

We gebruiken voor deze oefeningen de 'restaurant' collectie.
Laad het bestand restaurant.js in je MongoDB omgeving.
Het document vind je op het leerplatform ([restaurant.json](#)).
Maak hiervoor een nieuwe database horeca aan.

1. **Toon alle documenten in de collectie restaurant.**

Toon de velden restaurant_id, naam, borough (wijk), adres en keuken (cuisine), toon het veld _id niet.



The image shows the MongoDB Compass interface. In the 'Filter' section, there is a search icon and a dropdown menu. In the 'Project' section, the query `{restaurant_id:1,name:1,borough:1,cuisine:1,_id:0}` is entered. Below this, a code editor shows the equivalent MongoDB query:

```
db.getCollection('restaurant').find(
  {},
  {
    restaurant_id: 1,
    name: 1,
    borough: 1,
    cuisine: 1,
    _id: 0
  }
);
```

2. **Toon de eerste 10 restaurants in de wijk Staten Island.**

**Sorteer hiervoor alfabetisch op naam en neem enkel de restaurants in je overzicht op waarvoor de naam ingevuld is, dus ook geen lege string.
Toon alleen de naam.**



The image shows the MongoDB Compass interface. In the 'Filter' section, the query `{"$and":[{"borough":"Staten Island"}, {"name":{"$ne":""}}]}` is entered. In the 'Project' section, the query `{name:1, _id:0}` is entered. In the 'Sort' section, the query `{name:1}` is entered. In the 'Collation' section, the query `{ locale: 'simple' }` is entered. On the right, there are buttons for 'Generate query', 'Explain', 'Reset', and 'Find'. Below these, the 'Skip' value is 0 and the 'Limit' value is 10. Below the interface, a code editor shows the equivalent MongoDB query:

```
db.getCollection('restaurant').find(
  {
    $and: [
      { borough: 'Staten Island' },
      { name: { $ne: '' } }
    ]
  },
  { name: 1, _id: 0 }
).sort({ name: 1 })
.limit(10);
```

3. Toon de restaurants die geen enkele 'Iranian' keuken (cuisine) bereiden en waarvan de score hoger is dan 70 en op een breedtegraad liggen die lager is dan -65,75. Sorteer op naam.

🏠 restaurant	
	name String
1	"Anella"
2	"B.B. Kings"
3	"Baluchi'S Indian Food"
4	"Bella Napoli"
5	"Bistro Caterers"
6	"Brothers Fish Market"
7	"Cafe R"
8	"Cheikh Umar Futiyu Restauran..."
9	"Concrete Restaurant"

⇒ Etc...38 documents

```
db.getCollection('restaurant').find({
  $and: [
    { cuisine: { $ne: 'Iranian' } },
    { 'grades.score': { $gt: 70 } },
    { 'address.coord': { $lt: -65.75 } }
  ]
});
```

4. Toon de naam en borough (wijk) voor de restaurants die behoren tot de wijk Staten Island of Queens of Staten Island of Brooklyn. Sorteer eerst op naam en vervolgens op wijk. Toon geen restaurants waarvan de naam niet ingevuld is.
Tip : gebruik \$in en gebruik geen \$and

	name String	borough String
1	"#1 Garden Chinese"	"Brooklyn"
2	"(Lewis Drug Store) Locanda V..."	"Brooklyn"
3	"(Library) Four & Twenty Blac..."	"Brooklyn"
4	"1 Or 8"	"Brooklyn"
5	"10 Devoe"	"Brooklyn"
6	"100 Fun"	"Brooklyn"
7	"1001 Nights"	"Brooklyn"
8	"1001 Nights Cafe"	"Staten Island"
9	"101 Restaurant And Bar"	"Brooklyn"

etc => 7009

```
db.getCollection('restaurant').find(
  {
    borough: {
      $in: [
        'Staten Island',
        'Queens of Staten Island',
        'Brooklyn'
      ]
    },
    name: { $ne: '' }
  },
  { name: 1, _id: 0, borough: 1 }
)
.sort({ name: 1, borough: 1 });
```

5. **Toon de restaurants die een cijfer hebben met een score van 2 of een cijfer met een score van 6 en gevestigd zijn in de wijk Manhattan of Brooklyn, en hun keuken (cuisine) mag niet 'American' of 'Italian' zijn. Sorteer op naam, wijk.**

Dit geeft dan 2490 documents

```
db.restaurant.find(
  {
    "grades.score": {$in: [2, 6]},
    borough: {$in: ["Manhattan", "Brooklyn"]},
    cuisine: {$not: {$in: ['American', 'Italian']}}
  },
  {}
).sort(
  {
    name: 1,
    borough: 1
  }
)
```

6. Vanaf hier moeten aggregatie-methodes gebruikt worden. (group by)
Kijk hiervoor even naar :

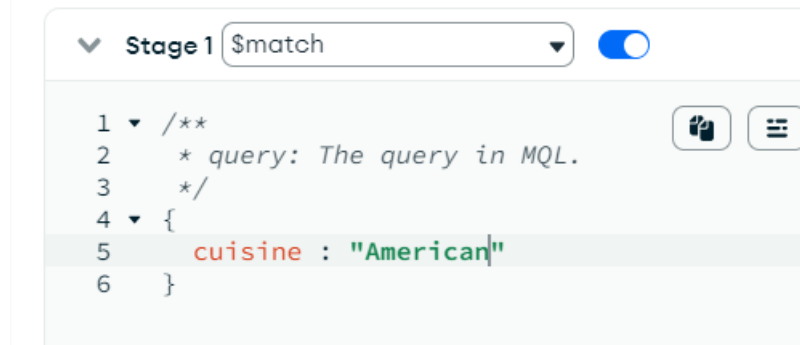
<https://www.mongodb.com/docs/manual/core/aggregation-pipeline/>

Toon het aantal Amerikaanse restaurants in elke borough (wijk).

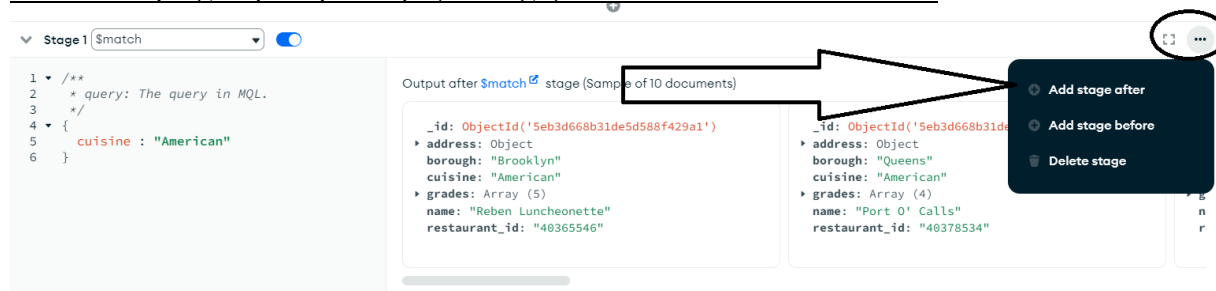
=> Klik op Aggregations.

Eerste stap : filter op Amerikaans.

Kies in Stage1 voor \$match.



Tweede stap : groepeer per wijk (borough) en tel het aantal restaurants.



Voeg een tweede stage toe.



Klik op export.

```
db.getCollection('restaurant').aggregate([
  { $match: { cuisine: 'American' } },
  {
    $group: {
      _id: '$borough',
      aantal: { $count: {} }
    }
  }
])
```

7. Toon de hoogste score voor elk restaurant.

Bekijk het veld grades. Grades is een array van documenten.

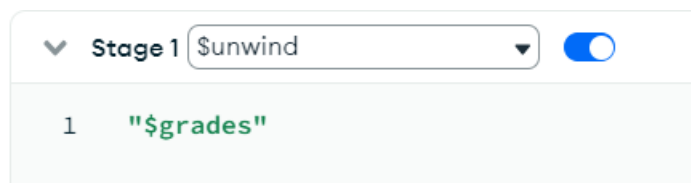
Wanneer je een aggregate function wil toepassen op het field score zal je in een eerste stap een “unwind” moeten doen van grades.

Je moet unwind enkel toepassen wanneer je dus max, min, avg, count of sum wil berekenen!!

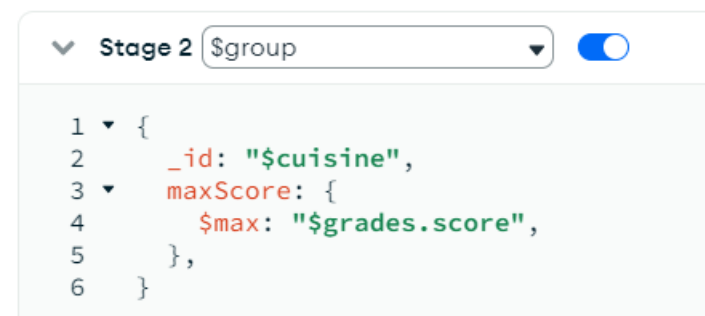
Daarom “ontrafelen” we de array van scores.

```
"grades": [
  {
    "date": {
      "$date": "2014-10-06T00:00:00.000Z"
    },
    "grade": "A",
    "score": 2
  },
  {
    "date": {
      "$date": "2013-10-10T00:00:00.000Z"
    },
    "grade": "A",
    "score": 12
  },
  {
    "date": {
      "$date": "2012-10-24T00:00:00.000Z"
    },
    "grade": "A",
    "score": 10
  }...
]
```

Eerste stap : unwind grades.



Tweede stap : groepeer per keuken en toon de hoogste score



```
db.getCollection('restaurant').aggregate([
  { $unwind: '$grades' },
  {
```

```

    $group: {
      _id: '$cuisine',
      maxScore: { $max: '$grades.score' }
    }
  }
]
);

```

```

{
  _id: 'Mexican',
  maxScore: 73
}
{
  _id: 'Steak',
  maxScore: 49
}
{
  _id: 'African',
  maxScore: 78
}
{
  _id: 'Seafood',
  maxScore: 77
}
{
  _id: 'Greek',
  maxScore: 39
}

```

Etc...

8. Toon de hoogste score voor de combinatie keuken (cuisine) en borough (wijk).

Group by op twee fields!

```

db.getCollection('restaurant').aggregate([
  {
    $unwind: "$grades",
  },
  {
    $group: {
      _id: {
        cuisine: "$cuisine",
        borough: "$borough",
      },
      maxScore: {
        $max: "$grades.score",
      },
    },
  },
]
);

```

```
    },  
  },  
]  
)
```

```
< {  
  _id: {  
    cuisine: 'Other',  
    borough: 'Queens'  
  },  
  maxScore: 44  
}  
{  
  _id: {  
    cuisine: 'Indian',  
    borough: 'Manhattan'  
  },  
  maxScore: 98  
}  
{  
  _id: {  
    cuisine: 'Mediterranean',  
    borough: 'Queens'  
  },  
  maxScore: 28  
}
```

Etc...

9. Toon de borough (wijk) met het hoogste aantal restaurants met het cijfer 'A' en een score groter dan of gelijk aan 90.

Tip : werk met limit (equivalent top 1 sql server).

▼ Stage 1 \$match ☒

```
1  ▼ {"grades.grade": 'A',  
2    "grades.score": {$gte: 90}}
```

Stage 2
\$group

```

1
2 {
3   _id: "$borough",
4   count: {
5     "$count": {}
6   }
7 }

```

Stage 3
\$sort

```

1 /**
2  * Provide any number of fields
3  */
4 {
5   count: -1
6 }

```

Stage 4
\$limit

```

1 /**
2  * Provide the number of documents
3  */
4 1

```

```

db.getCollection('restaurant').aggregate(
[
  {
    $match: {
      'grades.grade': 'A',
      'grades.score': { $gte: 90 }
    }
  },
  {
    $group: {
      _id: '$borough',
      count: { $count: {} }
    }
  },
  { $sort: { count: -1 } },
  { $limit: 1 }
],
{ maxTimeMS: 60000, allowDiskUse: true }
);

```