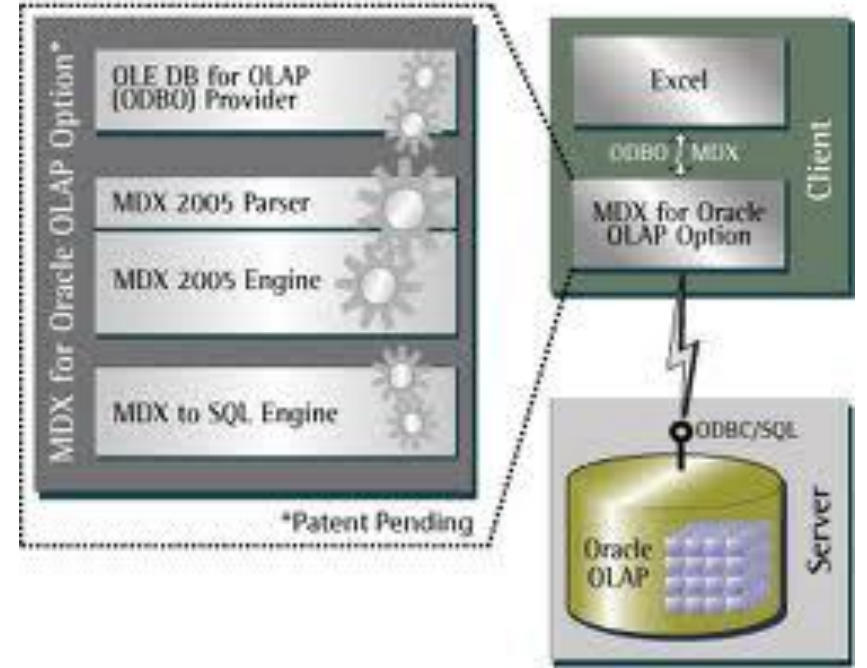


# 08118 Veri Tabanı I



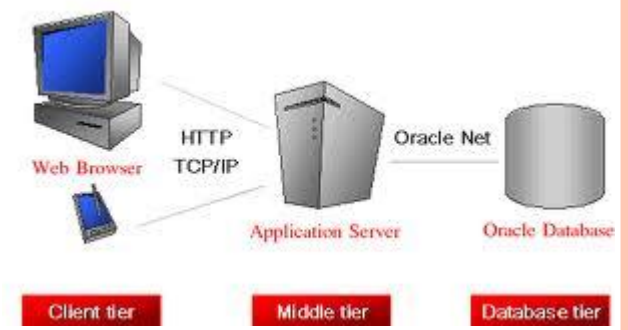
## Database Management System

Elbistan Meslek Yüksek Okulu  
2013 – 2014 Bahar Yarıyılı

# Veri Tabanı I :: DBMS ::

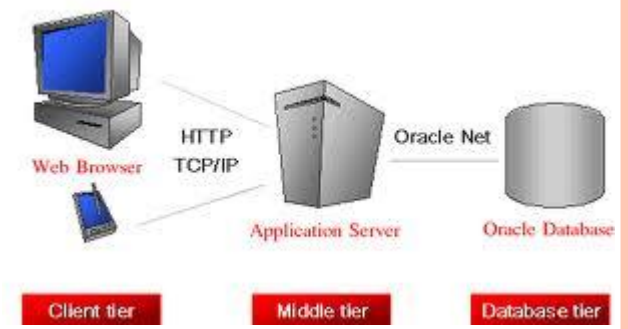
## Bu Haftanın Ders Konuları

- Veri Tabanı Temel Kavramları
- Varlık-İlişki Modeli
- Varlık ve Varlık dizisi
- İlişki ve İlişki Kümeleri
- Nitelikler
- Varlıkların Eşlendirilmesi
- Varoluş Koşulu
- Anahtarlar



# Veri Tabanı I ::: DBMS :::

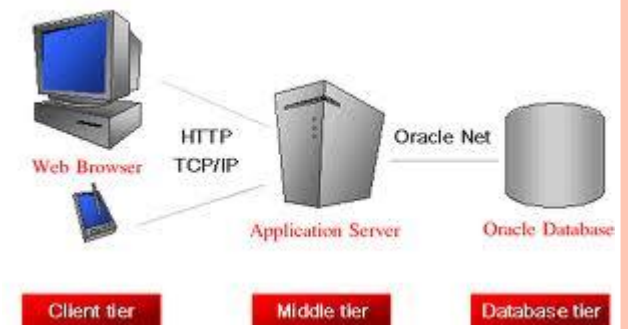
## Model Nedir ?



- Model kelimesi; isim, sıfat ve fiil olarak ve her birinde oldukça farklı çağrışımlar yapacak şekilde kullanılmaktadır.
- İsim olarak “model”, bir temsili ifade eder. Bu temsil; bir mimarın, bir binanın küçük ölçekli modeli veya bir fizikçinin bir atomun büyük ölçekli modelini oluşturması anlamındadır.

# Veri Tabanı I :: DBMS ::

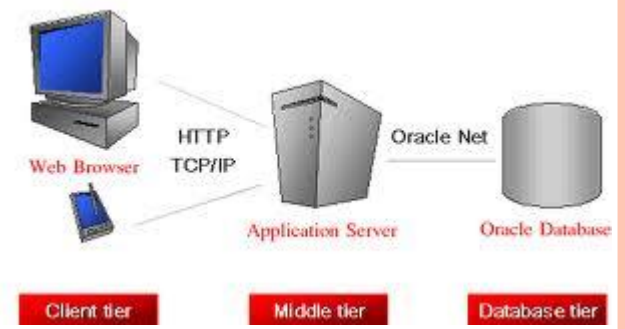
## Model Nedir ?



- Sıfat olarak “model”, mükemmeliyetin/idealin ölçüsünü ifade eder.
  - “Model ev”, “model öğrenci”-“model eş” ifadelerinde olduğu gibi.
- Fiil olarak “model” ise, bir şeyin nasıl olduğunu ispat etmek, açıklamak, göstermek anlamındadır.

# Veri Tabanı I :: DBMS ::

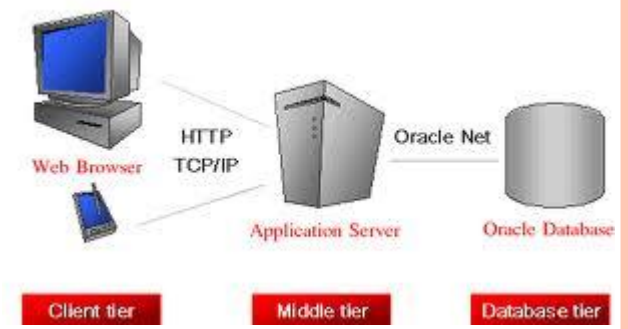
## Model Nedir ?



- Bilimsel modeller bütün bu çağrışımları bünyelerinde bulundurlar. Onlar; durumların, nesnelerin ve olayların temsilleridir.
- Örneğin, yeryüzünün bir kısmının modeli olan bir yol haritasında, bitki örtüsü gösterilmez. Çünkü bu durum, o haritanın bir yol haritası olarak kullanımı açısından uygun değildir.
- Güneş sisteminin bir modelinde, gezegenleri temsil eden topların, gezegenlerle aynı maddeden yapılmış olmaları veya aynı sıcaklığa sahip olmalarına ihtiyaç yoktur.

# Veri Tabanı I .:: DBMS ::.

## Model Tipleri

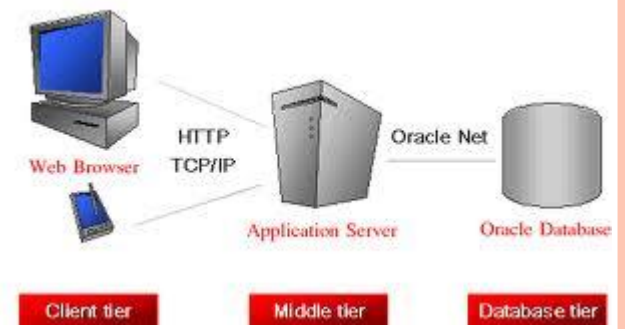


Bilimde, sıradan işlemlerde olduğu gibi, değişik tipte modeller kullanılır:

- Simgesel Model,
- Benzetim Modeli,
- Sembolik Model.

# Veri Tabanı I :: DBMS ::

## Simgesel Model



- Simgesel model, durumların büyük veya küçük ölçekli temsilleridir. Gerçek şeylerin uygun özelliklerini temsil ederler.
- Şekilleri, temsil ettikleri şeylere benzer. Yol haritaları, hava fotoğrafları bu tip modellere örnek verilebilir.

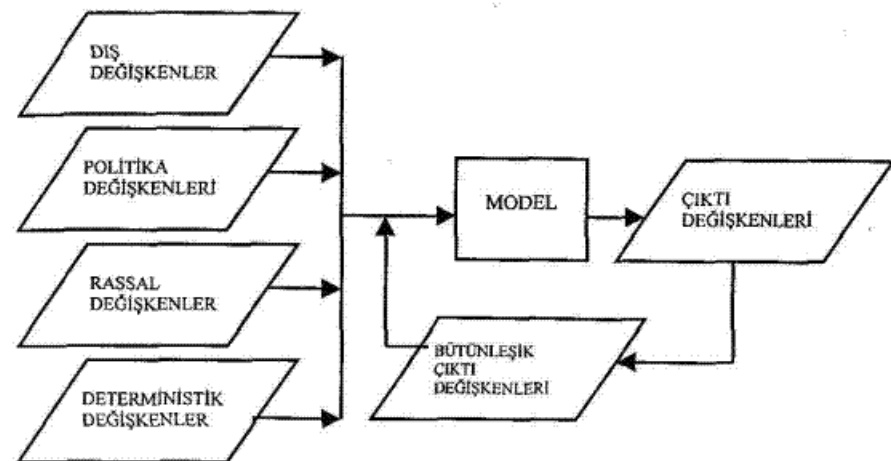




# Veri Tabanı I :: DBMS ::

## Benzetim Modeli

- Bazı durumlarda ise; haritada yükselteler, yol genişlikleri gibi özellikleri belirtmek gerekebilir.
- O zaman, renkler ve kontur çizgileri gibi bir takım açıklayıcı özelliklere ihtiyaç duyulur. Benzetim modeli, bir olayın simüle edilmiş halidir.
- Bu simülasyon benzetim modelinde çalışmalıdır.



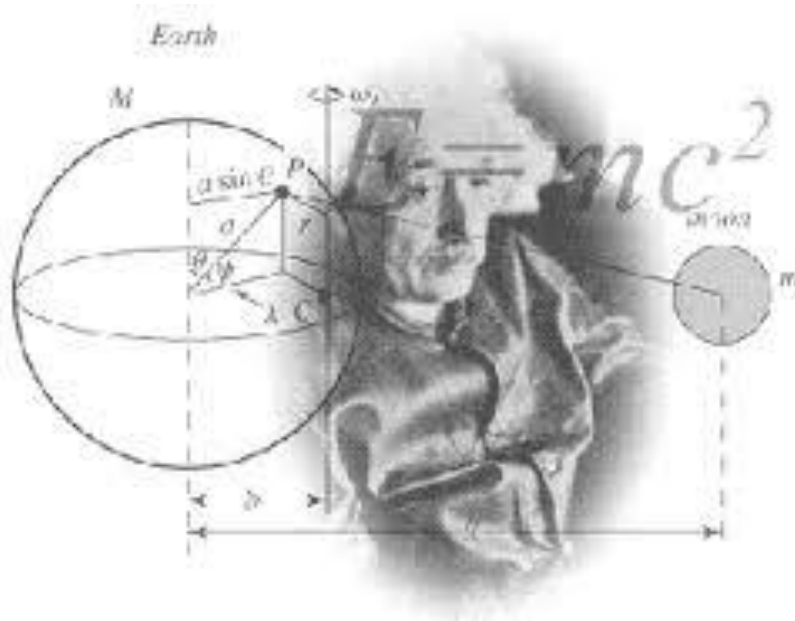
Şekil 2.5 Simülasyon modellerinde kullanılan değişkenler



# Veri Tabanı I :: DBMS ::

## Sembolik Model

- Temsil edilen şeylerin özellikleri sembollerle ifade edilir.
- Böylece, bir grafik ile gösterilen ilişki (benzetim modeli), bir eşitlik olarak da ifade edilebilir.
- Bu tip modellere Matematiksel Modeller de denilmektedir.

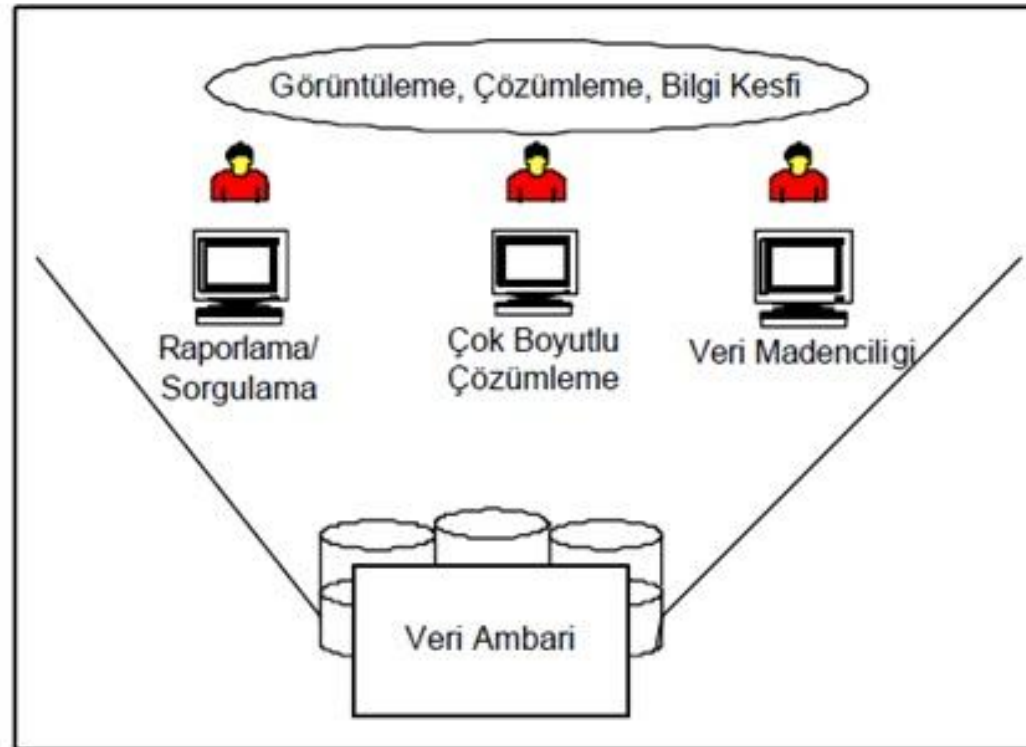


# Veri Tabanı I ::: DBMS :::

## Veri Nedir ?



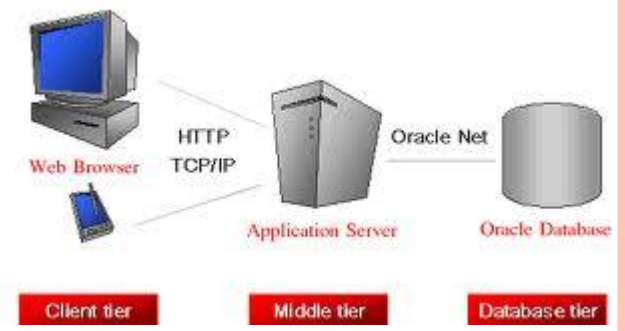
- Geçen haftaki dersimizde; veri kelimesi tanımlanmış ve yanlış verinin depolanmasını ve/veya verinin istenmeyen kişilerin kullanımına sunulmasını engelleyen bir takım imkânların olması gerektiğini belirtmiştik.



# Veri Tabanı I .:: DBMS ::.

## Yanlış verinin kaynakları

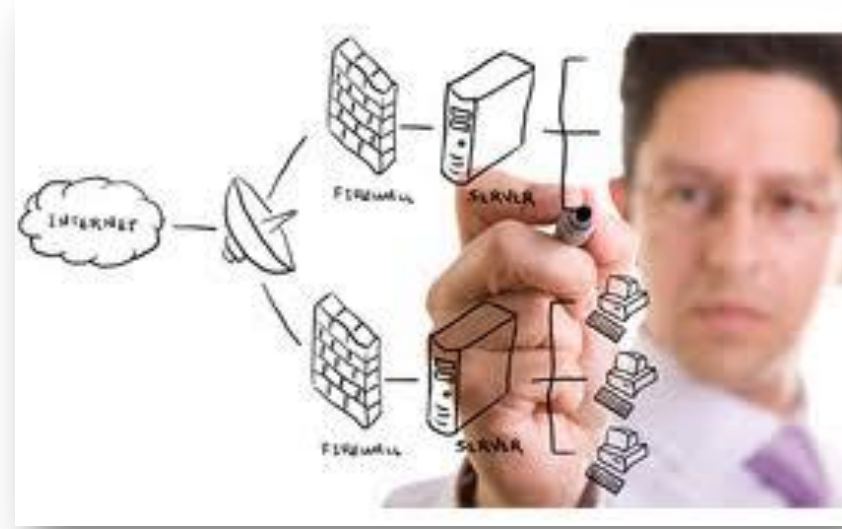
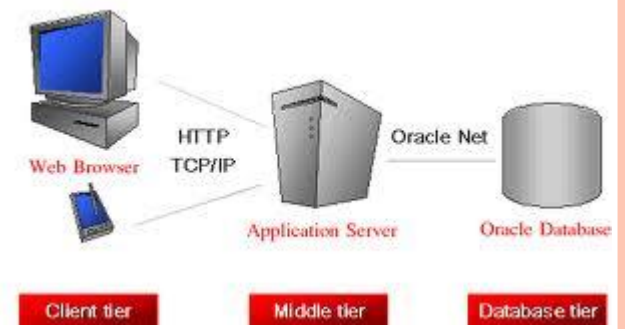
- Programlama hataları,
- Klavyeden hatalı giriş nedeniyle oluşan yanlışlıklar
- Veri tabanı programının kötü niyetli kullanımı.



# Veri Tabanı I .:: DBMS ::.

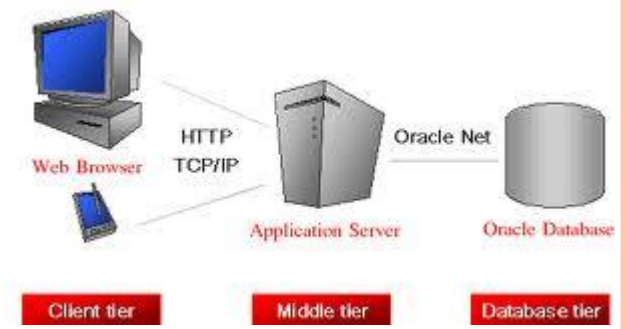
## Veri tabanlarının korunması

1. Veri güvenliği sağlanması
2. Veri bütünlüğünün sağlanması.



# Veri Tabanı I .:: DBMS ::.

## Veri güvenliği



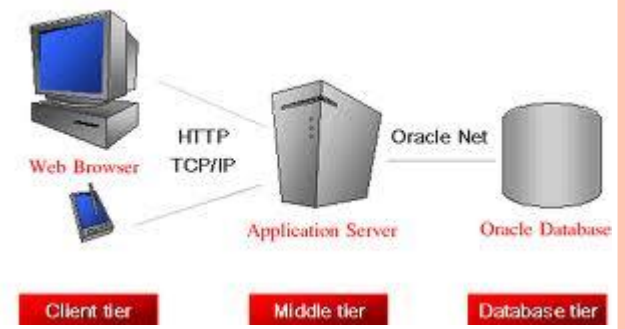
- Veri güvenliğinin konusu, veri tabanını, dolayısıyla veriyi yetkisiz kullanımlara karşı korumaktır.

Bu konuda çok çeşitli yaklaşımlar vardır;

- Hem verinin istenmeyen şekilde değiştirilmesine veya zarar görmesine hem de yetkisiz kullanımlara engel olmak gerekir.

# Veri Tabanı I ::: DBMS :::

## Veri güvenliđi



Veri Güvenliđini sađlamak için bazı genel teknikler geliřtirilmiřtir.

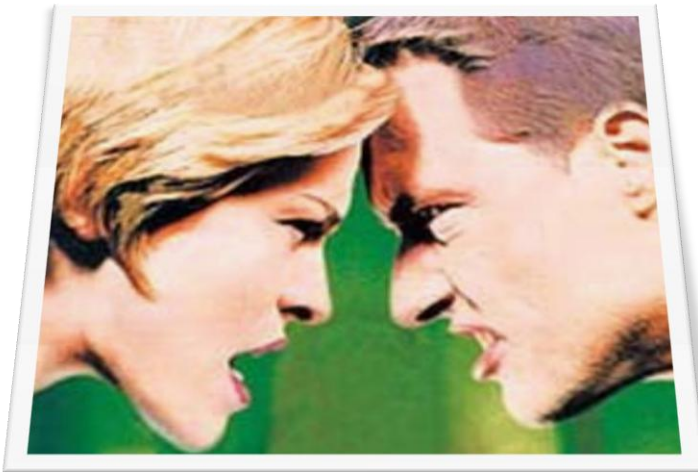
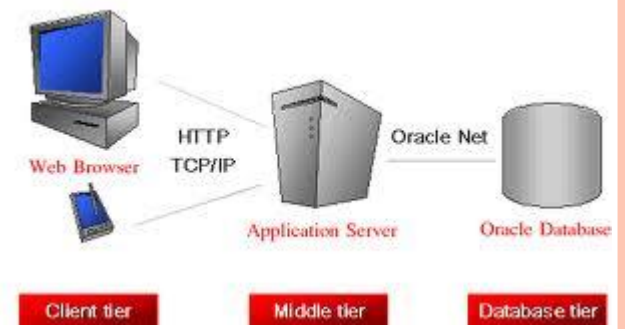
- Çoklu ortamlarda farklı yetkilere sahip kullanıcılar vardır.
- Farklı yetkilere sahip kiřilerin, DBMS ulařabilecekleri veri farklıdır.



# Veri Tabanı I :: DBMS ::

## Veri güvenliği

- Örneğin, bilgisayara veri girişi yapan bir işletimcinin, kurumun muhasebe kayıtlarına, muhasebe müdürü kadar yetkiliymiş gibi girerek değişiklikler yapması engellenmelidir.
- Bu amaçla, hangi kullanıcıların hangi yetkilerinin olduğu ve bu yetkilerini kullanabilmek için gerekli şifreler daha önceden tasarlanmalıdır.

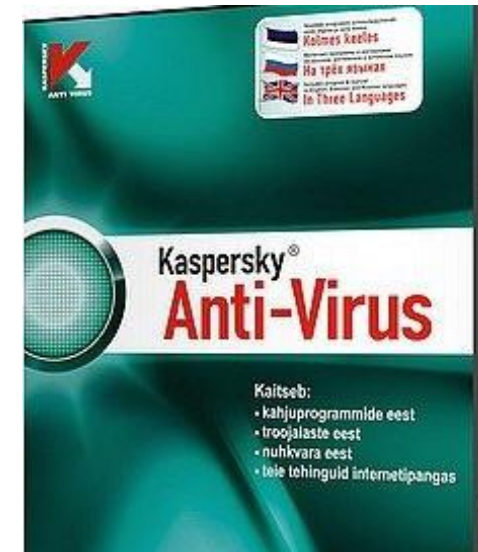
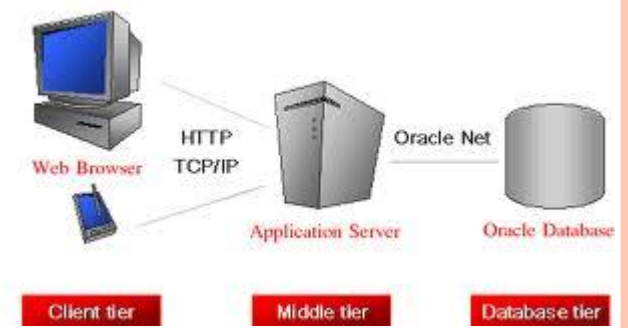




# Veri Tabanı I ::: DBMS :::

## Veri güvenliği

- Şifre sisteminin yeterli olmadığı durumlarda, verinin fiziksel koruma altına alınması gerekir.
- Yangın veya hırsızlığa karşı verinin yedeklenmesinin yapılması gibi.



# Veri Tabanı I ::: DBMS :::

## Kullanıcı haklarının temin edilmesi



- Sistemde hangi kullanıcının hangi yetkilere ve haklara sahip olduğu ve neler yapabileceğinin önceden belirlenmiş olması gerekir.
- DBMS tasarımı yaparken işin en başında yapılması gerekenlerden biri de budur.
- Bir kişinin yetkisini veya hakkını başka bir kişiye vermesi ise, sistemin müdahalesi dışında gerçekleşen bir durumdur.

# Veri Tabanı I .:: DBMS ::.

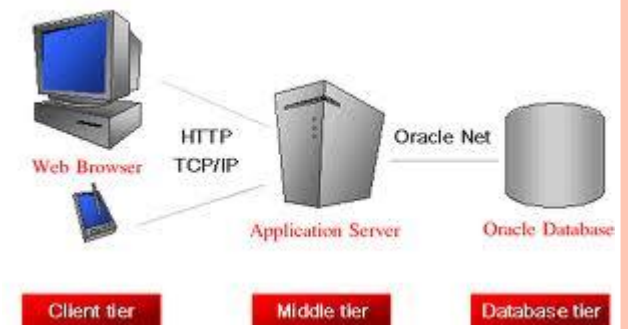
## Kullanıcı haklarının temin edilmesi



- 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 I .:: DBMS ::.

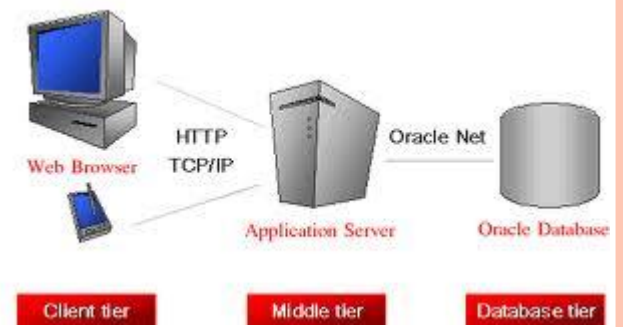
## Veri tekrarı ve veri bütünlüğü



- Bir veri tabanı yönetim sisteminde farklı veri dosyalarında; isim, adres, numara gibi bilgilerin bulunması gerekebilir.
- Örneğin, hem müşteri bilgilerini içeren bir veri tabanı dosyasında, hem de satılan malların sevkiyatının yapılacağı adreslerin bulunduğu başka bir veri dosyasında, müşteri adresi bilgilerinin yer alması gerekebilir.
- Yani, pek çok durumda, aynı verinin birden fazla veri dosyasında bulunması gerekebilir.
- Bu durum, veri tekrarı olarak ifade edilmektedir.

# Veri Tabanı I ::: DBMS :::

Bu durumu bir VT tablosunda açıklayalım.

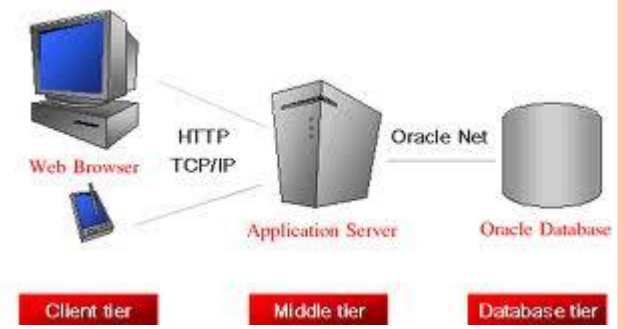


CustomerID	Name	Address	City
1	Julie Smith	25 Oak Street	Airport West
2	Alan Wong	1/47 Haines Avenue	Box Hill
3	Michelle Arthur	357 North Road	Roma

OrderID	Amount	Date	CustomerID	Name	Address	City
11	127.50	02-Apr-2000	02-Apr-2000	Julie Smith	25 Oak Street	Airport Street
20	112.99	15-Apr-2000	15-Apr-2000	Julie Smith	25 Oak Street	Airport Street
34	74.00	19-Apr-2000	19-Apr-2000	Julie Smith	25 Oak Street	Airport Street

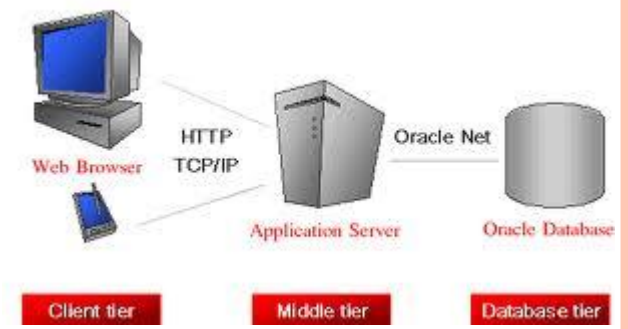
Böyle bir durum, veri bütünlüğünün bozulmasına neden olur.

## Veri Tabanı I ::: DBMS :::



- Veri üzerinde yapılacak deęişiklik, silme, ekleme gibi işlemlerin, o verinin bulunduğu bütün dosyalarda gerçekleştirilmesi gerekir.
- Özellikle çok kullanıcıli ortamlarda bu işlem oldukça önemlidir. Aksi takdirde, veri tabanında uygun olmayan veri ile çalışılmış olur.
- Veri bütünlüğünün bozulmasının bir sebebi, **veri tekrarı**dır.

## Veri Tabanı I ::: DBMS :::



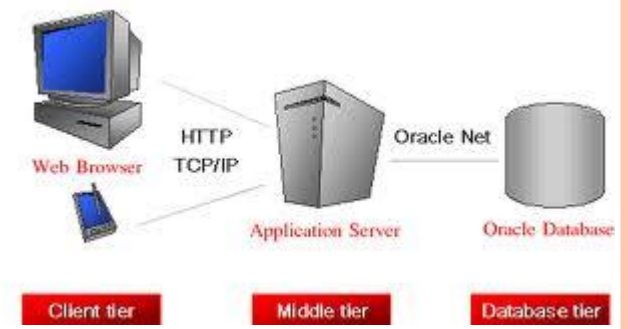
Bir başka sebep de,verinin zayıf geçerlilik kontrolüdür.

Bunun sebepleri de şu şekilde sıralanabilir:

- Veri güvenliğinin yetersiz oluşu,
- Veri tabanının zarar görmesi durumunda kurtarma yöntemlerinin yetersiz oluşu,
- Uzun kayıtların idaresinin zorluğu,
- Değişikliklerin esnek olmaması,
- Programlama ve bakım masraflarının yüksek olması,
- İnsandan kaynaklanan hatalar.



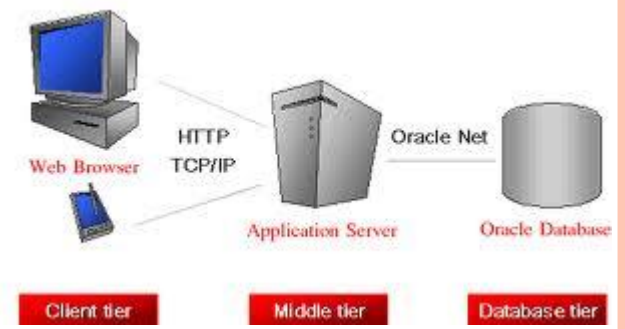
## Veri Tabanı I ::: DBMS :::



- Günümüzde kullanılan çeşitli veri tabanı yönetim sistemi programları, yukarıda sayılan bütün problemlerin üstesinden hemen hemen gelebilecek çözümler üretmişler ve bunları kolay kullanılabilir hale getirmişlerdir.
- Kullanıcıların, bir veri tabanı oluştururken, ayrıca bu problemler için önlem almalarına gerek kalmamaktadır.

# Veri Tabanı I :: DBMS ::

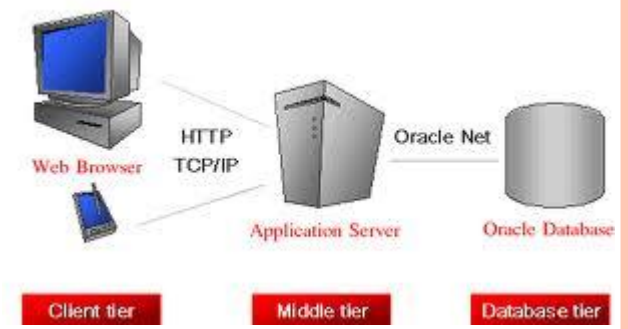
## Veri Modeli



- Bir veri modeli, verinin hangi kurallara göre yapılandırıldığını belirler.
- Fakat yapılar, verinin anlamı ve nasıl kullanılacakları hakkında tam bir açıklama vermezler.
- Veri üzerinde yapılmasına izin verilen işlemlerin belirlenmesi de gerekir.
- İşlemler, bu yapının sunduğu çerçeve içinde çalıştırılırlar.

# Veri Tabanı I ::: DBMS :::

## Yapılar



- Bir veri yapısı oluşturulurken, verinin bir şekilde bilgisayara yerleştirilmesi söz konusu olduğu için, nesnelere ve onların arasındaki ilişkilerin temsil edilmesi gerekir.
- Bu tür bir temsil tablolarla yapılabilir. Bir tabloda sütun başlıkları olarak öznitelikler ve satırlarda da bu özniteliklerin aldığı değerler (kayıt birimleri) yer alır.
- Tablodaki her bir sütun, bir veri birimidir.
- Her sütun aynı zamanda bir değişkendir.

# Veri Tabanı I ::: DBMS :::



TCKIMLIK	BABAADI	SOYAD	ADI	ADRES	PKODU	İL
02513794991	İBRAHİM	ÇİĞİLOĞLU	MURAT	GÜVERCİNLİK GAZ	00000	90
02530909792	ZEKERİYA	DİREK	DİLAY	YEŞİLKÖY YENİ	00000	90
10001724386	NECATİ	ALKAN	UĞUR MELİH	KİLECI MAH.1001 SOK. I	42550	42
10006357732	RAHMİ	GÜLER	YUSUF	BÜYÜKMANGIT BELDEŞ	1920	0
10006443794	CELAL	SARIKÇI	ERCAN	DUMLUPINAR MAH.106	1070	0
10009380652	ALİ	DEVECİ	ERSAN	CUMHURİYET MAH.	01700	1
10013865442	YUSUF	BÜLBÜL	NEZAKET	Ş.HALİS ŞİŞMAN MAH.	80750	80
10015403802	A CEVDET	YILMAZ	ARİFE	CUMHURİYET MAH. AY	01510	1
10019396728	HASAN	UĞUR	ŞERMİN	NUSRATİYE MAH.5005	33050	0
10021532510	EKREM	SARIKOÇ	MUSTAFA	VAKIF M. MOLLAÇAKIR	03700	03
10028417070	İSMAİL	KURT	RAMAZAN	MEVLANA M. AHMET UI	46100	46
10030319614	ÖMER	GENÇ	ŞANVER	ŞEHİTLİK MAH. KAF	6900	6
10030453028	ALİ	ÖZER	OYA	HAMZALI KÖYÜ	01680	1
10034289838	MUSTAFA	GENÇ	FİLİZ	KARAYILAN CAD. OÜ	27010	0
10039275782	HÜSEYİN	BEŞLİOĞLU	MEHMET	OSMANGAZİ MAH. K	06280	06
10045655882	ORHAN	AKABALI	İLKNUR	ZİRAAT MAH. SAĞ SOK.	6110	6
10048400298	ABDULLAH	EMİNE	KUNT	TUFAN PAŞA MAH. E	01520	01
10048459180	SÜLEYMAN	AKGÜL	PINAR	BURUK BEL. CUMHURİ	1440	0
10052194282	ÖMER	ARSLAN	GÜRKAN	OLGUNLAR MAH. AKINI	23100	23

# Veri Tabanı I :: DBMS ::



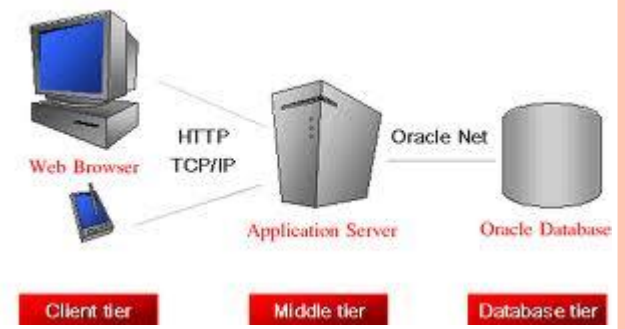
- Düz bir dosyadan oluşan veri tabanları olabileceği gibi (örneğin isim ve adres alanlarından oluşan adres veri tabanları), birden fazla dosyadan oluşan veri tabanları da vardır ve daha yaygın bir şekilde kullanılmaktadır.
- Bir veri tabanında temsil edilebilecek genel kayıt ilişkilendirme tipleri vardır.

TABLE

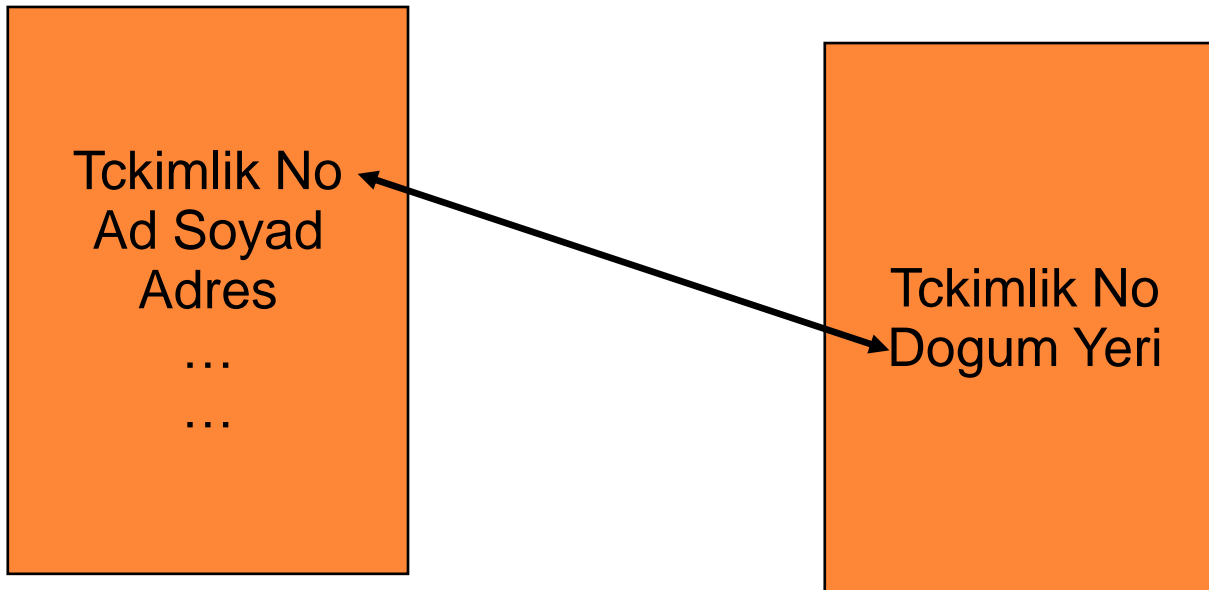
CustomerID	Name	Address	City
1	Julie Smith	25 Oak Street	Airport West
2	Alan Wong	1/47 Haines Avenue	Box Hill
3	Michelle Arthur	357 North Road	Roma

# Veri Tabanı I :: DBMS ::

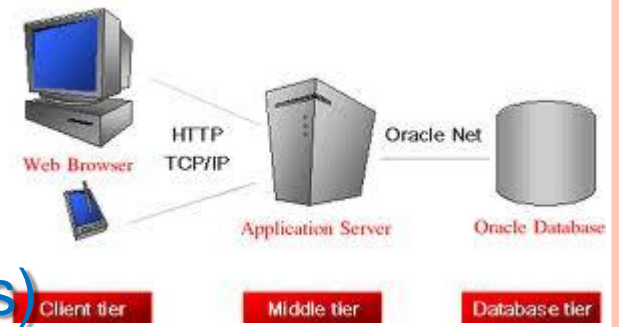
## Birden bire ilişkiler (one-to-one relationships)



- Aralarında bir ilişki olan iki tablo arasında, tablolardan birindeki asıl anahtar alanın kayıt değerinin, diğer tablodaki sadece bir kayıta karşılığının olması durumunu gösteren ilişki tipi.
- **Örnek** : bir işçinin doğum yeri bilgisinin doğum yerleri tablosunda bir şehre karşılık gelmesi gibi.

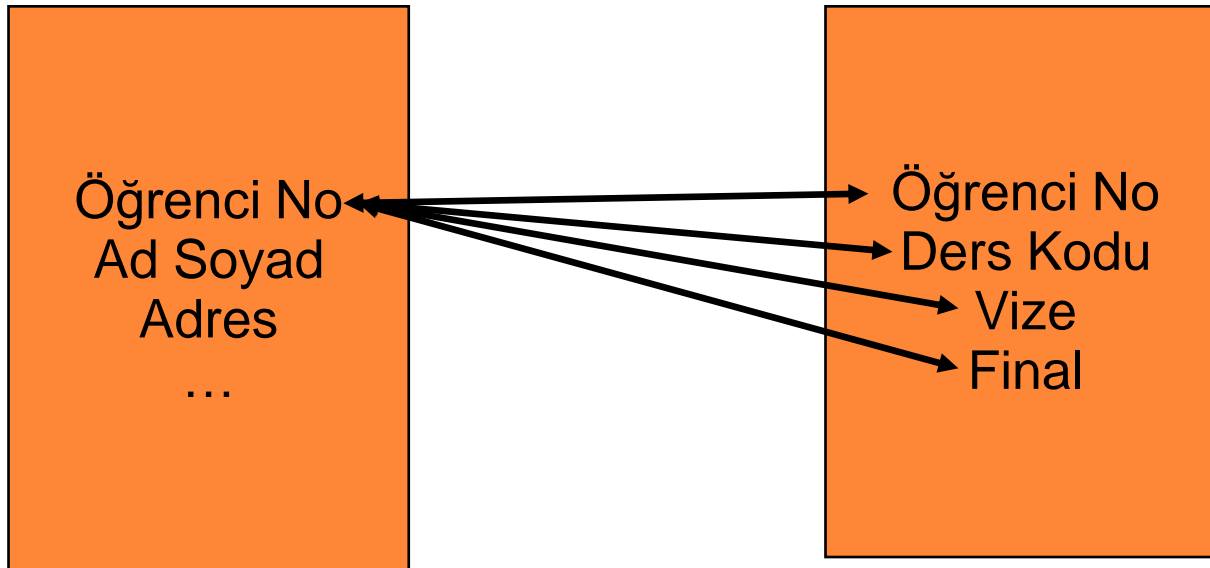


# Veri Tabanı I :: DBMS ::



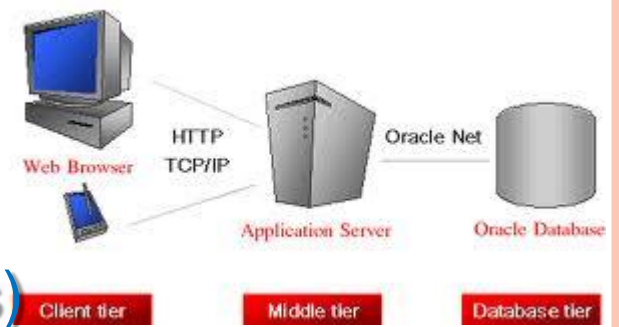
## Birden çoğa ilişkiler (one-to-many relationships)

- Aralarında bir ilişki olan iki tablo arasında, asıl anahtar alanın kayıt değerinin, diğer tablodaki birden fazla kayıta karşılığının olması durumunu gösteren ilişki tipi.
- **Örnek** : Bir öğrencinin birden fazla almış olduğu derse ve bu derse ait vize final sınav sonuçları gibi.
- Bir öğrenciye karşılık birden fazla ders ve notları





# Veri Tabanı I :: DBMS ::



## Çoktan bire ilişkiler (many-to-one relationships)

- Aralarında bir ilişki olan iki tablo arasında, tablolardan birindeki bir kaydın değerinin, asıl anahtar alanın olduğu diğer tabloda, birden fazla kayıta karşılığının olması durumunu gösteren ilişki tipi.

CustomerID	Name	Address	City
1	Julie Smith	25 Oak Street	Airport Street
2	Alan Wong	1/47 Haines Avenue	Box Hill
3	Michelle Arthur	357 North Road	Roma

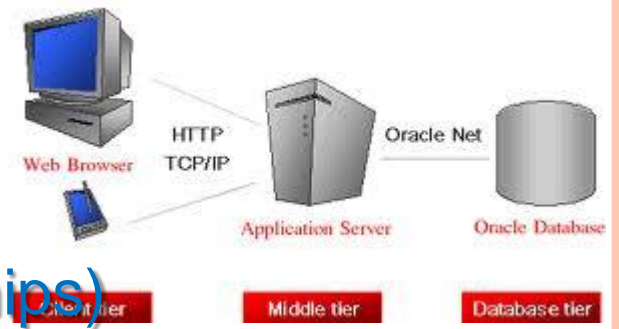
CUSTOMERS

ORDERS

OrderID	CustomerID	Amount	Date
1	3	27.50	02-Apr-2000
2	3	12.99	15-Apr-2000
3	2	74.00	19-Apr-2000

# Veri Tabanı I :: DBMS ::

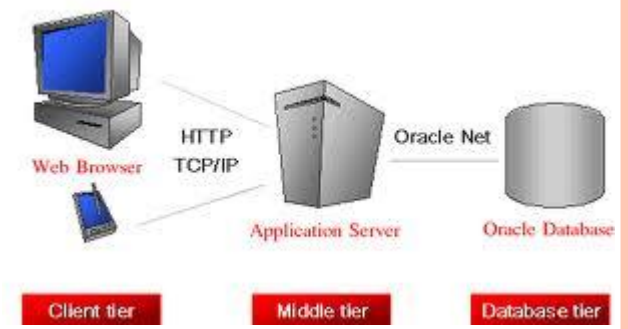
## Çoktan çoğa ilişkiler (many-to-many relationships)



- Aralarında bir ilişki olan iki tablo arasında, tablolardan herhangi birindeki herhangi bir kaydın, diğer tablodaki birden fazla kayıt ile ilişkilendirilebildiği ilişki tipi.
- Anne ve babanın çocukları ile olan ilişkisi tablosunda olduğu gibi.
- Bir sınıfa ait tüm öğrenciler ile dersleri arasındaki ilişki gibi.

# Veri Tabanı I :: DBMS ::

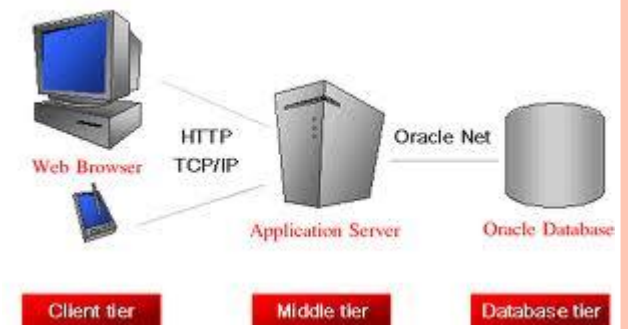
## Kısıtlar



- Veri üzerindeki mantıksal sınırlamalara kısıt adı verilir.
- Kısıtların genel olması tercih edilen bir durumdur.
- Örneğin, “Tüm yöneticilerin maaşları, işçilerinden daha fazladır” ifadesi, “Ahmet Bey’in maaşı Mehmet Bey’in maaşından daha fazladır” ifadesinden daha geneldir ve dolayısıyla daha kullanışlıdır.

# Veri Tabanı I :: DBMS ::

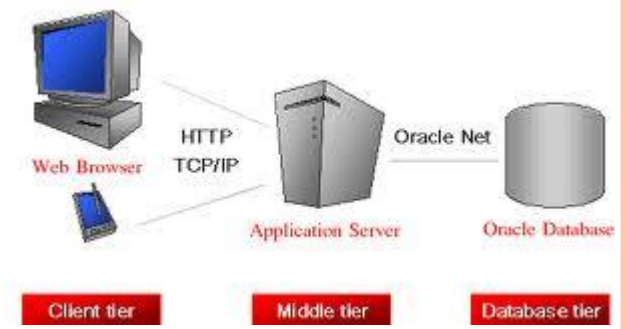
## Kısıtlar



- Kısıtlar, veri modellerinde bütünlük sağlamak ve semantik nedenlerle kullanılır.
- Örneğin, “İŞÇİ tablosunun YAŞ alanında 15 ve 65 arasında değer alabilir” şeklinde bir kısıtlama veri modelinde uygulanabilir.
- Bu sayede gerçek dünyada karşılaşılan bir özellik, oluşturulacak veri tabanına yansıtılabilir.

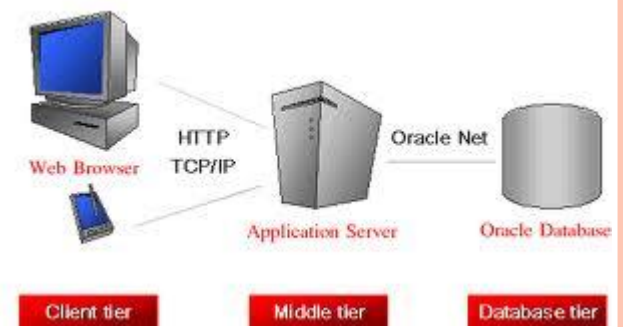
# Veri Tabanı I :: DBMS ::

## Kısıtlar



- Tckimlik No=11 Karakter olmalıdır.
- Tablolarda kısıtların kullanılması, fonksiyonel bağımlılıkların belirlenmesi amacını taşımaktadır.
- Örneğin, İŞÇİ varlık tipinde İŞÇİ\_NO özniteliği bir aday anahtar olabilir. Çünkü, diğer öznitelikler, bu öznitelige bağlı olabilir.
- Başka aday anahtarlar da olabilir (İSİM, ADRES gibi).

# Veri Tabanı I :: DBMS ::



- Bir tabloda, aday niteliklerinden biri asıl anahtar olarak belirlenir. İki ayrı ilişki tipi arasında bir bağlantı kurabilmek, o ilişkinin bir tablosundaki bir anahtarın (Primary key), diğer tabloya eklenmesi ile mümkün olur.
- Buna türetme, ikinci tabloya eklenen anahtara da yabancı anahtar (Foreign key) denir.

CUSTOMERS

CustomerID	Name
1	Julie Smith
2	Alan Wong
3	Michelle Arthur

AddressID	Address	City
1	25 Oak Street	Airport Street
2	1/47 Haines Avenue	Box Hill
3	357 North Road	Roma

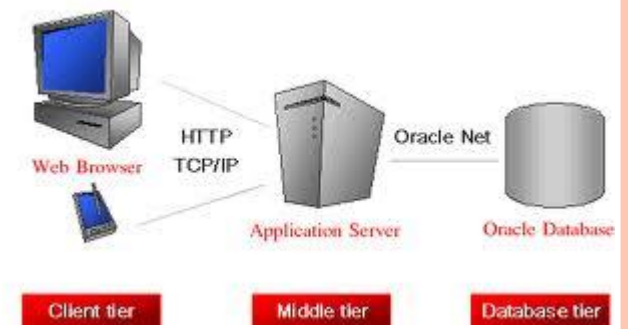
ADDRESS

ORDERS

OrderID	CustomerID	Amount	Date
1	3	27.50	02-Apr-2000
2	1	12.99	15-Apr-2000
3	2	74.00	19-Apr-2000

# Veri Tabanı I ::: DBMS :::

## İşlemler

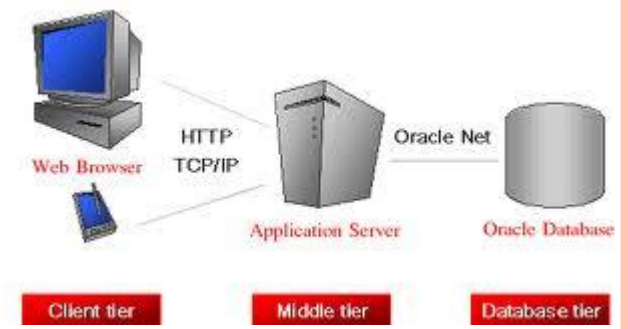


- İşlemler, bir veri tabanı durumundan, bir başka veri tabanı durumu elde etmek için yapılan operasyonlara (Operation) denir.
- Bunlar;
  - Verinin çağırılması,
  - Güncellenmesi,
  - Eklenmesi
  - Silinmesiile ilgili işlemlerdir.



# Veri Tabanı I ::: DBMS :::

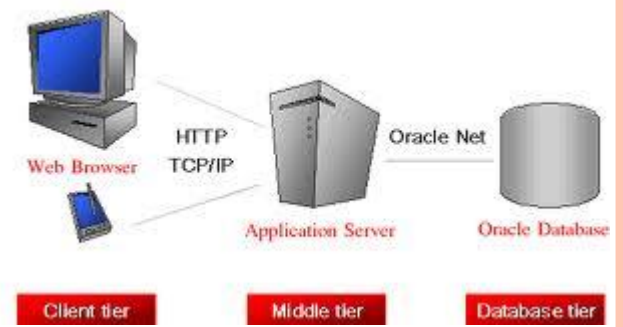
## Basit Veri Modelleri



- Basit veri modelleri olarak ayrılan ilk grup veri modelleri, bilgisayarlarda veri işleme ihtiyacının ortaya çıkması ve dosyalama sistemleri oluşturmak amacıyla kullanılmaya başlanan **Hiyerarşik** ve **Şebeke** veri modelleridir.
- **Hiyerarşik veri** modellerinde çoklu ilişkileri temsil edebilmek için, varlık tiplerinin her ilişki için ayrı ayrı tanımlanması gerekir.
- Bu da gereksiz veri tekrarına sebep olur.
- Hiyerarşik model, bir ağaç yapısına benzer. Model dahilindeki herhangi bir düğüm, altındaki n sayıda düğüme bağlanırken, kendisinin üstünde ancak bir düğüme bağlanabilir.

# Veri Tabanı I ::: DBMS :::

## Hiyerarşik veri modeli



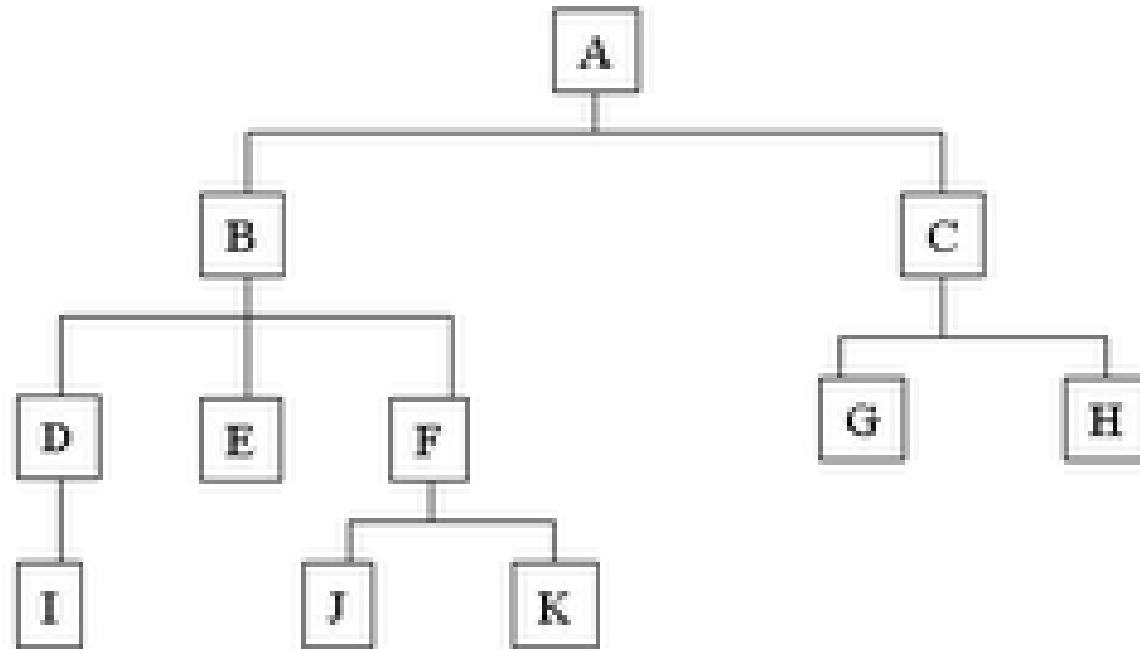
- Hiyerarşik yapının en tepesindeki düğüm noktasına kök (root) denir ve bu düğümün sadece bağımlı düğümleri bulunur.
- Bu veri yapısını gösteren grafiğe de hiyerarşik tanım ağacı denir.

Root Segment

Level 1 segments  
(root children)

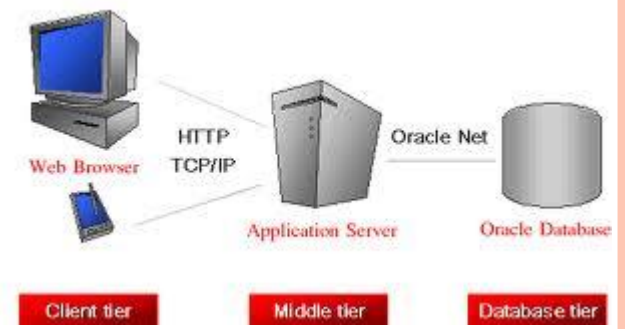
Level 2 segments  
(level 1 children)

Level 3 segments  
(level 2 children)



# Veri Tabanı I .:: DBMS ::.

## Şebeke veri modelleri

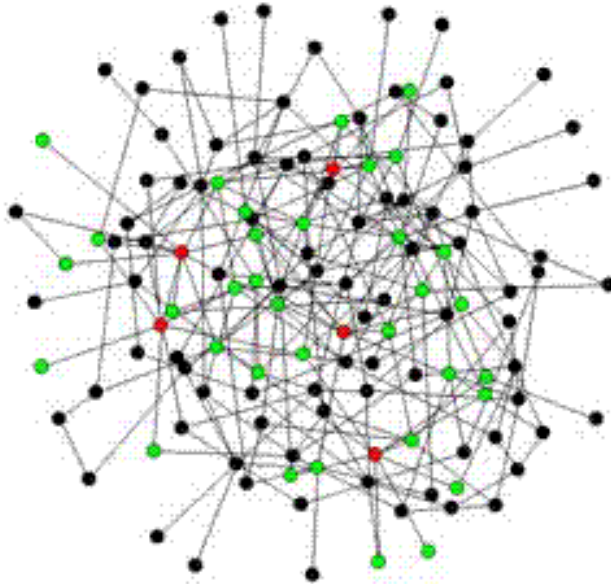
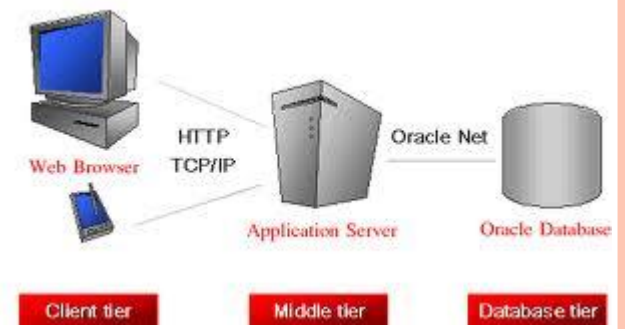


- Şebeke veri modelleri, **tablo ve grafik** temellidir.
- **Grafikteki düğümler** varlık tiplerine karşılık gelir ve tablolar şeklinde temsil edilir.
- **Grafiğin okları ilişkileri** temsil eder ve **tabloda bağlantılar olarak** temsil edilir.
- İki ayrı veri yapılandırma aracı vardır: **Kayıt tipi ve bağlantı**.
- Kayıt tipleri varlık tiplerini belirler. Bağlantılar ise, ilişki tiplerini belirler. Bu yapıyı gösteren grafiğe **de veri yapısı grafiği** adı verilir.

# Veri Tabanı I :: DBMS ::

## Şebeke veri modelleri

- Şebeke veri modeli, veri modelleri içinde en genel olanlarından biridir.
- Şebeke içinde bir eleman, herhangi bir başka elemana bağlanabilir. Hiyerarşik yapılardan farklı olarak, şebeke yapılarında bağlantı açısından herhangi bir sınırlama yoktur.



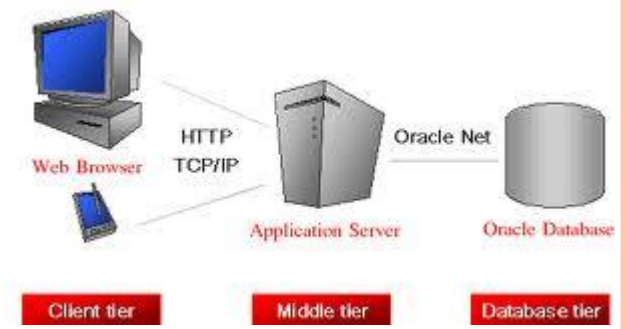
Şebeke veri modelleri

Karmaşık bağlantı,

Düğümdeñ düğüme bağlantılar

# Veri Tabanı I :: DBMS ::

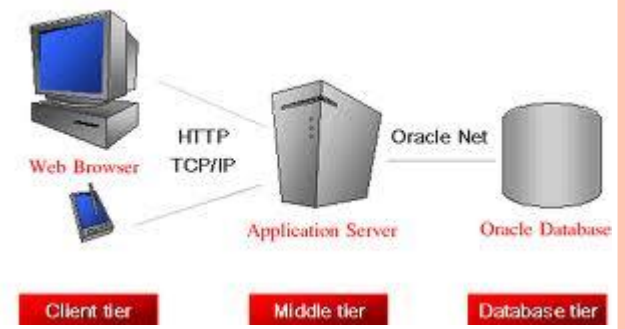
## Varlık ilişki (VI) modelleri



- Bir veri tabanı uygulaması için varlık, hakkında tanımlayıcı bilgi saklanabilen her şey olarak kabul edilmektedir.
- Varlık, bağımsızdır ve tek başına tanımlanabilir.
- Bir varlık, ev, öğrenci, araba gibi bir nesne ya da futbol maçı, tatil, satış gibi olaylar olabilir.
- En anlamlı şekilde kendi nitelikleri tarafından temsil edilir.
- Örneğin, bir EV; nitelikleri olan ADRES, STİL, RENK ve MALZEME ile tanımlanabilir.

# Veri Tabanı I :: DBMS ::

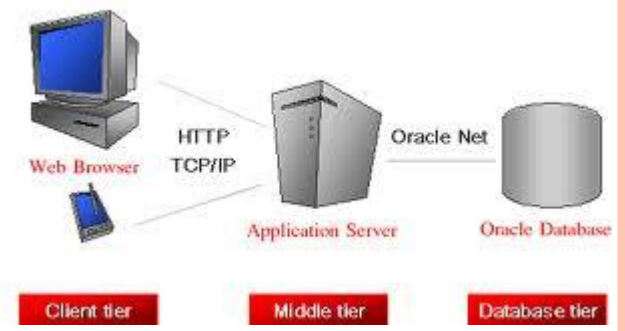
## Varlık ilişki (Vi) modelleri



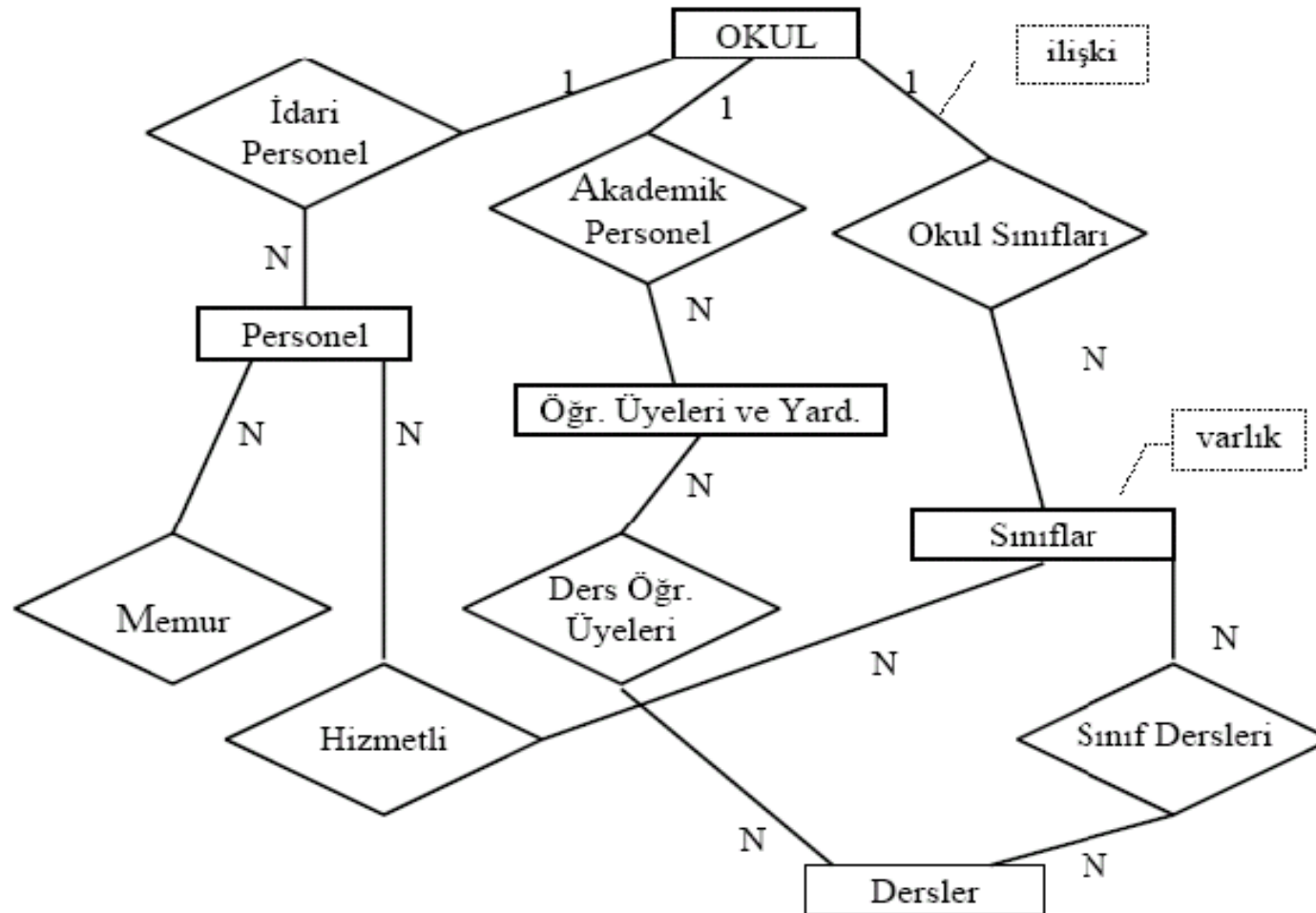
- Eğer bir özneliğin kendisi tanımlayıcı bilgi içeriyorsa, onu varlık olarak tanımlamak gerekir.
- Örneğin, eğer evin malzemesi hakkında ek bilgi depolamak gerekiyorsa MALZEME'yi de varlık (tablo) olarak sınıflamak gerekir.
- Varlık-İlişki veri modelleri (Vi), sütunlarında, öznelikleri temsil eden değişkenlerin yer aldığı ve satırlarında da enstantanelerin temsil edildiği tablolar, varlıklar ve aralarındaki ilişkileri oklarla göstermek için kullanılan grafikler üzerine kurulmuş veri modelleridir.
- Ticari veri tabanlarında yaygın olarak kullanılan veri modellerinden biridir.

# Veri Tabanı I ::: DBMS :::

## Varlık ilişki (Vİ) modelleri



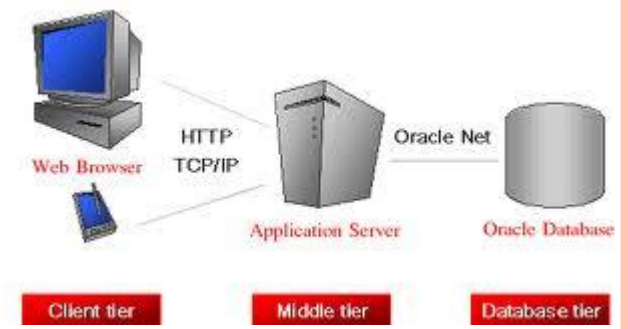
- Vİ modeline göre düzenlenen veri tabanının yapısı, **Varlık-İlişki Diyagramı** ile gösterilir.





# Veri Tabanı I :: DBMS ::

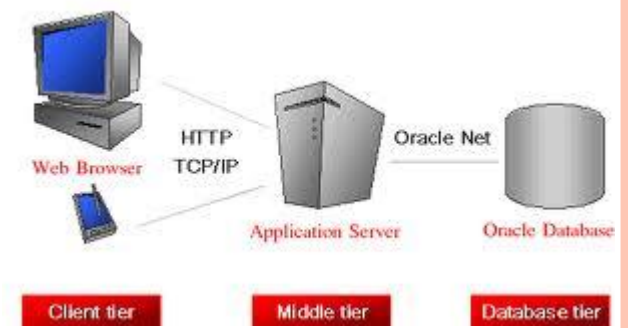
## Varlık ilişki (VI) modelleri



- İlişkisel şema listesini oluşturan her bir satır, bir tablo olarak temsil edilir.
- Tablonun sütunları nitelik olarak isimlendirilir.
- Örneğin, HASTANE tablosunun nitelikleri; Hastane\_Kodu, Hastane\_Adı, Adres, Tel\_No ve Yatak\_Sayısı'dır.
- Tablonun satırlarında bütün özniteliklerin aynı değerler aldığı iki satır olamaz.
- Her satır diğerinden mutlaka farklıdır. Aksi halde veri tekrarı söz konusu olur.
- Veri tabanlarındaki ilişki kavramı, matematikteki küme kavramını esas aldığı için aynı satırın bir tabloda birden fazla yer alması mümkün değildir.

# Veri Tabanı I :: DBMS ::

## Varlık ilişki (VI) modelleri



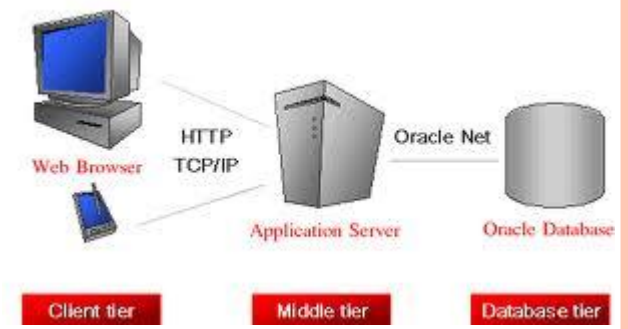
- Bu nedenle, ilişki için bir **anahtar** kullanmak gerekir. Anahtar, bir satırı tek başına tanımlayabilen nitelikler kümesidir.
- Anahtar kavramı, ilişkisel veri modelinde kullanılan önemli bir kısıttır.

**COLUMN**

CustomerID	Name	Address	City
1	Julie Smith	25 Oak Street	Airport West
2	Alan Wong	1/47 Haines Avenue	Box Hill
3	Michelle Arthur	357 North Road	Roma

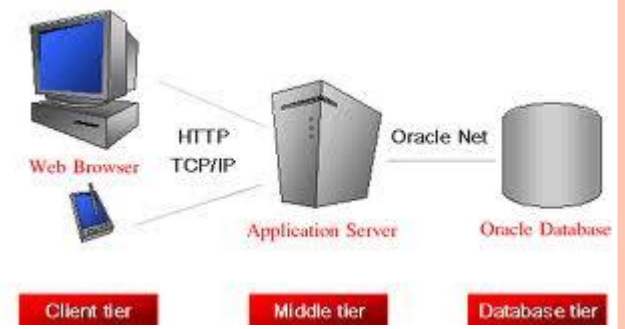
# Veri Tabanı I :: DBMS ::

## İlişkisel Veri Modeli Dilleri



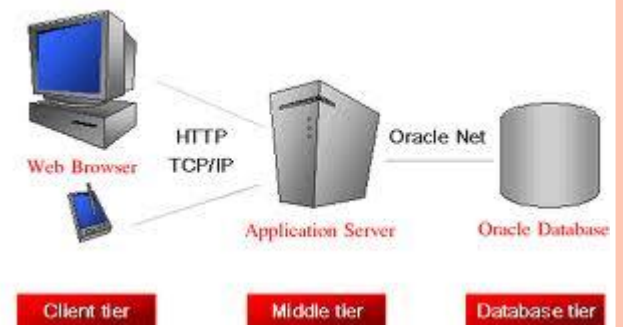
- Veri üzerinde yapılacak işlemler için, ilişkisel veri modellerinde üç tip dil kullanılır.
- Birincisi, **matematikteki ilişkisel işlemlere** dayanır. Bu tip dillere örnek olarak **INGRES** ve **QUEL** verilebilir.
- İkinci tip dil, görüntü yönelimlidir. **Boşluk doldurma** yöntemiyle çalışır. Örneğin, **QBE (Query By Example)** ve **CUPID** bu tür dillerdendir.
- Üçüncü tip dil, **haritalandırma yönelimli** dildir. Bu tip diller, bilinen bir özniteliğin ya da öznitelik kümesinin, aranan bir özniteliğin ya da öznitelik kümesinin üzerinde, bir ilişki yoluyla haritalandırılması prensibiyle çalışır. Örneğin, **yapısal sorgulama dili (SQL)** bu tip bir veri dilidir

## Veri Tabanı I .:: DBMS ::.

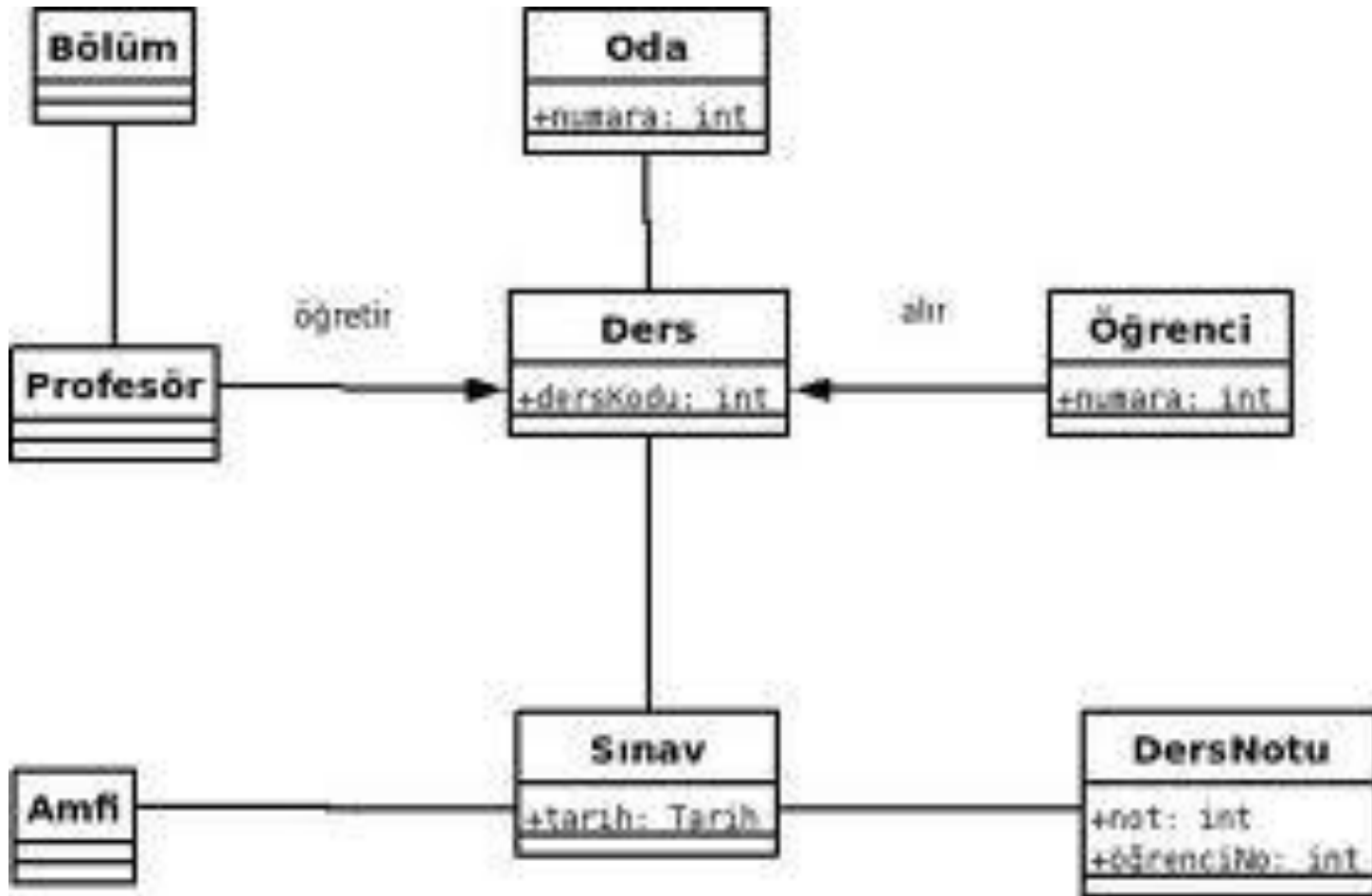


- Nesne yönelimli veri modelinde, bir sorgunun karşılığında mutlaka önceden tanımlanmış belirli bir nesne kümesi olması gerekir. Bir sorgunun sonucu olarak tesadüfi bir nesne kümesinin elde edilmesi mümkün değildir.
- Çünkü bütün nesnelerin, modelde önceden tanımlanmış olması gerekmektedir. İlişkisel modeldeki ilişki kavramı, nesne yönelimli modelde sınıf kavramına karşılık gelmektedir.

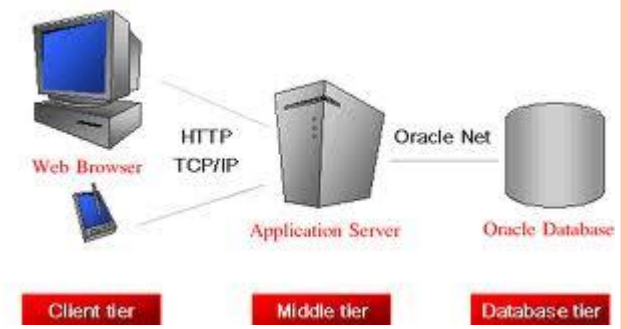
# Veri Tabanı I ::: DBMS :::



- Nesne yönelimli veri modeli;

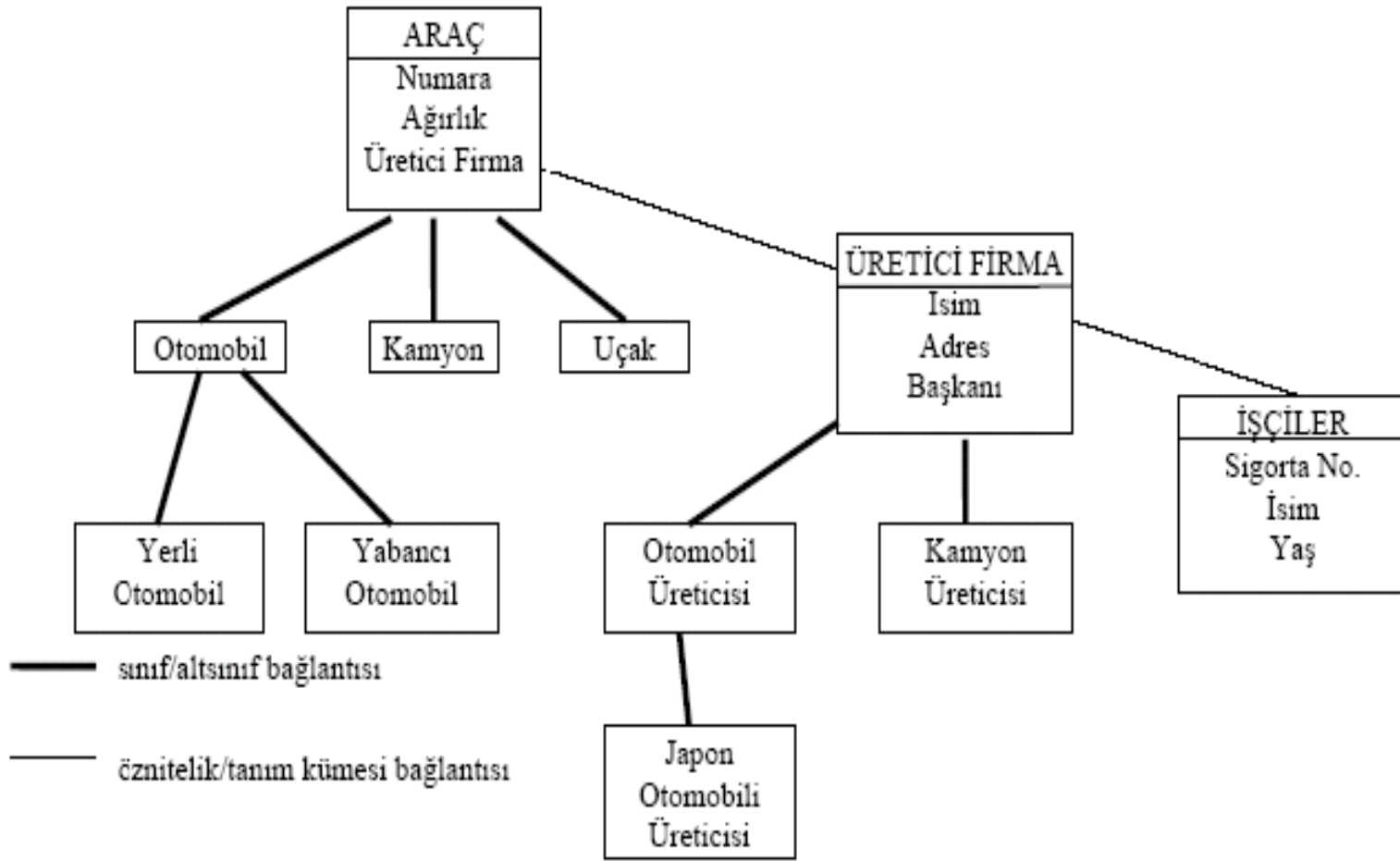


## Veri Tabanı I ::: DBMS :::



- Nesne yönelimli modellemenin en önemli faydalarından bir tanesi de, modeldeki nesnelere tanımlarken, ortak öznelik ve metotlara sahip nesnelere kullanıldıkları her farklı ortamda, tekrar tanımlanmalarına gerek duyulmamasıdır.
- Aksi takdirde bu durum, hem tekrardan dolayı yer kaybına, hem de modeldeki dinamik değişikliklerin pratik olmamasına sebep olacaktır.

# Veri Tabanı I :: DBMS ::



# Veri Tabanı I :: DBMS ::



- Kurum nesnesinin bir elemanı,

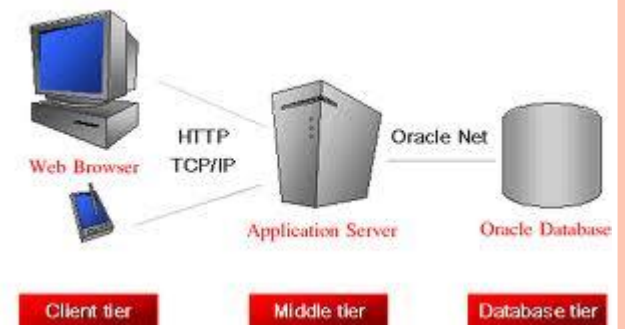
İsim : İstanbul Otelcilik A.Ş.  
İdari Yeri : İstanbul  
Telefon : 0212-555 5555

yönetici nesnesinin bir elemanı da, aşağıdaki gibi tanımlanabilir.

İsim : Ahmet Gel  
Adres : Fenerbahçe, İstanbul  
Doğum\_Tar : 1943



# Veri Tabanı I ::: DBMS :::

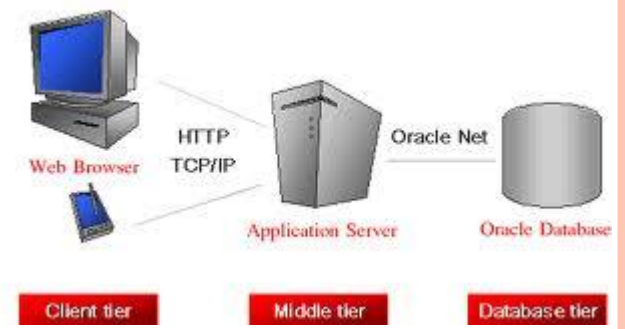


- **Tanımlama:** Bu işlem hem soyut kavramların (sınıf), hem de somut kavramların (elemanlar), teker teker tanımlanması ile ilgilidir ve anahtar değerler yardımıyla yapılır.

Bilgisayar ve çalışanlar sınıfları  
Sınıfların üyesi elemanlar

Bilgisayar Ekibi (Analizci, Programcı, İşletimci)  
Çalışanlar (Bakım Ekibi, Bilgisayar Ekibi, Yönetici)

# Veri Tabanı I ::: DBMS :::



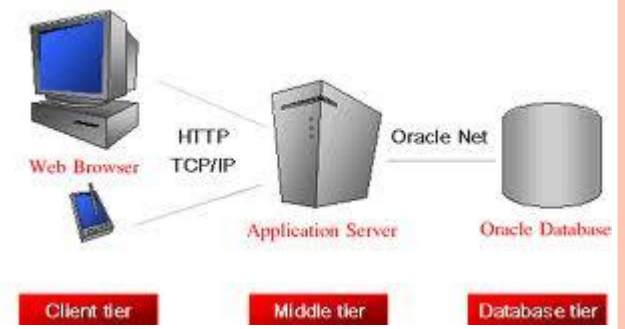
- **Toplam:** Nesnelar arasındaki ilişkilerin daha üst düzeyde, bir toplam nesne (veya tip) tarafından temsil edilmesi ile ilgili bir soyutlama yöntemidir.
- Bu toplam tipe genellikle anlamlı bir isim verilir ve bu isim modelin başka yerlerinde, ona ait özellikler referans olarak verilmeden kullanılabilir.

Bilgisayar ve çalışanlar toplamı  
Toplamın üyesi elemanlar



Bilgisayar Ekibi (Analizci, Programcı, İşletimci)  
Çalışanlar (Bakım Ekibi, Bilgisayar Ekibi, Yönetici)

# Veri Tabanı I :: DBMS ::



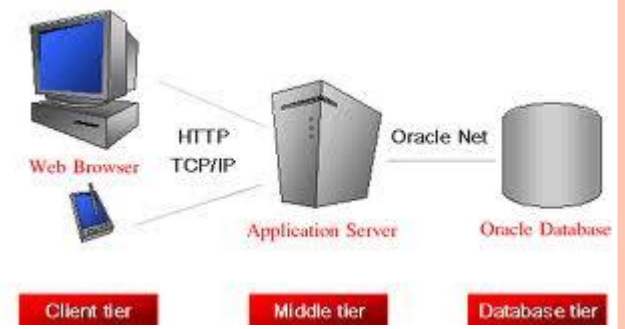
- **Genelleştirme:** Aynı özelliklere sahip bir grup nesnenin, sosyal nesne olarak temsil edilmesi ile ilgili bir soyutlama yöntemidir. Örneğin, bir kurumda çalışan personel şu şekilde düzenlenebilir:
- “Bilgisayar Ekibi” nesnesi; Analizci, Programcı ve İşletimci elemanları (herbiri bir alt nesnedir) için bir sosyal nesnedir.
- Aynı şekilde Çalışanlar nesnesi de Bakım Ekibi, Bilgisayar Ekibi ve Yönetici nesnelere için sosyal nesne durumundadır.

Genel Nesne

Eleman (alt nesne)

Bilgisayar Ekibi (Analizci, Programcı, İşletimci)  
Çalışanlar (Bakım Ekibi, Bilgisayar Ekibi, Yönetici)

# Veri Tabanı I :: DBMS ::



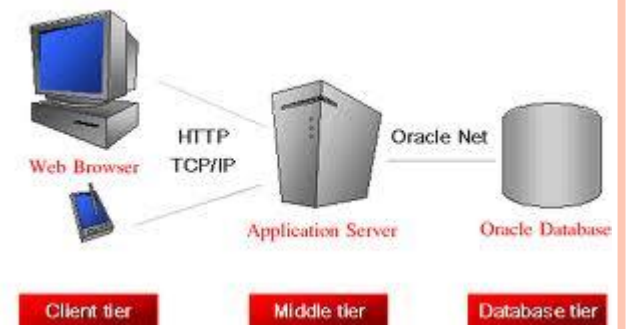
NYVM, ilişkisel veri modellerine karşı üstünlükleri vardır.

- Bunlar; NYVM'nde veri tiplerinin İLVM'e göre daha esnek olması,
- nesne tanımlarında soyutlama yapılabilmesine imkân vermesi ve bu tanımların semantik içeriklerinin de olması
- Veri bütünlüğünün daha kolay sağlanabilmesi ve ilişkisel veri modellerine göre, mevcut veri yapısında daha fazla genişleme ve yeniden düzenleme imkânlarına sahip olması sayılabilir.

## NESNEYE YÖNELİK VERİ MODELİ

Bilgisayar Ekibi (Analizci, Programcı, İşletimci)  
Çalışanlar (Bakım Ekibi, Bilgisayar Ekibi, Yönetici)

# Veri Tabanı I ::: DBMS :::



... Devamı Haftaya