Dökümanı forklayan adının yanına bir tik atıp PR atsın.

Emre Çalışkan x

Senem Canpolat x

Berkay Gülşen

Eda Kurnaz x

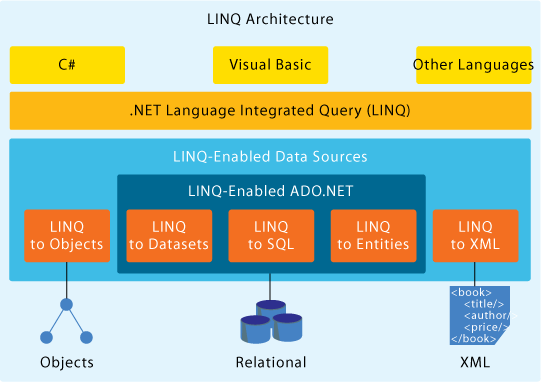
Rıdvan Öztürk x

KISACA LINQ NEDIR?

LINQ açılımı dile entegre edilmiş sorgu. Hayatımıza C# ile girmiştir. Farklı veri kaynaklarından sorgulama yapma imkanı sunar.

LINQ Mimarisi 3 katmandan oluşmaktadır. En üstteki katman LINQ yapısını kullanacak olan programlama dilleridir. Ortadaki katman LINQ sorgularının işlendiği katmandır. En alttaki katman ise LINQ için çeşitli veri yapılarını bulundurur. Bu veri yapıları ile LINQ Query katmanı arasında LINQ provider (sağlayıcı) iletişim kurar. Sağlayıcı sayesinde sorgular anlaşılır hale gelir.

LINQ farklı veri kaynaklarına erişim sağlayabilir. Veri üzerindeki işlemleri kolaylaştırır.



LINQ VERİ SAĞLAYICILARI

LINQ to Object : List<T> tipindeki yapıların sorgulanması amacıyla geliştirilmiştir.

LINQ to SQL: Veriyi tutan ortamlardaki verilere erişmek ve erişilen verilerin üzerinde işlem yapmak amacıyla geliştirilimiştir.

LINQ to XML: Veri kaynağına parsel kullanılmadan hızlı bir biçimde erişilmek için geliştirilmiştir.

NEDEN LINQ TERCIH ETMELIYIZ?

-Sorguların yazımında üretkenlik düşmektedir.

-Çalışma esnasında veri tipi uyuşmazlığı ile karşılaşırız.

-Tip kontrolü yapılmaz. Bundan ötürü hatalar meydana gelebilir.

LINQ to Object

Hafızada bulunan ve Enumerable<T> arayüzünü uygulayan bütün koleksiyonlarda kullanılabilir. From where ve select gibi yapıları SQL ortamındaki gibi kullanarak işlem yapabiliriz.

LINQ to SQL

LINQ ifadeleri arka planda Queryable arayüzü ile uyumlu nesneler üretir. Bu arayüz içinde LINQ ifadeleri kollara ayrılır. Kollara ayıran nesnenin adı Expressiondur. Queryprovider isimli arayüz sayesinde ise dinamik sorgu cümleleri oluşturur.

Bu sorguda çalışma esnasında t-sql yapısına çevrilir. Çevirilen yapı veri tabanı üzerinde direkt olarak sorgulama yapabilir.

LINQ to XML

LINQ mimarisinin XML verilerini sorgulaması için üretilmiştir. Bu sınıfların tamamı Syste.Xml.Linq olarak adlandırılır.

**LINQ - EF İlişkisi**

**Entity Framework:** .Net platformunda ORM (Object Relational Mapping) araçlarından birisidir. ORM (Object Relational Mapping) ise veritabanı ile nesneye yönelik programlama (OOP) arasındaki ilişkiyi kuran teknolojidir. Basite indirgemek istersek, Nesnelerimizi ilişkisel veritabanındaki tablomuza bağlayan ve veri alış-verişini bizim için yapan bir tekniktir/metodtur. ORM tekniği belli bir programlama diline bağlı değildir ve her nesneye yönelik programlama dillerinde yazılabilir/kullanılabilir. Yani Entity Framework, nesne tabanlı programlamada veritabanındaki tablolara uygun nesneler oluşturma tekniğidir.

Entity Framework ‘ün amacı yazılım geliştiricileri katı SQL sorgularından kurtararak ORM (Object Relational Mapping) imkânı sağlar. LINQ sayesinde yazılım geliştirici katı, uzun ve karmaşık SQL sorgularını yazmaktan kurtulur. Bunun yanında Entity Framework CRUD (Create, Read, Update, Delete) işlemleri ile uygulama ve veritabanı arasındaki iletişimin yapılmasını sağlar.

* CRUD (Create, Read, Update, Delete) işlemleri ile uzun, karışık ve zahmet verici SQL kodlarından kurtulmamızı sağlar.
* Kodu veritabanından soyutlar, yani kod tarafında EF için oluşturulan entity’ler aslında veritabanı tarafındaki tabloları ve bu tablolar arasındaki bağlantıları (one to one, one to many ve many to many gibi) temsil ederler.
* Veritabanı işlemlerinde nesneye yönelik kod yazmamızı sağlar.
* Daha sade ve zahmetsiz SQL sorguları sayesinde veritabanı performansını artırır.
* Yazılım geliştiricinin data işlemleri ile haşır neşir olmadan sadece uygulama üzerinde odaklanmasına olanak sağlar.
* Kod yazma süresini kısaltarak daha az zamanda daha çok iş yapmayı sağlar.
* SQL tarafındaki bütün sorguları (query’leri) LINQ kullanarak kod tarafında yazmamızı sağlar.

Entity Framework ile 3 farklı yöntem ile proje geliştirilebilir. Bu yöntemler;

1. Model First (New Database)

2. Database First (Existing Database)

3. Code First (New Database)

**Model First (Önce Model):** Bu yöntemde Visual Studio üzerinde boş bir model dosyası (.edmx) eklenerek veri tabanı bu model üzerinde tasarlanır. Derleme adımında verilen script dosyasi ile veri tabanı oluşturulur.

**Database First (Önce Veritabanı):** Bu yöntemde hali hazırda var olan veritabanı projeye model dosyası ile bağlanır ve gerekli class’lar EF tarafından üretilir.

**Code First (Önce Kod — Yeni Veritabanı):** Bu yöntemde classlar ve mapping kodları yazılımcı tarafından oluşturulur. Daha sonra veri tabanı bu class’lardan türetilir.

**REFERANSLAR**

* <http://hurkanseyhan.blogspot.com/2014/04/linq-nedir.html>
* <https://medium.com/t%C3%BCrkiye/entity-framework-nedir-bff3943bec72>
* <https://www.mehmetaltunel.com/csharp/entity-framework-nedir-ne-ise-yarar/>
* <https://www.yusufsezer.com.tr/entity-framework/>
* <https://coskunkurtuldu.medium.com/net-core-entity-framework-code-first-yakla%C5%9F%C4%B1m%C4%B1-6ac02867b3e5>
* <https://hakanyalitekin.medium.com/linq-nedir-f55957d5bf52>