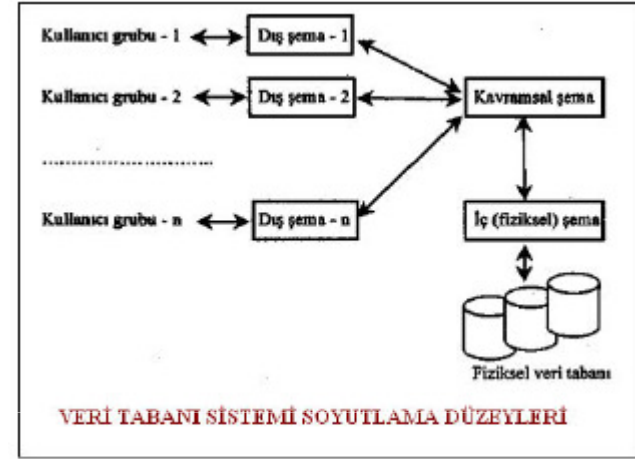


# Soyutlama Düzeyleri

Dr. İsmail İŞERİ  
OMÜ Bilgisayar Mühendisliği

# Soyutlama Düzeyleri

- Fiziksel veri tabanı ile uç kullanıcı arasında birçok soyutlama düzeyi bulunur.
- Yandaki çizimde üç soyutlama düzeyinin yer aldığı örnek bir mimari yer almaktadır.



# İç (fiziksel) Şema

- İç Şema, veri tabanının fiziksel yapısı ile ilgili tanımları içerir.
- Veri tabanı bilgisayarda bir disk dosyası biçiminde yer aldığı için, bu dosyanın disk üzerindeki adresi ve özellikleri ile ilgili tanımlar iç Şemayı oluşturur.
- Aslında fiziksel veri tabanı olarak adlandırılan katman da kendi içinde, dosyalar ve mantıksal tutanaklar düzeyinden başlayıp, bitler ve fiziksel adresler düzeyine kadar uzanan birçok soyutlama düzeyinde görülebilir.

# Kavramsal Şema

- Tüm veri tabanının kuruluş düzeyindeki mantıksal yapısıdır.
- Her Veri Tabanı Yönetim Sistemi, veri tabanının kavramsal şemasını tanımlamak ve bu şemanın fiziksel gerçekleştirimi olan fiziksel şemanın kimi özelliklerini belirtmek için gerekli veri tanımlama olanaklarını sağlar.
- Bunlardan biri olan Veri Tanımlama Dili (DDL), kavramsal veri tabanının, veri modeli terimleri ile tanımlanmasını sağlayan yüksek düzeyli bir dildir.

# Dış (alt) Şema

- Kavramsal Şemanın bir alt kesiminin soyut bir modelidir.
- Dış şemaları tanımlamak için de genellikle veri tanımlama dili kullanılır.
- Dış şema, bir anlamda, küçük bir veri tabanının kavramsal semasıdır ve genellikle kavramsal şema ile dış şema aynı soyutlama düzeyindedir.

# Dış (alt) Şema

- Bazı yönleriyle dış şema kavramsal şemadan daha soyut olabilir.
- Fiziksel veri tabanında ve kavramsal şemada yer almayan, ancak kavramsal şemadaki verilerden türetilen verilere dış şemada yer verilebilir.
  - Örneğin, kavramsal şemada kişilerin doğum tarihleri yer alırken, dış şemada kişilerin yaşlarına yer verilebilir.

# Veri Bağımsızlığı

- Veri bağımsızlığı sayesinde uygulamaların, veri saklama yapıları ve erişim yöntemlerinden bağımsızlaştırılması sağlanır.
- Dış şema, kavramsal şema ve iç şemadan oluşan soyutlama zinciri, iki farklı veri bağımsızlığı düzeyi sağlar.
  - Fiziksel Veri Bağımsızlığı
  - Mantıksal Veri Bağımsızlığı

# Fiziksel Veri Bağımsızlığı

- Fiziksel Veri Bağımsızlığı, “bellekte saklı verilerin yapı ve erişim yöntemi değişikliklerinden uygulamaların etkilenmemesi” olarak tanımlanabilir.
- Eğer bir bilişim sisteminde fiziksel veri bağımsızlığı sağlanmışsa, bellek yapıları ve erişim yöntemleri, hatta bellek ortamları uygulamalardan etkilenmeden değiştirilebilir.

# Fiziksel Veri Bağımsızlığı

- Fiziksel veri bağımsızlığının önemini daha iyi anlamak için, örnek olarak önce veri tabanı olanakları kullanılmadan, genel amaçlı bir programlama dili ile geliştirilen bir bilişim sistemini düşünelim.
- Bu örnekte veriler, kullanılan işletim sistemi ile programlama dilinin sağladığı dosya yapıları ve erişim yöntemleri kullanılarak ikincil belleklerde saklanır.
- Saklanan verilerin hacminin büyüklüğü ve veriler arası bağıntıların çokluğu, verileri saklamak ve verilere erişmek için kullanılan yapı ve yöntemlerin karmaşıklaşmasına yol açar.

# Fiziksel Veri Bağımsızlığı

- Veriler üzerinde işlem yapacak kullanıcıların, karmaşık veri düzenleme ve erişim tekniklerini tüm ayrıntıları ile bilmeleri, yazacakları uygulama programlarında yapıları doğru olarak tanımlamaları ve bu yapılarla uyumlu işlem algoritmalarını oluşturmaları gereklidir.
- Bu nitelikteki uygulama programlarının yalnız uzman bilişim teknik personeli tarafından yazılabileceği açıktır.

# Fiziksel Veri Bağımsızlığı

- Bilişim uygulamaları genellikle sürekli değişim halindedir. Bir kez geliştirildikten sonra birkaç yıl değiştirilmeden kullanılan uygulamaların sayısı son derece azdır.
- Değişiklikler işlevsel ya da işletimsel gereksinimlerden kaynaklanabileceği gibi, teknolojik gelişmelerden veya iyileştirme isteklerinden de kaynaklanabilir.
- Uygulamaların işlevsel ya da işletimsel gereksinimlerinden kaynaklanan değişiklikler kullanıcıların tümünü değil, yalnız belirli bir kesimini doğrudan ilgilendirir. Diğer değişiklikler ise kullanıcıların hiçbirini doğrudan ilgilendirmez.

# Fiziksel Veri Bağımsızlığı

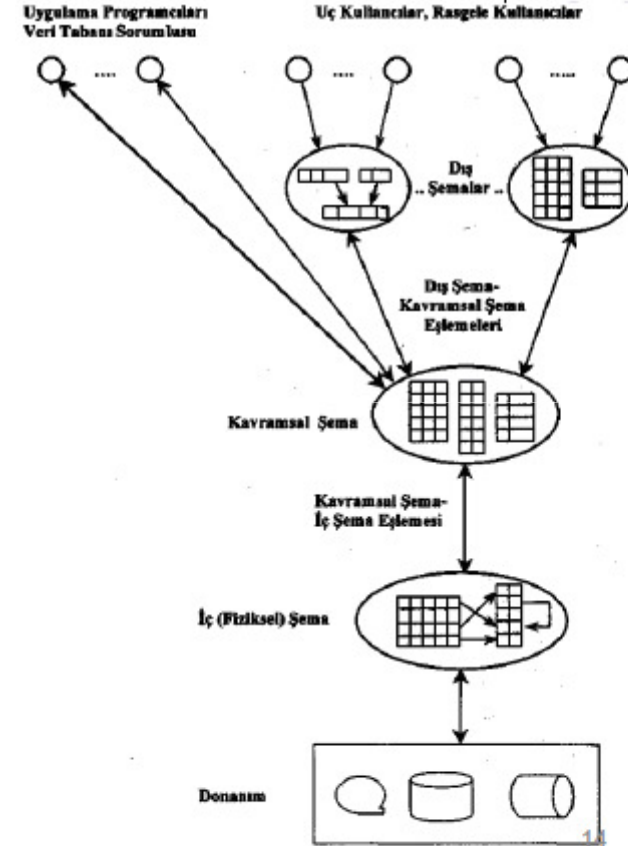
- Veri tabanında yapılan her değişiklikten sonra;
  - tüm uygulama programlarının uyarlanması,
  - yeniden derlenmesi,
  - gerekiyorsa yeniden sınanması,
  - tüm kullanıcıların da yapılan değişikliklerle ilgili bilgi sahibi olmaları, gerekir.
- Yukarıdaki örnekte olduğu gibi “uygulamaların veriyi bellekte saklamak için kullanılan yapılara ve veriye erişmek için kullanılan yöntemlere bağımlı olduğu bilişim uygulamalarına” veriye bağımlı uygulamalar denir.

# Mantıksal Veri Bağımsızlığı

- Mantıksal veri bağımsızlığı, kavramsal şema değişikliklerinden kullanıcıların olabildiğince korunması anlamına gelmektedir. Özellikle, yapılan değişiklik ile ilgisi olmayan dış şemaları kullanan kullanıcıların, bu değişiklikten etkilenmemesi amaçlanmaktadır.
- Kavramsal şema değişiklikleri; kullanıcı gereksinimlerini karşılamak için mantıksal düzeyde yapılan değişiklikler olduğu için, genellikle kavramsal şema değişikliği ile birlikte dış şemalardan bir kısmında da değişiklik yapılır.

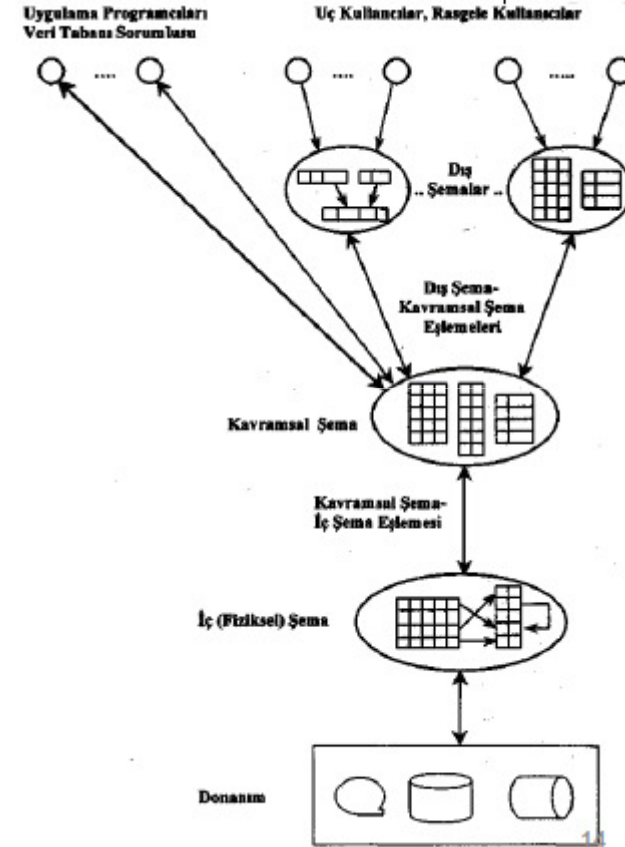
# Örnek bir VTYS mimarisi

- Bir veri tabanı kullanıcısı için, veri tabanı dış şemadır.
- Kullanıcı dış şemayı görür ve sağlanan yazılım olanakları ile dış şemada yetkili olduğu işlemleri gerçekleştirir.
- Kullanıcıların çoğu kavramsal ve iç şemadan habersizdir, verilerin dış şemaya uygun biçimde saklandığını düşünür.
- Oysa dış ve kavramsal şemalar tümüyle mantıksaldır.



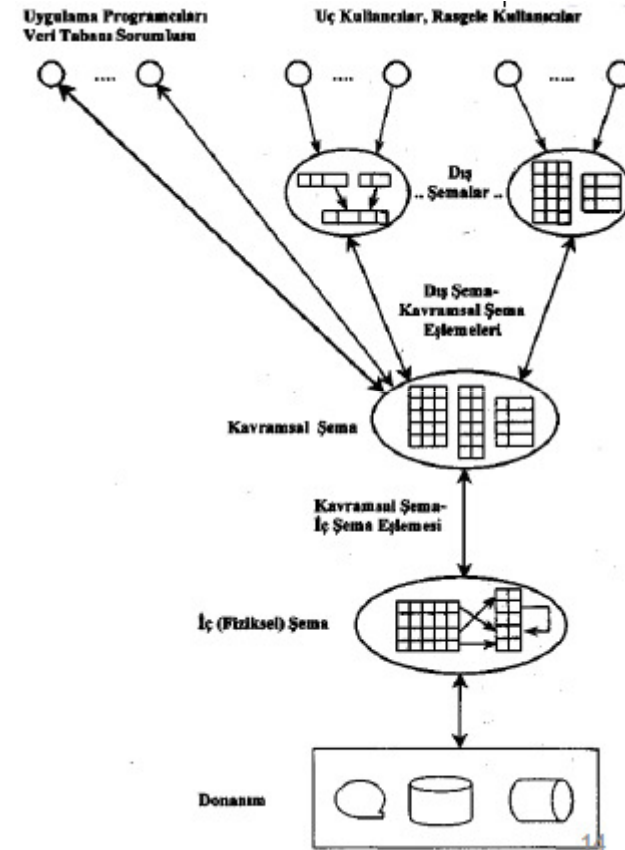
# Örnek bir VTYS mimarisi

- Kullanıcı tarafından dış şemaya göre oluşturulacak isteklerin iç şemada karşılanması gerekir.
- Bunun için de kullanıcı tarafından dış şemaya göre tanımlanan verilerin önce kavramsal şemadaki, sonra da iç şemadaki karşılıklarının belirlenmesi ve kullanıcı isteğinin fiziksel veri tabanı üzerinde gerçekleştirilmesi gerekir.



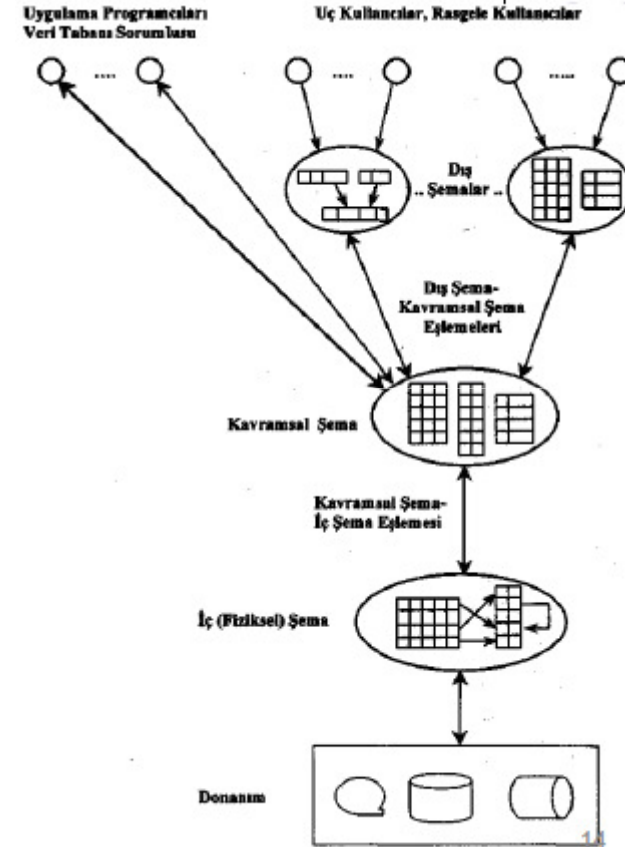
# Örnek bir VTYS mimarisi

- İstek bir sorgu ise, iç şemaya göre seçilen veriler, bu defa önce kavramsal sonra da dış şemaya göre dönüştürülerek kullanıcıya sunulmalıdır.
- Bir VTYS'nin mantıksal ve fiziksel veri bağımsızlığının sağlanabilmesi için, şema tanımlarına ek olarak şemalar arası eşleme tanımlarının da saklanması gerekir.



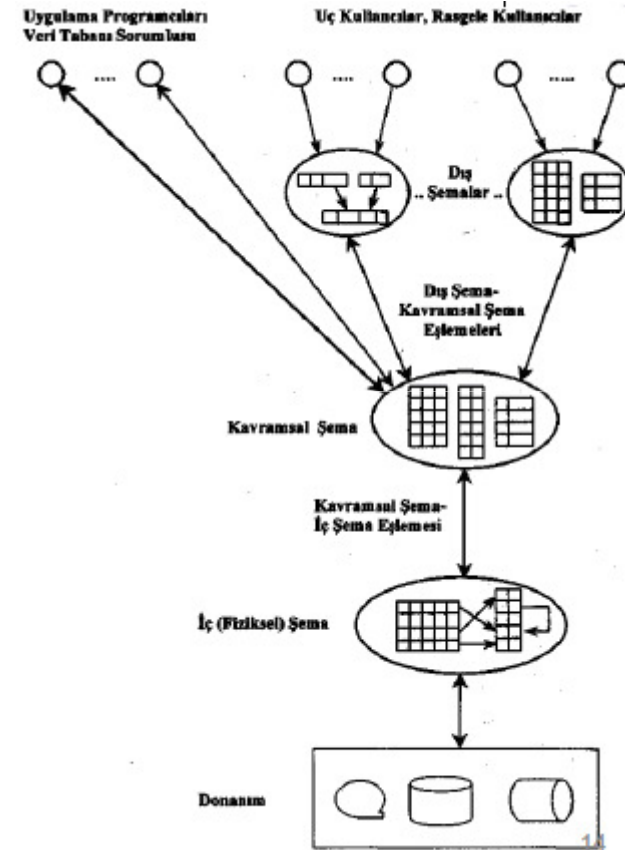
# Örnek bir VTYS mimarisi

- İç şemada bir değişiklik yapıldığında, iç şema - kavramsal şema eşlemede gerekli uyarlamalar yapılarak değişikliğin kavramsal ve dış şemaları, dolayısıyla da kullanıcıları etkilemesi önlenerek fiziksel veri bağımsızlığı sağlanmış olur.



# Örnek bir VTYS mimarisi

- Kavramsal Şemada bir değişiklik yapıldığında ise, bir yandan kavramsal Şema - iç Şema eşlemeleri, diğer taraftan da bazı dış Şemalar ile kavramsal Şema arasındaki eşlemeler uyarlanır.
- Böylece kavramsal Şemadaki değişiklikten iç Şema (fiziksel veri bağımsızlığı) ve dış Şemaların en azından bir kesimi (mantıksal veri bağımsızlığı) etkilenmemiş olur.



# Örnek bir VTYS mimarisi

- Örneğimizdeki üç düzeyli mimari ilk defa 1975 yılında “Bilgisayarlar ve Bilgi İşlem konusundaki Amerikan Ulusal Standartlar Komitesinin VTYS Çalışma Takımı” tarafından önerilmiş ve zamanla benimsenmiştir.
- Veri bağımsızlığının sağlanması için VTYS mimarisinin mutlaka üç düzeyli mimaride olması gerekli değildir.

