**İlişkisel ve İlişkisel Olmayan (NoSQL) Veri Tabanı Sistemleri**

Kullanıcılar tarafından sürekli daha kısa sürede erişilmek istenen bilgi; bir amaca ulaşmak için veri veya ham bilginin işlenerek kullanıcılara fayda sağlayacak biçimde dönüştürülmüş halidir.

Temel bir kurumdan, orta ve büyük ölçekli işletmelerin kurumsal ve ticari bilgilerinin düzenlenerek saklanmasına kadar farklı alanlarda veri depolama gerekliliği ortaya çıkmıştır. Verinin büyüklüğü, miktarı ve karmaşıklığı gibi farklı etkenlere bağlı olarak veri modelleme, veri depolama ve sorgulama yöntemleri ortaya çıkmıştır.

Okuma ve yazma gibi işlemlerin yoğun olarak kullanıldığı taban, ilişkisel veri tabanıdır. Bunun yanı sıra performans ve esneklik özellikleri ile ilişkisel olmayan veri tabanı yöntemleri sistemleri (NoSQL) ise ‘eBay’ ve ‘Amazon’ gibi şirketlerde tercih edilmiştir.

**BİLİŞİM SİSTEMLERİ VE YÖNETİMİ:**

Bilişim sistemi; organizasyonlarda karar verme aşamasına kadar bilgiyi toplar, düzenler, işler ve saklar. Bilişim sistemlerinde 3 aşama bilgiyi üretmek için gereklidir:

1-Girdi: Veriyi toplar.

2-İşlem: Toplanan veriyi daha anlamlı hale getirir.

3-Çıktı: İşlenen veriyi, kullanıcılara aktarır.

İşletmeler için bilişim sistemleri, girdi olarak alınan verinin işlenip çıktısını almaktan daha fazlasını ifade eder.

**VERİ TABANI VE VERİ TABANI YÖNETİM SİSTEMİ:**

Veri Tabanı; birbirleriyle ilişkileri olan verilerin, kullanımına uygun düzenlenmiş, mantıksal ve fiziksel olarak tanımlarının olduğu bilgi depolarıdır. Veri Tabanı Yönetim Sistemleri, verilere aynı anda birden fazla bağlantı sağlayabilme özelliği sunar. Bu sistemler; verinin nasıl depolanacağını, kullanılacağını ve ulaşabileceğini gösteren sistemdir.

Veri Tabanı Sistemi; “Veri tabanı”, ”Veri tabanı yönetim sistemi” ve “Uygulama programları/Kullanıcı arayüzü” nü bir arada içerir.

Veri Tabanı modelleri 8’e ayrılır:

•**Tablo Model (Düz Model**): İki boyutlu veri grubundan oluşarak; sütunlarda verilerin benzer özellikleri, satırlarda ise veri grupları yer alır.

•**Hiyerarşik Veri Modeli:** Bu veri tabanının depoladığı yapısal verilere “kayıt” adı verilir. Kayıtlar ağaç şeklinde yukarıdan aşağıya sıralanır. “Kök” adı verilen ilk kaydın bir veya birden fazla alt kaydı vardır. Kayıtlar dallanabilir.

•**Ağ Veri Modeli:** Hiyerarşik veri modelinin geliştirilmiş halidir. Ağ modelinde bir veri diğer verilerle ilişkili olup, uç düğümdeki veri iç düğümdeki veriye bağlıdır.

•**İlişkisel Veri Modeli:** Satır ve sütundan oluşan iki boyutlu tablolardan elde edilir. Tablonun her satırında birbiriyle ilişkili veriler bulunurken, sütunlarda nitelikler bulunur.

•**Nesne Yönelimli Veri Modeli:** Nesne yönelimli programlamaya dayanan veri modelidir.

•**Nesne İlişkisel Veri Modeli:** Nesne yönelimli özellikler içerir. Nesne ilişkisel ilk veri modeli Oracles8’dir.

•**Çoklu Ortam Veri Modeli:** Bu model nesne ilişkisel veri tabanı ile büyük benzerlik gösterir. Model; veri miktarını, sürekliliği ve senkronizasyonu destekler. Özellikle tıp bilgi sistemlerinde kullanılır.

•**Dağıtık Veri Modeli:** Birden fazla bilgisayarlarda depolanarak, bir ağ üzerinde dağıtılan bilgiler için kullanılır. Veri tabanını parçalara ayırmak, sorguların daha hızlı işlenmesini sağlar.

**VERİ TABANI TASARIMI**

Gerçeğin, ihtiyaçlar çerçevesinde işlenerek veri tabanına aktarılması gerekir. Veri tabanı tasarımında ilk olarak kullanıcıların ihtiyaçlarının belirlenmesi gerekir. Bilgilerin,veri tabanı sisteminde gerek kullanıcılar gerekse bilgisayarlar tarafından anlaşılabilecek şekilde tanımlanması gerekir.

Kullanıcı ve bilgisayar düzeyleri sırasıyla “kavramsal” ve “fiziksel” düzeyler olarak bilinir. Çeşitli veri modelleri geliştirilmiştir. Kavramsal şema, ortalama veri tabanı kullanıcısı için, veri tabanının yapısını genel olarak tanımlar. Kavramsal şema, yazılım ve donanımdan bağımsız olup son kullanıcı tarafından anlaşılması daha kolaydır.

Fiziksel tasarım aşamasında, veri tabanında fiziksel olarak nasıl organize edilmesi gerektiği belirlenir.

**İLİŞKİSEL VE İLİŞKİSEL OLMAYAN (NoSQL) VERİ TABANI SİSTEMLERİ**

İlişkisel Veri Tabanı , günümüzde yaygın kullanılan veri tabanı sistemlerinden biridir. Satır ve sütunların meydana getirdiği tablolardan oluşur. Bir veri tabanında ilişkiden söz edebilmek için en az iki tablonun yer alması ve bu iki tablodaki verilerin birbiri ile ilişkili olması gerekir.

**-**İlişkisel veri tabanı sistemlerinde sağlanan temel özellikler:

• Bölünmezlik

• Tutarlılık

• İzolasyon

• Dayanıklılık

İlişkisel Olmayan (NoSQL) Veri tabanı, Carlo Strozzi sayesinde ortaya çıkmıştır. İlişkisel olmayan veri tabanları,yatay olarak ölçeklendirilen bir veri depolama sistemidir. Ölçek sorununa, en iyi cevap veren NoSQL’dir.

Çok büyük verilerin depolanması ve yazılmasında ilişkisel veri tabanlarının eksik kaldığı zamanlarda, NoSQL çözüm sağlamıştır.

NoSQL veri tabanının kullanılma nedenleri:

• Düşük maliyet

• Yüksek performans

• Verilerin ölçeklenebilirliği

• Esneklik

**VERİ TABANI MİMARİLERİNİN PERFORMANS KARŞILAŞTIRMASI**

İlişkisel veri tabanı olan MySQL ve ilişkisel olmayan (NoSQL) veri tabanı olarak, ilişkisel veri tabanı sistemlerine alternatif çözüm olan MongoDB veri tabanı sistemi kullanılmıştır.

MySQL ve MongoDB veri tabanı sistemlerinin performans ve yatay ölçeklenebilirlik incelemesi için aşağıdaki işlemlerin uygulanması hedeflenmiştir:

• Veri tabanı sunucu sistemleri özellikleri belirlenmesi,

• Veri tabanı şemaları oluşturulması,

• Sorguların belirlenmesi,

• Veri tabanı ayarlarının yapılması,

• Ölçümler ve ölçüm metrikleri bilgileri,

• Performans analizi ve sonuçlarıdır.

**Veri Tabanı Sorguları:**

• “SELECT” deyimi içeren basit bir sorgu hazırlanmıştır.

•İkinci sorgu için daha karmaşık “INNER JOIN” olan bir sorgu hazırlanmıştır.

•Üçüncü sorgu “INNER JOIN” ve “WHERE” olan detaylı karmaşık bir sorgu hazırlanmıştır

**Projede zaman ölçümleri için 3 yöntem kullanılır:**

•Birinci Yöntem: Clock() fonksiyonu ile belirli bir süre CPU üzerinde harcanan zamanın sonuçlarını gösterir.

•İkinci Yöntem: Milisaniye zamanlamalarını hesaplayan Gettimeofday() fonksiyonu kullanılır.

•Üçüncü Yöntem (Slow Query Log): Veri tabanı için belirlenmiş uzun süren sorguları kaydeder.

**‘Saniyedeki Sorgu=(Toplam Sorgu Sayısı\*Toplam İş Parçacığı Sayısı)/Ortalama Sorgu Süresi**‘ şeklinde bulunurken,

**‘İş Parçacığı Başına Saniyedeki Sorgular=Toplam Sorgu Sayısı/Ortalama Sorgu Süresi**’ şeklinde hesaplanır.

••MySQL ve MongoDB veri tabanları sistemlerine eşit miktarda sorgular yapılır.

İlk olarak bu iki veri tabanına basit sorgu testi uygulanmıştır. Test sonucunda 3 işlemci sayısı-1 işlemci çekirdeği sayısına göre incelendiğinde MySQL veri tabanının, sorgu sayıları arttığında MongoDB sistemine göre daha avantajlı olduğu ortaya çıkar. Fakat 2 işlemci sayısı-3 işlemci çekirdeği sayısına göre incelendiğinde ise MongoDB daha avantajlı görülmüştür.

İkinci sorguda; MongoDB veri tabanı sisteminin, daha kısa sürede daha çok sorgu yürütmesinin mümkün olduğu, sorgu sayısı değiştikçe daha iyi performans sergilediği gözlemlenmiştir.

Üçüncü sorguda; MySQL veri tabanının, MongoDB’ye göre veri kayıt sayısı farkı arttıkça daha iyi sonuç verdiği ortaya çıkmıştır. Ortalama sorgu süreleri sonuçlarına göre, veri kayıt sayısı farkı arttıkça MongoDB’nin MySQL veri tabanına göre daha iyi performans gösterdiği görülmüştür.

MongoDB daha belirgin ve karalı performans gösterdiği görülür.

••Son olarak iki veri tabanlarına **veri ekleme (INSERT)** ve **silme(DELETE)** işlemleri uygulanmıştır. Veri ekleme işleminde MongoDB, MYSQL veri tabanına göre daha iyi performans gösterir. Fakat veri silme işleminde komut sayılarının artışı ile MySQL veri tabanı silme işleminde daha iyi performans sergilediği gözlenmiştir.