

CMD üzerinden Kullanım Sağlamak:

Bunun için komut sistemine Mongosh yazmanız yeterlidir. Lakin CMD, Mongosh Shell'in komutlarını tanımayabilir. Kendim, aşağıda verdiğim videodaki adımları uygulayarak sorunu çözdüm.

LINK: <https://www.youtube.com/watch?v=xxMAhi8sxoE>

Veri tabanını silmek

```
test> show dbs
admin      40.00 KiB
config    108.00 KiB
local      40.00 KiB
test1      40.00 KiB
test> use test1
switched to db test1
test1> show collections
products
test1> db.dropDatabase()
{ ok: 1, dropped: 'test1' }
test1> show dbs
admin      40.00 KiB
config    108.00 KiB
local      40.00 KiB
test1>
```

'show dbs' komutu bize mevcut veritabanlarımızın listesini verir. 'use +veritabanıAdı' komutu ise o veritabanıyla işlem yapacağımızı belirtir.

```
test1> use acme
switched to db acme
acme> show dbs
admin      40.00 KiB
config    108.00 KiB
local      40.00 KiB
acme>
```

'use acme' komutu ile acme veritabanını kullanmaya çalıştık fakat böyle bir veritabanı olmadığından bu veritabanı oluşturuldu. Ardından veritabanlarını listeledik fakat 'acme' adında bir veri tabanı göremedik. Bunun nedeni acme için henüz bir collection ya da başka bir şeyin mevcut olmamasıdır.

```
acme> db
acme
acme>
```

Eğer kontrol sağlamak istersek bu veritabanının varlığını bu şekilde sorgulayabiliriz.

Hadi bir collection oluşturalım

```
acme> db.createCollection('posts')
{ ok: 1 }
acme> show collections
posts
acme>
```

db.createCollection komutu ile 'posts' adında bir collection oluşturduk ve show collections diyerek bu oluşturduğumuz collection'u listeleyebildik.

Şimdi bu collection'a insert işlemi yapalım

```
acme> db.posts.insert(
... {
...   title: 'Post One',
...   body : 'Body of post one',
...   category: 'News',
...   likes: 4,
...   tags: ['news', 'events'],
...
...   user: {
...     name: 'John Doe',
...     status: 'author'
...   },
...   date: Date()
... })
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63ec01aac5953e297be4c063") }
}
acme>
```

Birden fazla veri eklemek istiyorsak

insertMany ifadesini kullanarak birden fazla veriyi aynı anda ekleme imkanı elde ettik.

```
acme> db.posts.insertMany( [
...   {
...     title: 'Post Two',
...     body: 'Body of post two',
...     category: 'Technology',
...     date: Date()
...   },
...   {
...     title: 'Post Three',
...     body: 'Body of post three',
...     category: 'News',
...     date: Date()
...   },
...   {
...     title: 'Post four',
...     body: 'Body of post four',
...     category: 'Entertainment',
...     date: Date()
...   }
... ] )
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63ec0421c5953e297be4c064"),
    '1': ObjectId("63ec0421c5953e297be4c065"),
    '2': ObjectId("63ec0421c5953e297be4c066")
  }
}
```

Verileri listelemek istersek

```
acme> db.posts.find().pretty()
[
  {
    _id: ObjectId("63ec01aac5953e297be4c063"),
    title: 'Post One',
    body: 'Body of post one',
    category: 'News',
    likes: 4,
    tags: [ 'news', 'events' ],
    user: { name: 'John Doe', status: 'author' },
    date: 'Wed Feb 15 2023 00:48:26 GMT+0300 (GMT+03:00)'
  },
  {
    _id: ObjectId("63ec0421c5953e297be4c064"),
    title: 'Post Two',
    body: 'Body of post two',
    category: 'Technology',
    date: 'Wed Feb 15 2023 00:58:57 GMT+0300 (GMT+03:00)'
  },
  {
    _id: ObjectId("63ec0421c5953e297be4c065"),
    title: 'Post Three',
    body: 'Body of post three',
    category: 'News',
    date: 'Wed Feb 15 2023 00:58:57 GMT+0300 (GMT+03:00)'
  },
  {
    _id: ObjectId("63ec0421c5953e297be4c066"),
    title: 'Post four',
    body: 'Body of post four',
    category: 'Entertainment',
    date: 'Wed Feb 15 2023 00:58:57 GMT+0300 (GMT+03:00)'
  }
]
```

db.posts.find() komutu ile tüm verileri listeleyebiliyoruz. Bu komutun sonuna .pretty() komutunu da dahil edersek, verilerimize daha okunaklı bir şekilde erişim sağlayabiliyoruz.

Collection verileri içerisinde filtreleme yapmak

```
acme> db.posts.find({ category: 'News' }).pretty()
[
  {
    _id: ObjectId("63ec01aac5953e297be4c063"),
    title: 'Post One',
    body: 'Body of post one',
    category: 'News',
    likes: 4,
    tags: [ 'news', 'events' ],
    user: { name: 'John Doe', status: 'author' },
    date: 'Wed Feb 15 2023 00:48:26 GMT+0300 (GMT+03:00)'
  },
  {
    _id: ObjectId("63ec0421c5953e297be4c065"),
    title: 'Post Three',
    body: 'Body of post three',
    category: 'News',
    date: 'Wed Feb 15 2023 00:58:57 GMT+0300 (GMT+03:00)'
  }
]
acme>
```

'db.posts.find()' komutunun içerisine süslü parantezler dahilinde, istediğimiz koşulu belirtip sorgulamamızı yapabiliriz. Örnekte

{category: 'News'} ifadesi ile, kategorisi sadece News olan verilere ulaşmış olduk.

Pretty komutu ise, bizlere gelen çıktının daha okunur gözükmesini sağlıyor, fakat komut sisteminde her iki **kullanım da aynı** çıktıyı veriyor. İhtiyaç duyulması halinde kullanılabilir.

Sıralama işlemleri

```
acme> db.posts.find().sort({ title: 1 })
[
  {
    _id: ObjectId("63ec01aac5953e297be4c063"),
    title: 'Post One',
    body: 'Body of post one',
    category: 'News',
    likes: 4,
    tags: [ 'news', 'events' ],
    user: { name: 'John Doe', status: 'author' },
    date: 'Wed Feb 15 2023 00:48:26 GMT+0300 (GMT+03:00)'
  },
  {
    _id: ObjectId("63ec0421c5953e297be4c065"),
    title: 'Post Three',
    body: 'Body of post three',
    category: 'News',
    date: 'Wed Feb 15 2023 00:58:57 GMT+0300 (GMT+03:00)'
  },
  {
    _id: ObjectId("63ec0421c5953e297be4c064"),
    title: 'Post Two',
    body: 'Body of post two',
    category: 'Technology',
    date: 'Wed Feb 15 2023 00:58:57 GMT+0300 (GMT+03:00)'
  },
  {
    _id: ObjectId("63ec0421c5953e297be4c066"),
    title: 'Post four',
    body: 'Body of post four',
    category: 'Entertainment',
    date: 'Wed Feb 15 2023 00:58:57 GMT+0300 (GMT+03:00)'
  }
]
```

Sıralamak için `sort()` metodundan faydalanırız. Metodun içerisine süslü parantezler dahilinde sıralamak istediğimiz alanı ve karşısına da '1' veya '-1' değerini gireriz. 1, artan sırada sıralama (ASC) -1 ise azalan sırada sıralama (DESC) anlamına gelmektedir.

Verilen çıktıda 'Post four' başlıklı veri, ilk olarak listelenmesi gerekirken en sonda listelenmiştir. Bunun nedeni sort işleminin büyük-küçük harf duyarlı olmasıdır. Eğer verideki başlığı 'Post Four' şeklinde girmiş olsaydık ilk sırada gözükecekti.

Count Kullanımı

```
acme> db.posts.find( { category: 'News' } ).count()
2
acme>
```

Girilen sorgu, bize kategorisi 'News' olan verilerin kaç adet olduğunu geri döndürür. Görüldüğü üzere kategorisi 'News' olan 2 adet verimiz var imiş.

Limit Kullanımı

```
acme> db.posts.find().sort( { title: -1 } ).limit(2)
[
  {
    _id: ObjectId("63ec0421c5953e297be4c066"),
    title: 'Post four',
    body: 'Body of post four',
    category: 'Entertainment',
    date: 'Wed Feb 15 2023 00:58:57 GMT+0300 (GMT+03:00)'
  },
  {
    _id: ObjectId("63ec0421c5953e297be4c064"),
    title: 'Post Two',
    body: 'Body of post two',
    category: 'Technology',
    date: 'Wed Feb 15 2023 00:58:57 GMT+0300 (GMT+03:00)'
  }
]
```

`limit(2)` komutunu vererek, verileri azalan şekilde sıraladıktan sonra, SADECE 2 TANE değer listelenmesini sağlamış olduk.

forEach Kullanımı

```
acme> db.posts.find().forEach( function(doc) { print('Blog Post : ' + doc.title) } )
Blog Post : Post One
Blog Post : Post Two
Blog Post : Post Three
Blog Post : Post four

acme>
```

find() ile verileri aldıktan sonra, forEach kullanarak bu verilerin her biri üzerinde işlem yapabiliriz. forEach bizden bir adet fonksiyon isteyecektir. function(doc) diyerek, her bir satırdaki değeri temsilen artık 'doc' değişkenimizi kullanabiliriz. İsterseniz siz başka bir isim de verebilirsiniz.

Sonrasında ise süslü parantezler içerisinde her bir doc verisi için yapmak istediğimiz işlemi belirtiyoruz. Biz bu örnekte yazdırma işlemi yapacağımız için 'print' metodunu kullandık ve

'Blog Post : ' + doc.title komutu sayesinde her bir değerın başına 'Blog Post : ' ifadesi eklenmiş ve sonrasında da bu değerlerin başlıkları basılmış oldu.

Eğer tek bir spesifik değeri kullanacaksak

```
acme> db.posts.findOne( {category: 'News' } )
{
  _id: ObjectId("63ec01aac5953e297be4c063"),
  title: 'Post One',
  body: 'Body of post one',
  category: 'News',
  likes: 4,
  tags: [ 'news', 'events' ],
  user: { name: 'John Doe', status: 'author' },
  date: 'Wed Feb 15 2023 00:48:26 GMT+0300 (GMT+03:00)'
}
acme>
```

Örnekte findOne() metodu ile sadece tek bir satır dönüş almamız sağlandı. Metodun içerisinde bir filtreleme yaptık ve category değeri 'News' olanlardan seçim sağlamış olduk.

Update İşlemi

```
acme> db.posts.update(
...   {title:'Post Two'},
...   {$set:
...     {
...       title: 'Post Two',
...       body: 'New post 2 body',
...       date: Date()
...     }
...   },
...   {
...     upsert: true
...   }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

title: 'Post Two' olan verileri, \$set ile yeni değerlerine update ettik. Süslü parantezler içerisinde 3. bir parametre olarak 'upsert: true' ifadesini belirterek, eğer 'Post Two' başlığında değer yoksa, girdiğimiz değerler ile birlikte yeni bir değer insert itmesini söyledik.

```
acme> db.posts.find({title: 'Post Two'})
[
  {
    _id: ObjectId("63ec0421c5953e297be4c064"),
    title: 'Post Two',
    body: 'New post 2 body',
    category: 'Technology',
    date: 'Wed Feb 15 2023 10:51:09 GMT+0300 (GMT+03:00)'
  }
]
```

Increment işlemi

```
acme> db.posts.update({title: 'Post One'}, { $inc: { likes: 2} })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
acme> db.posts.find({title: 'Post One'})
[
  {
    _id: ObjectId("63ec01aac5953e297be4c063"),
    title: 'Post One',
    body: 'Body of post one',
    category: 'News',
    likes: 6,
    tags: [ 'news', 'events' ],
    user: { name: 'John Doe', status: 'author' },
    date: 'Wed Feb 15 2023 00:48:26 GMT+0300 (GMT+03:00)'
  }
]
acme>
```

‘Post One’ başlığına ait verideki likes değeri 4 iken, girdiğimiz increment komutu ile birlikte bu değeri 2 arttırdık. Veriyi tekrar yazdırdığımızda ise değerin 6 olarak güncellendiğini gördük.

\$inc → increment, \$set → set

Rename

```
acme> db.posts.update({title: 'Post One'}, {$rename: {likes: 'views'} })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
acme> db.posts.find({title: 'Post One'})
[
  {
    _id: ObjectId("63ec01aac5953e297be4c063"),
    title: 'Post One',
    body: 'Body of post one',
    category: 'News',
    tags: [ 'news', 'events' ],
    user: { name: 'John Doe', status: 'author' },
    date: 'Wed Feb 15 2023 00:48:26 GMT+0300 (GMT+03:00)',
    views: 6
  }
]
```

Gördüğünüz üzere ‘Post One’ başlıklı degree ait likes anahtar sözcüğü, views olarak **güncellendi**.

SİLME İŞLEMİ

```
acme> db.posts.remove({title: 'Post four'})
{ acknowledged: true, deletedCount: 1 }
acme> db.posts.find()
[
  {
    _id: ObjectId("63ec01aac5953e297be4c063"),
    title: 'Post One',
    body: 'Body of post one',
    category: 'News',
    tags: [ 'news', 'events' ],
    user: { name: 'John Doe', status: 'author' },
    date: 'Wed Feb 15 2023 00:48:26 GMT+0300 (GMT+03:00)',
    views: 6
  },
  {
    _id: ObjectId("63ec0421c5953e297be4c064"),
    title: 'Post Two',
    body: 'New post 2 body',
    category: 'Technology',
    date: 'Wed Feb 15 2023 10:51:09 GMT+0300 (GMT+03:00)'
  },
  {
    _id: ObjectId("63ec0421c5953e297be4c065"),
    title: 'Post Three',
    body: 'Body of post three',
    category: 'News',
    date: 'Wed Feb 15 2023 00:58:57 GMT+0300 (GMT+03:00)'
  }
]
```

Burada remove() metodu kullanarak veri silme işlemi gerçekleştirdik ve title: 'Post four' ifadesi ile eşleşen veri silinmiş oldu. Tekrar verileri listelediğimizde ise bu verinin listelenmediğini görüyoruz.

Böylelikle sağlamasını da yapmış bulunduk.

Alt Belgeler

```
acme> db.posts.update({title: 'Post One' },
...   {
...     $set: {
...       comments: [
...         {
...           user: 'Marry Williams',
...           body: 'Comment One',
...           date: Date()
...         },
...         {
...           user: 'Harry White',
...           body: 'Comment Two',
...           date: Date()
...         }
...       ]
...     }
...   }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

'Post One' başlıklı veriye, comments adında yeni bir alan oluşturduk ve burada 2 tane farklı veri girişi sağladık.

Dizideki Öğeye Göre Bul (\$elemMatch)

```
acme> db.posts.find({
...   comments: {
...     $elemMatch: {
...       user: 'Marry Williams'
...     }
...   }
... })
[
  {
    _id: ObjectId("63ec01aac5953e297be4c063"),
    title: 'Post One',
    body: 'Body of post one',
    category: 'News',
    tags: [ 'news', 'events' ],
    user: { name: 'John Doe', status: 'author' },
    date: 'Wed Feb 15 2023 00:48:26 GMT+0300 (GMT+03:00)',
    views: 6,
    comments: [
      {
        user: 'Marry Williams',
        body: 'Comment One',
        date: 'Wed Feb 15 2023 11:31:56 GMT+0300 (GMT+03:00)'
      },
      {
        user: 'Harry White',
        body: 'Comment Two',
        date: 'Wed Feb 15 2023 11:31:56 GMT+0300 (GMT+03:00)'
      }
    ]
  }
]
acme>
```

Sadece 'Post One' verisi görüntülendi çünkü comments: içeriği olan ve içerisinde 'Marry Williams' kullanıcısı bulunan tek veri 'Post One' verisidir.

Index Ekle

```
acme> db.posts.createIndex({title: 'text'})
title_text
```


Metin Ara

```
acme> db.posts.find({
...   $text: {
...     $search : "\"Post T\""
...   }
... })
[
  {
    _id: ObjectId("63ec0421c5953e297be4c065"),
    title: 'Post Three',
    body: 'Body of post three',
    category: 'News',
    date: 'Wed Feb 15 2023 00:58:57 GMT+0300 (GMT+03:00)'
  },
  {
    _id: ObjectId("63ec0421c5953e297be4c064"),
    title: 'Post Two',
    body: 'New post 2 body',
    category: 'Technology',
    date: 'Wed Feb 15 2023 10:51:09 GMT+0300 (GMT+03:00)'
  }
]
```

Not : “\ ve \” kaçış amaçlı kullanılmıştır. SQL sorgularındaki % operatörü gibi düşünebilirsiniz.

Karşılaştırma Operatörleri

```
acme> db.posts.find({ views : {$gte:6} })
[
  {
    _id: ObjectId("63ec01aac5953e297be4c063"),
    title: 'Post One',
    body: 'Body of post one',
    category: 'News',
    tags: [ 'news', 'events' ],
    user: { name: 'John Doe', status: 'author' },
    date: 'Wed Feb 15 2023 00:48:26 GMT+0300 (GMT+03:00)',
    views: 6,
    comments: [
      {
        user: 'Marry Williams',
        body: 'Comment One',
        date: 'Wed Feb 15 2023 11:31:56 GMT+0300 (GMT+03:00)'
      },
      {
        user: 'Harry White',
        body: 'Comment Two',
        date: 'Wed Feb 15 2023 11:31:56 GMT+0300 (GMT+03:00)'
      }
    ]
  },
  {
    _id: ObjectId("63ec0421c5953e297be4c064"),
    title: 'Post Two',
    body: 'New post 2 body',
    category: 'Technology',
    date: 'Wed Feb 15 2023 10:51:09 GMT+0300 (GMT+03:00)',
    views: 10
  }
]
```

NOT:

\$gte → büyük eşittir

\$gt → büyüktür

\$lte → küçük eşittir

\$lt → küçüktür

Bazı komutlar

Exit → çıkış yapar

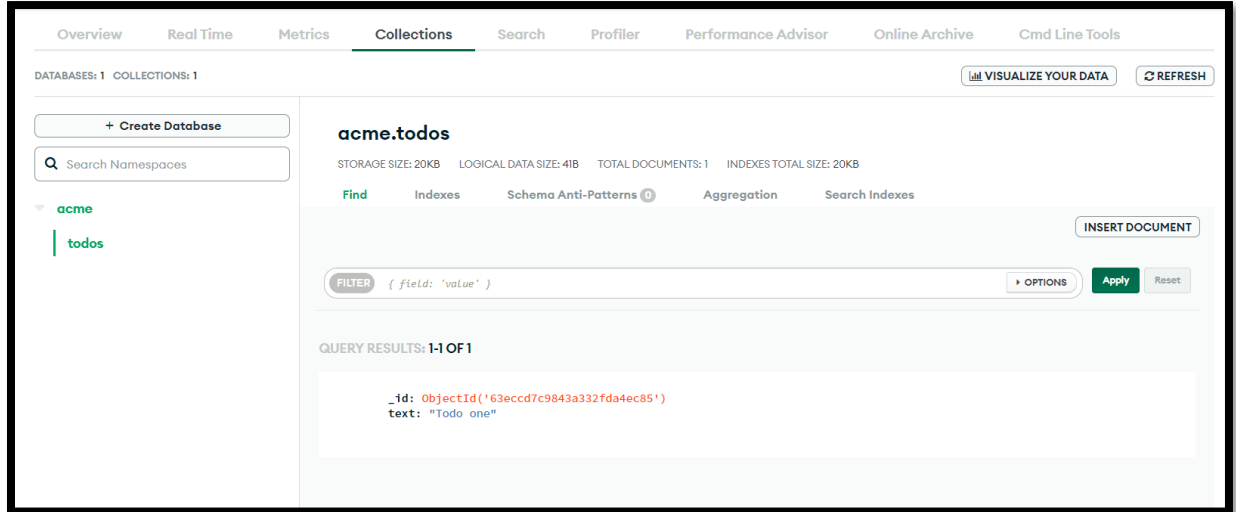
Buluttaki Veritabanına Bağlanmak ve İşlem Yapmak

```
C:\Users\kkes->mongosh "mongodb+srv://testcluster1.ujuouvl.mongodb.net/myFirstDatabase" --apiVersion 1 --username gorkem
Enter password: ****
Current Mongosh Log ID: 63eccce30dcfd33e95b77c01
Connecting to:      mongodb+srv://<credentials>@testcluster1.ujuouvl.mongodb.net/myFirstDatabase?appName=mongosh+1.7
.1
Using MongoDB:      5.0.14 (API Version 1)
Using Mongosh:       1.7.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-x8b5gb-shard-0 [primary] myFirstDatabase> show dbs
admin   336.00 KiB
local   1.93 GiB
Atlas atlas-x8b5gb-shard-0 [primary] myFirstDatabase> use acme
switched to db acme
Atlas atlas-x8b5gb-shard-0 [primary] acme> db.createCollection('todos')
{ ok: 1 }
Atlas atlas-x8b5gb-shard-0 [primary] acme> db.todos.insert({ text: 'Todo one' } )
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63eccd7c9843a332fda4ec85") }
}
Atlas atlas-x8b5gb-shard-0 [primary] acme> db.todos.find()
[ { _id: ObjectId("63eccd7c9843a332fda4ec85"), text: 'Todo one' } ]
Atlas atlas-x8b5gb-shard-0 [primary] acme> ,
```

Veritabanımızı kodlar ile oluşturduk. MongoDBCompass'ta olduğu gibi burada da arayüz aracılığıyla işlemler yapabiliriz. Fakat biz yine de terminal kullanarak işlemlerimizi yapmış olduk.



Gördüğünüz üzere todos collection'ü eklenmiş halde ve içerisinde 'Todo one' metnine sahip bir de document mevcut.