

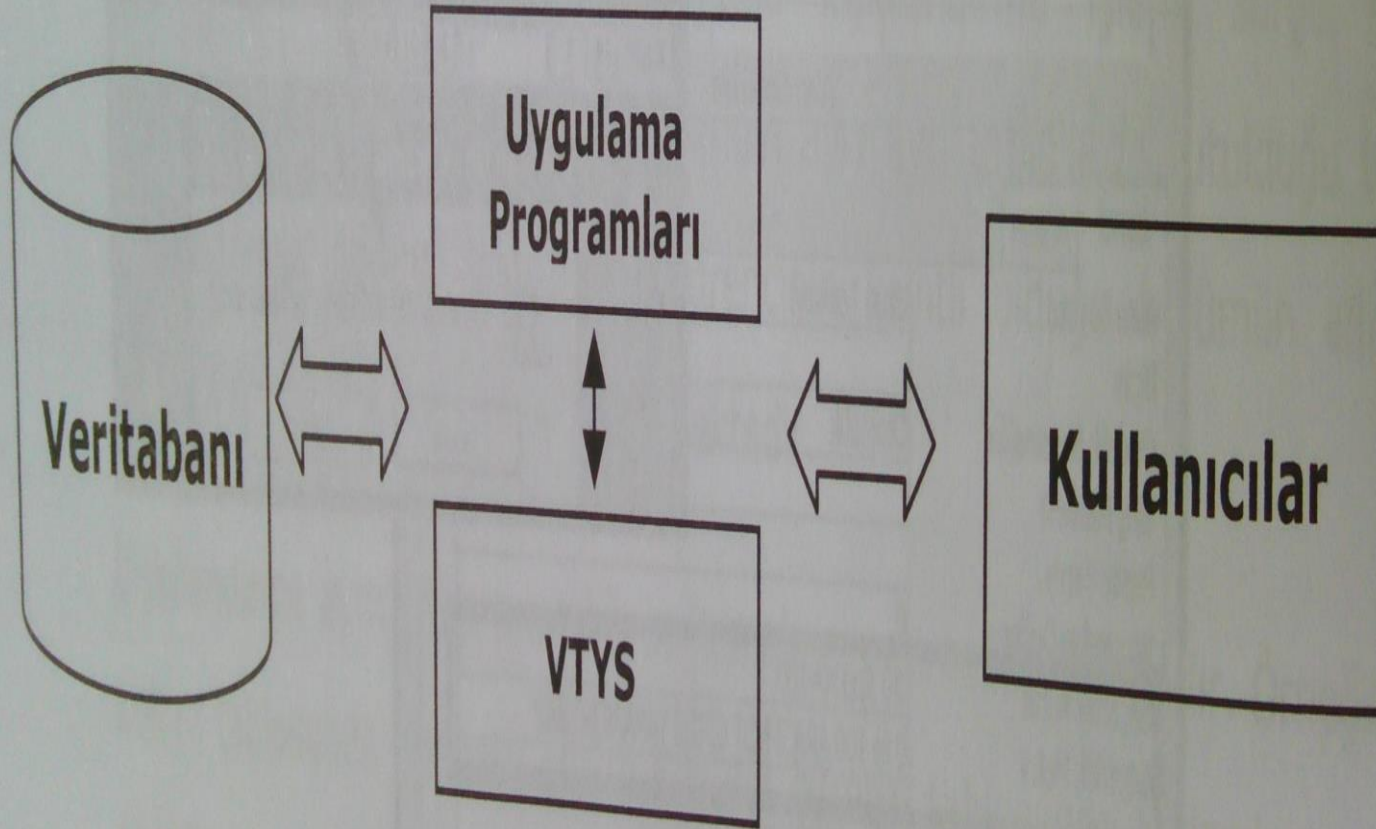
Veritabanı Yönetim Sistemleri



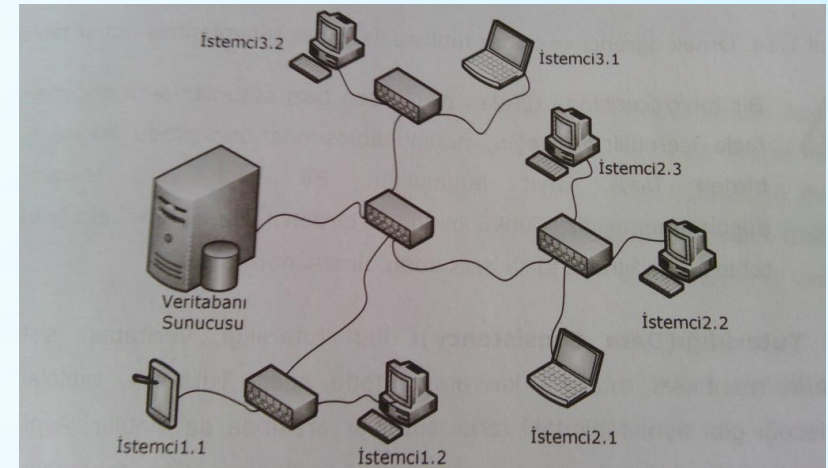
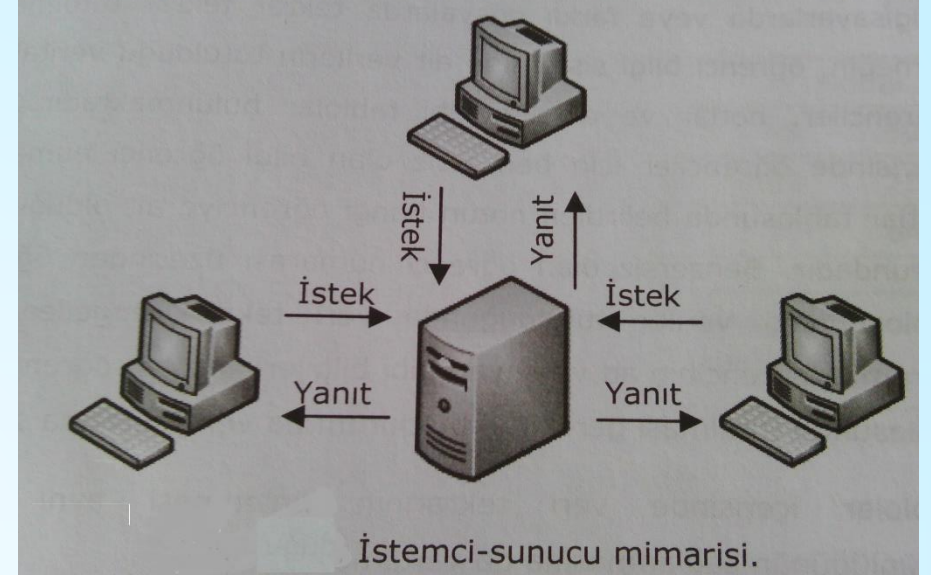
Dr. Öğr. Üyesi Durmuş ÖZDEMİR

e-mail: durmus.ozdemir@dpu.edu.tr

VTYS



Şekil 1.11. Veritabanı Yönetim Sistemi.

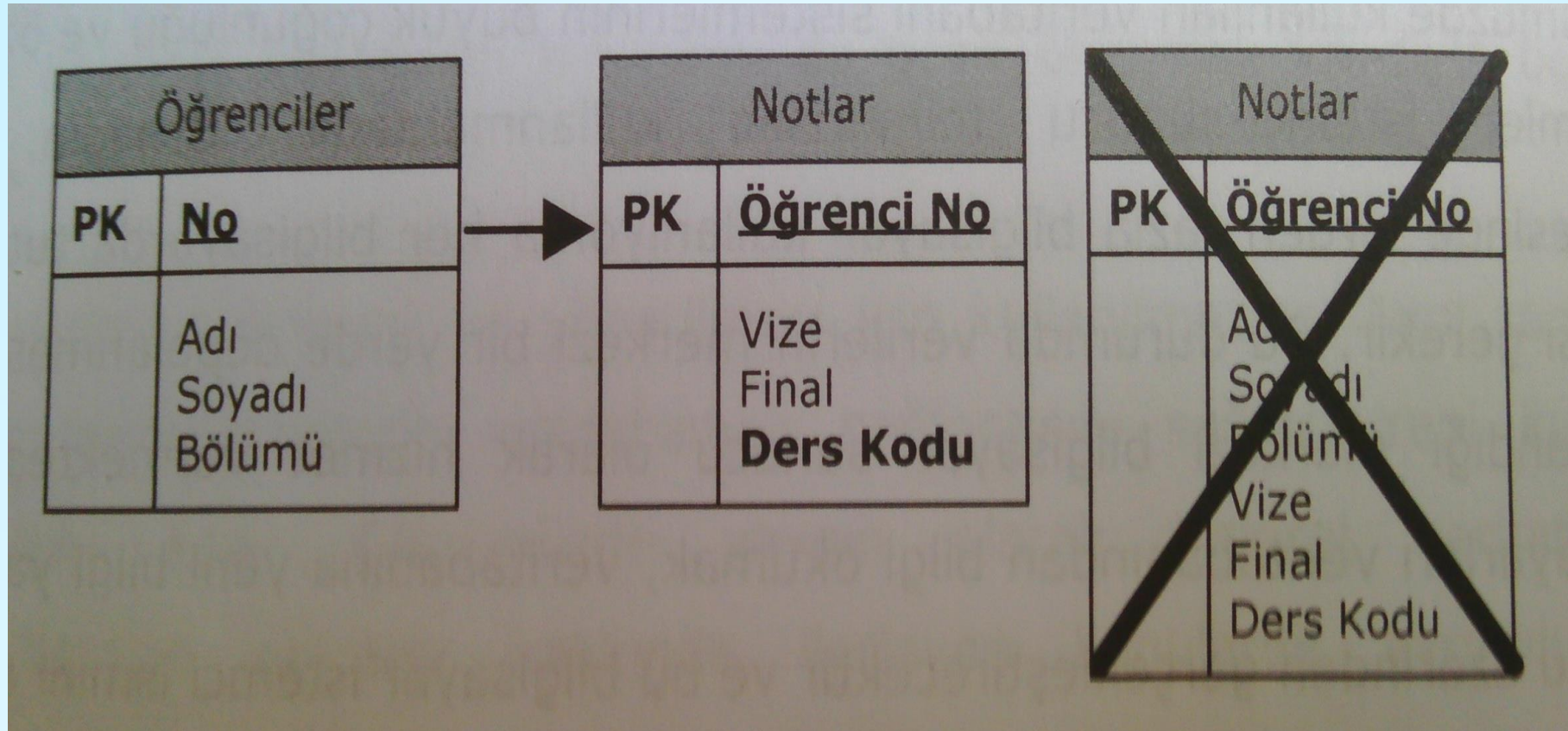


Şekil 1.13. Örnek bir öğrenci bilgi sisteminin veritabanı alt yapısı.

VTYS'nin Yararları

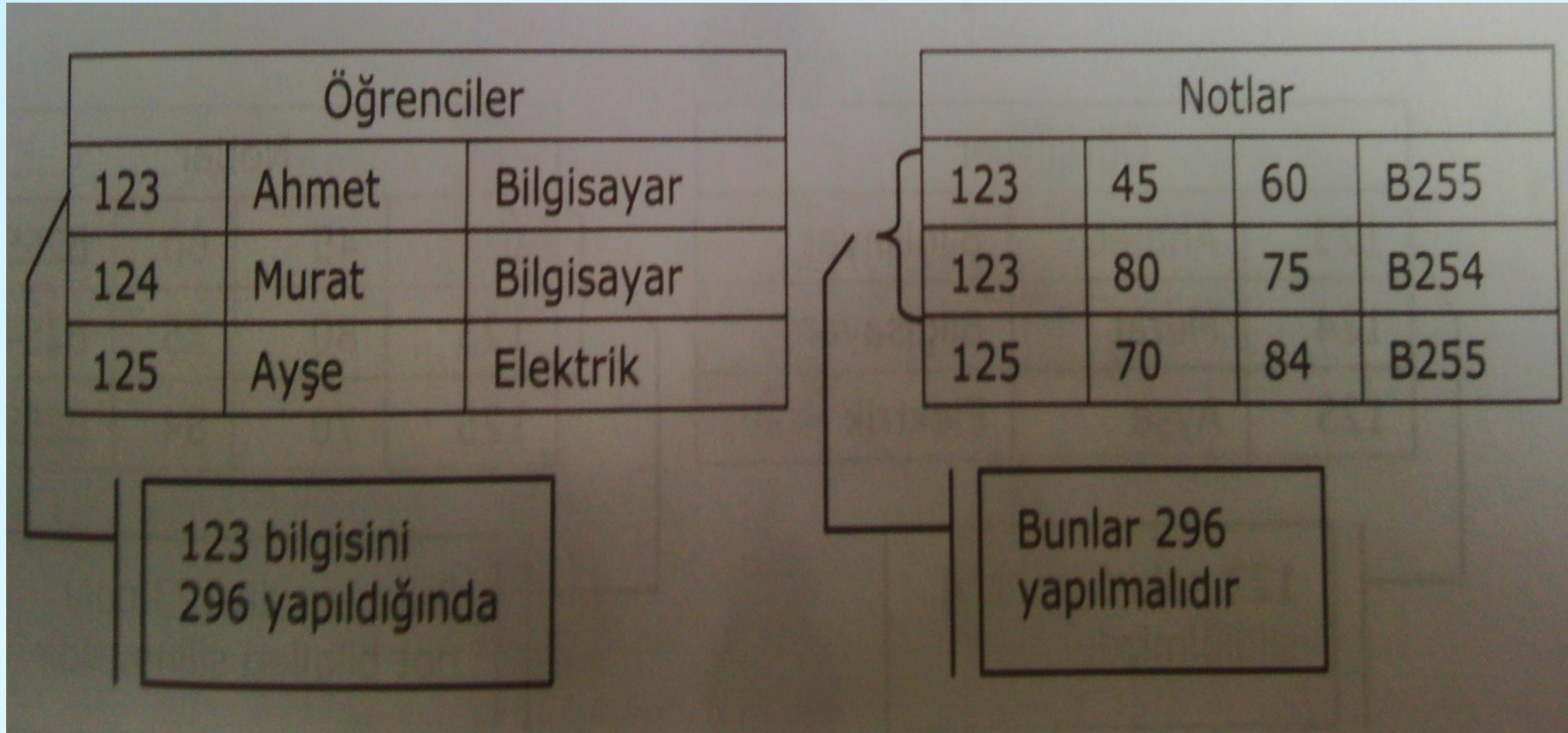
- A) Veri Tekrarı (Data Redundancy):
- B) Veri Tutarlılığı (Data Consistency):
- C) Verinin Eşzamanlı Paylaşımı (Data Concurrency):
- D) Veri Bütünlüğü (Data Integrity):
- E) Veri Güvenliği (Data Security):
- F) Veri Bağımsızlığı (Data Independence):

A) Veri Tekrarı (Data Redundancy)



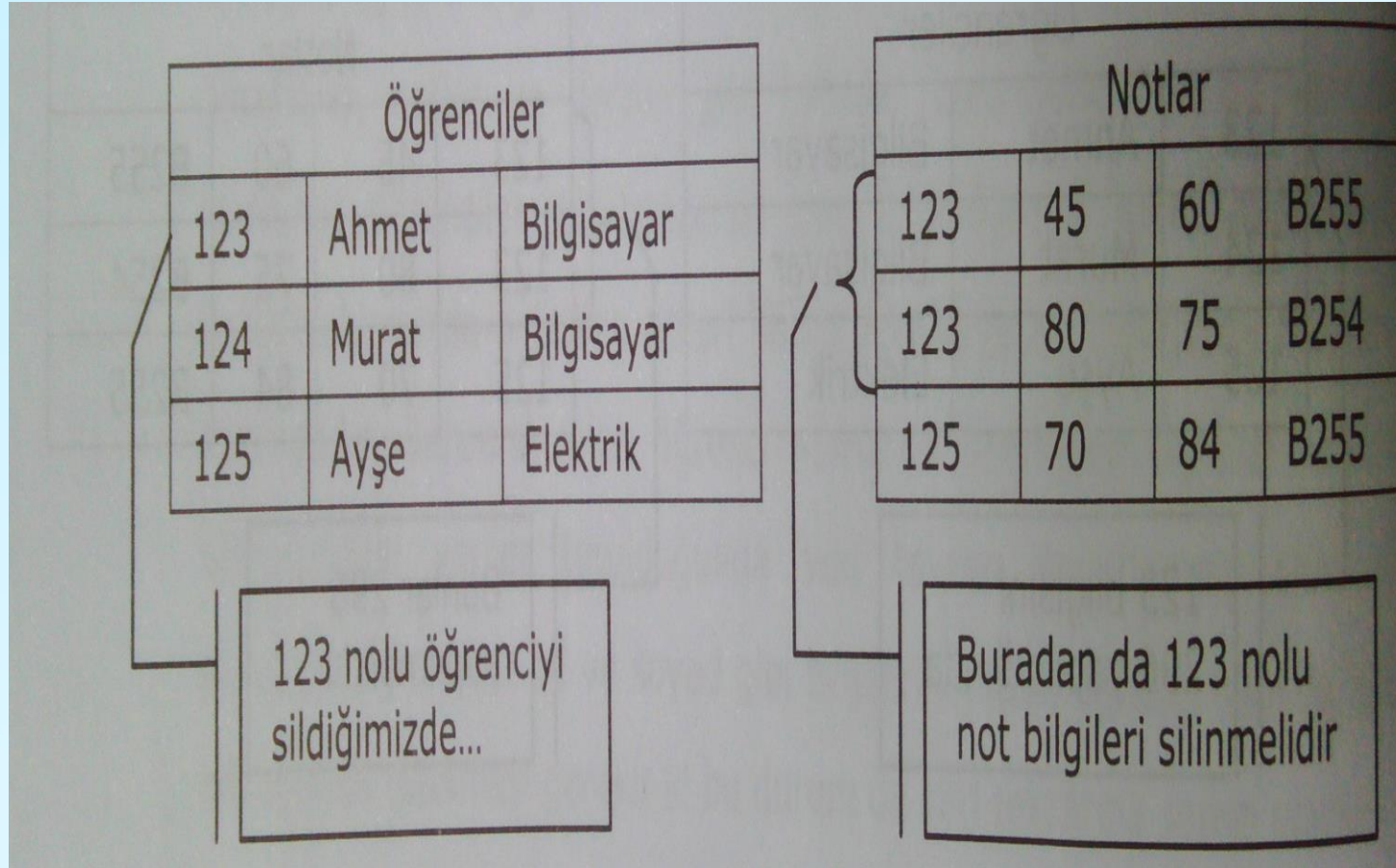
- Tablo içerisine girilen bir bilginin diğer tabloya tamamen girilmesi engellenerek veri tekrarı ortadan kaldırılır.

B) Veri Tutarlılığı (Data Consistency):



- Bir yerdeki veri güncellenirse diğer tablodaki verinin de güncellenmesi gerekir. Bu işlem gerçekleşmez ise veri tutarsızlığı oluşur.

C) Veri Bütünlüğü (Data Integrity):

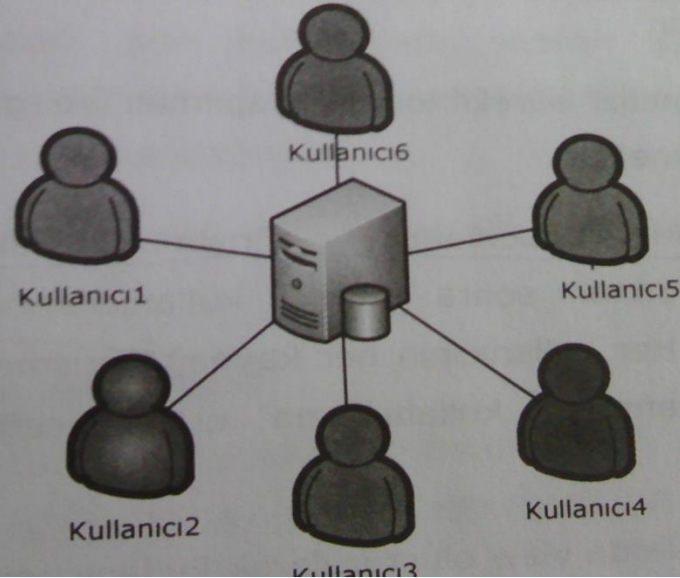


- Verinin farklı tablolarda bölündüğü durumlarda, tüm verilerin bir bütün olarak kullanılmasını sağlar. Örneğin bir veri silindiğinde o verinin ilişkili olduğu tüm tablolardan o veri silinmelidir.

D) Veri Güvenliđi (Data Security):

Kullanıcı Adı	Okuma	Ekleme	Deđiřtirme	Silme
Kullanıcı1	Evet	Hayır	Hayır	Hayır
Kullanıcı2	Evet	Evet	Evet	Evet
Kullanıcı3	Hayır	Evet	Hayır	Hayır
Kullanıcı4	Evet	Evet	Evet	Hayır
Kullanıcı5	Hayır	Hayır	Hayır	Hayır
Kullanıcı6	Evet	Hayır	Evet	Hayır

- Yetki ve sorumluluk dahilinde veriye ulaşabilme...



E) Veri Paylaşımı

- İstemci-Sunucu mimarisi kullanılarak tek bir veritabanını ağ üzerinden birden fazla istemci için kullanıma açabilir ve istenirse istemci sayısı kısıtlanabilir.

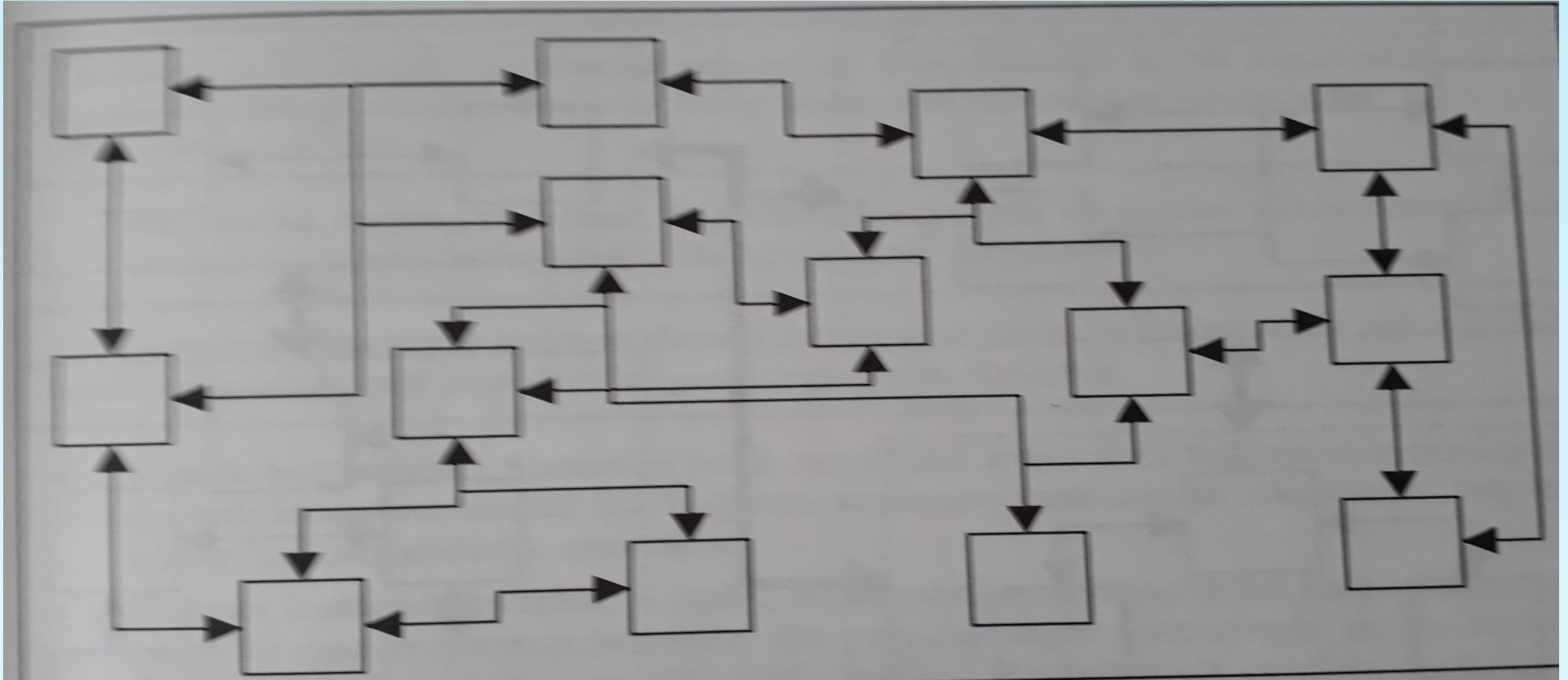
F) Veri Bağımsızlığı

- Karmaşık veri gösterimleri ve algoritmalarla ilgilenmeden kullanıcının isteği doğrultusunda isteklerini veri üzerinden yapabilmesidir. Fiziksel yapı kullanıcıdan gizlidir,

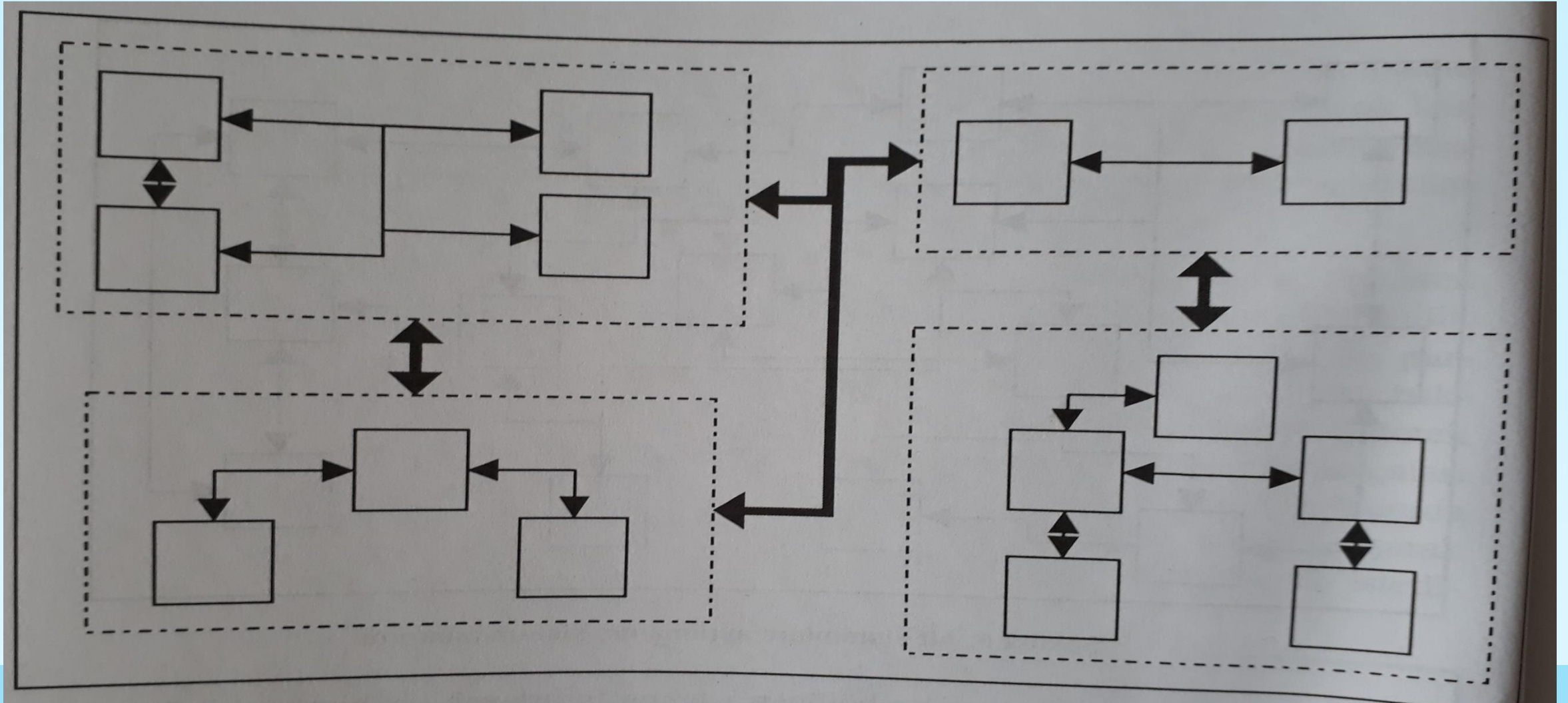
Problem Çözme ve Modelleme

- Problemi fark etmek, tanımlamak, gerekli bilgileri toplamak ve bir çözüm modeli tasarlamak çözüme gidişte genel kabul görmüş bir yaklaşımdır.
- Kısacası; tespit, teşhis, tedavi, analiz sentez; **tahlil, terkip ve inşa...**
- **Batılı bir yaklaşımla ☺ böl ve yönet...**
- **Analiz**, problemin **küçük parçalara ayrılması...**
- **Sentez**, ise bunların bütünleşik çözümü oluşturacak şekilde tekrar **birleştirilmesi işlemidir.**
- *Veriler analiz edilirken varlık-ilişki diyagramı, tasarım halinde ise ilişkisel tablo model diyagramları kullanılır...*

Alt sistemlere ayrılmamış sistem tasarımı (X)



Alt Sistemlerden oluşan Sistem Tasarımı (✓)



Modelin/Problemin Parçalarının Belirlenmesi

- Problemi çözmek için hangi temel bilgileri bilmemiz gerekir?

Sorusuna cevap bulabilmek için...

- **5N1K (Ne, Nerede, Ne zaman, Niçin, Nasıl, Kim)**
 - **Kim ?** (Projede takip edilecek kişi, birim, ünvanlar... Genel Müdür, memur, işçi gibi ünvanlar ile, Üretim, satış, mali işler gibi birimler)
 - **Ne?** (İşin Tanımı ve niteliği ile ilgili bilgiler, açık, net ve diğer işlerden ayrılan özellikler..)
 - **Ne zaman?** (Projenin veya işin Başlangıç ve Bitiş tarihleri, Kişilerin işe giriş ve çıkış tarihleri, tekrarlı işlerin başlangıç ve bitişi, bütçe, vergi dönemi vs..)
 - **Nerede?** (Yer kavramı: coğrafi konum, adres, bina, otelde belirli odalarda kalma, bilet yeri vs..)
 - **Niçin?** (Hedef: Müşteri memnuniyeti, satış performansı, ürün geliştirme sürecinin verimi vs..)
 - **Nasıl?** (Yöntem: İzlenecek yol, kullanılacak araçlar, metot ve biçim ne olacak...)

Modelin Kalitesi

- Bir yazılımın çalışması kaliteli olduğunu göstermez.
- **İhtiyaca cevap vermesi (Beklenileni karşılıyor mu?)**
- **Kemalat** (Yeterince gelişmiş mi, olgun mu?, ekleme veya çıkarmaya gerek duyuyor mu? Tüm problemlere çözüm sağlamış mı?)
- **Basitlik ve Kolay anlaşılabilme (Karmaşa yönetilebiliyor mu? yoksa Basitlik izafidir, kime göre neye göre...)**
- **Standartları taşıyor mu? (Geliştirmeye ve anlamaya uygun mu?)**
- **Esnek ve bağımlılık (Modül içi bağlantı max, modüller arası bağlantı min.)**
- **Tekrar Kullanılabilirlik** (Alt sistemler başka sistemlerde de kullanılabilir mi?)
- **Esneklik (sonradan çıkan ihtiyaçlara cevap verebilir mi, entegrasyon)**
- **Performans** (Ölçülebilir kriterleri karşılıyor mu?)
- **Bakım Kolaylığı (Sonradan destek verecek kişiler anlayabilir mi?)**
- **Hata Toleransı** (Sistemin hata tespit, kendi içerisinde çözümleme ve hatayı anlaşılır şekilde diğer sistemler ve kullanıcılara iletme kabiliyeti)

Denge Kavramı

Aşağıdaki yapılar birbiri ile ahenk ve uyum içinde olmalıdır.

- **Performans**
- Esnek Yapı
- Anlaşılabilirlik
- Kolay Kullanabilirlik
- **Verimlilik**

Örneğin; çok esnek bir yapı performans kaybına yol açar. Fakat sistem esnek değilse bu sefer de ortaya sonradan çıkan en basit ihtiyaç karşılanamaz.

Yukarıdaki değişkenlerden bir tanesinin aşırıya gitmesi diğerini için tam ters bir netice verebilir. (Haddini aşan zıddına dönüşür.)

Bu yapılar bir projenin başarılı/başarısız olması durumunun ölçütüdür.

Modelin Tanımı Ne zaman biter?

Bir modelin tamamlandığı ne zaman anlaşılır?

Test ederek denenebilir. Ancak, test model üzerinden sistem gerçekleştirilerek yapılabilir. Fakat maliyeti oldukça yüksektir.

Modelin sistem tasarımına başlanması için yeterli hale geldiğini de fark etmek gerekir. Aşağıdaki 2 soru bize yardımcı olabilir.

- **Modelin kullanıcı isteklerinin tamamını nasıl karşılayacağını açıkça belli olması? Tamamlandığına işarettir. // şunları yeter... ya şu da olsaydı**
- **Ayrıca Modeli tekrar tekrar incelediğimizde modele önemli eklemeler yapılmıyor sa artık sonraki aşamaya geçmeye hazırız demektir.**

İhtiyaç Sınırlarının belirlenmesi (Onun için yazılım şartnameleri hazırlanıyor...)
Versiyonlu ve Uyarlamalı çalışma..

Özetle

- Problem çözme ve sistem modelini oluşturma önemlidir.
- Problemi çözmek için problemi oluşturan parçaların tespit edilmesi gerekmektedir.
- Sistemi anlayabileceğimiz büyüklükte parçalara ayırmamız gerekmektedir.
- Parçalar arası ilişkileri belirleyerek genel model oluşturulur. Bu aşamada modelleme yöntemlerinden uygun olanı seçiniz.
- Modelin kalitesini basitlik, esneklik, standartlara uygunluk, bakım vs. gibi objektif kriterlere göre ölçmek gereklidir.

Modelleme Yöntemleri

- Modelleme yöntemleri, modeli oluşturan parçaların incelenmesi ve parçalardan genel sisteme ulaşmak için izlenecek yolu ifade eder.

1-) Özetten Ayrıntıya Modelleme

2-) Ayrıntıdan Özete Modelleme

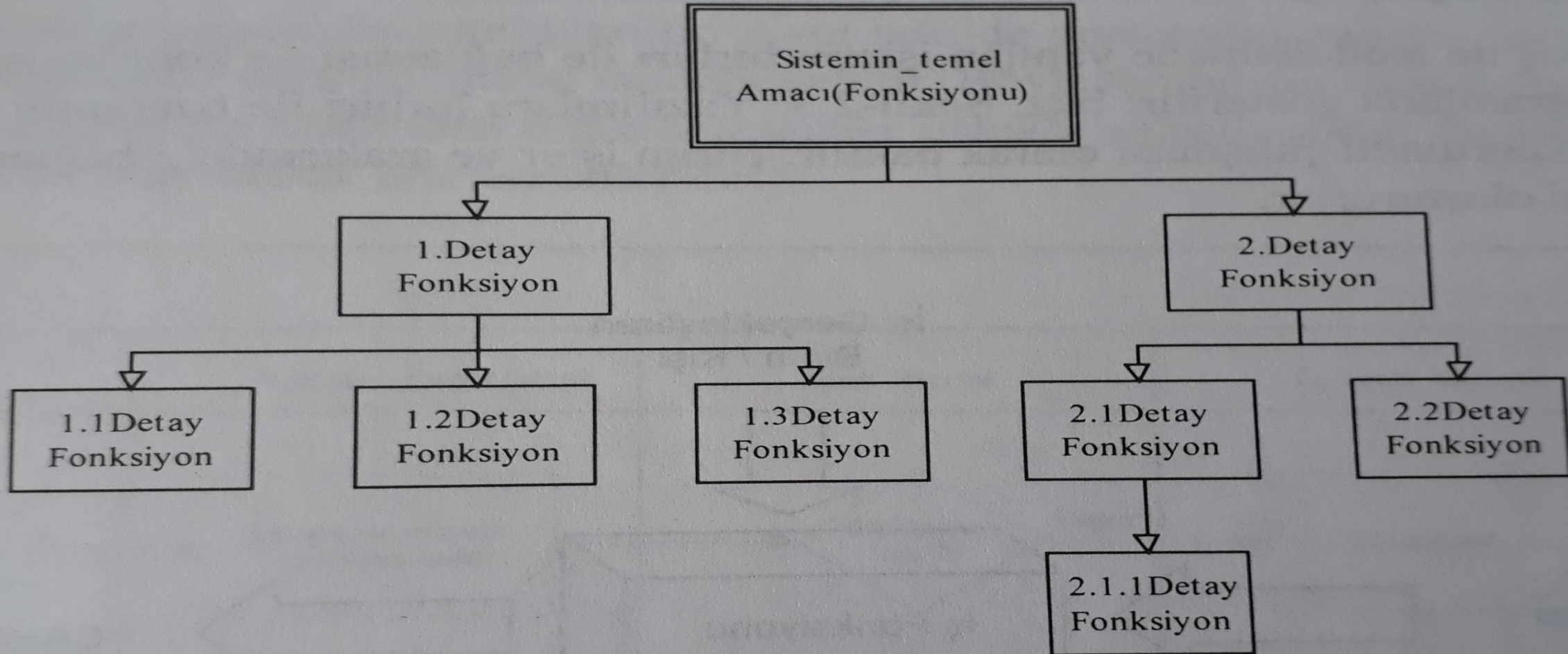
3-) İş Akış Temelli Sistem Modellenmesi

4-) Nesne Tabanlı Modelleme

5-) Varlık-İlişki Modelleme

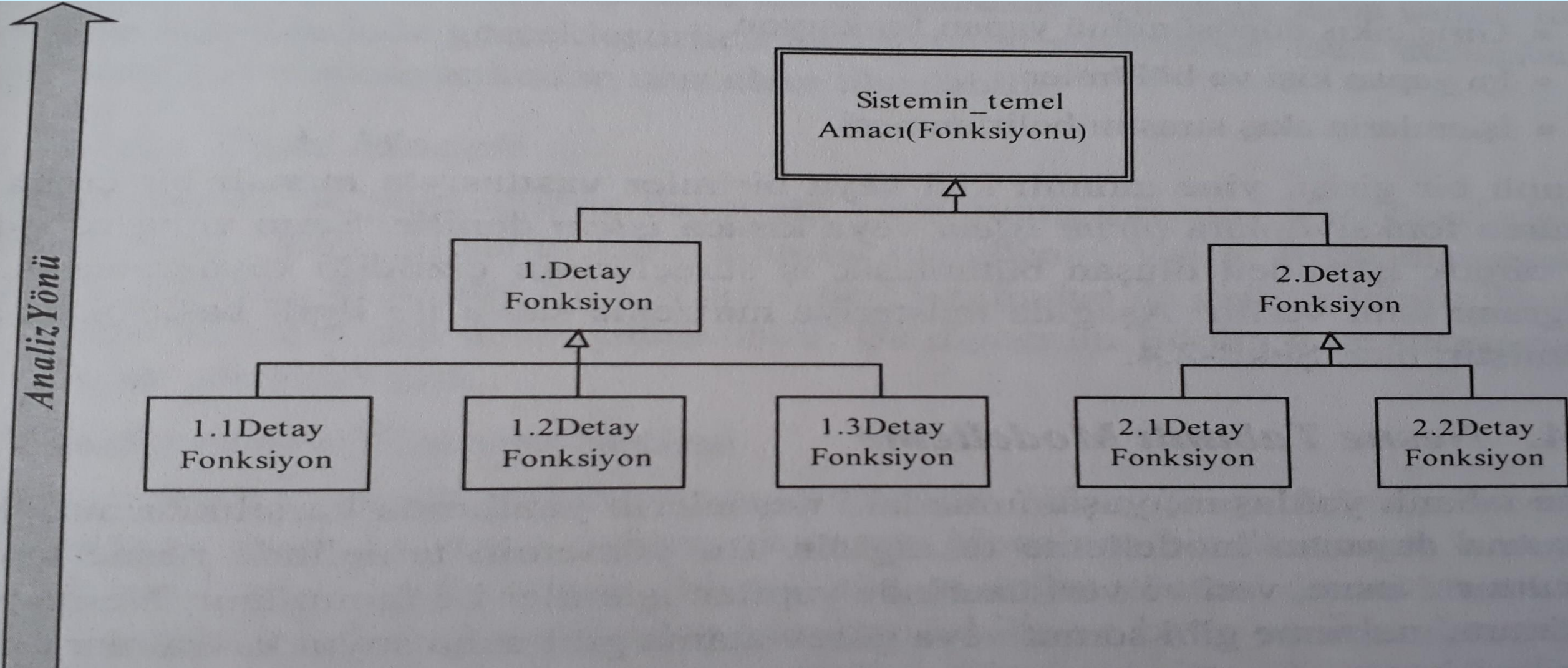
1-) Özetten Ayrıntıya Modelleme

- Özetten ayrıntıya modelleme yapılırken öncelikle sistemin temel fonksiyonu belirlenir. Daha sonra en alt fonksiyonlara kadar ulaşılır.



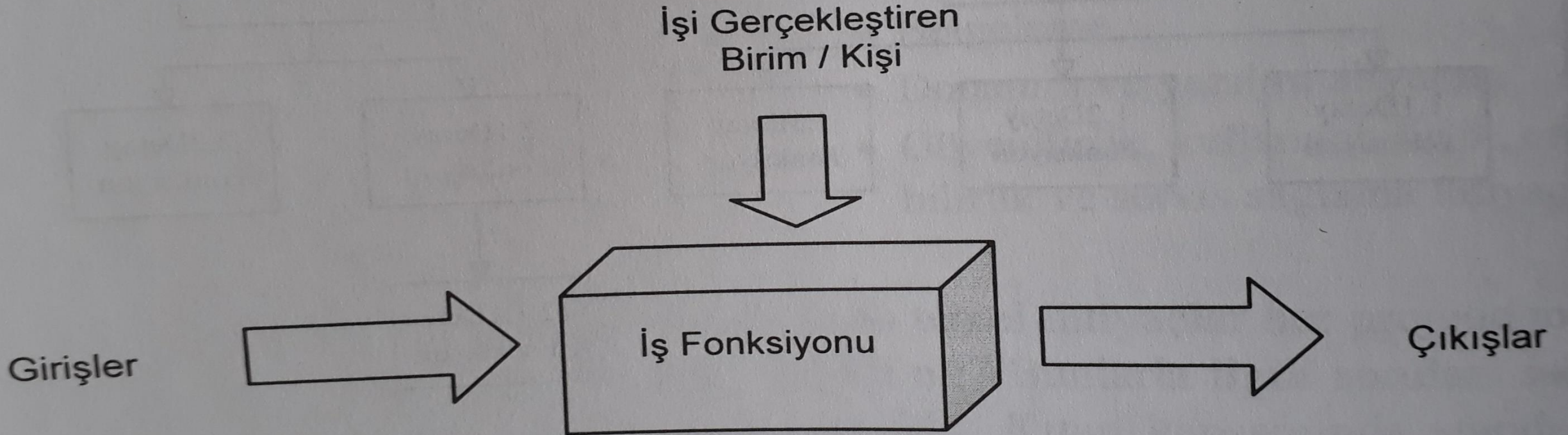
2-) Ayrıntıdan Özete Modelleme

- Beyin fırtınası yaklaşımı ile yakın bir inceleme yapılır. İlk olarak sistemle ilgili ilgili bilgiler hızlı bir şekilde toplanır, birleştirilir ve üst fonk. oluşur.

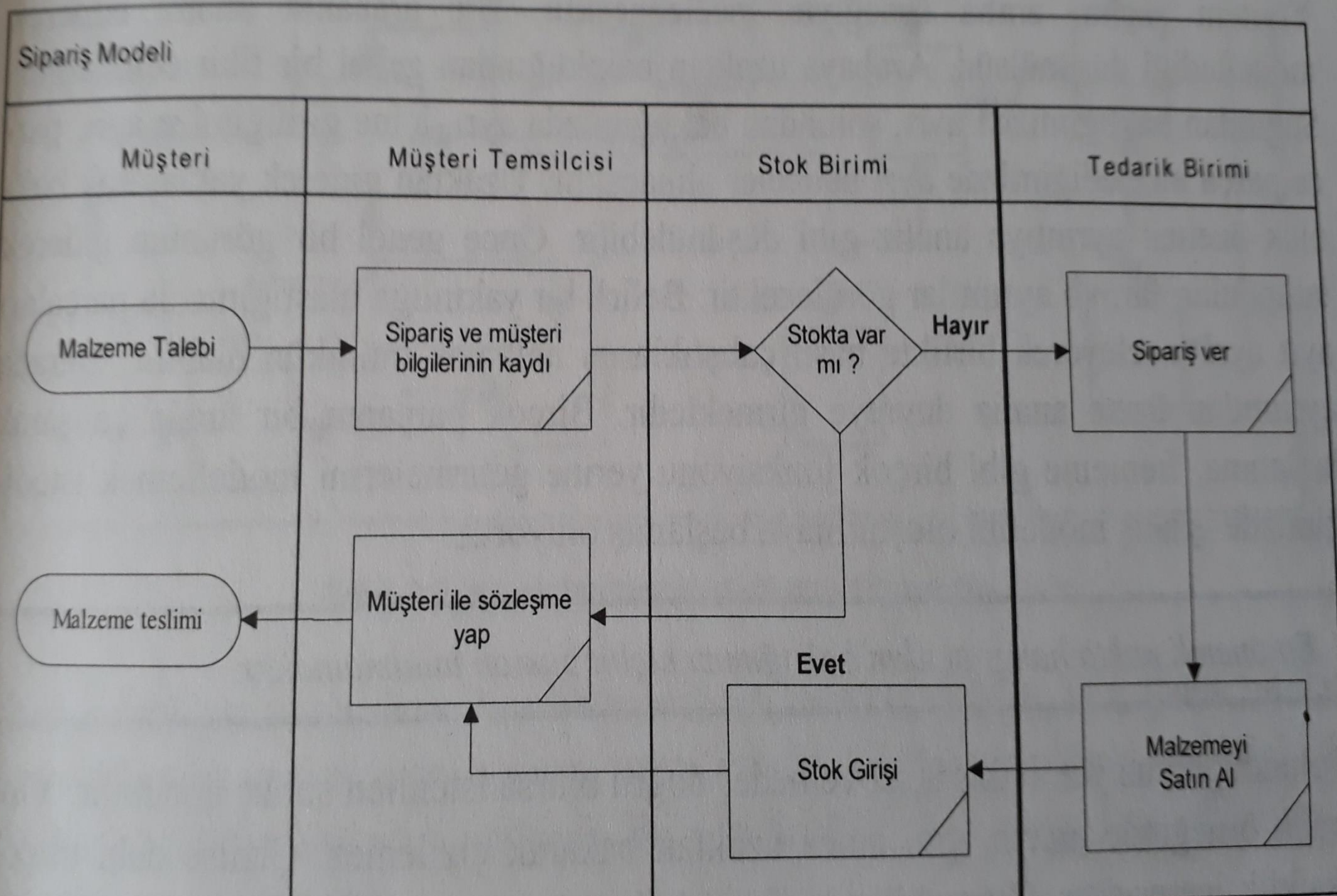


3-) İş Akış Temelli Sistem Modellenmesi

- Bu modelde yapıların birbiri ile bağlantısı ve koordinasyonu görsel diyagramlarla gösterilir. Yazılımlara birbiri ile bağlantılı birçok işlemin koordineli çalışması olarak bakılır. Birim işler arasındaki bağlantılar iş modelini oluşturur.



4-) Nesne Tabanlı Modelleme



- Nesne, veri ve veri üzerinde yapılan işlemler olarak tanımlanır. Nesne personel, fatura malzeme gibi somut ya da görev, atama gibi soyut kavramlar olabilir. Nesne, nitelik, olay yapıları ile model kurulur.
- Nesne = Personel, Nitelikleri= adı, soyadı, sicili vb. , methodu= personelin transferi gibi model kurulabilir, olay= işten çıkış

5-) Varlık-İlişki (Entitiy-Relation) Modelleme

- ER modelleme veriler arası ilişkilerin görsel (şema) sunumudur. ER Modelinde, verileri varlıklara ayırır, varlıklar arasındaki öznitelikleri ve kurulum ilişkilerini kurarız, bunların tümü ER şeması kullanılarak görsel olarak gösterilebilir.

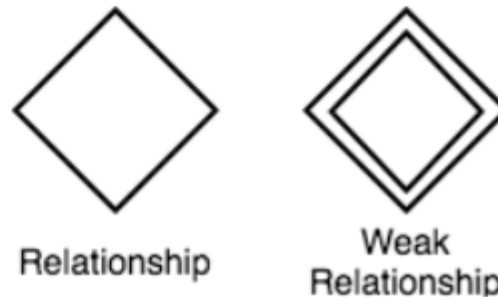
Entity

Simple rectangular box represents an Entity.



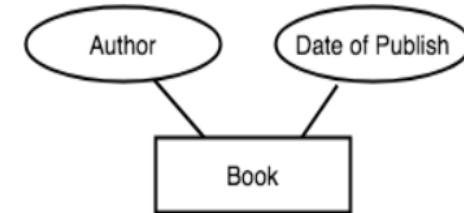
Relationships between Entities - Weak and Strong

Rhombus is used to setup relationships between two or more entities.



Attributes for any Entity

Ellipse is used to represent attributes of any entity. It is connected to the entity.



Weak Entity

A weak Entity is represented using double rectangular boxes. It is generally connected to another entity.



Key Attribute for any Entity

To represent a Key attribute, the attribute name inside the Ellipse is underlined.



5-) Varlık-İlişki (Entity-Relation) Modelleme

Derived Attribute for any Entity

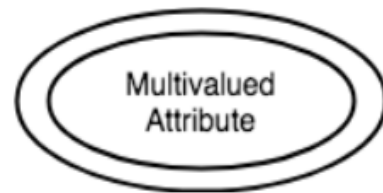
Derived attributes are those which are derived based on other attributes, for example, age can be derived from date of birth.

To represent a derived attribute, another dotted ellipse is created inside the main ellipse.



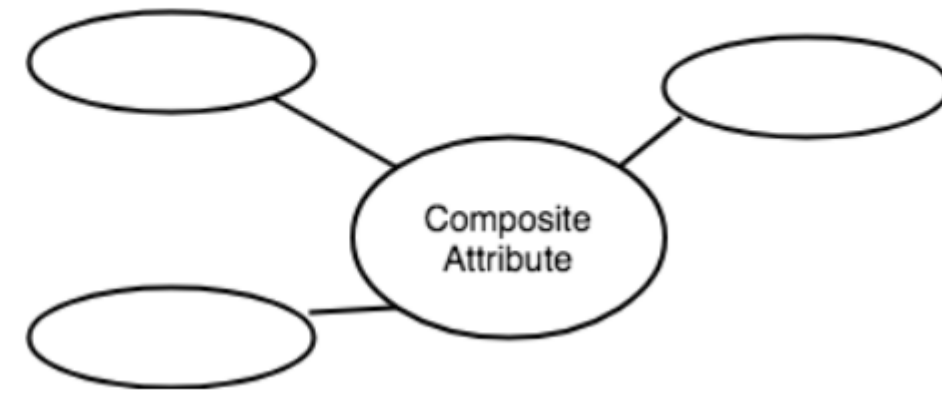
Multivalued Attribute for any Entity

Double Ellipse, one inside another, represents the attribute which can have multiple values.



Composite Attribute for any Entity

A composite attribute is the attribute, which also has attributes.



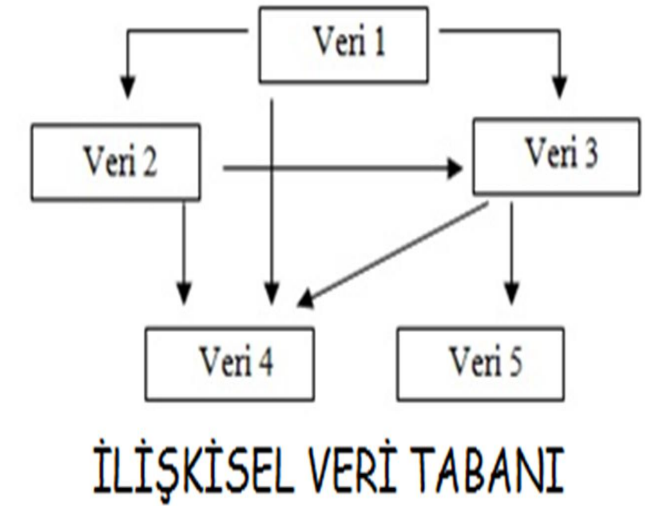
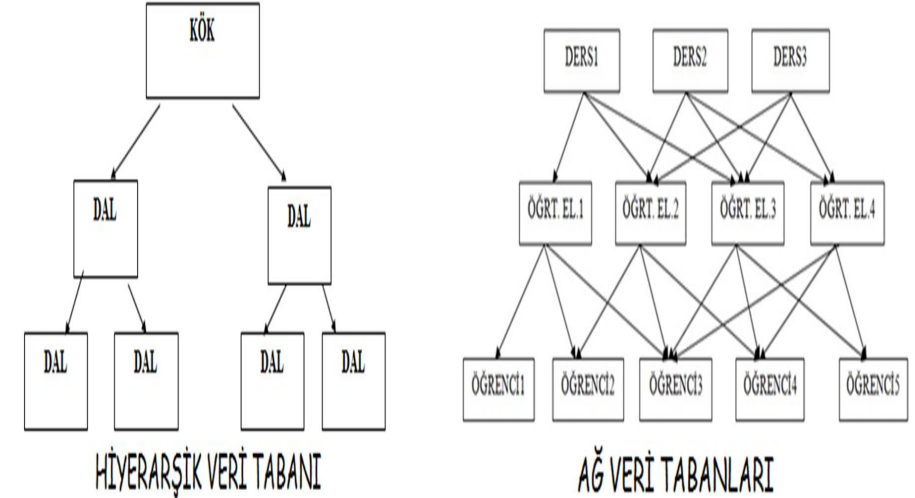
<https://www.studytonight.com/dbms/er-diagram.php>

Model Seçimi ve Yazılımın Modüler Geliştirilmesi

- Bir projede farklı yöntemlerin aynı anda kullanılması mümkündür.
- Amaca ve ihtiyaca göre kullanılması gereklidir.
- Yapılması gereken bir defada tüm karmaşıklığı yönetmeye çalışmak değil, alt parçalara/problemlere/**alt sistemlere/bölmelere** bölerek parça parça yönetmek veya yaklaşımdır.
- Avantajları: Tasarım kolaylığı, Tekrar Kullanabilme, Projedeki Görev Paylaşımı Kolaylaşır, Kurumların çalışması ve modülerlik

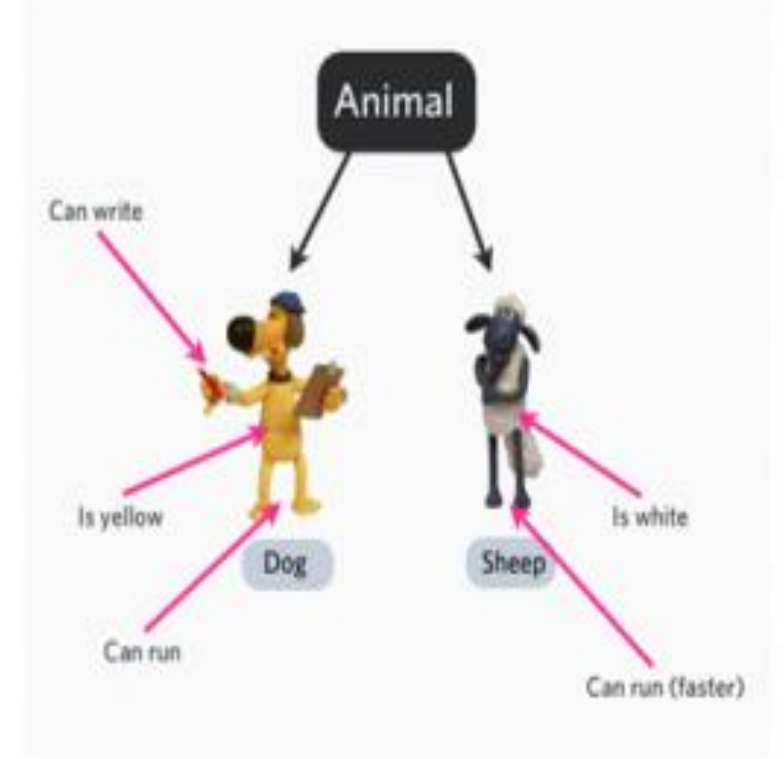
VERİ MODELLERİ

- Veri Modeli: Verilerin depolanması, işlenmesi, veriler arası ilişkilerin kurulması biçimidir. Verileri mantıksal düzeyde düzenlemek için her VTYS belirli bir veri modelini kullanır.
 - SIRADÜZENSEL VERİ MODELİ (HIERARCHICAL DATA MODEL)
 - AĞ VERİ MODELİ (NETWORK DATA MODEL)
 - İLİŞKİSEL VERİ MODELİ (RELATIONAL DATA MODEL)
 - NESNEYE YÖNELİK VERİ MODELİ (OBJECT ORIENTED DATA MODEL)
- Günümüzde kullanılan VTYS'lerin hemen hemen tümü ilişkisel veri modeline dayanır.



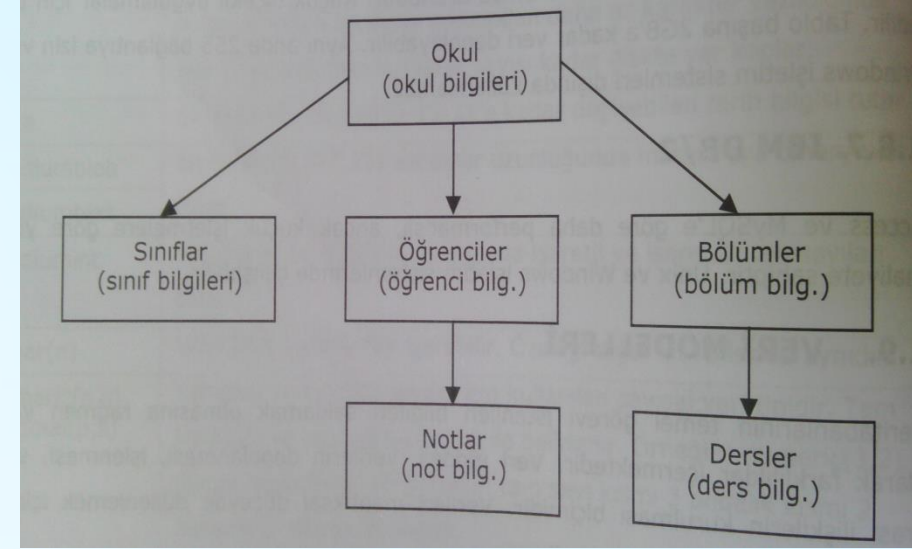
Nesneye Yönelik Veri Modeli

- Veriler nesne olarak modellenir ve yaratılır.
- Nesneye-Yönelik Programlama (OOP) da olan sınıf ve miras kavramlarına sahiptir.
- Karmaşık veriler üzerinde işlem yaparken yüksek performans sunan bir yaklaşımdır.
- İlişkisel Modelin iyi bilinen özelliklerini Nesne Tabanlı özelliklerle birleştirir. Bunlar:
 - Kullanıcı tanımlı veri türleri
 - Kullanıcı tanımlı fonksiyonlar
 - Kalıtım ve alt sınıflar



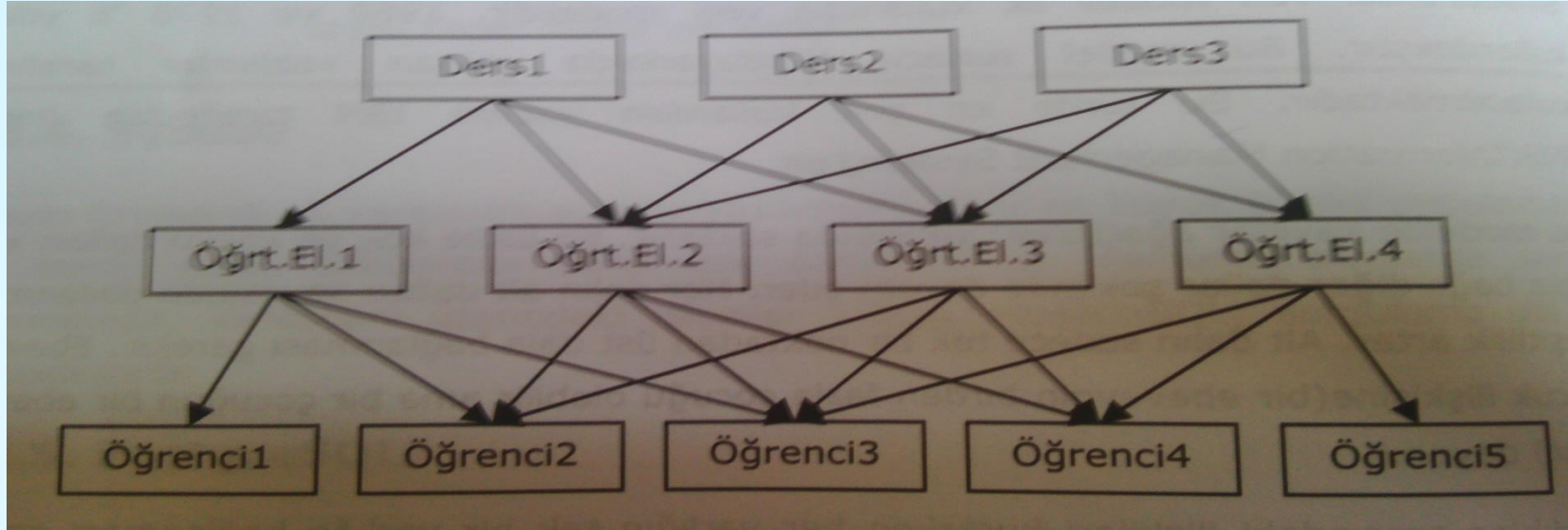
SIRADÜZENSEL VERİ MODELİ (HIERARCHICAL DATA MODEL)

- 1960 ve 1970 li yıllarda yaygın olarak kullanılmıştır. Sunucu bilgisayarlarda kullanılan yazılımdır
- IBM tarafından IMS (Information Management System) yazılımı geliştirilmiştir.
- Bu modelde kayıtlar AĞAÇ (TREE) yapısında saklanır. En üstte KÖK dizin ve KÖKÜN dalları, ve bu dallara dayalı alt dallar şeklinde devam eder.
- Ebeveyn Çocuk ilişkisi modeline dayanır.
- Herhangi biri bulunmak istendiğinde arama tepe noktadan köke doğru devam eder.



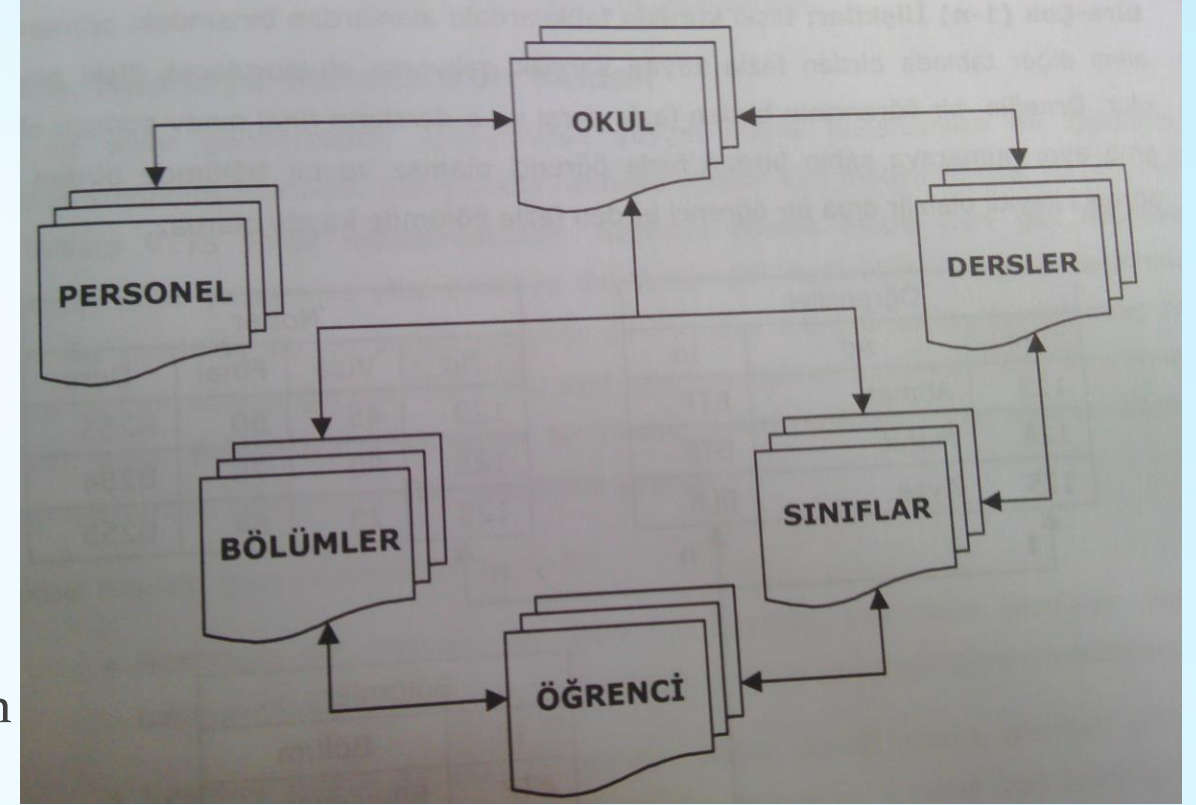
AĞ VERİ MODELİ (NETWORK DATA MODEL)

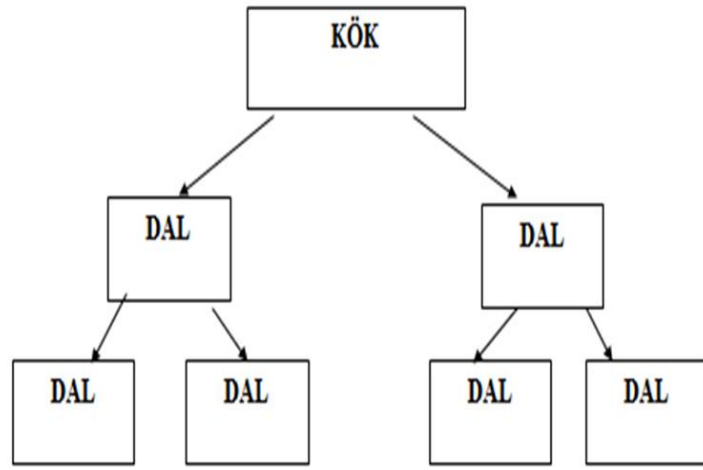
- 1970-1985 arası yaygındır.
- Sıradüzensel model yetersiz kalınca, bir verinin birden fazla veri ile ilişki kurabilmesi açısından ağ veri modeli fikri çıkmıştır.
- Her kaydın birden çok ebeveyni ve birçok çocuk kaydı olabilir.



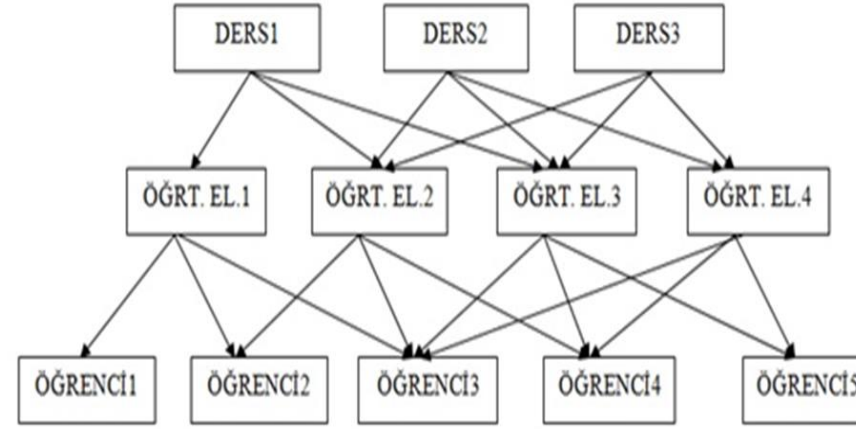
İLİŞKİSEL VERİ MODELİ (RELATIONAL DATA MODEL)

- 1985 lerde günümüze kadar hemen hemen tüm sistemlerin ilişkisel olduğu söylenebilir.
- Bu modelde ortak özelliğe sahip veriler tablolar aracılığı ile tutulurlar.
- Tablolar satır ve sütunlardan oluşur. Her sütun ayrı bir veri tutar. Ve buraya ALAN (field) denir. Her alanın tutacağı veri tipi belirlenir.
- Aynı satırdaki veriler aynı kaydı ifade eder.
- Anahtarlar kullanılır.
- Anahtarlar sayesinde indexleme yapıldığı için işlemler daha hızlı yapılır.

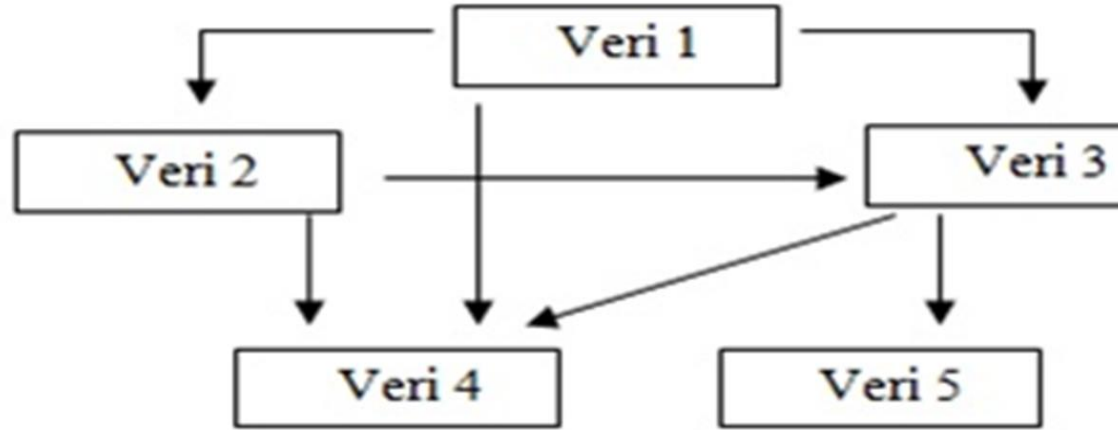




HİYERARŞİK VERİ TABANI



AĞ VERİ TABANLARI



İLİŞKİSEL VERİ TABANI

student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88