Aggregation

db.Person.find({}) = db.Person.aggregate([])

db.Person.insertMany([- - _])

db.Person.aggregate([

  {stage1},
   {stage2},
      ⋮
    {stageN}

{stage 1}

* find() ve aggregate([]) eşittir. Sadece ve sadece aggregate([]) tek bir stage'e sahip olduğu zaman.

db.Person:aggregate([])

{ $group : { _id : "$age" }}

{ $group : { _id : "$company.location.county" }}

{ $group : { _id : "$name", total : { $sum : "$price" }}}

$MATCH STAGE

{ $match : { city : "New York" }}

{ $match : { age : { $gt : 25 }}}

{ $match : { $and : [{ gender : "female" }, { age : { $gte : 25 }}] }}

* $match → tüm operatörleri destekler.

```
db.Person.aggregate([
  {$match:{age:{$gt:25}}}
])
```

```
db.Person.aggregate([
  {$match:{isActive:true}}
])
```

```
db.Person.aggregate([
    {$match:{tags:{$size:3}}}
])
```

$GROUP → _id → zorunlu alan | # sort ve count
başka stage'de
yapılır.

{ $group : {_id:"$age" }}

{ $group: {_id : {age:"$age", gender : "$gender"}}}

```
db.Person.aggregate([
    {$group:{_id:"$gender"}}
])
```

```
db.Person.aggregate([
    {$group:{_id:"$age"}}   → Bütün yaş olasılıklarını
])                            getirir. (Gruplar)
```

```
db.Person.aggregate([
    {$group:{_id:"$name"}}   → Bütün isim olasılıklarını getirir.
])
```

```
db.Person.aggregate([
    {$group:{_id:"$company.location.country"}}
])          → distinct gibi çalışır.
```

```
db.Person.aggregate([
    {$group:{_id:{age:"$age",gender:"$gender"}}}
])          21      x     2 = 42  farklı olasılık olışur.
```

```
db.Person.aggregate([
    {$group:{_id:{eyeColor:"$eyeColor",favoriteFruit:"$favoriteFruit"}}}
])          3        x     3   = 9 olasılık
```

```
db.Person.aggregate([
    {$group:{_id:{eyeColor:"$eyeColor",   → 3
                                            x
```

```

```
db.Person.aggregate([
    {$group:{_id:{eyeColor:"$eyeColor",   → 3
        favoriteFruit:"$favoriteFruit",   → 3 ×
        age:"$age"}}}   → 21 ×
])   ───────────
                                              189 olasılık
```

```
db.Person.aggregate([
    {$match:{favoriteFruit:"apple"}},
    {$group:{_id:{eyeColor:"$eyeColor",age:"$age"}}}   → 63 olasılık
])
```

```
db.Person.aggregate([
    {$match:{gender:"female"}},
    {$group:{_id:{eyeColor:"$eyeColor",age:"$age"}}}   → 63 olasılık
])
```

```
db.Person.aggregate([
    {$match:{gender:"female"}},
    {$group:{_id:{eyeColor:"$eyeColor",age:"$age",gender:"$gender"}}}
])
```

⇓

```
{
    "_id" : {
        "eyeColor" : "brown",
        "age" : 28,
        "gender" : "female"
    }
}
```

```
{
    "_id" : {
        "eyeColor" : "brown",
        "age" : 32
    }
}
```

→ Sonuç 0 ,

```
db.Person.aggregate([
    {$group:{_id:{eyeColor:"$eyeColor",age:"$age"}}},
    {$match:{gender:"female"}}
])
```

↳ $match . $group'tan sonra gelir

```
])
```
↳ $match, $group'tan sonra gelir.

```
db.Person.aggregate([
  {$group:{_id:{eyeColor:"$eyeColor",age:"$age"}}},
  {$match:{"_id.age":{$gt:30}}}
])
```
↳ Bu şekilde kullanılınca $match, $group'tan sonra gelir.

```
db.Person.aggregate([
  {$group:{_id:{eyeColor:"$eyeColor",age:"$age"}}},
  {$match:{"_id.eyeColor":"brown"}}
])
```
* grup içindeki alanlardan birinde şart varsa, match aşağı yazılabilir.

$COUNT

```
db.Person.aggregate([                    →63//
  {$group:{_id:{eyeColor:"$eyeColor",age:"$age"}}},
  {$count:"eyeColorAndAge"}
])
```

```
db.Person.aggregate([              {
  {$count:"allDocuments"}    ⟹        "allDocuments" : 1000
])                                 }
```

```
db.Person.aggregate([
  {$group:{_id:{eyeColor:"$eyeColor",age:"$age"}}},
]).toArray().length    ⟹ Sayısını verir, count gibi
```

```
db.Person.aggregate([
  {$group:{_id:{eyeColor:"$eyeColor",age:"$age"}}},
```

```
db.Person.aggregate([
   {$group:{_id:{eyeColor:"$eyeColor",age:"$age"}}},
]).itcount()
```

⇒ count gibi sayıyı verir.

count → 0,21 sn → Server side
(toArray().length → 1,7 sn   } cevap süreci
(itCount() → 1,46 sn
  → Client-side
  Curso tüm belgede dolaşır. O yüzden uzun sürer.

db.Person aggregate ([{ $count: "total" }]) → 0,21 sn } Server
db.Person.find ({ }).count () ——————→ 0,21 sn } Side Count

```
db.Person.aggregate([
   {$group:{_id:"$company.location.country"}},
   {$count:"countriesCount"}
])
```

```
db.Person.aggregate([                    {
   {$group:{_id:"$age"}},        ⇒         "differentAge" : 21
   {$count:"differentAge"}                }
])
```

```
db.Person.aggregate([
   {$group:{_id:{eyeColor:"$eyeColor",gender:"$gender"}}},
   {$count:"eyeColorAndGender"}
])
```

```
{
   "eyeColorAndGender" : 6
```

```
{
   "eyeColorAndGender" : 6
}
```

```
db.Person.aggregate([
   {$match:{age:{$gte:25}}},
   {$group:{_id:{eyeColor:"$eyeColor",age:"$age"}}},
   {$count:"eyeColorAndAge"}
])
```

```
{
   "eyeColorAndAge" : 48
}
```

$SORT

```
db.Person.aggregate([
   {$sort:{name:1}},
])
```

```
db.Person.aggregate([
   {$sort:{age:-1}},
])
```

```
db.Person.aggregate([
   {$sort:{age:-1,gender:-1}},
])
```

```
db.Person.aggregate([
   {$sort:{age:-1,gender:-1,eyeColor:1}},
])
```

```
db.Person.aggregate([
   {$group:{_id:"$favoriteFruit"}},
   {$sort:{_id:1}},
])
```

```
db.Person.aggregate([
  {$group:{_id:"$age"}},
  {$sort:{_id:1}},
])

db.Person.aggregate([
  {$group:{_id:"$eyeColor"}},
  {$sort:{_id:-1}},
])

db.Person.aggregate([
  {$group:{_id:{eyeColor:"$eyeColor",favoriteFruit:"$favoriteFruit"}}},
  {$sort:{"_id.eyeColor":1,"_id.favoriteFruit":-1}},
])

db.Person.aggregate([
  {$match:{eyeColor:{$ne:"blue"}}},
  {$group:{_id:{eyeColor:"$eyeColor",favoriteFruit:"$favoriteFruit"}}},
  {$sort:{"_id.eyeColor":1,"_id.favoriteFruit":-1}},
])
```