

Veritabanı Yönetim Sistemleri

Dr. Öğr. Üyesi Ahmet Arif AYDIN

2-

Dosya Sistemleri ve Problemler

Veritabanı Yönetim Sistemlerinin Kuramsal Seviyeleri

Veritabanı Tasarım Aşamaları

- ☐ Veri ve bilgi kavramlarını açıklayınız?
- ☐ Veritabanı (database) nedir? Neden kullanılır
- ☐ VTYS 'nin kullanım alanları nelerdir ?
- ☐ IDS (Integrated Data Store) nedir?
- ☐ SABRE ?
- ☐ Edgar Codd kimdir? Bu dersimizle olan bağlantısı nedir?

Bir Elektronik-ticaret firmasının

- ❑ çok sayıda çalışanı, ürünleri, ve satışları bulunmaktadır
- ❑ Firmanın depolanması gereken verisinin boyutu da yaklaşık olarak 1TB olacağını varsayalım .



- ☐ Birden fazla kullanıcının (n) eş zamanlı olarak erişimi sağlanabilecek
- ☐ Yeni veri eklenebilecek
- ☐ Var olan kayıtlar güncellenebilecek
- ☐ İstenilen kayıtlar silinebilecek (çok nadir kullanılabilir)
- ☐ Sorgular makul süre içerisinde cevaplandırılacak
- ☐ Verinin istenilen bazı kısımlarına erişim kısıtlanabilecek

Eğer veri işletim sistemlerinin sağlamış olduğu dosyalara yazılacak olursa

- ❑ 1 TB'lık ana bellek (RAM) olmadığından verinin hepsi aynı anda belleğe alınamayacak.
- ❑ Veri hard disk üzerine kaydedilecek ve her defasında verinin bir parçası hafızaya alınıp işlenmesi gerekmektedir (access).
- ❑ Farklı formatlarda ve sayıda dosyalar kaydedilebilecek. Birden fazla kopya olabilecek (redundancy)

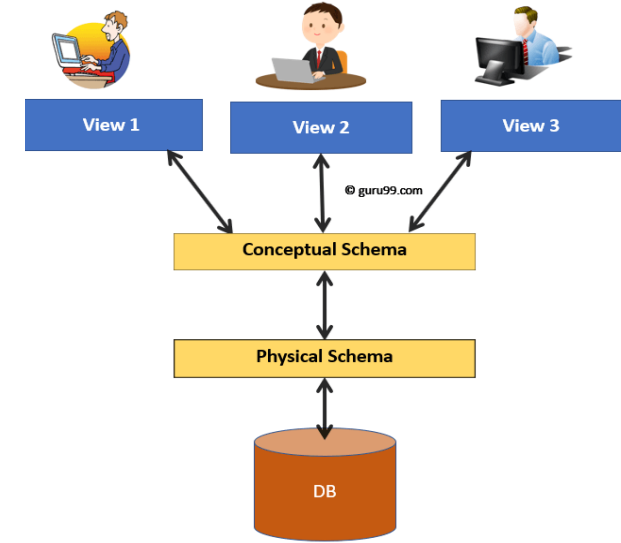
Eğer veri işletim sistemlerinin sağlamış olduğu dosyalara yazılacak olursa

- ❑ Kullanıcı sorularını cevaplamak için özel programların geliştirilmesi gerekmektedir.
- ❑ Farklı kullanıcılar tarafından eş-zamanlı erişilen verinin tutarlılığını korumak gerekmektedir (concurrent access). Ve her bir uygulamanın bunu sağlaması oldukça zordur.

Eğer veri işletim sistemlerinin sağlamış olduğu dosyalara yazılacak olursa

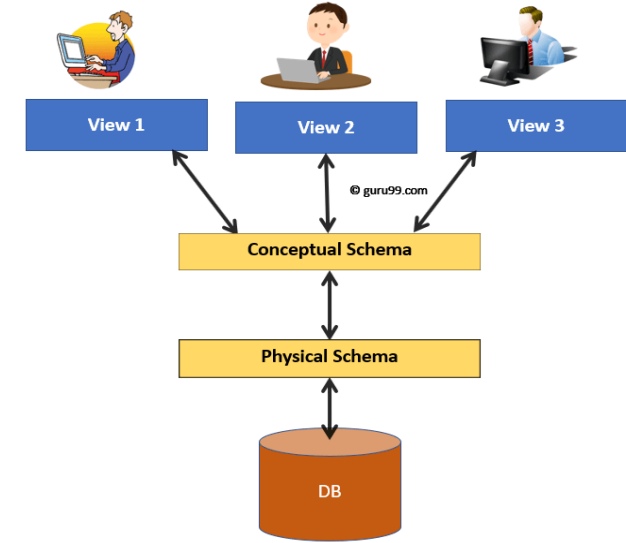
- ❑ Sistem çökmelerinde ve sistem üzerinde değişiklikler yapıldığında verinin tutarlı bir biçimde saklandığı garanti edilmelidir (consistency) fakat dosya sistemleri ile bu ihtiyacın karşılanması çok zordur.
- ❑ İşletim sistemleri sadece şifre güvenliği sağlamaktadır (security). Farklı kullanıcılar için izinleri ve şifreleri düzenleme noktasında dosya sistemleri yeterli seviyede esnek değildir.

- ❑ VTYS sağlamış olduğu veri bağımsızlığı (data independence), veritabanı katmanlarının birbirinden bağımsız olarak işlem gerçekleştirebilme imkanı sağlar
- ❑ Uygulama programları için farklı erişim seviyeleri tanımlanmasına imkan sağlar.
- ❑ VTYS 'leri verinin genel bir görünüşünü verip diğer detaylarını izole eder (abstraction)



Etkili Veri Bağlantısı (efficient data access)

- ❑ Veritabanı yönetim sistemleri veri ekleme, güncelleme, sonra tekrar okuma ve elde etme(erişim) işlemleri (CRUD) için sofistike teknikler sunmaktadır.



Veri Bütünlüğü ve Güvenliği (Data Integrity and Security)

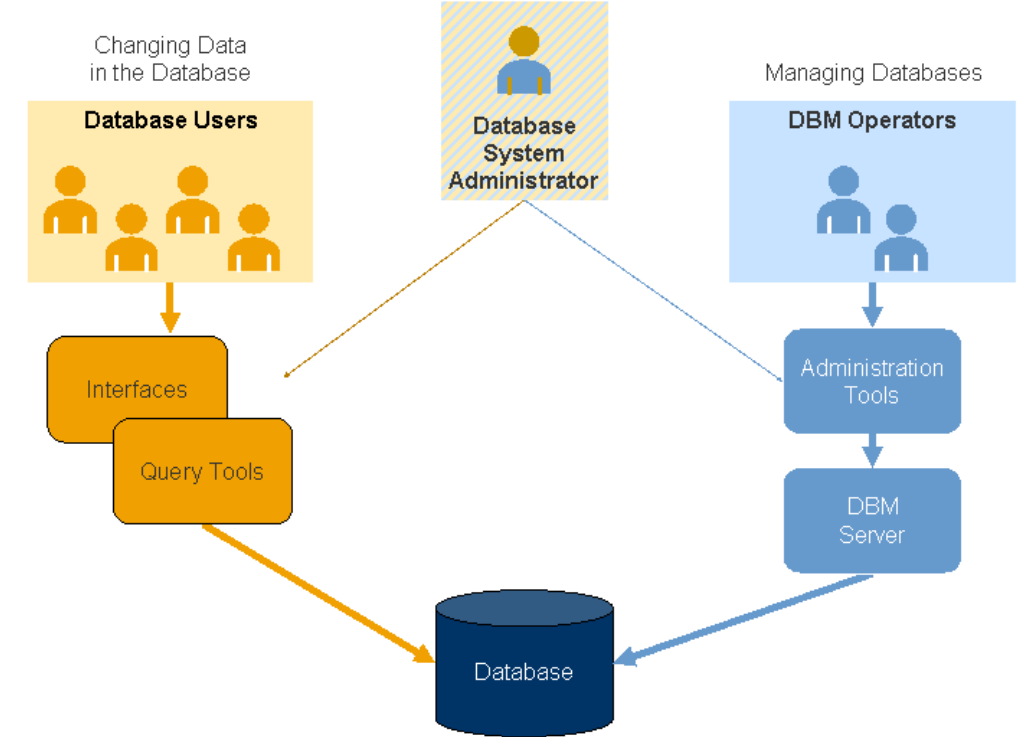
- ❑ VTYS 'leri üzerinden erişilen verinin bütünlüğü korunur.
- ❑ Birden fazla değişim isteği kısıtlanır.
- ❑ Farklı kullanıcılar için erişim denetimi (access control) sağlanır.



<https://2wtech.com/difference-between-data-security-and-data-integrity/>

Veri Yönetimi (Data Administration)

- ❑ Ortak kullanılan verinin yönetimi bir çok profesyonel tarafından tasarlanır ve düzenlenir.
- ❑ Verinin sunumunda tekrar ortadan kaldırılarak verinin çağırılması optimize edilir.



Eş zamanlı erişim ve kurtarma (Concurrent Access and Crash Recovery):

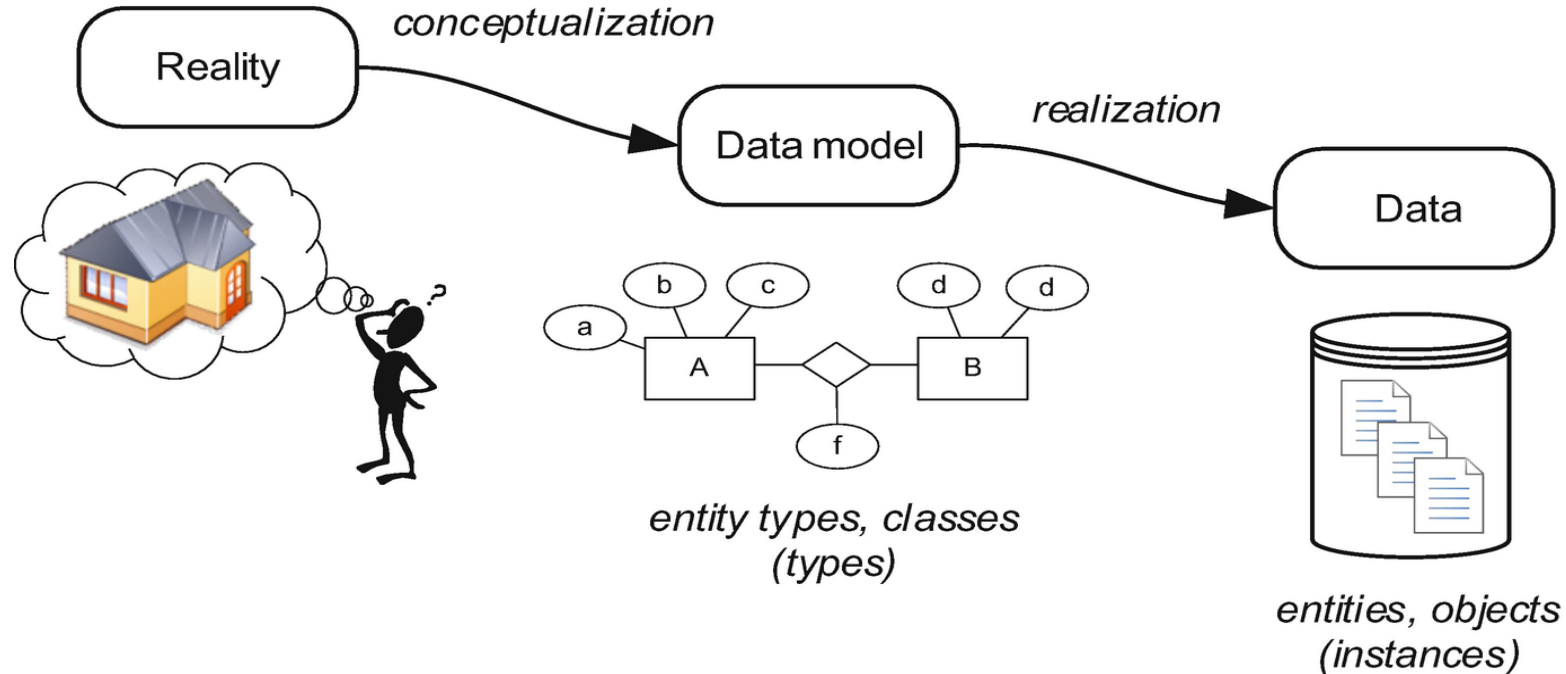
- ❑ VTYS eş-zamanlı erişimlerde her seferinde bir kullanıcının aynı veriye erişimini düzenler
- ❑ Sistem hatalarından ve etkilerinden kullanıcıları ve verileri korur.

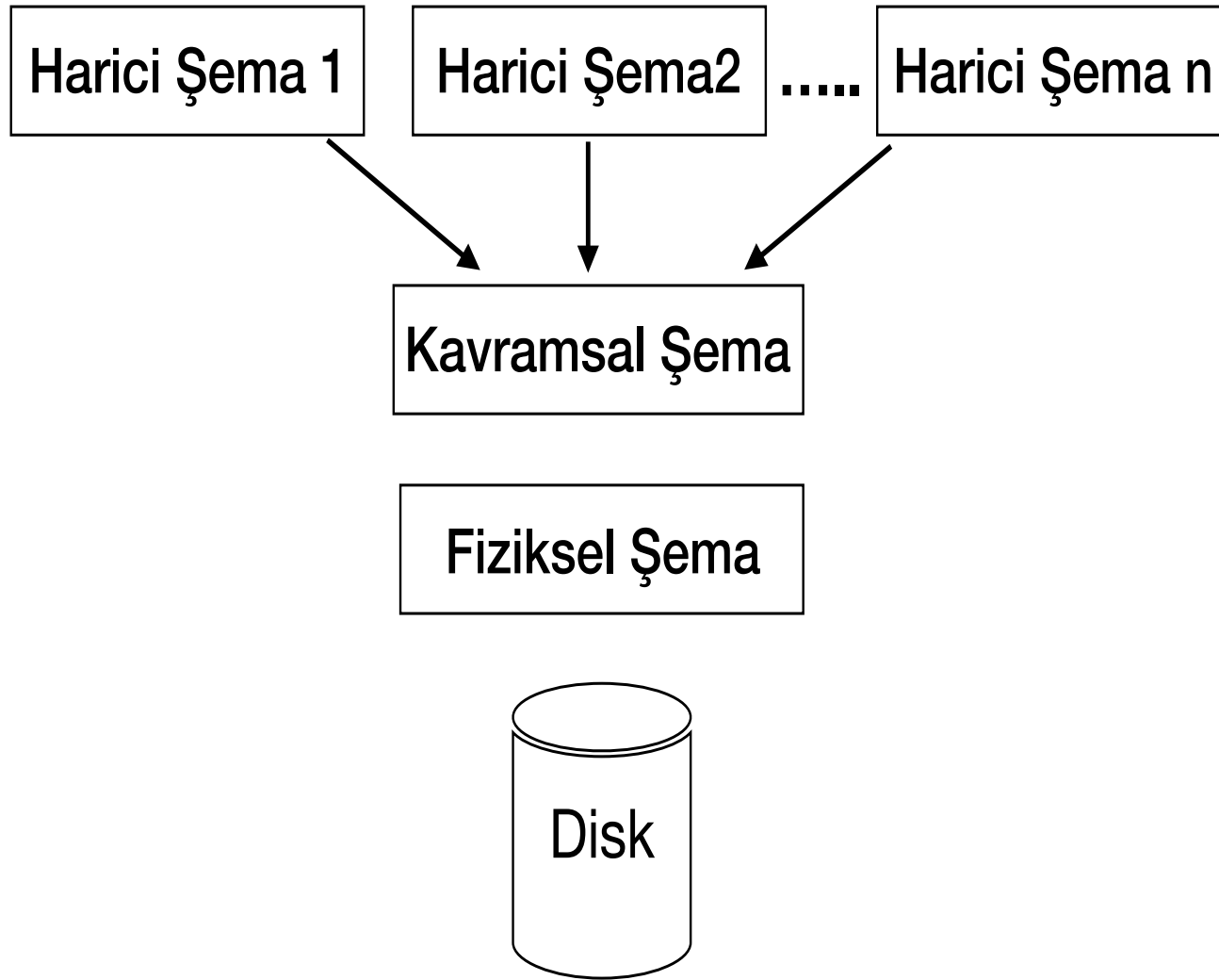
Azalan Uygulama geliştirme Zamanı (Reduced Application Development Time)

- ❑ VTYS' leri önemli fonksiyonları sağladığından tekrar en başından itibaren program yazılması için gereken zaman harcanmasını ortadan kaldırır.

Veri modelleme (data modeling)

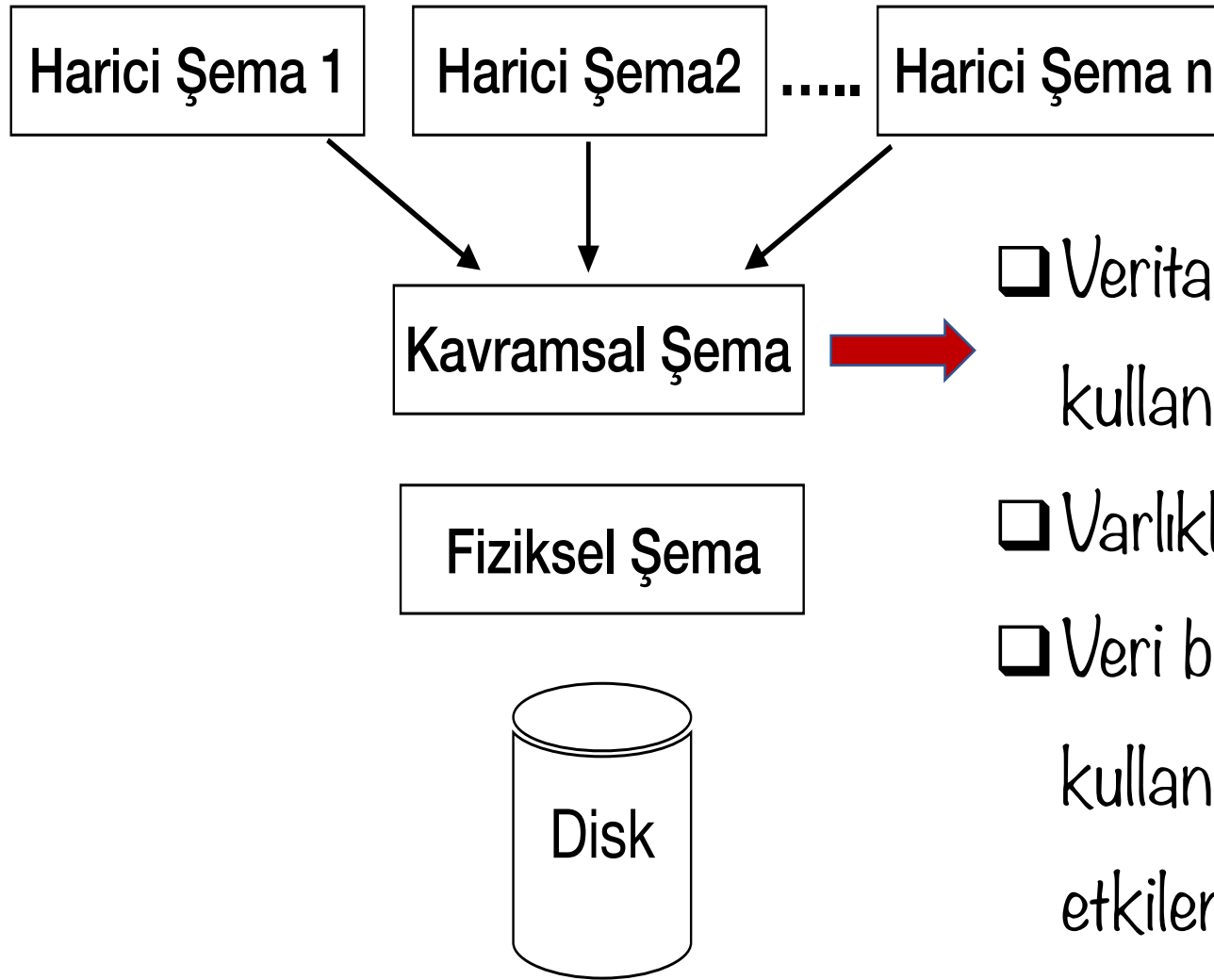
- ❑ Veri kolleksiyonlarının yüksek-seviyeli olarak tanımlanıp ve düşük-seviyeli bilgilerin gizlenmesini sağlayan temsil veri modeli (data model) olarak tanımlanmaktadır.
- ❑ VTYS'leri kullanıcılarına veri modelinin oluşturulması ve yönetimini sağlamaktadır.





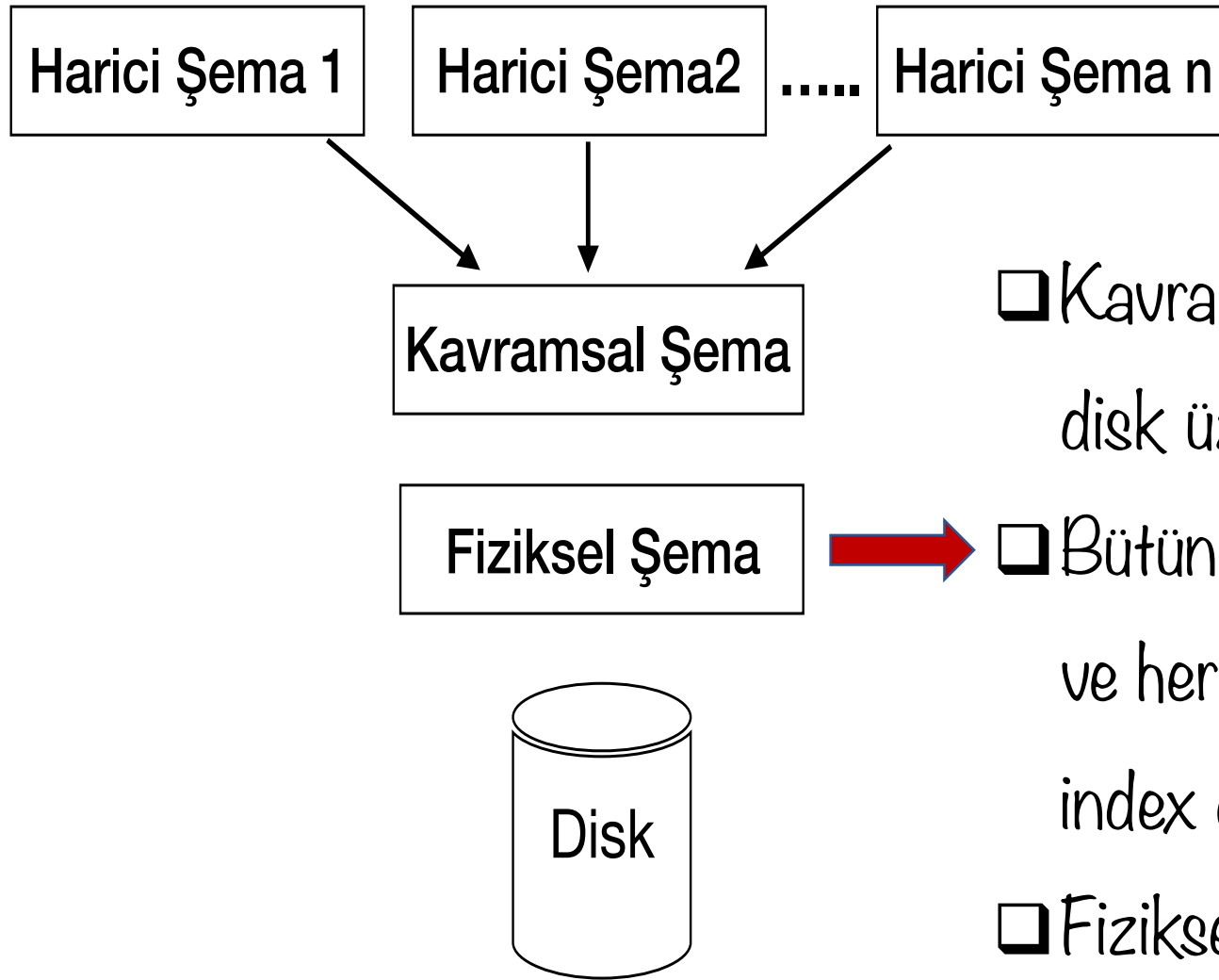
External Schema

- ❑ N tane harici şema olabilir.
- ❑ Farklı kullanıcı gruplarının ihtiyaçlarını karşılamak ve çeşitli isteklerine cevap verebilmek için kullanılır
- ❑ Harici şema ile verinin farklı alanları ve görünüşleri (view) sunulur.
- ❑ Kısıtlama gerçekleştirmenizi sağlar



Conceptual or Logical Schema

- ❑ Veritabanında kayıtlı olan verileri veri modelini kullanarak tanımlar ve bütün ilişkileri tanımlar.
- ❑ Varlıklar ve varlıklar arasındaki ilişkiler tanımlanır.
- ❑ Veri bağımsızlığı (data independance) sayesinde kullanıcıların kayıt seviyesindeki değişikliklerden etkilenmelerini minimum seviyeye indirir.



Physical Schema

- ❑ Kavramsal şemada tanımlanan verinin gerçekte disk üzerine nasıl kaydedileceğinin bilgilerini içerir.
- ➔ ❑ Bütün ilişkileri sıralı olmayan dosyalara kaydeder ve her bir şemanın (tablo) ilk kolonunu kullanarak index oluşturur.
- ❑ Fiziksel şemanın tasarımı ile ilgili kararlar ilerde veriye erişimi birinci dereceden etkilemektedir.

Geliştiriciler (Developers)

- ❑ IBM, Oracle, DataStax gibi büyük ölçekli firmalar adına çalışmaktadırlar.
- ❑ Geliştirilen VTYS'leri
 - ❑ çok çeşitli disiplinlerden (eğitim, sanayi, ticaret, havayolu)
 - ❑ farklı **tecrübesi** olan organizasyonlar kullanmaktadır.

Kullanıcılar (Users)

- ❑ Teknik bilgisi olmayan kullanıcılar
 - ❑ veritabanı uygulamalarını programcılarının hazırladığı form web arayüzlerini kullanarak sorgularını veritabanına yönlendirirler.
- ❑ Sofistike kullanıcılar
 - ❑ VTYS'ler kendi sorgularını SQL ile yazmalarına imkan sağlamaktadır

Uygulama Programcıları (Application Developers)

- ❑ Veritabanı sorgulama dilini kullanarak istenilen sorguları içeren uygulamaları geliştirirler.
 - ❑ raporlar, istatistikler, tablolar,
- ❑ Bu uygulamalar harici şema kullanılarak yazılmaktadır.

VTYS Admin (Database Administrator (DBA))

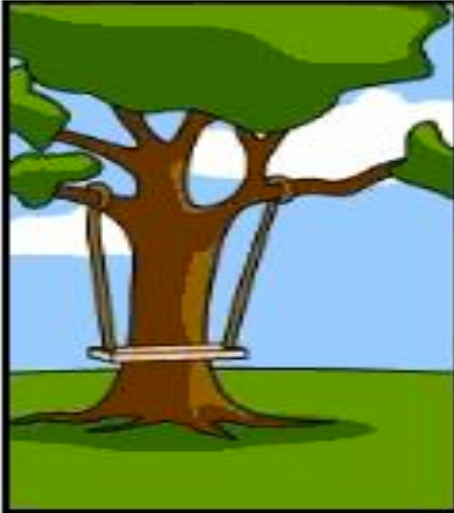
- ❑ Büyük ölçekli firmaların veritabanlarının tasarımı, yönetimi ve mevcut durumunu koruması için profesyonel destek (DBA) gerekmektedir.

Admin Görevler

- ❑ Kavramsal ve fiziksel Şema tasarımı (Conceptual and Physical Schema Design)
- ❑ Güvenlik ve Yetkilendirme (Security and Authorization)
- ❑ Verinin kullanılabilirliği ve hatalardan kurtarılması (Data availability and Recovery from failures)
- ❑ Veritabanının Düzenlemesi (Database Tuning)
 - ❑ Kullanıcı ihtiyaçlarına ve değişim isteklerine cevap verebilme
 - ❑ Donanım ve altyapı değişikliklerini gerçekleştirebilme



How the customer explained it



How the Project Leader understood it



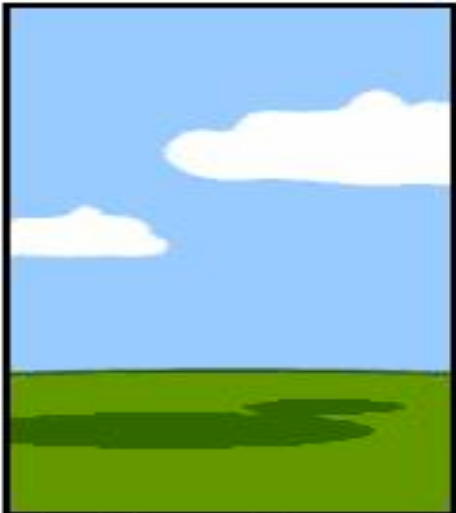
How the Analyst designed it



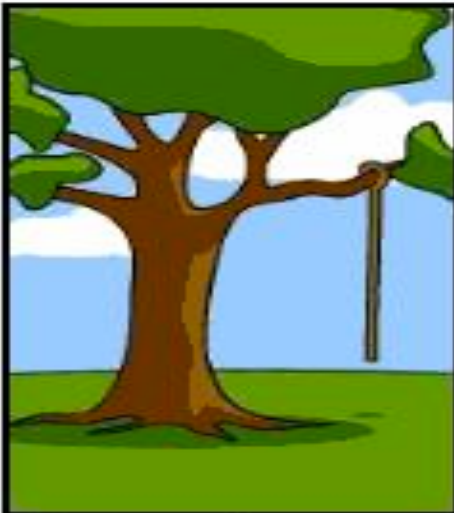
How the Programmer wrote it



How the Business Consultant described it



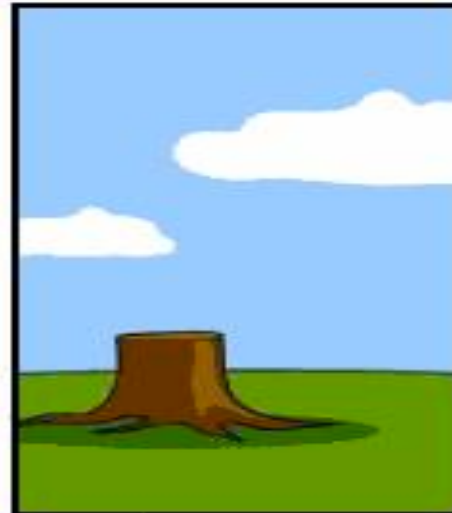
How the project was documented



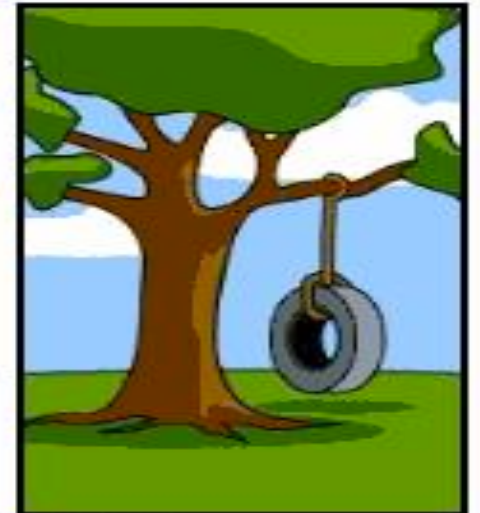
What operations installed



How the customer was billed



How it was supported



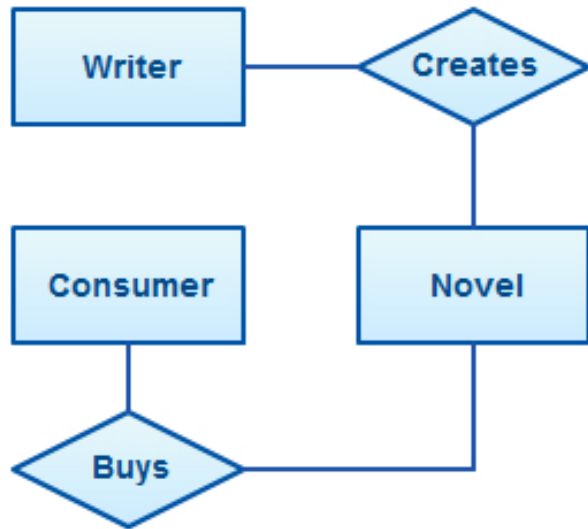
What the customer really needed

1- Gereksinim Analizi (Requirement Analysis)

- ☐ Kullanıcı isteklerinin toplantılarla belirlendiği aşamadır.
- ☐ Kullanıcıların tasarlanacak veritabanından beklentileri belirlenir.
- ☐ Veritabanına kaydedilecek veriler tanımlanır.
- ☐ Veriyi işleyecek uygulamalardan beklenen özellikler belirlenir.
- ☐ Öncelik ve performans gerektiren işlemler belirlenir.
- ☐ Kullanıcı isteklerinin öncelik sırası belirlenerek bir seti elde edilir.

1- Gereksinim Analizi (Requirement Analysis)

2- Kavramsal Veritabanı Tasarımı (Conceptual Database Design)



- ❑ Gereksinim aşamasında belirlenen istekler doğrultusunda veritabanına kaydedilecek verinin varlık-iliski (ER-Entity-relationship) modeli oluşturulur.
- ❑ Veritabanı'na saklanacak verinin kullanıcılar ve geliştiriciler tarafından ortak kullanılabilen ve yüksek-seviyeli görüntüsü oluşturulur.
- ❑ ER Modeli teknik bilgi sahibi olmayan kullanıcıların verinin nasıl kaydedileceğini anlamasına ve tasarım sürecine dahil edilmesine yardımcı olur.
- ❑ Veritabanı tasarımcısı istenilen özellikler doğrultusunda tasarımı gerçekleştirir.

1- Gereksinim Analizi
(*Requirement Analysis*)

2- Kavramsal Veritabanı Tasarımı
(*Conceptual Database Design*)

3- Mantıksal Veritabanı Tasarımı
(*Logical Database Design*)

❑ İkinci aşamada belirlenen varlık-ilişki (ER) modeli ilişkisel şemaya dönüştürülür.

1- Gereksinim Analizi
(*Requirement Analysis*)

2- Kavramsal Veritabanı Tasarımı
(*Conceptual Database Design*)

3- Mantıksal Veritabanı Tasarımı
(*Logical Database Design*)

4- Şema Yenilenmesi
(*Schema Refinement*)

□ 3. aşamada belirlenen ilişkisel şemanın problemleri ortaya çıkarılır ve tekrar düzenlenir.

1- Gereksinim Analizi
(*Requirement Analysis*)

2- Kavramsal Veritabanı Tasarımı
(*Conceptual Database Design*)

3- Mantıksal Veritabanı Tasarımı
(*Logical Database Design*)

4- Şema Yenilenmesi
(*Schema Refinement*)

5- Fiziksel Veritabanı Tasarımı
(*Physical DB Design*)

- ❑ Index tabloları ve yeni Tablolar oluşturulur.
- ❑ Oluşturulan tablolar kullanılarak istenilen performans kriterlerinin kontrolü yapılır.

1- Gereksinim Analizi (Requirement Analysis)

2- Kavramsal Veritabanı Tasarımı (Conceptual Database Design)

3- Mantıksal Veritabanı Tasarımı (Logical Database Design)

4- Şema Yenilenmesi (Schema Refinement)

5- Fiziksel Veritabanı Tasarımı (Physical DB Design)

6- Uygulama ve Güvenlik Tasarımı (Application and Security Design)

- ❑ Uygulama ve güvenlik işlemlerinin tanımlandığı aşamadır.
- ❑ UML diyagramları yardımıyla uygulamaların ve tanımlanan görevlerin işlem aşamaları belirlenir.
- ❑ Her bir varlığın iş akışı şemasında görev ve etki alanı tanımlanır.
- ❑ Veritabanına erişilebilen ve erişimi sınırlı olan kısımları tanımlanır.