# İLİŞKİSEL VE İLİŞKİSEL OLMAYAN(NoSQL) VERİ TABANI YÖNETİM SİSTEMLERİ MİMARİ PERFORMANSININ YÖNETİM BİLİŞİM SİSTEMLERİ TARAFINDAN İNCELENMESİ

## 1.GİRİŞ:

Bilgisayar ve iletişim teknolojilerinde yaşanan hızlı gelişim her geçen gün farklı çözümler üretmeye zorlamaktadır. Belli bir amaca ulaşmak için bilginin ham hali organizasyonlar tarafından yararlı haline çevrilir bilginin en etkili ve hızlı biçimi ortaya çıkar. Verinin özelliklerine bağlı olarak veri modelleme, depolama ve sorgulama gibi yöntemler geliştirilmiştir. Bu kapsamda okuma ve yazma gibi işlemlerin olduğu veri tabanlarında ilişkisel veri tabanlarının yanı sıra ilişkisel olmayan veri tabanı yönetim sistemleri de kullanılmaktadır.

Bu çalışmada bilişim sistemleri ve veri tabanı imcelenerek iki veri tabanı yönetim sistemi performansı incelenecktir.

## 2.BİLİŞİM SİSTEMLERİ VE YÖNETİMİ:

Bilişim sistemi, organizasyonlarda karar verme aşamasına kadar bilgiyi toplamak,düzenlemek olarak tanımlanabilir. Bilişim sistemlerinde bilgiyi üretmek için gerekli aktiviteler:girdi,işlem ve çıktıdır. Girdi, organizasyonun içinden veya dışından veriyi toplamaktır. İşlem, veriyi daha anlamlı hale getirir. Çıktı, işlenmiş bilgiyi, insanlara aktarır. Bilişim sistemleri, bilişim teknolojileri alt yapısından yararlanan çözümlerdir.

## 2.VERİ TABANI VE VERİ TABANI YÖNETİM SİSTEMLERİ(VTYS):

Veri tabanı genel tanımı ile kullanıma uygun hale getirilmiş veriler topluluğudur. Birbirleriyle ilişkileri olan verilerin tutulduğu alanlardır. Gerçekte var olan birbirileriyle ilişkisi olan nesneleri modeller.

VTYS, verilere aynı anda birden çok bağlantı sağlayabilme özelliği sağlar. Verinin nasıl depolanacağı kullanılacağı gibi mantıksal olarak yönlendiren kurallar sistemidir.

Veri tabanı, VTYS ve uygulama programlarını ile kullanıcı ara yüzlerini içeren yapıya “veri tabanı sistemi (VTS)” denir.

Veri tabanı modellerini sekiz kategoriye ayırabiliriz:

* Düz model veya tablo modeli : İki boyutlu veri grubundan oluşur. Sütunlarda verilerin benzer özellikleri, satırlarda ise veri grupları yer alır. Kullanıcı adlarının ve şifrelerinin tutulduğu veri tabanı buna örnek olarak verilebilir. Böyle bir veri tabanında her satırda bir kullanıcıya ait şifre bilgileri, sütunlarda ise tipleri aynı olan veriler yer alır. Düz veri modeli tek tablodan oluşan bir model olarak düşünülebilir.
* Hiyerarşik Veri Modeli: Bu veri tabanının depoladığı yapısal verilere “kayıt” adı verildi. Kayıtlar ağaç mimarisi şeklinde yukarıdan aşağı sıralanmaktadır. Kök adı verilen ilk kaydın bir veya daha çok çocuk kayıtları vardır. Çocuk kayıtlarında kendi çocuk kayıtları olabilir. Kök haricinde bütün kayıtların bir ebeveyni vardır.
* Ağ veri modeli: Ağ modelinin hiyerarşik modelden en önemli farkı, uç- düğüm pozisyonundaki verinin iç-düğüme işaret edebilmesidir. Böylelikle ağ modelinde bire-çok ilişkiler yanında, çoka-çok ilişkiler de modellenebilir. Bu veri tekrarını önemli ölçüde azaltır.
* İlişkisel Veri Modeli: İlişkiler yardımıyla, veri içerisindeki ilişkiler modellenir. Dolayısıyla, ilişkisel bir veri tabanı, çeşitli ilişki örneklerinden oluşur. Kavramsal olarak ilişkiler, satır ve sütunlardan oluşan iki boyutlu tablolarla karakterize edilir. Genellikle veri tabanında her tablo için bir dosya bulunur. Tablonun her satırı birbiriyle ilişkili verilerin bir topluluğudur. Sütunlarda ise nitelikler bulunur .
* Nesne İlişkisel Veri Modeli: Nesne ilişkisel veri tabanı, ilişkisel işlevselliğin üzerine nesne yönelimli özellikler içerir. İlişkisel veri tabanları içinde nesne yönelimli karakteristikler içeren ilk veri tabanı 1997 yılında piyasaya sunulan Oracle8’dir.
* Çoklu Ortam Veri Modeli: Çoklu ortam veri tabanları nesne ilişkisel veri tabanları ile büyük benzerlikler gösterir. Bununla birlikte, film, müzik, metin ve video gibi büyük nesneleri işlemek ve aynı zamanda işleme sırasındaki adımları kullanıcıya göstermemek için farklı özellikler taşır. Çoklu ortam veri tabanlarının desteklemesi gereken üç temel özellik; Veri miktarı, Süreklilik ve Senkronizasyondur. Çoklu ortam veri tabanı uygulaması, imge görüntüleme, uzaktan görüntülü eğitim, üç boyutlu tıbbi görüntü kayıtları depolanması konularında özellikle tıp bilgi sistemlerinde kullanılmaktadır
* Dağıtık Veri Modeli: Dağıtık veri tabanları, iki ya da daha fazla bilgisayarda depolanan ve bir ağ üzerinde dağıtılan bilgiler için kullanılan veri tabanı grubudur. Veri tabanını ağ üzerinden paralel kullanmak için parçalara ayırmak, sorguların daha hızlı işlenmesini sağlar. Böyle bir sistemde, birden fazla veri tabanına erişilmesine rağmen, kullanıcı bir tek veri tabanıyla çalışıyormuş gibi işlem yapar .

## 4. VERİ TABANI TASARIMI

Veri tabanı tasarımında; gerçeğin, modellenerek veri tabanına aktarılması gerekir. Veri tabanı tasarımında ilk adım, olası kullanıcı gereksinimlerinin belirlenmesidir. Bu gereksinimler, veri tabanında bulunacak veri gruplarını, veri tiplerini ve depolama yapılarını tanımlar. Gerçeğin veri tabanındaki sayısal temsili, bir model olarak tanımlanmalı ve kullanıcılar ile bilgisayarlar tarafından anlaşılabilir bir şekilde ifade edilmelidir. Bu tanımlama sürecine "şema" adı verilir.

Şema, kullanıcı ve bilgisayar düzeyleri için sırasıyla "kavramsal" ve "fiziksel" düzeylerde bulunur. Kavramsal şema, kullanıcıların anlayabileceği bir modeli temsil ederken, iç şema ise bilgisayarların anlayabileceği fiziksel bir modeli temsil eder. Her iki düzeyde de farklı anlayış mekanizmalarına hitap edildiğinden, kullanılacak veri modelleri de farklılık gösterebilir.

Bu süreçte, çeşitli veri modelleri kullanılarak kavramsal ve fiziksel düzeylerde şemalar geliştirilir. Bu modeller, veri tabanı tasarımında kullanılan temel yapı taşlarıdır.

Veri tabanı tasarımı, kullanıcı ihtiyaçlarına göre üç ana aşamada gerçekleşir. İlk aşama, kavramsal tasarımdır. Bu aşamada, genel veri tabanı yapısı belirlenir ve kavramsal şema oluşturularak, kullanıcıların veri tabanını anlaması ve uygulamalarını modellemesi sağlanır. İkinci aşama, mantıksal tasarımdır. Burada, uygun bir veri tabanı yönetim sistemi seçilir ve kavramsal model, seçilen sisteme uyarlanarak kullanıcı ihtiyaçları ile veri tabanı sistemi arasında bir köprü oluşturulur. Son aşama ise fiziksel tasarımdır. Bu aşamada, verinin fiziksel olarak nasıl depolanacağı ve organize edileceği belirlenir. Veri tabanı, gerçek dünya uygulamalarına uygun hale getirilerek, fiziksel tasarım tamamlanır.

## 5. İLİŞKİSEL VE İLİŞKİSEL OLMAYAN (NoSQL) VERĠİ TABANI SİSTEMLERİ

## 5.1 İlişkisel Veri Tabanı (Relational Database System)

Günümüzde en yaygın kullanılan veri tabanı sistemleri, satır ve sütunlardan oluşan tablolardan meydana gelir. Bu tablolar arasında ilişkiler bulunur ve ilişkisel veri tabanları, bu ilişkileri kullanarak büyük dosyalardan oluşur. Her bir tablo, belirli bir yapıya uygun verileri saklamak üzere tasarlanmıştır.

ACID, klasik ilişkisel veri tabanı sistemlerinde sağlanan temel özellikleri tanımlar:

* Bölünmezlik (Atomicity)
* Tutarlılık (Consistency)
* İzolasyon (Isolation)
* Dayanıklılık (Durability)

Bu özellikler, veri tabanı işlemlerinin güvenilirliğini sağlar ve veri bütünlüğünü korur.

## 5.2 İlişkisel Olmayan (NoSQL) Veri tabanı (Non-Relational Database System)

NoSQL veri tabanı sistemlerine bir alternatif olarak ortaya çıkmıştır. Veri tabanı sorunlarından biri olan ölçek sorununa en iyi çözüm ilişkisel olmayan veri tabanı olarak belirlenmiştir.

* Kolay Ulaşılabilirlik (Basically Available): Veri erişim sorunlarını önlemek için kopyalar kullanır ve veriyi birçok sunucudan alır.
* Esnek Durum (Soft state): NoSQL sistemler, ACID mantığındaki sıkı veri tutarlılığına odaklanmaz ve tutarsız ve geçici verilere izin verir.
* Eninde sonunda Tutarlı (Eventually consistent): NoSQL sistemler, uygulamalar anlık tutarlılıkla ilgilense de gelecekte belirli bir zaman diliminde verilerin tutarlı olacağını varsayar. Bu, ACID'in zorunlu kıldığı anlık tutarlılıktan farklı bir yaklaşımdır.

Formun Üstü

## VERİ TABANI MİMARİLERİNİN PERFORMANS KARŞILAŞTIRMASI (PERFORMANCE COMPARISON OF DATABASE ARCHITECTURE)

Bu çalışmada, günümüzde yaygın olarak kullanılan MySQL ve MongoDB veri tabanı sistemleri performans ve yatay ölçeklenebilirlik açısından incelenmiştir. Aşağıdaki işlemler yapılmış ve sonuçlar ortaya çıkarılmıştır:

1. Veri tabanı sunucu sistemlerinin özellikleri belirlenmiştir.
2. MySQL ve MongoDB için veri tabanı şemaları oluşturulmuştur.
3. İlgili sorgular belirlenmiştir.
4. Veri tabanı ayarları yapılmıştır.
5. Ölçümler ve ölçüm metrikleri toplanmıştır.
6. Performans analizi yapılmış ve sonuçlar elde edilmiştir.

Veri Tabanı Şeması: Proje kapsamında, MySQL ve MongoDB için iki farklı veri tabanı şeması tasarlanmıştır. MySQL şeması, MongoDB şeması gibi bir müzik uygulaması etrafında modellenmiştir ve farklı algoritmalar kullanarak diğer kullanıcılara şarkı önerileri sunmak için tasarlanmıştır. Veri tabanı şemaları, veri tekrarını ortadan kaldırmak için normalizasyon prensiplerine uygun olarak oluşturulmuştur.

Bu çalışmada üç farklı veri tabanı sorgusu kullanılmıştır:

1. Birinci sorgu, sadece "SELECT" deyimini içeren basit bir sorgudur.
2. İkinci sorgu, daha karmaşık bir yapı olan "INNER JOIN" deyimini içeren bir sorgudur.
3. Üçüncü sorgu ise "SELECT" deyimi ile birlikte iç içe geçmiş "JOIN", "INNER JOIN" ve "WHERE" deyimlerini içeren detaylı ve karmaşık bir sorgudur

Projede ölçümler için zaman kavramı ön planda tutulmuştur ve üç farklı yöntem kullanılmıştır:

1. Birinci yöntem, CPU üzerinde harcanan zamanı ölçmek için Clock() fonksiyonunu kullanmaktır.
2. İkinci yöntem, milisaniye hassasiyetiyle zamanlamaları sağlamak için Gettimeofday() fonksiyonunu kullanmaktır.
3. Üçüncü yöntem ise Slow Query Log (Yavaş sorgu kaydı) olarak adlandırılmaktadır. Her veri tabanı, kendi zaman ölçüm yöntemini sunar. Bu yöntemde, önceden belirlenmiş uzun süren sorguları kaydedebilir ve mikrosaniye doğruluğunda yapılandırılabilir.

Ölçüm Metrikleri: Veri tabanlarının performansını ölçmek için ortak bir metrik gereklidir. Bir işlemi tamamlamak için gereken süre ve veri tabanının işlemi tamamlama durumu için gereken zaman gibi faktörler önemlidir. Bu kavramlar iyi anlaşılmalı ve birbirinden ayrı tutulmalıdır.

## 7. SONUÇ VE DEĞERLENDĠRME (RESULT AND EVALUATION)

Bu çalışmada, yönetim bilişim sistemleri çerçevesinde veri tabanlarının modellenmesi, niteliklerinin belirlenmesi, en uygun performans ölçümleri, sürecin uygun hale getirilmesi ve en uygun veri tabanının seçimi konuları ele alınmıştır. Teknolojik ilerlemelerin etkisiyle veri tabanı sistemlerinde NoSQL kavramının önemi artmıştır. Bu makalede, ilişkisel ve ilişkisel olmayan veri tabanı yönetim sistemlerinin performanslarının karşılaştırılması ve her bir veri tabanını etkileyen faktörlerin incelenmesi amaçlanmıştır.

Literatür taraması sonucunda benzer çalışmalara rastlanmıştır. Örneğin, Ahmet ALADILY tarafından yapılan 2015 yılındaki tezde Couchdb, Mongodb, Cassandra ve Hbase gibi dört farklı dağıtık veri tabanının performansı karşılaştırılmıştır. Yılmaz GÖKŞEN ve Hakan AŞAN tarafından yapılan başka bir çalışmada ise NoSQL veri tabanları özelinde esneklik, tutarlılık, performans, sorgulama dili ve veri bağlantıları gibi özellikler açısından ilişkisel veri tabanları ile genel bir karşılaştırma yapılmıştır.

Bu çalışmada ise yapılan uygulama ile tutarlılık, esneklik gibi özelliklerden ziyade performans üzerinde durulmuş ve farklı senaryolar incelenmiştir. Analizler sonucunda, NoSQL ağırlıklı bir veri tabanının büyük veri çiftleri içerebilme ve basit şeması sayesinde karmaşık sorguları daha hızlı çalıştırabilme yeteneği olduğu görülmüştür. Özellikle MongoDB, büyük bir avantaj sağlayarak MySQL'e göre daha iyi performans göstermiştir.

Sonuçlar, işletmelere farklı senaryolarda hangi veri tabanı yönetim sisteminin daha uygun olduğunu belirlemede yardımcı olacak niteliktedir. İlişkisel olmayan (NoSQL) veri tabanlarının hız, geliştirme zamanı ve ölçeklenebilirlik gibi özellikler açısından, ilişkisel veri tabanı yönetim sistemlerine göre daha etkin sonuçlar sunduğu sonucuna varılmıştır.