BMB202. Veritabanı Yönetimi

Ders 3.
Varlık-İlişki Çizelgelerinin İlişkisel Veri Tabanına Dönüştürülmesi (Tablolar, UML)

Bilgisayar Mühendisliği Bölümü

Dersin Planı

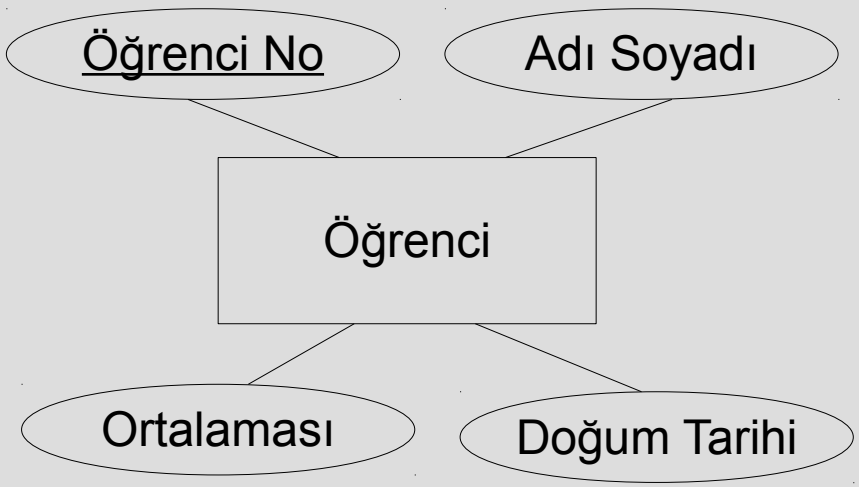
- E-R Çizelgesi >> İlişki Şeması (Tablolar)
- UML Sınıf Diyagramları

E-R Çizelgesi >> İlişki Şeması veya UML Diyagram

- Varlık-ilişki modeli kullanılarak veri modelleme yapıldıktan sonra, eğer veri tabanını gerçekleştirmek için ilişkisel bir VTYS kullanılacaksa, oluşturulan varlık-ilişki çizelgesinin tablolara (ilişki şemalarına) veya UML (Unified Modeling Language) sınıf diyagrama dönüştürülmesi gerekir.
- İlişki şemaları, VTYS yazılımları üzerinde yapılabilmektedir.
- UML ise nesneye yönelik programlama ile gelişmiş bir gösterim şeklidir ve programlama dili tarafında genelde tercih edilir. Birçok diyagram türü vardır.

E-R >> İlişki Şeması

- Varlık-ilişki modelindeki her varlık kümesi için ilişkisel modelde bir tablo oluşturulur.
- İlişkinin nitelikleri olarak da varlık kümesinin nitelikleri kullanılır.
  - Öğrenci (Öğrenci No, Adı Soyadı, Ortalaması, Doğum Tarihi)

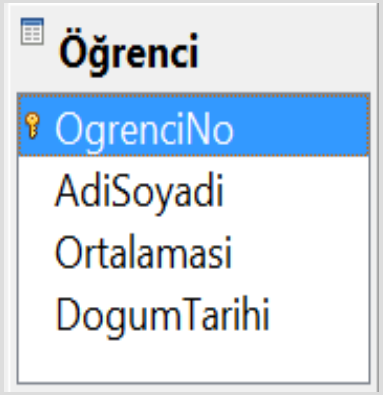


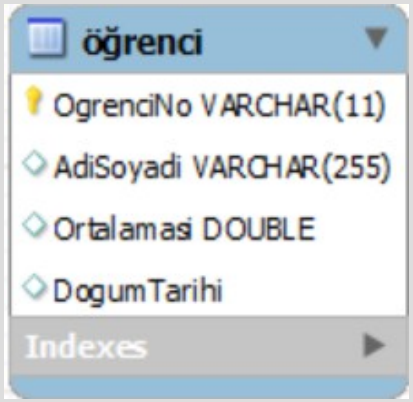
Öğrenci
Öğrenci No (PK)
Adı Soyadı
Ortalaması
Doğum Tarihi

Primary Key (PK) Birincil Anahtar

- Birincil anahtar hakkında bilgi verilmişti.
- E-R diyagramda altı çizili gösterim varken tablo gösteriminde PK kelimesi kullanılabilir. Bu kullanım VTYS yazılımına göre değişiklikler gösterebilir.
  - LibreOffice, MS Office Access, MS SQL Server, MYSQL Workbench: Anahtar resmi
  - (Tasarımda, Türkçe karakter kullanmamanız tavsiye edilir.)

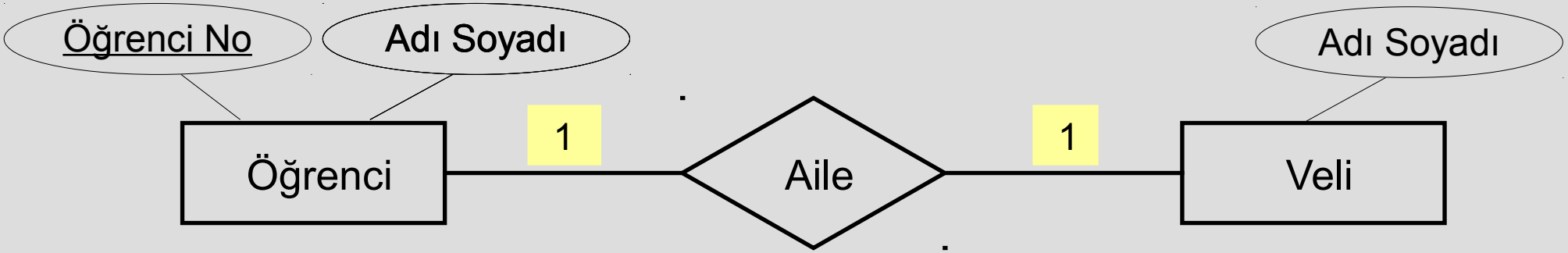
Ders
Öğrenci No (PK)
Adı Soyadı
Ortalaması
Doğum Tarihi

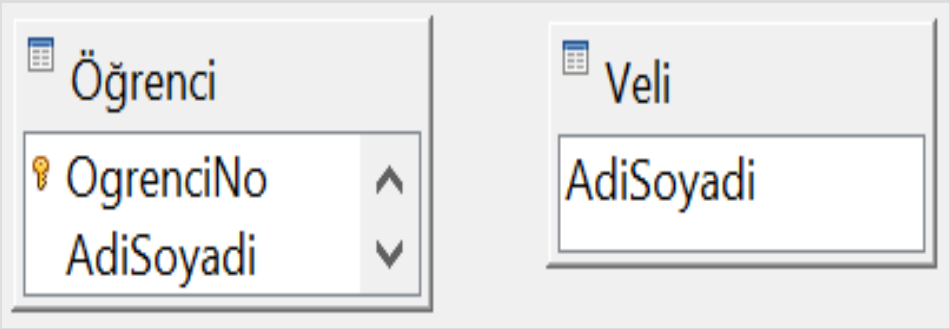




Birden-bire (1-1) ilişki tablo şeklinde gösterimi

- Bir öğrencinin bir velisi olur.
  - Aile bölümü için ek tablo mu?
  - Veli hangi öğrencinin?
  - Tek tabloda yapılabilir mi?



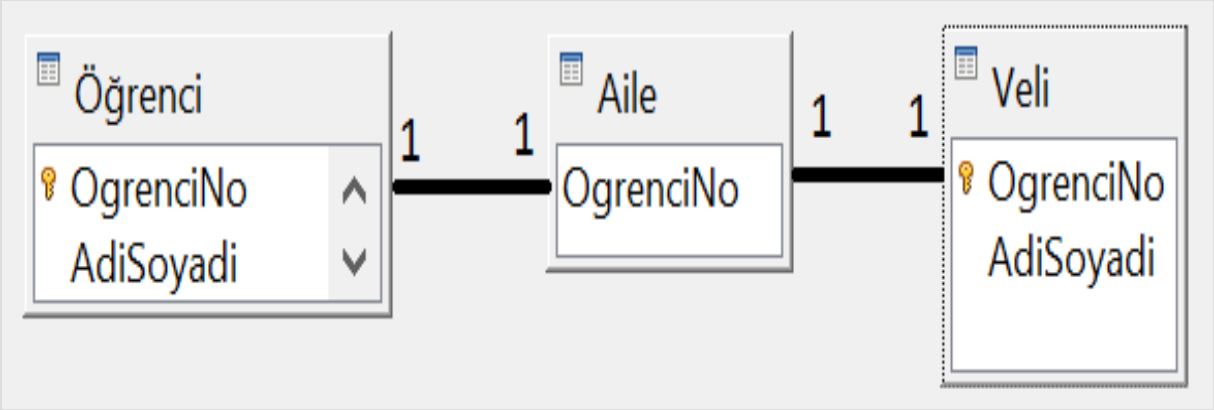


1-1

- Veri tablosu üzerinden düşünme

Öğrenci	
Öğrenci No	Adı Soyadı
11	Metin
10	Ali
7	Feyyaz

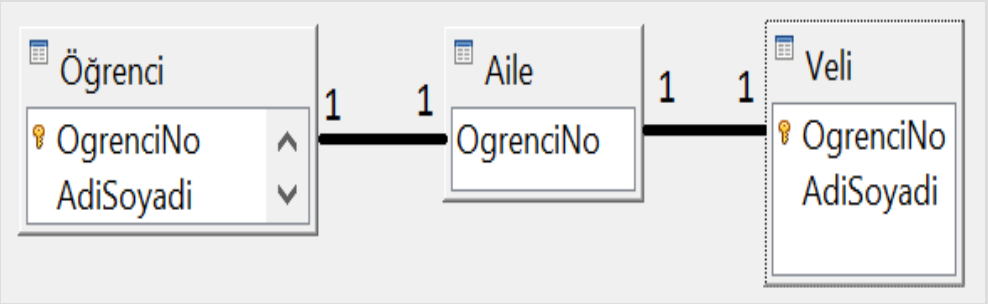
Veli
Adı Soyadı
Ayşe
Fatma
Zübeyde



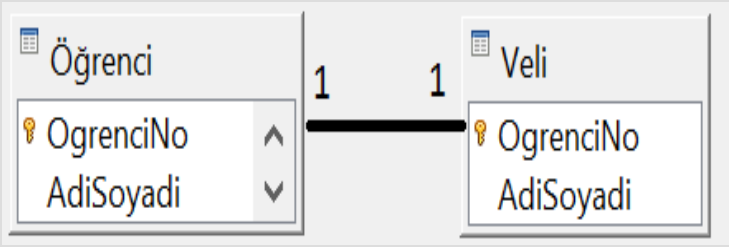
Sizce bu çözüm Doğru mu?

1-1

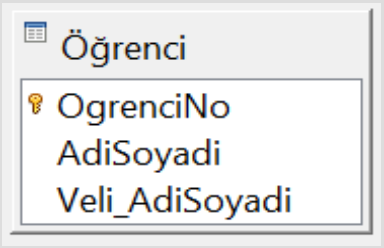
1



2



3



- Sizce üç şekil arasında fark var mıdır?
  - Hangisini seçersiniz?
- 2-3 arasında fark nedir?
  - Ne zaman ayıracağız?

1-1

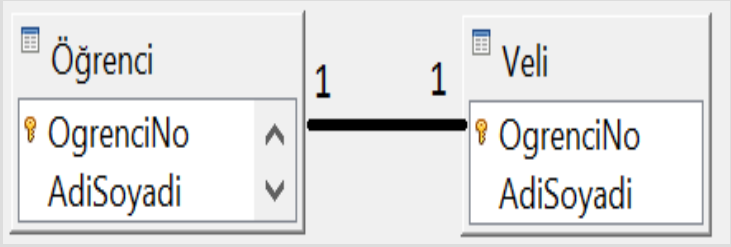
Öğrenci	
Öğrenci No	Adı Soyadı
11	Metin
10	Ali
7	Feyyaz

Veli
Adı Soyadı
Ayşe
Fatma
Zübeyde

Öğrenci	
Öğrenci No	Adı Soyadı
11	Metin
10	Ali
7	Feyyaz

1 — 1

Veli	
Öğrenci No	Adı Soyadı
10	Ayşe
7	Fatma
11	Zübeyde

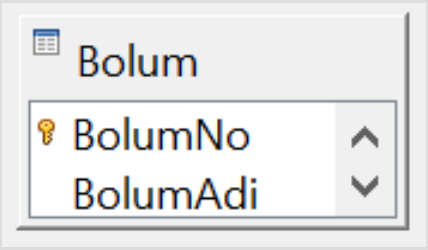


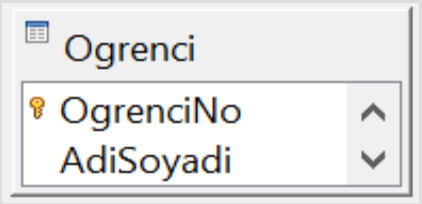
1-1 Sonuç

- Sonuçlar
  - Bire-bir ilişki tek tabloda ifade edilebilir.
  - Ancak özellik gereksiz ve az kullanılacak bir nitelik ise her ekleme işleminde boşu boşuna gereksiz nitelik alanı kadar alan açılır.
  - Bu sorun iki tablo ile engellenebilir.
  - Üç tablo ile 1-1 ilişki kurmak son derece gereksiz ve tercih edilmeyen bir işlemdir.
  - Gereksiz nitelik gerekli nitelik haline geldiğinde tek tabloya geçiş tavsiye edilebilir.
    - Her zaman tek tabloyu yönetmek iki tabloyu yönetmekten daha kolaydır.

Birden-çoğa (1-n) ilişki tablo şeklinde gösterimi

- Bir bölümde okuyan öğrenciler
  - 3 tabloya gerek var mı?
  - 2 tablo olur mu?
  - Tek tablo yeterli mi?



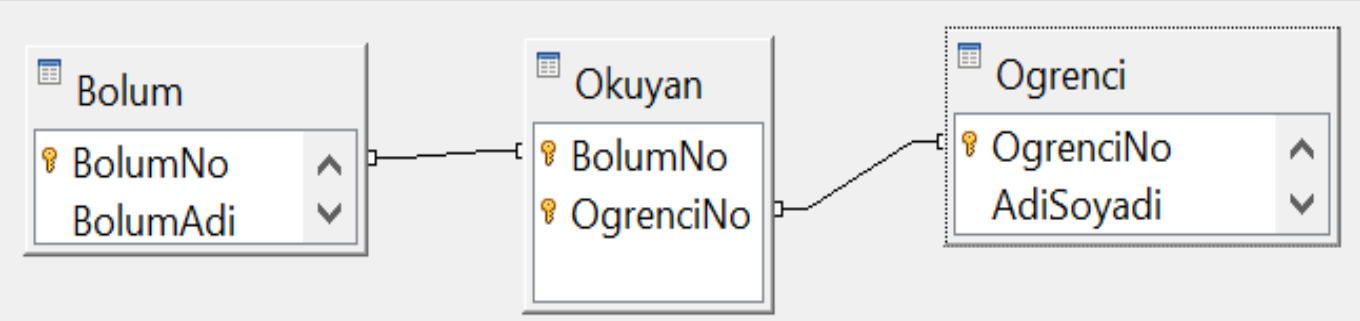


1-n

- Bölüm tablosunda Bölüm No'ya ihtiyaç var mı?

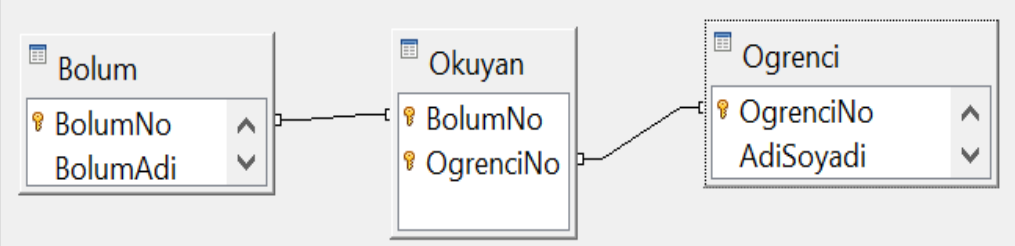
Bölüm	
Bölüm No	Bölüm Adı
1	Bilgisayar Müh.
2	Endüstri Müh.
3	Yazılım Müh.

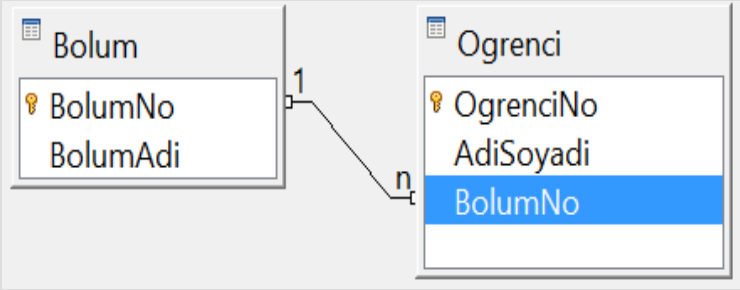
Öğrenci	
Öğrenci No	Adı Soyadı
11	Metin
10	Ali
7	Feyyaz



Sizce bu çözüm Doğru mu?

1-n

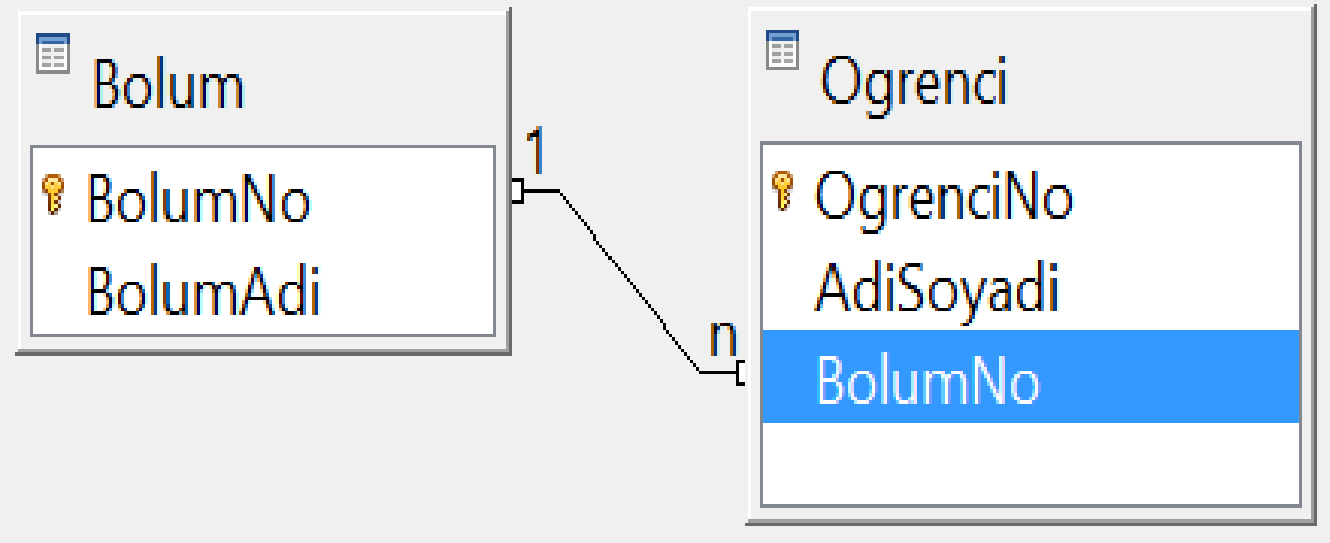




- Ogrenci tablosundaki BolumNo **ikinci anahtar** olarak isimlendirilir. Tekrarlayabilir. Bu sebepten dolayı Ogrenci tablosu yanında “**n**” eklenir.
- Sizce hangisi daha doğru yaklaşım
  - BolumNo doğru yerde mi?
  - Bolum içine OgrenciNo eklesek olmaz mı?

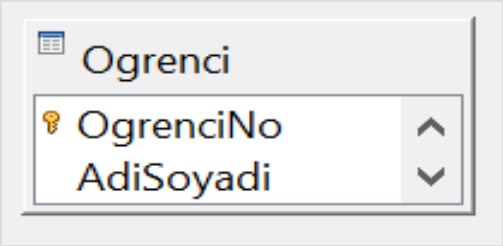
1-n Sonuç

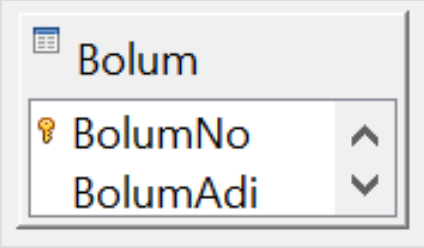
- Sonuçlar
  - Bire çok ilişki için iki tablo yeterlidir.
  - İlişki için ikinci anahtar doğru tabloya eklenmelidir.
    - İkinci anahtarın eklendiği tablo “n” özelliğine sahiptir.



Çoktan-bire (n-1)

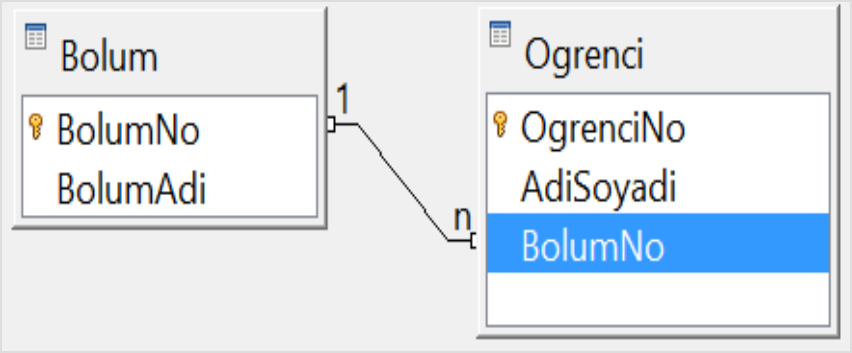
- Öğrenci – Bölüm İlişkisi
  - 3 tabloya gerek var mı?
  - 2 tablo olur mu?
  - Tek tablo yeterli mi?

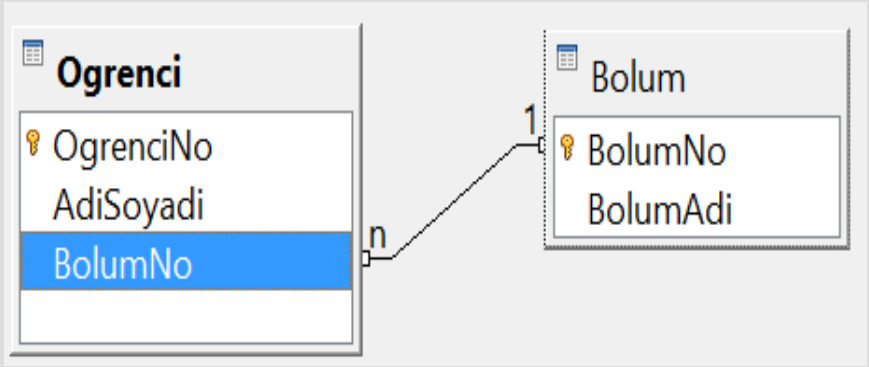




Çoktan-bire (n-1) Sonuç

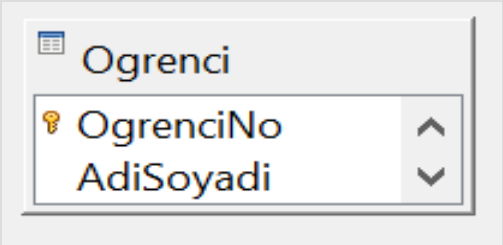
- Sonuçlar
  - İki tablo yeterlidir.
  - Görüldüğü gibi 1-n ile n-1 ilişki sadece bağlantının ara ismini değiştirir.
    - 1-n: okuyan
    - n-1: okuduğu

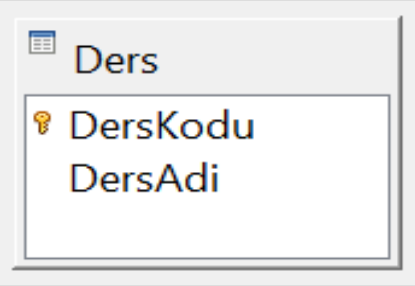




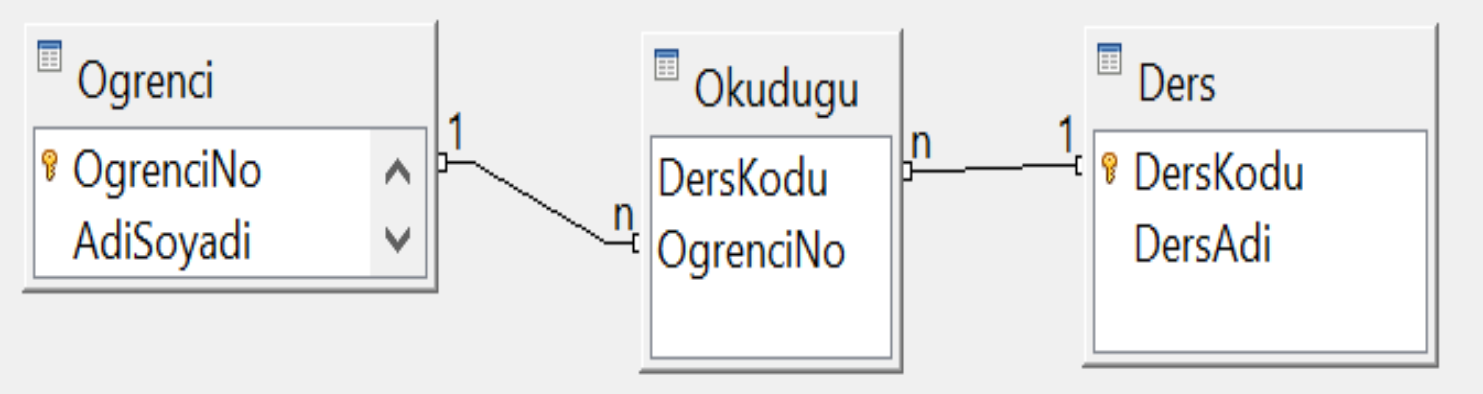
Çoktan-bire (n-1)

- Öğrenci – Bölüm İlişkisi
  - 3 tabloya gerek var mı?
  - 2 tablo olur mu?
  - Tek tablo yeterli mi?





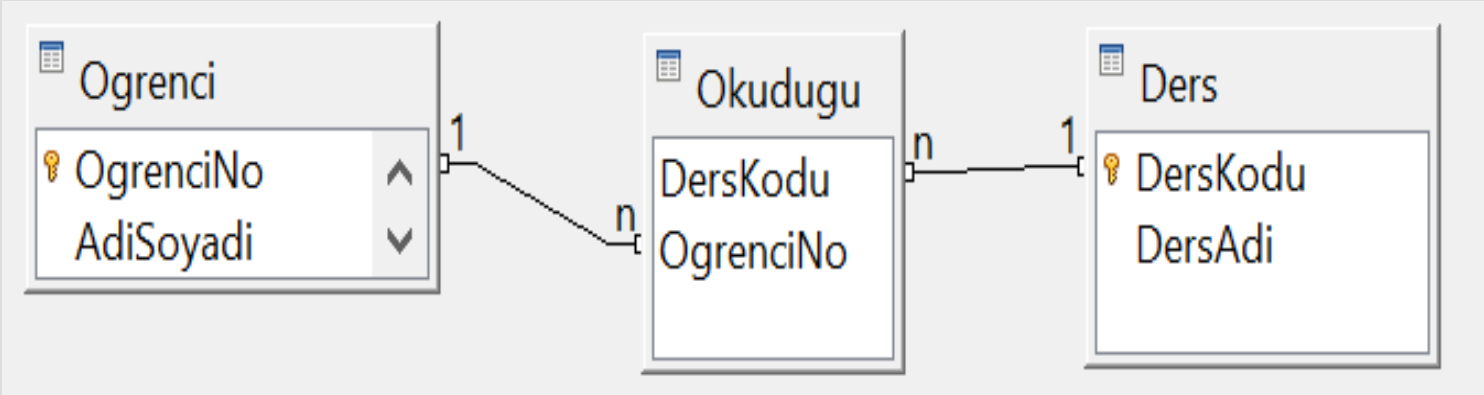
n-m



- “Okudugu” tablosu olmadan ilişki kurulabilir mi?
- Okudugu tablosunda “DersKodu” ve “OgrenciNo” birlikte birincil anahtar seçilmesi sizce doğru mu?
- İki tabloda bu ilişkiyi tanımlamak mümkün mü?

n-m Sonuç

- Sonuçlar
  - En az üç tablo gereklidir.
  - Ara tablonun ikincil anahtarın her ikisi de birlikte birinci anahtar seçilebilir. Bu sayede tekrar önlenir.



UML (Unified Modelling Language)

- UML yazılım sistemlerinin olgularını
  - tanımlamak,
  - görselleştirmek,
  - oluşturmak ve
  - dokümante etmek için

  kullanılan bir standartlar dilidir.
- UML, OMG (Object Management Group) tarafından nesne yönelimli programları modellemek üzere standart olarak kabul edilmiştir.
- UML, yazılım projelerinin tasarımını daha çok grafiksel öğeler ile belirtir.
- UML kullanımı ekip içerisindeki iletişimi güçlendirir, potansiyel tasarımların ortaya çıkmasını sağlar ve yazılımın mimari tasarımını ortaya koyar.

• UML’in amaçları

- Kullanıcılara geliştirip paylaşabilecekleri anlamlı bir görsel modelleme dili sunmak.
- Temel kavramları genişletmeye yönelik olarak genişleyebilir ve özelleştirilebilir bir mekanizma sunmak.
- Uygulama geliştirme dillerinden ve geliştirme süreçlerinden bağımsızlığı sağlamak.
- Modelleme dillerini anlamak üzere biçimsel bir zemin hazırlamak.
- Bileşen, pattern, framework ve bir arada yürütülen projelerin ileri seviyede geliştirilmesine destek sağlamak.

UML Diyagram Tipleri

- UML'de 9 farklı diyagram tipi vardır:
  - Sınıf (Class) ***
  - Nesne (Object)
  - Kullanım Durumu (Use-Case)
  - Sıralı (Sequence)
  - İşbirliği (Collaboration)
  - StateChart
  - Activity
  - Bileşen (Component)
  - Dağıtım (Deployment)
- Bu derste sadece Sınıf diyagramları üzerinde duracağız. Diğer diyagramlar çeşitlerini diğer derslerinizde göreceksiniz.

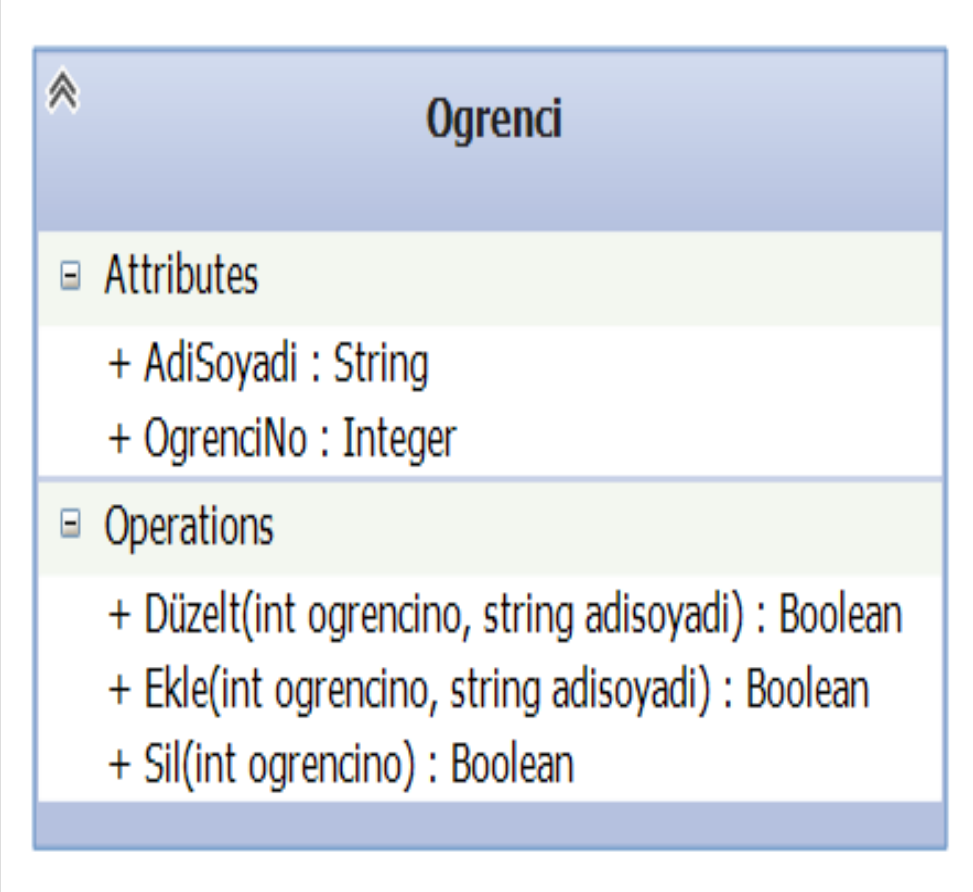
UML Sınıf Diyagramı

- Sınıf diyagramları hemen her nesne yönelimli yaklaşımlı programlama dillerinde çoğunlukla kullanılır.
- Bu diyagramları kullanarak programınızı sahip olduğu tüm sınıflarınızı yine sahip oldukları özellikleri ile gösterebilirsiniz.

UML Sınıf Bileşenleri

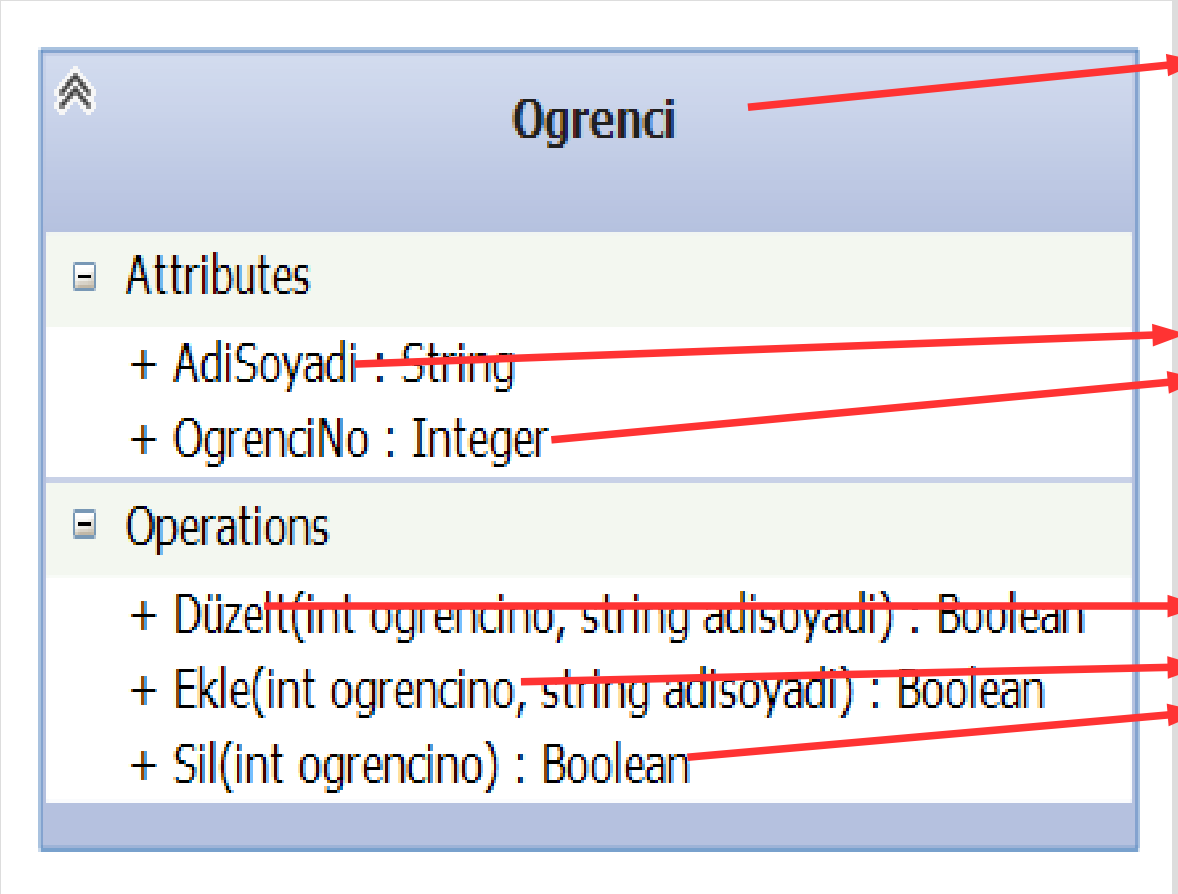
- İsim (Name)
- Nitelikler (Attributes)
- İşlemler (Operations)

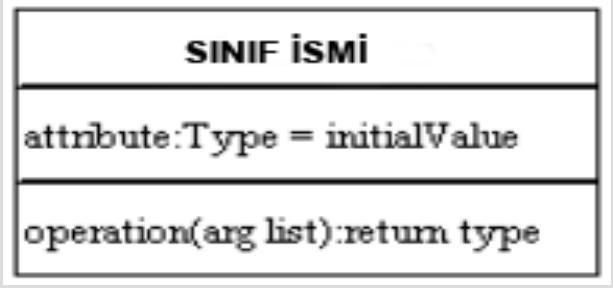
- İlk örnekte görüldüğü gibi “İlişkisel şema (tablolara)” göre işlemler seklemesi gelmiştir.
- Bu eklentinin özellikle uygulama geliştirmede çok yararı olacaktır.
- Ayrıca, bu diyagramlar otomatik olarak koda çevrilebilir.



UML Class Diyagram - Microsoft Visual Studio

UML Sınıf Diyagramı



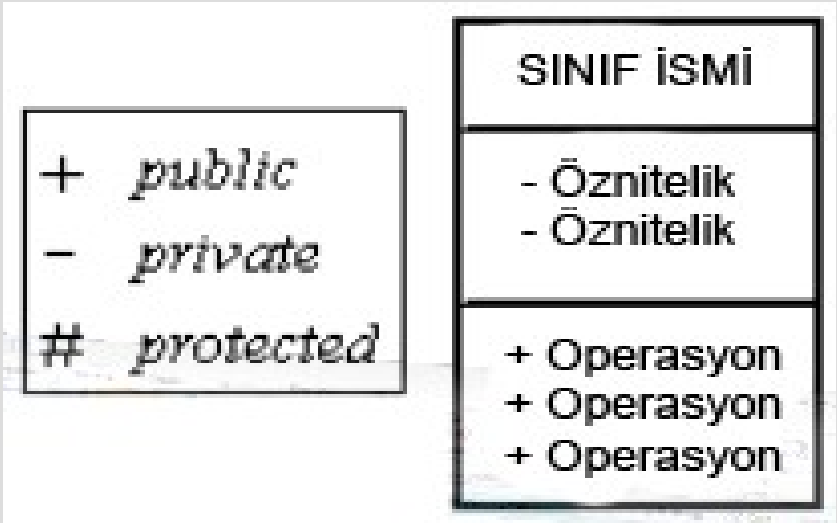


Sınıflar

- Sınıflar iki ana sınıfa ayrılmışlardır.
  - Aktif Sınıf: o sınıfın hareketliliğini/aktifliğini göstermektedir.
  - Pasif Sınıf: daha çok diğer sınıflara veri sunmak amacı ile kullanılırlar
    - Bir sınıfın aktif olduğunu belirlemek için yapmanız gereken ise diyagramınızı hazırlarken daha kalın bir çerçeve ile çizmektir.

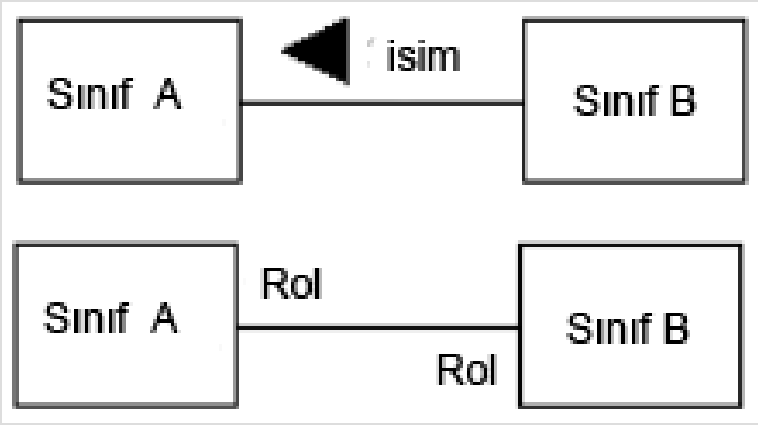
Erişilebilirlik / Görünürlük (Visibility)

- Public
- Private
- Protected



Bağlantılar (Associations)

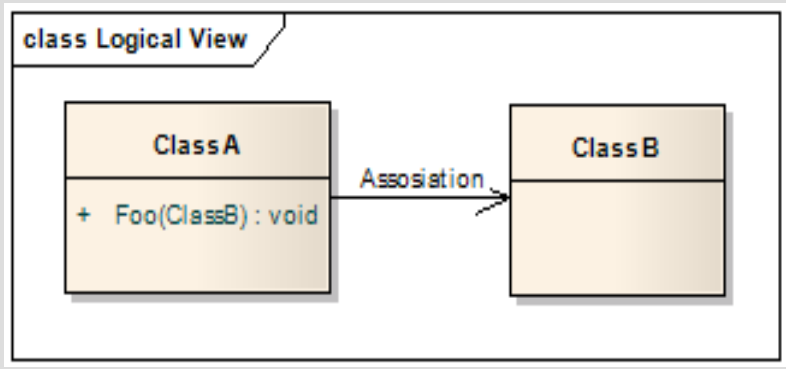
- Sınıf diyagramlarında bulunan işbirliği işaretçisini sınıflarınız arası statik (static - durağan) işbirliklerini göstermek amacı ile kullanabilirsiniz.
- Bağlantı adını bağlantı çizgisinin üst kısmına, üzerine yada altına yazabilirsiniz.
- Ok ile bağlantının yönünü belirleyebilirsiniz.
- Rolleri ise her iki sınıfa yakın yerlere yazabilirsiniz.
- Unutmayın ki hem bağlantıya isim vermek hemde rolleri yazmak pek sık başvurulan bir yöntem değildir.



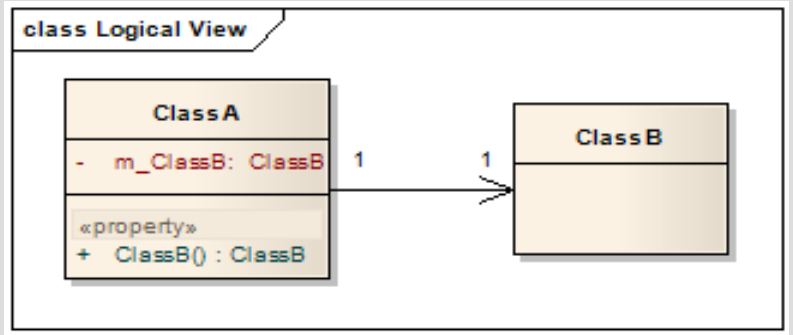
Bağlantılar (Associations)

- İki türlü iş birlikteliği söz konusudur.
  - Zayıf Bağlantı (Weak Association): ClassA işlemlerinden biri ClassB örneğinin parametre içerir veya ClassB örneğini döndürür göstermek için ClassB bağlı olabilir.
  - Güçlü Bağlantı (Strong Association): ClassA de ClassB örneği başvuru içerir olduğunu göstermek için ClassB bağlı olabilir.

Zayıf

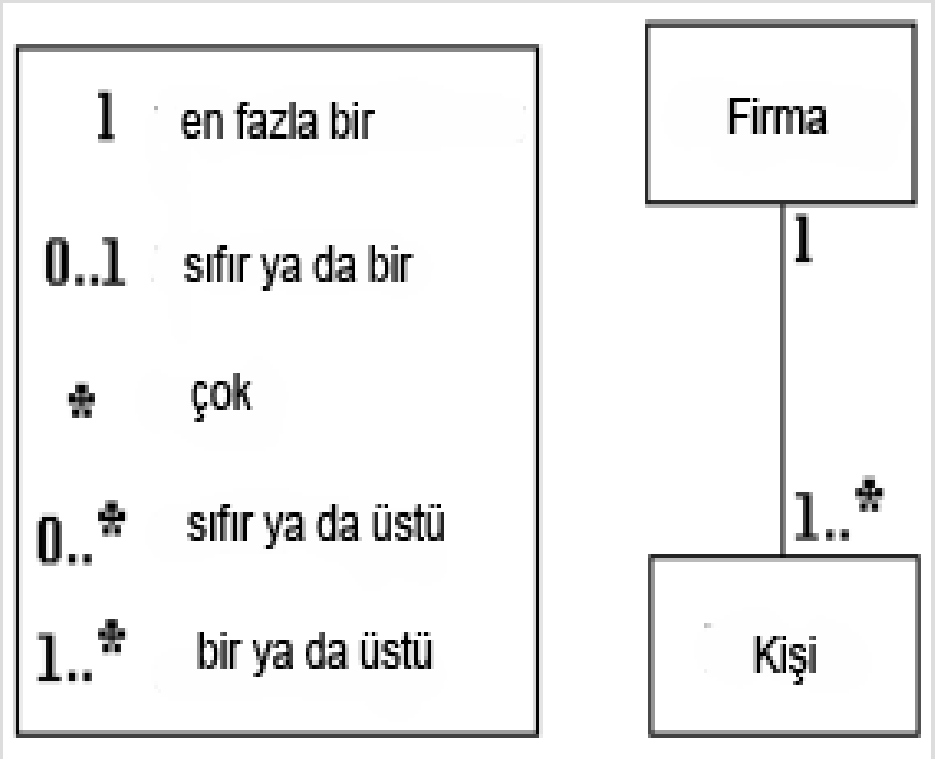


Güçlü



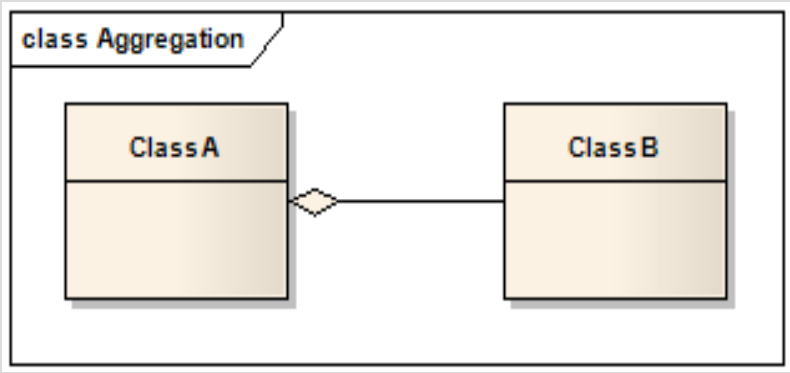
Çokluluk/Çeşitlilik (Multiplicity (Cardinality))

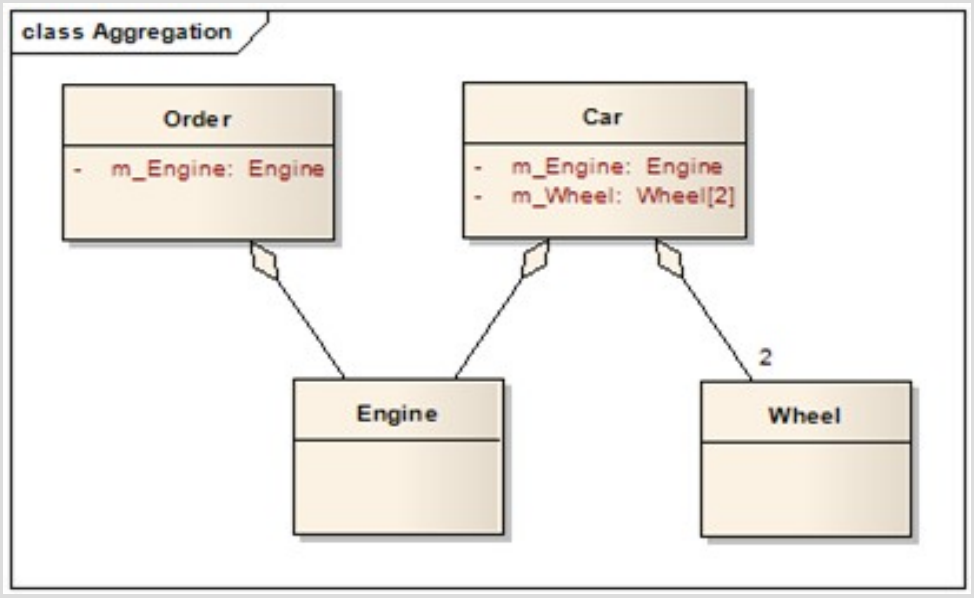
- Veritabanındaki ilişkilerin genişletilmiş şeklidir.
- Çokluluk işaretini bağlantının sonuna yakın bir yere yerleştirmelisiniz.
- Bu semboller bir sınıf örneğinin diğer bir sınıf örneğine bağlantı sayısını göstermektedir.
  - Örnek olarak bir firmanın birden fazla çalışanı yada ofisi yada deposunun olabilmesini gösterebiliriz.



Aggregation (Toplanmış) (Shared Association)

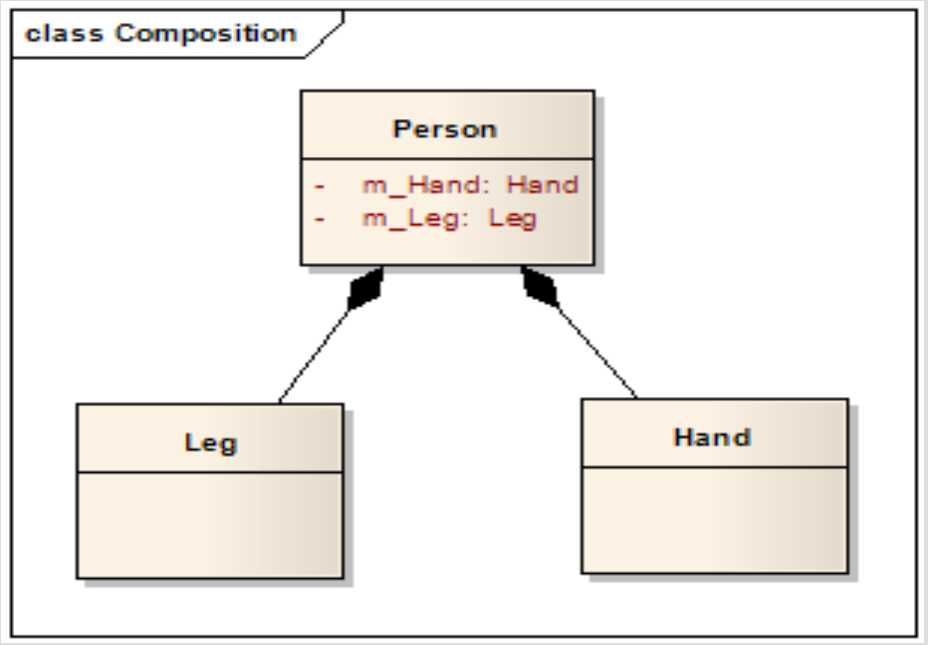
- ClassA (tamamı) ve ClassB (parçası) arasındaki ilişkide Class B diğer sınıflarca toplanabilir.
- Bu durumda normal bağlantı linki yerine toplama linki kullanılır.
- Toplamayı ise içi boş bir eşkenar dörtgen ile gösteririz.
- Burada dikkat edilmesi gereken nokta ise "tümü" ile ifade edilen sınıfın "kısmi" olarak ifade edilen sınıfa göre daha önemli bir rol oynadığıdır ve ayrıca, her iki sınıfta birbirlerine bağımlı değildirler.
- Eşkenar dörtgenin "tümü" ile ifade edilen sınıfa yakın olarak yerleştirilmesi gerektiğini unutmamalısınız.





Composition (Çalışma) (Not-Shared Association)

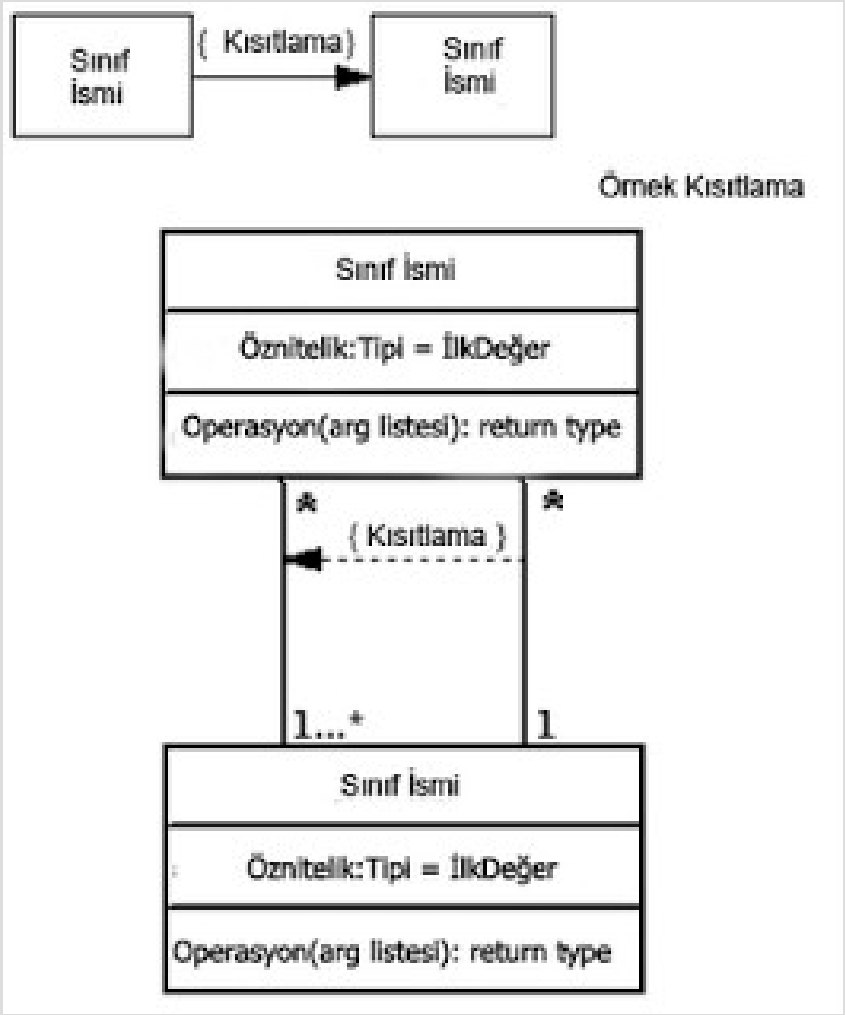
- Çalışma, Toplamanın özel bir tipidir ve Sınıf A nın 'tümü'nün Sınıf B'nin bir 'kısmı'na karşı güçlü sahipliğini gösterir.
- Çalışma, içi dolu bir eşkenar dörtgen ile gösterilir.



Kısıtlama (Constraint)

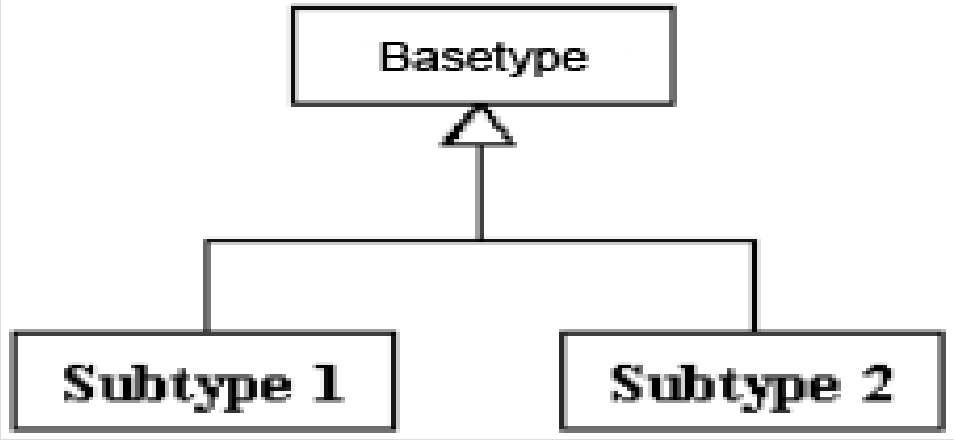
- Kısıtlamayı oklu parantez {} işaretlerinin arasında yazarak gösterebilirsiniz.
  - Örneğin bir sınıfın soyut olduğunu ifade eden şey: {abstract} kısıtlamasıdır.
  - {frozen} bir değişkenin değerinin asla değişemeyeceği anlamına gelir
- +baslangic:Date=Now1

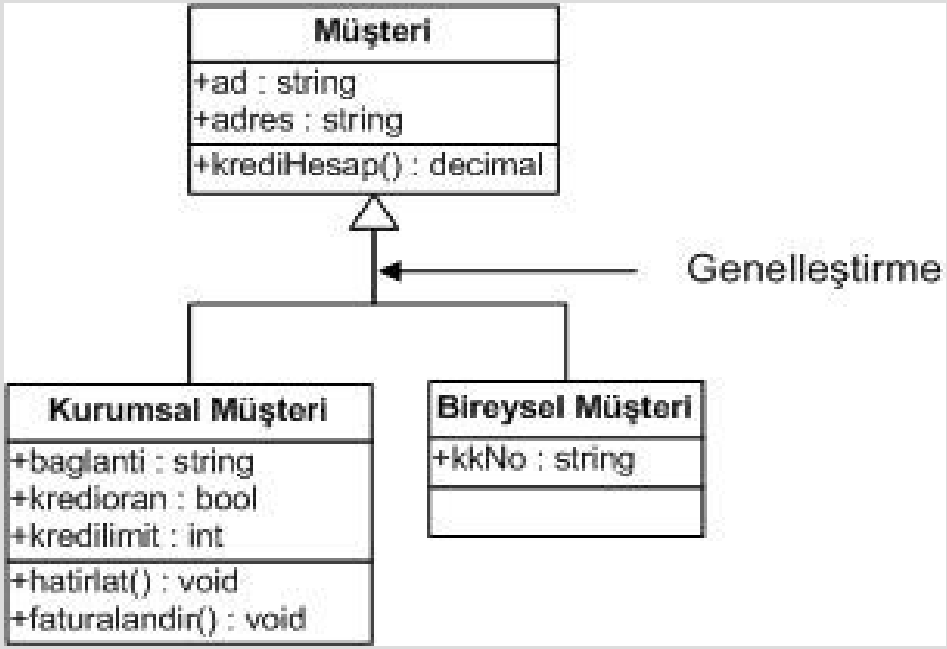
  +bitis: Date=Now2{>=Baslangic}
  - İkinci tarih başlangıçtan büyük olmalıdır kısıtı.



Genelleştirme (Generalization)

- Genelleştirme miras'ın (inheritence) veya 'is a' (sahiplik) bağlantısının bir diğer adıdır.
- İki sınıf arasındaki bağlantıda birinin diğerinin özelleştirilmiş şekli olduğunu ifade eder.
  - Örnek olarak; Boeing 777 bir uçak markasıdır. Bu örnekte 'Boeing 777' uçak sınıfının özel bir çeşididir.

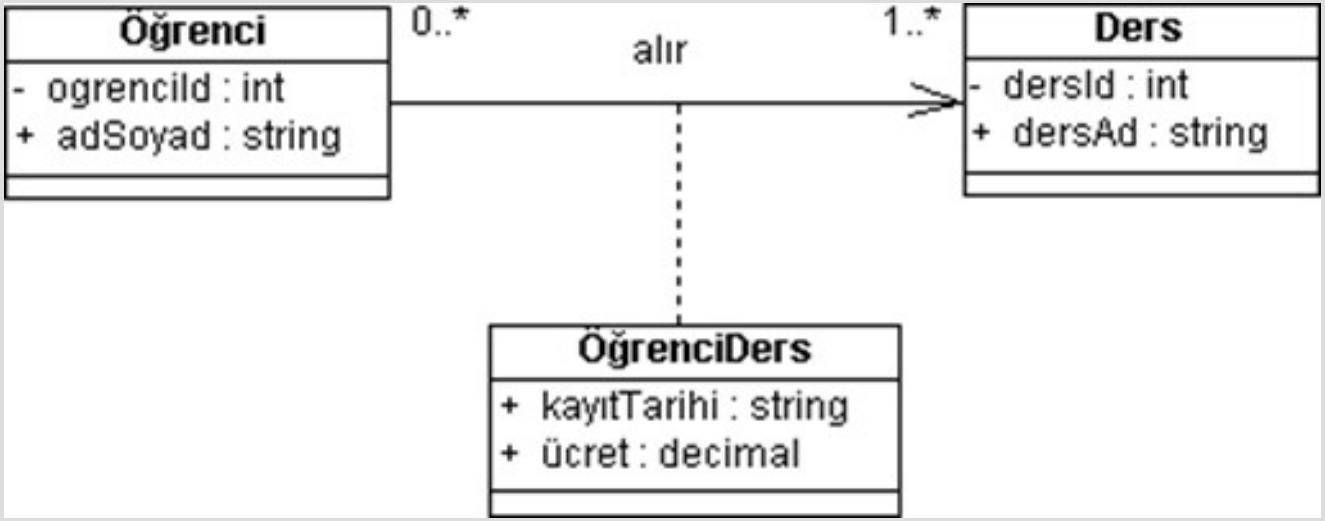




Örnek 1

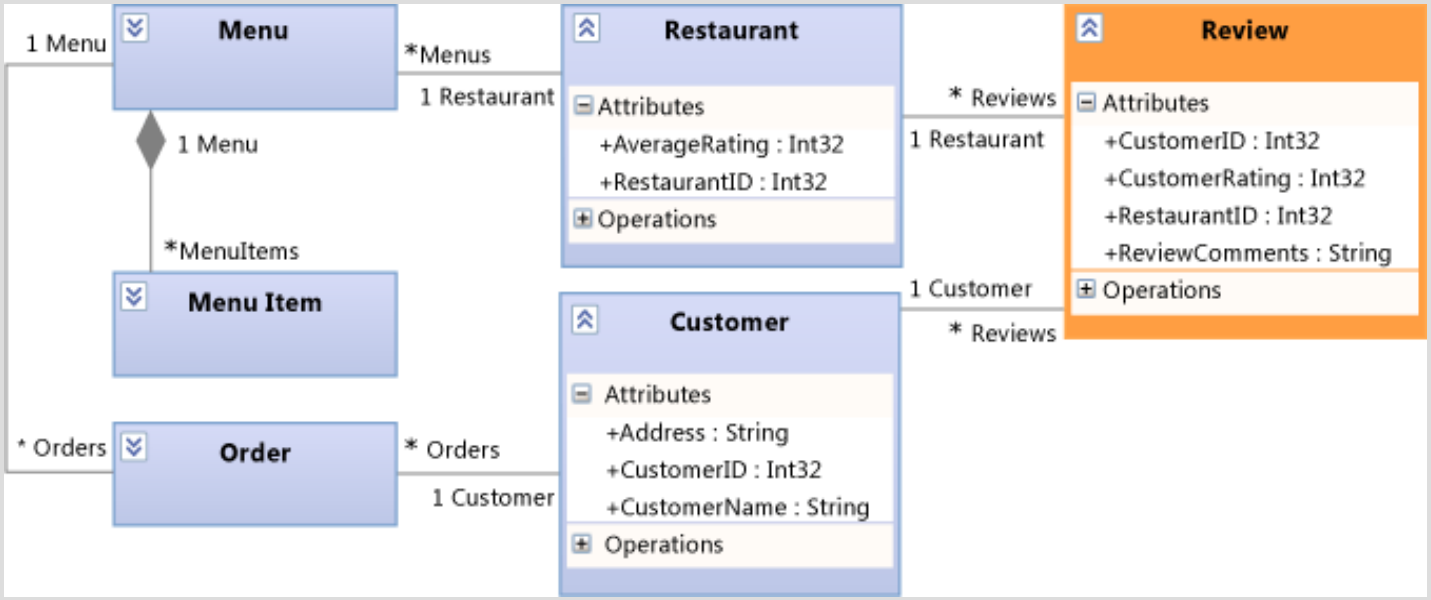
- Öğrenci, Ders ilişkisini
  - E-R diyagramı
  - İlişkisel Şema (Tablo)
  - İlişkisel Model (Veri içeren tablo)
  - UML Sınıf Diyagramı

  şeklinde çiziniz.



Örnek 2

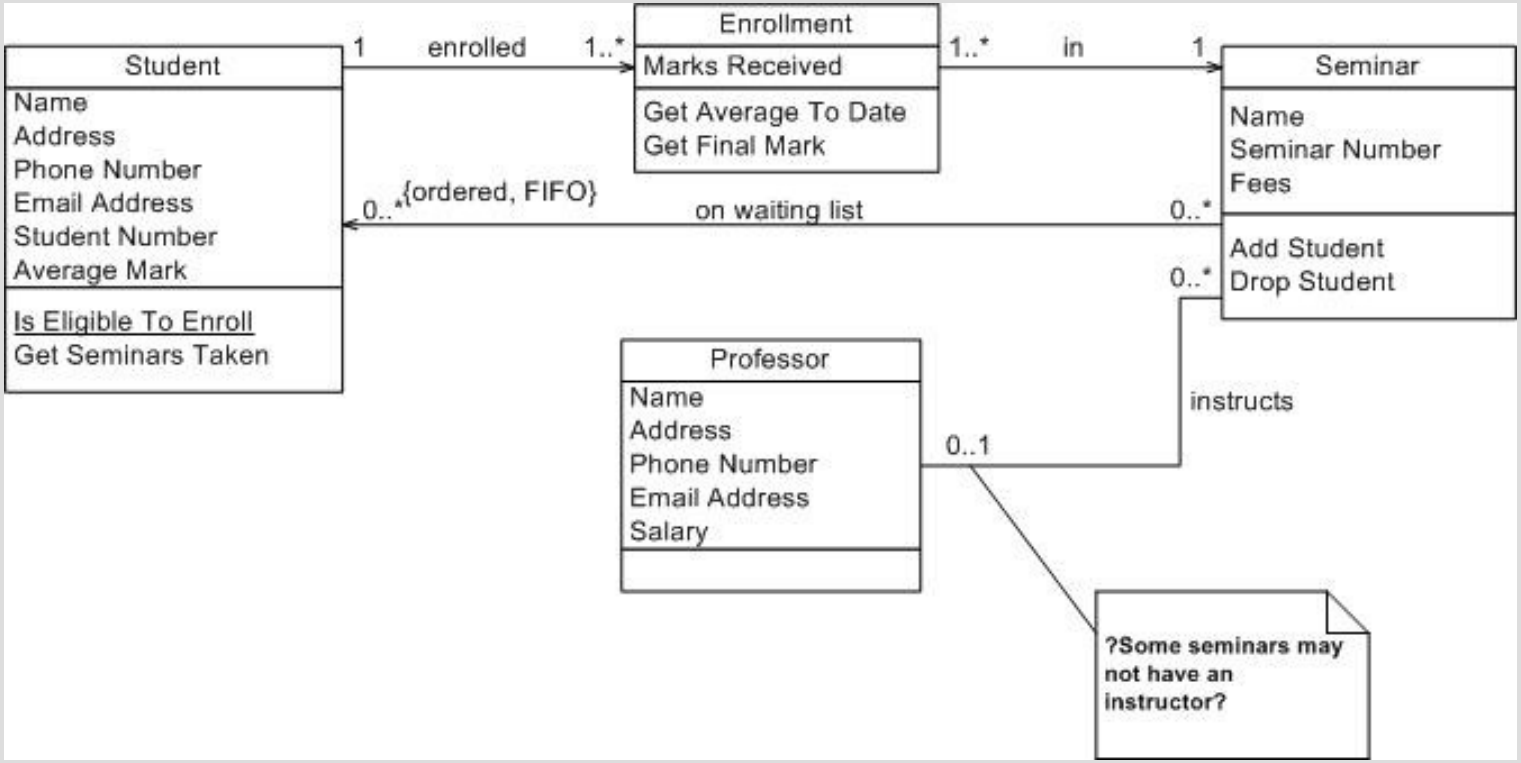
- Restoran hakkında yorum paylaşımı yapan bir web sitesi düşünüyorsunuz.
  - UML Diyagramı şeklinde tasarlayın.



Restoran uygulamanıza yenileme yapmayı düşündünüz:
Menu, Menü İçeriği ve Müşteri İstekleri (Order) eklemeyi düşünüyorsunuz.
Acaba, yukarıda kapalı olarak gösterilen bölümlere nasıl özellikler eklersiniz.

Örnek 3

- Bir profesör, bir seminer sistemi geliştirmeyi düşünmüştür.
- Bu sistemde öğrenci istediği seminere kayıt olacak (enrollment).
- Profesör, kayıt olan öğrencilerden istediklerini seminere alacak, istemediklerini çıkarabilecek.



Kaynaklar

- http://www.smartdraw.com/resources/tutorials/uml-class-diagrams/#/resources/tutorials/Introduction-to-UML
- http://yazilimdizayn.blogspot.com.tr/2012/09/
- http://umlnedir.blogspot.com.tr/2009/08/umle-giris.html
- http://aviadezra.blogspot.com.tr/2009/05/uml-association-aggregation-composition.html