

# *2.Temel Kavramlar*

*OLAP,OLTP*

*Veri Güvenliği*

*Veri Tabanı Yönetim Sistemlerinin Sınıflandırılması*

*Neden veri tabanı kullanılır?*

*Veri Tabanı Yaklaşımının Avantajları*

*Veri Tabanı Yapısı*

*Veri Tabanı Nesneleri*

# *Kullanım amacına göre veri tabanları*

*Kullanım amacına göre veri tabanı çözümlerini iki gruba ayırılır;*

- ***OLTP*** (*On Line Transactional Processing*)( *Açık İşlemsel süreçler*)
- ***OLAP*** (*On Line Analytical Processing*- *Çevrimiçi analitik işleme*),

# OLTP

- *(On Line Transactional Processing)( Açık İşlemsel süreçler) :Canlı sistem. veri girişi, veri güncelleme, veri silme gibi işlemlere olanak tanıyan sistemlerdir.*
- *Üzerinde sürekli işlem yapılan veri tabanları sistemleridir. insert, update, delete şeklilde DML işlemleri içeren veri depolama sistemleridir. OLTP sistemlerde sürekli yoğun işlemler yapılır.*

# OLAP

- *(On Line Analytical Processing- Çevrimiçi analitik işleme), Veri madenciliği veriyi depolayan, raporlama ve analiz için kullanılan, veriye hızlı erişim sağlayan yapılardır.. İş zekası çözümleri için OLAP kullanılır.*
- *OLAP fazla miktarda verinin iki boyutlu tablolar yerine çok boyutlu küpler biçiminde tutulduğu ve bu veriler üzerinde istatistiksel analizlerin yapılabilirdiği veri tabanlarını ifade eder. Özellikle karar destek sistemlerinde kullanılan bu tür veri tabanlarının arkasında çoğu zaman yine OLTP türü veri tabanları bulunmaktadır.*
- *Ayrıca OLAP veri tabanları eş zamanlı (concurrent) birden çok kullanıcıya OLTP'ye göre çok daha performanslı sorgu sonuçları üretebilir.*
- *Var olan verileri anlamlandırmayı ve raporlamayı sağlar.*

## *Veri güvenliği kavramı*

- *Veri güvenliğinin konusu, veri tabanını, dolayısıyla veriyi yetkisiz kullanımlara karşı korumaktır. Bu konuda çok çeşitli yaklaşımlar vardır. Hem verinin istenmeyen şekilde değiştirilmesine veya zarar görmesine hem de yetkisiz kullanımlara engel olmak gerekir. Bunu sağlamak için bazı genel teknikler geliştirilmiştir.*

- Kullanıcıların tanımlanması: Çok kullanıcıli ortamlarda farklı yetkilere sahip kullanıcılar vardır. Farklı yetkilere sahip kişilerin, veri tabanında ulaşabilecekleri veri farklıdır. Örneğin, bilgisayara veri girişı yapan bir işletimcinin, kurumun muhasebe kayıtlarına, muhasebe müdürü kadar yetkiliymiş gibi girerek değışiklikler yapması engellenmelidir. Bu amaçla, hangi kullanıcıların hangi yetkilerinin olduđu ve bu yetkilerini kullanabilmek için gerekli şifreler daha önceden tespit edilmelidir.

- Fiziksel koruma: Şifre sisteminin yeterli olmadığı durumlarda, verinin fiziksel koruma altına alınması gerekir. Yangın veya hırsızlığa karşı verinin yedeklenmesinin yapılması gibi
- Ağ Koruması: Firewall IPX vb uygulamalar ile ağ üzerinden veritabanına gelebilecek saldırılara karşı güvenlik önlemlerini almak.

- Kullanıcı haklarının temin edilmesi: Sistemde hangi kullanıcının hangi yetkilere ve haklara sahip olduğu ve neler yapabileceğinin önceden belirlenmiş olması gerekir. Bir kişinin yetkisini veya hakkını başka bir kişiye vermesi ise, sistemin müdahalesi dışında gerçekleşen bir durumdur.
- Özellikle veri tabanının sorgulanmasında güvenlik problemleri ortaya çıkmaktadır. Hangi tür kullanıcının, hangi sorgu tiplerini sisteme yöneltebileceğinin daha önceden tespit edilmesi gerekmektedir. Fakat, yukarıda bahsedilen önlemlerden hiç biri tam bir koruma sağlamaz. Bu yüzden, birden fazla önlem kullanarak güvenlik artırılabilir.

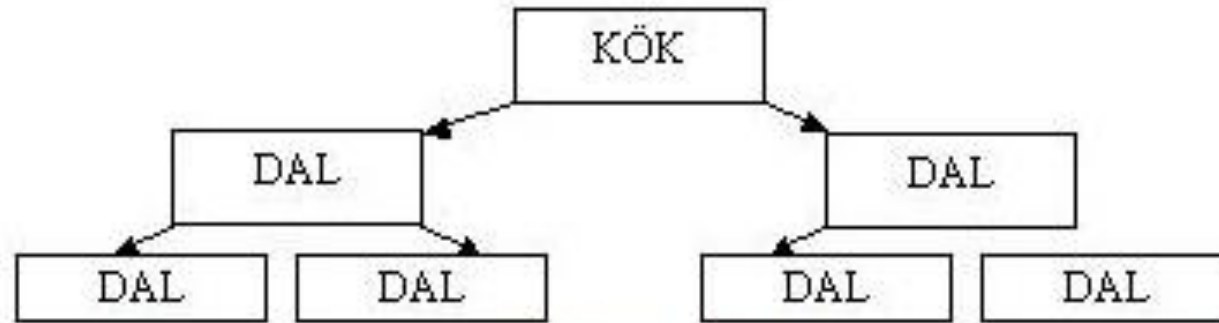


# *Veri Tabanı Yönetim Sistemlerinin Sınıflandırılması*

- *Veri Modeline Göre*
  - *Hiyerarşik*
  - *Ağ*
  - *İlişkisel*
  - *Nesneye Yönelik*
- *Kullanıcı Sayısına Göre*
  - *Tek kullanıcı*
  - *Çok kullanıcı*

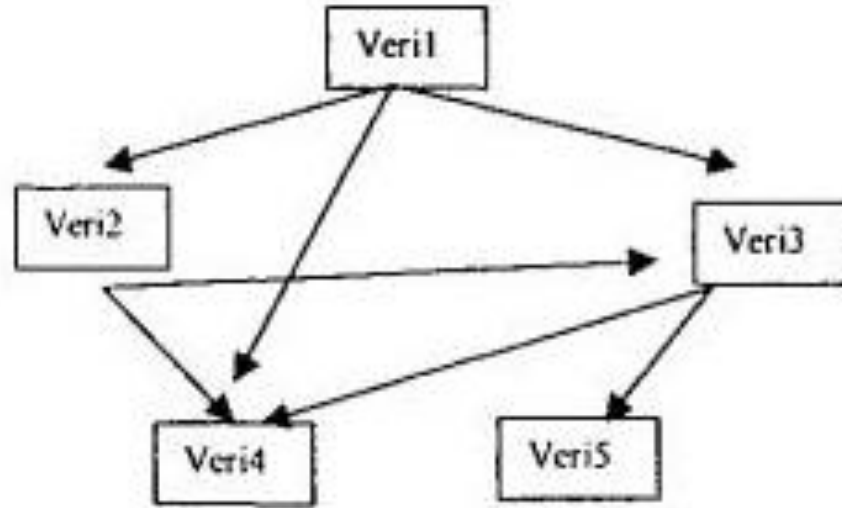
# *Hiyerarşik veri tabanları*

- *Veri tabanları için kullanılan ilk modeldir*
- *Hiyerarşik veri tabanları bilgileri bir ağaç yapısında saklarlar.*



# Ağ Veri Tabanları

- *Hiyerarşik veri tabanları yetersiz kalınca 1960'ların sonunda verilerin ağaçların daha gelişmiş hali olan graflar şeklinde saklandığı yapı ortaya çıkmıştır.*



# *Nesneye Yönelik veri tabanları*

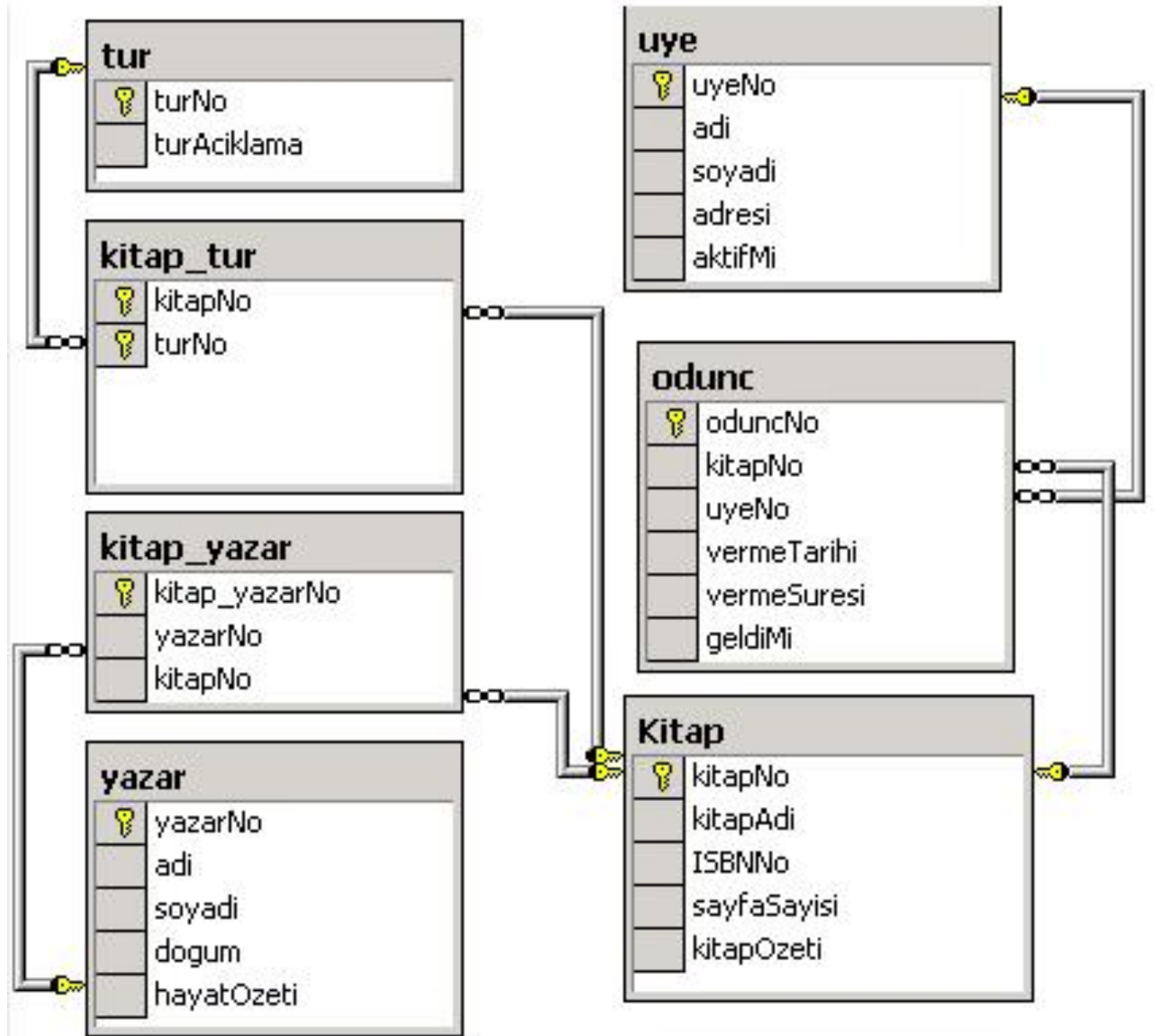
- *Günümüzdeki pek çok kelime işlemci ve hesap tablosu programında kullanılan nesneler artık veri tabanlarında da kullanılmaktadır.*
- *Nesneye yönelik veri tabanı C++ gibi nesneye yönelik bir dille oluşturulan ve yine bu tarz bir dille kullanılan veri tabanı anlamına gelir.*

# *İlişkisel veri tabanları*

- *1970'lerin başında geliştirilmiştir. Daha önceki veritabanı sistemlerinde veriler üzerinde değişiklik yapmak oldukça zordu. Veriler arasında bütünlüğü sağlamak amacıyla verilerin eşleştirilmesi ve ilişki kurulması gerektiğinde mevcut sistemler yetersiz kalmakta ve güvenlik sağlanamamaktaydı. Bu gibi durumları ortadan kaldırmak için ilişkisel veri tabanlarına ihtiyaç duyulmuştur.*
- *Tablolar arasındaki bağlantılar matematiksel ilişkilerle gösterilir.*
- *Günümüzdeki hemen hemen tüm veri tabanı programları bu yapıdadır.*

- *Günümüzde en çok kullanılan bu ilişkisel veri tabanlarında veriler; tablolar, satırlar ve alanlar şeklinde tutulmaktadır. Bu sayede veriler arasında ilişkilendirmeler yapılabilmekte, verilerin kullanımı ve yönetimi daha verimli bir hâle getirilmektedir.*
- *Günümüzde en sık kullanılan ilişkisel veritabanı yönetim sistemleri arasında Microsoft SQL Server, Oracle, Microsoft Access, PostgreSQL ve MySQL gösterilebilir.*

# İlişkisel veritabanları



# *Neden veri tabanı kullanılır?*

- *Verilerin tutulması, saklanması ve erişilmesinde geleneksel yaklaşım verilerin ayrı ayrı dosyalarda gruplanması yaklaşımını kullanmaktadır.*
- *Verilerin artması, verilere aynı anda erişme ve düzenlenme ihtiyacı ile geleneksel yaklaşım yetersiz kalmıştır.*



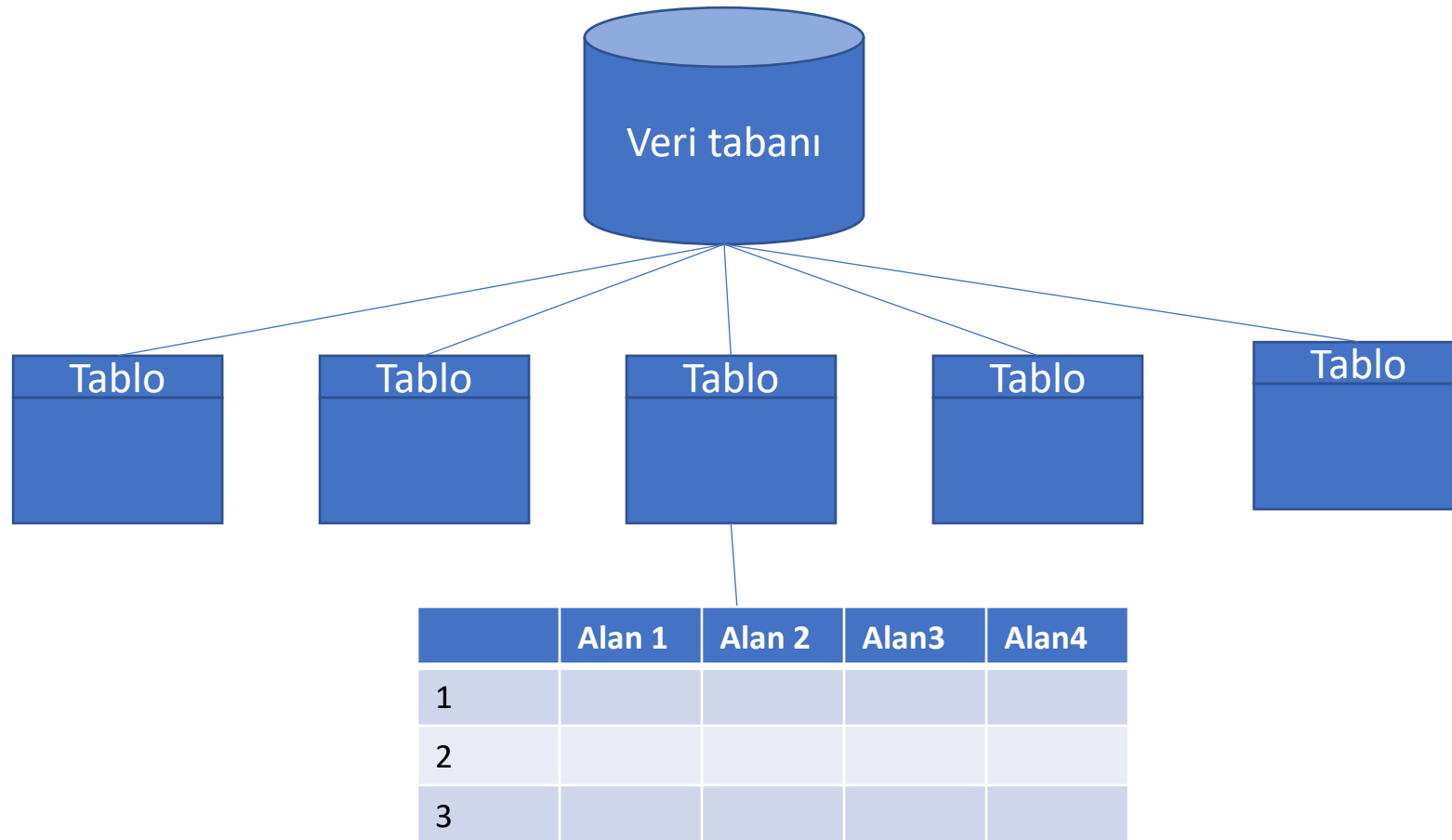
# *Veri Tabanı Yaklaşımının Avantajları*

- *Ortak verilerin tekrarının önlenmesi,*
- *Verilerin merkezi denetiminin ve tutarlılığının sağlanması*
- *Veri paylaşımının sağlanması*
- *Fiziksel yapı ve erişim yöntemi karmaşıklıklarının, çok katmanlı*
- *Her kullanıcıya yalnız ilgilendiği verilerin, alışık olduğu kolay, anlaşılır yapılarda sunulması mimarilerle kullanıcıdan gizlenmesi*

# *Veri Tabanı Yaklaşımının Avantajları*

- *Sunulan çözümleme, tasarım ve geliştirme araçları ile uygulama yazılımı geliştirmenin kolaylaşması.*
- *Veri bütünlüğünün gerekli olanakların sağlanması, mekanizmaların kurulması,*
- *Güvenlik ve gizliliğin istenilen düzeyde sağlanması*
- *Yedekleme, yeniden başlatma, onarma gibi işletim sorunlarına çözüm getirilmesi*

# *Veri Tabanı Yapısı*



## *Basit Bir örnek;*

- *İnternet üzerinden ticaret yapan bir firmamız olduğunu düşünelim.*
  - *Temelde basit olarak;*
    - *Ürünler,*
    - *Müşteri(Üyeler)*
    - *Sipariş*
- Şeklinde üç adet tablo oluşturalım;*

Ürün Listesi

Barkod	Ürün Adı	Fiyatı
1	Java	50
2	Delphi	120
3	Linux	60
4	XML	200
5	Perl	70
6	C++	80

Sipariş Listesi

Sipariş Kodu	Ürün Kodu	Kullanıcı Kodu
1	1	1
2	2	1

Kullanıcı Kod	
1	Ali
2	Ahmet
3	Ayşe
4	Fatma

- Tablo
- İlişki
- Birincil Anahtar
- Yabancı Anahtar

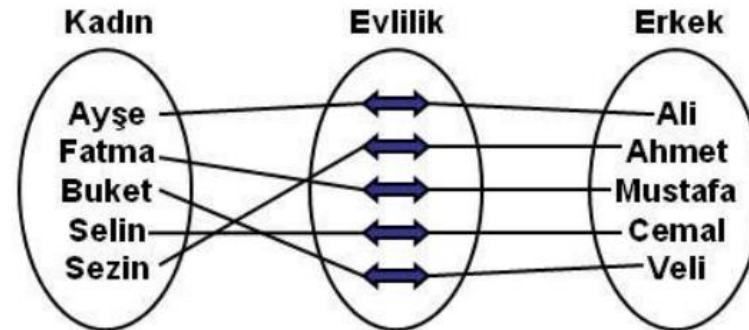
# *İlişkilendirme Tipleri*

- Bire bir ilişkiler ( One-to-one relationships):

## *Bire bir ilişkiler ( One-to-one relationships):*

- Aralarında ilişki olan iki tablo arasında, tablolardan birindeki asıl anahtar alanın kayıt değerinin, diğer tablodaki sadece bir kayıta karşılığının olması durumunu gösteren ilişki tipidir.*

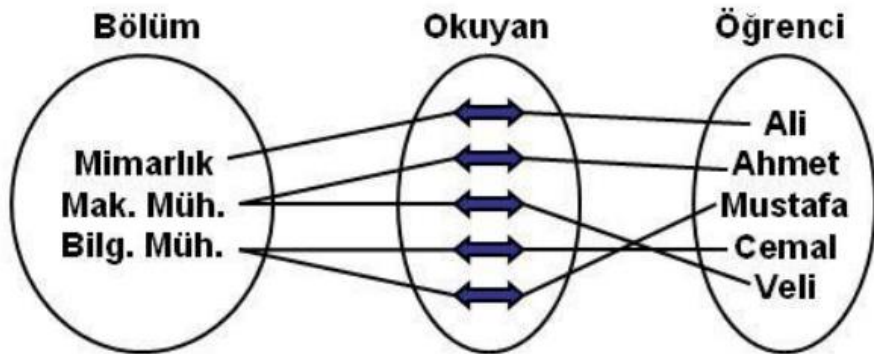
Örnek: “Evlilik” ilişkisi T.C. Medeni Kanunu’na göre birden-bire’dir.



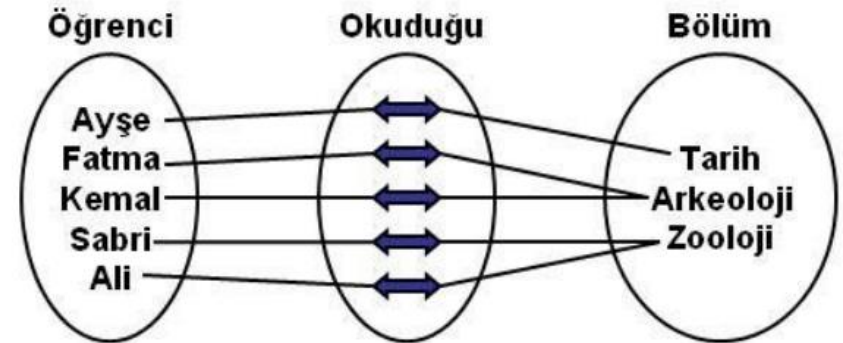
# Bire çok ilişkiler ( One-to-many relationships)

- Aralarında bir ilişki bulunan iki tablo arasında, anahtar alanının kayıt değerinin, diğer tablodaki birden fazla kayıta karşılığının olması durumunu gösteren ilişki tipidir.

Örnek: “Bölüm” ve “Öğrenci” varlık kümeleri arasındaki “Okuyan” ilişkisi, bölümden öğrenciye doğru birden-çoğa şeklindedir.



Örnek: “Öğrenci” ve “Bölüm” varlık kümeleri arasındaki “Okuduğu” ilişkisi, öğrenciden bölüme doğru çoktan-bire şeklindedir.

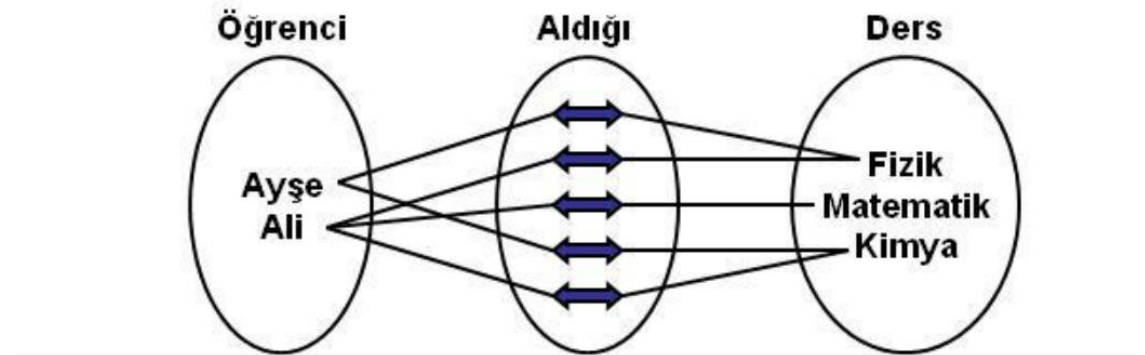




# Çoka çok ilişkiler( *Many-to-many relationships*):

- *Aralarında bir ilişki bulunan iki tablo arasında, tablolardan herhangi birindeki herhangi bir kaydın, diğer tablodaki birden fazla kayıtle ilişkilendirilebildiği ilişki tipidir.*

Örnek: “Öğrenci” ve “Ders” varlık kümeleri arasındaki “Aldığı” ilişkisi, çoktan-çoğa şeklinde bir ilişkidir.



# *Veri Tabanı Nesneleri*

- *Veri tabanı (Database)*
- *Tablo (Table)*
- *Sonuç Kümesi (ResultSet)*
- *Satır (Row)*
- *Sütun (Column)*
- *Alan (field)*
- *Veri Tipi (Data Type)*

# *Veritabanı (Database)*

- *SQL Server açısından bakıldığında içerisinde tabloları, aralarındaki ilişkileri, sorguları, kullanıcı hak ve yetkileri vb. bileşenleri barındıran mantıksal yapıdır.*

# *Tablo (Table)*

- *Bir veri tabanı tablolarda saklanan verilerden oluşur.*
- *Satır ve sütunlardan oluşan verileri tutan temel veri tabanı nesnesidir.*
- *Tablolar verilerin satırlar ve sütunlar halinde düzenlenmesiyle oluşan veri grubudur.*
- *Örneğin ders içeriği ve öğrenci bilgilerini veri tabanında saklamak için 2 tablo oluşturulur:*
  - *Ogrenci\_bilgileri*
  - *icerik*

# Tablo

Alan

Ogr_no	Ad_soyad	d_tarih	d_yeri	e-mail
1	Ayşe Öztürk	01.11.1979	Konya	ayse@gazi.edu.tr
2	Sema Özdemir	24.05.1975	Ankara	sema@gazi.edu.tr
3	Serdar Gülpınar	06.06.1983	Adana	serdar@gazi.edu.tr
4	Mehmet Efe	11.02.1978	Niğde	mehmet@gazi.edu.tr
5	Zerrin Polat	22.08.1980	Antalya	zerrin@gazi.edu.tr
6	Ulviye Ata	12.12.1984	İstanbul	ulviye@gazi.edu.tr

Kayıt

# *Sonuç Kümesi (Resulset)*

- *Bir sorgunun sonucunda anlık olarak üretilen ve tabloyu andıran yapıdır.*

# Satır(Row)

- *Bir tabloda yer alan her bir kayıt bir satıra karşılık gelir.*

	Alan 1	Alan 2	Alan3	Alan4
1				
2				
3				

# *Sütun(Column)*

- Tablolar dikey sütunların yan yana gelmesiyle meydana gelir.

	Alan 1	Alan 2	Alan3	Alan4
1				
2				
3				



# *Alan(field)*

- *Her bir sütunun adı ile birlikte diğer bilgilerinin(en fazla kaç birimlik veri tutulacağı, ne tür veri tutulacağının(tarih, para birimi, rakam, karakter, metin vb.) ) ortaya konulduğu tanıma alan denir.*

# *Veri Tipi(Data Type)*

- *Tutulan kayıtların özelliklerinin tanımlanmasıdır.*
  - *Her bir alanın tipinin;*
    - *Tam sayı*
    - *Sadece harf,*
    - *Ondalıkli sayı,*
    - *Tarih vb*

*Olarak önceden tanımlanması gerekir.*

# *Kayıt(Record)*

- *Yapılandırılmış verilerden her birine bir kayıt denir.*
- *Yani alan bilgileri ile birlikte her bir satır bir kayıttır.*

# Nesne ve Değişken İsimlendirme Kuralları

- Karakterler ile başlamalısınız. Harf ([a-z] ya da [A-Z]) ile ya da alt çizgi (\_) ile başlamalıdır
- 30 karakteri geçmemelisiniz.
- A-Z, a-z, 0-9, \_, \$, ve # kullanabilirsiniz.
- Bir isimlendirmeyi sadece bir yerde kullanabiliriz. ARMUT diye tablo varsa armut diye view oluşturamayız.
- Temel kullanılan ifadeler isim olarak verilemez. Mesela 'Select', 'Insert','Delete' diye bir tablo ismi yapamayız.
- İsimlendirme de **büyük harfle başlayıp küçük harfle biter**. Eğer küçük büyük harf duyarlı olmasını istersek isimlendirmeyi **tırnak(")** içerisinde yazmalıyız.
- Tanımlayıcı isimlerinde Türkçe harfler (Ü, ü, Ç, ç, Ö, ö, İ, ı, Ğ, ğ, Ş, ş) ve boşluk yer almamalıdır
- İsimleri kısa ve anlamlı olmalıdır

# *İsimlendirme Notasyonları*

- *Değişken ve nesne isimlendirmelerinde bazı kısıtlamalar ve uyulması gereken sistematiikler mevcuttur.*
- *Notasyonlar tanımlama işlemlerinde bazı kurallar ve düzen getirmeyi hedefler.*
- *Tanımlama işlemini büyük ya da küçük harf ile başlatmak, kelimeler arasına alt çizgi koymak gibi sıralanabilir.*

# *Deve Notasyonu*

- *İlk kelimenin tamamen küçük harf, geri kalan tüm kelimelerin baş harflerinin büyük şekilde yazılmasıdır.*
- *ArifGünel*

# *Alt Çizgi (Underscore) Notasyonu*

- *Kelimelerin arasına altçizgi koyulmasıdır.*
- *Arif\_Gunel*

# *Macar Notasyonu*

- *İsimlendirmenin başına, kullanılan değişken tipinin veya kontrolün kısaltma ile yazılmasıdır.*
- *tblArif, dbGunel*



# *Açıklama Satırları*

- *Tüm programlama dilleri ve veri tabanı yazılımlarında var olan ve geliştiricilerin kod blokları arasında açıklamalar, bilgi ve hatırlatıcı metinler girebilmesini sağlayan bir özelliktir.*
- *Açıklama haline getirilen satırlar otomatik olarak açık yeşil ile renklendirilir ve kodların veri tabanı motoruna gönderilmesi durumunda veri tabanı motoru buradaki metinleri kod olarak görmediği için çalıştırmaz.*

- *Bu açıklama satırı, tek satır ve çok satırlı olabilmektedir.*
- *Açıklama satırı için iki farklı yöntem vardır.*
  - *Tek satırda açıklama bitecek ise iki tane yan yana tire işareti (--)*
  - *Birden fazla satırda açıklama için /\* başlanıp \*/ ile bitirilmelidir.*

--fdfdffdfd

```
- select *From Musteri  
  /*dfdsfsdfs*/
```

# Kaynakça

- <http://www.sqlserveronculeri.com/1/Article/143/analysis-service-nedir--temel-kavramlari.aspx> Aykut Taşdelen C ve Sistem Programcıları Derneği  
Eğitmeni Tarih:05.09.2011