## SQL



## Entity Framework

Entity Framework ORM(Object Relational Mapping) araçlarından biridir. ORM nedir dersek: İlişkisel veritabanı ile nesneye yönelik programlama(OOP) arasında bir köprü görevi gören araçtır. Bu köprü, ilişkisel veritabanındaki bilgilerimizi yönetmek için nesne modellerimizi kullandığımız bir yapıdır. Kısaca veritabanına bizim nesnelerimizi bağlayan ve bizim için veri alışverişini yapan Microsofttarafından geliştirilmiş bir framework’tür.

Entity Framework ile 4 farklı yöntem kullanarak proje geliştirebilirsiniz. Bu yöntemler;

* Model First (Yeni Veritabanı Oluşturma Visual Studio İle)
* Database First (Var Olan Veritabanını Kullanma)
* Code First (Yeni Veritabanı Kod Yazarak)
* Code First(Var Olan Veritabanını Kullanma)

**Model First:**Bu yöntem Visual Studio üzerinde boş bir model dosyası (.edmx) eklenerek veritabanını bu model üzerinde tasarlanabilmesine olanak sağlıyor. Derleme adımında verilen script dosyası veritabanını oluşturur.

**Database First:**Bu yöndem önceden oluşturulmuş olan veritabanını projeye model olarak bağlayarak gerekli classlarımız Entity Framework tarafından oluşturulmaktadır.

**Code First (Kod Yazarak):**Bu yöntem (Ki benim en sevdiğim) classlarımızı visual studio ortamında oluşturmaya başlayarak gerçekleştirdiğimiz bir yöntemdir. Veritabanımız bu classlardan türetilmektedir. Burada Mapping işlemleri yazılımcı tarafından classlar oluşturulurken **Attribute**’lar sayesinde yapılabilmektedir. Bu arada Mapping işlemi kısaca tablolarımızdaki kısıtlarımızı belirlediğimiz olaydır. Attribute’ların yanında farklı yöntemlerlede bu işlemleri gerçekleştirebilmekteyiz. Örnek vermek gerekirse **Fluent Api**veya **Fluent Validation**gibi araçlar Mapping işlemleri için popüler olarak kullanılmaktadır.

**Code First (Var Olan Veritabanını Kullanma):**Bu yöntemde de classlar ve mapping kodları yazılımcı tarafından oluşturulmaktadır. Veritabanımız classlarımızın ve modellemenin durumuna göre güncellenmektedir.

*Kaynak <*[*https://medium.com/t%C3%BCrkiye/entity-framework-nedir-bff3943bec72*](https://medium.com/t%C3%BCrkiye/entity-framework-nedir-bff3943bec72)*>*

Yani veritabanındaki tabloları doğrudan **C# sınıfları** olarak kullanabilmeyi sağlar. SQL sorgusu yazmak yerine LINQ ile (C# sözdizimiyle) veritabanına erişilir.

### Self Notes on Entity Framework

**Entity Framework** kullanarak;

* SQL Server'daki **KampusTek veritabanı** ile
* Visual Studio'daki **C# uygulaması** arasında bir **bağlantı (bir köprü, bir tercüman)** kurdum.

Bu sayede artık veritabanı işlemlerini SQL komutları yazarak değil, C# kodları yazarak yapabileceğim.

**"Scaffolding" (İskele Kurma), var olan bir veritabanı şemasını okuyarak, o şemayla birebir uyumlu C# sınıflarını ve DbContext'i otomatik olarak oluşturan bir süreç.** Bu sürece aynı zamanda **"Reverse Engineering" (Tersine Mühendislik)** de denir. Çünkü normalde koddan veritabanı oluşturulurken, bu işlemde veritabanından kod oluşturulur.

**Database-First (Önce Veritabanı) Yaklaşımı** kullandım çünkü elimde önceden oluşturduğum veritabanım vardı. Bu durumda C# sınıflarının, var olan veritabanıyla **mükemmel bir şekilde eşleşmesi** gerekiyordu.

* Bu yüzden Scaffold-DbContext komutu, bu eşleşmeyi sağlamak için kullanılan standart araç olduğu için onu kullandım.

Scaffold-DbContext "Server= \SQLEXPRESS;Database=KampusTekDBv2;Trusted\_Connection=True;TrustServerCertificate=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models -Force