



```
import pymongo
```

```
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
```

```
mydb = myclient["fakulte"]
```

```
mycol = mydb["ogrenci"]
```

```
mydict = { "ogrenciAd": "Ayşe", "Sehir": "Tokat", "bolum": "yaz muh" }
```

```
x = mycol.insert_one(mydict)
```




<https://www.mongodb.com/cloud/atlas/register>



MongoDB[®]

Atlas



```
from pymongo.mongo_client import MongoClient
```

```
uri = "mongodb+srv://mrturkoglu23:<password>@cluster0.zuyakjp.mongodb.net/?retryWrites=true&w=majority"
```

```
# Create a new client and connect to the server
```

```
client = MongoClient(uri)
```

```
# Send a ping to confirm a successful connection
```

```
try:
```

```
    client.admin.command('ping')
```

```
    print("Pinged your deployment. You successfully connected to MongoDB!")
```

```
except Exception as e:
```

```
    print(e)
```





```
mydb = client["fakulte"]
```


```
mycol = mydb["ogrenci"]
```

```
data = { "ad": "Mehmet", "soyad": "Çınar" ,  
"bolum":"Yazılım Muhendisliği","Sehir":"Samsun","Yas":20}
```



```
x=mycol.insert_one(data)
```






```
data2=[{"ad": "Mehmet", "soyad": "Yıldız" ,"bolum":"Yazılım  
Muhendisliği","Sehir":"Samsun","Yas":20},  
        {"ad": "Ahmet", "soyad": "Tek" ,"bolum":"İnşaat  
Muhendisliği","Sehir":"Trabzon","Yas":22},  
        {"ad": "Ayşe", "soyad": "Çınar" ,"bolum":"Makine  
Muhendisliği","Sehir":"Samsun","Yas":25},  
        {"ad": "Fatma", "Sehir":"Ordu","Yas":20},  
        {"ad": "Çınar", "Sehir":"Elazığ","Yas":18}]
```



```
x=mycol.insert_many(data2)
```



```
data3={ "_id":1,"ad": "Eymen", "soyad": "Çınar", "bolum":"Yazılım  
Muhendisliği","Sehir":"Elazığ","Yas":20}
```

```
x=mycol.insert_one(data3)
```



Find Komutu

- `find_one()`, seçimdeki ilk oluşumu döndürür.

```
x=mycol.find_one()  
print(x)
```

- `find()`, seçimdeki tüm oluşumları döndürür.

```
for x in mycol.find():  
    print(x)
```



Koşullar Görmek istediğim bilgiler

```
for x in mycol.find({},{ { "_id": 0,"ad": 1, "soyad": 1 } }):  
    print(x)
```

Not: id dışında görmek istemediğimiz sütunlar için 0 yazmaya gerek yoktur.

Filtreleme Sorguları

- Memleketi Samsun olanları getir;
`mysorgu={"Sehir":"Samsun"}`
`mydoc=mycol.find(mysorgu)`
`for x in mydoc:`
 `print(x)`
- Yaşı 23'den büyük olanları getir;
`mysorgu1={"Yas": { "$gt": 23}}`
`mydoc=mycol.find(mysorgu1)`
`for x in mydoc:`
 `print(x)`

- Yaşı 18 ve 18'den düşük olanların isimlerini getir;

```
mysorgu1={"Yas": { "$lte": 18}}
```

```
for x in mycol.find(mysorgu1,{"_id":0,"ad":1}):  
    print(x)
```

- Bölümü Yazılım Mühendisi olmayanların isim ve soyadlarını getir;

```
mysorgu1={"bolum": { "$ne": "Yazılım Muhendisliği"}}
```

```
for x in mycol.find(mysorgu1,{"_id":0,"ad":1,"soyad":1}):  
    print(x)
```

- Memleketi Samsun ve Elazığ olanların bölümlerini getir;

```
mysorgu1={"Sehir": { "$in": ["Samsun","Elazığ"]}}
```

```
for x in mycol.find(mysorgu1,{"_id":0,"bolum":1}):  
    print(x)
```

```
mysorgu1={"Sehir": { "$nin": ["Samsun","Elazığ"]}}
```

```
for x in mycol.find(mysorgu1,{"_id":0,"bolum":1}):  
    print(x)
```


- Bolumu Yazılım Mühendisi ve Yaşı 20 olanları getir:

```
mysorgu1={"$and": [{"Yas":20}, {"bolum":"Yazılım Muhendisliği"}]}
```

```
for x in mycol.find(mysorgu1):  
    print(x)
```

```
mysorgu1={"$or": [{"Yas":20}, {"bolum":"Yazılım Muhendisliği"}]}
```

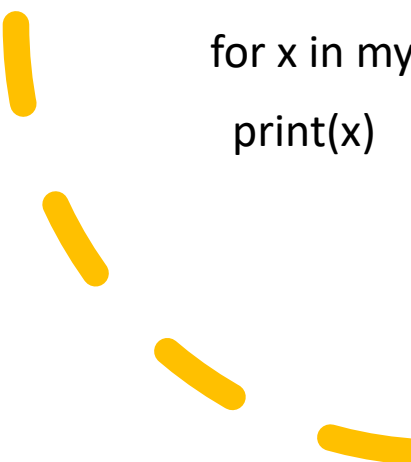
```
for x in mycol.find(mysorgu1):  
    print(x)
```




```
mysorgu1={"ad": { "$regex": "e"}}  
for x in mycol.find(mysorgu1):  
    print(x)
```

```
mysorgu1={"ad": { "$regex": "^M"}}  
for x in mycol.find(mysorgu1):  
    print(x)
```

```
mysorgu1={"ad": { "$regex": "e$"}}  
for x in mycol.find(mysorgu1):  
    print(x)
```




```
mysorgu={"ad":{"$regex":"e"}}  
for x in mycol.find(mysorgu1):
```



```
mysorgu1={"ad": { "$regex": "E$"}}  
for x in mycol.find(mysorgu1):  
    print(x)
```

```
mysorgu1={"ad": { "$regex": "E$","$options":"i"}}  
for x in mycol.find(mysorgu1):  
    print(x)
```





```
myresult = mycol.find().limit(2)
```

```
#print the result:
```

```
for x in myresult:
```

```
    print(x)
```



Sıralama (Sort) Sorguları

Ad sütündeki değerler dikkate alarak küçükten büyüğe doğru sıralama işlemi;

```
mydoc = mycol.find().sort("ad")
```

```
for x in mydoc:
```

```
    print(x)
```

```
mydoc = mycol.find().sort("Yas",-1)
```

```
for x in mydoc:
```

```
    print(x)
```



Update Sorguları

```
myquery = { "bolum": "Yazılım Mühendisliği" }
```

```
newvalues = { "$set": { "bolum": "Yazılım  
Mühendisliği" } }
```

```
mycol.update_one(myquery, newvalues)
```

```
for x in mycol.find():  
    print(x)
```



Adı «A» ile başlayanların yaşını 30 yapan güncelleme işlemi;

```
myquery = { "ad": { "$regex": "^A" } }
```

```
newvalues = { "$set": { "Yas": 30 } }
```

```
x = mycol.update_many(myquery, newvalues)
```

```
print(x.modified_count, "documents updated.")
```

```
for x in mycol.find():
```

```
    print(x)
```



```
myquery = { "ad": "Fatma" }
```

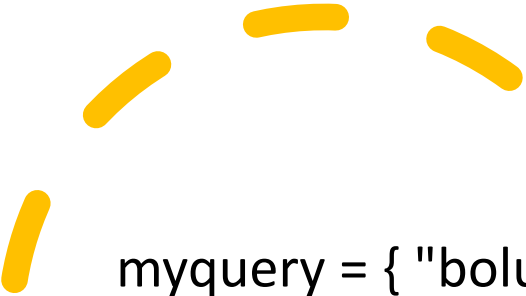
```
mycol.delete_one(myquery)
```

```
#print the result:
```

```
for x in mycol.find():  
    print(x)
```



Delete
Sorguları



```
myquery = { "bolum": { "$regex": "^Mak" } }

x = mycol.delete_many(myquery)

print(x.deleted_count, " documents deleted.")

#print the result:
for x in mycol.find():
    print(x)
```



Delete
Sorguları

- Delete All Documents in a Collection

```
x = mycol.delete_many({})
```

```
print(x.deleted_count, " documents deleted.")
```

```
#print the result:
```

```
for x in mycol.find():
```

```
    print(x)
```

- Delete Collection

```
mycol.drop()
```