

The background is a digital illustration of a server room aisle. On both sides are rows of server racks, each filled with numerous server units that have glowing blue lights. The floor is a light-colored square tile, and the ceiling is a grid of square panels with recessed lighting. In the distance, a door is visible at the end of the aisle. Labels 'A3', 'B3', 'A4', and 'B4' are visible on the ceiling. The title 'Entity Framework Core' is centered in the middle of the image in a white, sans-serif font.

Entity Framework Core

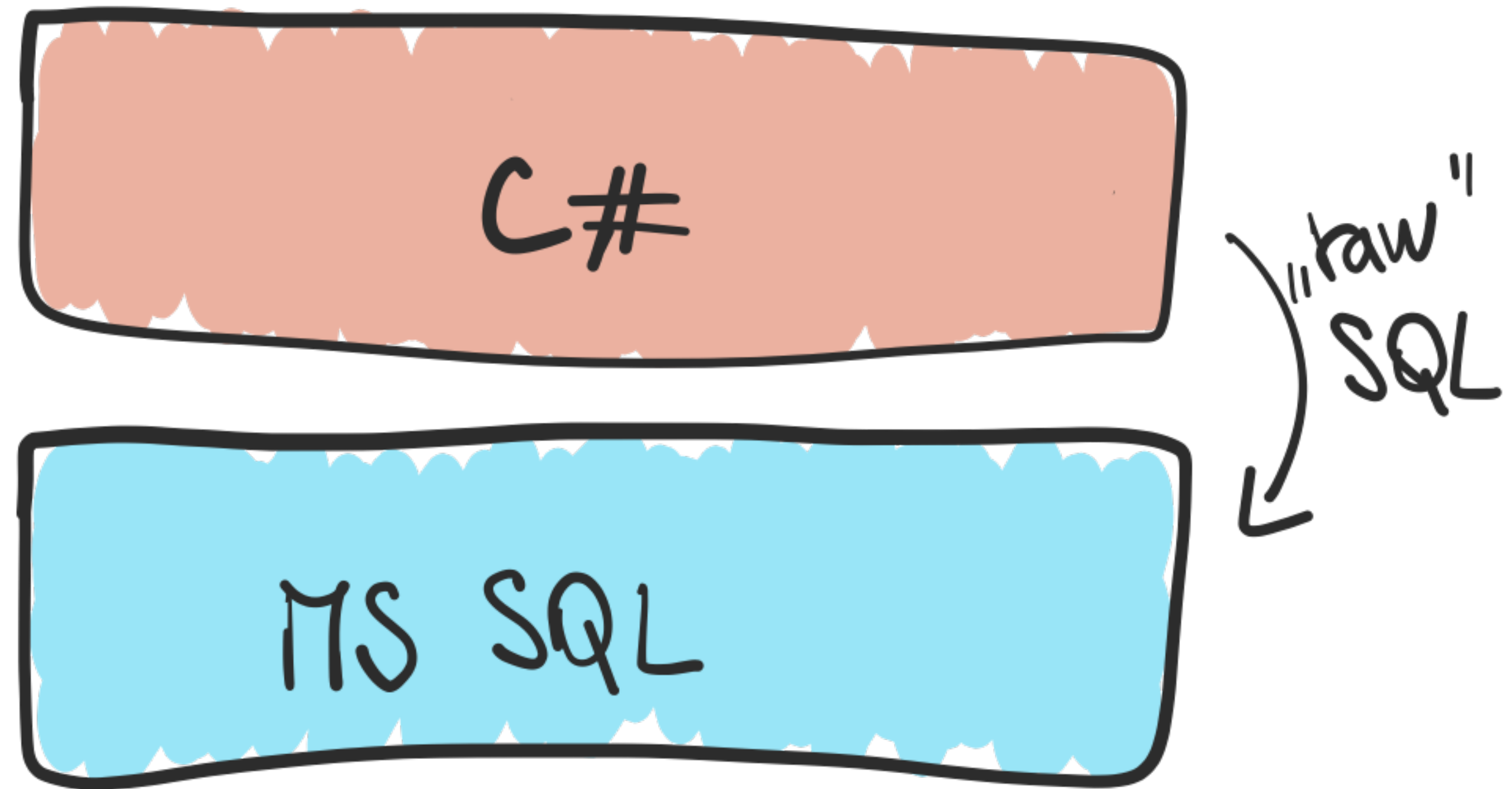
LUKÁŠ VAVREK

Short Definition

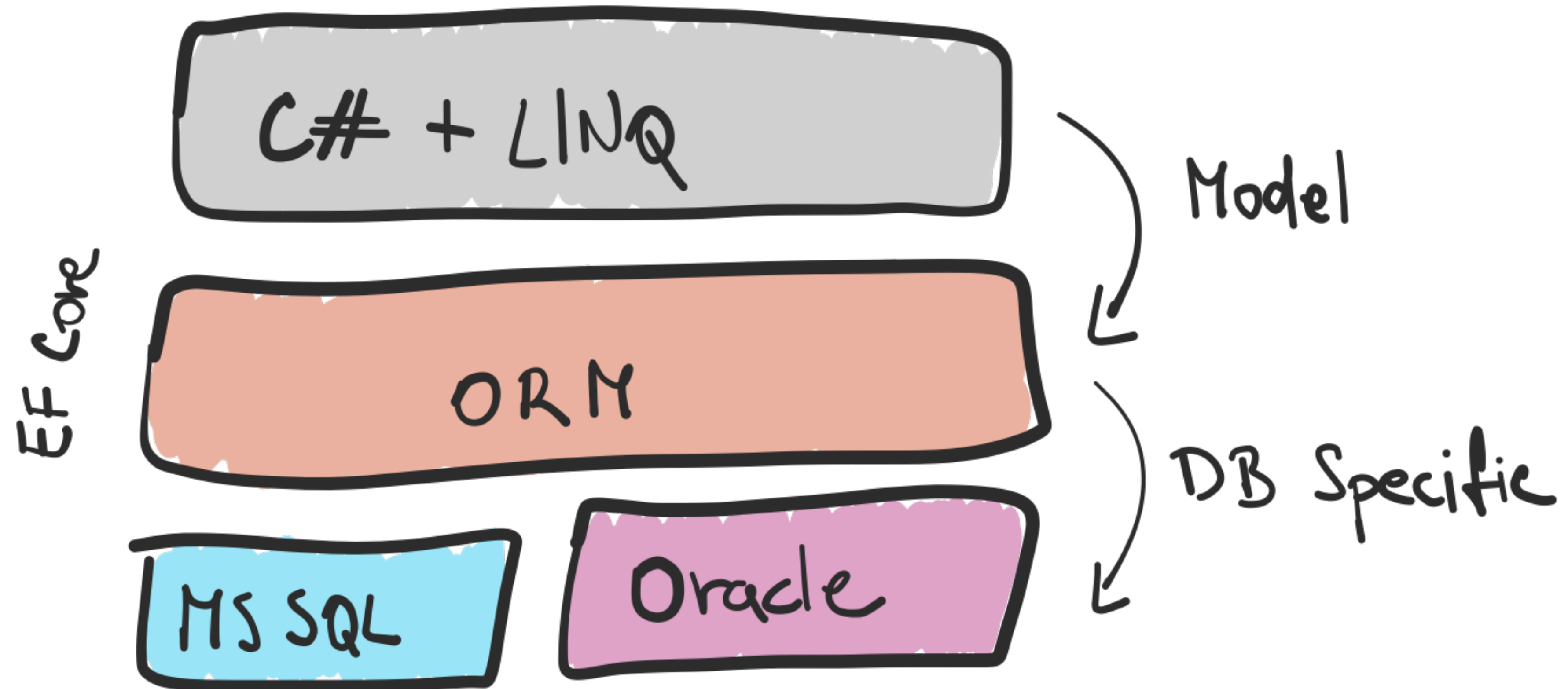
EF Core can serve as an object-relational mapper (O/RM), enabling .NET developers to work with a database using .NET objects, and eliminating the need for most of the data-access code they usually need to write.

EF Core supports many database engines.

Platform Specific



ORM



The Model

The model is responsible for performing data access. It is made up of entity classes and a context object that represents a session with the database, allowing you to query and save data.

Model can be hardcoded to match existing DB, generated from existing DB or created by hand and propagated to DB using migrations.

Code first

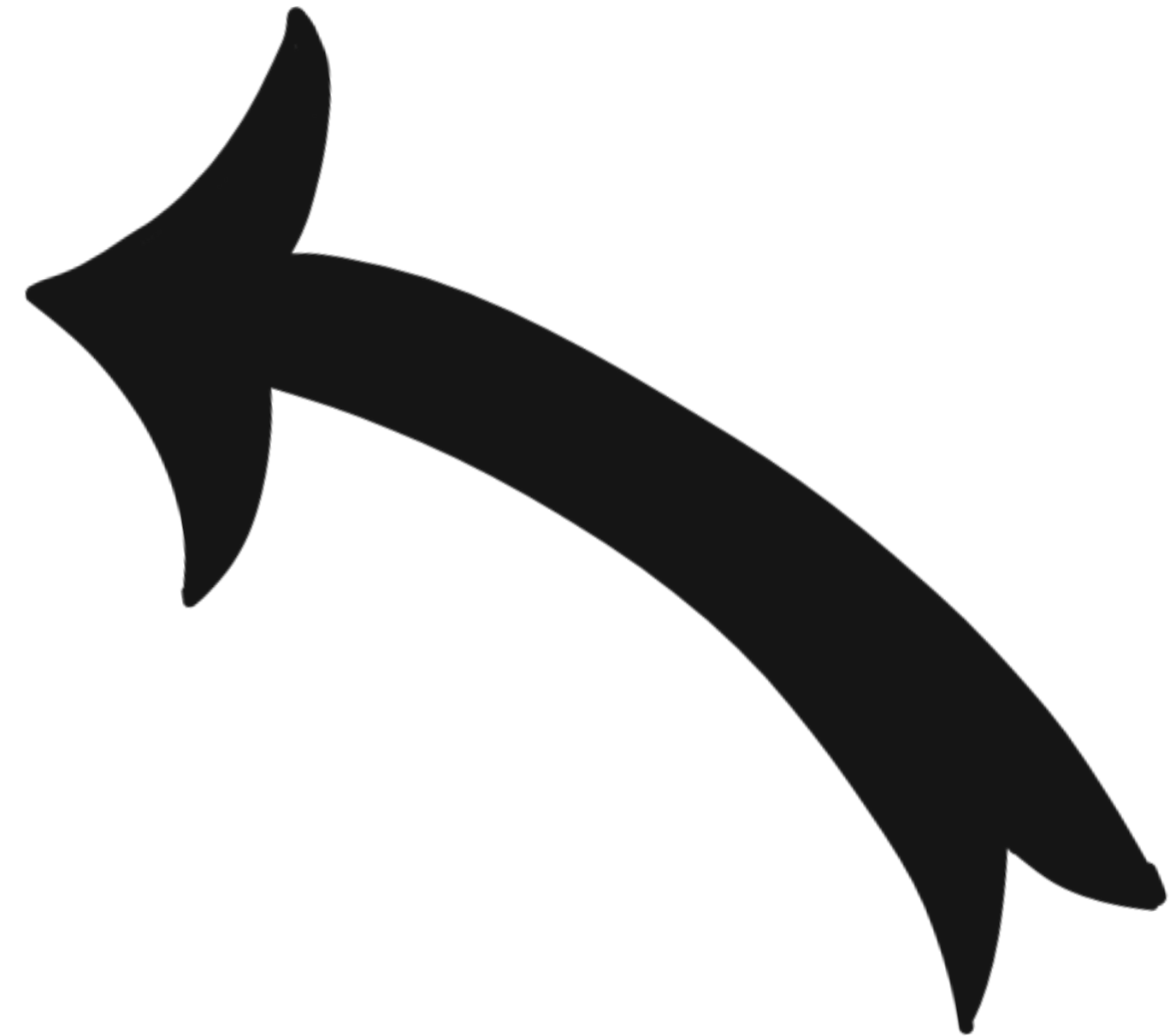
```
public class Blog
{
    public int BlogId { get; set; }
    public string Url { get; set; }
}
```



```
create table Blogs (
    blogId int,
    url varchar(100)
)
```

DB first

```
public class Blog
{
    public int BlogId { get; set; }
    public string Url { get; set; }
}
```



```
create table Blogs (
    blogId int,
    url varchar(100)
)
```

Querying

```
using (var db = new BloggingContext())
{
    var blogs = db.Blogs
        .Where(b => b.Rating > 3)
        .OrderBy(b => b.Url)
        .ToList();
}
```


Saving Data

```
using (var db = new BloggingContext())
{
    var blog = new Blog {
        Url = "http://sample.com"
    };
    db.Blogs.Add(blog);
    db.SaveChanges();
}
```




Q & A