VERİ ORGANİZASYONU

CEREN SUDE FIRAT

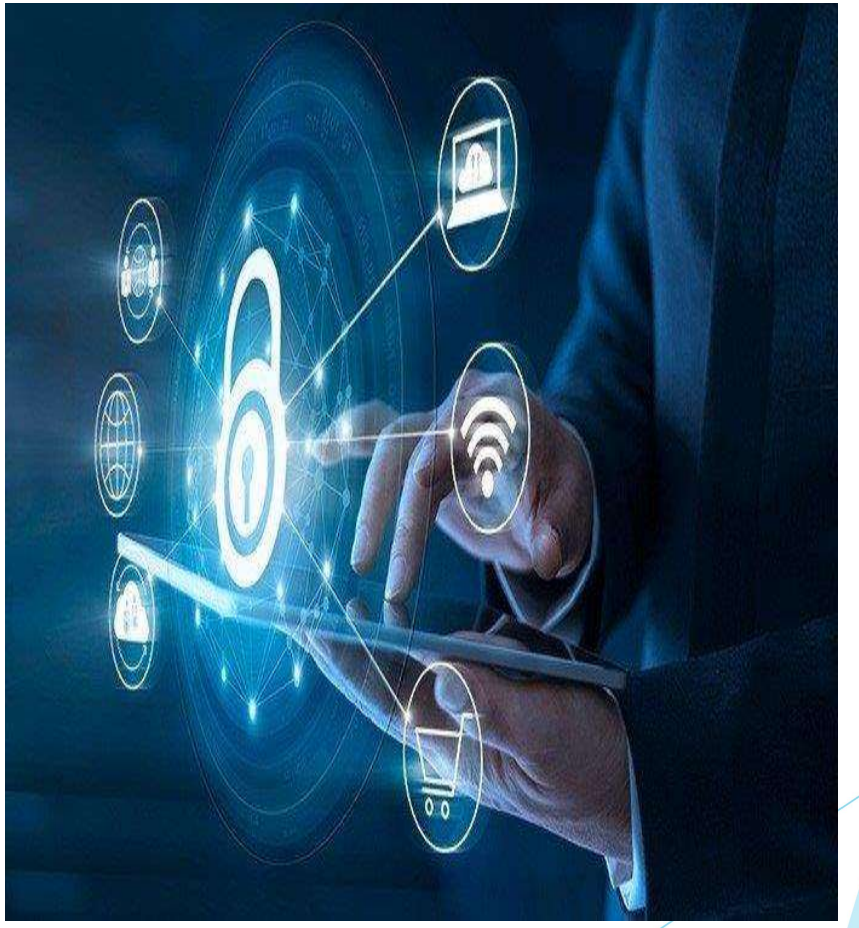
02220224007

VERİ TABANI NEDİR?

- Veri tabanları, veri fazlalığını kontrol eden ve veri tutarlılığını koruyan en önemli sistemlerden biridir. Veri entegrasyonu ile verilere basit şekilde ulaşılması, düzenlenmesi ve paylaşılması gibi avantajlar sağlamaktadır.

BİLİŞİM SİSTEMİ NEDİR?

Bilişim sistemi, organizasyonlarda karar verme aşamasına kadar bilgiyi toplamak, düzenlemek, işlemek ve saklamak olarak tanımlanabilir. Bilişim sistemleri, bilişim teknolojileri altyapısından yararlanan yönetsel çözümlerdir.



BİLİŞİM SİSTEMİ NEDİR?

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

VERİ TABANI SİSTEMLERİ NEDİR?

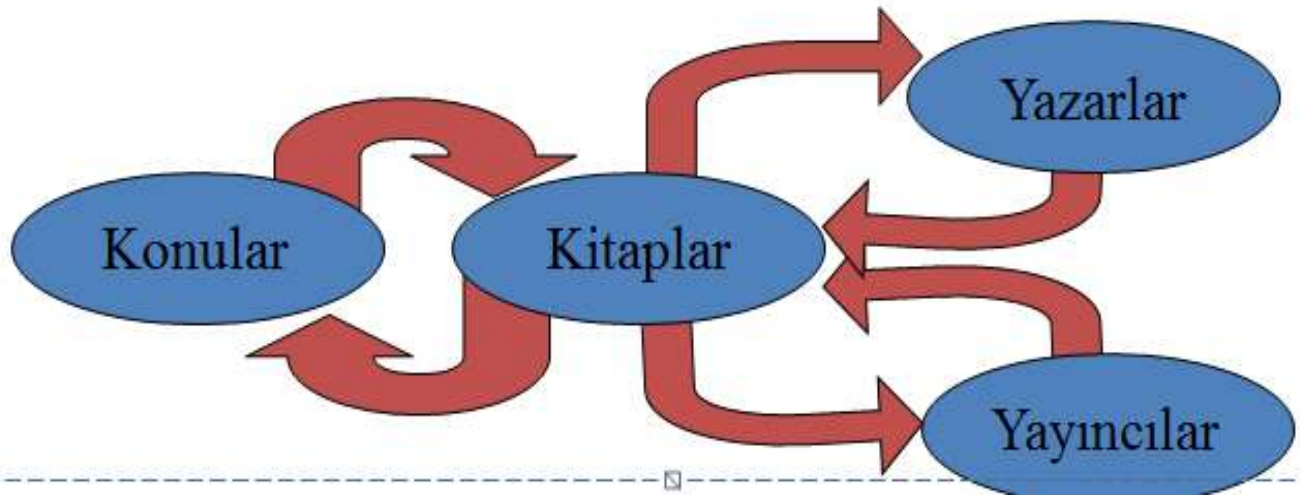
- Yeni bir veri tabanı oluşturmak, oluşturulan veri tabanlarını düzenlemek, yönetmek, geliştirmek, belirli amaçlar için kullanmak ve bu veri tabanlarının bakımlarını yapmak için kullanılan yazılıma veri tabanı yönetim sistemi denir. Veri tabanı, VTYS ve uygulama programlarını ile kullanıcı ara yüzlerini içeren yapıya “veri tabanı sistemi ” denir.

ÖĞRENCİ NO	ADI	SOYADI	BÖLÜM
1	AHMET	ŞEN	BÖLÜM1
2	AYŞE	MUTLU	BÖLÜM2
3	AKİF	İNCE	BÖLÜM2
4	ALİ	ÇELİK	BÖLÜM2
5	AYÇA	YILMAZ	BÖLÜM3

BÖLÜM	DANIŞMAN
BÖLÜM1	DANIŞMAN1
BÖLÜM2	DANIŞMAN2
BÖLÜM2	DANIŞMAN3

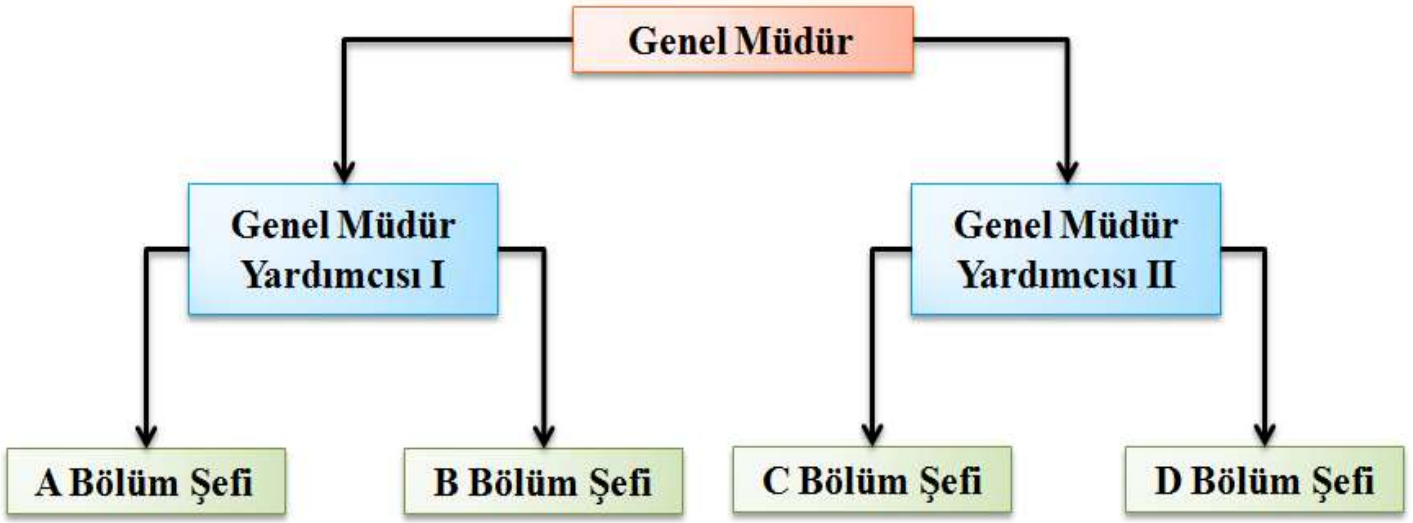
Düz model veya tablo modeli:

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.



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

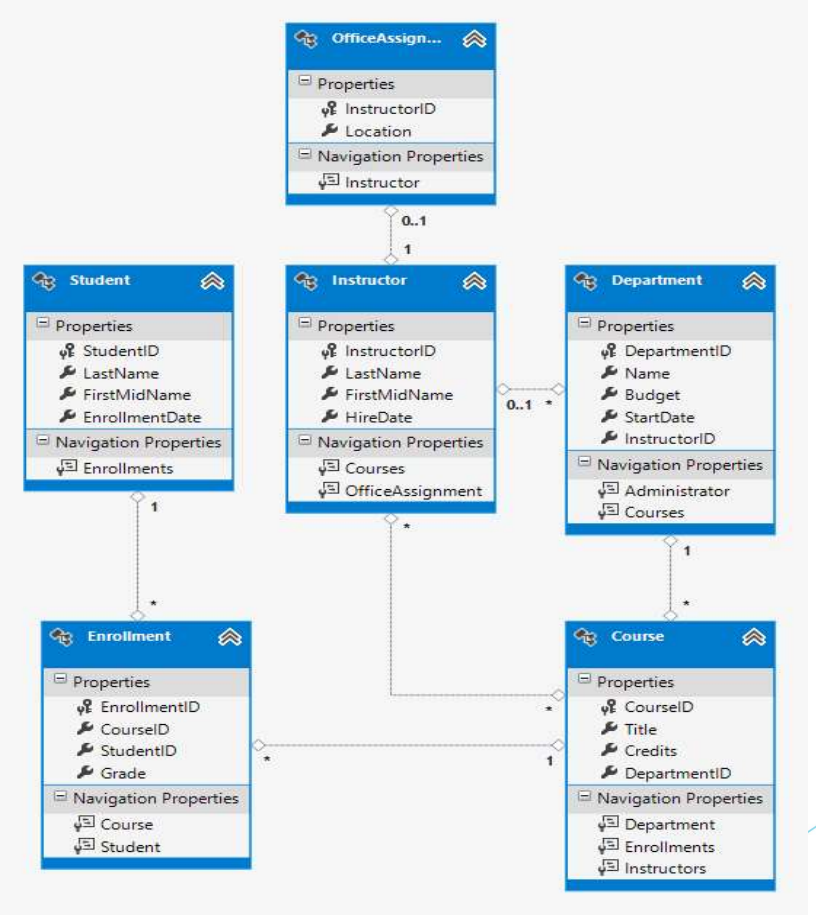


Ağ veri modeli:

Hiyerarşik veri modelinin geliştirilmiş halidir.Ağ modelinin hiyerarşik modelden en önemli farkı, uç düğüm pozisyonundaki verinin iç-düğüme işaret edebilmesidir. Bu veri tekrarını önemli ölçüde azaltır

İlişkisel Veri Modeli:

İlişkiler yardımıyla, veri içerisindeki ilişkiler modellenir. Kavramsal olarak ilişkiler, satır ve sütunlardan oluşan iki boyutlu tablolarla karakterize edilir.veri tabanında her tablo için bir dosya bulunur. Tablonun her satırı birbiriyle ilişkili verilerin bir topluluğudur. Sütunlarda ise nitelikler bulunur

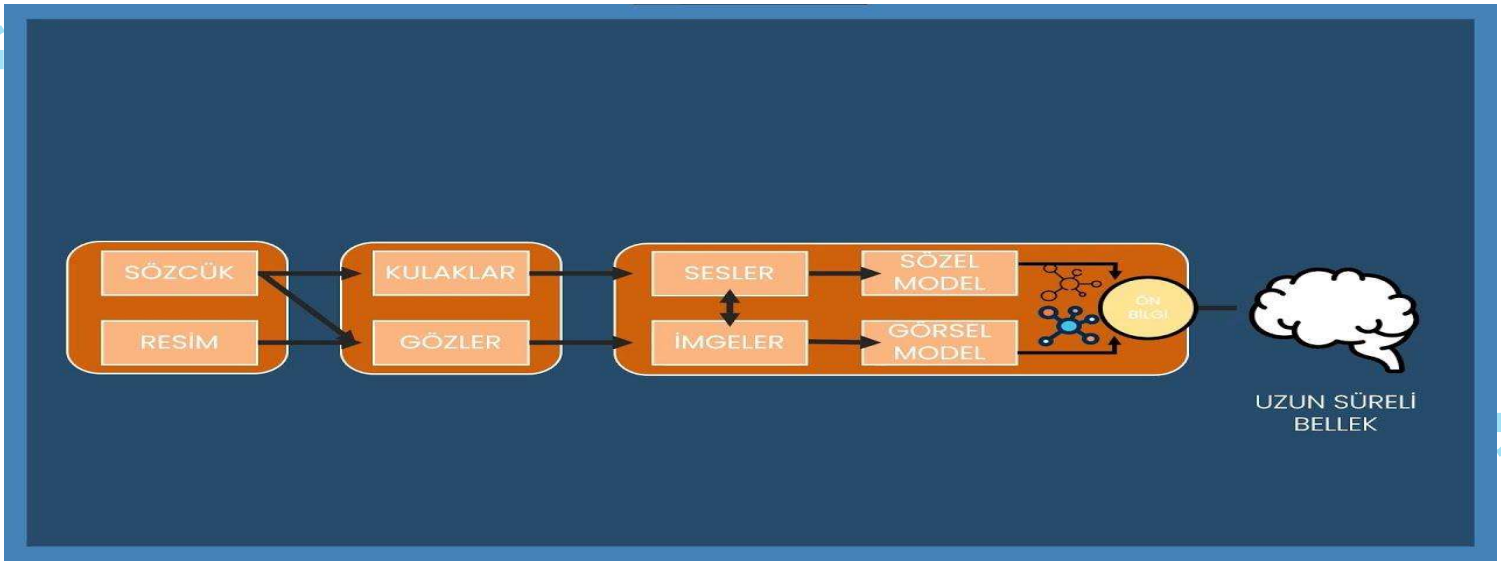


Nesne Yönelimli Veri Modeli:

- Nesne yönelimli programlamaya dayanan veri modelidir.

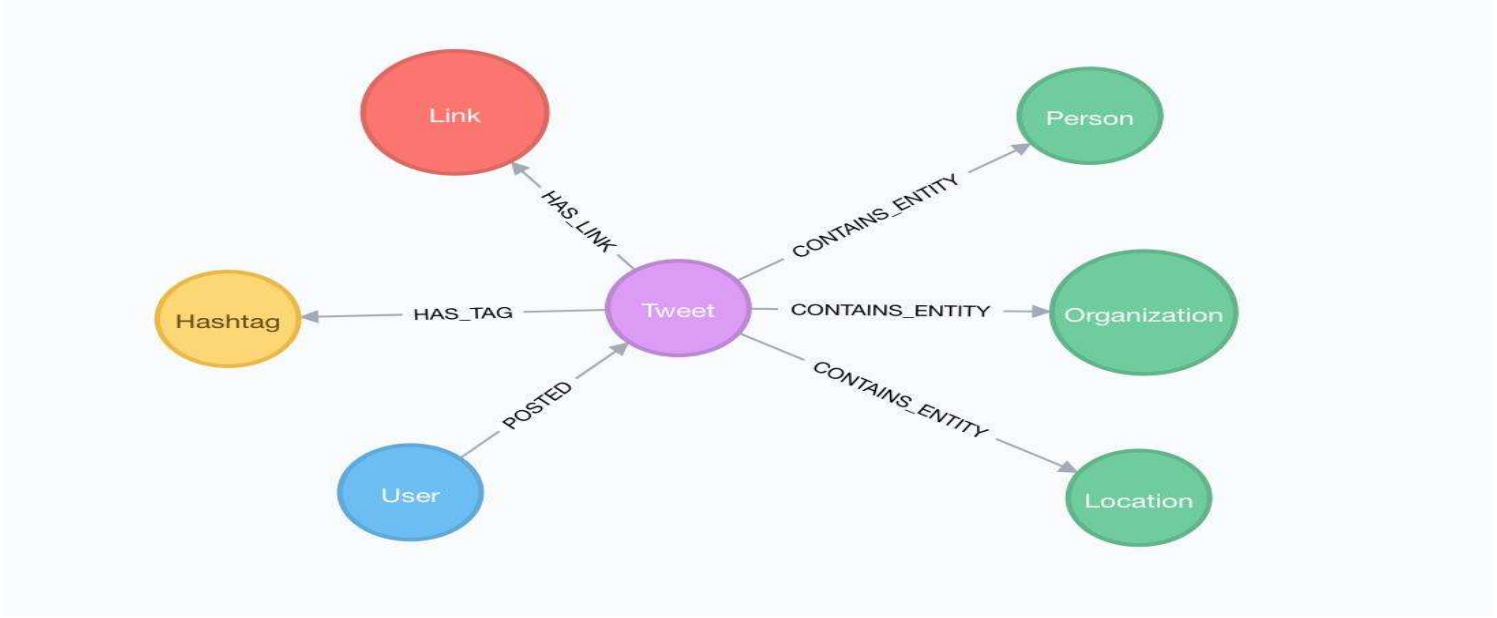
Nesne İlişkisel Veri Modeli:

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



Çoklu Ortam Veri Modeli:

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.



Dağıtık Veri Modeli:

iki ya da daha fazla bilgisayarda depolanan ve bir ağ üzerinde dağıtılan bilgiler için kullanılan veri tabanı grubudur.Birden fazla veri tabanına erişilmesine rağmen, kullanıcı bir tek veri tabanıyla çalışıyormuş gibi işlem yapar.

VERİ TABANI TASARIMI

- 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 anlaşılabilecek bir tarzda tanımlanması gerekir. Buna "Şema" denir.Kullanıcı ve bilgisayar düzeyleri sırasıyla "kavramsal" ve "fiziksel" düzeyler, bu düzeylerdeki şemalar da “kavramsal şema” ve “iç şema” olarak anılırlar.

VERİ TABANI TASARIMI

- 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. Kavramsal şema, fiziksel depolama yapılarının ayrıntılarına girmeden, varlıklar, veri tipleri, varlıklar arasındaki ilişki tipleri ve kısıtlayıcılar üzerinde yoğunlaşır. kavramsal şema, yazılım ve donanımdan bağımsızdır ve son kullanıcı tarafından anlaşılması da daha kolaydır.
- geleneksel veri tabanı tasarımında, kavramsal tasarımdan sonraki adım, çoğunlukla, gerçekleştirim için kullanılacak bir veri tabanı yönetim sisteminin seçimidir.O nedenle mantıksal veri modelleri, gerçekleştirim veri modelleri olarak da bilinirler.
- İç şema depolama yapılarını, kayıt formatlarını, kayıt alanlarını, veri tabanına giriş yol ve yöntemleri ile veri tabanının fiziksel gerçekleştirimini ilgilendiren diğer bütün detayları tanımlar.

İLİŞKİSEL VE İLİŞKİSEL OLMAYAN (NoSQL) VERİ TABANI SİSTEMLERİ

İlişkisel Veri Tabanı

- Satır ve sütunların meydana getirdiği tablolardan oluşur. Bu tablolar birbiri ile ilişkileri olan tablolardır. Veri tabanı denilen büyük dosyalardan 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

- NoSQL, ilişkisel veri tabanı sistemlerine alternatif bir çözüm olarak ortaya çıkmıştır. İlişkisel olamayan veri tabanları yatay olarak ölçeklendirilen bir veri depolama sistemidir.
- Veri tabanlarına ilişkin problemlerden biri olan ölçek sorununa, diğer çözümlerin içinde en iyi cevap vereni NoSQL'dir.
- Çok büyük verilerin depolanması ve yazılmasında ilişkisel veri tabanlarının eksik kaldığı hususlarda, yatay ölçekleme yapan dağıtık NoSQL çözümleri geliştirilmiştir.

- İlişkisel veri tabanını yerine NoSQL veri tabanını tercihi, özellikle hız ve yatay büyüme ile gereksiz ek maliyetten kurtulmaya dayanmaktadır.

- NoSQL "BASE" (Basically Available-Soft stateEventually consistent) kısaltması ile ifade edilir.

- Kolay Ulaşılabilirlik (Basically Available): Veri erişim sorunlarını ortadan kaldırmak için kopyaları kullanır ve paylaşılmış ya da bölümlenmiş veriyi birçok sunucudan alır.
- Esnek Durum (Soft state): NoSQL sistemler tutarsız ve süreksiz verilerin barınmasına da izin verir.
- Eninde sonunda Tutarlı (Eventually consistent): ACID'in zorunlu tuttuğu tutarlılığa karşın NoSQL'de tanımlanmayan bir zamanda tutarlılığın oluşacağı garanti edilir

VERİ TABANI MİMARİLERİNİN PERFORMANS KARŞILAŞTIRMASI

- Veri tabanı mimarilerinde çok fazla seçenek vardır.Bu çalışmada MySQL ve NoSQL veri tabanı sistemlerine alternatif ortaya çıkan MongoDB veri tabanı sistemi kullanılmıştır.
- Çalışmada MySQL ve MongoDB veri tanı sistemlerinin performans ve yatay ölçeklenebilirlik incelenmesi belirli işlemlerle belirlenmeye çalışılmıştır.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.

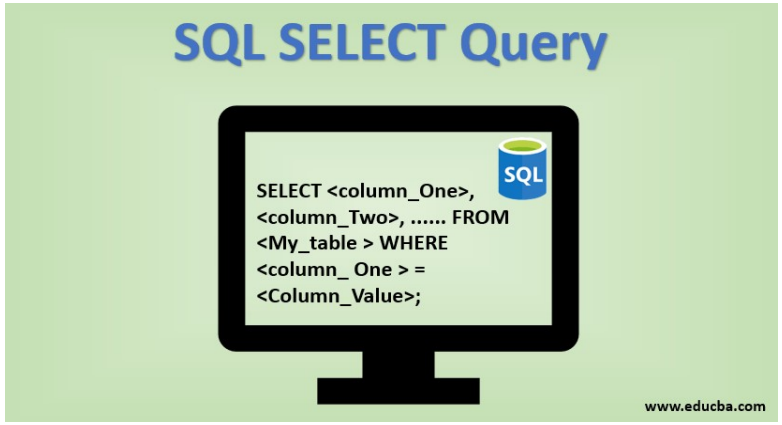
- Veri Tabanı Şeması:Çalışmada MySQL ve MongoDB olan 2 veri tabanı şeması tasarlanmıştır.Şemalar müzik uygulaması etrafında modellenmiştir.
- Şemalar arasında veri tekrarını ortadan kaldırmak için normalizasyon değerlendirmesi sağlanmıştır.

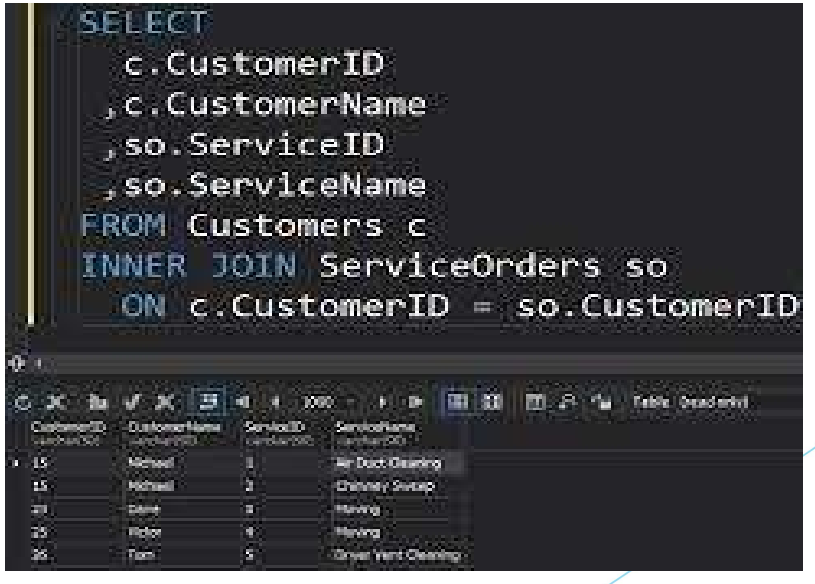
```
MariaDB [(none)]> show variables like '%version%';
+---------------------------------+-----------------------------------------+
| Variable_name                   | Value                                   |
+---------------------------------+-----------------------------------------+
| in_predicate_conversion_threshold | 1000                                  |
| innodb_version                  | 10.4.12                                 |
| protocol_version                | 10                                      |
| slave_type_conversions          |                                         |
| system_versioning_alter_history | ERROR                                   |
| system_versioning_asof          | DEFAULT                                 |
| tls_version                     | TLSv1.1,TLSv1.2,TLSv1.3                 |
| version                         | 10.4.12-MariaDB                         |
| version_comment                 | MariaDB Server                          |
| version_compile_machine         | x86_64                                  |
| version_compile_os              | Linux                                   |
| version_malloc_library          | system                                  |
| version_source_revision         | ba6bfc402c352372cc1a9ec20b5dc50b2204549f |
| version_ssl_library             | OpenSSL 1.1.1c FIPS  28 May 2019        |
| wsrep_patch_version             | wsrep_26.22                             |
+---------------------------------+-----------------------------------------+
15 rows in set (0.009 sec)

MariaDB [(none)]>
```

Veri Tabanı Sorguları:

- Çalışmada 3 farklı veri tabanı sorgusu kullanılmıştır.1.sorgu sadece "SELECT" deyimi içerir.2. sorgu daha karmaşık olan "INNER JOIN" deyimi içerir. 3. sorgu "SELECT" ile birlikte "JOIN","INNER JOIN" ve "WHERE" deyimi içerir.





Ölçümler:

- Çalışmada zaman kavramı ön planda tutulması hedeflenmiştir. 3 yöntem kullanılmıştır.
- 1. Clock() fonksiyonu kullanımı ile belirli bir süre CPU üzerinde harcanan zaman sonuçlarının elde edilmesini sağlamaktır.
- 2. Milisaniye hassasiyetiyle zamanlamaları sağlayan Gettimeofday() fonksiyonu kullanılarak sonuçların elde edilmesini sağlamaktır.
- 3.Slow Query Log (Yavaş sorgu kaydı) olarak adlandırılmaktadır.

Ölçüm Metrikleri:

- Bir uygulama için en önemli faktör, bir görevi tamamlamak için gereken süre ve veri tabanının bir işlemi tamamlaması durumu için gerekli zamandır.
- Saniyedeki sorguyu hesaplamak için toplum sorgu sayısını toplam iş parçacığı sayısı ile çarpıp ortalama sorgu süresine bölmeliyiz.
- İş parçacığı başına saniyedeki sorguyu hesaplamak için toplam sorgu sayısı ortalama sorgu süresine bölünmelidir.

Analiz ve sonuçlar:

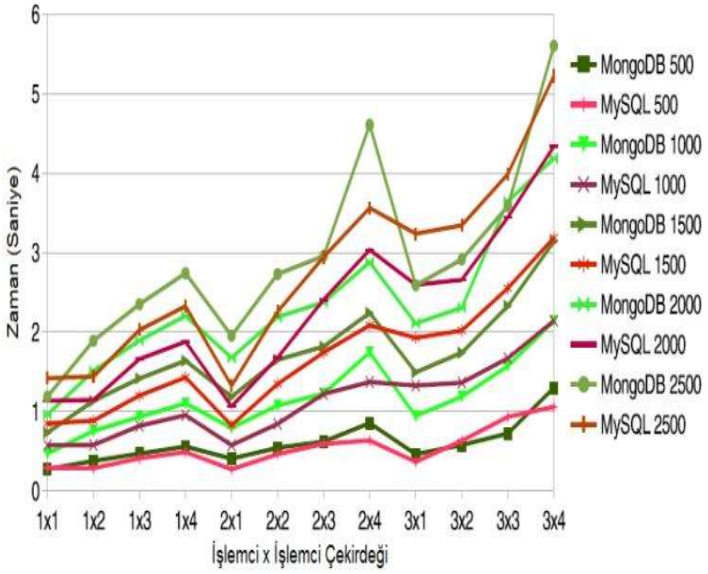
- veri tabanlarının farklı sorgu türlerine göre nasıl yanıt verdiği hem okuma hem yazma ile analiz edilen sorguların toplam sayısı ve sonuçları şekillerle gösterilmiştir.

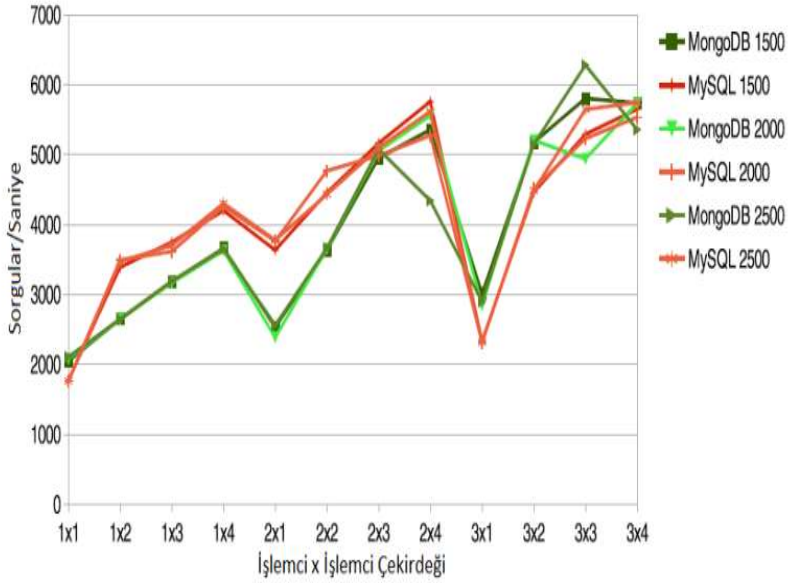
- Bu çalışmada MySQL ve MongoDB veri tabanları sistemlerinin her ikisine eşdeğer miktarda sorgular yapılmaktadır.

SONUÇLAR:

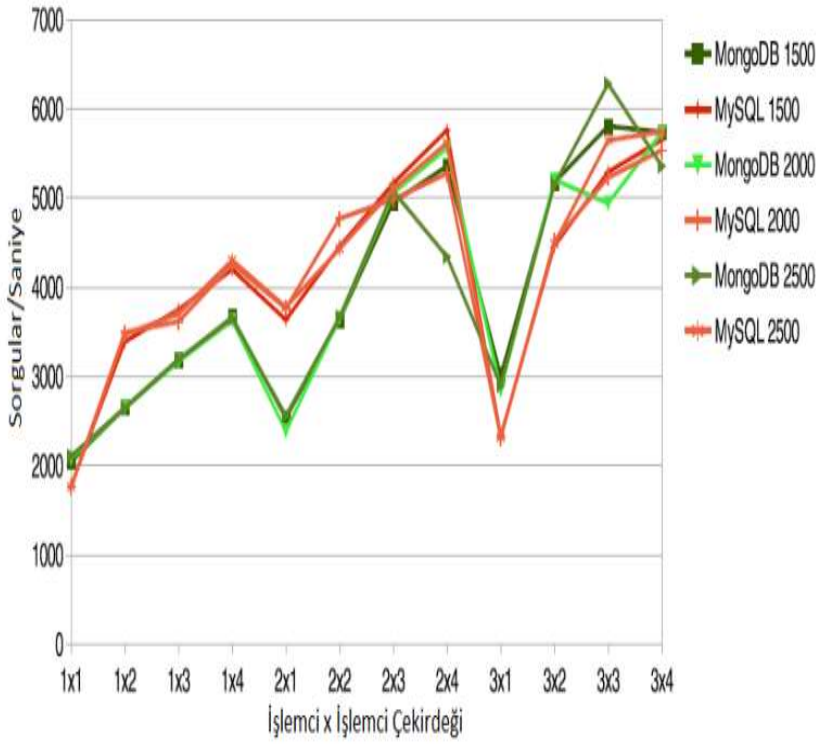
- MongoDB, sorgu sayısı farkı arttıkça daha belirgin bir performans kötülüğü gösterdiği tespit edilmiştir.

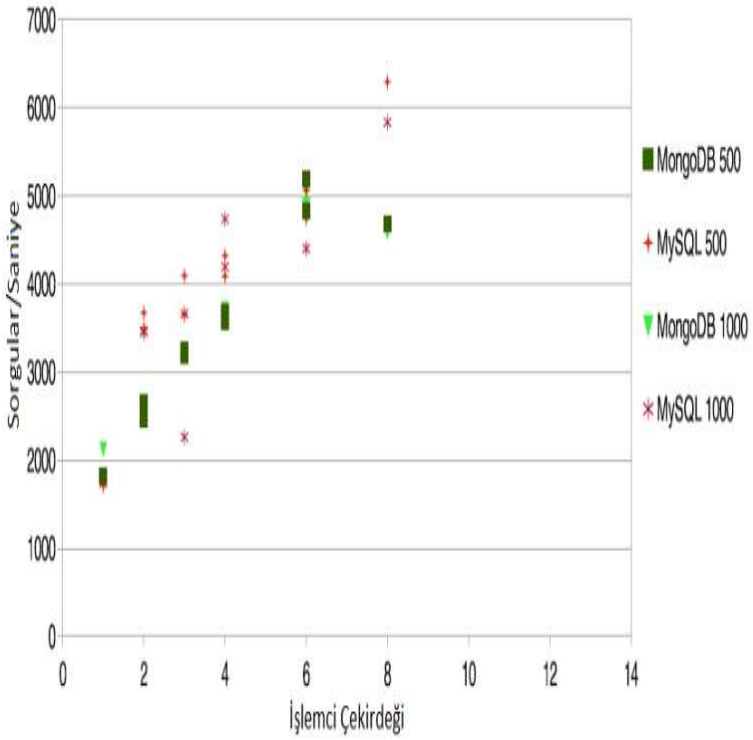
Ayrıca sorgular/saniye ölçüm metrik grafiği ile ortalama süre sonuçları elde edilmiştir.



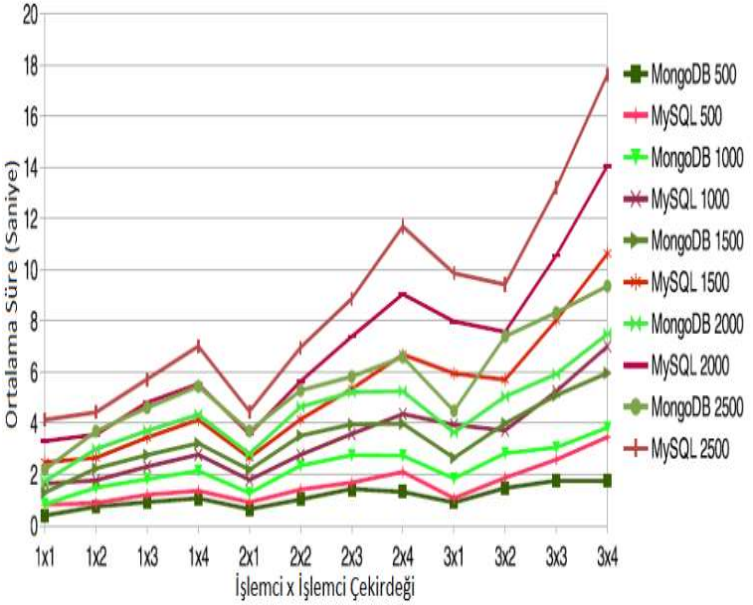


MySQL veri tabanı sisteminin, sorgu sayıları arttığında MongoDB üzerinde avantaj sahibi olduğu görülmektedir.İşlemci sayısı 2 ve 3e gelince grafikte azalma görülmüş.MongoDB bu yapılandırmalarda daha fazla avantaj göstermiş.

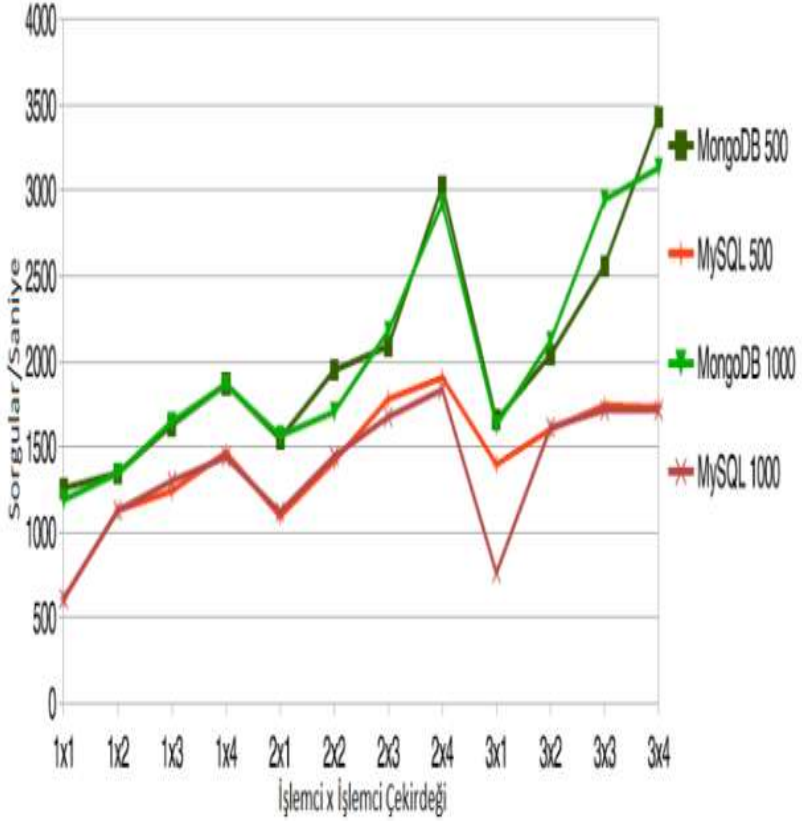




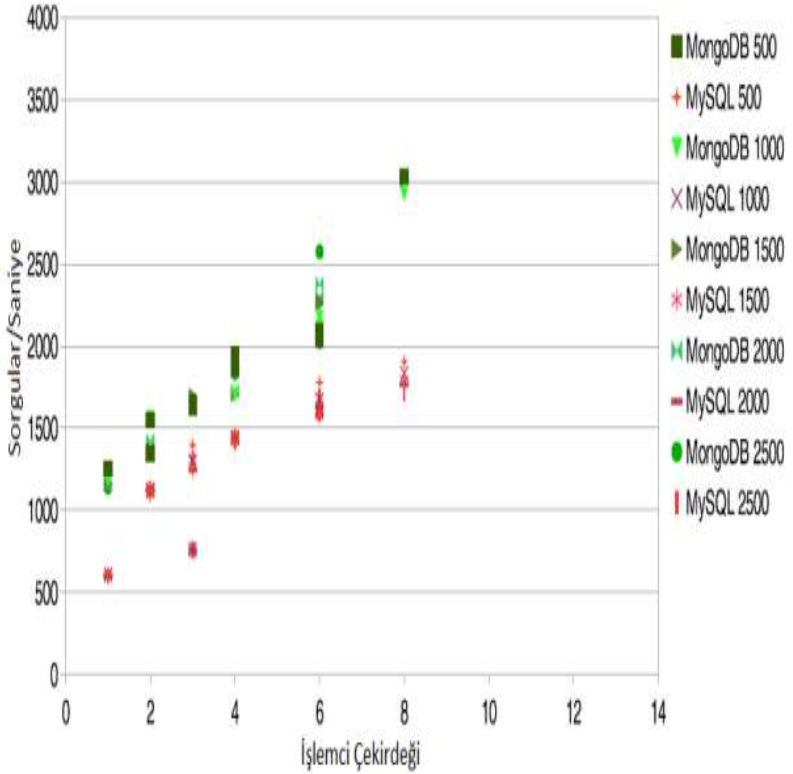
Sorgular/Saniye ile işlemci çekirdeği miktarı için analiz işlemi



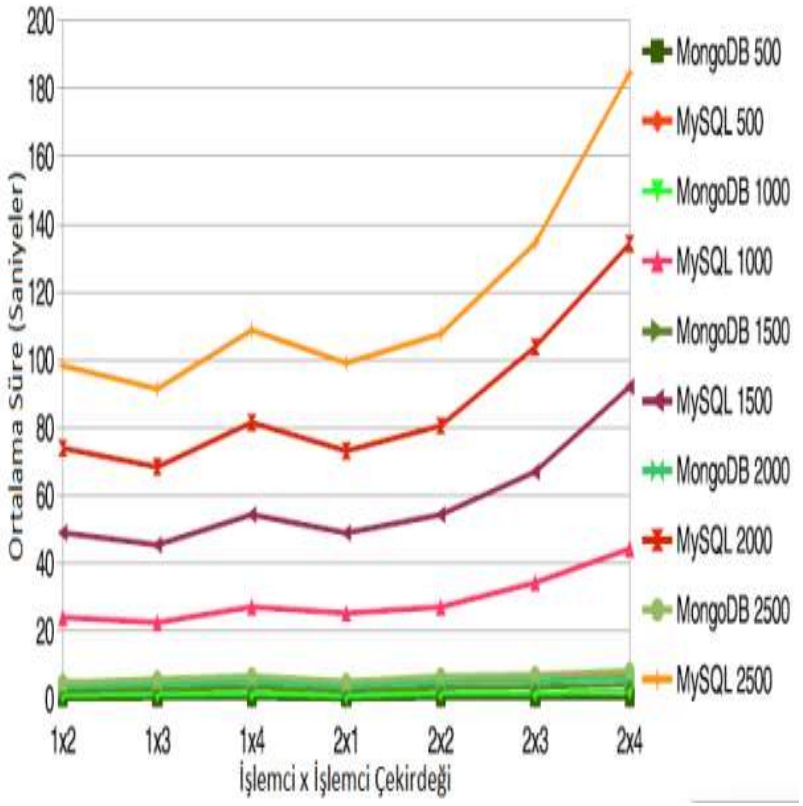
INNER JOIN ile karmaşık sorgu analizi işlemi



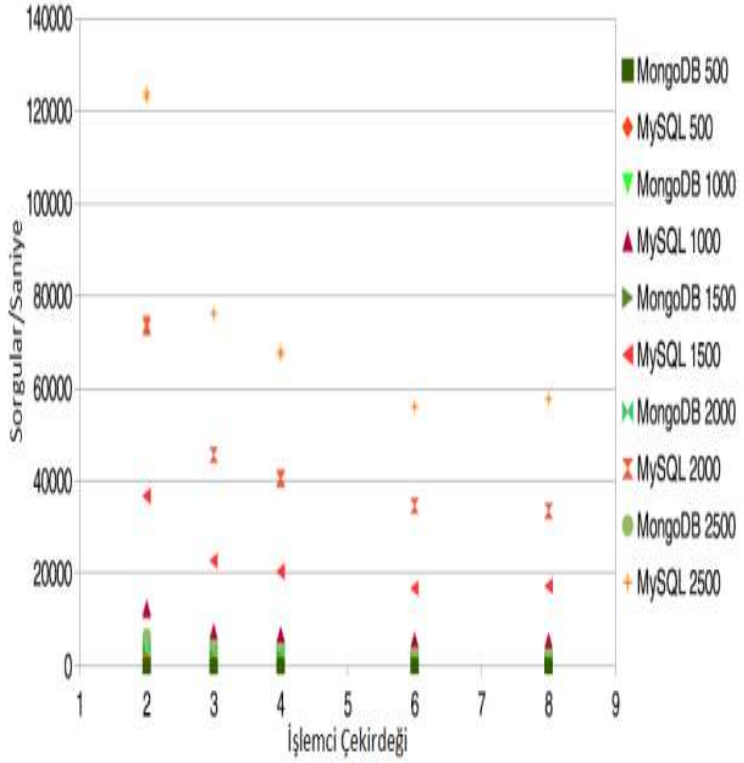
INNER JOIN ile 500 ve 1000 veri için sorgu/saniye analizi işlemi



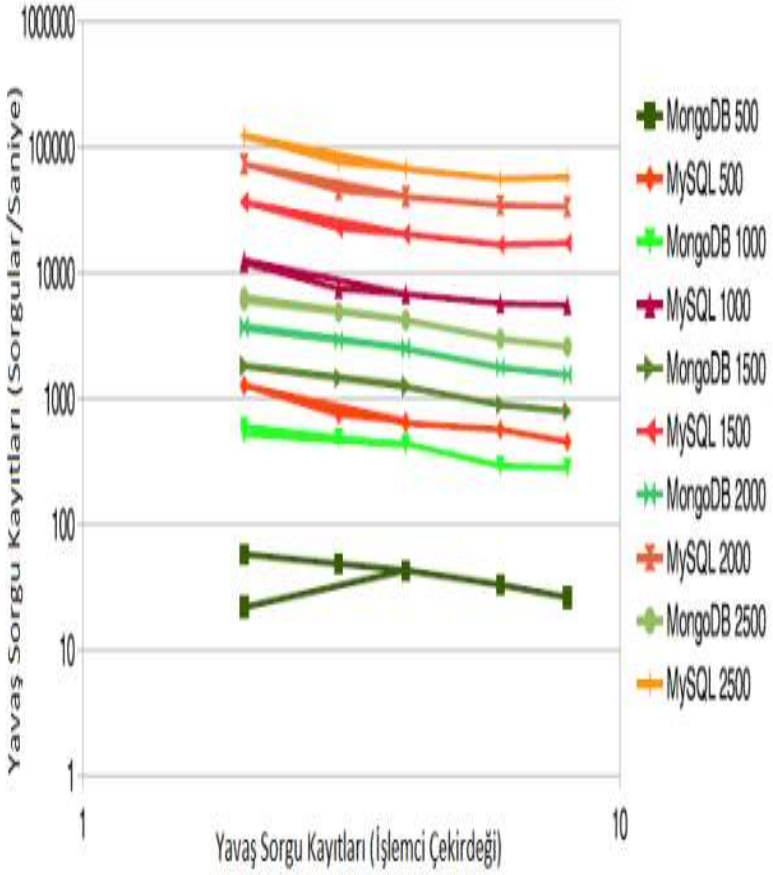
Detaylı ve karmaşık sorgu ile Sorgular/saniye analiz işlemi



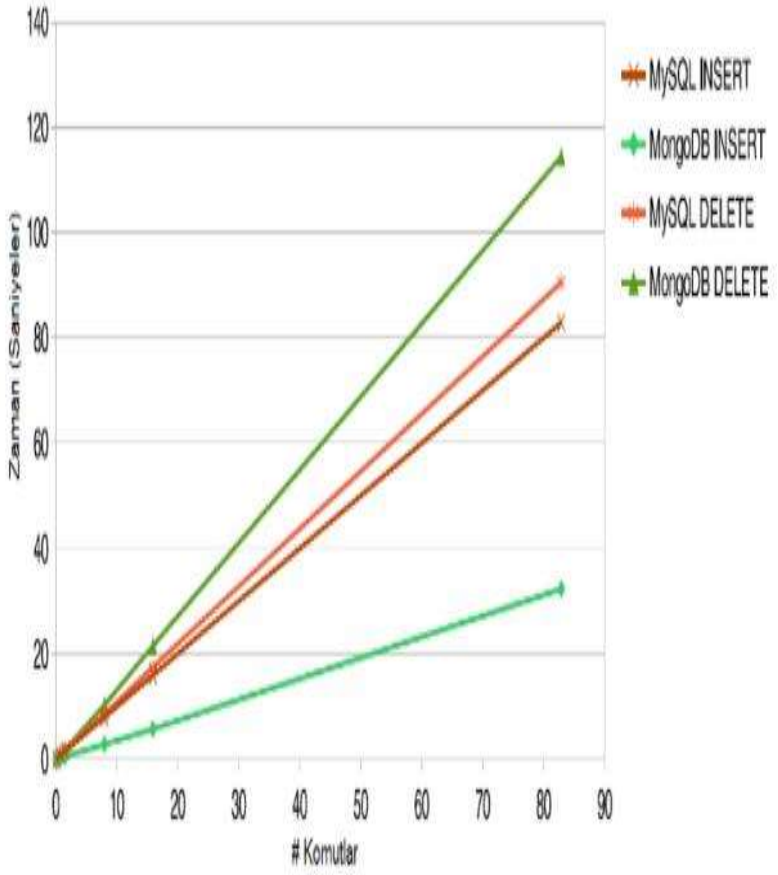
Detaylı ve karmaşık sorgu kodu ile ortalama süre analiz işlemi



Detaylı ve karmaşık sorgu ile işlemci çekirdeği üzerinde analiz işlemi



Detaylı ve karmaşık sorgu ile ölçeklendirilmiş analiz işlemi



INSERT ve DELETE işlemleri