**ENTITY FRAMEWORK**

Entity Framework (EF), Microsoft tarafından 2008 yılında kullanıcılar için piyasaya sürülmüş bir Object-Relational Mapping (ORM) aracıdır. Bu makalede EF'nin ne olduğunu, temel kavramlarını, nasıl kullanıldığını ve avantajlarını ele alacağız.

EF’nin bir Object-Relational Mapping (ORM) aracı olduğunu söyledik. Peki nedir bu ORM? ORM, ilişkisel veri tabanlarındaki (Relational Databases) tablolar ile nesne yönelimli programlama arasında eşleme yapan bir yapıdır. Otomatik olarak yapılan bu eşlemeler sayesinde, veri tabanı işlemleri (CRUD: Create, Read, Update, Delete) nesneler üzerinden daha efektif ve basit bir şekilde gerçekleştirilebilir.

Peki EF’nin bize yararı nedir ve neden bir ORM aracı olan EF’ye ihtiyaç duyarız? Daha önce belirttiğimiz gibi EF, veri tabanı tablolarını nesnelerle eşleyerek veri işlemlerini daha soyut ve kolay bir şekilde yapmamıza olanak tanır. ADO.NET gibi farklı veri erişim seçeneklerimiz olsa da bu teknolojiler bu tür bir soyutlama sağlamaz. Bunun yerine veri tabanı işlemlerini doğrudan SQL komutları ve veri erişim nesneleri kullanarak gerçekleştirir. Bu nedenle, genellikle daha efektif bir seçenek olan EF’yi kullanmayı tercih ederiz.

**Entity Framework’ün Temel Kavramları**

EF kullanırken bilmemiz gereken ve uygulamalı olarak yaptığımız bazı temel özellikler vardır.  
**Veri Modelleme:**   
Bunlardan birisi olan veri modellemede EF sayesinde veri tabanı tablolarını sınıflar olarak modellememiz mümkündür. Proje içinde oluşturduğumuz her bir model sınıfı veri tabanındaki tablolara denk gelir.

**Entity Classes:**  
Entity sınıfları (classes), veri tabanındaki tabloları temsil eder ve bu sınıfları model yapmada kullanırız. Bu sınıflar, uygulamamız ile veri tabanı arasındaki veri aktarımını sağlamaktadır.

**DbContext:**DbContext, modellerimizi veri tabanına aktarmak için kullanacağımız bir ana sınıftır. Bu sınıf veri tabanı ile etkileşime geçmek için kullanılır.

**Migrations:**Migrationlar, veri tabanında yapılan değişiklikleri yönetmek ve veri tabanına aktarmak için kullanılan araçtır. Veri tabanını kodla uyumlu hale getirmek için kullanılır.

**Entity Framework’ün Proje Türleri**

EF ile yapabileceğiniz 4 farklı proje türü vardır. Bunlar Model First, Database First, Code First(Yeni veri tabanı kodu yazarak) ve tekrar Code First(Var olan veri tabanını kullanarak)’tür. Ben ise en yaygın olan Database First ve Code First(Yeni veri tabanı kodu yazarak) türlerini anlatacağım.  
  
**Database First:**

Bu proje türünde önceden oluşturduğumuz veya halihazırda oluşturulmuş bir veri tabanını projemize model olarak entegre eder, bağlarız ve veri tabanındaki tablolarımız projemizde yeni sınıflar olarak EF tarafından oluşturulur.

**Code First:**Bu yöntemde ise proje içinde oluşturduğumuz sınıf modellerimizi migration yoluyla veri tabanına aktarırız ve EF bu sınıfları veri tabanında tablolara karşılık gelecek şekilde oluşturur ve tutar.

**Avantajlar ve Dezavantajlar**Sonuç olarakEF, veri tabanı işlemlerini nesne yönelimli bir şekilde yönetmek için güçlü ve etkili bir teknolojidir. İlişkisel veri tabanları ile çalışmayı kolaylaştırır ve veri tabanı işlemlerini daha verimli hale getirir. Ancak, performans sorunları ve öğrenme açısından karmaşık olabileceği için biraz zaman alması gibi dezavantajları da vardır.

**Özet**Bu makalede genel olarak Entity Framework'ün ne olduğunu, temel kavramlarını, kullanım şekillerini ve avantajlarını anlatmaya çalıştım. EF, ilişkisel veri tabanları ile nesne yönelimli programlama arasında bir köprü oluşturarak, veri tabanı işlemlerini daha kolay ve verimli hale getiren bir teknolojidir. Veri modelleme, entity sınıfları, DbContext ve migration gibi temel kavramları anlamamız, EF'yi daha etkin ve verimli kullanmamızı sağlayacaktır.

**Berkan İrez**