

02225076553 -- Yasin Koer
02220201909 -- TURAN AYHAN
02215076001 -- Hamza Turgay Acı

02200201006 -- Samed Sonkaya
02200201013 -- Anıl Berkan Torun
02200201011 -- Habib ako

NoSQL Nedir ?

Son yıllarda verinin inanılmaz boyutlara ulaşması ve katlanarak artması sonucunda mevcut olarak kullanılan ilişkisel veritabanı sistemleri yerine ortaya atılmış bir kavramdır. İlişkili veritabanı sistemleri ile arasındaki en büyük fark ilişkisel veritabanı sistemlerinde veriler tablo ve sütunlar ile ilişkili bir şekilde tutulurken NoSQL’de json bir yapıda tutulmasıdır.

NoSQL sistemlerin avantajlarına değinmek gerekirse ilk olarak performans gösterilebilir. Okuma ve yazma işlemleri ilişkisel veritabanlarına göre ok daha hızlı olmaktadır. İkinci olarak ise NoSQL sistemler yatay olarak genişletilebilirler. Binlerce sunucu bir arada alışarak inanılmaz derecedeki veriler üzerinde işlemler yapılabilir.

Ayrıca eklemekte fayda var ki günümüzde Büyük Veri alanında yapılan alışmalarda NoSQL sistemler yoğun olarak kullanılmaktadır.

MongoDB Nedir?

MongoDB 2009 yılında geliştirilmiş açık kaynak kodlu bir NoSQL veritabanıdır. Bugün piyasada Cassandra, BigTable, Dynamo gibi birçok NoSQL veritabanı bulunmaktadır.

MongoDB’de her kayıt bir doküman olarak ifade edilir. Ve bu dökümanlar json formatı şeklinde saklanır. Daha önce ilişkisel veritabanlarıyla ilgilenenlerin bildiğı table yapısını burada collection, row yapısını document, column yapısını ise field alır.

Günümüzde aktif olarak kullanılan pek ok programlama dili için driver desteğı bulunması bakımından bugün NoSQL sistemler içerisinde en ok tercih edilenlerden biridir.

MongoDB’nin Faydaları

MongoDB kullanıcıları, sistemin sağladığı avantajları bilerek sisteme girerler. Bu sayede arzu ettikleri faydaya kısa sürede ulaşabilirler. Sistemin sağladığı ok önemli 4 faydayı aşağıda bulabilirsiniz:

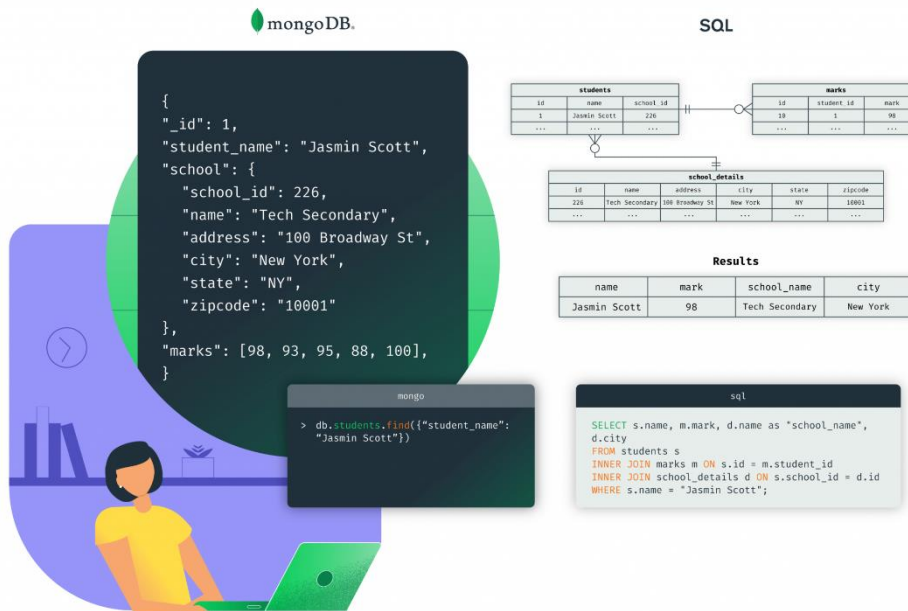
Kolay Kurulum Özelliğı

MongoDB ile birlikte gelen kurulum sihirbazını da kullanarak kurulum yapabilirsiniz. Linux, MacOS, Windows ve Docker için söz konusu sistem tercih edilebilir. Bununla

birlikte Linux kullanıyorsanız, CentOS, Red Hat, Ubuntu, SUSE, Debian ya da Amazon Linux arasında seçim yapmanız gerektiğini de belirtmekte fayda vardır. Yine de de hangi sistemi tercih ederseniz edin, MongoDB'yi aktif etmek için çok fazla gayret sarf etmenize gerek kalmayacaktır.

Şemasız Bir Veri Tabanınız Olacaktır

MongoDB'nin önemli avantajlarından biri de şemasız bir veri tabanına sahip olmasıdır. Şema, birçok veri tabanında kullanılan tablo ve sütunların yapısına sahiptir. SQL gibi bir sistem kullanıyorsanız, veri tabanını kurmadan önce bir şema tanımlaması yapmanız gerekir. Bunun yanında MongoDB, şema yapısına ihtiyaç duymadan birçok dosya türünün depolanmasını sağlar. Ayrıca söz konusu dosyalara hızlı bir şekilde erişmek için gelişmiş çok yönlülüğe izin vererek şema ihtiyacını ortadan kaldırır.



Kısaca belirtmek gerekirse, SQL veri tabanlarının aksine, MongoDB nesne odaklı bir veri tabanı yönetim sistemidir. Daha basit bir şekilde ifade edecek olursak, sistemde tablolar yerine koleksiyonlar halinde düzenlenmiş belgeler kullanılır. MongoDB kullanıcıları net bir hız artışına sahip olurlar. Şemasız veri tabanı sayesinde, yapmanız gereken tek şey bir veri noktasını dizine eklemektir. Bu da daha az işlem gücü demektir. Donanımınızın gerginliğini de bu sayede azaltabilirsiniz.

Yüksek Performans Ortaya Koyar

Veri tabanının performansı, dolaylı olarak web sitenizin ya da uygulamanızın hızını da etkiler. Bu durum da kullanıcı deneyimini (UX) iyileştirir. İçeriğin yüklenmesini beklemek web sitesi kullanıcılarının başını ağrıtabilecek ve onları rahatsız edecektir. Yukarıda da belirtildiği gibi MongoDB'nin şemasız bir şekilde kullanılması, içeriklerin de hızlı yüklenmesine neden olacaktır. Bunun yanında performans yetenekleri açısından sağlanan başka avantajlar da vardır.

MongoDB, SQL gibi tablo birleşimlerinin kullanılmasına izin vermez. SQL ile iç ve dış birleşimler yapabilirsiniz. Bu işlemler ile iki tabloyu kısmen ya da tamamen bir araya getirebilirsiniz.

Öte yandan tablolarınıza katılma yeteneği, veri tabanlarının satırları birleştirme ve bilgileri paylaşması, çok önemli bir özelliktir. Aynı yapı kurulmadan MongoDB'nin birleştirilecek tabloları yoktur. Bu durum, bir dezavantaj gibi görünse de birleştirme yapmak, performansı düşürücü bir etki yaratabilir. Bunun yerine belgelere güvenip yerden ve kaynaklardan tasarruf elde edebilirsiniz.

Uygun Maliyetli Bir Veri Tabanı Yönetim Sistemidir

SQL seçeneklerini inceliyorsanız; geliştirici seçenekler için en temel sürümün ücretsiz olduğunu göreceksiniz. Fakat daha kapsamlı bir sisteme ihtiyaç duyuyorsanız ödeyeceğiniz ücret yüksek aralıkta olacaktır. Maliyet-performans analizi yaptığınızda da MongoDB'nin çok önemli avantajları ile karşılaşabilirsiniz. Bahse konu sistem, sadece açık kaynaklı bir veri tabanı yönetim sistemi değildir. Sistem, aynı zamanda birden fazla dil seçeneği ile kullanıcıya uygun maliyetli bir hizmet sağlar.

Dil seçeneklerinden 10 tanesi resmi dildir. Ancak MongoDB tarafından ortaya konulan çok daha fazla dil seçeneği de vardır. MongoDB ile birden fazla sunucuya kolayca yayılma yeteneğine de sahip olabilirsiniz. Bu da size büyümek için gerekli altyapıyı sağlayacaktır. Ayrıca sistem, herhangi bir ölçüte ihtiyaç duymadan verileri bulmanıza yarayan bir dizin desteği ve dinamik bir altyapı da sağlayacaktır. Web sitenizin verimli çalışması için söz konusu sistem çok önemli bir çözüm ortaya koyabilir.

MongoDB Temel İşlemler

Bu aşamada ise MongoDB üzerinde veritabanlarını listelemeyi, collectionları görmeyi, veri eklemeyi, silmeyi, güncellemeyi ve listelemeyi görelim. MongoDB'yi bilgisayarımıza yükledikten sonra grafiksel bir arayüz içinde kullanmak isterseniz piyasada robomongo gibi birçok güzel seçenek mevcut. Ama ben işlemlerin daha net anlaşılması için komut ekranı aracılığıyla yapıyor olacağım. MongoDB'yi çalıştırmam için komut ekranını açtıktan sonra 'mongo' kodunu yazarak entere basıyorum. Artık MongoDB'yi çalıştırmış olduk.

MongoDB : Veritabanlarını Listelemek

Sistemimizde bulunan veritabanlarını listelemek için 'show databases' komutunu yazıyoruz. Sistemimizde kayıtlı olan veritabanları ve boyutları böylelikle karşımıza listelenecektir.

show databases

```
> show databases
2019berkekurnazdb  0.000GB
DbAnaliz           0.027GB
DbBooks            0.000GB
DbDiaryApp         0.000GB
FlaskMongo         0.000GB
MvcCrudDb          0.000GB
RehberUygulama     0.000GB
admin              0.000GB
config             0.000GB
local              0.000GB
myDatabase         0.000GB
testdb             0.000GB
>
```

MongoDB : Yeni Veritabanı Oluşturmak

Yeni bir veritabanı eklemek için şöyle bir yol izleyeceğiz. Önce olmayan bir veritabanını kullan komutunu verip içerisine veri eklediğimiz zaman veritabanının oluştuğunu göreceğiz. Şimdi ekleyeceğim veritabanımın ismini BenimVeritabanım olarak giriyorum.

Kodu yazdıktan sonra switched to db BenimVeritabanım şeklinde mesaj karşımıza gelecek.

use BenimVeritabanim

MongoDB : Var Olan Veritabanını Kullanmak

Sistemimizde daha önce kayıtlı bir veritabanını kullanmak istediğimizde yazacağımız kod yine aynı olacak. Önce use ardından ise veritabanının ismi.

use Veritabanınınİsmi

MongoDB : Veritabanı Silmek

Sistemimizde kayıtlı bir veritabanını silmek istediğimizde ise önce silmek istediğimiz veritabanına use ile geçiş yapıyoruz. Ardından ise db.dropDatabase() komutunu yazıyoruz. Bu işlemin ardından ilgili veritabanı silinecektir.

use Veritabanınınİsmi

db.dropDatabase()

MongoDB : Veritabanında Collection Oluşturma

Collection yani diğer anlamıyla tablo oluşturmak istediğimiz zaman önce ilgili veritabanının içine giriyor ardından ise db.createCollection("İsim") komutunu yazıyoruz.

use BenimVeritabanim

db.createCollection("musteriler")

MongoDB : Veritabanında Collectionları Listeleme

Veritabanımızda bulunan collectionları listelemek için show collections komutunu kullanmamız yeterli. Ben bir yukarıda müşteriler isimli bir collection oluşturduğum için bana onu listeleyecektir.

show collections

```
> show collections
musteriler
>
```

MongoDB : Veritabanında Collection Silme

Veritabanımızda var olan bir collectionu silmek istersek db.CollectionAdi.drop() komutunu giriyoruz.

db.musteriler.drop()

MongoDB : Collection İçerisine Veri Ekleme

Collection içerisine veri eklemek için db.Collectionİsmi.insert() diyerek insert içerisine json formatında veri giriyoruz. Örnek vermek gerekirse ben yukarıda oluşturduğumuz müşteriler collectionuna ismi berke soyismi kurnaz ve yaşı 20 olan bir kayıt eklemek istersem aşağıdaki gibi bir kod yazacağım.

db.musteriler.insert({"name": "Berke", "surname": "Kurnaz", "age":20})

MongoDB : Collection İçerisindeki Verileri Listelemek

Ben yukarıda yaptığımız ekleme işlemlerine örnek birkaç veri daha girdim. Bu aşamada ise bu verileri listeleyelim. Bunun için yazmamız gereken komut `db.Collectionİsmi.find()` olacaktır. Bunun sonucunda veriler satır satır listelenir fakat biz `db.Collectionİsmi.find().pretty()` dersek veriler daha düzenli olarak listelenir. Tabii ki de hangisini kullanmak isterseniz tercih sizin.

`db.musteriler.find()`

`db.musteriler.find().pretty()`

db.musteriler.find() komutu sonrası çıktımız

```
db.musteriler.find()
{ "_id" : ObjectId("5cdbe3de5bf1023ab8014c2f"), "name" : "Berke", "surname" : "Kurnaz", "age" : 20 }
{ "_id" : ObjectId("5cdbe4665bf1023ab8014c30"), "name" : "Ali", "surname" : "Demir", "age" : 25 }
{ "_id" : ObjectId("5cdbe47a5bf1023ab8014c31"), "name" : "Ayse", "surname" : "Ozturk", "age" : 38 }
```

db.musteriler.find().pretty() komutu sonrası çıktımız

```
db.musteriler.find().pretty()
{
  "_id" : ObjectId("5cdbe3de5bf1023ab8014c2f"),
  "name" : "Berke",
  "surname" : "Kurnaz",
  "age" : 20
}
{
  "_id" : ObjectId("5cdbe4665bf1023ab8014c30"),
  "name" : "Ali",
  "surname" : "Demir",
  "age" : 25
}
{
  "_id" : ObjectId("5cdbe47a5bf1023ab8014c31"),
  "name" : "Ayse",
  "surname" : "Ozturk",
  "age" : 38
}
```

MongoDB : Collection İçerisindeki Veriyi Güncellemek

Collection içerisindeki veriyi güncellemek istediğimiz zaman kullanacağımız komut ise `db.Collectionİsmi.update()`. Burada update içerisine önce hangi şartlı kaydı güncellemek istediğimizi söylüyoruz. Bu genellikle her kayda otomatik olarak verilen `_id` değeri olur. Ardından ise hangi alanını değiştirmek istediğimizi söylüyoruz.

Örnek olması açısından müşteriler collectionunun içerisinde Berke'nin ismini Mehmet olarak değiştirelim.

```
db.musteriler.update({"_id": ObjectId("5cdbe3de5bf1023ab8014c2f")}, {$set: {"name": "Mehmet"}})
```

MongoDB : Collection İçerisindeki Veriyi Silmek

Bir collection içerisindeki veriyi silmek için kullandığımız komut ise

db.Collectionİsmi.remove(). Yine güncelleme işleminde olduğu gibi silmek istediğimiz kaydın değerini remove içerisine json formatında giriyoruz. Ve bu değer genellikle kaydın _id kısmı oluyor.

Örnek vermek gerekirse collectionum içerisinde yer alan Ayşe'yi silmek istiyorum. Yazacağım komut aşağıdaki gibi olacaktır.

```
db.musteriler.remove({"_id": ObjectId("5cdbe47a5bf1023ab8014c31")})
```

Windows İçin MongoDB Kurulumu

- MongoDB'nin resmi web sitesinden en son sürümü indirin. İndirme sayfasından "MSI" paketini seçin.

Version

7.0.5 (current)

▼

Platform

Windows x64

▼

Package

msi

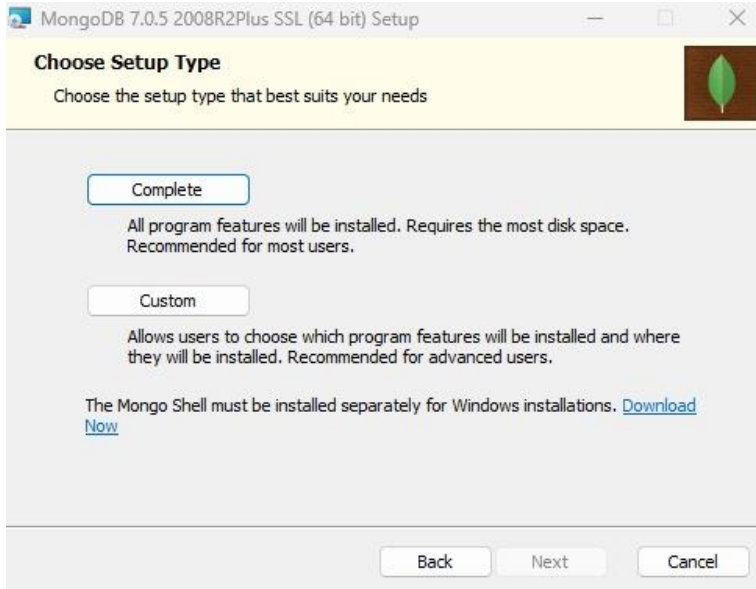
▼

Download

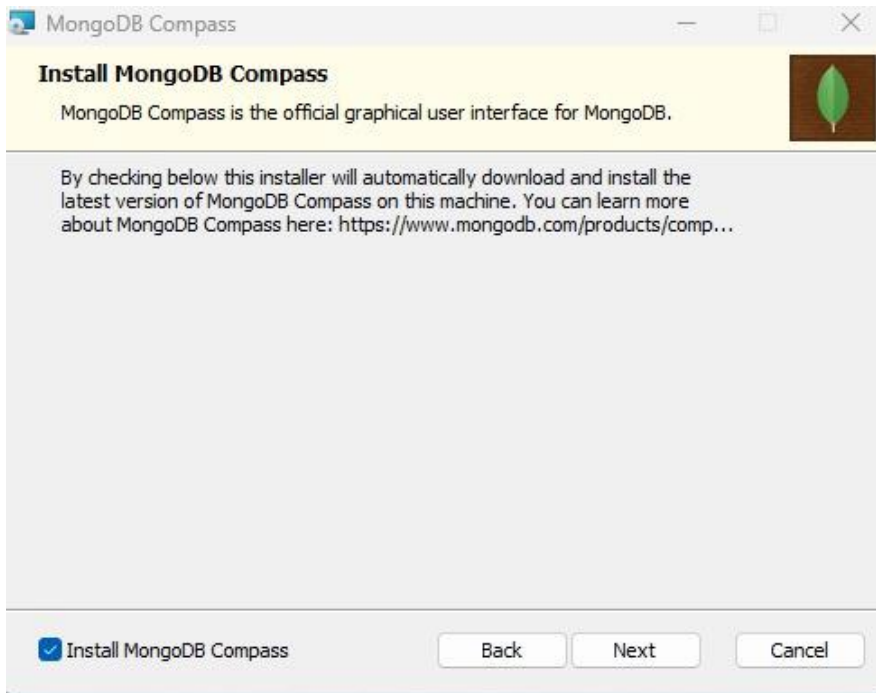
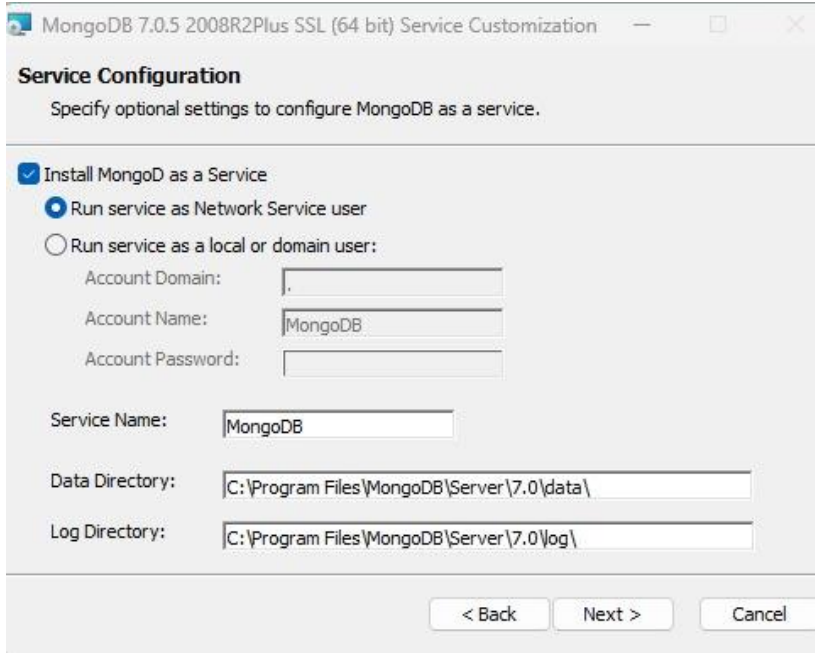
Copy link

More Options

- İndirilen MSI dosyasını çift tıklayarak çalıştırın.
- MongoDB'nin lisans sözleşmesini kabul edin ve "Next" butonuna tıklayın.
- "Complete" seçeneğini seçerek tüm bileşenleri yükleyin ve "Next" butonuna tıklayın.



- Kurulum dizinini seçin (varsayılan olarak "C:\Program Files\MongoDB\Server{version}") ve "Next" butonuna tıklayın.



- "Install" butonuna tıklayarak kurulumu başlatın.
- C:\Program Files dosyasına gelip bir tane data klasörü açıp içine db klasörünü oluşturuyoruz.
- Bilgisayarımızı yeniden başlattıktan sonra kurulumu tamamlamış oluyoruz.

SWinREAgent	10.01.2024 03:25	Dosya klasörü
android	1.11.2023 08:55	Dosya klasörü
data	29.01.2024 22:54	Dosya klasörü
Kullanıcılar	15.12.2022 08:59	Dosya klasörü
Microsoft	11.02.2023 19:12	Dosya klasörü
OneDriveTemp	29.01.2024 11:08	Dosya klasörü
PerfLogs	7.05.2022 08:24	Dosya klasörü
Program Dosyaları (x86)	15.01.2024 11:15	Dosya klasörü
Program Files	29.01.2024 22:53	Dosya klasörü
ProgramData	28.01.2024 01:57	Dosya klasörü
Recovery	17.11.2022 02:07	Dosya klasörü
Riot Games	1.02.2023 13:29	Dosya klasörü
Temp	25.12.2022 23:07	Dosya klasörü
Windows	10.01.2024 03:49	Dosya klasörü

- Connect butonuna basıp MongoDB'yi başlatıyoruz.

New Connection

Connect to a MongoDB deployment

URI ⓘ

Edit Connection String ☒

mongodb://localhost:27017/

> Advanced Connection Options

Save

Save & Connect

Connect

- Yeni bir database oluşturuyoruz.
- Collection ismini vermemiz gerekiyor aksi takdirde database'imizi oluşturamayız.
- Create Database butonuna tıklayıp yeni database'imizi oluşturuyoruz.

Create Database



Database Name

BooksCollection

Collection Name

books

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

➤ **Additional preferences** (e.g. Custom collation, Capped, Clustered collections)

Cancel

Create Database

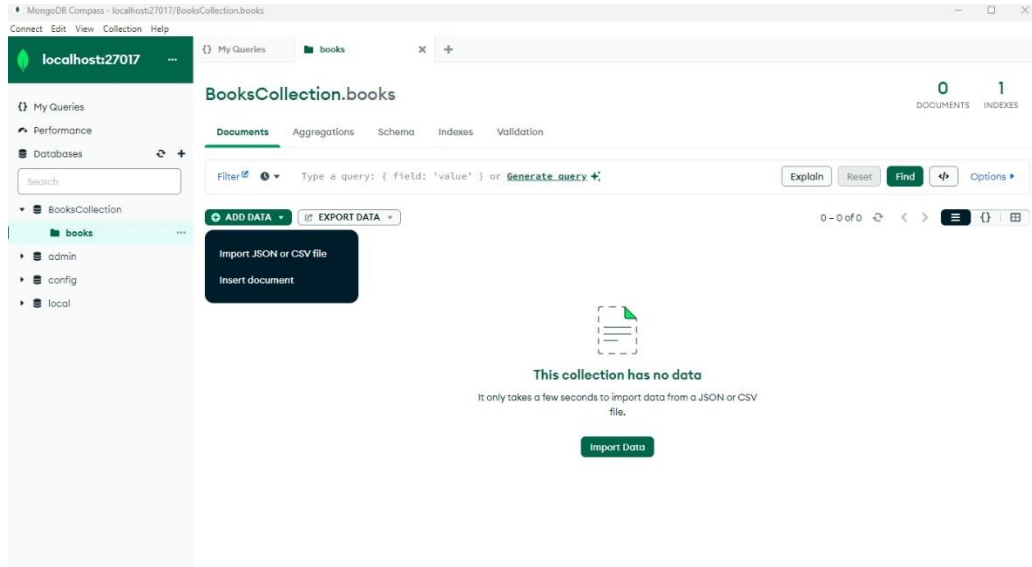
Senaryo

Kaggle'dan alacağımız "Top Goodreads Books Collection (1980-2023)" veri modelini MongoDB'ye entegre aşamasını aşağıda görebilirsiniz.

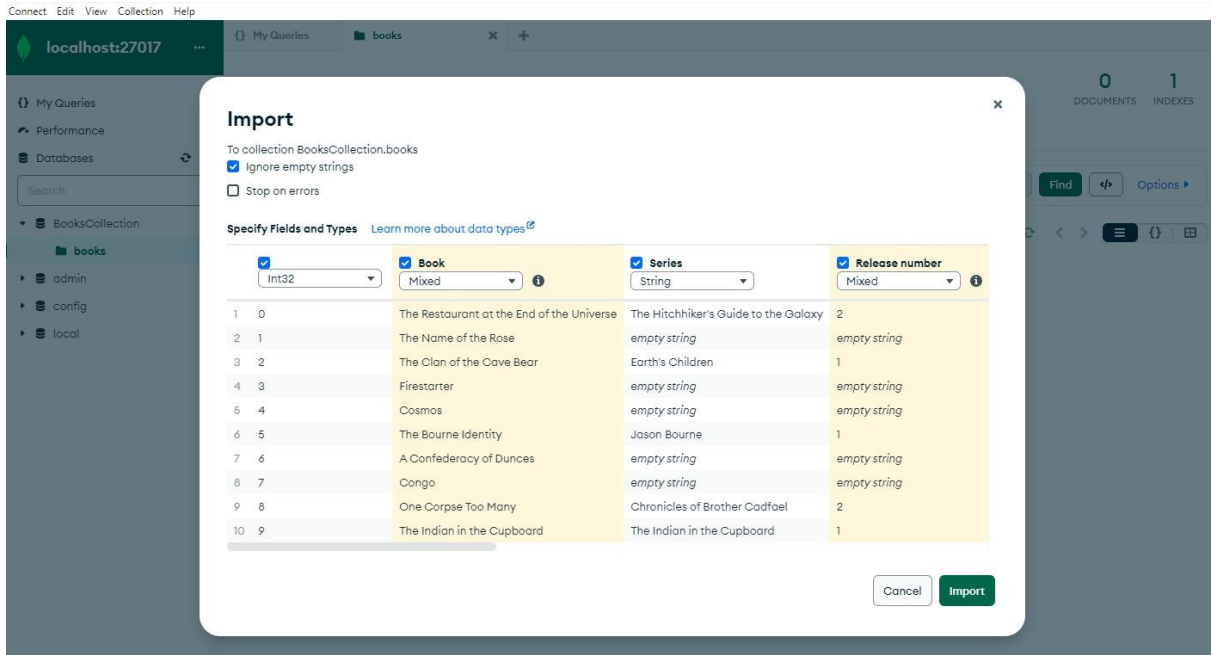
<https://www.kaggle.com/datasets/cristaliss/ultimate-book-collection-top-100-books-up-to-2023> sitesinden “En İyi Goodreads Kitap Koleksiyonu (1980-2023)” veri modelini csv uzantılı olarak indiriyoruz.



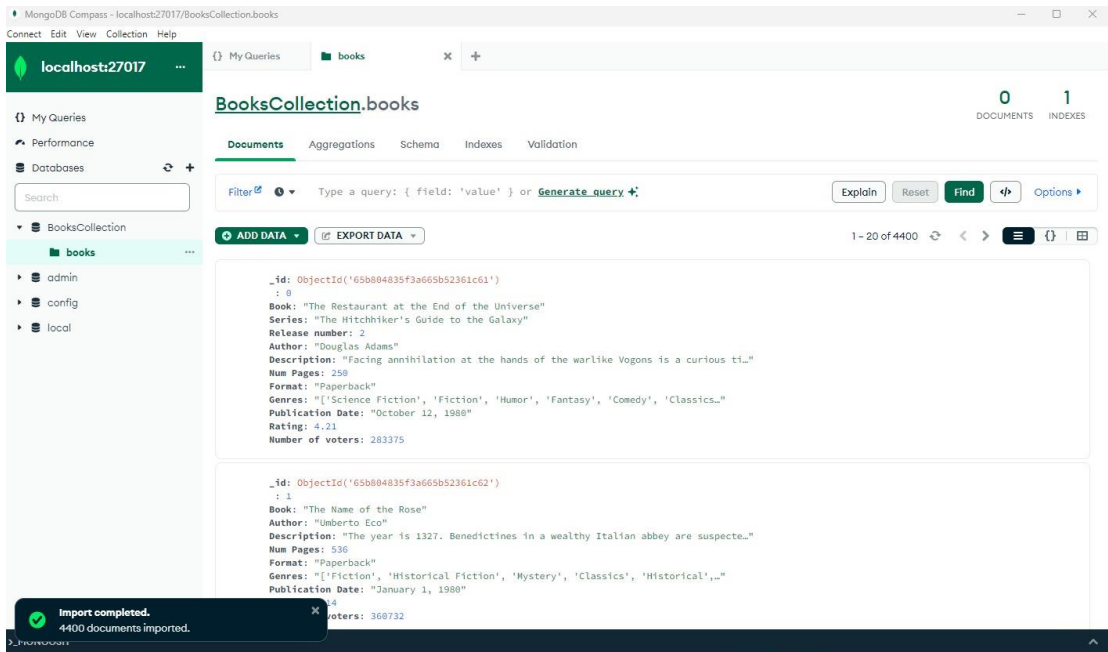
- App data butonuna tıklayıp “import json or csv” butonuna tıklayarak indirdiğimiz dosyayı import ediyoruz.



- Import ekranında hangi verileri ve veri tipini seçeceğimizi belirliyoruz.



- Import completed mesajını aldık.
- Dosyamızı import ettik.



MongoDB Mimarisi

MongoDB, NoSQL (Not Only SQL) veritabanı yönetim sistemlerinden biridir ve belge tabanlı bir veritabanıdır. Geleneksel SQL tabanlı veritabanlarına göre farklı bir mimariye sahiptir.

MongoDB'nin mimarisi şu ana unsurlardan oluşur:

Veri Modeli:

MongoDB, BSON (Binary JSON) formatında belgeleri depolar. Bu belgeler JSON benzeri bir formatta yazılmış anahtar-değer çiftlerinden oluşur. Belgeler, koleksiyonlarda gruplanır. Koleksiyonlar, SQL tablolarına benzer şekilde düşünülebilir.

Veri Tabanı Sunucusu (MongoDB Server):

MongoDB, çoklu platformlarda çalışabilen ücretsiz ve açık kaynaklı bir yazılımdır. Veri tabanı sunucusu, MongoDB istemcilerinin isteklerini işler ve veritabanı işlemlerini gerçekleştirir.

İstemci Uygulamaları:

MongoDB'ye erişmek için kullanılan uygulamalar. İstemci uygulamaları, MongoDB sürücülerini kullanarak MongoDB sunucusuna bağlanır ve isteklerini gönderir.

Sharding (Parçalama):

Büyük ölçekli MongoDB dağıtımlarında performans ve ölçeklenebilirlik sağlamak için kullanılır.

Veri, shard adı verilen parçalara bölünür ve bu parçalar farklı sunuculara dağıtılır.

Replikasyon:

Yüksek kullanılabilirlik ve veri yedekleme için kullanılır.

Veritabanı, birden fazla sunucu (replica set) üzerinde kopyalanır ve bu sunucular arasında otomatik olarak senkronize edilir.

Yönetim ve İzleme Araçları:

MongoDB'nin yönetimi ve izlenmesi için bir dizi araç vardır.

Bu araçlar, performans izleme, yedekleme ve veritabanı yapılandırmasını yönetmeyi içerir.

Veri Depolama Mimarisi:

Veriler, disk üzerinde BSON belgeleri olarak depolanır.

MongoDB, verileri yazarken ön bellek kullanır ve yazma işlemlerini onaylamak için birçok veri yapısı kullanır.

Küme (Cluster) Mimarisi:

MongoDB, küme oluşturma ve yönetme yeteneklerine sahiptir.

MongoDB kümesi, birden fazla MongoDB sunucusundan oluşur ve yüksek kullanılabilirlik ve ölçeklenebilirlik sağlar.

Bu unsurlar, MongoDB'nin genel mimarisini oluşturur. Ancak, bu unsurların yanı sıra, güvenlik, izinler, veri yönetimi ve performans gibi diğer önemli konular da MongoDB mimarisinin bir parçasıdır.