

İLKER KALECİK

***İlişkisel ve NoSQL
Veritabanı Sistemlerinin
Performans Mimarisinin
Yönetim Bilgi Sistemleri
İçinde İncelenmesi***

**VERİ
ORGANİZASYONU**

Konu Başlıkları

01

- Giriş

02

- Bilişim Sistemleri ve yönetimi

03

- Veri tabanı ve veri tabanı yönetim sistemleri

04

- Veri tabanı tasarımı

05

- İlişkisel ve NoSQL veri tabanı sistemleri

06

- Veri tabanı mimarilerinin performans karşılaştırması

01

Giriş

Giriş



- **Bilgi Sistemleri Nedir?**

Bilgi sistemleri, bir işletmenin bilgi ihtiyaçlarını karşılamak için kullanılan teknolojik sistemlerdir. Bunlar, bilgiyi toplar, depolar, işler, ileter ve kullanıcıların ihtiyaçlarına göre sunar.



Günümüzde artan veri sayısı, artan veri karmaşıklığı nedeniyle verinin saklanma yöntemi değişmiştir. Bu karmaşıklığı önlemek adına bazı yöntemler geliştirilmiştir. İlişkisel olmayan veritabanı sistemleri buna örnektir. Dünyaca ünlü dev şirketler de bu yöntemi kullanmaktadır.

02

Bilişim sistemleri ve yöntemi



Bilişim Sistemleri

Bilişim Sistemi bilgiyi işlemek yönetmek organize etmek ve saklamak olarak tanımlanabilir.

Giriş

- Bilgi sistemlerinin
- Bölümleri:



Girdi

Girdi, organizasyonun içinden veya dış çevresinden hem bilgileri toplamaktır



İşlem

İşlem, toplanan ham veriyi daha anlamlı biçimde işleyerek bilgiye dönüştürür.

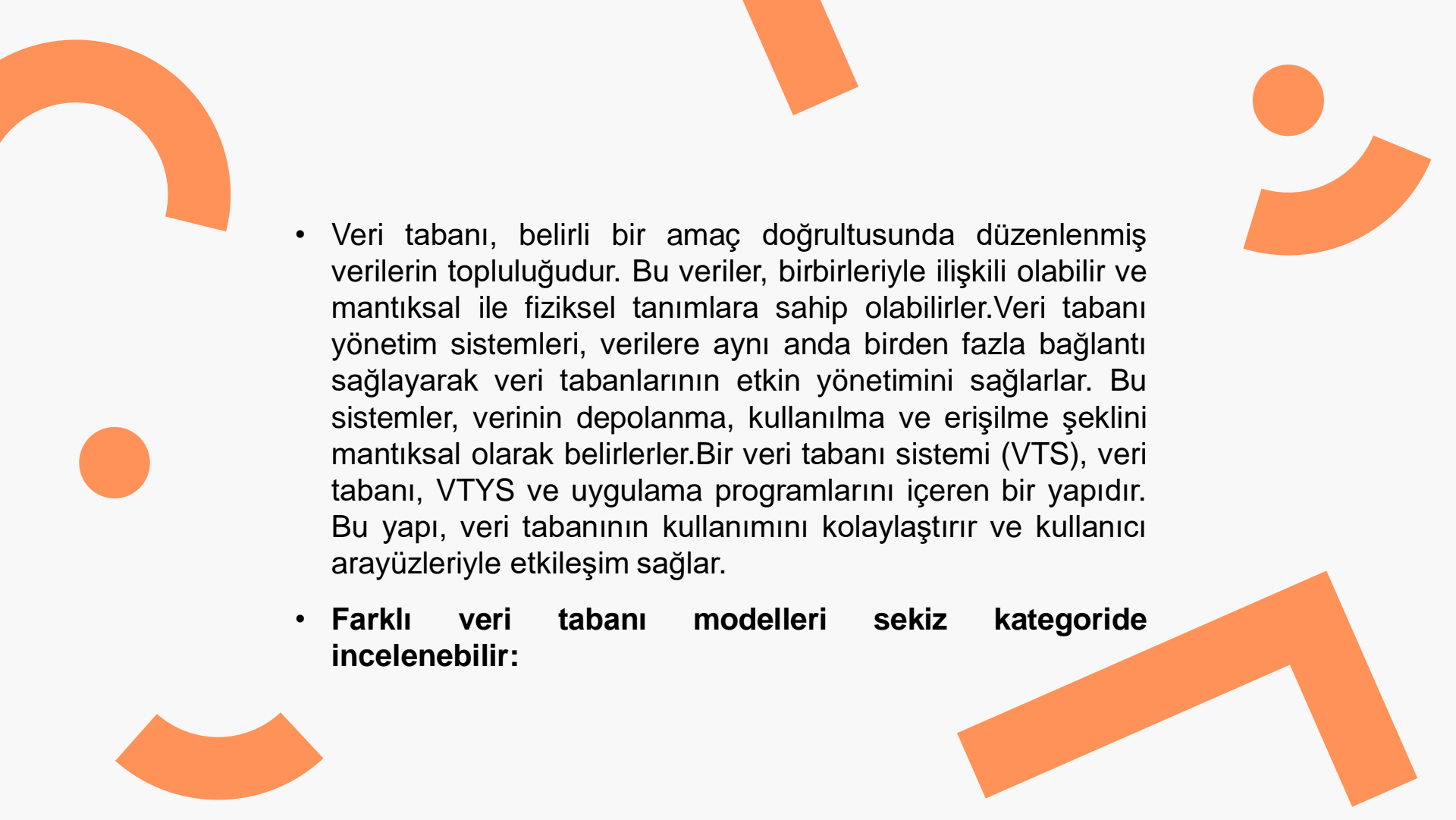


Çıktı

Çıktı, işlenmiş bilgiyi (enformasyonu) insanlara veya kullanılacak olan aktivitelere aktarır.

03

Veri tabanı ve veri tabanı yönetim sistemi

- 
- Veri tabanı, belirli bir amaç doğrultusunda düzenlenmiş verilerin topluluğudur. Bu veriler, birbirleriyle ilişkili olabilir ve mantıksal ile fiziksel tanımlara sahip olabilirler. Veri tabanı yönetim sistemleri, verilere aynı anda birden fazla bağlantı sağlayarak veri tabanlarının etkin yönetimini sağlarlar. Bu sistemler, verinin depolanma, kullanılma ve erişilme şeklini mantıksal olarak belirlerler. Bir veri tabanı sistemi (VTS), veri tabanı, VTS ve uygulama programlarını içeren bir yapıdır. Bu yapı, veri tabanının kullanımını kolaylaştırır ve kullanıcı arayüzleriyle etkileşim sağlar.
 - **Farklı veri tabanı modelleri sekiz kategoride incelenebilir:**

Düz Model Veya Tablo Modeli

İki boyutlu veri grubundan oluşur ve benzer özelliklere sahip veriler sütunlarda, veri grupları ise satırlarda yer alır.

Hiyerarşik veri modeli

Veriler ağaç yapısı şeklinde düzenlenir, her kayıt bir veya daha fazla çocuk kaydına sahiptir.

- Ağ Veri Modeli

- Doğal olarak ilişkili verileri modeller ve çok-açok ilişkileri destekler.

İlişkisel Veri Modeli

- İlişkilerin temel alındığı bir modeldir ve veriler ilişkisel tablolar aracılığıyla temsil edilir.
- Nesne Yönelimli Veri Modeli
- Nesne yönelimli programlamaya dayanan bir veri modelidir.

- Nesne İlişkisel Veri Modeli

- Nesne yönelimli ve ilişkisel özellikleri bir araya getirir.

- Çoklu Ortam Veri Modeli

- Büyük medya dosyalarını işleyebilen bir modeldir ve kullanıcıya adımları göstermez.

- Dağıtık Veri Modeli

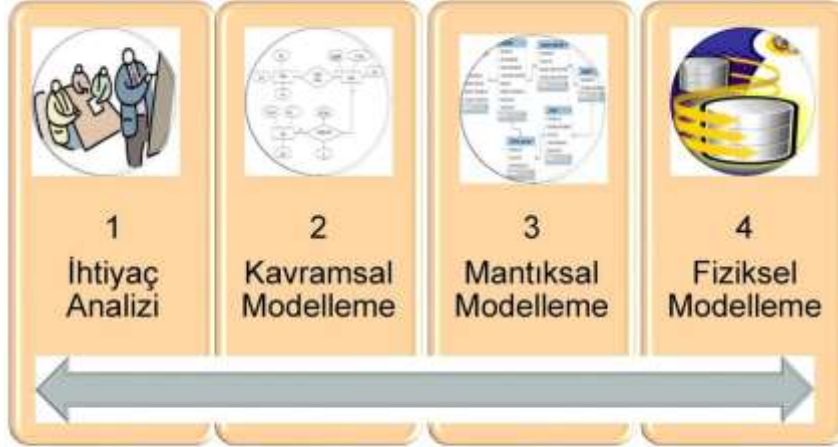
- Verilerin iki veya daha fazla bilgisayar üzerinde dağıtıldığı bir modeldir ve paralel kullanım sağlar.

04

Veri Tabanı Tasarımı

- Veri tabanı tasarımı, gerçek dünyadaki ihtiyaçları ve beklentileri göz önünde bulundurarak yapılır. İlk adım, hangi bilgilerin saklanacağını, nasıl saklanacağını ve nasıl kullanılacağını belirlemektir. Bu, kullanıcıların ve bilgisayarların anlayabileceği bir modelle yapılır.
- Veri tabanı tasarımı, farklı seviyelerde yapılır. Kullanıcılar için genel bir yapı belirlenirken, daha teknik detaylar fiziksel düzeyde ele alınır. Bu, veri tabanının genel yapısını ve içinde neyin bulunacağını tanımlar.
- Geleneksel olarak, kullanıcı gereksinimlerine dayanarak bir kavramsal model oluşturulur. Bu model, genel olarak veri tabanının yapısını belirler. Ardından, bu model, gerçekleştirilebilecek bir veri tabanı sistemine dönüştürülür.
- Fiziksel tasarım aşamasında, verilerin nasıl organize edileceği belirlenir. Bu, veri tabanının iç yapısını ve depolama şeklini içerir. Bu aşamada, genellikle fiziksel veri modelleri kullanılır ve bu model, yazılım ve donanıma bağlıdır.

Veritabanı Tasarım Aşamaları



©Veritabanı Yöntemleri
Prof. Dr. İbrahim Çiğ

Veritabanı Tasarım aşamaları

05

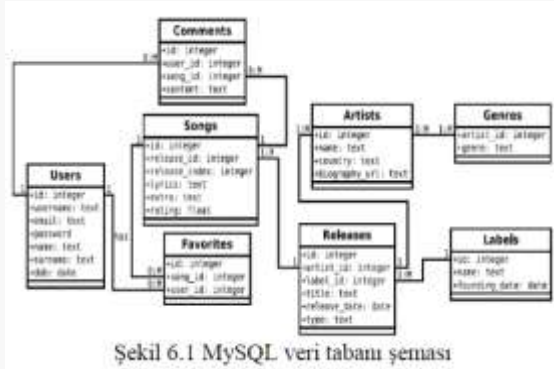
İlişkisel ve NoSQL veri tabanı sistemleri

-
- NoSQL veri tabanları, ilişkisel veri tabanlarına alternatif olarak geliştirilmiştir ve özellikle büyük ölçekli veri depolama ve işleme ihtiyaçlarını karşılamak için kullanılır. Bu sistemler, yatay olarak ölçeklendirilebilen dağıtık veri depolama sistemleri olarak tanımlanabilir. Örneğin, sosyal ağlar gibi büyük veri miktarlarına sahip platformlar, NoSQL veri tabanlarını tercih etmektedir. Ölçek sorunlarına cevap olarak geliştirilen NoSQL çözümleri, özellikle Twitter ve Facebook gibi devasa işlem hacimlerine sahip platformlar için idealdir. NoSQL'in tercih edilme nedenleri arasında, hızlı erişim ve yatay büyüme imkanı sağlaması, ilişkisel veri tabanlarının ek maliyetlerinden kaçınmayı mümkün kılması yer almaktadır. NoSQL veri tabanları, ilişkisel veri tabanlarının ACID (Atomicity, Consistency, Isolation, Durability) işlemselliğine karşılık olarak BASE (Basically Available, Soft state, Eventually consistent) prensibini benimserler. Bu prensip, temelde sürekli kullanılabilirlik, yumuşak durum ve sonunda tutarlılık anlamına gelir. Bu sayede NoSQL veri tabanları, büyük ölçekli ve dağıtık sistemlerde esneklik sağlar ve performanslarını artırır.

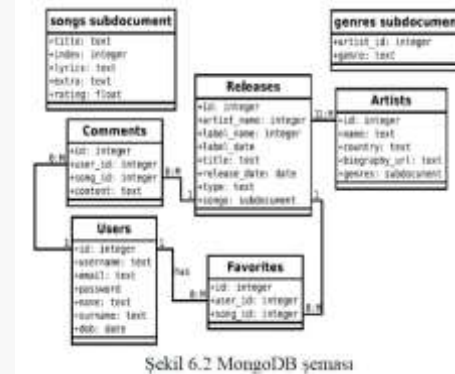
DEPOLAMA	PROTOKOL	MODEL	LİSANS	. DİL	
MENTROY MAPPED B- TRESS	BSON	DOCUMENT	AGPL	C++	MONGO DB
COW-B-TREE	HTTP/REST	DOCUMENT	APACHE	ERLANG	COUCHDB
PLUGGABLE INNODB	HTTP/REST TSP	KEY/VALUE	APACHE	ERLANG	REAKIN
INMEMORY SNAPSHOT TO DISK	BUFS TCP	KEY/VALUE	BSD	C++	REDIS
PLUGGABLE BSV MY SQL		KEY/VALUE	APACHE	JAVA	VOLDEMORT
MENTABLE- SSTABLE	TCP/THRIFT	WIDE COLUMN	APACHE	JAVA	CASSANDRA
HDFS	HTTP REST	WIDE COLUMN	APACHE	JAVA	H-BASE

06

Veri tabanı mimarilerinin
performans karşılaştırması

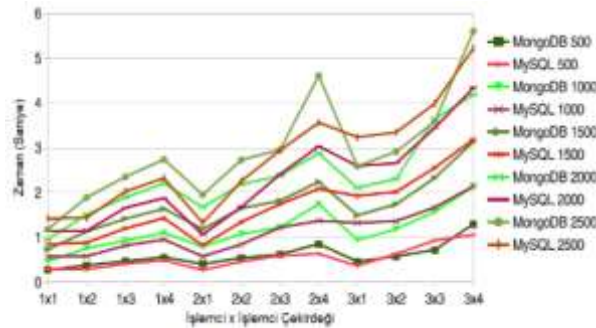


MYSQL

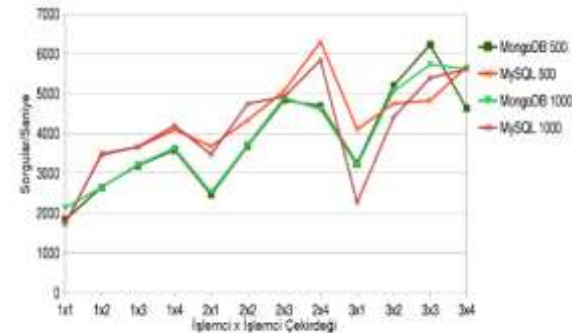


MONGODB

MONGO DB VS MY SQL

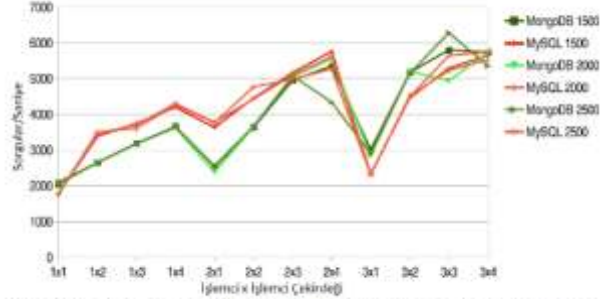


Şekil 6.3 Sorgu 1- Analiz işlemi

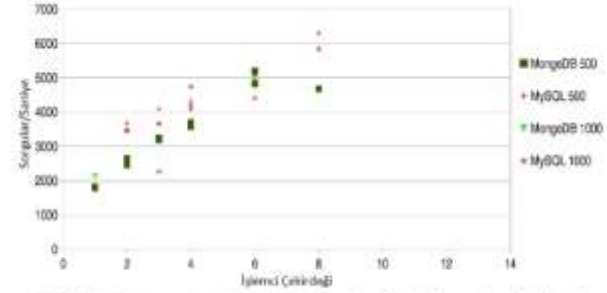


Şekil 6.4 Sorgu 1 - Sorgu/saniye analiz işlemi

MONGO DB VS MY SQL

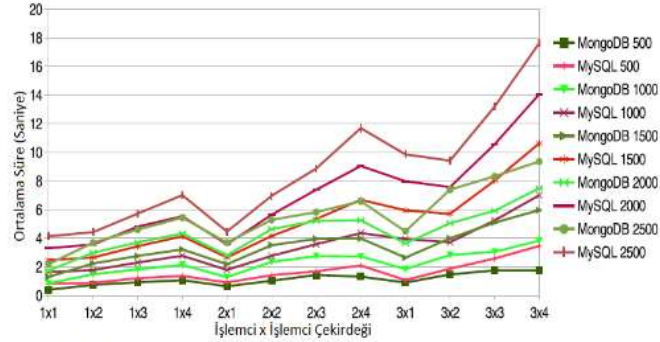


Şekil 6.5 Sorgu 1-Çok sayıdaki sorgu miktarı analiz işlemi



Şekil 6.6 Sorgu 1-Sorgular/Saniye ile işlemci çekirdeği miktarı için analiz işlemi

MONGO DB VS MY SQL

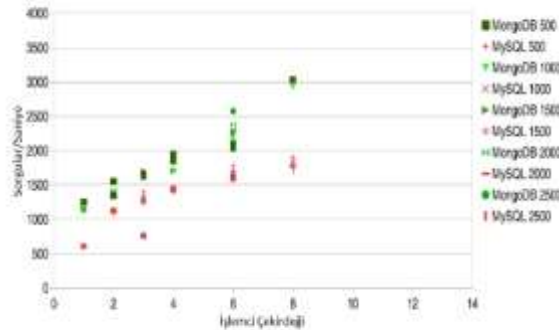


Şekil 6.7 Sorgu 2 - INNER JOIN ile karmaşık sorgu analizi işlemi

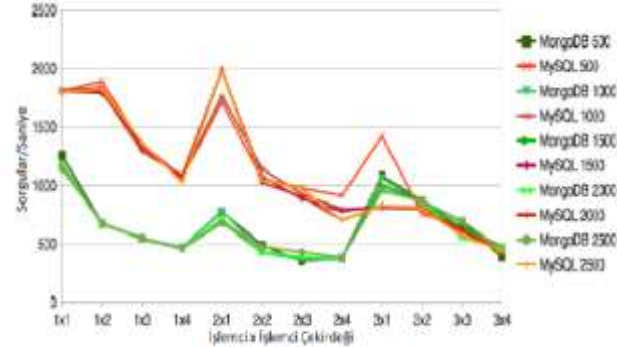


Şekil 6.8 Sorgu 2- INNER JOIN ile 500 ve 1000 veri için sorgu/saniye analizi işlemi

MONGO DB VS MY SQL

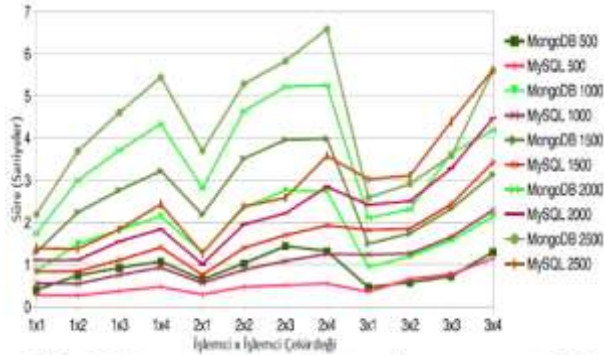


Şekil 6.9 Sorgu 2- INNER JOIN ile işlemci çekirdeği miktarı üzerinde analiz işlemi

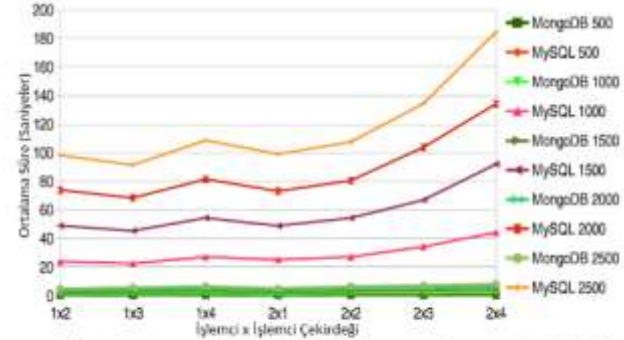


Şekil 6.11 Sorgu 3- Detaylı ve karmaşık sorgu ile Sorgular/saniye analiz işlemi

MONGO DB VS MY SQL

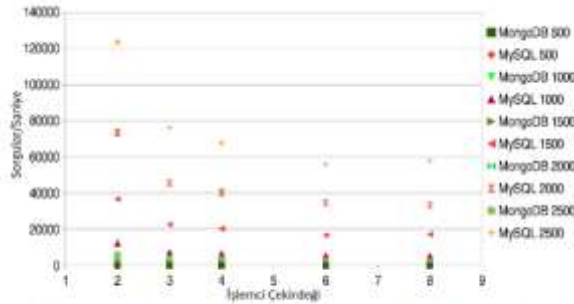


Şekil 6.10 Sorgu 3 – Detaylı karmaşık sorgu süre analizi

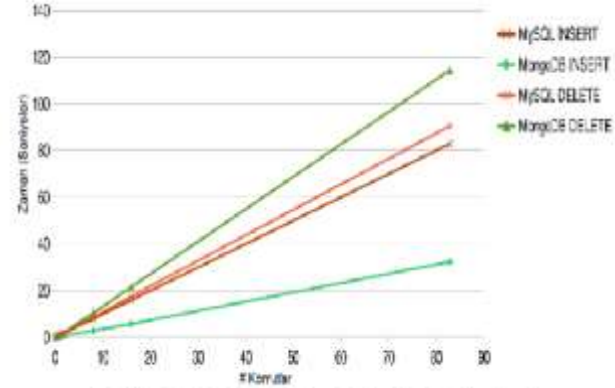


Şekil 6.12 Sorgu 3 – Detaylı ve karmaşık sorgu kodu ile ortalama süre analiz işlemi

MONGO DB VS MY SQL



Şekil 6.13 Sorgu 3 - Detaylı ve karmaşık sorgu ile işlemci çekirdeği üzerinde analiz işlemi



Şekil 6.15 INSERT ve DELETE işlemleri

• İLKER KALECİK

02220224041

