Aggregation Framework

```
>mongoimport -d test -c products < products.json
connected to: 127.0.0.1
2014-04-27T08:03:06.506+0200 imported 10 objects
```

Aggregation simple

Trouver le nombre de produit par manufacturer.

```
> db.products.aggregate([ {$group: { _id:"$manufacturer", num_products:{$sum:1} } } ])
{ "_id" : "Amazon", "num_products" : 2 }
{ "_id" : "Sony", "num_products" : 1 }
{ "_id" : "Samsung", "num_products" : 2 }
{ "_id" : "Google", "num_products" : 1 }
{ "_id" : "Apple", "num_products" : 4 }
```

Groupe

Nombre de products de chaque manufacturer pour chaque category.

```
> db.products.aggregate([
...   {$group:
...   {
... _id: {
...   "manufacturer":"$manufacturer",
...   "category" : "$category"},
... num_products:{$sum:1}
...   }
...   }
... ])
```

```
{ "_id" : { "manufacturer" : "Amazon", "category" : "Tablets" }, "num_products" : 2 }
{ "_id" : { "manufacturer" : "Apple", "category" : "Laptops" }, "num_products" : 1 }
{ "_id" : { "manufacturer" : "Google", "category" : "Tablets" }, "num_products" : 1 }
{ "_id" : { "manufacturer" : "Sony", "category" : "Laptops" }, "num_products" : 1 }
{ "_id" : { "manufacturer" : "Samsung", "category" : "Tablets" }, "num_products" : 1 }
{ "_id" : { "manufacturer" : "Samsung", "category" : "Cell Phones" }, "num_products" : 1 }
{ "_id" : { "manufacturer" : "Apple", "category" : "Tablets" }, "num_products" : 3 }
```

$sum

La somme des prix pour chaque mnufacturer.

```
> db.products.aggregate([
...    {$group:
...     {
... _id: {
...     "maker":"$manufacturer"
... },
... sum_prices:{$sum:"$price"}
...     }
...    }
... ])
```

```
{ "_id" : { "maker" : "Amazon" }, "sum_prices" : 328 }
{ "_id" : { "maker" : "Sony" }, "sum_prices" : 499 }
{ "_id" : { "maker" : "Samsung" }, "sum_prices" : 1014.98 }
{ "_id" : { "maker" : "Google" }, "sum_prices" : 199 }
{ "_id" : { "maker" : "Apple" }, "sum_prices" : 2296 }
```

Write an aggregation query to sum up the population (pop) by state and put the result in a field called population.

```
> db.zips.aggregate([{"$group":{"_id":"$state",  "population":{$sum:"$pop"}}}])
{ "_id" : "WV", "population" : 1793477 }
{ "_id" : "WA", "population" : 4866692 }
...
Type "it" for more
>
```

$avg

Trouver le prix moyen par category.

```
> db.products.aggregate([
...    {$group:
...     {
... _id: {
...     "category":"$category"
... },
... avg_price:{$avg:"$price"}
...     }
...    }
... ])
```

{ "_id" : { "category" : "Laptops" }, "avg_price" : 499 }
{ "_id" : { "category" : "Cell Phones" }, "avg_price" : 563.99 }
{ "_id" : { "category" : "Tablets" }, "avg_price" : 396.4271428571428 }

$addToSet

Lister les « category » par « manufacturer ». Cette liste contient des éléments uniques.

```
> db.products.aggregate([
...    {$group:
...     {
... _id: {
...     "maker":"$manufacturer"
... },
... categories:{$addToSet:"$category"}
...     }
...    }
... ])
```

{ "_id" : { "maker" : "Amazon" }, "categories" : ["Tablets"] }
{ "_id" : { "maker" : "Sony" }, "categories" : ["Laptops"] }
{ "_id" : { "maker" : "Samsung" }, "categories" : ["Tablets", "Cell Phones"] }
{ "_id" : { "maker" : "Google" }, "categories" : ["Tablets"] }
{ "_id" : { "maker" : "Apple" }, "categories" : ["Laptops", "Tablets"] }

$push

Lister les « category » par « manufacturer ».

```
> db.products.aggregate([
...    {$group:
...     {
... _id: {
...     "maker":"$manufacturer"
... },
... categories:{$push:"$category"}
...     }
...    }
... ])
```

{ "_id" : { "maker" : "Amazon" }, "categories" : ["Tablets", "Tablets"] }

{ "_id" : { "maker" : "Sony" }, "categories" : ["Laptops"] }
{ "_id" : { "maker" : "Samsung" }, "categories" : ["Cell Phones", "Tablets"] }
{ "_id" : { "maker" : "Google" }, "categories" : ["Tablets"] }
{ "_id" : { "maker" : "Apple" }, "categories" : ["Tablets", "Tablets", "Tablets", "Laptops"] }

$max et $min

Déterminer le prix « price » maximum de chaque « manufacturer ».

```
> db.products.aggregate([
...    {$group:
...     {
... _id: {
...     "maker":"$manufacturer"
... },
... maxprice:{$max:"$price"}
...     }
...    }
... ])
```

{ "_id" : { "maker" : "Amazon" }, "maxprice" : 199 }
{ "_id" : { "maker" : "Sony" }, "maxprice" : 499 }
{ "_id" : { "maker" : "Samsung" }, "maxprice" : 563.99 }
{ "_id" : { "maker" : "Google" }, "maxprice" : 199 }
{ "_id" : { "maker" : "Apple" }, "maxprice" : 699 }

$project

```
> db.products.findOne()
{
	"_id" : ObjectId("535c9d9a3a9816733480ee86"),
	"name" : "iPad 16GB Wifi",
	"manufacturer" : "Apple",
	"category" : "Tablets",
	"price" : 499
}
```

```
db.products.aggregate([
    {$project:
     {
         _id:0,
         'maker': {$toLower:"$manufacturer"},
         'details': {'category': "$category",
                     'price' : {"$multiply":["$price",10]}
                    },
         'item':'$name'
     }
    }
])
```

{ "maker" : "apple", "details" : { "category" : "Tablets", "price" : 4990 }, "item" : "iPad 16GB Wifi" }
{ "maker" : "apple", "details" : { "category" : "Tablets", "price" : 5990 }, "item" : "iPad 32GB Wifi" }
...

Importer zips.json

$match

Filtre n : 1

```
use test
> db.zips.findOne()
{
    "_id" : "35004",
    "city" : "ACMAR",
    "loc" : [
        -86.51557,
        33.584132
    ],
    "pop" : 6055,
    "state" : "AL"
}
```

Rechercher tous les documents avec « state » egal « NY »

```
db.zips.aggregate([
    {$match:
     {
         state:"NY"
     }
    }
])
```

{ "_id" : "06390", "city" : "FISHERS ISLAND", "loc" : [-72.017834, 41.263934], "pop" : 329, "state" : "NY" }
{ "_id" : "10001", "city" : "NEW YORK", "loc" : [-73.996705, 40.74838], "pop" : 18913, "state" : "NY" }

{ "_id" : "10002", "city" : "NEW YORK", "loc" : [-73.987681, 40.715231], "pop" : 84143, "state" :"NY" }

…

Rechercher la population totale par « city » de l'etat « state » de « NY »

```
db.zips.aggregate([
   {$match:  { state:"NY"  } },
   {$group:  { _id: "$city",  population: {$sum:"$pop"} } }
])
```

{ "_id" : "ELMIRA HEIGHTS", "population" : 7918 }
{ "_id" : "WELLSVILLE", "population" : 9645 }
{ "_id" : "WATKINS GLEN", "population" : 4584 }
{ "_id" : "VAN ETTEN", "population" : 1477 }
…

$sort

```
db.zips.aggregate([
   {$match:  { state:"NY"  } },
   {$group:  { _id: "$city",  population: {$sum:"$pop"} } },
   {$sort:  { population:-1  } }
])
```

{ "_id" : "BROOKLYN", "population" : 2300504 }
{ "_id" : "NEW YORK", "population" : 1476790 }
{ "_id" : "BRONX", "population" : 1209548 }
{ "_id" : "ROCHESTER", "population" : 396013 }
….

$limit et $skip

A utiliser avec sort sinon résultat non définie.

```
db.zips.aggregate([
   {$match:  { state:"NY"  } },
   {$group:  { _id: "$city",  population: {$sum:"$pop"} } },
   {$sort:  { population:-1  } },
   {$skip: 10},
   {$limit: 5}
])
```

{ "_id" : "ASTORIA", "population" : 165629 }
{ "_id" : "JACKSON HEIGHTS", "population" : 145967 }
{ "_id" : "FAR ROCKAWAY", "population" : 100646 }
{ "_id" : "RIDGEWOOD", "population" : 85732 }
{ "_id" : "BINGHAMTON", "population" : 83017 }

$first et $last

Find the largest city in every state.

Phase 1 :

```
db.zips.aggregate([
    /* Trouver la population de chaque ville de chaque */
    {$group:
     {
         _id: {state:"$state", city:"$city"},
         population: {$sum:"$pop"},
     }
    }
])
```

{ "_id" : { "state" : "WY", "city" : "THAYNE" }, "population" : 505 }
{ "_id" : { "state" : "WY", "city" : "SMOOT" }, "population" : 414 }
{ "_id" : { "state" : "WY", "city" : "LA BARGE" }, "population" : 606 }
….

Phase 2 :

```
db.zips.aggregate([
    /* Trouver la population de chaque ville de chaque */
    {$group:
     {
         _id: {state:"$state", city:"$city"},
         population: {$sum:"$pop"},
     }
    },
     /* trier comme sur population */
    {$sort:
     {"_id.state":1, "population":-1}
    }
])
```

Phase 3 :

```
db.zips.aggregate([
    /* get the population of every city in every state */
    {$group:
     {
         _id: {state:"$state", city:"$city"},
         population: {$sum:"$pop"},
     }
    },
     /* trier comme sur population */
    {$sort:
     {"_id.state":1, "population":-1}
    },
    /* grouper par etat, prendre le premier de chaque groupe */
    {$group:
     {
         _id:"$_id.state",
         city: {$first: "$_id.city"},
         population: {$first:"$population"}
     }
    }
])
```

{ "_id" : "WV", "city" : "HUNTINGTON", "population" : 75343 }
{ "_id" : "WA", "city" : "SEATTLE", "population" : 520096 }
{ "_id" : "VT", "city" : "BURLINGTON", "population" : 39127 }
{ "_id" : "VA", "city" : "VIRGINIA BEACH", "population" : 385080 }
{ "_id" : "UT", "city" : "SALT LAKE CITY", "population" : 186346 }