

İlişkisel ve İlişkisel Olmayan (NoSQL) Veri Tabanı

- 1. GİRİŞ
- 2. BİLİŞİM SİSTEMLERİ VE YÖNETİMİ
- 3. VERİ TABANI VE VERİ TABANI YÖNETİM SİSTEMLERİ
- 4. VERİ TABANI TASARIMI
- 5. İLİŞKİSEL VE İLİŞKİSEL OLMAYAN (NoSQL) VERİ TABANI SİSTEMLERİ
- 6. VERİTABANI MİMARİLERİNİN PERFORMANS KARŞILAŞTIRMASI
- 7. SONUÇ VE DEĞERLENDİRME

GİRİŞ

- . Bilgisayar ve iletişim teknolojilerinde yaşanan hızlı gelişim her geçen gün daha fazla organizasyonu etkileyerek farklı çözümler üretmeye zorlamaktadır .
- . Belli başlı bir amaca ulaşmak için veri veya ham bilginin işlenerek ilgililere yarar sağlayacak biçime dönüştürülmüş hali olan bilgi, organizasyonlar tarafından sürekli daha kısa sürede erişilmek istenen en etkili faktör haline gelmiştir .
- . Günümüzde yaşanan bu değişim ve gelişim, verilerin modellenerek saklanması ve dolayısıyla veri tabanı kullanımını zorunlu kılmaktadır.
- . Bu kapsamda, okuma ve yazma gibi işlemlerin yoğun olarak kullanıldığı veri tabanlarında ilişkisel veri tabanlarının yanı sıra ilişkisel olmayan veri tabanı yönetim sistemleri de kullanılmaktadır. Performans ve esneklik özellikleri ile ilişkisel olmayan veri tabanı yönetim sistemleri (NoSQL) Amazon gibi dünyaca ünlü şirketler tarafından tercih edilebilir hale gelmiştir.
- . Bu çalışmada “bilgi sistemleri” ve “veri tabanı” kavramları incelenerek ilişkisel ve ilişkisel olmayan veri tabanı yönetim sistemleri mimari performansının detaylı karşılaştırılması yapılmıştır.

BİLİŞİM SİSTEMLERİ VE YÖNETİMİ

Bilişim sistemlerinde üç aktivite bilgiyi üretmek için gereklidir. Bu aktiviteler: girdi, işlem ve çıktıdır. Girdi, organizasyonun içinden veya dış çevresinden, ham bilgileri (veriyi) toplamaktır. İşlem, bu ham veriyi daha anlamlı biçime çevirir. Çıktı, işlenmiş bilgiyi (enformasyon), insanlara veya kullanılacak olan aktivitelere aktarır.

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

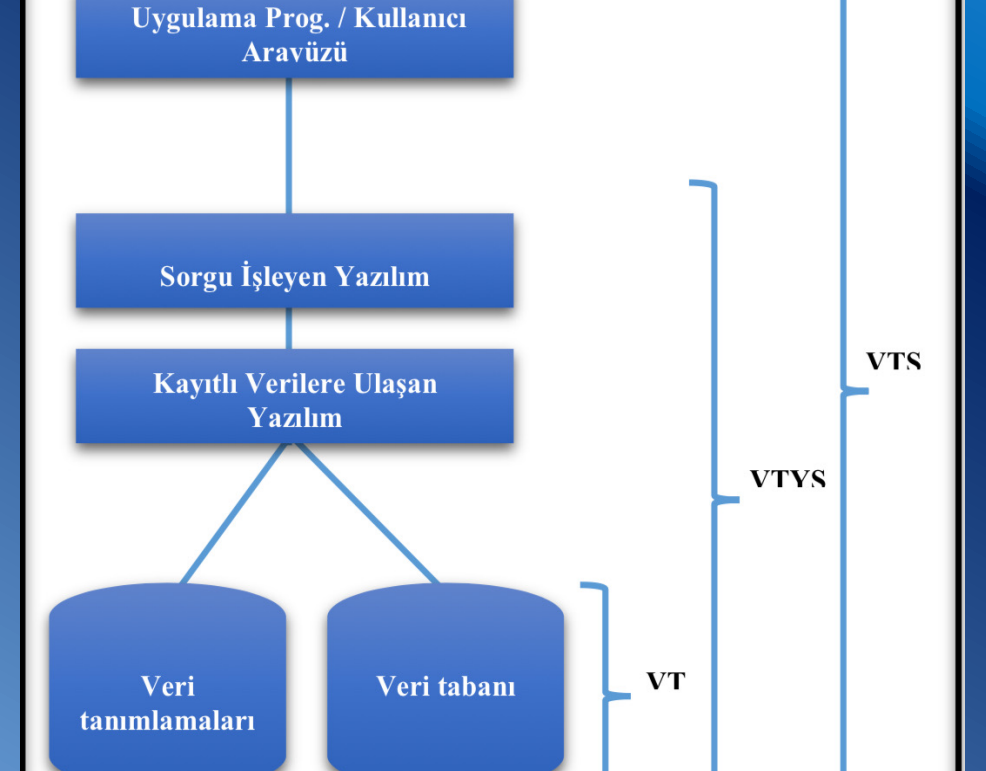


VERİ TABANI VE VERİ TABANI YÖNETİM SİSTEMLERİ

Veri tabanı en genel tanımıyla, kullanım amacına uygun olarak düzenlenmiş veriler topluluğudur. Veri tabanları gerçekte var olan ve birbirleriyle ilişkisi olan nesneleri ve ilişkileri modeller .

Veri tabanı yönetim sistemleri (VTYS), verilere aynı anda birden çok bağlantı sağlayabilme özelliği sağlar .

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 şekil 3.1’de gösterilmiştir.



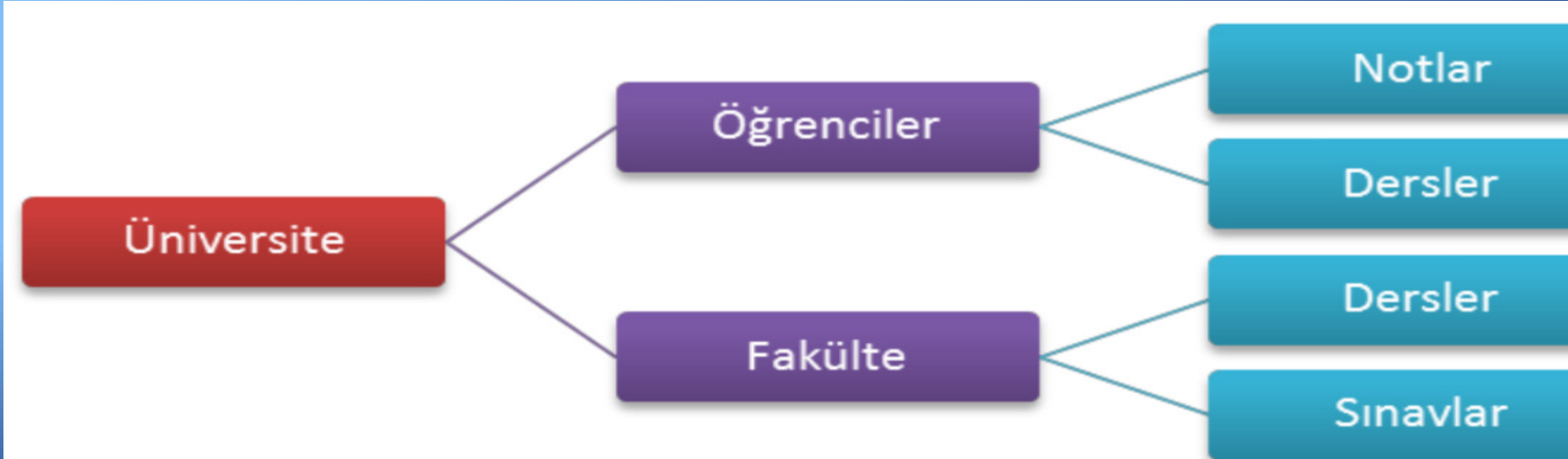
- **Düz model veya tablo modeli:**

iki 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 .

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

- **Hiyerarşik Veri Modeli:**

İlk olarak 1960'lı yıllarda ortaya çıkmış ve adını veriyi depolama yönteminden almıştır. 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:

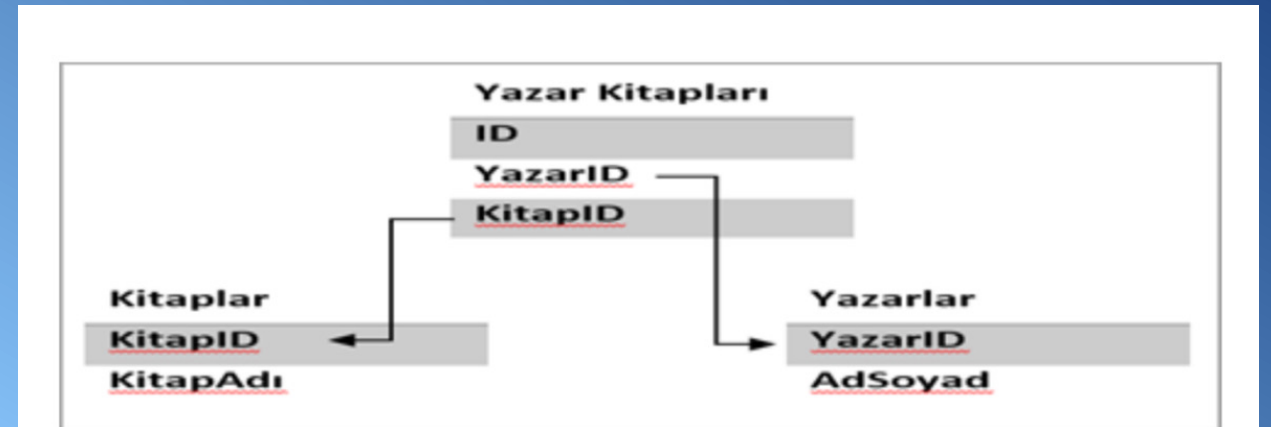
Bu model 1970'li yılların başında geliştirilmiştir. Hiyerarşik veri modelinin geliştirilmiş halidir. Hızlıca kabul görmesinin nedeni bir verinin doğal olarak başka veriler ile ilişkili olmasıdır. Ağ modelinin hiyerarşik modelden en önemli farkı, uç- düğüm pozisyonundaki verinin iç-düğümüne işaret edebilmesidir. Böylelikle ağ modelinde bire-çok ilişkiler yanında , çoka-çok ilişkiler de modellenebilir. Bu veri tekrarını önemli ölçüde azaltır



• İlişkisel Veri Modeli:

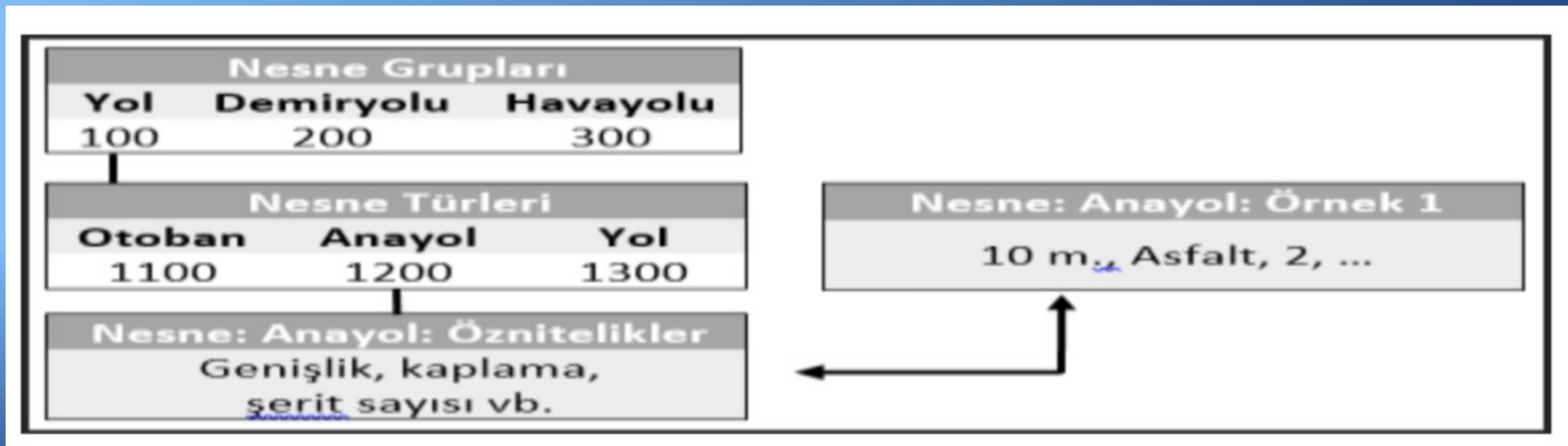
Hiyerarşik ve ağ veri modellerinin, çeşitlenen beklentileri karşılamakta yetersiz kalması, yeni bir model arayışını başlatmış ve ilişkisel veri modeli geliştirilmiştir. E. F. Codd'un 1970'de yazmış olduğu "A Relational Model of Data for Large Shared Data Banks" makalesi ile ilişkisel veri yapılarında büyük bir ilerleme kaydedilmiştir. İlişkisel veri modelinin temel kavramı, ilişkidir.

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.



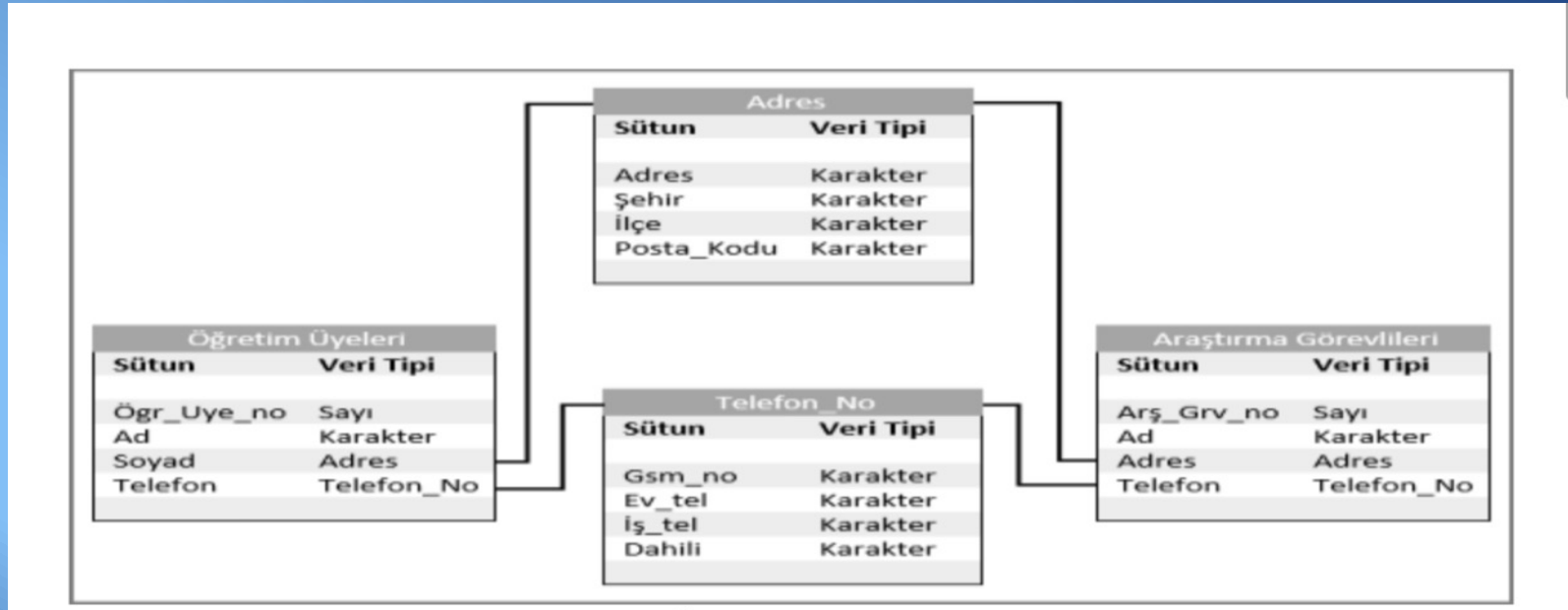
- Nesne Yönelimli Veri Modeli:

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:

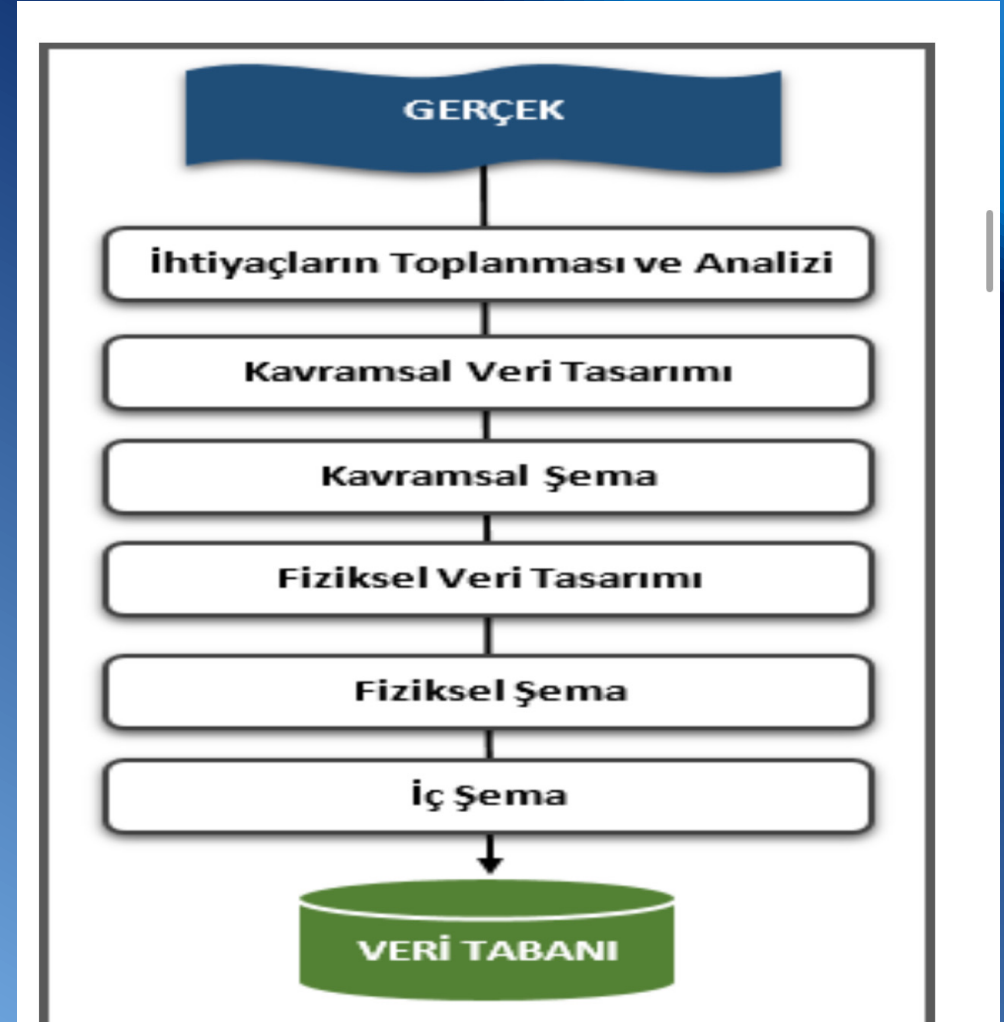
- Çoklu ortam veri tabanları nesne ilişkisel veri tabanları ile büyük benzerlikler gösterir. Bununla birlikte, film, müzik, metin ve video gibi büyük nesneleri işlemek ve aynı zamanda işleme sırasındaki adımları kullanıcıya göstermemek için farklı özellikler taşır. Çoklu ortam veri tabanlarının desteklemesi 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.

VERİ TABANI TASARIMI

Veri tabanı tasarımında; gerçeğin, gereksinim ve beklentiler çerçevesinde modellenerek veri tabanına aktarılması gerekir. Veri tabanı tasarımında ilk olarak, olası veri tabanı kullanıcı gereksinimlerinin belirlenmesi gerekir. Gerçeğin veri tabanındaki sayısal temsili, onun belli bir perspektiften bir modeli olup, bir veri tabanı sisteminde gerek kullanıcılar ve gerekse bilgisayar tarafından anlaşılabilir bir tarzda tanımlanması gerekir. Böyle bir tanımlama, veri tabanı literatüründe “şema” olarak adlandırılır. Kullanıcı ve bilgisayar düzeyleri sırasıyla "kavramsal" ve "fiziksel" düzeyler, bu düzeylerdeki şemalar da “kavramsal şema” ve “iç şema” olarak anılırlar. Kavramsal ve fiziksel düzeylerdeki şemalar, farklı anlayış mekanizmalarına hitap ettiklerinden, kullanılacak veri modelleri de farklı olacaktır. Her iki düzeyde kullanılmak üzere, çeşitli veri modelleri geliştirilmiştir .



İLİŞKİSEL VE İLİŞKİSEL OLMAYAN (NoSQL) VERİ TABANI SİSTEMLERİ

Günümüzde en yaygın kullanılan veri tabanı sistemlerinden biridir. Satır ve sütunların meydana getirdiği tablolardan oluşur. Bu tablolar birbiri ile ilişkileri olan tablolardır. Dolayısıyla bir veri tabanında ilişkiden söz edebilmek için en az iki tablonun yer alması ve bu iki tablodaki verilerin birbiri ile bir şekilde ilişkilendiriliyor olması gerekir.

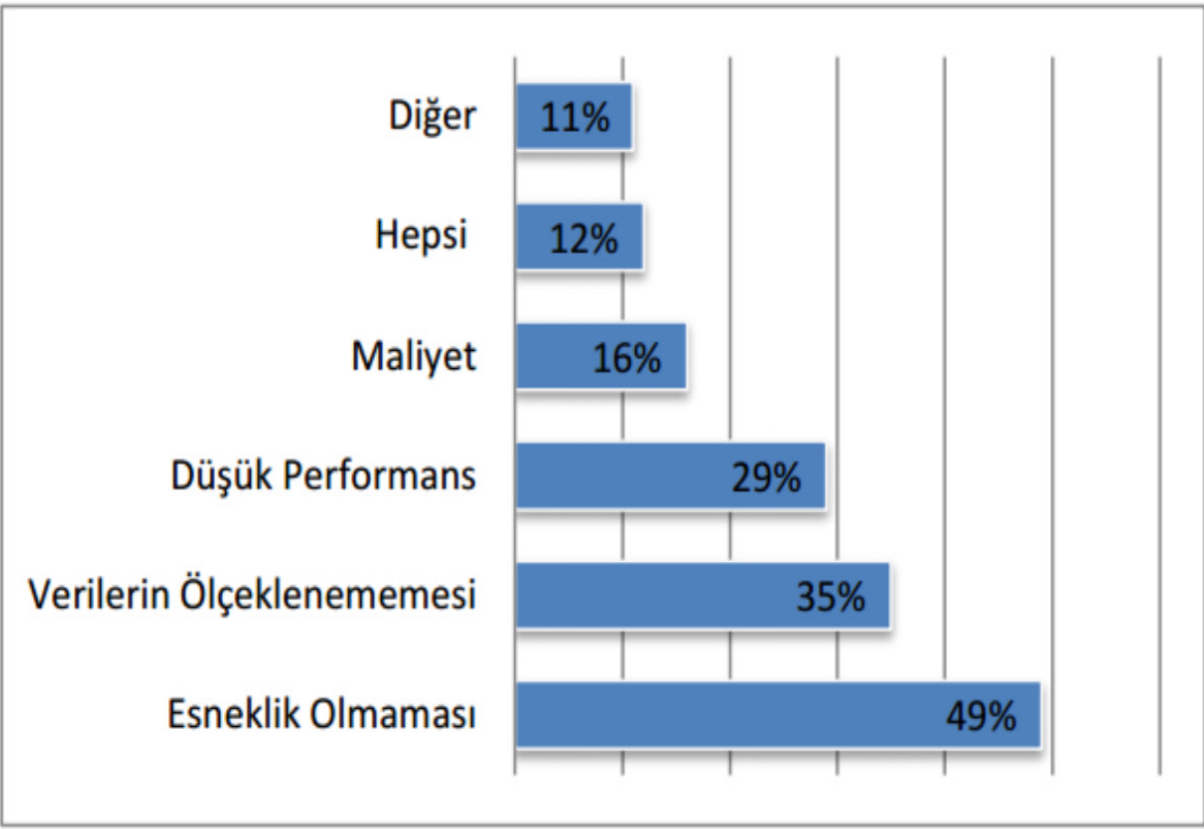
ACID; klasik ilişkisel veri tabanı sistemlerinde sağlanan temel özellikler aşağıda sunulmuştur :

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

İlişkisel olmayan (NoSQL) veri tabanı; 1998 yılında ilk olarak Carlo Strozzi tarafından öne sürülen bir kavramdır.

NoSQL, ilişkisel veri tabanı sistemlerine alternatif bir çözüm olarak ortaya çıkmıştır. İlişkisel olmayan veri tabanları yatay olarak ölçeklendirilen bir veri depolama sistemidir.

Dünya'da NoSQL örneklerini incelediğimizde; sosyal ağlarda Digg'in 3 TB'lık çözümü, Facebook'un gelen postaları arama için 50 TB ve eBay'ın bütün verileri için 2 PB'lık çözümleri vardır. Veri tabanlarına ilişkin problemlerden biri olan ölçek sorununa, diğer çözümlerin içinde en iyi cevap vereni NoSQL'dir. Günlük 7 TB'lık işlem hacmine sahip Twitter [16] ve 10 TB'lık Facebook örneğindeki gibi, çok büyük verilerin depolanması ve yazılmasında ilişkisel veri tabanlarının eksik kaldığı hususlarda, yatay ölçekleme yapan dağıtık NoSQL çözümleri geliştirilmiştir. İlişkisel veri tabanı kullanıcılarının, araştırmalar neticesinde NoSQL veri tabanına geçmek istemelerinin nedenleri şekil 5.1'de yüzde olarak gösterilmiştir.



VERİTABANI MİMARİLERİNİN PERFORMANS KARŞILAŞTIRMASI

Veri tabanı mimarilerinde oldukça bol çeşit ve bir o kadar da seçenek vardır. Bu çalışmada ilişkisel veri tabanı olarak günümüzde en yaygın kullanılan veri tabanı sistemlerinden biri olan MySQL ve ilişkisel olmayan (NoSQL) veri tabanı olarak ilişkisel veri tabanı sistemlerine alternatif bir çözüm olarak ortaya çıkan, yatay olarak ölçeklendirilen bir veri depolama sistemi olan MongoDB veri tabanı sistemi kullanılmıştır.

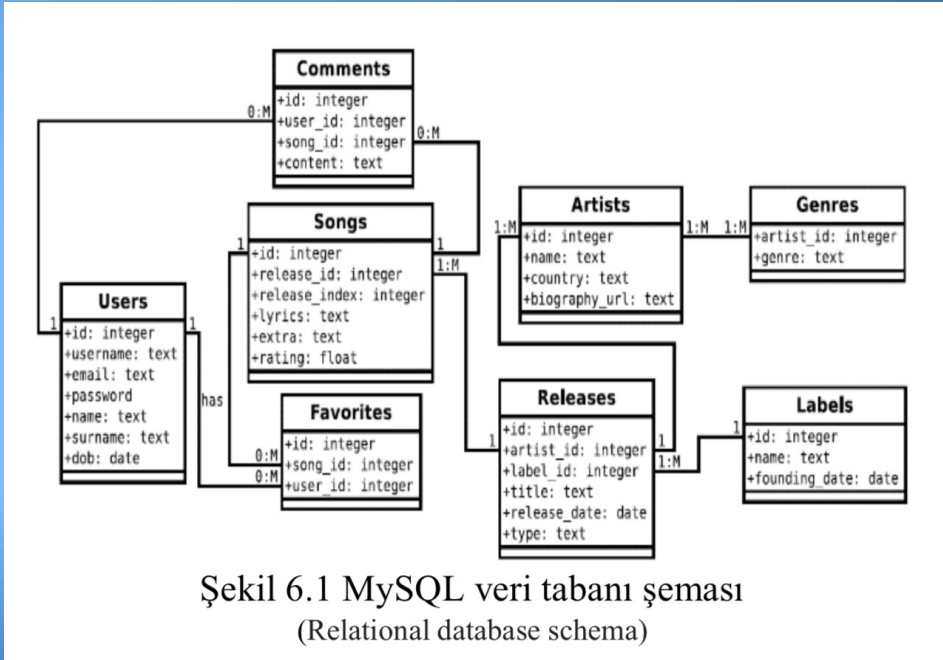
Yapılan çalışmada; MySQL ve MongoDB veri tabanı sistemlerinin performans ve yatay ölçeklenebilirlik incelemesi için aşağıdaki işlemlerin uygulanması ve sonuçlarının ortaya çıkarılması hedeflenmiştir.

Bunlar;

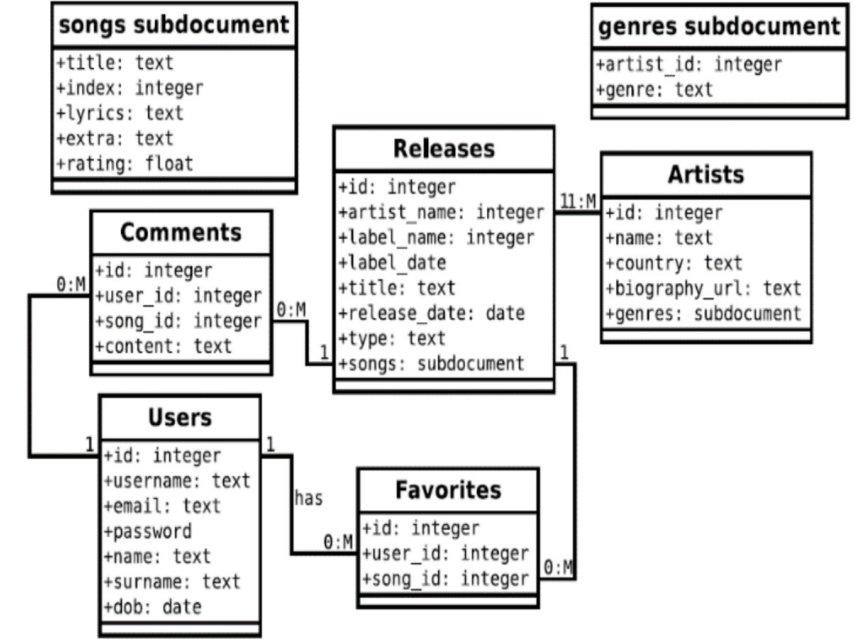
- Veri tabanı sunucu sistemleri özellikleri belirlenmesi,
- Veri tabanı şemaları oluşturulması,
- Sorguların belirlenmesi,
- Veri tabanı ayarlarının yapılması,
- Ölçümler ve ölçüm metrikleri bilgileri,
- Performans analizi ve sonuçlarıdır.

Veri Tabanı Şeması:

Projede iki adet veri tabanı şeması tasarlanmıştır. Biri MySQL (şekil 6.1), diğeri ise MongoDB (şekil 6.2) veri tabanıdır. Şemalar, kendi zevk ve tercihleri doğrultusunda diğerkullanıcılara şarkılar önermek için tasarlanmış farklı algoritmalar kullanan bir müzik uygulaması etrafında modellenmiştir. Tablolar arasında herhangi bir veri tekrarını ortadan kaldırmak için normalizasyon değeriendirmesi sağlanmıştır.



Şekil 6.1 MySQL veri tabanı şeması
(Relational database schema)



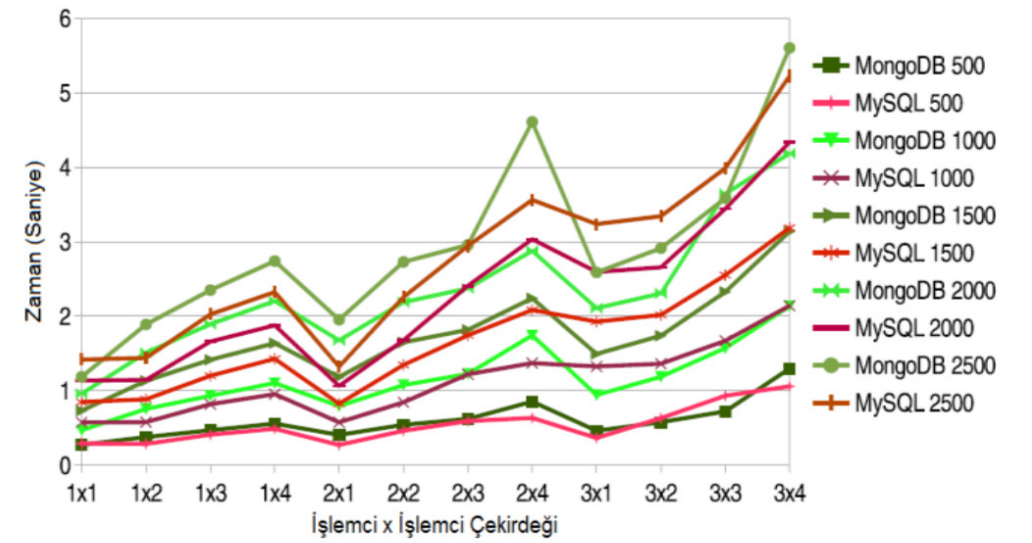
Şekil 6.2 MongoDB şeması
(MongoDB schema)

Analiz ve sonuçlar:

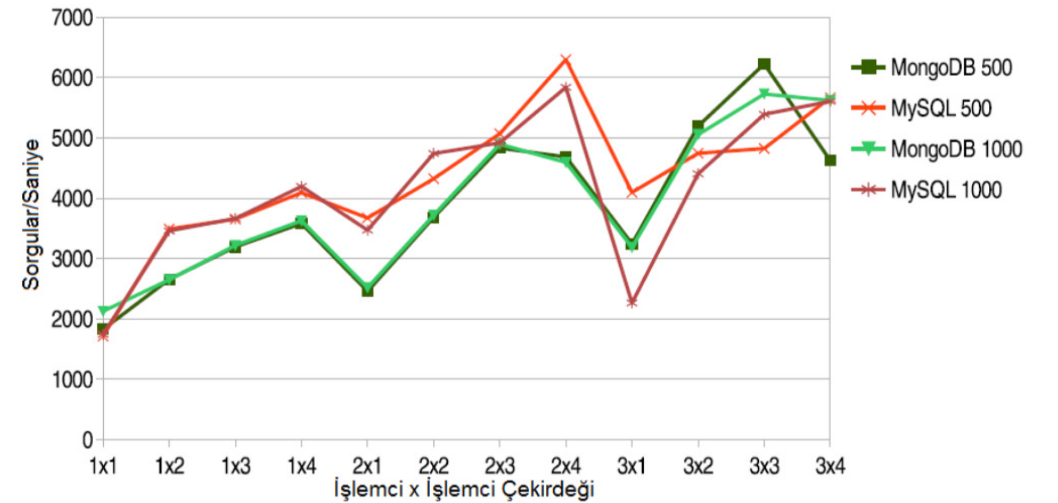
Burada öncelikle veri tabanlarının farklı sorgu türlerine göre nasıl yanıt verdiği hem okuma hem yazma ile analiz edilen sorguların toplam sayısı ve sonuçları şekillerle gösterilmiştir. Son olarak veri tabanı boyutunun performansa etkisi konusunda inceleme yapılmıştır.

Şekil 6.3'de MySQL ve MongoDB veri tabanlarına sorgu 1 (basit sorgu) ile karşılaştırma testi uygulanmıştır. Yapılan analizde; MongoDB, sorgu sayısı farkı arttıkça daha belirgin bir performans kötülüğü gösterdiği tespit edilmiştir. Bu karşılaştırma, işlemci çekirdeği sayılarının toplam sayısı aynı olduğu zaman, 2 ya da 1 işlemci kullanımının değişmez olduğunu açıkça ortaya koymuştur (1x2 ve 2x1). MySQL veri tabanının, özellikle 3 işlemci sayısı ile 1 işlemci çekirdeği sayısına göre incelendiğinde daha kötü performans gösterdiği görülmektedir.

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

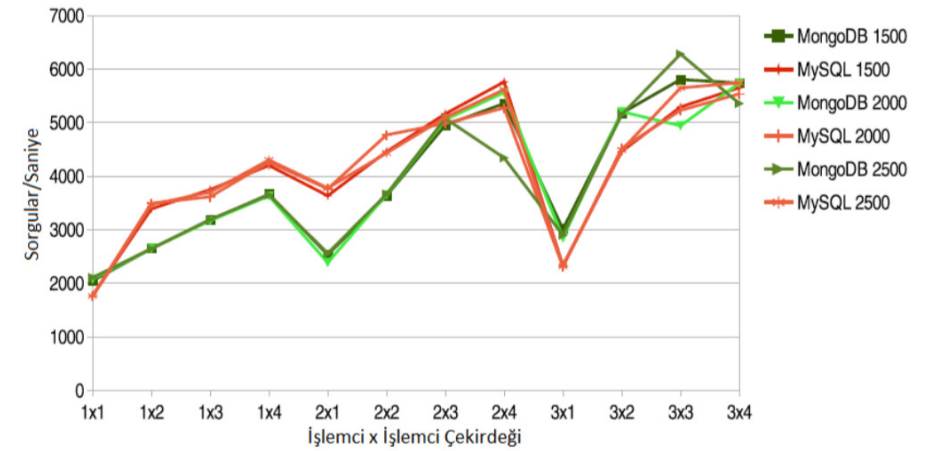


Şekil 6.3 Sorgu 1- Analiz işlemi
(Query 1 - Analysis process)



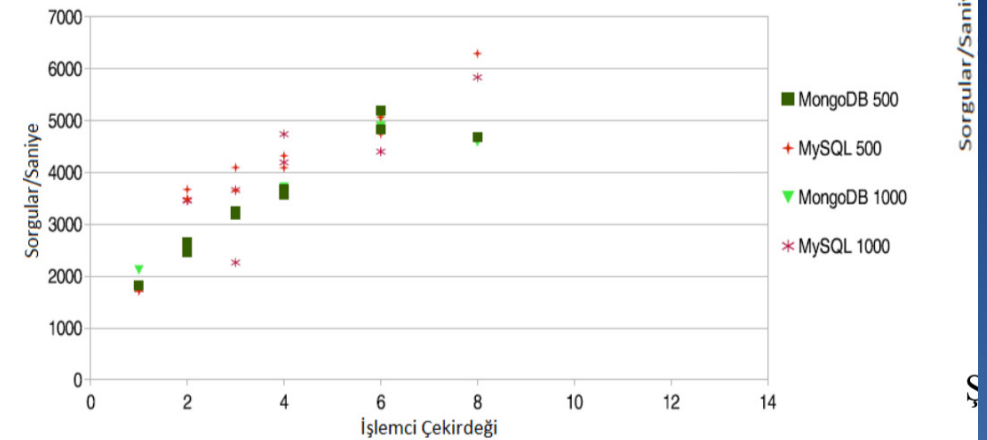
Şekil 6.4 Sorgu 1 - Sorgu/saniye analiz işlemi
(Query 1 - Queries/second analysis process)

Şekil 6.5’de MySQL veri tabanı sisteminin, sorgu sayıları arttığında MongoDB üzerinde avantaj sahibi olduğu görülmektedir. Fakat 2 işlemci ve 3 işlemci çekirdeği yapılandırmasından sonraki diğer yüksek işlemci-işlemci çekirdeği sayılarında sorgu/saniye grafiğinde keskin bir şekilde azalma görülmektedir. MongoDB bu yapılandırmalarda daha fazla avantaj göstermiştir.



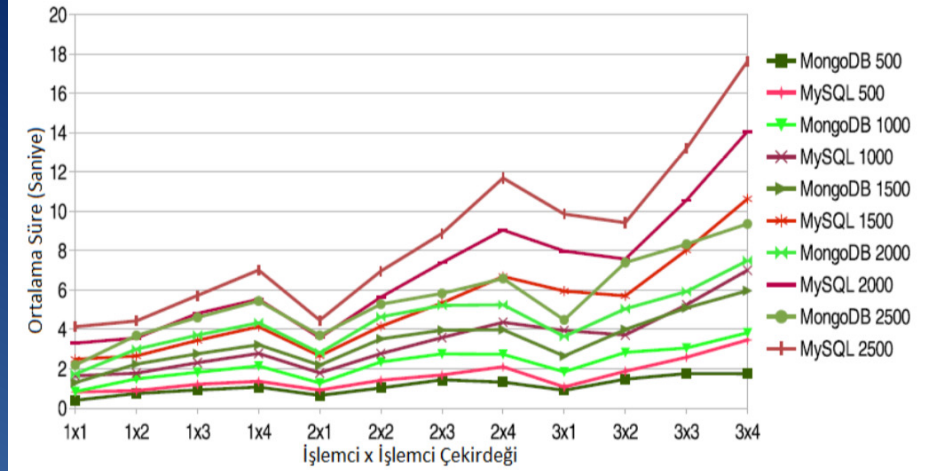
Şekil 6.5 Sorgu 1-Çok sayıdaki sorgu miktarı analiz işlemi
(Query 1- Many query amount analysis process)

Şekil 6.6’da işlemci çekirdeği miktarı ile saniye başına yapılan sorgu sayıları arasındaki ilişki analizi gösterilmektedir. MySQL için biraz daha iyi olan performans 4 işlemci çekirdeğine kadar hemen hemen aynıdır.



Şekil 6.6 Sorgu 1-Sorgular/Saniye ile işlemci çekirdeği miktarı için analiz işlemi
(Queries/second and amount of processor core analysis process)

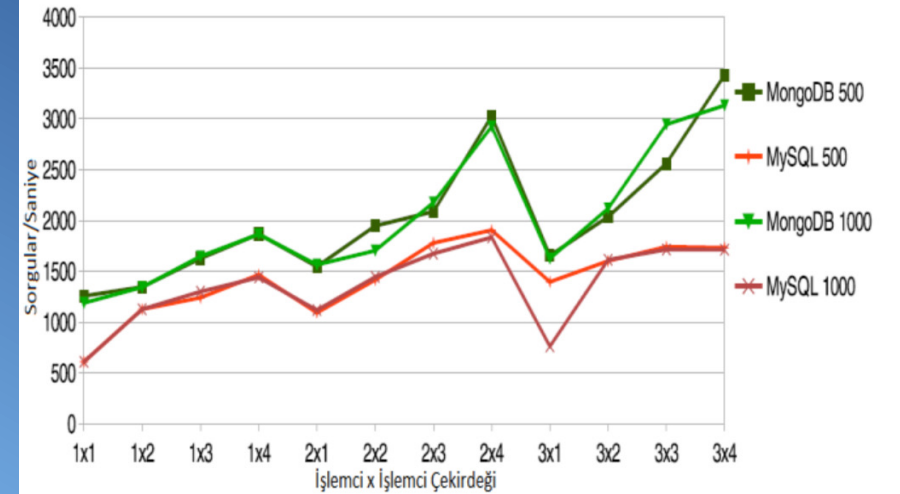
Şekil 6.7’de MySQL ve MongoDB veri tabanlarına ikinci sorgu kodu ile karşılaştırma testi uygulanmıştır. Yapılan analizde; MySQL veri tabanı sisteminin MongoDB’ye göre ortalama sorgu süreleri sonuçları, sorgu sayısı farkı arttıkça daha belirgin bir performans kötülüğü göstermiştir.



Şekil 6.7 Sorgu 2 - INNER JOIN ile karmaşık sorgu analizi işlemi

(complex query analysis process with INNER JOIN)

Şekil 6.8’de MySQL ve MongoDB veri tabanlarına ikinci sorgu kodu ile karşılaştırma testi uygulanmıştır. Bu test 500 ve 1000 gibi küçük veri kayıtları üzerinde yapılmıştır. 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.



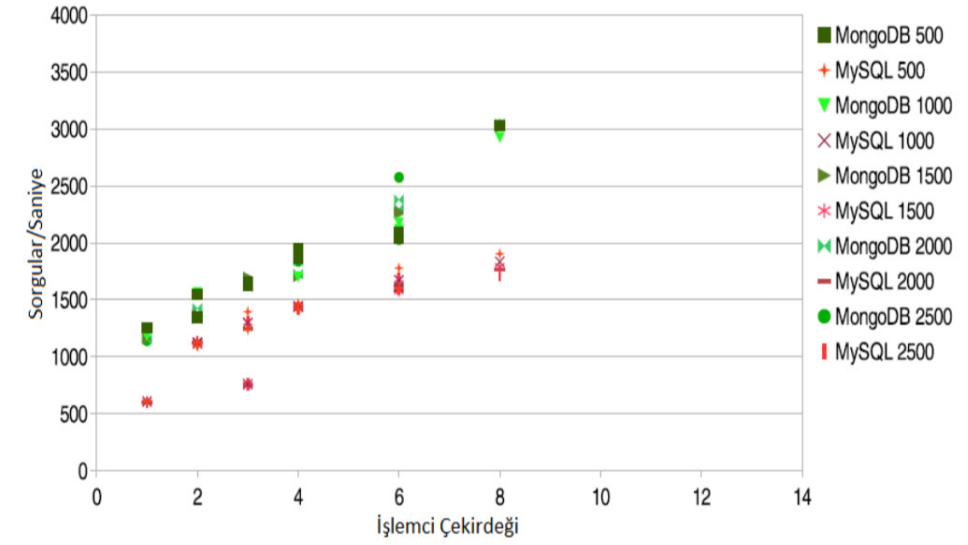
Şekil 6.8 Sorgu 2- INNER JOIN ile 500 ve 1000 veri için sorgu/saniye analizi işlemi

(500 and 1000 data query / second analysis process with INNER JOIN)

İşlemci çekirdeği miktarı ile saniye başına yapılan sorgu sayıları arasındaki ilişki analizi şekil 6.9’da gösterilmektedir. İkinci sorgu kodu ile karşılaştırma testi uygulanmıştır. Bu test 500 ile 2500 veri kayıt setleri üzerinde yapılmıştır. MySQL veri tabanı sistemi, veri kayıt miktarı ve sorgu sayısı artışına göre öncelikle kademeli bir düşüş ve ardından küçük performans artışları gösterdiği tespit edilmiştir. MySQL, artan işlemci çekirdeği ve veri kayıt miktarlarında birbirine çok yakın sorgu/saniye performansı göstermiştir. MongoDB veri tabanı sisteminin, MySQL’e göre oldukça yüksek bir performans sergilediği gözlemlenmiştir. Aynı veri kayıt setlerinde MongoDB’nin MySQL üzerindeki belirgin performans farkı ve avantajı görülmektedir.

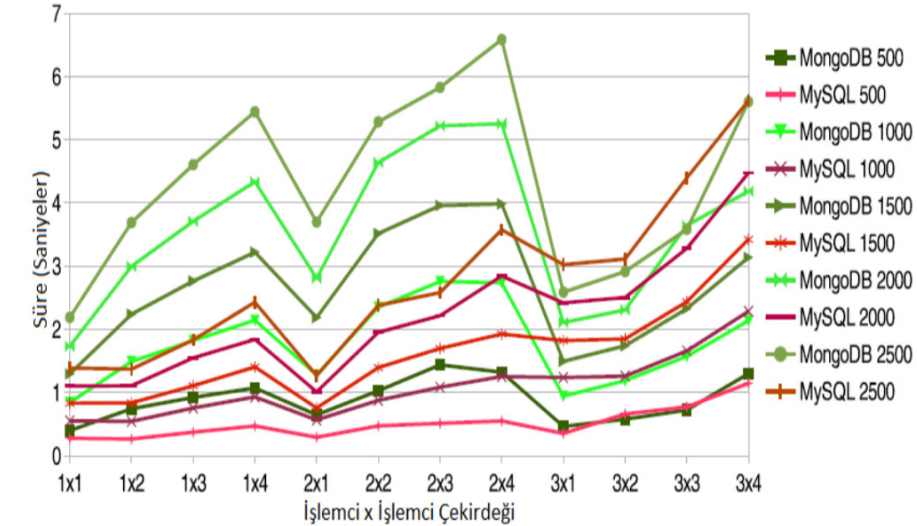
Şekil 6.10’da iç içe geçmiş “SELECT” ve “WHERE” işlemlerini içeren üçüncü sorgu neticesinde ortaya çıkan performans değerleri gösterilmektedir.

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.



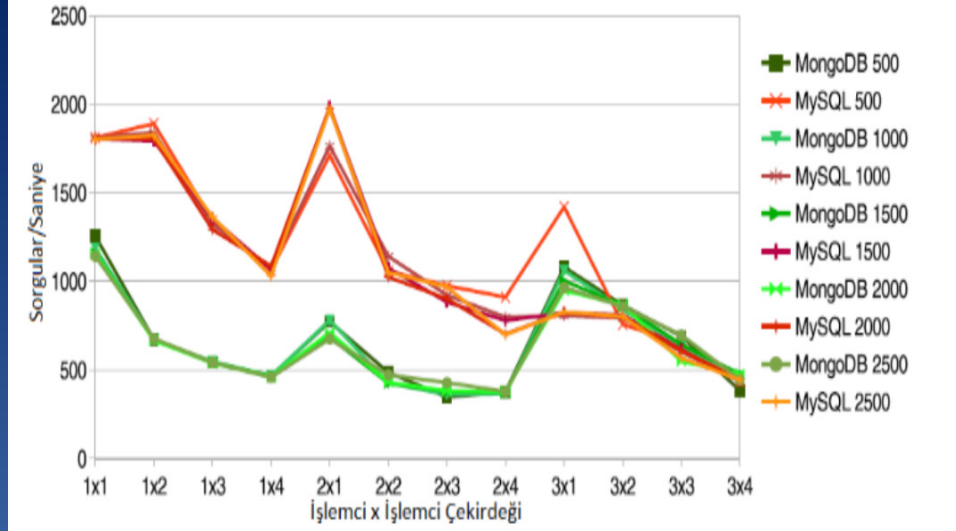
Şekil 6.9 Sorgu 2- INNER JOIN ile işlemci çekirdeği miktarı üzerinde analiz işlemi

(analysis on the amount of processor core operation with INNER JOIN)



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

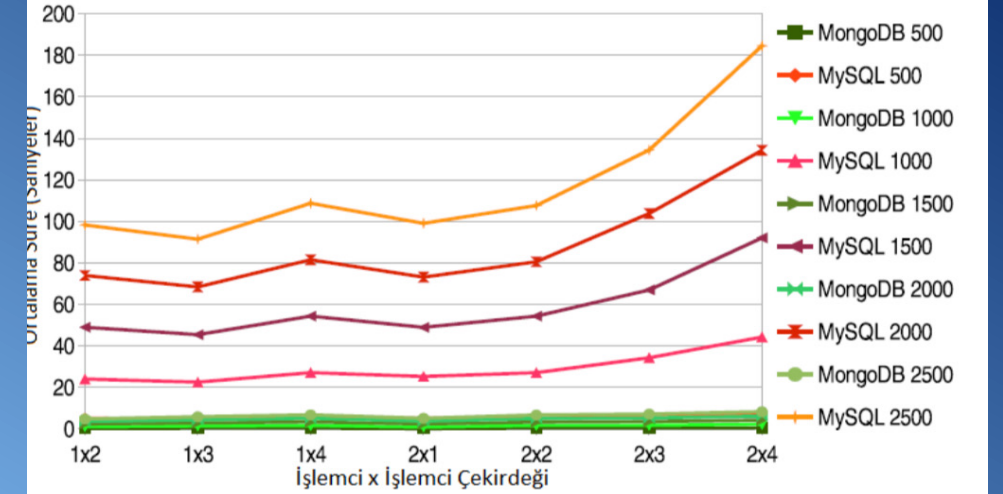
MySQL ve MongoDB veri tabanlarına üçüncü sorgu kullanılarak uygulanan karşılaştırma testi sonuçları Şekil 6.11’de daha iyi anlaşılmaktadır. MySQL veri tabanı sisteminin 2x4 işlemci ve işlemci çekirdeği yapılandırmasında en iyi performansı gösterdiği açık bir şekilde görülmektedir. Fakat 2x1 ve 3x1 işlemci ve işlemci çekirdeği yapılandırmalarında her iki veri tabanı için performans sorunu olduğu gözlemlenmektedir. Bu performans sorunu MySQL’de daha belirgin haldedir.



Şekil 6.11 Sorgu 3- Detaylı ve karmaşık sorgu ile Sorgular/saniye analiz işlemi

(Queries/second analysis process with detailed and complex queries)

Şekil 6.12’de üçüncü sorgu ile ortalama süre ölçümleri gösterilmiştir. Yapılan analizde; MySQL veri tabanı sisteminin MongoDB’ye göre ortalama sorgu süreleri sonuçları, veri kayıt sayısı farkı arttıkça oldukça belirgin bir performans kötülüğü gösterdiği gözlemlenmiştir. Bu karşılaştırma, eşdeğer veri kayıt setlerinde yapılmasına rağmen MySQL veri tabanının sorguları işleme ve sonuçlandırma süresi ortalamasının oldukça yüksek olduğu görülmektedir. MongoDB daha belirgin ve istikrarlı bir performans göstermektedir.

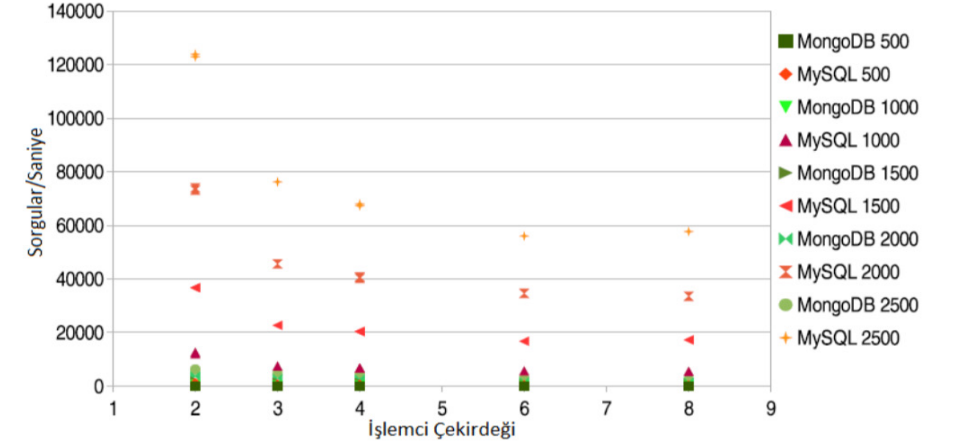


Şekil 6.12 Sorgu 3 – Detaylı ve karmaşık sorgu kodu ile ortalama süre analiz işlemi

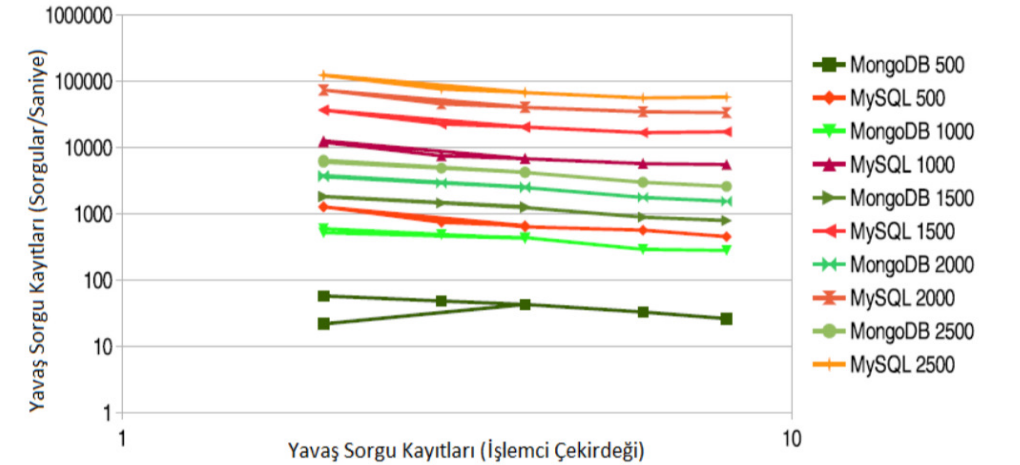
(The average time analysis process with detailed and complex query code)

MySQL ve MongoDB veri tabanlarına üçüncü sorgu olarak tanımlanan detaylı ve karmaşık sorgu kodu içeren karşılaştırma testi analizi Şekil 6.13’de gösterilmiştir. Bu test 500 ile 2500 veri kayıt setleri üzerinde yapılmıştır. MySQL veri tabanı sistemi, iki eksen boyunca logaritma kullanılarak çizilen grafikte logaritmik bir eğilim olduğu görüntüsü sergilemektedir. MongoDB’nin ise eğilimi nerede ve nasıl gösterdiği net olarak görülmemektedir.

Zamanlama ölçeği büyütülerek veri tabanları sistemleri arasındaki performans farkının Şekil 6.14’de daha anlaşılabilir hale geldiği görülmektedir. Ölçek büyüdükçe MySQL’in performansındaki dezavantaj açıkça görülmektedir. MongoDB tüm veri kayıt setlerinde oldukça iyi bir performans gösterdiği ortaya konulmuştur.



Şekil 6.13 Sorgu 3 - Detaylı ve karmaşık sorgu ile işlemci çekirdeği üzerinde analiz işlemi
(Process analysis of processor cores with detailed and complex queries)

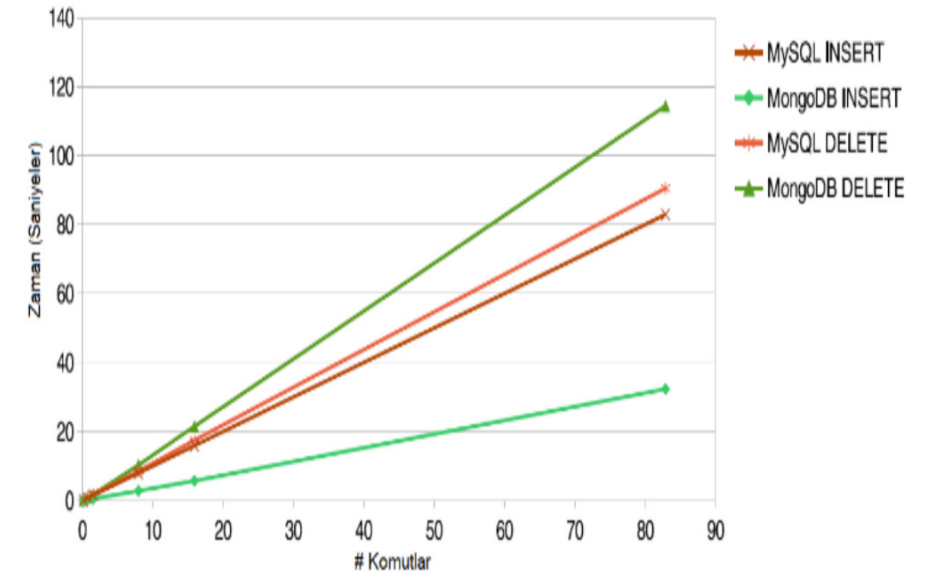


Şekil 6.14 Sorgu 3- Detaylı ve karmaşık sorgu ile ölçeklendirilmiş analiz işlemi
(Scaled analysis process detailed and complex queries)

Son olarak MySQL ve MongoDB veri tabanlarına veri ekleme “INSERT” ve silme “DELETE” işlemleri uygulanmıştır. Veri tabanı komutlarının çoğu veri okuma ve yazma sorguları olmasına rağmen silme sorguları da önemli bir faktör olarak görülmektedir. Şekil 6.15’de her iki veri tabanı sisteminin INSERT ve DELETE işlemlerine ait performans grafiği gösterilmektedir.

Yapılan analizde; her iki veri tabanının komut sayılarına göre işlem süreleri doğrusal bir eğilim göstermektedir. MongoDB’nin veri ekleme işlemi MySQL’e göre çok daha iyi bir performansa sahiptir.

Veri silme işleminde ise MongoDB’nin MySQL ile benzer bir performansa sahip olduğu fakat veri silme komut sayılarının artışı ile MySQL veri tabanı sisteminin silme işleminde iyi bir performans sergilediği gözlemlenmiştir.



Şekil 6.15 INSERT ve DELETE işlemleri
(INSERT and DELETE operations)

SONUÇ VE DEĞERLENDİRME

Klasik ilişkisel veri tabanlarını altküme olarak gören, aynı zamanda SQL ve Daha Fazlası (Not Only SQL) olarak da adlandırılan dağıtık mimari ile oluşturulmuş veri tabanları ile ilişkisel veri tabanları karşılaştırılmış ve yönetim bilişim sistemleri açısından incelenmiştir.

Bu çalışmada, ilişkisel veri tabanları ile NoSQL veri tabanları karşılaştırılarak yönetim bilişim sistemleri açısından incelenmiştir. Veri tabanlarının modellenmesi, performans ölçümleri, süreçlerin uygun hale getirilmesi ve veri tabanı seçimi konularında kullanıcılara rehberlik edilmiştir. NoSQL'un teknolojik ilerleme ile önemi artmış ve performans açısından avantajları ortaya çıkmıştır. Literatür taramasında yapılan çalışmalar, NoSQL veri tabanlarının performansını vurgulamıştır. Yapılan analizlerde, MongoDB'un karmaşık sorgularda MySQL'den daha iyi performans gösterdiği ve veri büyüklükleri ile ilgili avantajları olduğu belirlenmiştir. Sonuç olarak, NoSQL'un hız, geliştirme zamanı ve ölçeklenebilirlik gibi özellikleri ile ilişkisel veri tabanlarından daha etkin sonuçlar sağladığı sonucuna varılmıştır.