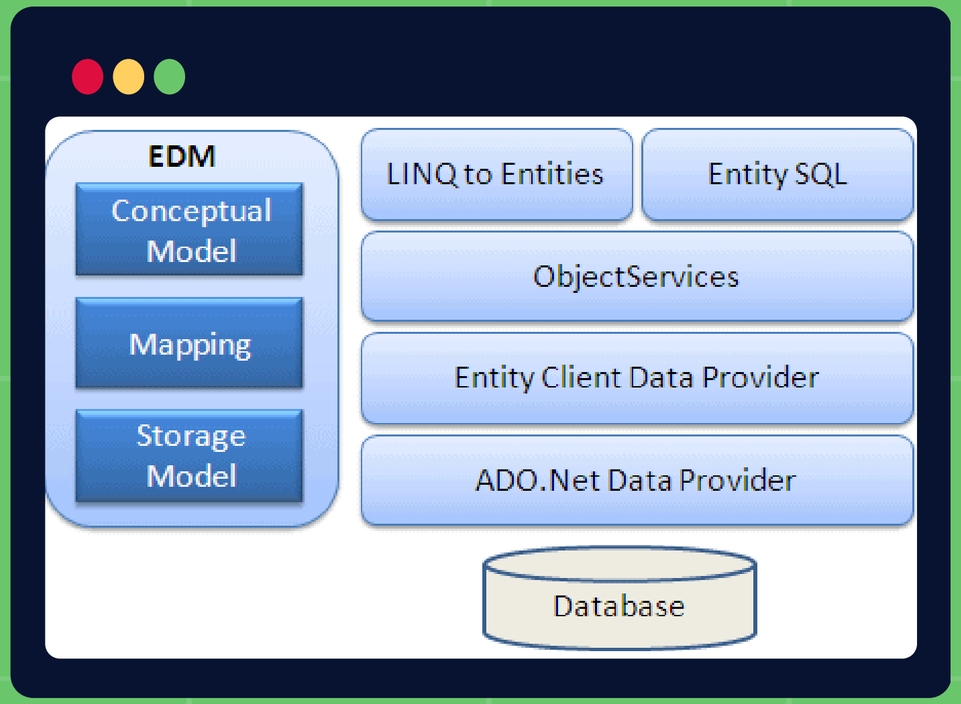
Entity Framework (EF)

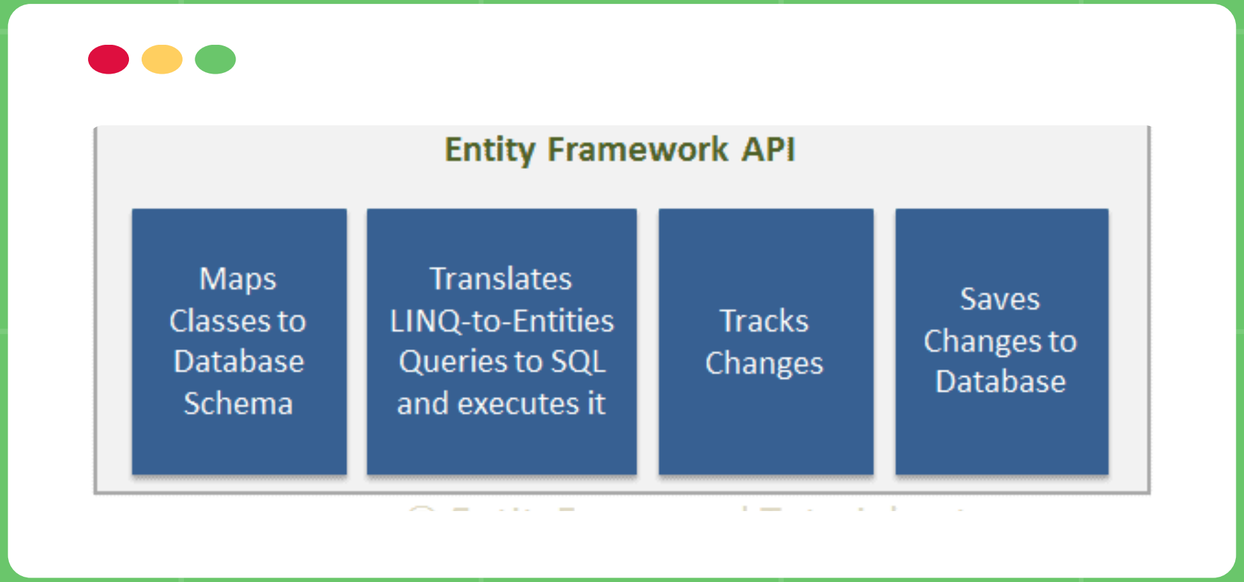
Entity Framework automates database operations and allows you to work with data as if it were regular objects in C#

- ORM (Object-Relational Mapping) framework for C# and .NET
- Works with databases as objects (instead of SQL queries)
- Automates CRUD operations

```
using var context = new LibraryContext();

// CREATE
context.Books.Add(new Book { Title = "The Hobbit", Author = "J.R.R. Tolkien", Year = 1937 });
context.SaveChanges();

// READ & UPDATE
var book = context.Books.FirstOrDefault(b => b.Title == "The Hobbit");
if (book != null) { book.Year = 1951; context.SaveChanges(); }

// DELETE
context.Books.Remove(book);
context.SaveChanges();
```

How does Entity Framework work?





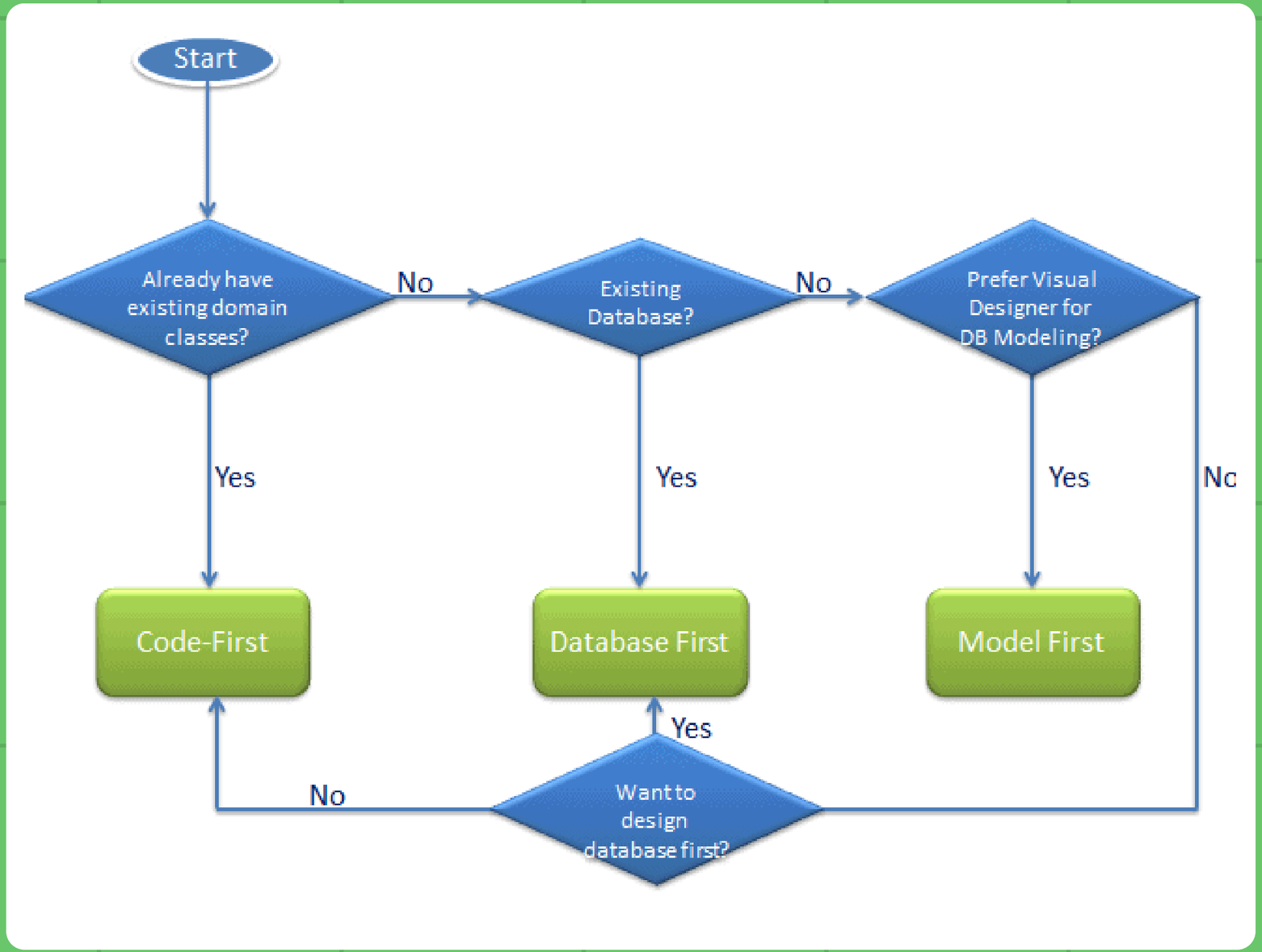
- Creates a data model (EDM – Entity Data Model)
- Maps classes to database tables
- Converts LINQ queries to SQL
- Tracks object changes
- Saves data using SaveChanges()

What is Entity & Context class in EF?

Entity = C# class mapped to a table
Scalar properties = columns in the database
Navigation properties = relationships between entities
DbSet = collection of entities in a context
Entity states (Added, Modified, Deleted...)

Represents a database connection
Inherits from DbContext
Contains DbSet<TEntity> – a collection of entities
Allows CRUD operations
Manages transactions and object changes

Development approaches



Thank you for your attention.

For English speakers, there is a readme in English on GitHub where all tasks are explained.

http://bit.ly/3FAYjX8