

# İlişkisel ve İlişkisel Olmayan (NoSQL) Veritabanı Sistemlerinin Mimari Performanslarının Bilgi Sistemleri Yönetimi Açısından İncelenmesi

---

AD SOYAD:ESİN DÖVER  
NUMARA:02210224072  
DERS:VERİ ORGANİZASYONU

# GİRİŞ

- Bilgisayar ve iletişim teknolojilerindeki hızlı gelişimin organizasyonları etkileyerek farklı çözümler üretmeye zorluyor. Verinin işlenerek bilgiye dönüştürülmesi, organizasyonlar için giderek daha önemli hale gelmiştir. Bilgi sistemleri, bu süreçte önemli bir rol oynamaktadır.
- Günümüzdeki değişim ve gelişim, veri modellenmesi ve depolanmasının önemini artırmaktadır. Veri modelleme ve depolama gereklilikleri, kurum rehberinden büyük işletmelerin bilgilerinin organize edilmesine kadar farklı alanlarda ortaya çıkmaktadır. Bu ihtiyaçlar doğrultusunda, ilişkisel ve ilişkisel olmayan veri tabanı yönetim sistemleri kullanılmaktadır. İlişkisel olmayan veri tabanı yönetim sistemleri, performans ve esneklik özellikleri nedeniyle tercih edilebilir hale gelmiştir.
- Bu çalışmada, bilişim sistemleri ve veri tabanı kavramları incelenerek ilişkisel ve ilişkisel olmayan veri tabanı yönetim sistemlerinin mimari performansları detaylı bir şekilde karşılaştırılmıştır.



## 2. BÖLÜM BİLİŞİM SİSTEMLERİ VE YÖNETİMİ (INFORMATION SYSTEMS AND MANAGEMENT)

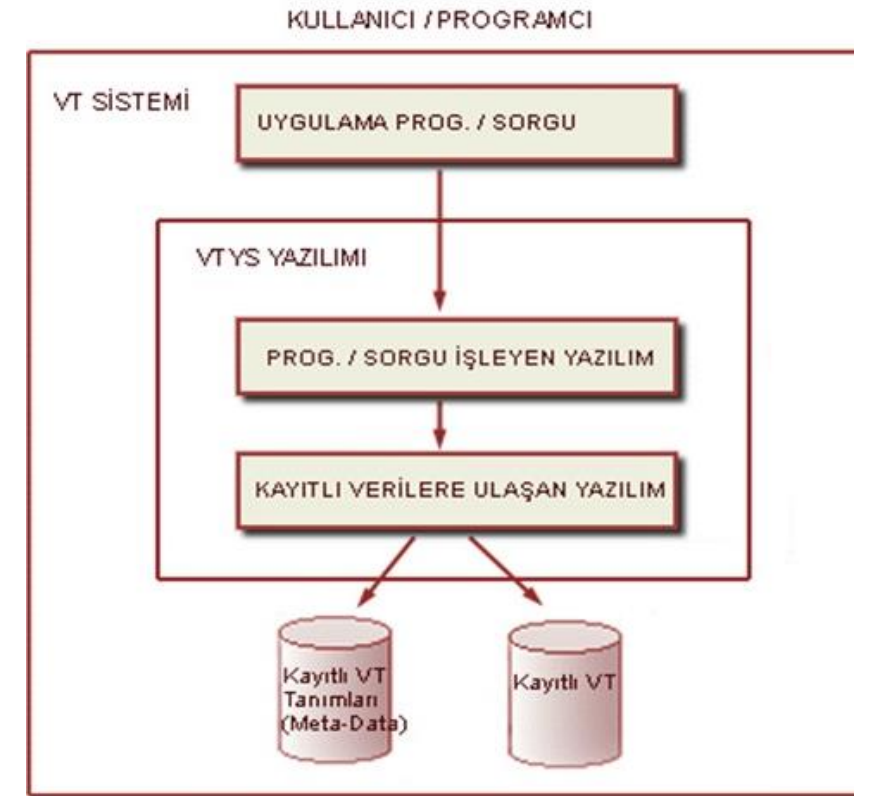


Bilişim sistemi, organizasyonlarda karar verme aşamasına kadar bilgiyi toplamak, düzenlemek, işlemek ve saklamak olarak tanımlanabilir. Bilişim sistemlerinde üç aktivite bilgiyi üretmek için gereklidir. Bu aktiviteler: girdi, işlem ve çıktıdır. Girdi ham bilgileri toplamaktır. İşlem bilginin daha anlamlı hale getirilmesidir, çıktı ise işlenen bilginin kullanıcılara , insanlara aktarılmasıdır.

Bilişim sistemleri, bilişim teknolojileri altyapısından yararlanan yönetsel çözümlerdir. Bilişim sistemlerini etkin bir şekilde kullanmak için organizasyon, yönetim ve teknolojiye hâkim olmak gerekmektedir [

### 3. VERİ TABANI VE VERİ TABANI YÖNETİM SİSTEMLERİ(DATABASE AND DATABASE MANAGEMENT SYSTEM)

- Veri tabanının genel tanımı kullanım amacına uygun olarak düzenlenmiş veri topluluğu ve bilgi deposu olmasıdır. Veri tabanları ilişkileri modeller. Veri tabanı yönetim sistemleri verilere aynı anda birden fazla bağlantı sağlayabilir. Bu sistemler verinin nasıl depolanacağını nasıl kullanılacağını ve erişileceğini gösteren kurallar sistemidir.
- Veri tabanı, VTYS ve uygulama programlarını ile kullanıcı ara yüzlerini içeren yapıya “veri tabanı sistemi (VTS)” denir. Veri tabanı, veri tabanı yönetim sistemi ve veri tabanı sistemi arasındaki ilişki ve işlevler gösteren grafik yandadır.

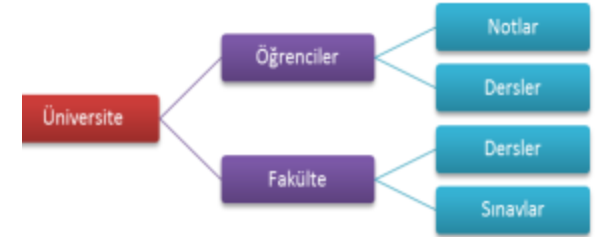


# Veri tabanı modellerini sekiz kategoriye ayırabiliriz:

- **Düz model veya tablo modeli**
- İki boyutlu veri grubundan oluşur. Bu modelde, veriler sütunlarda benzer özelliklere sahip olarak düzenlenirken, satırlarda ise veri grupları bulunur. Örneğin, kullanıcı adlarının ve şifrelerinin tutulduğu bir veri tabanı bu modele örnek olarak verilebilir. Bu tür bir veri tabanında, her satır bir kullanıcıya ait şifre bilgilerini içerirken, sütunlar ise aynı tipte verileri barındırır. Düz veri modeli, tek bir tablodan oluşan bir model olarak düşünülebilir.
- **Hiyerarşik Veri Modeli**
- Veriyi depolama yönteminden adını almıştır. Bu modelde, depolanan yapısal verilere "kayıt" adı verilir. Kayıtlar, ağaç mimarisi şeklinde yukarıdan aşağıya sıralanır. İlk kayıta "kök" adı verilir ve bir veya daha fazla çocuk kaydı bulunabilir. Her çocuk kaydı kendi altında başka çocuk kayıtları içerebilir. Kök haricindeki tüm kayıtların bir ebeveyni vardır.

	Ad Soyad	Kullanıcı Adı	Parola
Kayıt 1	Murat ERGİN	Mergin	kjVdb125
Kayıt 2	Ayşe YILMAZ	Ayılmaz	Bks46db7
Kayıt 3	Can TÜRK	Cturk	fhG8dbt9

Şekil 3.2 Düz veri modeli örneği  
(Instance of flat data model)



Şekil 3.3 Hiyerarşik Veri Tabanı Modeli  
(Hierarchical Database Model)

- **Ağ Veri Modeli**
- Hiyerarşik veri modelinin geliştirilmiş halidir. Hızla kabul görmesinin sebebi verilerin doğal olarak birbirleriyle ilişkili olmasıdır. Bu modelde, verinin iç düğüme işaret etme özelliği bulunur. Bu sayede bir veri birden fazla veriyle etkileşime geçebilir bu da veri tekrarını önemli ölçüde azaltır.
- **İlişkisel Veri Modeli**
- Hiyerarşik ve ağ veri modellerinin çeşitlenen beklentileri karşılamakta yetersiz kalması, ilişkisel veri modelinin geliştirilmesi için bir itici güç olmuştur. E. F. Codd'un 1970'te yazdığı "A Relational Model of Data for Large Shared Data Banks" makalesi, ilişkisel veri yapılarında önemli bir ilerleme kaydedilmesini sağlamıştır. İlişkisel veri modelinin temel kavramı, ilişkidir. İlişkiler sayesinde, veri içindeki ilişkiler modellenir ve ilişkisel veri tabanı, çeşitli ilişki örneklerinden oluşur. İlişkisel veri tabanında, kavramsal olarak ilişkiler, satır ve sütunlardan oluşan iki boyutlu tablolarla temsil edilir. Her tablo genellikle bir dosyaya karşılık gelir ve tablonun her satırı, birbiriyle ilişkili verilerin bir koleksiyonunu temsil ederken sütunlarda ise nitelikler bulunur.
- **Nesne Yönelimli Veri Modeli**
- Daha sonraları ortaya çıkmış ve başarısını kanıtlamıştır. Nesne yönelimli programlamaya dayanan veri modelidir.
- **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

- Nesne ilişkisel veri tabanlarıyla benzerlik gösterir ancak büyük nesneleri işlemek ve kullanıcıya işleme adımlarını göstermemek için farklı özelliklere sahiptir. Bu model, film, müzik, metin, video gibi büyük nesneleri işlerken veri miktarı, süreklilik ve senkronizasyon gibi temel özellikleri desteklemelidir. Özellikle tıp bilgi sistemlerinde, görüntüleme, uzaktan eğitim ve üç boyutlu tıbbi görüntü kayıtlarının depolanması gibi alanlarda kullanılır.

- Dağıtık Veri Modeli

- Birden fazla bilgisayar üzerinde depolanan ve ağ üzerinde dağıtılan veriler için kullanılan bir veri tabanı grubunu ifade eder. Bu modelde, veri tabanı parçalara ayrılarak ağ üzerinde paralel kullanım sağlanır, böylece sorgular daha hızlı işlenir. Kullanıcı, birden fazla veri tabanına erişse de, tek bir veri tabanı ile çalışıyormuş gibi işlem yapar.



## 4. VERİ TABANI TASARIMI (DATABASE DESIGN)

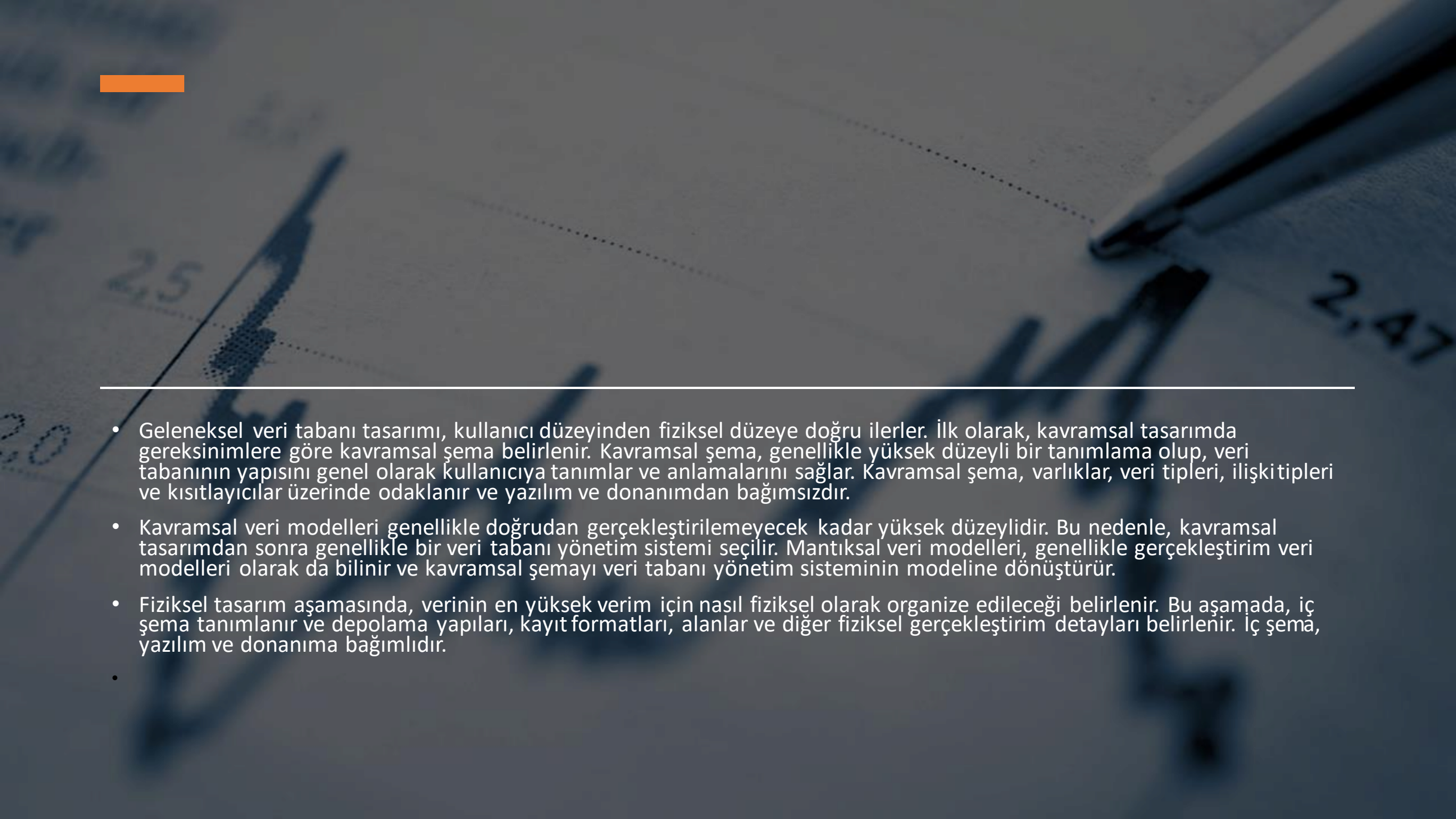



Veritabanı tasarım safhaları

- Veri tabanı tasarımında, gerçeğin gereksinimler ve beklentiler doğrultusunda modellemesi yapıp veri tabanına aktarılmalıdır. İlk olarak, olası kullanıcı gereksinimleri belirlenir. Bu gereksinimler, veri gruplarını, veri tiplerini ve depolama yapılarını tanımlar. Veri tabanındaki sayısal temsili, kullanıcılar ve bilgisayarlar tarafından anlaşılabilir bir şekilde tanımlanır ve genellikle "şema" olarak adlandırılır. Kullanıcı ve bilgisayar düzeylerindeki şemalar farklı anlayış mekanizmalarına hitap eder, bu nedenle kullanılacak veri modelleri de farklılık gösterebilir.

-



- 
- 
- 
- Geleneksel veri tabanı tasarımı, kullanıcı düzeyinden fiziksel düzeye doğru ilerler. İlk olarak, kavramsal tasarımda gereksinimlere göre kavramsal şema belirlenir. Kavramsal şema, genellikle yüksek düzeyli bir tanımlama olup, veri tabanının yapısını genel olarak kullanıcıya tanımlar ve anlamalarını sağlar. Kavramsal şema, varlıklar, veri tipleri, ilişki tipleri ve kısıtlayıcılar üzerinde odaklanır ve yazılım ve donanımdan bağımsızdır.
  - Kavramsal veri modelleri genellikle doğrudan gerçekleştirilemeyecek kadar yüksek düzeylidir. Bu nedenle, kavramsal tasarımdan sonra genellikle bir veri tabanı yönetim sistemi seçilir. Mantıksal veri modelleri, genellikle gerçekleştirim veri modelleri olarak da bilinir ve kavramsal şemayı veri tabanı yönetim sisteminin modeline dönüştürür.
  - Fiziksel tasarım aşamasında, verinin en yüksek verim için nasıl fiziksel olarak organize edileceği belirlenir. Bu aşamada, iç şema tanımlanır ve depolama yapıları, kayıt formatları, alanlar ve diğer fiziksel gerçekleştirim detayları belirlenir. İç şema, yazılım ve donanıma bağımlıdır.
  -

# 5. İLİŞKİSEL VE İLİŞKİSEL OLMAYAN (NoSQL) VERİ TABANI SİSTEMLERİ (RELATIONAL AND NONRELATIONAL DATABASE (NoSQL) SYSTEMS)

- ✓ Günümüzde en yaygın kullanılan veri tabanı sistemlerinden biri, ilişkisel veri tabanlarıdır. Bu sistemler, satır ve sütunlardan oluşan tablolardan oluşur ve bu tablolar arasında ilişkiler bulunur. İlişkisel veri tabanlarında en az iki tablonun bulunması ve bu tablolar arasındaki verilerin ilişkilendirilmesi gerekmektedir. ACID özellikleri, klasik ilişkisel veri tabanı sistemlerinde temel olarak sağlanır.
- ✓ ☐ Bölünmezlik (Atomicity) ☐ Tutarlılık (Consistency) ☐ İzolasyon (Isolation) ☐ Dayanıklılık (Durability)
- ✓ İlişkisel olmayan (NoSQL) veri tabanları ise, ilişkisel veri tabanı sistemlerine alternatif bir çözüm olarak geliştirilmiştir. Bu sistemler, yatay olarak ölçeklendirilebilen bir veri depolama sistemidir. NoSQL, özellikle büyük veri depolama ve işleme gereksinimlerinde tercih edilir.
- ✓ NoSQL'un popülerliği, sosyal medya platformları gibi büyük veri hacimleriyle başa çıkabilme kabiliyetinden kaynaklanmaktadır. İlişkisel veri tabanlarının ölçeklenebilirlik ve performans açısından yetersiz kaldığı durumlarda, NoSQL çözümleri daha etkili olabilir.
- ✓ İlişkisel veri tabanı kullanıcılarının NoSQL'a geçme isteklerinin nedenleri araştırmalarla incelenmiş ve bu geçiş eğilimleri yüzde olarak gösterilmiştir.

Tablo 1: Lider NoSQL ürünlerinin teknik karşılaştırması  
(The Technical comparison about the leader NoSQL's products)

	MongoDB	CouchDB	Riak	Redis	Voltdb	Cassandra	HBase
Dil	C++	Erlang	Erlang	C++	Java	Java	Java
Lisans	AGPL	Apache	Apache	BSD	Apache	Apache	Apache
Model	Document	Document	Key/Value	Key/Value	Key/Value	Wide Column	Wide Column
Protokol	BSON	HTTP/REST	HTTP/REST or TCP/Protocol	bufs TCP		TCP/Thrift	HTTP/REST or
Depolama	Memory mapped b-trees	COW-BTree	Pluggable: InnoDB, LevelDB, Bitcask	In memory, snapshot to disk	Pluggable: BSV, MySQL, in-	Memtable/SSTable	HDFS

- Amazon, "DynamoDB" ve Google, "BigTable" adını verdikleri NoSQL veri tabanı sistemleriyle bu gereksinimi karşılamaktadır. İlişkisel veri tabanlarına tercih edilen NoSQL'un seçilme nedenleri, özellikle hız ve yatay büyümeyle ek maliyetten kurtulma isteğine dayanmaktadır.
- NoSQL, ilişkisel veri tabanlarının ACID işlemselliğine karşılık "BASE" (Basically Available- Soft state-Eventually consistent) prensibiyle işler. Bu prensip, aşağıdaki şekilde ifade edilir:
- Basically Available (Kolay Ulaşılabilirlik): Veri erişim sorunlarını kopyaları kullanarak ve birçok sunucudan veri alarak çözer.
- Soft state (Esnek Durum): NoSQL sistemler tutarsız ve süreksiz verilerin barınmasına izin verir.
- Eventually consistent (Eninde sonunda Tutarlı): NoSQL, gelecekte bir zaman diliminde tutarlı olacağını garanti eder.
- NoSQL veri tabanları, özellikle e-ticaret, internet arama motorları ve sosyal medya gibi büyük ölçekli internet uygulamaları için güvenilirliklerini kanıtlamıştır. NoSQL ürünleri, her biri kendine özgü karakteristiklere sahip olan birçok kayıt saklama teknolojisini içerir.

# 6. VERİTABANI MİMARİLERİNİN PERFORMANS KARŞILAŞTIRILMASI (PERFORMANCE COMPARISON OF DATABASE ARCHITECTURE)

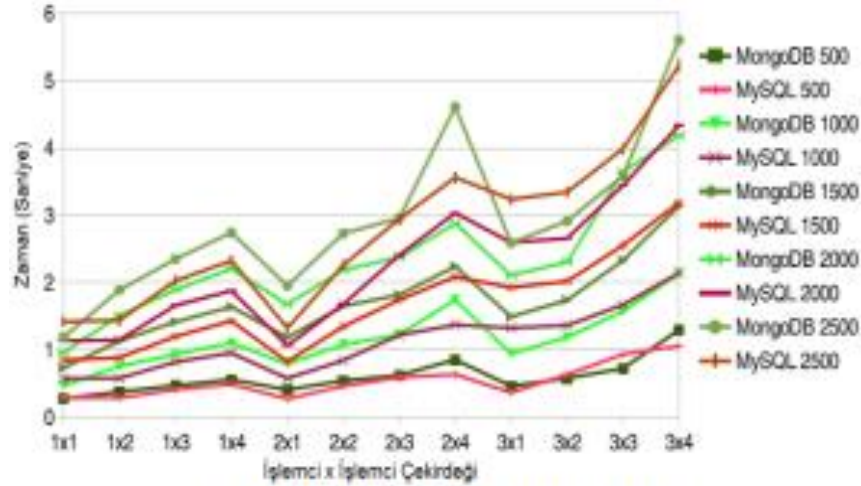
Bu çalışmada, günümüzde en yaygın kullanılan MySQL ve ilişkisel olmayan MongoDB veri tabanı sistemleri performans ve yatay ölçeklenebilirlik açısından incelenmiştir. İnceleme için aşağıdaki adımlar izlenmiştir:

1. Veri tabanı sunucu sistemlerinin özelliklerinin belirlenmesi.
2. Veri tabanı şemalarının oluşturulması.
3. Sorguların belirlenmesi.
4. Veri tabanı ayarlarının yapılması.
5. Ölçümler ve ölçüm metriklerinin toplanması.
6. Performans analizi ve sonuçlarının çıkarılması.

Proje kapsamında, MySQL ve MongoDB veri tabanı sistemlerine özgü şemalar tasarlanmıştır. Bu şemalar, farklı algoritmalar kullanarak müzik uygulaması üzerinden diğer kullanıcılara şarkı önermek için modellenmiştir. Normalizasyon değerlendirmesi yapılarak, veri tekrarlarının önlenmesi sağlanmıştır.

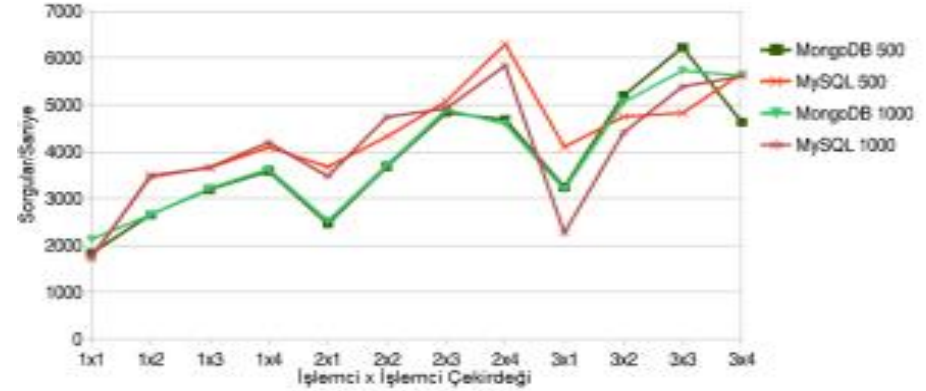
- **Veri Tabanı Sorguları:** Bu çalışmada üç farklı veri tabanı sorgusu kullanılmıştır. Birinci sorgu için sadece “SELECT” deyimi içeren basit bir sorgu hazırlanmıştır. İkinci sorgu için daha karmaşık “INNER JOIN” deyimi içeren bir sorgu hazırlanmıştır. Üçüncü sorgu için ise “SELECT” ile birlikte iç içe “JOIN”, “INNER JOIN” ve “WHERE” deyimi içeren detaylı karmaşık bir sorgu hazırlanmıştır.
- Sorgu 1: Basit
- Sorgu 2: Karmaşık
- Sorgu 3: Detaylı ve karmaşık
- **Analiz ve sonuçlar:** Burada öncelikle veri tabanlarının farklı sorgu türlerine göre nasıl yanıt verdiği hem okuma hem yazma ile analiz edilen sorguların toplam sayısı ve sonuçları şekillerle gösterilmiştir. Son olarak veri tabanı boyutunun performansa etkisi konusunda inceleme yapılmıştır.

- Şekil 6.3'de MySQL ve MongoDB veri tabanlarına sorgu 1 (basit sorgu) ile karşılaştırma testi uygulanmıştır.



Şekil 6.3 Sorgu 1- Analiz işlemi  
(Query 1- Analysis process)

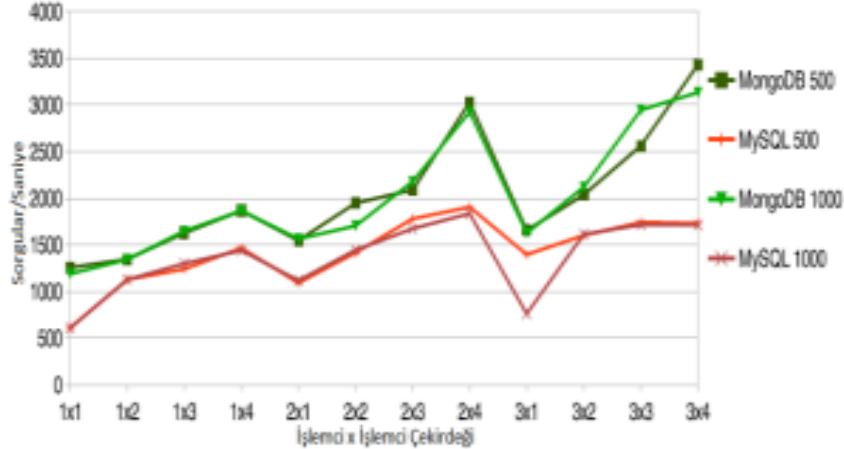
Ayrıca, sorgular/saniye ölçüm metrik grafiği ile de şekil 6.4'de görüldüğü üzere ayrıntılı ortalama süre sonuçları elde edilmiştir.



Şekil 6.4 Sorgu 1 - Sorgu/saniye analiz işlemi  
(Query 1 - Queries/second analysis process)

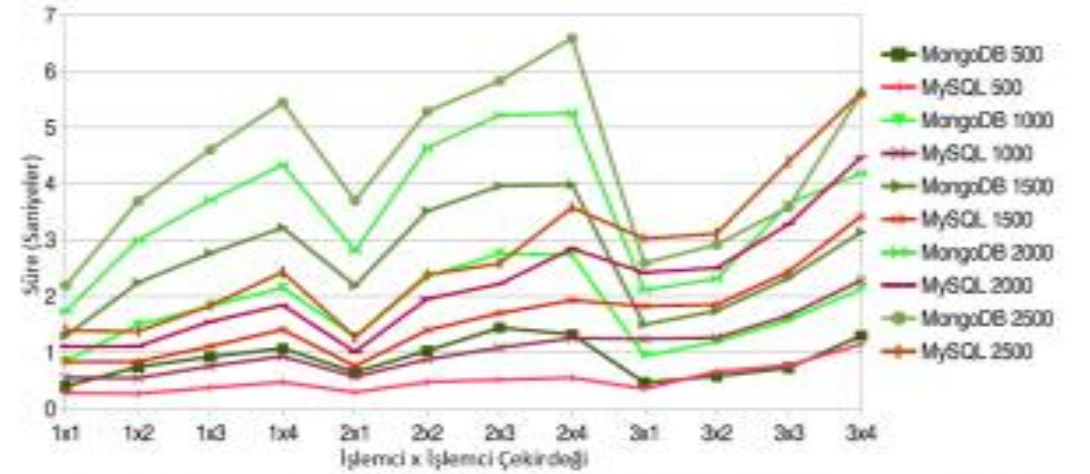
# YAPILAN ANALİZLER

Yapılan analizde; MongoDB veri tabanı sisteminin, daha az bir sürede daha çok sorgu yürütmesinin mümkün olduğu, sorgu sayısı değiştikçe performans ölçümünün daha belirgin hale gelerek sorgu/saniye başına %40 oranında daha iyi performans sergilediği gözlemlenmiştir.



Şekil 6.8 Sorgu 2- INNER JOIN ile 500 ve 1000 veri için sorgu/saniye analizi işlemi

Yapılan analizlerde; MySQL veri tabanı sisteminin MongoDB 'ye göre sorgu süresi sonuçları, veri kayıt sayısı farkı arttıkça iyi bir performans göstermiştir. Fakat işlemci ve işlemci çekirdeği yapılandırmalarının değiştirildiği durumlarda performans farklılıkları daha belirgin hale gelmiştir. Bu karşılaştırma, işlemci ve işlemci çekirdeği sayılarının 3x1, 3x2, 3x3 ve 3x4 şeklinde yapılandırıldığı anlarda iki veri tabanı birbiri üzerine hemen hemen aynı performansı gösterdiği gözlemlenmiştir.



Şekil 6.10 Sorgu 3 – Detaylı karmaşık sorgu süre analizi (Detailed and complex query time analysis)



# 7. SONUÇ VE DEĞERLENDİRME (RESULT AND EVALUATION)

- Bu çalışmanın amacı, yönetim bilişim sistemleri çerçevesinde veri tabanlarının modellenmesi, niteliklerinin belirlenmesi ve uygun performans ölçümlerinin yapılmasıdır. Ayrıca, sürecin uygun hale getirilmesi ve en uygun veri tabanı seçimi konusunda kullanıcılara rehberlik edilmesi amaçlanmıştır. Son yıllardaki teknolojik gelişmelerin etkisiyle, NoSQL kavramı da ön plana çıkmıştır. Bu makalede, ilişkisel ve ilişkisel olmayan veri tabanı yönetim sistemlerinin performans karşılaştırması yapılarak, her bir veri tabanının farklı faktörlerden nasıl etkilendiği ve hangi durumlarda diğerine göre daha uygun olduğu araştırılmıştır.
- Artan yazılım rekabetinde kullanıcının en önemli tercih kriterlerinden biri olan çalışma hızı dikkate alınarak, oldukça yaygın kullanım alanına sahip veri tabanı yönetim sistemlerinden MongoDB ve MySQL'in mümkün olduğunca eşit koşullarda işlem süreleri hesaplanarak performansları karşılaştırılmıştır. Farklı sorgu tipleri çalıştırılan testlerde, detaylı ve karmaşık yapılandırmalar ile veri tabanları analiz edilmiştir.
- NoSQL veri tabanlarının, büyük veri çiftlerini hızlı ve basit bir şekilde işleyebilmesi ve MongoDB'un karmaşık sorguları MySQL'e kıyasla daha iyi bir performansla gerçekleştirebilmesi gözlemlenmiştir. Özellikle, MongoDB'un alt belge koleksiyonu kullanımı, büyük veri tabanı boyutlarına rağmen depolama ve bellek maliyetlerinde önemli bir avantaj sağlamaktadır. Bu nedenle, büyük veri tabanı sistemlerinde NoSQL veri tabanlarının alternatif olarak değerlendirilmesi önemlidir.
- Sonuç olarak incelenen veri tabanları arasında avantaj ve dezavantajlar bulunmaktadır. İlişkisel veri tabanlarının NoSQL sistemlerine taşınması zor olabilir ve veri güvenliği konusunda bazı eksiklikler yaşanabilir. Ancak NoSQL veri tabanları, hız, geliştirme zamanı ve ölçeklenebilirlik gibi özelliklerle performans açısından daha etkin sonuçlar sağlayabilir.