İlişkisel ve İlişkisel Olmayan (NoSQL) Veri Tabanı Sistemleri Mimari Performansının Yönetim Bilişim Sistemleri Kapsamında İncelenmesi

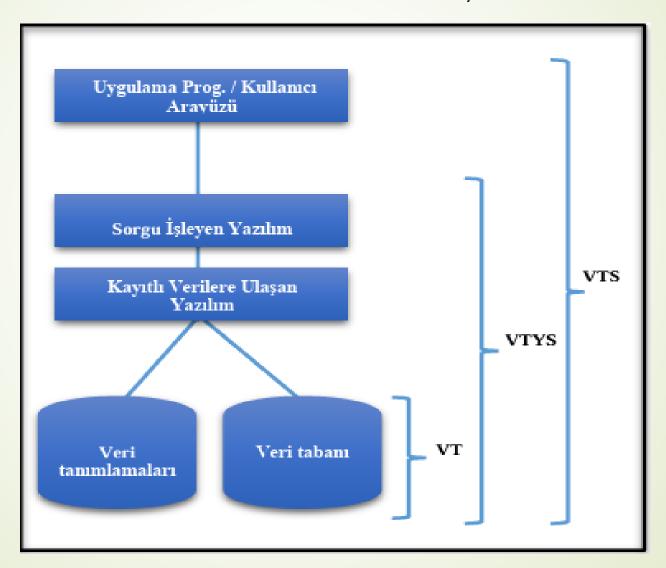
# BİLİŞİM SİSTEMLERİ VE YÖNETİMİ (INFORMATION SYSTEMS AND MANAGEMENT)

- Bilişim sistemleri, bilginin toplanmasında, işlenmesinde, depolanmasında, ağlar aracılığıyla bir yerden bir yere iletilip kullanıcıların hizmetine sunulmasında kullanılan iletişim ve bilgisayarlar dâhil bütün teknolojileri kapsar. Bilişim sistemlerinde girdi, işlem ve çıktı bilgiyi üretmek için gereklidir.
- Girdi: Ham bilgileri toplar.
- İşlem: Bu ham veriyi daha anlamlı biçime çevirir.
- Çıktı: İşlenmiş bilgiyi insanlara ve kullanılacak olan aktivitelere aktarır.

### VERI TABANI VE VERI TABANI YÖNETIM SISTEMLERI (DATABASE MANAGEMENT SYSTEM) AND DATABASE

- Veri tabanı kullanım amacına uygun olarak düzenlenmiş veriler topluluğudur. Birbiriyle ilişkili verilerin tutulduğu bilgi depolarıdır. Veri tabanları gerçekte var olan ve birbirleriyle ilişkisi olan nesneleri ve ilişkileri modeller.
- Veri tabanı yönetim sistemleri, verilere aynı anda birden çok bağlantı sağlayabilme özelliği sağlar. Bu sistemler verini nasıl depolanacağını, kullanılacağını ve erişileceğini mantıksal olarak yönlendiren kurallar sistemidir. Veri tabanı VTYS ve uygulama programları ile kullanıcı ara yüzlerini içeren yapıya «veri tabanı sistemi» denir.

# VT-VTYS-VTS Arasındaki İlişki ve İşlevler (Relation and Functions between DB-DBMS-DBS)

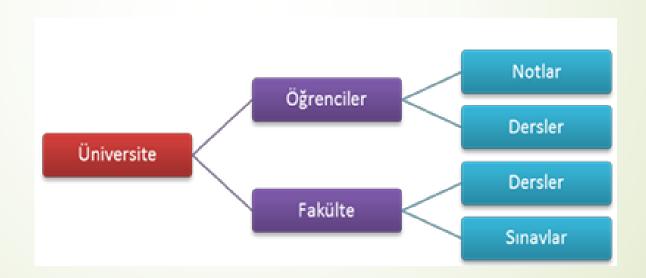


# Veri tabanı sistemleri sekiz kategoriye ayrılır.

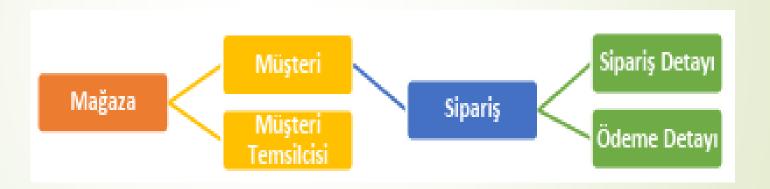
1- Düz model ve ya tablo modeli: İki boyutlu veri grubundan oluşur. Sütunlarda verilerin benzer özelliklerini, satırlarda ise veri grupları yer alır.

	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			

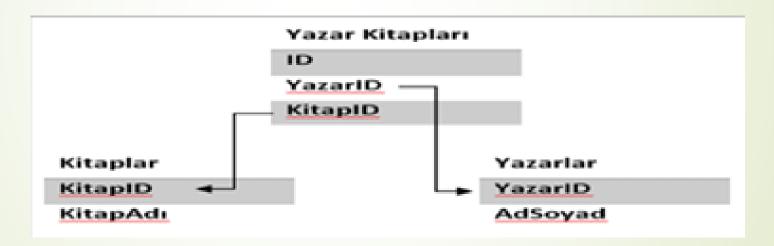
2- Hiyerarşik Veri Modeli: Adını veriyi depolama yönteminden almıştır. Bu veri tabanının depoladığı yapısal verilere «kayıt» adı verildi. Kayıtlar ağaç mimarisi şeklinde yukarıdan aşağıya sıralanmaktadır. Kök adı verilen ilk kaydın bir veya daha çok çocuk kayıtları olabilir. Kök haricinde bütün kayıtların bir ebeveyni vardır.



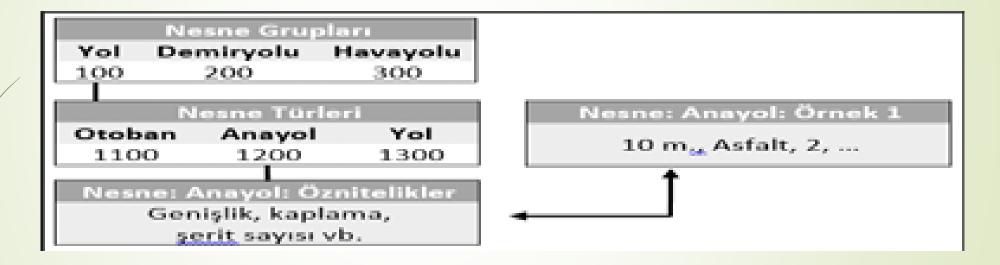
3- Ağ veri modeli: Hiyerarşik modelin geliştirilmiş halidir. Hiyerarşik modelden an önemli farkı uç-düğüm pozisyonundaki verinin iç-düğüme işaret edebilmesidir.



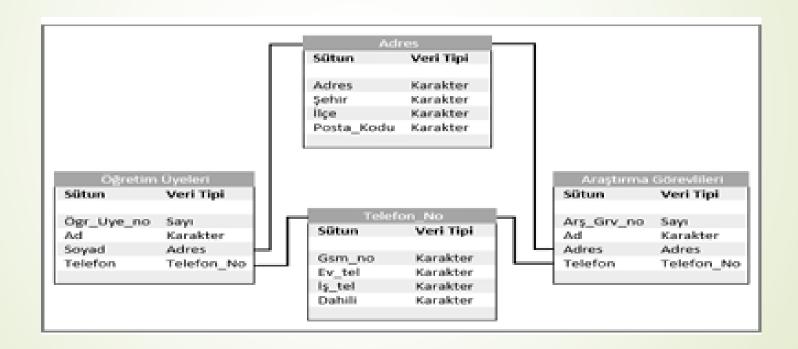
4- İlişkisel veri modeli: Hiyerarşik ve ağ veri modellerinin çeşitlenen beklentilerinin karşılanmakta yetersiz kalması nedeniyle geliştirilmiştir. Temel kavramı, ilişkidir. Genellikle veri tabanında her tablo için bir dosya bulunur. Tablonun her satırı birbiriyle ilişkili verilerin bir topluluğudur. Sütunlarda ise nitelikler bulunur.



5- Nesne yönelimli veri modeli: Nesne yönelimli programlamaya dayanan veri modelidir.



 6- Nesne ilişkisel veri modeli: Nesne ilişkisel veri tabanı, ilişkisel işlevselliğin üzerine nesne yönelimli özellikler içerir.



7- Çoklu Ortam Veri Modeli: Çoklu ortam veri tabanları nesne ilişkisel veri tabanları ile büyük benzerlikler gösterir. . Çoklu ortam veri tabanlarının desteklemesi gereken üç temel özellik; Veri miktarı, Süreklilik ve Senkronizasyondur.

8- 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.

### VERI TABANI TASARIMI (DATABASE DESIGN)

- Veri tabanı tasarımında gereksinimler ve beklentiler gerçeklikle modellenerek veri tabanına aktarılır.
- İlk olarak kullanıcı gereksinimleri belirlenir. Bu gereksinimler, veri tabanında yer alacak veri gruplarını, verilerin tiplerini ve verinin fiziksel olarak depolanması için kullanılacak olan veri yapılarını belirler. Şema oluşturulur. Kullanıcı ve bilgisayar düzeyleri sırasıyla "kavramsal" ve "fiziksel" düzeyler, bu düzeylerdeki şemalar da "kavramsal şema" ve "iç şema" olarak anılırlar.



- Geleneksel veri tabanı tasarımı, kullanıcı düzeyinden fiziksel düzeye doğrudur. Kavramsal tasarımda gereksinimlere göre kavramsal şema belirlenir. Kavramsal şema, ortalama veri tabanı kullanıcısı için, veri tabanının yapısını genel olarak tanımlar. Kullanıcıların veri tabanının yapısını anlamalarına ve böylece uygulamalarını modellemelerini sağlar. Kavramsal şema, yazılım ve donanımdan bağımsızdır ve son kullanıcı tarafından anlaşılması da daha kolaydır.
- Fiziksel tasarım aşamasında, verinin en yüksek verim için, veri tabanında fiziksel olarak nasıl organize edilmesi gerektiği belirlenir. Sonuç, iç şemadır. İç şema depolama yapılarını, kayıt formatlarını, kayıt alanlarını, veri tabanına giriş yol ve yöntemleri ile veri tabanının fiziksel gerçekleştirimini ilgilendiren diğer bütün detayları tanımlar.

## ilişkisel ve ilişkisel olmayan (Nosql) veri tabanı Sistemleri (Relational and non relational database (Nosql) systems)

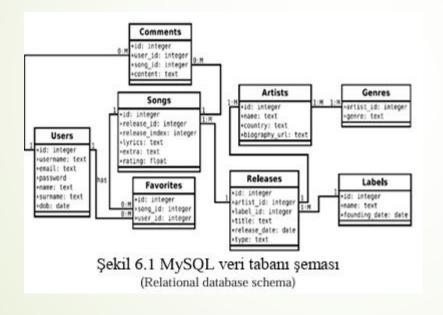
- İlişkisel Veri Tabanı (Relational Database System): Günümüzde en yaygın kullanılan veri tabanı sistemlerinden biridir. Satır ve sütunların meydana getirdiği tablolardan oluşur. Bu tablolar birbiri ile ilişkileri olan tablolardır. En az iki tablodan oluşur. Her bir tablo, belli yapıya uygun verileri saklamak üzere tasarlanır.
- ACID; klasik ilişkisel veri tabanı sistemlerinde sağlanan temel özellikler:
- Bölünmezlik (Atomicity)
- Tutarlılık (Consistency)
- İzolasyon (Isolation)
- Dayanıklılık (Durability)

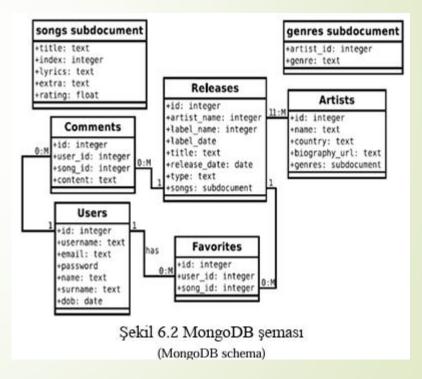
- İlişkisel Olmayan (NoSQL) Veri tabanı (Non-Relational Database System): İlişkisel veri tabanı sistemlerine alternatif bir çözüm olarak ortaya çıkmıştır. İlişkisel olamayan veri tabanlar,ı yatay olarak ölçeklendirilen bir veri depolama sistemidir
- İlişkisel veri tabanlarının kullandığı ACID işlemselliğine karşın NoSQL "BASE" (Basically Available- Soft state- Eventually consistent) kısaltması ile ifade edilir.
- Kolay Ulaşılabilirlik (Basically Available)
- Esnek Durum (Soft state)
- Eninde sonunda Tutarlı (Eventually consistent)

# VERI TABANI MIMARILERININ PERFORMANS KARŞILAŞTIRMASI (PERFORMANCE COMPARISON OF DATABASE ARCHITECTURE)

- Veri tabanı mimarisinde birçok seçenek vardır. Bu çalışmada 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:
- Veri tabanı sunucu sistemleri özellikleri belirlenmesi,
- Veri tabanı şemaları oluşturulması,
- Sorguların belirlenmesi,
- Veri tabanı şemaları oluşturulması,
- Ölçümler ve ölçüm metrikleri bilgileri
- Performans analizi ve sonuçları

İşlemleri uygulanmıştır. Veri tabanı şeması: Projede iki adet veri tabanı şeması tasarlanmıştır. Biri MySQL, diğeri ise MongoDB veri tabanıdır. Şemalar bir müzik uygulaması etrafında modellenmiştir. Tablolar arasında herhangi bir veri tekrarını ortadan kaldırmak için normalizasyon değerlendirmesi sağlanmıştır.





Veri tabanı sorguları: Bu çalışmada üç veri tabanı sorgusu kullanılmıştır. Birinci sorgu için "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

SELECT \* FROM Users WHERE username = 'username'

#### Sorgu 2: Karmaşık

SELECT' Favourites. song\_id' AS fSID,' Favourites. user\_id' AS fUID
FROM Favourites AS b INNER JOIN Favourites AS a
ON b. user\_id = a. user\_id
WHERE a. song\_id = 123456 AND a. user\_i d!= 987654

#### Sorgu 3: Detaylı ve karmaşık

SELECT' Songs. release\_id' AS sld,' Releases. id' ASrld
FROM Songs INNER JOIN Releases
ON Songs. release\_id = Releases. id
WHERE artist\_id IN
SELECT' Genres. artist\_id' AS gAID
FROM Genres AS c
INNER JOIN Artists AS d
ON c.artist\_id = d.id WHERE d.name = ' artist\_name'

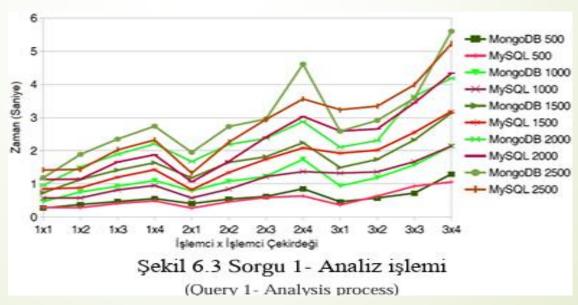
- Ölçümler: Zaman kavramı önemlidir. Zaman ölçümleri için üç yöntem ile hareket edilmiştir.
- Birinci yöntem; Clock() fonksiyonu kullanımı ile belirli bir süre CPU üzerinde harcanan zaman sonuçlarının elde edilmesini sağlamaktır.
- İkinci yöntem; milisaniye hassasiyetiyle zamanlamaları sağlayan Gettimeofday() fonksiyonu sonuçların elde edilmesini sağlamaktır. kullanılarak
- Üçüncü yöntem; Slow Query Log (Yavaş sorgu kaydı) olarak adlandırılmaktadır. Her veri tabanı zamanı ölçmek için kendi yöntemini sunmaktadır.

Ölçüm metrikleri: Veri tabanlarının performansını ölçmek için ortak bir metrik gereklidir. Bir uygulama için en önemli faktör, bir görevi tamamlamak için gereken süre ve veri tabanının bir işlemi tamamlaması durumu için gerekli zamandır.

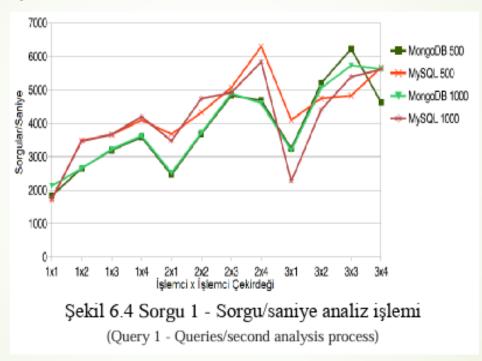
Her iş parçacığının saniye saniye sorgu başına nasıl tepki verdiğini ölçmek için aşağıdaki formül kullanılır.

Analiz ve sonuçlar: 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. Yapılan çalışmada daha önce açıklanan koşullar kapsamında veri tabanlarının detaylı olarak karşılaştırılabilmesi için çok çeşitli durumlar yaratılmak istenmiştir. Ölçüm için kullanılan yapılandırmalar 1'den 3'e kadar işlemci sayısı ve 1'den 4'e kadar işlemci çekirdek sayısı olarak değişmektedir. Ölçümlerde yapılan sorgu sayısı 500 ile 2500 arasındadır. Her bir ölçüm beş adet test ile bitirilmiştir. Her test sonucunda sorgulardan her birini gerçekleştirmek için alınan ortalama süreler hesaplanarak raporlanmıştır.

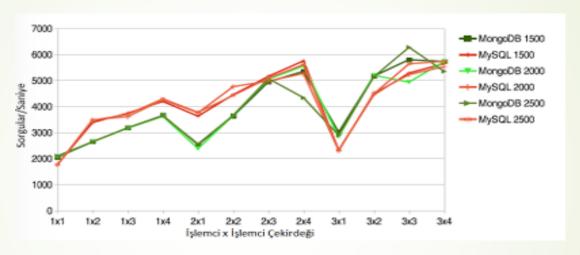
Şekil 6.3'de MySQL ve MongoDB veri tabanlarına sorgu 1 (basit sorgu) ile karşılaştırma testi uygulanmıştır. Yapılan analizde; MongoDB, sorgu sayısı farkı arttıkça daha belirgin bir performans kötülüğü gösterdiği tespit edilmiştir. Bu karşılaştırma, işlemci çekirdeği sayılarının toplam sayısı aynı olduğu 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



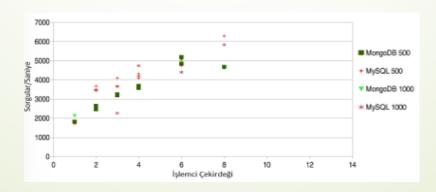
 sorgular/saniye ölçüm metrik grafiği ile de ayrıntılı ortalama süre sonuçları elde edilmiştir.



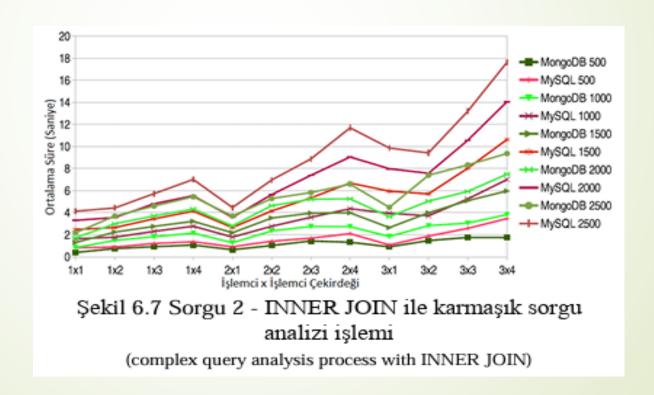
Şekil 6.5'de MySQL veri tabanı sisteminin, sorgu sayıları arttığında MongoDB üzerinde avantaj sahibi olduğu görülmektedir.



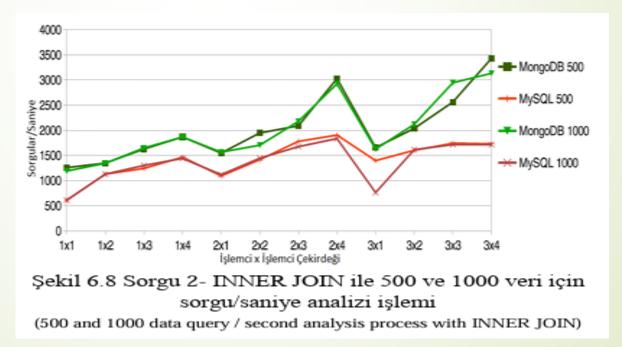
Şekil 6.6'da işlemci çekirdeği miktarı ile saniye başına yapılan sorgu sayıları arasındaki ilişki analizi gösterilmektedir.



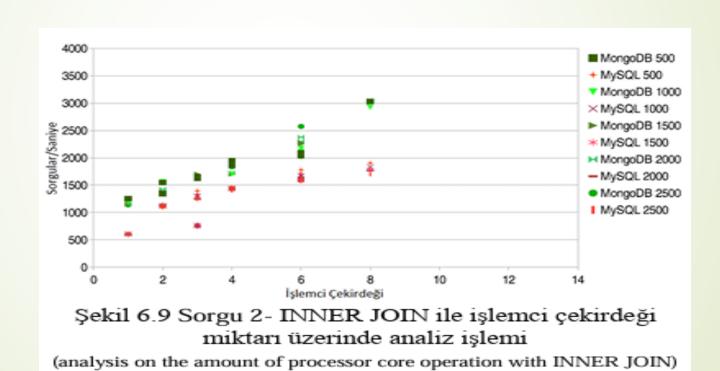
Şekil 6.7'de MySQL ve MongoDB veri tabanlarına ikinci sorgu kodu ile karşılaştırma testi uygulanmıştır. Yapılan analizde; MySQL veri tabanı sisteminin MongoDB'ye göre ortalama sorgu süreleri sonuçları, sorgu sayısı farkı arttıkça daha belirgin bir performans kötülüğü göstermiştir.



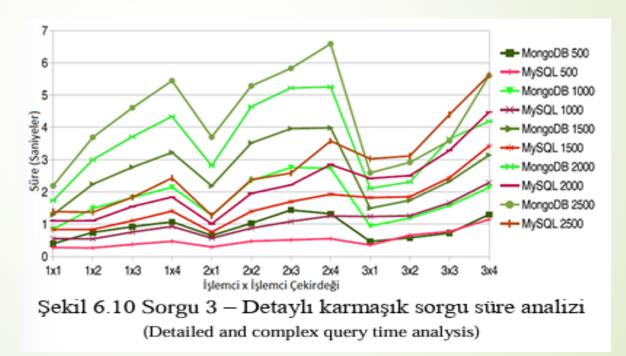
Şekil 6.8'de MySQL ve MongoDB veri tabanlarına ikinci sorgu kodu ile karşılaştırma testi uygulanmıştır. MongoDB veri tabanı sisteminin, daha az bir sürede daha çok sorgu yürütmesinin mümkün olduğu, sorgu sayısı değiştikçe performans ölçümünün daha belirgin hale gelerek sorgu/saniye başına %40 oranında daha iyi performans sergilediği gözlemlenmiştir.



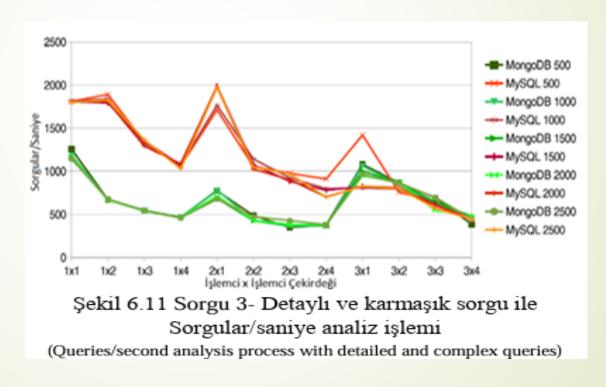
İşlemci çekirdeği miktarı ile saniye başına yapılan sorgu sayıları arasındaki ilişki analizi şekil 6.9'da gösterilmektedir. MongoDB veri tabanı sisteminin, MySQL'e göre oldukça yüksek bir performans sergilediği gözlemlenmiştir.



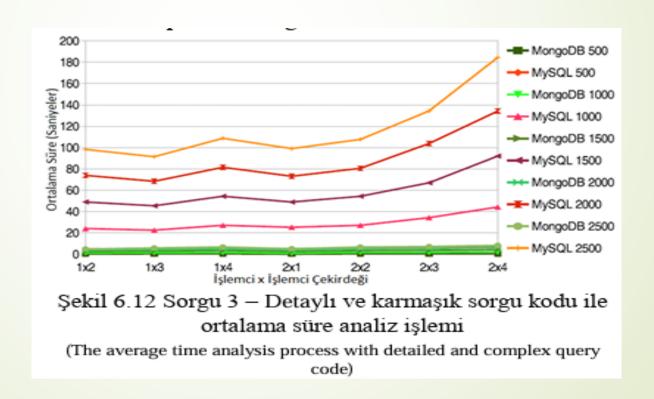
Şekil 6.10'da iç içe geçmiş "SELECT" ve "WHERE" işlemlerini içeren üçüncü sorgu neticesinde ortaya çıkan performans değerleri gösterilmektedir. Yapılan analizlerde; MySQL veri tabanı sisteminin MongoDB 'ye göre sorgu süresi sonuçları, veri kayıt sayısı farkı arttıkça iyi bir performans göstermiştir.



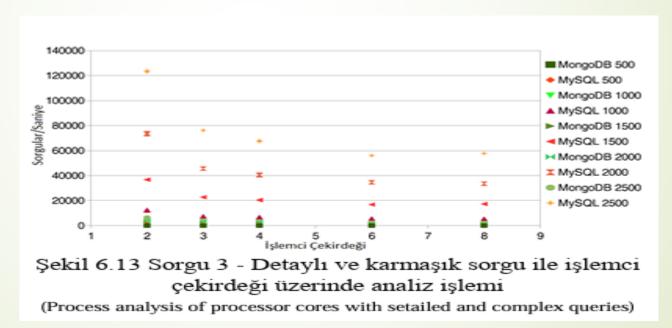
MySQL ve MongoDB veri tabanlarına üçüncü sorgu kullanılarak uygulanan karşılaştırma testi sonuçları Şekil 6.11'de gösterilmiştir. MySQL veri tabanı sisteminin 2x4 işlemci ve işlemci çekirdeği yapılandırmasında en iyi performansı gösterdiği görülmektedir.



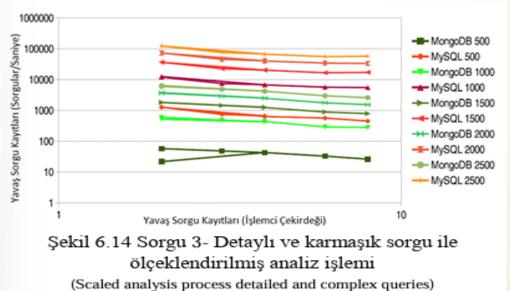
Şekil 6.12'de üçüncü sorgu ile ortalama süre ölçümleri gösterilmiştir. Yapılan analizde; MySQL veri tabanı sisteminin MongoDB'ye göre ortalama sorgu süreleri sonuçları, veri kayıt sayısı farkı arttıkça oldukça belirgin bir performans kötülüğü gösterdiği gözlemlenmiştir. MongoDB daha istikrarlı performans sergilemiştir.



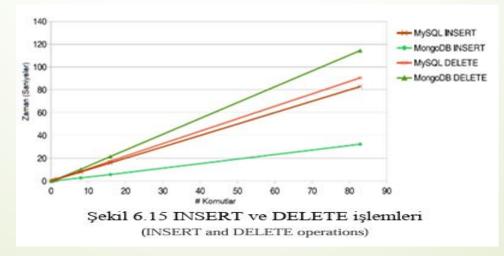
MySQL ve MongoDB veri tabanlarına üçüncü sorgu olarak tanımlanan detaylı ve karmaşık sorgu kodu içeren karşılaştırma testi analizi Şekil 6.13'de gösterilmiştir. MySQL veri tabanı sistemi, iki eksen boyunca logaritma kullanılarak çizilen grafikte logaritmik bir eğilim olduğu görüntüsü sergilemektedir.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



Zamanlama ölçeği büyütülerek veri tabanları sistemleri arasındaki performans farkı anlaşılır bir şekilde Şekil 6.14'te gösterilmiştir. Ölçeğin büyümesi MySQL'in performansı için dezavantajlı olduğu görülmektedir. MongoDB'nin tüm veri kayıt setlerinde oldukça iyi bir performans gösterdiği görülmektedir.



- Son olarak MySQL ve MongoDB veri tabanlarına veri ekleme "INSERT" ve silme "DELETE" işlemleri uygulanmıştır. Şekil 6.15'de her iki veri tabanı sisteminin INSERT ve DELETE işlemlerine ait performans grafiği gösterilmektedir.
- Yapılan analizde MongoDB'nin veri ekleme işlemi MySQL'e göre çok daha iyi bir performansa sahip olduğu görülmüştür.
- 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.



## SONUÇ VE DEĞERLENDİRME (RESULT AND EVALUATION)

- Dağıtık veri tabanları ile ilişkisel veri tabanları karşılaştırılmış ve yönetim bilişim sistemleri açısından incelenmiştir. Bu çalışmada veri tabanlarının modellenmesi ve veri tabanı seçiminde kullanıcılara ışık tutulması hedeflenmiştir. NoSQL kavramı incelenmiştir.
- Yapılan analizlerde NoSQL ağırlıklı bir veri tabanının büyük miktarda veri çiftleri içerebildiği, veri çoğaltmada da basit şeması nedeniyle MongoDB kullanılarak daha hızlı daha karmaşık sorgu tiplerinin çalıştırılabildiği izlenmiştir. Her iki veri tabanı sisteminde farklı yapılandırma durumlarında ikinci sorgu tipi ile yapılan performans testlerinde, MongoDB veri tabanı sistemi MySQL'e göre en iyi performansı göstermiştir. MongoDB, alt belge koleksiyonu kullanımı nedeniyle MySQL üzerinde çok büyük bir avantaja sahip olduğunu gösterilmiştir. Yapılan yazma ve silme işlemlerinde MySQL iyi bir performans sergilemiştir. MongoDB ve MySQl karşılaştırmasında mongoDB daha iyi performans sergilemiştir.

Sonuç olarak her iki veri tabanın avantaj ve dezavantajları vardır. İlişkisel veri tabanı yönetim sistemlerinin kullanıldığı uygulamalar ilişkisel olmayan sistemlere taşınmada veri kaybı söz konusu olabilir, bu da dezavantajlı olmaktadır. Diğer yandan hız, geliştirme zamanı ve ölçeklenebilirlik gibi özellikler ilişkisel olmayan veri tabanlarının kullanılması avantajlı olacaktır.