

Nesneye Dayalı Programlarla Nesne İlişki Haritalanması

Mustafa Cem Kasapbaşı

İstanbul Ticaret Üniversitesi Bilgisayar Mühendisliği Bölümü
mckasapbasi@iticu.edu.tr

Özet: Nesne İlişki Haritalama (Object Relation Mapping, ORM) nesneye dayalı programlama aracılığı ile daha çok, veri tabanı bağlı işler yapılmak istendiğinde ve bu veri tabanındaki nesnelere (Tablolar, Görünümler, Kayıtlı yordamlar, Fonksiyonlar vb.) arasındaki ilişkileri, programlama dilinin bir nesnesi olarak kullanılmasının sağlanmasına yarayan bir programlama yöntemidir. Bu çalışmada gerek paralı gerekse bedava kullanıma sunulan ORM uygulamaları değerlendirilip, değişik programlama dillerindeki ORM yapıları hakkında bilgi verilip, kıyaslamaları sunulmaya çalışılacaktır.

Anahtar Kelimeler: Object Relation Mapping, ORM, .Net entity framework, Java JPA

Object Relation Mapping with Object Oriented Languages

Abstract: Object Relation Mapping, (ORM) is mostly used in object oriented languages programming technique when tasks related to database needed to be done and also the relations between objects in that database system (tables, views, stored procedures etc.) are represented as object of the common programming languages. In this study, free and commercial ORM applications are evaluated, and ORM tools of different programming languages are represented and compared.

Keywords: Object Relation Mapping, ORM, .Net entity framework, Java JPA

1 ORM nedir?

Programlama ile uğraşanlar er ya da geç veri tabanı ile program yazmaları gerektiğinde bazı kodları tekrarladıklarının farkına varılmaktadır. Özellikle veri tabanlarındaki nesnelere (Tablolar, Görünümler, Kayıtlı yordamlar, Fonksiyonlar vb.) kullanırken bu problem daha fazla karşılaşılmaktadır. Bu kullanılan nesneye dayalı programlama dilinden bağımsız olarak her programcının karşılaştığı bir problemdir. Çözümü için geliştirilen kavramlardan biri de ORM dir[1-3].

1.1 ORM neden önemlidir?

İlişkisel modellemede amaç, verileri normalize etmektir, buna karşılık nesneye dayalı tasarımın amacı gerçek dünya iş süreçlerini modelleyen veriler ve nesnelere oluşturmaktır. Sağlam bir nesneye dayalı uygulama, benzerlikler ve fark-

lılıkların tam anlamıyla anlaşılması ve Farklı amaçlar için oluşturulan bu nesnelere arasında yapılacak uygun bir haritalama stratejisi belirlenmesi ile mümkün olabilir.

1.1.1 ORM nin avantaj ve dezavantajları

ORM nin avantajları ve dezavantajları bu bölümde verilmeye çalışılacaktır [4]:

Avantajları :

- Daha hızlı ve verimli şekilde iş süreçlerinin modellenmesi
- Kod tekrarının oldukça azalması
- Karmaşık sorguları daha kolay şekilde çözümlenmesine olanak sağlaması
- Kodların ayıklanması daha kolaydır [5].

Dezavantajları

- Kütle halindeki silme işlemlerinde bazı ORM araçlarının performansı başarılı değildir.

- Kayıtlı yordamların daha iyi performans sağladığı düşünülmektedir.
- Her dil için farklı araçlar olması ve getirdikleri yeni kodlama tekniklerinin öğrenilmesi. Buna örnek olarak .Net 3.5 altyapısı içinde kullanılan Linq (language Integrated Query) sorgu dili ve Lamda ifadeleri verilebilir.

ORM kullanarak kod yazmak yazılması gereken kod miktarını azaltır ve daha az hatalı kod yazılmasını sağlar[1].

2. ORM Haritalama Stratejileri

Hiç bir strateji yoktur ki tüm durumlar için hep en iyi sonucu versin. Aşağıda belirtilen stratejiler tek bir uygulamada da birleştirilebilir [6]:

- Hiyerarşi başına bir tablo: Basit ve /veya derin olamayan sınıf hiyerarşileri için uygun bir çözüm olur. Sınıf hiyerarşileri içinde veya veri tiplerinde hiç örtüşme, üst üste binme yok ise.
- Somut bir sınıfa bir tablo: Tiplerin farklılığı ve/veya tipler arası örtüşme az rastlanıyor ise kullanımı uyum olabilir.
- Bir sınıfa bir tablo: Tipler arasında belirgin bir örtüşme varsa ve tiplerin değiştirilmesi olağan ise kullanılmalıdır.
- Tablodaki ilişkilere göre alınacak stratejiler: Dikkat edilmesi gereken diğer bir stratejide tablodaki ilişkilere dir. Bunlar birden bire, birden çoğa, ve çoktan çoğa olmak üzere tiplendirilebilir. Bu tiplere göre veri ekleme ve silmede getirilmesi gereken kısıtların ORM nesnelere yansıtılıp yansıtılmayacağı da verilmesi gereken önemli bir karardır.

Bu çalışmada stratejiler sadece tanıştırılmıştır.

Bir sınıf bir tablo için bir örnek aşağıdaki verilmektedir:

Column Name	Data Type	Allow Nulls
KatilimciID	int	<input type="checkbox"/>
KAdi	nvarchar(50)	<input type="checkbox"/>
KSoyadi	nvarchar(50)	<input type="checkbox"/>
Eposta	nvarchar(50)	<input type="checkbox"/>
Adres	nvarchar(50)	<input type="checkbox"/>
Telefon	nvarchar(50)	<input type="checkbox"/>
DogumTarihi	datetime	<input type="checkbox"/>
Maas	money	<input type="checkbox"/>

Şekil 1 Bir sınıf başına bir tablo stratejisinde kullanılacak tablo

Şekil 1 deki Katilimci adlı tablo için kullanılacak ORM nesnesi aşağıdaki Şekil 2 de gibi tasarlanabilir:

```
public class Katilimci
{
    //properties
    public int _KatilimciID { get; set; }
    private string _Ad;
    public string Ad
    {
        get { return _Ad; }
        set {
            _Ad = value;
            //event listener
            if (AdDegisti != null)
                AdDegisti();
        }
    }
    public string _Soyad { get; set; }
    public string _Adres { get; set; }
    public string _Telefon { get; set; }
    public string _Eposta { get; set; }
    public DateTime _DogumTarihi { get; set; }
    public Decimal _Maas { get; set; }

    //methods
    public DataTable KatilimcileriGetir()
    { //methodun kodu }
    public int KatilimciEkle()
    { //methodun kodu }
    public int KatilimciGuncelle()
    { //methodun kodu }
    public int KatilimciSil()
    { //methodun kodu }

    //events
    public delegate void AdDegistiHandler();
    public event AdDegistiHandler AdDegisti;
}
}
```

Şekil 2 Katilimci tablosu için kullanılacak nesne kodu

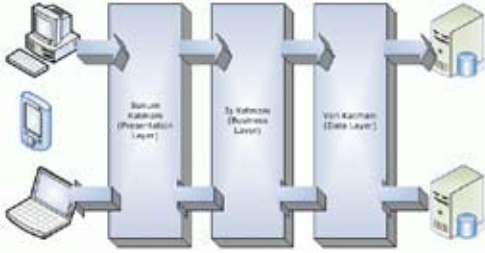
2.2 ORM Veri Yükleme Teknikleri

Ağır yükleme (Lazy Loading) bilgisayar programlamada sıklıkla kullanılan bir tarım örüntüsüdür (design pattern). Ne nesnenin oluşturulmasını ihtiyacı duyulana kadar geciktirmesi mantığına dayanır. Uygun bir şekilde kullanılırsa

performansı arttırmaktadır [7]. Bu şekilde tasarlanan bir ORM sınıfında nesnelere ihtiyaç duyulduğunda ve istenildiğinde yüklenmektedir.

Diğer bir yükleme tipi ise istekli yükleme (Eager Loading) dir. Bu nesne oluşturulur oluşturulmaz ilgili verinin de yüklenmesi anlamında kullanılır.

3.3 katmanlı Mimari ve ORM nin Kullanımı



Şekil 3 3 katmanlı mimari görünümü [8]

3 katmanlı mimari yazılım mühendisliği konularındandır ve bu makalede derinlemesine anlatılamayacak kadar kapsamlıdır. Bu sebepten sadece ORM kapsamı içinde kalan kısmı hakkında bilgi verilecektir. Şekil 3 de 3 katmanlı yapıyı oluşturan katmanlar gösterilmektedir. Burada 3 ayrı mantıksal/fiziksel ayırım söz konusu olabilir. Her bir katman farklı fiziksel lokasyonlarda olabileceği gibi, tüm katmanlar tek bir fiziksel lokasyonda da toplanabilir. Görevleri kısaca

- Veri Katmanı: Verilerin tutulduğu ortamı temsil etmektedir. Herhangi tipte bir depolama ortamını ifade edebilir. Bu ilişkisel bir veri tabanı olabilir.
- İş Katmanı: Bu ORM nesnelere içinde bulunduğu katmandır. Genellikle iki kısma ayrılır. Veri erişim katmanı, iş mantık katmanı (Data Access Layer ve bussines logic layer) adı verilir.
- Sunum Katmanı : Bu verilerin sunum ortamını ifade eder, bir Windows formundan, internet sayfasına kadar herhangi bir formda olabilir.

ORM nesnelere tam bir köprü olarak orta katmana yerleşmişlerdir. Burada sunum katmanından veri katmanına doğru tüm iletişim ORM nesnelere üzerinden yapılır.

4 ORM Uygulamaları

ORM Nesneye dayalı programlama dillerinde uygulamaları piyasada para ile satıldığı gibi bir bedava olanları da vardır aşağıda bu uygulamaları bir kaç kullanıldığı en çok tercih edilen programlama dillerine göre Tablo1 de listelenmiştir [9].

Java	.NET
Carbonado,	ADO.NET Entity Framework
Cayenne,	Base One Foundation Component Library,
Ebean	BootFX,
EclipseLink,	Castle ActiveRecord,
Enterprise Objects Framework,	DataObjects.NET
Hibernate,	Devart LINQ to SQL
iBATIS,	evForce,
Java Data Objects (JDO)	Habanero
JPOX,	LINQ to SQL,
MPF/J,	NHibernate,
Kodo.	ObjectMapper .NET
Object Relational Bridge (Apache OJB),	
OpenJPA,	
Torque,	
JPA	

Tablo1

Fark edileceği gibi bir çok platform tercih edilen programlama diline göre kullanılabilir. Ticari olarak satılan bazı araçlarda her iki dünyayı tek bir platformda birleştirmeyi de vaat etmektedirler.

[10] da ki yapılan seminer ve buluşmalarda java ve .Net farklı konularda ve platformlarda değerlendirilmiş karşılaştırılmıştır. Bu etkinliklerde konusunda uzman kişilerin verdiği kararlara göre .Net teknolojileri (linq ve Entity Framework) java Teknolojilerine göre (JPA ve hibernate) daha başarılı olmuştur. Bu sonuçlar bilimsel olmasa okuyucuya konusunda uzman

olan kişilerin yorumları hakkında fikir vermek için çalışmaya eklenmiştir.

5. Sonuçlar

ORM bir programlama tekniğidir ve kullanıldığında; Daha hızlı ve verimli şekilde iş süreçlerinin modellenmesi, Kod tekrarının oldukça azalması, Karmaşık sorguları daha kolay şekilde çözülmesine olanak sağlaması, Kodların ayıklanması daha kolay olması gibi bir çok avantajı bulunmaktadır. ORM nin daha yaygın uygulanabilmesi için gerek MYO Bilgisayar Programcılığı bölümü gerekse, Bilgisayar mühendisliği bölümü müfredatlarında yer bulması araştırmacı tarafından önerilmektedir.

Kaynaklar

[1] http://en.wikipedia.org/wiki/Object-relational_mapping

[2] Urban S. D. et al “A case study in mapping conceptual designs to object-relational schemas” *Concurrency: Practice And Experience* 2000; 12: 863–907

[3] Cobb M. A et al, “An OO Database Migrates to the Web” *IEEE Software* May/June 1998 22-30

[4] Jennings R., “Profesional ADO.NET 3.5 with Linq and Entity Framework”, Wiley Publishing, Inc. 2009

[5] <http://blog.taragana.com/index.php/archive/to-hibernate-or-not-a-commentary-on-orms-and-few-recommendations/>

[6] Anuja .K “Object Relational Mapping”, Master of Technology (M-Tech) partial fulfillment report for Department Of Computer Science Cochin University of Science And Technology 2007 KOCHI-682022

[7] http://en.wikipedia.org/wiki/Lazy_loading

[8] <http://www.bilgininadresi.net/Madde/2040/Katmanl%C4%B1-Mimari-nedir?-Faydalar%FD-nelerdir-?>

[9] http://en.wikipedia.org/wiki/List_of_object-relational_mapping_software

[10] <http://itboxing.devbg.org/>