

# Kalıcı Veri Yönetimi Veri Tabanı Sistemleri

Fehmi Şener – Tolunay Esergün

# DATA (VERİ) vs. INFORMATION (BİLGİ)

## DATA (Veri)

- Herhangi bir şey veri olabilir; sayı, kelime, görsel, ses vs. Tek başına bir anlam ifade etmesine gerek yoktur.
- Ham bilgi barındırır.
- Bilginin yapı taşlarıdır.
- Verinin düzenli bir şekilde yönetilmesi gerekir. Bunun içinde verilerin üretilmesi, depolanması, saklanması gibi kavramların uygulanması gerekir.
- Ham veriler depolama, işleme ve sunum için biçimlendirilmelidir.
- Bilgiye *bağlı değildir*.

## INFORMATION (Bilgi)

- Veriyi kullanışlı hale getirmek için bir bağlamda düzenlenir ve sunulursa bilgi denir.
- Verinin işlenmesi ile elde edilir/oluşturulur.
- Verinin anlamını ortaya çıkarır.
- İyi bir karar vermek için doğru, ilgili/alakalı ve zamanına uygun olmalıdır.
- Üstte ki şartın sağlanabilmesi için veri kaynaklarının da doğru, ilgili/alakalı ve zamanına uygun olmalıdır.
- Veriye *bağımlıdır*.

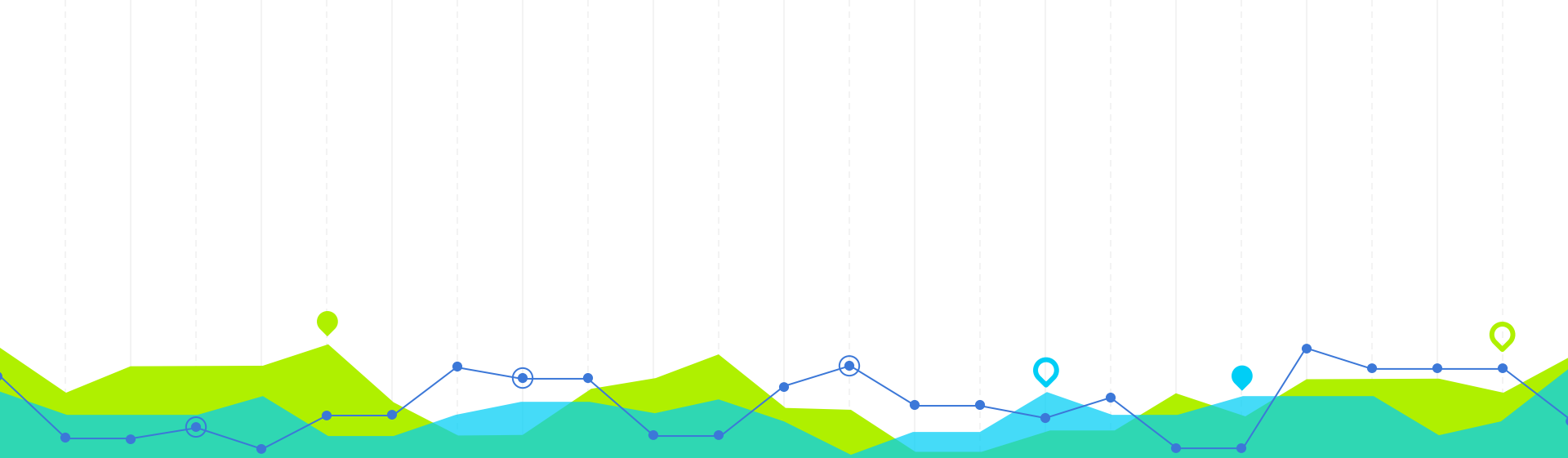
## DATA



## INFORMATION



\* [effectualsystems.com/data-need-information](http://effectualsystems.com/data-need-information)



# Veri Tabanı Nedir?

Veri Tabanı; düzenli veri koleksiyonlarının depolandığı ve ihtiyaç halinde kullanıldığı depolama ortamıdır.

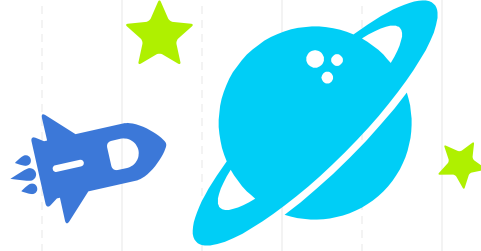
1

“

*In the 21st century, the database is the marketplace.*

*- Stan Rapp*

\* [azquotes.com/quote/678717](https://azquotes.com/quote/678717)



# 1960

Veri tabanı kavramı ilk olarak 1960'lı yıllarda ortaya atılmıştır.

\* [vargonen.com/blog/veritabani-nedir/](http://vargonen.com/blog/veritabani-nedir/)

# VERİ TABANI NEDİR?

- Veri tabanı kısaca, büyük çapta ilgili veri koleksiyonu platformudur.
- Veri tabanının önemi veriyi verimli ve hızlı bir şekilde yönetilebilir ve erişilebilir hale getirmesidir.
- İhtiyaç anında en hızlı şekilde ulaşmak temel esastır.
- Bilgi sistemlerinin kalbidir. Etkili ve verimli şekilde kullanmak gerekir.





# VERİ TABANI YÖNETİM SİSTEMİ NEDİR?

Bir veri tabanındaki verileri işlemek, yönetmek, almak, depolamak vb. işlemler için tasarlanmış yazılım sistemidir.

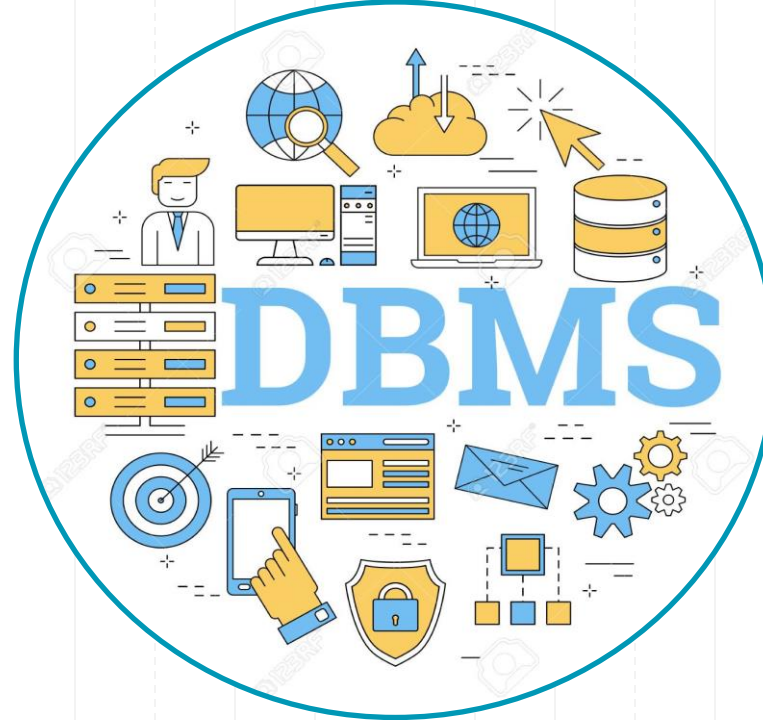
# 2



# VERİ TABANI YÖNETİM SİSTEMİ NEDİR?

Yeni bir veri tabanı  
oluşturmak.

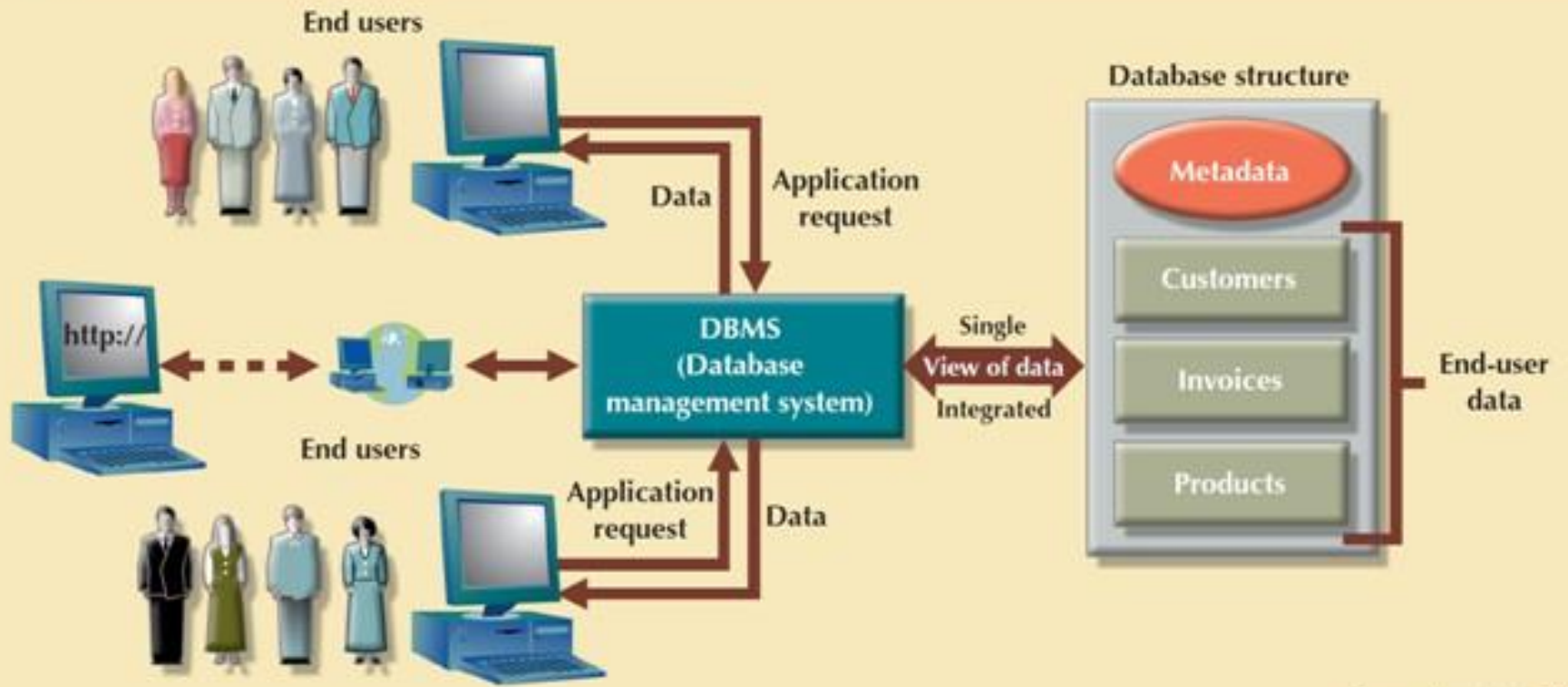
Veri tabanında  
düzenleme, ekleme  
çıkarma yapmak.



Geliştirme, bakım yapmak.

İhtiyaca bağlı farklı  
özellikleri kullanmak ve  
yönetimi kolaylaştırmak.

Verinin paylaşılmasına izin  
vermeli.



# VERİ TABANI YÖNETİM SİSTEMLERİNİN SINIFLANDIRILMASI

## Veri Modeline Göre

- Hiyerarşik
- Ağ / Graf
- İlişkisel
- **NoSQL**
- Nesneye yönelik

## Kullanıcı Sayısına Göre

- Tek Kullanıcılı (Access)
- Çok Kullanıcılı

## Fiziksel Konuma Göre

- Merkezi
- Dağıtık

# İLİŞKİSEL (RDBMS) vs. NOSQL VERİ TABANI YÖNETİM SİSTEMLERİ

## İlişkisel (RDBMS)

- Organize edilmiş verilerin tablolarda saklanması ve bu tablolar arasında kurulan bağ ile oluşan veri tabanı çeşididir.
- Tablolar satır ve sütunlardan oluşur.
- Her satır aynı sütunlara yani alanlara sahiptir.

## NoSQL

- NoSQL, ilişkisel veri tabanı sistemlerine alternatif bir çözüm olarak ortaya çıkar.
- Yatay olarak ölçeklendirilen bir veri depolama sistemidir.
- Belirli veri modelleri için özel olarak tasarlanmıştır ve esnek şemalara sahiptir.



# İLİŞKİSEL VERİ TABANI SİSTEMLERİ AVANTAJLARI



## Veri bütünlüğü

Veri bütünlüğü, verilerin doğruluğunu ve tutarlılığını sağlamaya yardımcı olur.



## Basitlik

İlişkisel model, verileri karmaşıklığı önleyecek şekilde yapılandırır.



## Normalizasyon

Veri tabanının bütünlüğünü ve doğruluğunu bozacak anomalilerin tespiti için metodoloji kuralları vardır.



## Veriye Ulaşma Kolaylığı

Herhangi bir tabloyu sorgulayabilir veya birleştirme işlevlerini kullanarak ilgili tabloları birleştirebilir.



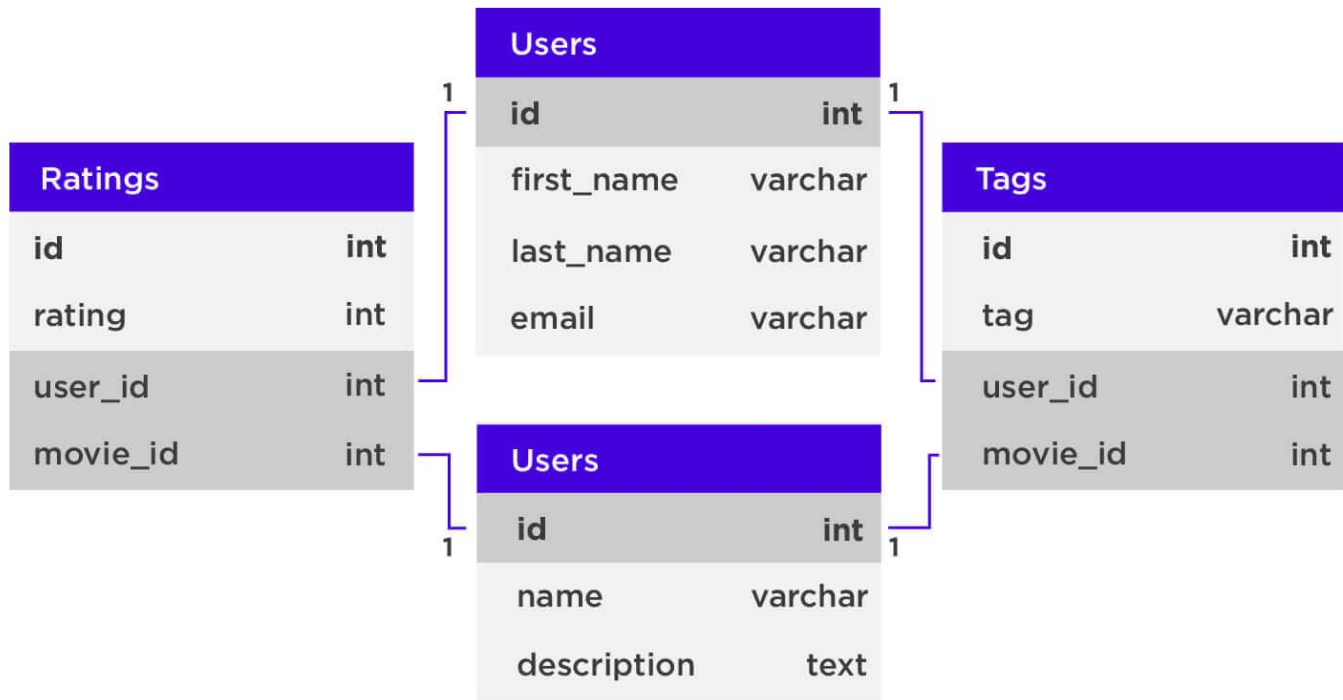
## Veri Bağımsızlığı

Verilerin fiziksel alan üzerinde nasıl depolandığı gibi konular ile ilgilenilmesine gerek yoktur.



## Veri Tekrarı

Aynı verilerin tekrarlanması önlenir.



Jelvix

Source: Assets Global

jelvix.com

# NoSQL VERİ TABANI SİSTEMLERİ AVANTAJLARI



## Ölçeklenebilirlik

Yatay olarak genişleyebilir; dağıtılmış donanım kümeleri kullanılarak ölçeği genişletilebilecek şekilde tasarlanır.



## Erişilebilirlik

Aynı anda sisteme çok sayıda kullanıcının bağlı olduğu durumlarda kontrol ve yönetim daha kolaydır.



## Maliyet Avantajı

Açık kaynak kodlu projelere ve bulut bilişim teknolojilerine uygun olduğu için maliyet olarak ilişkisel veri tabanı yönetim sistemlerine göre daha avantajlıdır..



## Esneklik

Esnek veri modeli sayesinde NoSQL veri tabanları yarı yapılandırılmış ve yapılandırılmamış veriler için idealdir.



## İşlevsellik

Yaygın Cloud desteği ve NoSQL veri tabanları, her biri ilgili veri modeli için özel olarak tasarlanmış yüksek oranda işlevsel API'ler ve veri türleri sağlar.


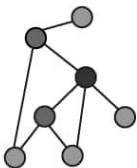
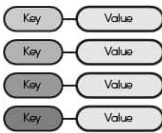
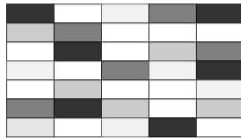
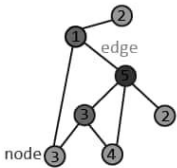
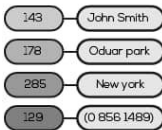
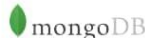













## Yüksek Performans

Daha yüksek performansı mümkün kılan belirli veri modelleri (belge, anahtar-değer, grafik gibi) destekler.



# NoSQL DATABASE TYPES

Document	Graph	Key-Value	Wide-Column															
																		
<pre>{   "user": {     "id": "143",     "name": "improgrammer",     "city": "New York"   } }</pre>			<table><tr><td>1</td><td>Fruit</td><td>A Foo</td><td>B Baz</td><td></td></tr><tr><td>2</td><td>City</td><td>E DC</td><td>D PLA</td><td>G FLD</td></tr><tr><td>3</td><td>State</td><td>A NZ</td><td>C CL</td><td></td></tr></table>	1	Fruit	A Foo	B Baz		2	City	E DC	D PLA	G FLD	3	State	A NZ	C CL	
1	Fruit	A Foo	B Baz															
2	City	E DC	D PLA	G FLD														
3	State	A NZ	C CL															
  	 	   	  															

[improgrammer.net/most-popular-nosql-database](http://improgrammer.net/most-popular-nosql-database)

ORACLE®

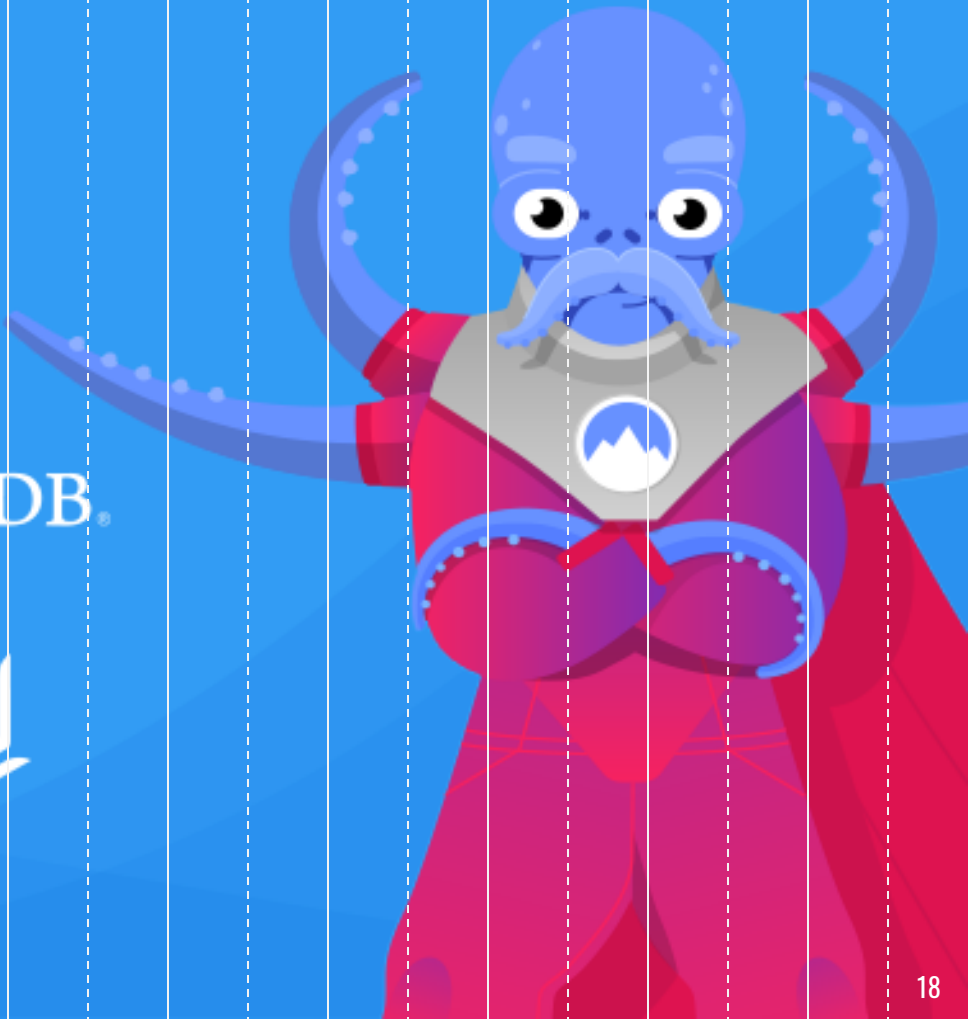
MySQL™

Microsoft®  
SQL Server

mongoDB®

cassandra

Xplenty



# SQL

VERTICAL  
SCALABILITY



# NoSQL

HORIZONTAL  
SCALABILITY



# İLİŞKİSEL (RDBMS) vs. NOSQL FARKLARI

	İlişkisel veritabanları	NoSQL veritabanları
En uygun iş yükleri	İşlemsel ve güçlü tutarlılığa sahip çevrimiçi işlem gerçekleştirme (OLTP) uygulamaları için tasarlanan ilişkisel veritabanları, çevrimiçi analitik işlem (OLAP) için uygundur.	NoSQL veritabanları, düşük gecikme süreli uygulamaları içeren çeşitli veri erişimi desenleri için tasarlanmıştır. NoSQL arama veritabanları, yarı yapılandırılmış veriler üzerinde analiz için tasarlanmıştır.
Veri modeli	İlişkisel model, verileri satır ve sütunlardan oluşan tablolar halinde normalleştirir. Tablolar, satırlar, sütunlar, dizinler, tablolar arasındaki ilişkiler ve diğer veritabanı öğeleri bir şema tarafından kesin bir şekilde tanımlanır. Veritabanı, tablolar arasındaki ilişkilerde başvurusal bütünlük uygular.	NoSQL veritabanları, performans ve ölçek için optimize edilmiş anahtar-değer, belge ve grafik gibi çeşitli veri modelleri sağlar.
ACID özellikleri	<p>İlişkisel veritabanları bölünmezlik, tutarlılık, yalıtım ve dayanıklılık (ACID) özelliklerini sağlar:</p> <ul style="list-style-type: none"><li>Bölünmezlik, bir işlemin ya tamamen yürütülmesini ya da hiç yürütülmemesini gerektirir.</li><li>Tutarlılık, bir işlem gönderildiğinde verilerin veritabanı şemasına uygun olmasını gerektirir.</li><li>Yalıtım, eş zamanlı işlemlerin birbirinden bağımsız olarak yürütülmesini gerektirir.</li><li>Dayanıklılık, beklenmeyen bir sistem hatasından veya güç kesintisinden son bilinen duruma kurtarma becerisi gerektirir.</li></ul>	NoSQL veritabanları yatay olarak ölçeklendirilebilen daha esnek bir veri modeli sağlamak için genellikle ilişkisel veritabanlarının bazı ACID özelliklerini esneterek bunlardan ödün verirler. Bu, tek bir bulut sunucusunun ulaşamayacağı derecede yatay ölçeklendirme gerektiren yüksek performanslı, düşük gecikme süreli kullanım örnekleri için NoSQL veritabanlarının mükemmel bir seçim olmasını sağlar.
Performans	Performans genellikle disk alt sistemine bağlıdır. En üst düzey performans için genellikle sorguların, dizinlerin ve tablo yapısının optimize edilmesi gerekir.	Performans genel olarak temel donanımın küme boyutu, ağ gecikme süresi ve çağrı yapan uygulama gibi etmenlerin birleşimine bağlıdır.
Ölçek	İlişkisel veritabanları genellikle donanımın işlem kapasitesini artırarak ölçeği artırır veya salt okunur iş yüklerine yönelik replikalar ekleyerek ölçeği genişletir.	Erişim desenleri aktarım hızını artırmak için neredeyse sınırsız ölçekte tutarlı performans sağlayan dağıtılmış mimariyi kullanarak ölçeği genişletebildiğinden, NoSQL veritabanları genellikle bölümlendirilebilen veritabanlarıdır.
API'ler	Veri depolama ve alma istekleri, yapılandırılmış sorgu diline (SQL) uygun sorgular kullanılarak iletilir. Bu sorgular ilişkisel veritabanı tarafından ayrıştırılır ve yürütülür.	Nesne tabanlı API'ler, uygulama geliştiricilerinin veri yapılarını kolayca depolamasına ve almasına imkan tanır. Bölüm anahtarları, uygulamaların anahtar-değer çiftlerini, sütun kümelerini veya seri hale getirilmiş uygulama nesneleri ve öznitelikleri içeren yarı yapılandırılmış belgeleri bulmasına imkan tanır.

\* [aws.amazon.com/tr/nosql](https://aws.amazon.com/tr/nosql)

# DEMO ZAMANI...



# Teşekkürler!

## Sorular?

[fehmiisener@gmail.com](mailto:fehmiisener@gmail.com) / [tolunay50@gmail.com](mailto:tolunay50@gmail.com)

