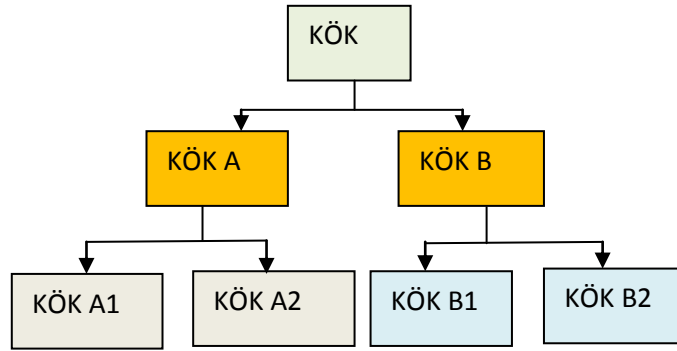


## Hiyerarşik Veri Tabanları

Hiyerarşik veri tabanı modeli bir ağaç yapısına sahiptir. Model dahilindeki herhangi bir düğüm, altındaki n sayıda düğüme bağlanırken, kendisinin üstünde ancak bir düğüme bağlanabilir. Hiyerarşik yapının en tepesindeki düğüm noktasına kök denir ve bu düğümün sadece bağımlı düğümleri bulunur. Bu veri yapısını gösteren grafiğe de hiyerarşik tanım ağacı denir. Sonuç olarak bu veri tabanı modeli **1:n** ilişkisini desteklemektedir.

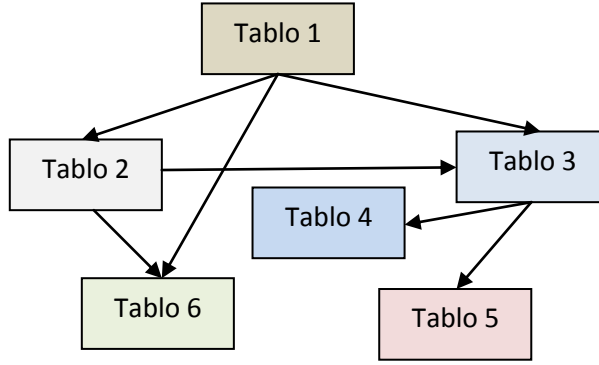
Bu veritabanı tipi, ana bilgisayar ortamlarında çalışan yazılımlar tarafından kullanılmaktadır. Bu türde en çok kullanılan yazılım, IBM tarafından çıkarılan IMS dir. Uzun bir geçmişe sahip olmasına rağmen, PC ortamına uyarlanan hiyerarşik veri tabanları yoktur.



Şekil x: Hiyerarşik Veri Tabanı Modeli

## Ağ Veri Tabanları

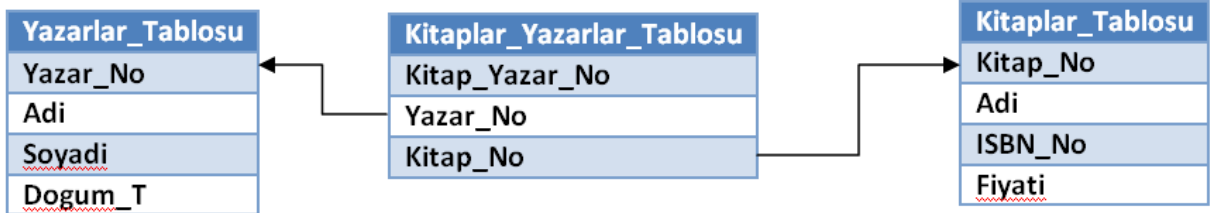
Hiyerarşik veri tabanlarının yetersiz kalmasından dolayı bilim adamlarının ortak çalışması sonucu ortaya konulmuş bir veri tabanı türüdür. Ağ Veri-Tabanı Modeli esasında hiyerarşik veri tabanı modelinin geliştirilmiş bir sürümüdür. Ağ veri tabanı modelinde şebeke içinde bir eleman, herhangi bir başka elemana bağlanabilir. Nitekim ortaya tablolar arasında kurulan bir ağ çıkmaktadır. Buradaki ilişki türü **n:n** dir. Ağ veri tabanları, verileri ağaçların daha da gelişmiş hali olan graflar (ağacın kendisinde özel bir graftır.) şeklinde saklarlar. Bu yapı en karışık yapılardan biridir. Bu veri tabanı modeli hiyerarşik veri tabanı modelinden çok daha esnek bir yapıya sahiptir.



Şekil x: Ağ Veri Tabanı Modeli

## İlişkisel Veri Tabanları

İlişkisel veri tabanı modeli, hiyerarşik modeldeki kısıtlamaya neden olan maddeleri, yani hiyerarşik yapıyı tamamen terk eden bir veri tabanı modelidir. E.F. Codd tarafından 1970 'li yıllarda geliştirilmiştir. Bu sistemde veriler tablo diye adlandırılan yapılarda tutulur. Tablolar arasında ilişkiler belirlenir. Bu ilişkiler matematiksel bağıntılarla (ilişkilerle) temsil edilir. Günümüzde en çok kullanılan veri tabanı modeli olmakla beraber, en başarılı veri tabanı modelidir.



Şekil x: İlişkisel Veri Tabanı Modeli

## Nesneye Yönelik Veri Tabanları

Nesneye yönelik veri tabanı da, C++ gibi nesneye dayalı bir dille (OOPL) yazılmış olan ve yine C++ gibi nesneye dayalı (OOPL) bir dille kullanılan veri tabanı anlamına geliyor. Nesne veri-tabanı modeli, verilerin herhangi bir noktadan çok kolayca alınabileceği, üç boyutlu bir yapıdan oluşur. İlişkisel veri tabanı verileri iki boyutlu tablolar halinde getirirken, nesne modelinde veriler tek parça olarak gelirler. Dolayısı ile birden fazla veri dönmesi arzulandığında nesne modeli performans olarak çok iyi değildir.

Günümüz teknolojisinde yüzde yüz nesneye yönelik bir veri tabanı yaygın olarak kullanıma sunulmuş değildir. Ancak nesneye yönelik veri tabanlarının bazı üstünlükleri olacağından söz ediliyor.

İlişkisel veri tabanları ile araştırıldığında nesneye yönelik veri tabanlarının sahip olması gereken üstünlükler şunlardır:

1. Nesneler, bir tabloda yer alan bir kayıttan çok daha karmaşık yapıya sahiplerdir ve daha esnek bir yapıda çok daha kullanışlı düzenlenebiliyorlar.
2. Nesneye dayalı bir veri tabanında, yapısı gereği arama işlemleri çok hızlı yapılabilir. Özellikle büyük tablolarla uğraşırken ilişkisel veri tabanlarından çok daha hızlı sonuca ulaşırlar. Ancak çalışma mantığı tümüyle değişir
3. Bu veri tabanı modelinde, türlerin kullanılmasına gerek olmamasıdır.
4. Çok karmaşık bir yapıya sahip olan büyük veri tabanı tasarımını kolaylaştırmasıdır. Bunu, nesne yöntembilimin prensiplerine uygun olarak tasarlanmış bir model olmasından kaynaklanır.

## **Nesne-İlişkisel Veritabanı Modeli (Object-Relational Database Model)**

---

Nesne-İlişkisel Veri tabanı modeli küresel bir yapıya sahiptir. Veri tabanı üzerindeki herhangi bir veriye, yüksek performansta erişim sağlar. Fakat yine de birden fazla veri istenildiği zaman bu modelde de veri tabanı performansı çok kötü bir darbe alır. Bu veri tabanı modeli, ilişkisel ve nesne veri tabanı modellerini bir şekilde aynı çatı altına almak için oluşturulmuştur.