

#MDBlocal



PIPELINE POWER

DOING MORE WITH MONGODB AGGREGATION

Asya Kamsky
Lead Know-It-All

MongoDB Inc

@asya999
#askAsya

APPLICATIONS & DATA



APPLICATIONS & DATA



find()

&

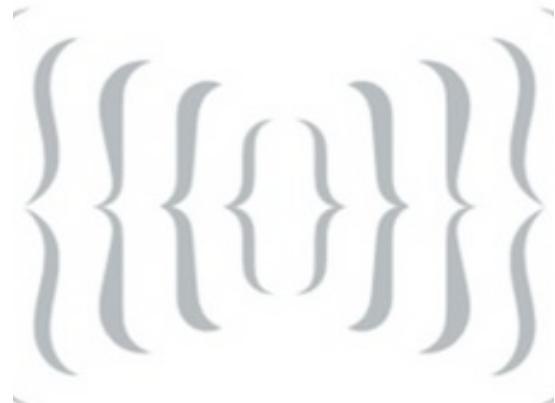
aggregate()



OPTIONS FOR ANALYTICS



pre-aggregate

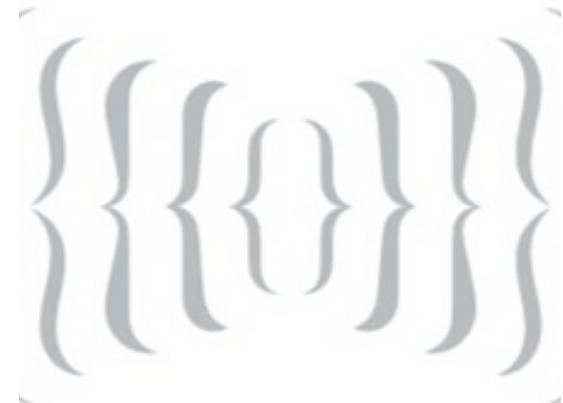


analyze
in MongoDB



analyze elsewhere

OPTIONS FOR ANALYTICS



analyze
in MongoDB



analyze elsewhere

Secure https://www.google.com/search?q=type+of+sa...&oq=type+of+sa...&gs=chrome_8957_183504&sourceid=ch... ⌂ ⌃ ⌁ ⌂ ⌃ ⌁

Show only Available nearby New items

Price Up to \$60 \$60 - \$150 \$150 - \$450 \$450 - \$1,000 Over \$1,000 \$ - \$10 \$ - \$50

Category Saw Blades Drill, Driver & Screwdriver Accessories

Brand baileigh bosch dewalt festool milwaukee hand trucks

More

Seller Acme Tools eBay HardwareAndTools.c Home Depot Neobits.com More

SKIL 3550-01 7 In. Wet Tile Saw with HydroLock System
\$999,999.99 from 25+ stores
★★★★★ 43 product reviews
SKIL
SKIL has the unique tools you need to deliver exceptional results. Skilsaws not only build projects, they build confidence.

17.25" SPECIAL 650 DI Semi Automatic Band Saw, Roller Table Support
\$38,995.00 from Circular Cold Saw Blades
17.25" SPECIAL 650 DI Semi Automatic Band Saw with Roller Table Support

Baileigh CNC Table Saw - STS-14120-CNC
\$24,245.00 from 3+ stores
Baileigh - Cabinet - 14 inch - 7.5 horsepower
The STS-14120-CNC table saw from Baileigh Industrial is based from our other sliding table saw designs, but adds a CNC twist. The CNC function ...

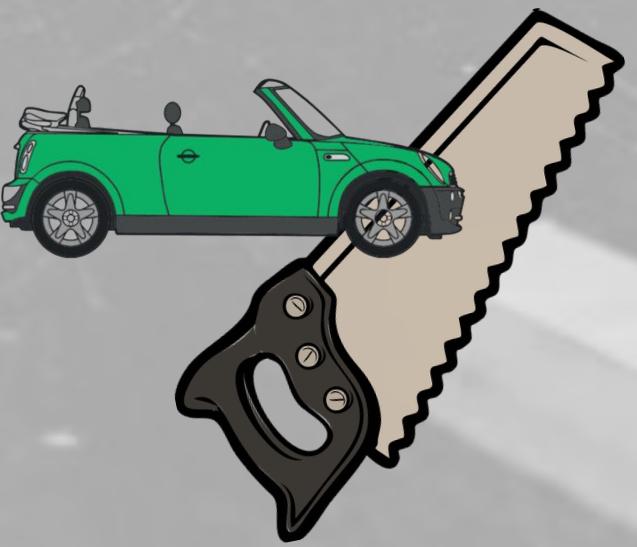
Safety Speed Cut Safety Speed ES5210M Panel Saw
\$19,949.00 from 2 stores
The ES5210M is the only panel saw in our lineup that features a carriage that traverses the full length of the machine frame. The traveling beam ...

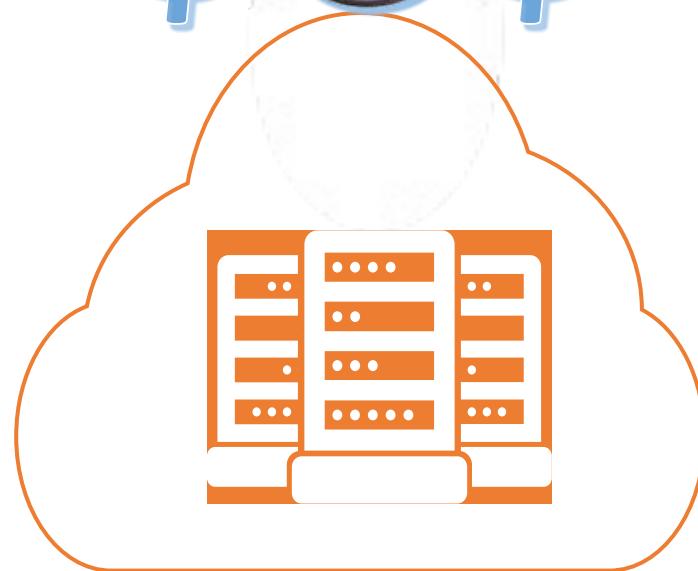
Hobart 6614-1 126" Vertical Meat Saw - 3 HP, 200-230/60/3V
\$8,481.55 from 4 stores
For precise cuts, simple operation and outstanding durability, look no further than this Hobart 6614-1 126" vertical meat saw! The 5 9/16" x 126" ...
Other size options: 142" (\$9,941)

Baileigh 21 IN x 60 IN Combination Woodworking Machine MF-3005
\$8,295.00 from 3 stores
Baileigh - 12 inch
The MF-3005 from Baileigh Industrial takes care of that problem. This 5 in 1 combination woodworking machine combines all of the most popular ...

Powermatic PM2000 Table Saw with Accu-Fence System, 10", 3 HP
\$7,179.98 from 20+ stores
Powermatic - Cabinet - 10 inch - 3 horsepower
The PM2000 table saw has an array of features and massive design. It has the heaviest cast iron base that eliminates table vibration. It has a ...

Skid Steer Tree Saw Attachment Fixed Boom High Flow 20-30 gpm
\$6,995.00 from Spartan Equipment





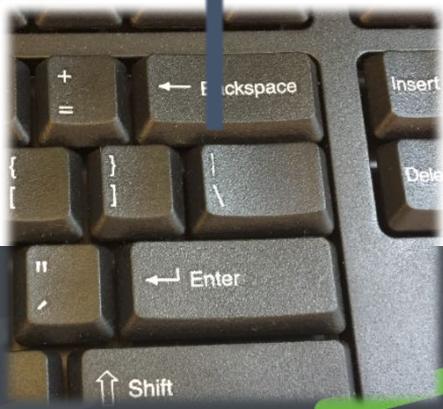
OPTIONS FOR ANALYTICS



ANALYTICS = AGGREGATION



PIPELINE



MONGODB.local
SYDNEY

#MDBlocal

PIPELINE

*nix command line pipe

```
ps ax | grep mongod | head 1
```



PIPELINE

MongoDB document pipeline

\$match | \$group | \$sort

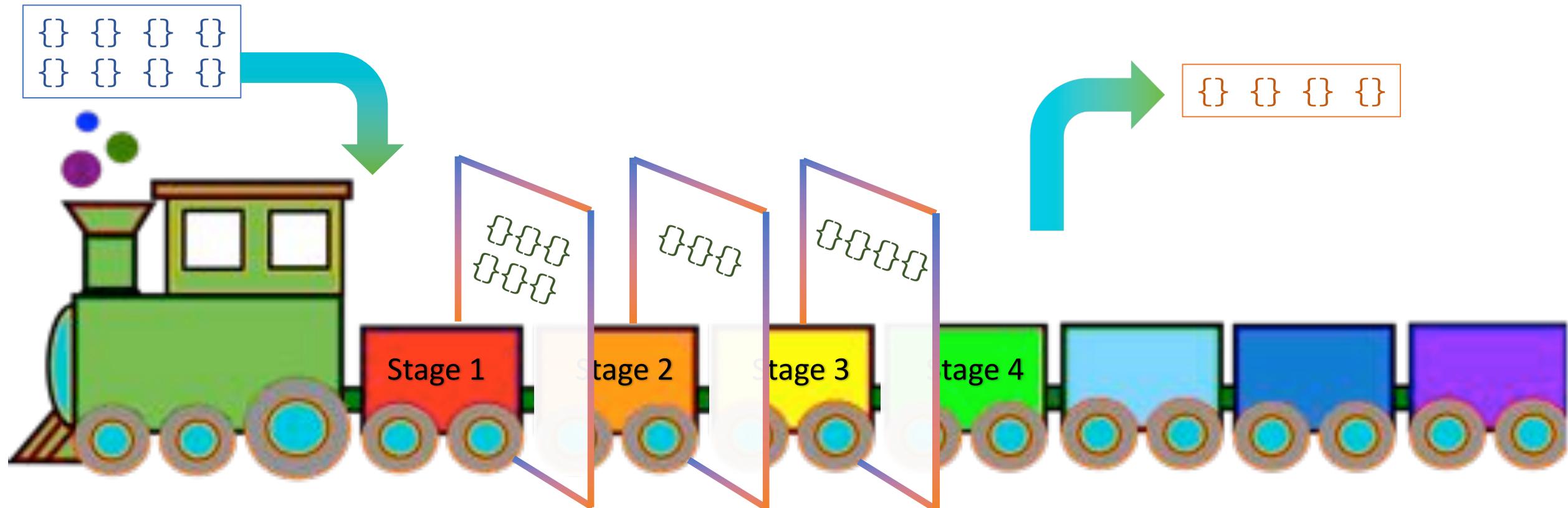


Input stream {} {} {} {} → Result {} {} ...

STAGES

START

Collection View Special stage



```
{"$stage":{ ... }}
```

DATA PIPELINE

```
{ title: "The Great Gatsby",
language: "English",
subjects: [
  "Long Island",
  "New York",
  "1920s" ],
{ title: "War and Peace",
language: "Russian",
subjects: [
  "Russia",
  "War of 1812",
  "Napoleon" ],
{ title: "Open City",
language: "English",
subjects: [
  "New York",
  "Harlem" ] },
```

```
{title: "The Great Gatsby",
language: "English",
subjects: [
  "Long Island",
  "New York",
  "1920s" ],
{title: "Open City",
language: "English",
subjects: [
  "New York"
  "Harlem" ] }
```

```
{title: "The Great Gatsby",
language: "English",
subjects: "Long Island"}
{title: "The Great Gatsby",
language: "English",
subjects: "New York"}
{title: "The Great Gatsby",
language: "English",
subjects: "1920s"}
{title: "Open City",
language: "English",
subjects: "New York"}
{title: "Open City",
language: "English",
subjects: "Harlem"}
```

```
{ _id: "Long Island",
count: 1 },
{ _id: "New York",
count: 2 },
{ _id: "1920s",
count: 1 },
{ _id: "Harlem",
count: 1 },
```

```
{"$unwind": "$subjects"} {"$sort": {"count": -1}}
{"$match": {"language": "English"} } {"$group": {"_id": "$subjects", "count": {"$sum": 1}} } {"$limit": 3} {"$project": "
```

\$match

\$unwind

\$group

\$sort

\$limit

\$project

\$skip

STAGES

\$sample
\$replaceRoot
\$out
\$skip
\$match
\$limit
\$sort
\$group
\$lookup
\$join
\$unwind
\$project
\$redact
\$count

\$match **\$geoNear**

\$limit

\$sample

\$skip

\$project

\$addFields

\$replaceRoot

\$lookup

\$graphLookup

\$sort

\$group

\$count

\$sortByCount

\$sample

\$replaceRoot

\$out

\$skip

\$match

\$limit

\$sort

\$group

\$project

\$indexStats

\$count

\$redact

\$bucket

\$bucketAuto

\$redact

\$facet

\$indexStats

\$out

\$collStats

\$geoNear

\$bucket

\$graphLookup

\$collStats

\$bucketAuto

\$indexStats

\$collStats

\$match
\$geoNear

\$limit
\$sample

\$skip

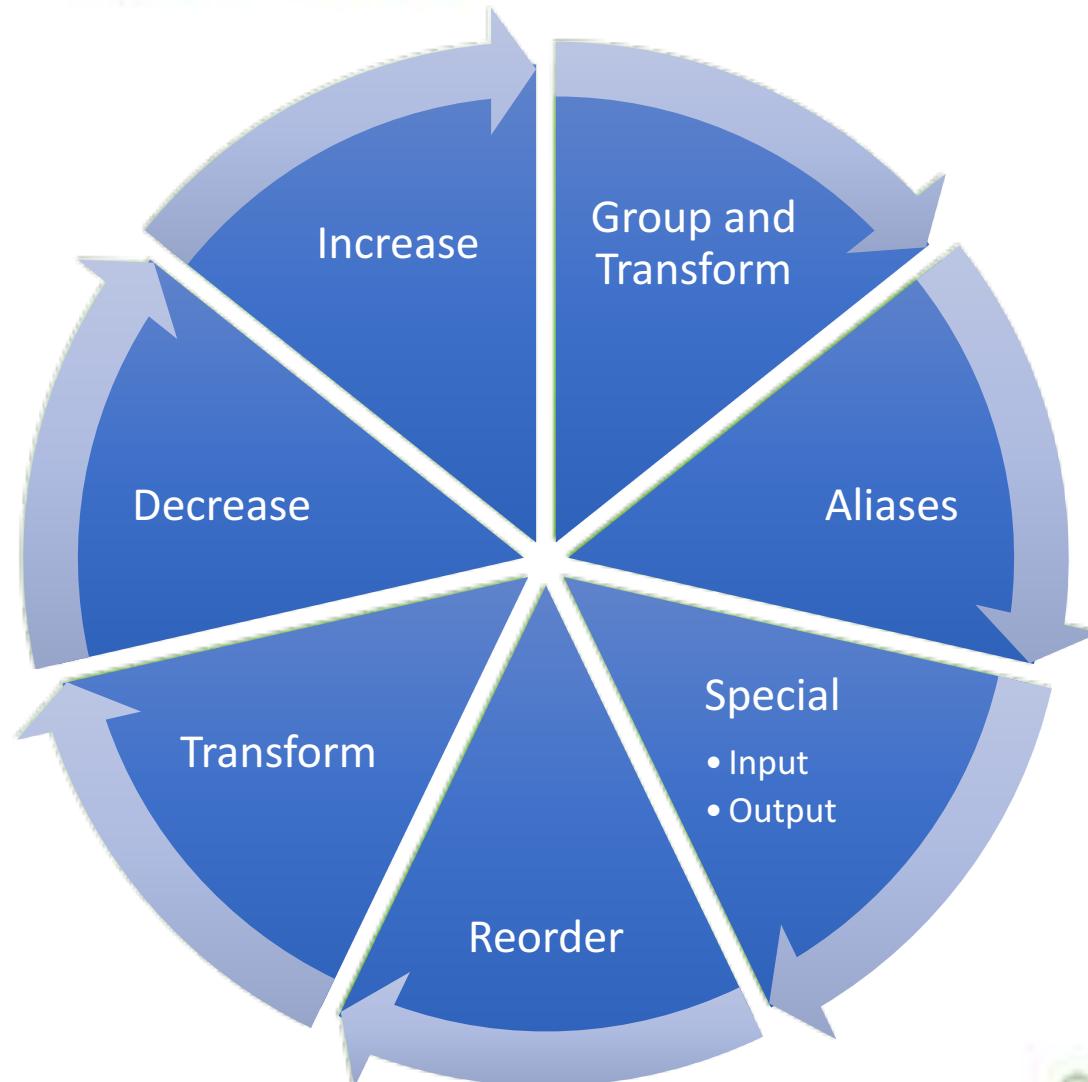
\$project
\$addFields

\$replaceRoot
\$lookup

\$graphLookup

\$unwind

\$sort



\$group

\$count
\$sortByCount

\$bucket
\$bucketAuto

\$facet \$redact

\$indexStats
\$scrollStats

\$out

```
{ title: "The Great Gatsby",
  language: "English",
  subjects: [
    "Long Island",
    "New York",
    "1920s" ],
{ title: "War and Peace",
  language: "Russian",
  subjects: [
    "Russia",
    "War of 1812",
    "Napoleon" ],
{ title: "Open City",
  language: "English",
  subjects: [
    "New York"
    "Harlem" ] },
```

```
{title: "The Great Gatsby",
language: "English",
subjects: [
  "Long Island",
  "New York",
  "1920s" ],
{title: "Open City",
language: "English",
subjects: [
  "New York"
  "Harlem" ] }
```

```
{title: "The Great Gatsby",
language: "English",
subjects: "Long Island"}
{title: "The Great Gatsby",
language: "English",
subjects: "New York"}
{title: "The Great Gatsby",
language: "English",
subjects: "1920s"}
{title: "Open City",
language: "English",
subjects: "New York"}
{title: "Open City",
language: "English",
subjects: "Harlem"}
```

```
{ _id: "Long Island",
count: 1 },
{ _id: "New York",
count: 2 },
{ _id: "1920s",
count: 1 },
{ _id: "Harlem",
count: 1 },
```

```
{"$unwind": "$subjects"} {"$sort": {"count": -1}}
{"$match": {"language": "English"} } {"$group": {"_id": "$subjects", "count": {"$sum": 1}} } {"$limit": 3} {"$project": "
```

\$match

\$unwind

\$group

\$sort

\$limit

\$project

\$skip

LET ME EXPLAIN...

```
db.books.aggregate([
  {$match:{"language":"English"}},
  {$unwind:"$subjects"},
  {$group:{_id:"$subjects",count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3}
],{explain:true})
```

```
{"stages" : [
  {"$cursor" : {"query" : { "language" : "English"},
    "fields" : { "subjects" : 1,"_id" : 0} ...
  }},
  {"$unwind" : {"path" : "$subjects"}, ...
  {"$group" : {"_id" : "$subjects","count" : [{"$sum" : {"$const" : 1}} ]},
  {"$sort" : {
    "sortKey" : {"count" : -1},
    "limit" : NumberLong(3)
  }}
] }
```

```
db.books.aggregate([
    {$match:{"language":"English"}},
    {$unwind:"$subjects"},
    {$group:{_id:"$subjects",count:{$sum:1}}},
    {$sort:{count:-1}},
    {$limit:3}
],{explain:true})
```

→

```
{"stages" : [
    {"$cursor" : {"query" : { "language" : "English"},  
                 "fields" : { "subjects" : 1,"_id" : 0} ...  
               },  
    {"$unwind" : {"path" : "$subjects"},  
    {"$group" : {"_id" : "$subjects","count" : {"$sum" : {"$const" : 1}} }},  
    {"$sort" : {  
        "sortKey" : {"count" : -1},  
        "limit" : NumberLong(3)  
      }  
  }  
]
```

```
db.books.aggregate([
    {$unwind:"$subjects"},
    {$group:{_id:"$subjects",count:{$sum:1}}},
    {$sort:{count:-1}},
    {$limit:3}
],{explain:true})

{"stages" : [
    {"$cursor" : {"query" : { },
                  "fields" : { "subjects" : 1,"_id" : 0} ...
                }},
    {"$unwind" : {"path" : "$subjects"}},
    {"$group" : {"_id" : "$subjects","count" : {"$sum" : {"$const" : 1}} }},
    {"$sort" : {
        "sortKey" : {"count" : -1},
        "limit" : NumberLong(3)
      }
  }
]} ] }
```

```
db.books.aggregate([
  {$unwind:"$subjects"},

  {$group:{_id:"$subjects",count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3}
],{explain:true})
```

```
db.books.aggregate([
    {$unwind:"$subjects"},
    {$match:{language:"English"}},
    {$group:{_id:"$subjects",count:{$sum:1}}},
    {$sort:{count:-1}},
    {$limit:3}
],{explain:true})

{"stages" : [
    {"$cursor" : {"query" : { "language" : "English"},  
                 "fields" : { "subjects" : 1,"_id" : 0} ...  
            },  
    {"$unwind" : {"path" : "$subjects"}},  
    {"$group" : {"_id" : "$subjects","count" : {"$sum" : {"$const" : 1}} }},  
    {"$sort" : {  
                "sortKey" : {"count" : -1},  
                "limit" : NumberLong(3)  
            }  
}] }
```

```
db.books.aggregate([
  {$unwind:"$subjects"},  

  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:"$subjects",count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3}
],{explain:true})  
  
{"stages": [  
    {"$cursor": {"query": { "language" : "English"},  
                 "fields" : { "subjects" : 1,"_id" : 0}}]  
  },  
  {"$unwind": {"path" : "$subjects"},  
  {"$match" : {"subjects" : {"$regex" : "ABC"}},  
  {"$group" : {"_id" : "$subjects","count" : [{"$sum" : {"$const" : 1}}]},  
  {"$sort" : {  
      "sortKey" : {"count" : -1},  
      "limit" : NumberLong(3)  
    }}  
] }
```

```
db.books.aggregate([
    {$unwind:"$subjects"},
    {$match:{language:"English",subjects:/^ABC/}},
    {$group:{_id:$subjects,count:{$sum:1}}},
    {$sort:{count:-1}},
    {$limit:3}
],{explain:true})

{"stages": [
    {"$cursor": {"query": {"language": "English"},  
        "fields": { "subjects": 1,"_id": 0} ...  
    }},  
    {"$unwind": {"path": "$subjects"}},  
    {"$match": {"subjects": {"$regex": "ABC"}}},  
    {"$group": {"_id": "$subjects","count": {"$sum": {"$const": 1}}}},  
    {"$sort": {  
        "sortKey": {"count": -1},  
        "limit": NumberLong(3)  
    }}  
] }
```

STREAMING VS BLOCKING

```
{ title: "The Great Gatsby",  
language: "English",  
subjects: [  
  "Long Island",  
  "New York",  
  "1920s" ],  
{ title: "War and Peace",  
language: "Russian",  
subjects: [  
  "Russia",  
  "War of 1812",  
  "Napoleon" ],  
{ title: "Open City",  
language: "English",  
subjects: [  
  "New York",  
  "Harlem" ] },
```

```
{title: "The Great Gatsby",  
language: "English",  
subjects: [  
  "Long Island",  
  "New York",  
  "1920s" ],  
  
{title: "Open City",  
language: "English",  
subjects: [  
  "New York"  
  "Harlem" ] }
```

```
{title: "The Great Gatsby",  
language: "English".  
subjects: "Long Island"}  
  
{title: "The Great Gatsby",  
language: "English",  
subjects: "New York"}  
  
{title: "The Great Gatsby",  
language: "English",  
subjects: "1920s"}  
  
{title: "Open City",  
language: "English",  
subjects: "New York"}  
  
{title: "Open City",  
language: "English",  
subjects: "Harlem"}
```

```
{ _id: "Long Island",  
count: 1 },  
  
{ _id: "New York",  
count: 1 },  
  
{ _id: "1920s",  
count: 1 },  
  
{ _id: "Harlem",  
count: 1 },
```

```
{ _id: "New York",  
count: 2 },  
  
{ _id:"Long Island",  
count: 1 },  
  
{ _id: "1920s",  
count: 1 },  
  
{ _id: "Harlem",  
count: 1 },
```

{"\$unwind": "\$subjects"}

{"\$sort": {"count": -1}}

{"\$match": {"language": "English"}}

{"\$group": {"_id": "\$subjects", "count": {"\$sum: 1}}}

**{"\$limit": 3}
 {"\$project":**

\$match

\$unwind

\$group

\$sort

\$limit

\$project

\$skip

\$match
\$geoNear

\$limit
\$sample

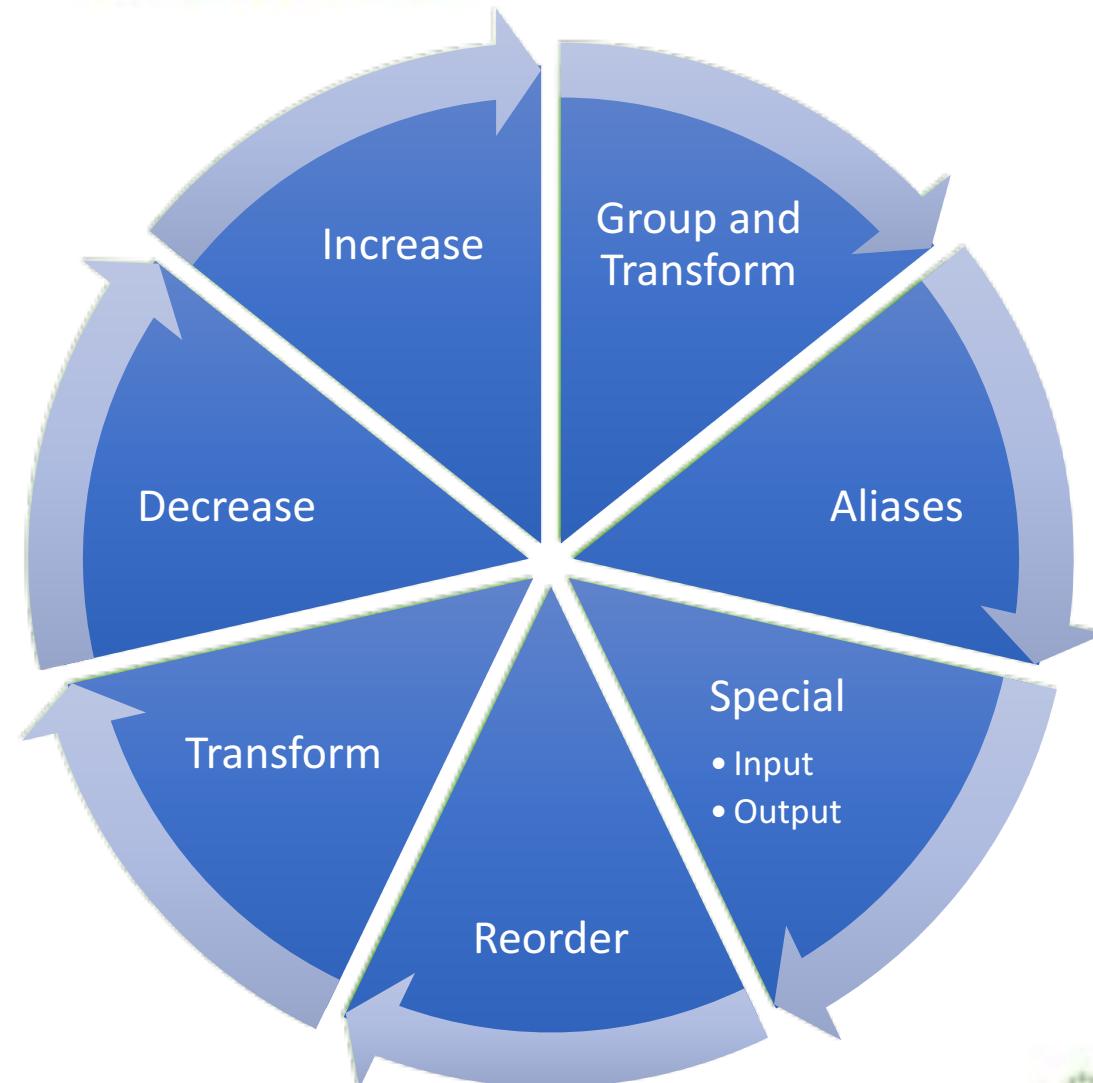
\$skip

\$project
\$addFields

\$replaceRoot
\$lookup

\$graphLookup

Sunwind



\$sort

\$group

\$count
\$sortByCount

\$bucket
\$bucketAuto

\$facet
\$redact

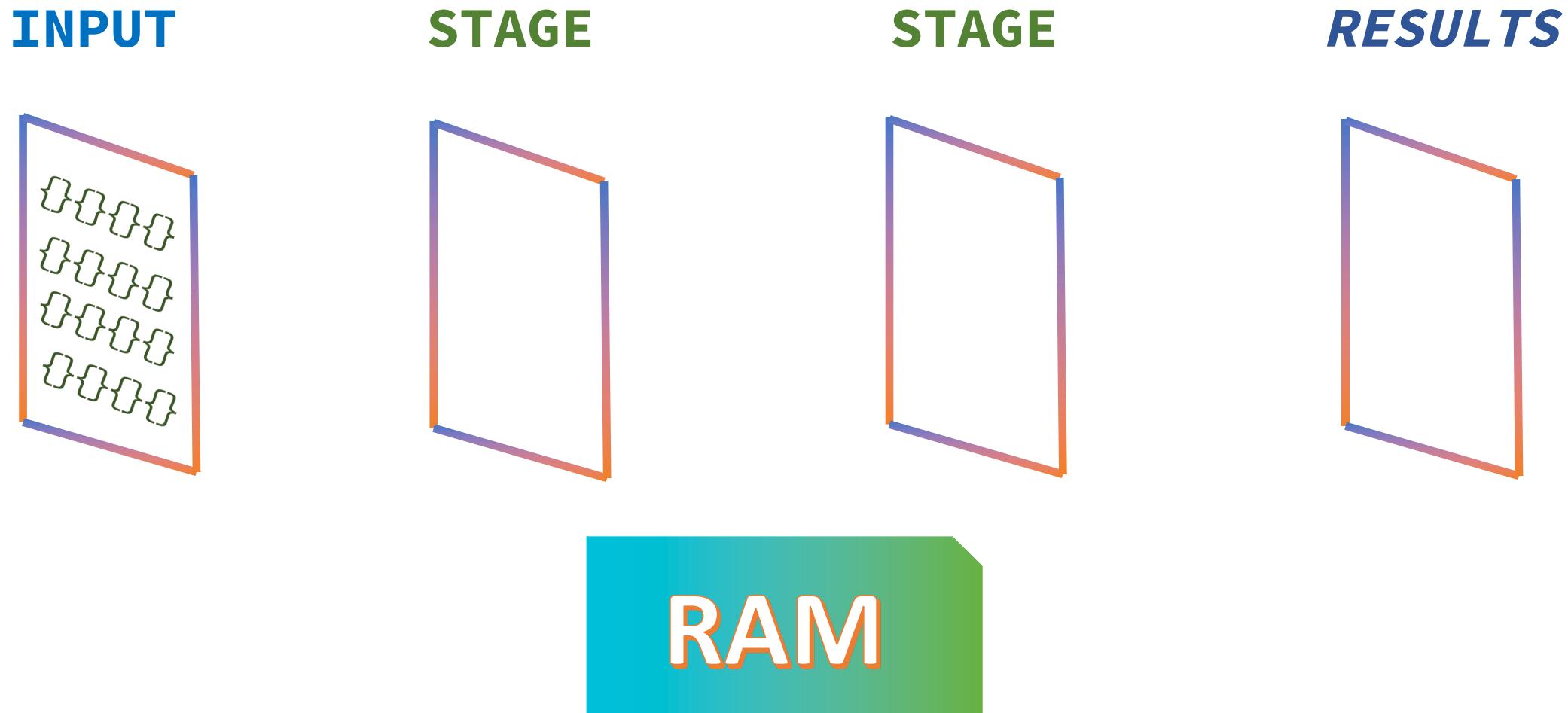
\$indexStats
\$collStats

\$out

STREAMING VS BLOCKING

RESOURCE USE

STREAMING RESOURCE USE

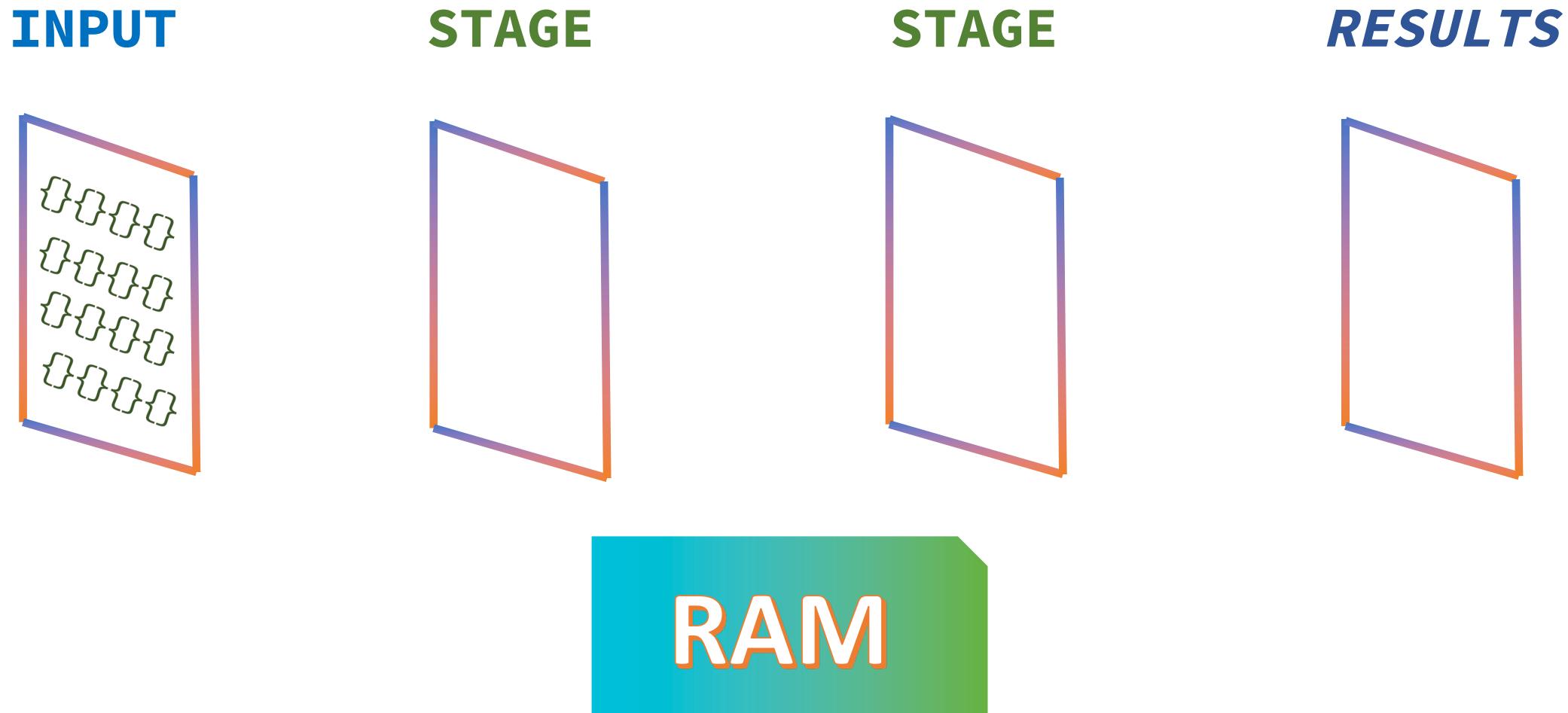


STREAMING RESOURCE USE



Each document is streamed through RAM

BLOCKING RESOURCE USE



BLOCKING RESOURCE USE



Sort with Limit:

$$\text{sizeOfAvgDoc} * \text{Limit} + \text{sizeOfAvgDoc}$$

Sort without Limit:

$$\text{sizeOfAllDocs}$$

Group:

$$\text{sizeOfAvgDoc} * \text{numberOfGroups} + \text{sizeOfAvgDoc}$$

STAGES, EXPRESSIONS, OPERATORS, ACCUMULATORS, OH MY!

EXPRESSIONS

`$zip`
`$sum`
`$scmp`
`$hour`
`$ssplit`
`$size`
`$mod`
`$avg`
`$isoWeek`
`$switch`
`$ceil`
`$lower`
`$month`
`$toLower`
`$add`
`$divide`
`$substrCP`
`$isoDayOfWeek`
`$millisecond`
`$abs`
`$type`
`$setEquals`
`$mergeObjects`
`$subtract`
`$dayOfMonth`
`$second`
`$log10`
`$dayOfYear`
`$strLenBytes`
`$reverseArray`
`$objectToArray`
`$let`
`$arrayToObject`
`$dateFromString`
`$filter`
`$setIntersection`
`$floor`
`$indexOfCP`
`$setDifference`
`$pow`
`$strcasecmp`
`$dateFromParts`
`$substrBytes`
`$concatArrays`
`$indexOfArray`
`$anyElement`
`$setIsSubset`
`$lne`
`$indexOfBytes`
`$dayOfWeek`
`$minute`
`$sqrt`
`$concat`
`$stoupper`
`$meta`
`$cond`
`$func`
`$reduce`
`$week`

\$substr
\$toUpperCase
\$indexOfCP
\$substrBytes
\$indexOfBytes
\$split
\$strLenBytes
\$concat
\$strLenCP
\$substrCP
\$toLower
\$cmp
\$monum
\$isoWeek
\$dayOfWeek
\$dateToParts
\$isoWeekYear
\$year
\$week
\$dateFromParts
\$dateFromString
\$isoDayOfWeek
\$millisecond
\$minute
\$dayOfMonth
\$dayOfYear
\$second
\$dateToString
\$dateToSet
\$add
\$ceil
\$trunc
\$log
\$divide
\$abs
\$mod
\$sub
\$mult
\$exp
\$floor
\$sqrt
\$avg
\$sum
\$log10
\$pow
\$max
\$last
\$addToSet
\$stdDevSamp
\$stdDevPop
\$min
\$push
\$type
\$lt
\$ifNull
\$let
\$mergeObjects
\$arrayToObject
\$objectToArray
\$switch
\$gte
\$cond
\$meta
\$literal

\$setIsSubset \$setEquals
\$arrayElemAt
\$indexOfArray
\$anyElementTrue
\$allElementsTrue
\$setIntersection
\$concatArrays
\$size
\$slice
\$zip
\$setDifference
\$map
\$reverseArray
\$filter
\$isArray
\$range

ARRAY EXPRESSIONS

ARRAY EXPRESSIONS

\$arrayElemAt

\$concatArrays

\$indexOfArray

\$isArray

\$size

\$range

\$reverseArray

\$map

\$reduce

\$filter

\$slice

\$zip

\$in

... plus all the set expressions

ARRAY EXPRESSIONS

`$arrayElemAt`

`$concatArrays`

`$indexOfArray`

`$isArray`

`$size`

`$range`

`$reverseArray`

`$map`

`$reduce`

`$filter`

`$slice`

`$zip`

`$in`

... plus all the set expressions

ARRAY EXPRESSIONS

\$map

input: array

output: array

\$reduce

input: array

output: anything

you

want

\$filter

input: array

output: subset of array

ARRAY EXPRESSIONS

```
"arr": [{"a":1}, {"a":99}, {"a":5}, {"a":3}]
```

```
{"$map": {  
  "input": "$arr",  
  "in": "$$this"  
}}
```

```
{"$map": {  
  "input": "$arr",  
  "as": "var",  
  "in": "$$var"  }}
```

```
{"$map": {  
  "input": "$arr",  
  "as": "eachElem",  
  "in": {"b": "$$eachElem.a"}  
} }
```

```
{"$map": {  
  "input": {"$range": [0, {"$size": "$arr"}]},  
  "as": "index",  
  "in": {"c": {"$arrayElemAt": ["$arr.a", "$$index"]}}  
} }
```

\$map

"a"	1
"a"	99
"a"	5
"a"	3



"a"	1
"a"	99
"a"	5
"a"	3

```
{"$map": {  
  "input": "$arr.a",  
  "as": "eachElem",  
  "in": {"b": "$$eachElem"} }}
```

"a"	1	"b"	1
"a"	99	"b"	99
"a"	5	"b"	5
"a"	3	"b"	3



0	"c"	1
1	"c"	99
2	"c"	5
3	"c"	3

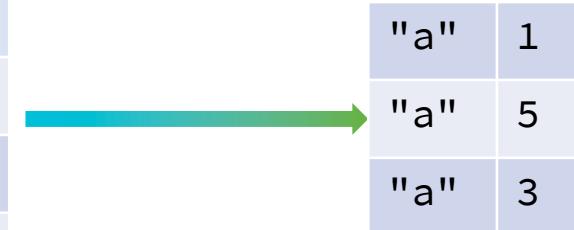
ARRAY EXPRESSIONS \$filter

```
"arr": [{"a":1}, {"a":99}, {"a":5}, {"a":3}]
```

```
{"$filter": {  
    "input": "$arr",  
    "cond": {"$lt": ["$$this.a", 10]}  
}}
```

```
{"$filter": {  
    "input": "$arr.a",  
    "as": "elem",  
    "cond": {"$lt": ["$$elem", 10]}  
} }
```

"a"	1
"a"	99
"a"	5
"a"	3

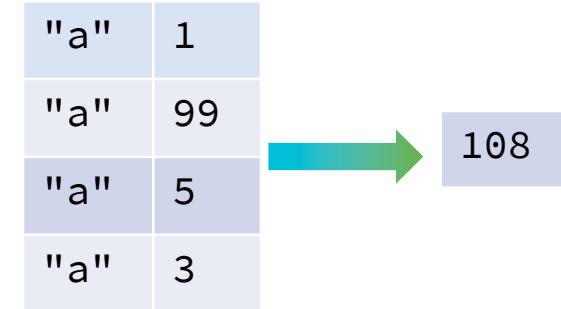


ARRAY EXPRESSIONS

```
"arr": [{"a":1}, {"a":99}, {"a":5}, {"a":3}]
```

```
{"$reduce": {  
    "input": "$arr",  
    "initialValue": 0,  
    "in": {"$add: ["$$value", "$$this.a"]}  
} }  
  
{$reduce":{  
    "input": "$arr",  
    "intialValue": [],  
    "in": {"$concatArrays": [  
        " $$this ",  
        " $$value"  
    ]}  
} }
```

\$reduce



READABILITY TIP



ENCAPSULATE COMPLEXITY

Functions for expressions

```
reverseArray = function(input) {  
    return {"$reduce":{  
        "input": input,  
        "intialValue": [],  
        "in": {"$concatArrays": [  
            [ "$$this" ],  
            "$$value"  
        ]}  
    }};  
};  
  
db.c.aggregate([ {"$addFields":{  
    "revArray":reverseArray("$origArray")  
} } ])
```

ENCAPSULATE COMPLEXITY

Functions for expressions

```
db.c.aggregate([ {"$addFields":{  
    "sortedArray":sortArray("$origArray")  
    sortArray= function( inputArray, sortField="", asc=false)  
    {  
        if( sortField == "" )  
            return inputArray;  
        else  
            return inputArray.sort(function(a,b){  
                if( asc )  
                    return a[sortField] - b[sortField];  
                else  
                    return b[sortField] - a[sortField];  
            });  
    } } ])
```

EXAMPLES



SCHEMA DISCOVERY

New Expressions (3.4.4, 3.6):

{"\$objectToArray": <object>}

=> array of k/v pairs

{"\$arrayToObject": <array of k/v pairs>}

=> object

Object:

```
{  
  "vid": {  
    "messageId": "bcd1d9",  
    "date": 1486546629585,  
    "status": "accept",  
    "comment": "hi"  
  }  
}
```

\$objectToArray:"\$vid"

```
[  
  { "k": "messageId",  
    "v": "bcd1d991"  
  },  
  { "k": "date",  
    "v": 1486546629585  
  },  
  { "k": "status",  
    "v": "accept"  
  },  
  { "k": "comment",  
    "v": "hi"  
  }  
]
```

\$objectToArray:"\$\$ROOT"

```
[ {  
    "k": "vid",  
    "v": {  
      "messageId": "bcd1d991",  
      "date": 1486546629585,  
      "status": "accept",  
      "comment": "hi"  
    }  
  }  
]
```

```
{  
  "vid": "072ade7d42d8",  
  "msgId": "bcd1d9",  
  "date": 1486546629585,  
  "status": "accept",  
  "comment": "hi"  
},  
{  
  "vid": "595d0a56cff2",  
  "msgId": "595d0a",  
  "date": 1486566646197,  
  "status": "reject",  
  "comment": "no good"  
},  
...
```

```
{"$project": {  
    "fields": {  
        "$objectToArray": "$$ROOT"  
    }  
}
```

```
{"fields": [  
    { "k": "vid", "v": "072ade7d42d8"},  
    { "k": "msgId", "v": "bcd1d991"},  
    { "k": "date", "v": 1486546629585 },  
    { "k": "status", "v": "accept"},  
    { "k": "comment", "v": "hi"}  
],  
{"fields": [  
    { "k": "vid", "v": "595d0a56cff2"},  
    { "k": "msgId", "v": "595d0a"},  
    { "k": "date", "v": 1486566646197},  
    { "k": "status", "v": "reject"},  
    { "k": "comment", "v": "no good"}  
],  
...  
}
```

\$project

```
{  
  "vid": "072ade7d42d8",  
  "msgId": "bcd1d9",  
  "date": 1486546629585,  
  "status": "accept",  
  "comment": "hi"  
},  
{  
  "vid": "595d0a56cff2",  
  "msgId": "595d0a",  
  "date": 1486566646197,  
  "status": "reject",  
  "comment": "no good"  
},  
...
```

```
{"$project": {  
    "fields": {  
        "$objectToArray": "$$ROOT"  
    }  
} },  
{"$project": {  
    "fields": {  
        "$map": {  
            "input": "$fields",  
            "in": {  
                "k": "$$this.k",  
                "t": {"$type": "$$this.v"}  
            }  
        }  
    }  
} },
```

```
{"fields": [  
    { "k": "vid", "v": "072ade7d42d8"},  
    { "k": "msgId", "v": "bcd1d991"},  
    { "k": "date", "v": 1486546629585 },  
    { "k": "status", "v": "accept"},  
    { "k": "comment", "v": "hi"}  
] },  
{"fields": [  
    { "k": "vid", "v": "595d0a56cff2"},  
    { "k": "msgId", "v": "595d0a"},  
    { "k": "date", "v": 1486566646197},  
    { "k": "status", "v": "reject"},  
    { "k": "comment", "v": "no good"}  
] },  
...
```

\$project

\$project

```
{  
  "vid": "072ade7d42d8",  
  "msgId": "bcd1d9",  
  "date": 1486546629585,  
  "status": "accept",  
  "comment": "hi"  
},  
{  
  "vid": "595d0a56cff2",  
  "msgId": "595d0a",  
  "date": 1486566646197,  
  "status": "reject",  
  "comment": "no good"  
},  
...
```

```
{"$project": {  
  "fields": {  
    "$map": {  
      "input": {  
        "$objectToArray": "$$ROOT",  
        "in": {  
          "k": "$$this.k",  
          "t": {"$type": "$$this.v"}  
        }  
      }  
    }  
  },  
  "$unwind": "$fields"}
```

```
{"fields": {"k": "vid", "t": "string"},  
 "fields": {"k": "msgId", "t": "string"},  
 "fields": {"k": "date", "t": double},  
 "fields": {"k": "status", "t": "string"},  
 "fields": {"k": "comment", "t": "string"},  
 "fields": {"k": "vid", "t": "string"},  
 "fields": {"k": "msgId", "t": "string"},  
 "fields": {"k": "date", "t": double},  
 "fields": {"k": "status", "t": "string"},  
 "fields": {"k": "comment", "t": "string"},  
 ...}
```

\$project

\$unwind

```
{  
  "vid": "072ade7d42d8",  
  "msgId": "bcd1d9",  
  "date": 1486546629585,  
  "status": "accept",  
  "comment": "hi"  
},  
{  
  "vid": "595d0a56cff2",  
  "msgId": "595d0a",  
  "date": 1486566646197,  
  "status": "reject",  
  "comment": "no good"  
},  
...
```

```
{"$project": {  
  "fields": {  
    "$map": {  
      "input": {  
        "$objectToArray": "$$ROOT"  
      },  
      "in": {  
        "k": "$$this.k",  
        "t": {"$type": "$$this.v"}  
      }  
    }  
  }},  
  {"$unwind": "$fields"},  
  {"$group": {  
    "_id": "$fields.k",  
    "types": {  
      "$addToSet": "$fields.t"  
    }  
  }}  
}
```

```
{  "_id": "comment", "types": ["string"]}  
{  "_id": "status", "types": ["string"]}  
{  "_id": "date", "types": ["double"]},  
{  "_id": "msgId", "types": ["string"]}  
{  "_id": "vid", "types": ["string"]},  
...
```

\$project

\$unwind

\$group

```
{  
  "vid": "072ade7d42d8",  
  "msgId": "bcd1d9",  
  "date": 1486546629585,  
  "status": "accept",  
  "comment": "hi"  
},  
,  
  {"vid": "595d0a56cff2",  
   "msgId": "595d0a",  
   "date": 1486566646197,  
   "status": "reject",  
   "comment": "no good"  
},  
...  
}
```

```
{"$project": {  
  "fields": {  
    "$map": {  
      "input": {  
        "$objectToArray": "$$ROOT"  
      },  
      "in": {  
        "k": "$$this.k",  
        "t": {"$type": "$$this.v"}  
      }  
    }  
  }  
},  
  {"$unwind": "$fields"},  
  {"$group": {  
    "_id": {  
      "k": "$fields.k",  
      "t": "$fields.t"  
    },  
    "c": {"$sum": 1}  
  }}  
}
```

```
{"_id": {"k": "status", "t": "string"},  
 "c": 100  
},  
 {"_id": {"k": "comment", "t": "string"},  
 "c": 100  
},  
 {"_id": {"k": "date", "t": "double"},  
 "c": 100  
},  
 {"_id": {"k": "msgId", "t": "string"},  
 "c": 100  
},  
 {"_id": {"k": "vid", "t": "string"},  
 "c": 100  
}  
...  
}
```

\$project

\$unwind

\$group

SCHEMA TRANSFORMATION

```
{"body": {  
  "VMESSAGE": {  
    "072ade7d42d8": {  
      "msgId": "bcd1d9",  
      "date": 1486546629585,  
      "status": "accept",  
      "comment": "hi"  
    } } } },  
{"body": {  
  "VMESSAGE": {  
    "595d0a56cff2": {  
      "msgId": "595d0a",  
      "date": 1486566646197,  
      "status": "reject",  
      "comment": "no good"  
    } } } },  
{"body": {  
  "VMESSAGE": {  
    "52ffd09bf5b5": {  
      "msgId": "1dadce",  
      "date": 1486568943752,  
      "status": "accept"  
    } } } }
```

```
{"message": {  
  "vid": "072ade7d42d8",  
  "msgId": "bcd1d9",  
  "date": 1486546629585,  
  "status": "accept",  
  "comment": "hi"  
} },  
{"message": {  
  "vid": "595d0a56cff2",  
  "msgId": "595d0a",  
  "date": 1486566646197,  
  "status": "reject",  
  "comment": "no good"  
} },  
{"message": {  
  "vid": "52ffd09bf5b5",  
  "msgId": "1dadce",  
  "date": 1486568943752,  
  "status": "accept"  
} }
```

\$project

\$addFields

```
{"body": {  
  "VMESSAGE": {  
    "072ade7d42d8": {  
      "msgId": "bcd1d9",  
      "date": 1486546629585,  
      "status": "accept",  
      "comment": "hi"  
    } } } },  
{"body": {  
  "VMESSAGE": {  
    "595d0a56cff2": {  
      "msgId": "595d0a",  
      "date": 1486566646197,  
      "status": "reject",  
      "comment": "no good"  
    } } } },  
{"body": {  
  "VMESSAGE": {  
    "52ffd09bf5b5": {  
      "msgId": "1dadce",  
      "date": 1486568943752,  
      "status": "accept"  
    } } } }
```

Object "body.VMESSAGE":

```
{  
  "072ade7d42d8": {  
    "messageId": "bcd1d9",  
    "date": 1486546629585,  
    "status": "accept",  
    "comment": "hi"  
  } }  
  
$objectToArray:"$body.VMESSAGE"  
[ {  
  "k": "072ade7d42d8",  
  "v": {  
    "messageId": "bcd1d991",  
    "date": 1486546629585,  
    "status": "accept",  
    "comment": "hi"  
  } }]  
  
Need array: [  
  {"k": "vid", "v": "072ade7d42d8"},  
  {"k": "messageId", "v": "bcd1d9"},  
  {"k": "date", "v": 1486546629585},  
  {"k": "status", "v": "accept"},  
  {"k": "comment", "v": "hi"}]
```

```
{"message": {  
  "vid": "072ade7d42d8",  
  "msgId": "bcd1d9",  
  "date": 1486546629585,  
  "status": "accept",  
  "comment": "hi"} },  
{"message": {  
  "vid": "595d0a56cff2",  
  "msgId": "595d0a",  
  "date": 1486566646197,  
  "status": "reject",  
  "comment": "no good"} },  
{"message": {  
  "vid": "52ffd09bf5b5",  
  "msgId": "1dadce",  
  "date": 1486568943752,  
  "status": "accept"} }
```

\$addFields

```
{"body": {  
  "VMESSAGE": {  
    "072ade7d42d8": {  
      "msgId": "bcd1d9",  
      "date": 1486546629585,  
      "status": "accept",  
      "comment": "hi"  
    } } },  
  {"body": {  
    "VMESSAGE": {  
      "595d0a56cff2": {  
        "msgId": "595d0a",  
        "date": 1486566646197,  
        "status": "reject",  
        "comment": "no good"  
      } } },  
  {"body": {  
    "VMESSAGE": {  
      "52ffd09bf5b5": {  
        "msgId": "1dadce",  
        "date": 1486568943752,  
        "status": "accept"  
      } } } }
```

```
  {"$addFields": {  
    "message": {  
      "$arrayToObject": ...  
      {"$objectToArray": "$body.VMESSAGE"}  
    } } }
```

```
  {"message": {  
    "vid": "072ade7d42d8",  
    "msgId": "bcd1d9",  
    "date": 1486546629585,  
    "status": "accept",  
    "comment": "hi" } },  
  {"message": {  
    "vid": "595d0a56cff2",  
    "msgId": "595d0a",  
    "date": 1486566646197,  
    "status": "reject",  
    "comment": "no good" } },  
  {"message": {  
    "vid": "52ffd09bf5b5",  
    "msgId": "1dadce",  
    "date": 1486568943752,  
    "status": "accept" } }
```

\$addFields

```
{"body": {  
  "VMESSAGE": {  
    "072ade7d42d8": {  
      "msgId": "bcd1d9",  
      "date": 1486546629585,  
      "status": "accept",  
      "comment": "hi"  
    } } } },  
{"body": {  
  "VMESSAGE": {  
    "595d0a56cff2": {  
      "msgId": "595d0a",  
      "date": 1486566646197,  
      "status": "reject",  
      "comment": "no good"  
    } } } },  
{"body": {  
  "VMESSAGE": {  
    "52ffd09bf5b5": {  
      "msgId": "1dadce",  
      "date": 1486568943752,  
      "status": "accept"  
    } } } }
```

```
{"$addFields": {  
  "bvm": {"$objectToArray": "$body.VMESSAGE"}  
} },  
{"$addFields": {  
  "bvm2": {"$arrayElemAt": ["$bvm", 0]}  
} },  
{"$addFields": {  
  "bvm": {"$objectToArray": "$bvm2.v"}  
} },  
{"$addFields": {  
  "msgarr": {  
    }  
} },  
{"$addFields": {  
  "message": {"$arrayToObject": "$msgarr"}  
} }
```

```
{"message": {  
  "vid": "072ade7d42d8",  
  "msgId": "bcd1d9",  
  "date": 1486546629585,  
  "status": "accept",  
  "comment": "hi"  
} },  
{"message": {  
  "vid": "595d0a56cff2",  
  "msgId": "595d0a",  
  "date": 1486566646197,  
  "status": "reject",  
  "comment": "no good"  
} },  
{"message": {  
  "vid": "52ffd09bf5b5",  
  "msgId": "1dadce",  
  "date": 1486568943752,  
  "status": "accept"  
} }
```

\$addFields

\$addFields

\$addFields

\$addFields

\$addFields

```
{"body": {  
  "VMESSAGE": {  
    "072ade7d42d8": {  
      "msgId": "bcd1d9",  
      "date": 1486546629585,  
      "status": "accept",  
      "comment": "hi"  
    } } } },  
{"body": {  
  "VMESSAGE": {  
    "595d0a56cff2": {  
      "msgId": "595d0a",  
      "date": 1486566646197,  
      "status": "reject",  
      "comment": "no good"  
    } } } },  
{"body": {  
  "VMESSAGE": {  
    "52ffd09bf5b5": {  
      "msgId": "1dadce",  
      "date": 1486568943752,  
      "status": "accept"  
    } } } }  
  
  {"$addFields": {  
    "bvm": {"$objectToArray": "$body.VMESSAGE"}  
  } },  
  {"$addFields": {  
    "bvm2": {"$arrayElemAt": ["$bvm", 0]}  
  } },  
  {"$addFields": {  
    "bvm": {"$objectToArray": "$bvm2.v"}  
  } },  
  {"$addFields": {  
    "msgarr": {  
      "$concatArrays": [  
        [ { "k": "vid", "v": "$bvm2.k" } ],  
        "$bvm"  
      ]  
    }  
  } },  
  {"$addFields": {  
    "message": {"$arrayToObject": "$msgarr"}  
  } }  
  
  {"message": {  
    "vid": "072ade7d42d8",  
    "msgId": "bcd1d9",  
    "date": 1486546629585,  
    "status": "accept",  
    "comment": "hi"  
  } },  
  {"message": {  
    "vid": "595d0a56cff2",  
    "msgId": "595d0a",  
    "date": 1486566646197,  
    "status": "reject",  
    "comment": "no good"  
  } },  
  {"message": {  
    "vid": "52ffd09bf5b5",  
    "msgId": "1dadce",  
    "date": 1486568943752,  
    "status": "accept"  
  } }  
}
```

\$addFields

\$addFields

\$addFields

\$addFields

\$addFields

```
{"body": {  
  "VMESSAGE": {  
    "072ade7d42d8": {  
      "msgId": "bcd1d9",  
      "date": 1486546629585,  
      "status": "accept",  
      "comment": "hi"  
    } } },  
{"body": {  
  "VMESSAGE": {  
    "595d0a56cff2": {  
      "msgId": "595d0a",  
      "date": 1486566646197,  
      "status": "reject",  
      "comment": "no good"  
    } } },  
{"body": {  
  "VMESSAGE": {  
    "52ffd09bf5b5": {  
      "msgId": "1dadce",  
      "date": 1486568943752,  
      "status": "accept"  
    } } } }
```

```
{"$addFields":{  
  "message":{  
    "$arrayToObject":{  
      "$let":{  
        "vars":{  
          "elem": {"$arrayElemAt:[  
            {"$objectToArray":"$body.VMESSAGE"},  
            0  
          ]}  
        },  
        "in": {"$concatArrays": [  
          { k: "vid", v: "$$elem.k" } ],  
          {$objectToArray:"$$elem.v"}  
        ]}  
      }  
    }  
  }  
}
```

```
{"message": {  
  "vid": "072ade7d42d8",  
  "msgId": "bcd1d9",  
  "date": 1486546629585,  
  "status": "accept",  
  "comment": "hi" } },  
{"message": {  
  "vid": "595d0a56cff2",  
  "msgId": "595d0a",  
  "date": 1486566646197,  
  "status": "reject",  
  "comment": "no good" } },  
{"message": {  
  "vid": "52ffd09bf5b5",  
  "msgId": "1dadce",  
  "date": 1486568943752,  
  "status": "accept" } }
```

\$addFields

```
{"body": {  
  "VMESSAGE": {  
    "072ade7d42d8": {  
      "msgId": "bcd1d9",  
      "date": 1486546629585,  
      "status": "accept",  
      "comment": "hi"  
    } } },  
{"body": {  
  "VMESSAGE": {  
    "595d0a56cff2": {  
      "msgId": "595d0a",  
      "date": 1486566646197,  
      "status": "reject",  
      "comment": "no good"  
    } } },  
{"body": {  
  "VMESSAGE": {  
    "52ffd09bf5b5": {  
      "msgId": "1dadce",  
      "date": 1486568943752,  
      "status": "accept"  
    } } } }
```

```
{"$addFields":{  
  "message":{  
    "$let":{  
      "vars":{  
        "elem": {"$arrayElemAt:[  
          {"$objectToArray":"$body.VMESSAGE"},  
          0  
        ]}  
      },  
      "in": {"$mergeObjects": [  
        {"vid": "$$elem.k"},  
        "$$elem.v"  
      ]}  
    }  
  }},  
  "message":{  
    "vid": "072ade7d42d8",  
    "msgId": "bcd1d9",  
    "date": 1486546629585,  
    "status": "accept",  
    "comment": "hi"  
  },  
  {"message": {  
    "vid": "595d0a56cff2",  
    "msgId": "595d0a",  
    "date": 1486566646197,  
    "status": "reject",  
    "comment": "no good"  
  },  
  {"message": {  
    "vid": "52ffd09bf5b5",  
    "msgId": "1dadce",  
    "date": 1486568943752,  
    "status": "accept"  
  }}}
```

EXPRESSIONS

AGGREGATION

FIND QUERY

UPDATE QUERY

DOCUMENT VALIDATION

```
{children: [  
    {name:"Max", dob:"1994-12-01", dep:true},  
    {name:"Sam", dob:"1997-09-28", dep:true},  
    {name:"Kim", dob:"2000-02-29", dep:true}  
]}
```

EXPRESSIONS

AGGREGATION

FIND QUERY

UPDATE QUERY

DOCUMENT VALIDATION

```
{children: [  
    {name:"Max", dob:"1994-12-01", dep:false},  
    {name:"Sam", dob:"1997-09-28", dep:true},  
    {name:"Kim", dob:"2000-02-29", dep:true}  
]}
```

```
db.c.aggregate([  
{$addFields:{  
    numChildren:{$size:"$children"},  
    numDependents:{$size:{  
        $filter:{  
            input:"$children.dep",  
            cond: "$$this"  
        }  
    }}  
}},  
...  
])
```

EXPRESSIONS

NEW IN 3.6

AGGREGATION

FIND QUERY

UPDATE QUERY

DOCUMENT VALIDATION

```
{children: [  
    {name:"Max", dob:"1994-12-01", dep:false},  
    {name:"Sam", dob:"1997-09-28", dep:true},  
    {name:"Kim", dob:"2000-02-29", dep:true}  
]}
```

```
db.c.find({$expr:{$gt:[ "$a", "$b" ]}})
```

EXPRESSIONS

NEW IN 3.6

AGGREGATION

FIND QUERY

UPDATE QUERY

DOCUMENT VALIDATION

```
{children: [  
    {name:"Max", dob:"1994-12-01", dep:false},  
    {name:"Sam", dob:"1997-09-28", dep:true},  
    {name:"Kim", dob:"2000-02-29", dep:true}  
]}
```

```
db.c.find({$expr: {  
    $lt:[ {$size:{$filter:{  
        input:"$children.dep",  
        cond:"$$this"  
    }}}],  
    2  
}})
```

EXPRESSIONS

NEW IN 3.6

AGGREGATION

FIND QUERY

UPDATE QUERY

DOCUMENT VALIDATION

```
{children: [
  {name:"Max", dob:"1994-12-01", dep:false},
  {name:"Sam", dob:"1997-09-28", dep:true},
  {name:"Kim", dob:"2000-02-29", dep:true}
]}
```

```
db.c.find({$expr:{$gt:[
  {$let:{
    vars:{dobs:{$map:{
      input:"$children.dob"
      in: {$year:{$dateFromString:{dateString:"$$this"
        }}}}}}},
    in:{$subtract:[
      {$max:"$$dobs"}, {$min:"$$dobs"}]}]}}, 10
]})
```

EXPRESSIONS

NEW IN 3.6

AGGREGATION

FIND QUERY

UPDATE QUERY

DOCUMENT VALIDATION

```
{children: [  
    {name:"Max", dob:"1994-12-01", dep:false},  
    {name:"Sam", dob:"1997-09-28", dep:true},  
    {name:"Kim", dob:"2000-02-29", dep:true}  
]}
```

```
db.c.update({$expr: {$anyElementTrue:{$map:{  
    input:"$children",  
    in: {$and:[  
        {$lt:["$$this.dob","1997-03-20"]},  
        " $$this.dep"  
    ]}  
}}}},  
{$set:{ audit:true }}  
)
```

EXPRESSIONS

NEW IN 3.6

AGGREGATION

FIND QUERY

UPDATE QUERY

DOCUMENT VALIDATION

```
{children: [  
    {name:"Max", dob:"1994-12-01", dep:false},  
    {name:"Sam", dob:"1997-09-28", dep:true},  
    {name:"Kim", dob:"2000-02-29", dep:true}  
]}
```

```
db.createCollection("c", validator:  
{$expr: {  
    <anything you can express>  
}})
```

NETWORK SUSPECT ACTIVITY DETECTION

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T05:28:13Z")  
}
```



```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}
```

```
{ _id: "303900",  
  ips: [  
    {ip:"71.56.112.56",  
     ts:ISODate("2017-05-08T08:54:04Z")  
    },  
    {ip:"71.56.112.56",  
     ts:ISODate("2017-05-09T09:01:11Z")  
    },  
    {ip:"12.130.117.87",  
     ts:ISODate("2017-05-09T09:04:59Z")  
    }  
]
```

\$match

\$sort

\$group

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}  
  
[ { _id: "303900",  
    ips: [  
        {ip:"71.56.112.56",  
         ts:ISODate("2017-05-08T...")  
        },  
        {ip:"71.56.112.56",  
         ts:ISODate("2017-05-09T...")  
        },  
        {ip:"12.130.117.87",  
         ts:ISODate("2017-05-09T...")  
        }  
    ]}  
  
    {$match:{ts:{$gte:start,$lt:end}}},  
    {$sort:{ts:1}},  
    {$group:{_id:"$user",ips:{$push:{ip:"$ipaddr", ts:"$ts"}}}},  
    {$addFields:{diffs: {$filter:{  
        input:{$map:{  
            input: {$range:[0,{$subtract:[{$size:"$ips"},1]}]}, as:"i",  
            in:{$let:{vars:{ip1:$arrayElemAt:["$ips","$$i"]},  
                    ip2:$arrayElemAt:["$ips",{$add:["$$i",1]}]}},  
            in:{  
                diff:{$divide:[{$subtract:["$$ip2.ts","$$ip1.ts"]}],60000},  
                ip1:"$$ip1.ip", t1:"$$ip1.ts",  
                ip2:"$$ip2.ip", t2:"$$ip2.ts"  
            }}}},  
            cond:{$and:[{$lt:[{$this.diff},10]},{$ne:[{$this.ip1},{$this.ip2}]}]}  
        ]}}},
```

\$match

\$sort

\$group

\$addFields

\$match

\$project

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}  
  
[ { _id: "303900",  
    ips: [  
        {ip:"71.56.112.56",  
         ts:ISODate("2017-05-08T...")  
        },  
        {ip:"71.56.112.56",  
         ts:ISODate("2017-05-09T...")  
        },  
        {ip:"12.130.117.87",  
         ts:ISODate("2017-05-09T...")  
        }  
    ]}  
  
    {$match:{ts:{$gte:start,$lt:end}}},  
    {$sort:{ts:1}},  
    {$group:{_id:"$user",ips:{$push:{ip:"$ipaddr", ts:"$ts"}}}},  
    {$addFields:{diffs: {$filter:{  
        input:{$map:{  
            input: {$range:[0,{$subtract:[{$size:"$ips"},1]}]}, as:"i",  
            in:{$let:{vars:{ip1:$arrayElemAt:["$ips","$$i"]},  
                    ip2:$arrayElemAt:["$ips",{$add:[ "$$i",1]}]}},  
            in:{  
                diff:{$divide:[{$subtract:["$$ip2.ts","$$ip1.ts"]}],60000},  
                ip1:"$$ip1.ip", t1:"$$ip1.ts",  
                ip2:"$$ip2.ip", t2:"$$ip2.ts"  
            }}}},  
            cond:{$and:[{$lt:[ "$$this.diff",10]},{$ne:[ "$$this.ip1","$$this.ip2"]}]}}},  
            {$match:{"diffs":{$ne:[]}}},  
            {$project:{_id:0, user:"$_id", suspectLogins:"$diffs"}}}
```

\$match

\$sort

\$group

\$addFields

\$match

\$project

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}  
  
[{"user": "35237073",  
 "suspectLogins": [  
     {"diff": 4.8333333333, "ip1": "106.220.151.16", "t1": "2017-05-08T06:58", "ip2": "223.182.113.15", "t2": "2017-05-08T07:03"},  
     {"diff": 8.3, "ip1": "223.182.113.15", "t1": "2017-05-08T07:03", "ip2": "49.206.217.26", "t2": "2017-05-08T07:11"}]  
}  
  
{$match: {ts: {$gte: start, $lt: end}}},  
{$sort: {ts: 1}},  
{$group: {_id: "$user", ips: {$push: {ip: "$ipaddr", ts: "$ts"}}}},  
{$addFields: {diffs: {$filter: {  
        input: {$map: {  
            input: {$range: [0, {$subtract: [{size: "$ips"}, 1]}]}, as: "i"},  
            in: {$let: {vars: {ip1: {$arrayElemAt: ["$ips", "$$i"]},  
                ip2: {$arrayElemAt: ["$ips", {$add: ["$$i", 1]}]}},  
                in: {  
                    diff: {$divide: [{subtract: ["$$ip2.ts", "$$ip1.ts"]}, 60000]},  
                    ip1: "$$ip1.ip", t1: "$$ip1.ts",  
                    ip2: "$$ip2.ip", t2: "$$ip2.ts"}  
                }}}},  
        cond: {$and: [{lt: ["$$this.diff", 10]}, {ne: ["$$this.ip1", "$$this.ip2"]}]}  
    }},  
    {$match: {"diffs": {$ne: []}}},  
    {$project: {_id: 0, user: "$_id", suspectLogins: "$diffs"}}
```

\$match

\$sort

\$group

\$addFields

\$match

\$project

```
start=ISODate("...")          {$match:{ts:{$gte:start,$lt:end}}},  
end=ISODate("...")           {$sort:{ts:1}},  
{                             {$group:{_id:"$user",ips:{$push:{ip:"$ipaddr", ts:"$ts"}}}},  
  user: "303900",  
  ipaddr: "71.56.112.56",  
  ts:ISODate("2017-05-08T...")  
}  
  
{ _id: "303900",  
  ips: [  
    {ip:"71.56.112.56",  
     ts:ISODate("2017-05-08T...")  
    },  
    {ip:"71.56.112.56",  
     ts:ISODate("2017-05-09T...")  
    },  
    {ip:"12.130.117.87",  
     ts:ISODate("2017-05-09T...")  
    }  
]  
  {$addFields:{diffs: {$filter:{  
    input:{'$map':{  
      input: {$range:[0,{$subtract:[{$size:'$ips'},1]}]}, as:"i",  
      in:{'$let':{vars:{ip1:{$arrayElemAt:["$ips","$$i"]},  
                     ip2:{$arrayElemAt:["$ips",{$add:[$$i,1]}]}},  
      in:{  
        diff:{$divide:[{$subtract:[$$ip2.ts,"$$ip1.ts"]},60000]},  
        ip1:"$$ip1.ip", t1:"$$ip1.ts",  
        ip2:"$$ip2.ip", t2:"$$ip2.ts"  
      }}}},  
    cond:{$and:[{$lt:[$$this.diff,10]}, {$ne:[$$this.ip1,"$$this.ip2"]}]}}},  
  {$match:{'diffs':{$ne:[]}}},  
  {$project:{_id:0, user:"$_id", suspectLogins:"$diffs"}}}
```

\$match

\$sort

\$group

\$addFields

\$match

\$project

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}  
  
[
```

```
    { _id: "303900",  
        ips: [  
            {ip:"71.56.112.56",  
                ts:ISODate("2017-05-08T...")  
            },  
            {ip:"71.56.112.56",  
                ts:ISODate("2017-05-09T...")  
            },  
            {ip:"12.130.117.87",  
                ts:ISODate("2017-05-09T...")  
            }  
        ]  
    }
```

```
    {$match:{ts:{$gte:start,$lt:end}}},  
    {$sort:{ts:1}},  
    {$group:{_id:"$user",ips:{$push:{ip:"$ipaddr", ts:"$ts"}}}},  
    {$addFields:{diffIpNum:{$size:{$setUnion:$ips.ip}}}},  
    {$match:{diffIpNum:{$gt:1}}},  
    {$addFields:{diffs: {$filter:  
        input:{$map:  
            input: {$range:[0,{$subtract:[{$size:$ips},1]}]}, as:"i",  
            in:{$let:{vars:{ip1:{$arrayElemAt:[$ips,"$$i"]},  
                ip2:{$arrayElemAt:[$ips,{$add:[$$i,1]}]}},  
                in:  
                    diff:{$divide:[{$subtract:[$$ip2.ts,"$$ip1.ts"]},60000]},  
                    ip1:"$$ip1.ip", t1:"$$ip1.ts",  
                    ip2:"$$ip2.ip", t2:"$$ip2.ts"  
                }}}},  
        cond:{$and:[{$lt:[$$this.diff,10]},{$ne:[$$this.ip1,"$$this.ip2"]}]}}},  
    {$match:{diffs:{$ne:[]}}},  
    {$project:{_id:0, user:"$_id", suspectLogins:"$diffs"}}
```

\$match

\$sort

\$group

\$addFields

\$match

\$addFields

\$match \$pr

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}  
  
[  
    { _id: "303900",  
        ips: [  
            {ip:"71.56.112.56",  
             ts:ISODate("2017-05-08T...")  
            },  
            {ip:"71.56.112.56",  
             ts:ISODate("2017-05-09T...")  
            },  
            {ip:"12.130.117.87",  
             ts:ISODate("2017-05-09T...")  
            },  
            diffIps: [  
                "71.56.112.56",  
                "12.130.117.87"  
            ]  
    }  
]
```

```
    {$match:{ts:{$gte:start,$lt:end}}},  
    {$sort:{ts:1}},  
    {$group:{_id:"$user",ips:{$push:{ip:"$ipaddr", ts:"$ts"}},  
              diffIps:{$addToSet:"$ipaddr"}}},  
    {$match:{$expr:{$gt:[{$size:"$diffIps"},1]}}},  
    {$addFields:{diffs: {$filter:  
        input:{$map:  
            input: {$range:[0,{$subtract:[{$size:"$ips"},1]}]}, as:"i",  
            in:{$let:{vars:{ip1:{$arrayElemAt:["$ips","$$i"]},  
                           ip2:{$arrayElemAt:["$ips",{$add:[ "$$i",1]}]}},  
                    in:  
                        diff:{$divide:[{$subtract:[ "$$ip2.ts", "$$ip1.ts"]},60000]},  
                        ip1:"$$ip1.ip", t1:"$$ip1.ts",  
                        ip2:"$$ip2.ip", t2:"$$ip2.ts"  
                    }}}},  
        cond:{$and:[{$lt:[ $$this.diff,10]}, {$ne:[ $$this.ip1, $$this.ip2]}]}  
    }}},  
    {$match:{"diffs":{$ne:[]}}},  
    {$project:{_id:0, user:"$_id", suspectLogins:"$diffs"}}
```

\$match

\$sort

\$group

\$match

\$addFields

\$match

\$project

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}  
  
[  
    { _id: "303900",  
        ips: [  
            {ip:"71.56.112.56",  
             ts:ISODate("2017-05-08T...")  
            },  
            {ip:"71.56.112.56",  
             ts:ISODate("2017-05-09T...")  
            },  
            {ip:"12.130.117.87",  
             ts:ISODate("2017-05-09T...")  
            },  
            diffIps: [  
                "71.56.112.56",  
                "12.130.117.87"  
            ]  
    }]
```

```
    {$match:{ts:{$gte:start,$lt:end}}},  
    {$sort:{ts:1}},  
    {$group:{_id:"$user",ips:{$push:{ip:"$ipaddr", ts:"$ts"}},  
              diffIps:{$addToSet:"$ipaddr"}}},  
    {$match:{"diffIps.1":{$exists:true}}},  
    {$addFields:{diffs: {$filter:{  
        input:{$map:{  
            input: {$range:[0,{$subtract:[{$size:"ips"},1]}]}, as:"i",  
            in:{$let:{vars:{ip1:{$arrayElemAt:["ips","$$i"]},  
                           ip2:{$arrayElemAt:["ips",{$add:[ "$$i",1]}]}},  
                           in:{  
                               diff:{$divide:[{$subtract:["$$ip2.ts","$$ip1.ts"]},60000]},  
                               ip1:"$$ip1.ip", t1:"$$ip1.ts",  
                               ip2:"$$ip2.ip", t2:"$$ip2.ts"  
                           }}}},  
            cond:{$and:[{$lt:[ "$$this.diff",10]},{$ne:[ "$$this.ip1", "$$this.ip2"]]}]}},  
        {$match:{"diffs":{$ne:[]}}},  
        {$project:{_id:0, user:"$_id", suspectLogins:"$diffs"}}}
```

\$match

\$sort

\$group

\$match

\$addFields

\$match

\$project

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}  
  
=====
```

```
{ _id: "303900",  
  ips: [  
    {ip:"71.56.112.56",  
     ts:ISODate("2017-05-08T...")  
    },  
    {ip:"71.56.112.56",  
     ts:ISODate("2017-05-09T...")  
    },  
    {ip:"12.130.117.87",  
     ts:ISODate("2017-05-09T...")  
    },  
  diffIps: [  
    "71.56.112.56",  
    "12.130.117.87"  
  ]  
}
```

```
  {$match:{ts:{$gte:start,$lt:end}}},  
  {$sort:{ts:1}},  
  {$group:{_id:"$user",ips:{$push:{ip:"$ipaddr", ts:"$ts"}},  
           diffIps:{$addToSet:"$ipaddr"}}},  
  {$match:{"diffIps.1":{$exists:true}}},  
  {$addFields:{diffs: {$filter:  
    input:{$map:{  
      input: {$range:[0,{$subtract:[{$size:"$ips"},1]}]}, as:"i",  
      in:{$let:{vars:{ip1:{$arrayElemAt:["$ips","$$i"]},  
                     ip2:{$arrayElemAt:["$ips",{$add:[ "$$i",1]}]}},  
                     in:{  
                       diff:{$divide:[$subtract:["$$ip2.ts","$$ip1.ts"],60000]},  
                       ip1:"$$ip1.ip", t1:"$$ip1.ts",  
                       ip2:"$$ip2.ip", t2:"$$ip2.ts"  
                     }}}},  
      cond:{$and:[{$lt:[ $$this.diff,10]},{$ne:[ $$this.ip1,"$$this.ip2"]}]}}},  
    {$match:{diffs:{$ne:[]}}},  
    {$project:{_id:0, user:"$_id", suspectLogins:"$diffs"}}}
```

\$match

\$sort

\$group

\$match

\$addFields

\$match

\$project

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}  
  
=====
```

```
{ _id: "303900",  
  ips: [  
    {ip:"71.56.112.56",  
     ts:ISODate("2017-05-08T...")  
    },  
    {ip:"71.56.112.56",  
     ts:ISODate("2017-05-09T...")  
    },  
    {ip:"12.130.117.87",  
     ts:ISODate("2017-05-09T...")  
    },  
  diffIps: [  
    "71.56.112.56",  
    "12.130.117.87"  
  ]  
}
```

```
  {$match:{ts:{$gte:start,$lt:end}}},  
  {$sort:{ts:1}},  
  {$group:{_id:"$user",ips:{$push:{ip:"$ipaddr", ts:"$ts"}},  
            diffIps:{$addToSet:"$ipaddr"}}},  
  {$match:{"diffIps.1":{$exists:true}}},  
  {$addFields:{diffs: {$filter:  
    input:{$map:  
      input: {$range:[0,{$subtract:[{$size:"$ips"},1]}]}, as:"i",  
      in:{$let:{vars:{ip1:{$arrayElemAt:["$ips","$$i"]},  
                    ip2:{$arrayElemAt:["$ips",{$add:[ "$$i",1]}]}},  
                  in:  
                    diff:{$cond:  
                      input:  
                        {$and:[  
                          {$lt:[{$this.diff},10]},  
                          {$ne:[{$this.ip1},{$this.ip2}]}  
                        ]}  
                      }},  
                      ip1:"$$ip1.ip", t1:"$$ip1.ts",  
                      ip2:"$$ip2.ip", t2:"$$ip2.ts"  
                    ]}}},  
    cond:{$and:[{$lt:[{$this.diff},10]},{$ne:[{$this.ip1},{$this.ip2}]}]}  
  ]}},  
  {$match:{"diffs":{$ne:[]}}},  
  {$project:{_id:0, user:"$_id", suspectLogins:"$diffs"}}}
```

\$match

\$sort

\$group

\$match

\$addFields

\$match

\$project

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}  
  
=====
```

```
{ _id: "303900",  
  ips: [  
    {ip:"71.56.112.56",  
     ts:ISODate("2017-05-08T...")  
    },  
    {ip:"71.56.112.56",  
     ts:ISODate("2017-05-09T...")  
    },  
    {ip:"12.130.117.87",  
     ts:ISODate("2017-05-09T...")  
    },  
  diffIps: [  
    "71.56.112.56",  
    "12.130.117.87"  
  ]  
}
```

```
  {$match:{ts:{$gte:start,$lt:end}}},  
  {$sort:{ts:1}},  
  {$group:{_id:"$user",ips:{$push:{ip:"$ipaddr", ts:"$ts"}},  
            diffIps:{$addToSet:"$ipaddr"}}},  
  {$match:{"diffIps.1":{$exists:true}}},  
  {$addFields:{diffs: {$filter:  
    input:{$map:  
      input: {$range:[0,{$subtract:[{$size:"$ips"},1]}]}, as:"i",  
      in:{$let:{vars:{ip1:{$arrayElemAt:["$ips","$$i"]},  
                    ip2:{$arrayElemAt:["$ips",{$add:[ "$$i",1]}]}},  
                  in:  
                    diff:{$cond:  
                      if:{$ne:["$$ip1.ip","$$ip2.ip"]},  
                      then:{$divide:[{$subtract:[ "$$ip2.ts","$$ip1.ts"]},60000]},  
                      else: 9999 },  
                      ip1:"$$ip1.ip", t1:"$$ip1.ts",  
                      ip2:"$$ip2.ip", t2:"$$ip2.ts"  
                    }}}}},  
    cond:{$and:[{$lt:[ "$$this.diff",10]},{$ne:[ "$$this.ip1","$$this.ip2"]}]}}},  
  {$match:{diffs:{$ne:[]}}},  
  {$project:{_id:0, user:"$_id", suspectLogins:"$diffs"}}}
```

\$match

\$sort

\$group

\$match

\$addFields

\$match

\$project

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}  
  
=====
```

```
{ _id: "303900",  
  ips: [  
    {ip:"71.56.112.56",  
     ts:ISODate("2017-05-08T...")  
    },  
    {ip:"71.56.112.56",  
     ts:ISODate("2017-05-09T...")  
    },  
    {ip:"12.130.117.87",  
     ts:ISODate("2017-05-09T...")  
    },  
  diffIps: [  
    "71.56.112.56",  
    "12.130.117.87"  
  ]  
}
```

```
  {$match:{ts:{$gte:start,$lt:end}}},  
  {$sort:{ts:1}},  
  {$group:{_id:"$user",ips:{$push:{ip:"$ipaddr", ts:"$ts"}},  
            diffIps:{$addToSet:"$ipaddr"}}},  
  {$match:{"diffIps.1":{$exists:true}}},  
  {$addFields:{diffs: {$filter:  
    input:{$map:  
      input: {$range:[0,{$subtract:[{$size:"$ips"},1]}]}, as:"i",  
      in:{$let:{vars:{ip1:{$arrayElemAt:["$ips","$$i"]},  
                    ip2:{$arrayElemAt:["$ips",{$add:[ "$$i",1]}]}},  
                  in:  
                    diff:{$cond:  
                      if:{$ne:["$$ip1.ip","$$ip2.ip"]},  
                      then:{$divide:[{$subtract:[ "$$ip2.ts","$$ip1.ts"]},60000]},  
                      else: 9999 },  
                      ip1:"$$ip1.ip", t1:"$$ip1.ts",  
                      ip2:"$$ip2.ip", t2:"$$ip2.ts"  
                    }}}},  
    cond:{$and:[{$lt:[ "$$this.diff",10]},{$ne:[ "$$this.ip1","$$this.ip2"]}]}}},  
  {$match:{"diffs":{$ne:[]}}},  
  {$project:{_id:0, user:"$_id", suspectLogins:"$diffs"}}}
```

\$match

\$sort

\$group

\$match

\$addFields

\$match

\$project

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}
```

```
[ "user" : "35237073",  
  "suspectLogins" : [  
      {"diff": 4.8333333333,  
       "ip1": "106.220.151.16",  
       "t1": "2017-05-08T06:58",  
       "ip2": "223.182.113.15"  
       "t2": "2017-05-08T07:03"},  
      {"diff": 8.3,  
       "ip1": "223.182.113.15",  
       "t1": "2017-05-08T07:03",  
       "ip2": "49.206.217.26",  
       "t2": "2017-05-08T07:11"}]
```

```
]
```

```
  {$match:{ts:{$gte:start,$lt:end}}},  
  {$sort:{ts:1}},  
  {$group:{_id:"$user",ips:{$push:{ip:"$ipaddr", ts:"$ts"}},  
           diffIps:{$addToSet:"$ipaddr"}}},  
  {$match:{"diffIps.1":{$exists:true}}},  
  {$addFields:{diffs: {$filter:  
          input:{$map:  
                  input: {$range:[0,{$subtract:[{$size:"$ips"},1]}]}, as:"i",  
                  in:{$let:{vars:{ip1:{$arrayElemAt:["$ips","$$i"]},  
                               ip2:{$arrayElemAt:["$ips",{$add:[ "$$i",1]}]}},  
                               in:  
                                   diff:{$cond:  
                                           if:{$ne:["$$ip1.ip","$$ip2.ip"]},  
                                           then:{$divide:[{$subtract:[ "$$ip2.ts","$$ip1.ts"]},60000]},  
                                           else: 9999 },  
                                   ip1:"$$ip1.ip", t1:"$$ip1.ts",  
                                   ip2:"$$ip2.ip", t2:"$$ip2.ts"}  
                  }}}}},  
  cond:{$lt:[ $$this.diff,10]}  
  }},  
  {$match:{"diffs":{$ne:[]}}},  
  {$project:{_id:0, user:"$_id", suspectLogins:"$diffs"}}
```

\$match

\$sort

\$group

\$match

\$addFields

\$match

\$project

```
start=ISODate("...")  
end=ISODate("...")  
{  
    user: "303900",  
    ipaddr: "71.56.112.56",  
    ts:ISODate("2017-05-08T...")  
}  
  
=====
```

```
{ "user" : "35237073",  
  "suspectLogins" : [  
      {"diff": 4.8333333333,  
       "ip1": "106.220.151.16",  
       "t1": "2017-05-08T06:58",  
       "ip2": "223.182.113.15"  
       "t2": "2017-05-08T07:03"  
     },  
      {"diff": 8.3,  
       "ip1": "223.182.113.15",  
       "t1": "2017-05-08T07:03",  
       "ip2": "49.206.217.26",  
       "t2": "2017-05-08T07:11"  
     }  
  ]  
}
```

```
  {$match:{ts:{$gte:start,$lt:end}}},  
  {$sort:{ts:1}},  
  {$group:{_id:"$user",ips:{$push:{ip:"$ipaddr", ts:"$ts"}},  
           diffIps:{$addToSet:"$ipaddr"}}},  
  {$match:{"diffIps.1":{$exists:true}}},  
  {$addFields:{diffs: {$filter:  
    input:{$map:{  
      input: {$range:[0,{$subtract:[{$size:"$ips"},1]}]}, as:"i",  
      in:{$let:{vars:{ip1:{$arrayElemAt:["$ips","$$i"]},  
                     ip2:{$arrayElemAt:["$ips",{$add:["$$i",1]}]}},  
            in:{  
              diff:{$cond:{  
                if:{$ne:["$$ip1.ip","$$ip2.ip"]},  
                then:{$divide:[{$subtract:[$$ip2.ts,"$$ip1.ts"]},60000]},  
                else: 9999 },  
                ip1:"$$ip1.ip", t1:"$$ip1.ts",  
                ip2:"$$ip2.ip", t2:"$$ip2.ts"  
              }}}},  
              cond:{$lt:["$$this.diff",10]}  
            }},  
  {$match:{"diffs":{$ne:[]}},  
   {$project:{_id:0, user:"$_id", suspectLogins:"$diffs"}}}
```

\$match

\$sort

\$group

\$match

\$addFields

\$match

\$project

More about performance

More about performance

INDEX USE

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3}
],{explain:true})

{"stages": [
  {"$cursor": {"query": { "language": "English"}, "fields": { "subjects": 1,"_id": 0} ...}},
  {"$unwind": {"path": "$subjects"}},
  {"$match": {"subjects": {"$regex": "ABC"}}, {"$group": {"_id": "$subjects","count": {"$sum": {"$const": 1}}}}, {"$sort": {"sortKey": {"count": -1}, "limit": NumberLong(3)}}
]} ] }
```

```
db.books.aggregate([
  {$unwind:"$subjects"},  

  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3}
],{explain:true})  
  
{"stages": [  
    {"$cursor": {"query": { "language": "English"},  
                 "fields": { "subjects": 1,"_id": 0},  
                 "...  
               },  
    {"$unwind": {"path": "$subjects"}},  
    {"$match": {"subjects": {"$regex": "^ABC"}}},  
    {"$group": {"_id": "$subjects","count": {"$sum": {"$const": 1}}}},  
    {"$sort": {  
        "sortKey": {"count": -1},  
        "limit": NumberLong(3)  
      }}  
  ] }
```

```
db.books.aggregate([
  {$unwind:"$subjects"},  

  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3}
],{explain:true})  
  
{"stages": [  
    {"$cursor": {"query": { "language": "English"},  
                 "fields": { "subjects": 1,"_id": 0},  
                 "queryPlanner": {}  
               },  
    {"$unwind": {"path": "$subjects"}},  
    {"$match": {"subjects": {"$regex": "^[ABC]"} }},  
    {"$group": {"_id": "$subjects","count": {"$sum": {"$const": 1}}}},  
    {"$sort": {  
        "sortKey": {"count": -1},  
        "limit": NumberLong(3)  
      }  
  }]  
}
```

```
db.books.aggregate([
  {$unwind:"$subjects"},  

  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3}
],{explain:true})  
  
{"stages": [  
    {"$cursor": {"query": { "language": "English"},  
                 "fields": { "subjects": 1,"_id": 0},  
                 "queryPlanner": {  
                   }  
               },  
    {"$unwind": {"path": "$subjects"}},  
    {"$match": {"subjects": {"$regex": "^[ABC]"}},  
     {"$group": {"_id": "$subjects","count": {"$sum": {"$const": 1}}}},  
    {"$sort": {  
      "sortKey": {"count": -1},  
      "limit": NumberLong(3)  
    }}  
  ] }
```

```
db.books.aggregate([
  {$unwind:"$subjects"},  

  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3}
],{explain:true})  
  
{"stages": [  
    {"$cursor": {"query": { "language": "English"},  
                 "fields": { "subjects": 1,"_id": 0},  
                 "queryPlanner": {  
                   }  
               },  
    {"$unwind": {"path": "$subjects"}},  
    {"$match": {"subjects": {"$regex": "ABC"}},  
    {"$group": {"_id": "$subjects","count": {"$sum": {"$const": 1}}}},  
    {"$sort": {  
      "sortKey": {"count": -1},  
      "limit": NumberLong(3)  
    }}  
  ] }
```

```
db.books.aggregate([
  {$unwind:"$subjects"},  

  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3}
],{explain:true})
```

```
{"stages": [  

  {"$cursor": {"query": { "language": "English"},  

    "fields": { "subjects": 1,"_id": 0},  

    "queryPlanner": {  

      "plannerVersion": 1,  

      "namespace": "test.books",  

      "indexFilterSet": false,  

      "parsedQuery": {  

        "language": {  

          "$eq": "English"  

        }  

      },  

      "winningPlan": { },  

      "rejectedPlans": [ ]  

    }  

  }]
```

```
db.books.aggregate([
  {$unwind:"$subjects"},  

  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3}
],{explain:true})
```

```
{"stages": [  

  {"$cursor": {"query": { "language": "English"},  

    "fields": { "subjects": 1,"_id": 0},  

    "queryPlanner": {  

      "plannerVersion": 1,  

      "namespace": "test.books",  

      "indexFilterSet": false,  

      "parsedQuery": {  

        "language": {  

          "$eq": "English"  

        }  

      },  

      "winningPlan": {  

        "stage": "PROJECTION",  

        "inputStage": "COLLSCAN",  

        "keys": [{"key": "language", "value": "English"}],  

        "conditions": [{"language": "English"}],  

        "output": {"fields": ["subjects", "_id"], "order": 1},  

        "stageDescription": "PROJECTION {language: $eq: English} {subjects: 1, _id: 0}"  

      }  

    }  

  }  

]
```

```
"winningPlan" : {
    "stage" : "PROJECTION",
    "transformBy" : {
        "subjects" : 1,
        "_id" : 0
    },
    "inputStage" : {
        "stage" : "IXSCAN",
        "keyPattern" : {
            "language" : 1,
            "subjects" : 1
        },
        "indexName" : "language_1_subjects_1",
        "isMultiKey" : false,
        "multiKeyPaths" : {
            "language" : [ ],
            "subjects" : [ ]
        },
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2,
        "direction" : "forward",
        "indexBounds" : {
            "language" : [

```

```
"winningPlan" : {
    "stage" : "PROJECTION",
    "transformBy" : { "subjects" : 1, "_id" : 0 },
    "inputStage" : {
        "stage" : "IXSCAN",
        "keyPattern" : {"language" : 1, "subjects" : 1},
        "indexName" : "language_1_subjects_1",
        "isMultiKey" : false,
        "multiKeyPaths": {"language":[],"subjects":[]},
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2,
        "direction" : "forward",
        "indexBounds" : {
            "language" : [
                "[\"English\", \"English\"]"
            ],
            "subjects" : [
                "[MinKey, MaxKey]"
            ]
        }
    }
},
"rejectedPlans" : [ ]
```

```
"winningPlan" : {
    "stage" : "PROJECTION",
    "transformBy" : { "subjects" : 1, "_id" : 0 },
    "inputStage" : {
        "stage" : "IXSCAN",
        "keyPattern" : { "language" : 1, "subjects" : 1 },
        "indexName" : "language_1_subjects_1",
        "isMultiKey" : false,
        "multiKeyPaths": {"language":[],"subjects":[]},
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2,
        "direction" : "forward",
        "indexBounds" : {
            "language" : [
                "[\"English\", \"English\"]"
            ],
            "subjects" : [
                "[MinKey, MaxKey]"
            ]
        }
    }
},
"rejectedPlans" : [ ]
```

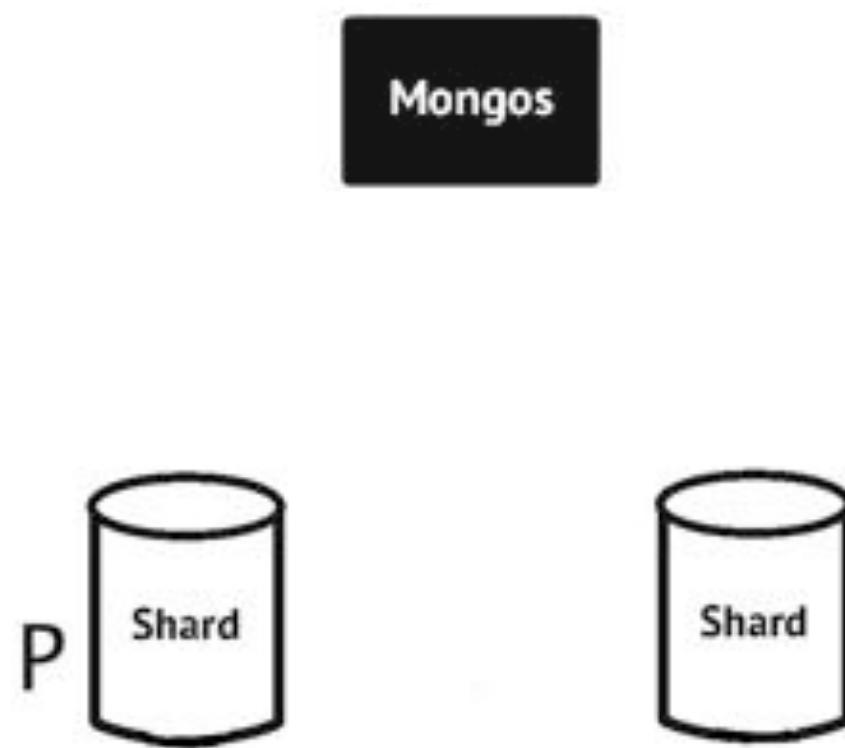
```
"winningPlan" : {  
    "stage" : "FETCH",  
    "inputStage" : {  
        "stage" : "IXSCAN",  
        "keyPattern" : {"language" : 1, "subjects" : 1},  
        "indexName" : "language_1_subjects_1",  
        "isMultiKey" : true,  
        "multiKeyPaths": {"language": [], "subjects": ["sub"]},  
        "isUnique" : false,  
        "isSparse" : false,  
        "isPartial" : false,  
        "indexVersion" : 2,  
        "direction" : "forward",  
        "indexBounds" : {  
            "language" : [  
                "[\"English\", \"English\"]"  
            ],  
            "subjects" : [  
                "[MinKey, MaxKey]"  
            ]  
        }  
    },  
    "rejectedPlans" : [ ]
```

streaming VS blocking

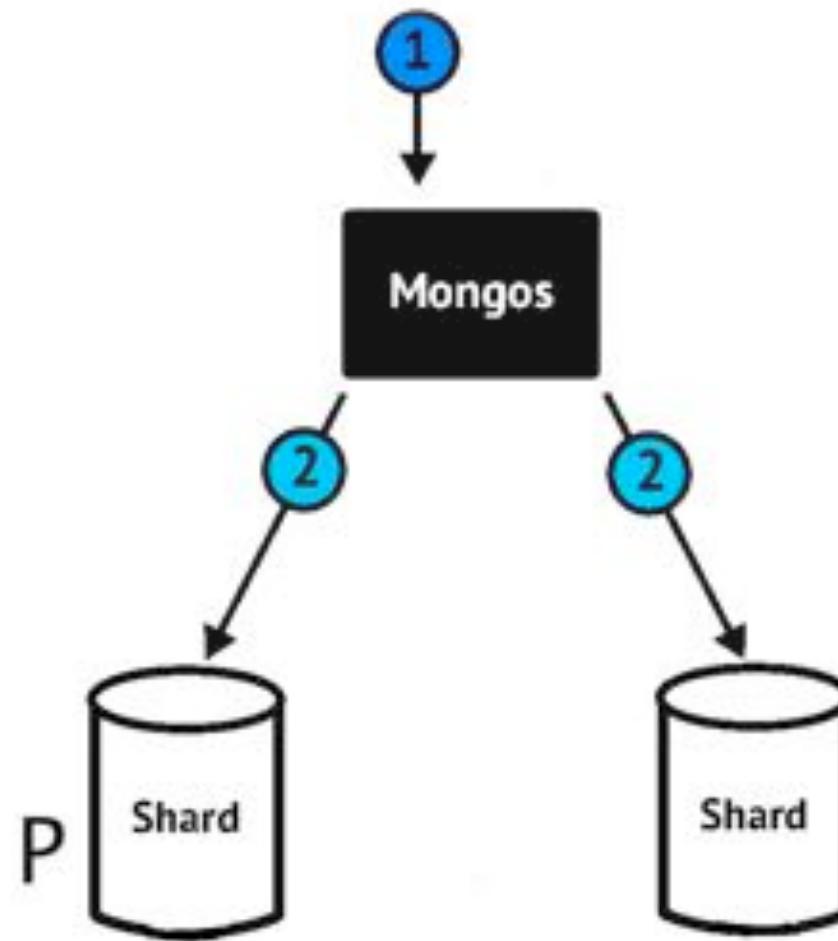
streaming VS blocking

SHARDING & MERGING

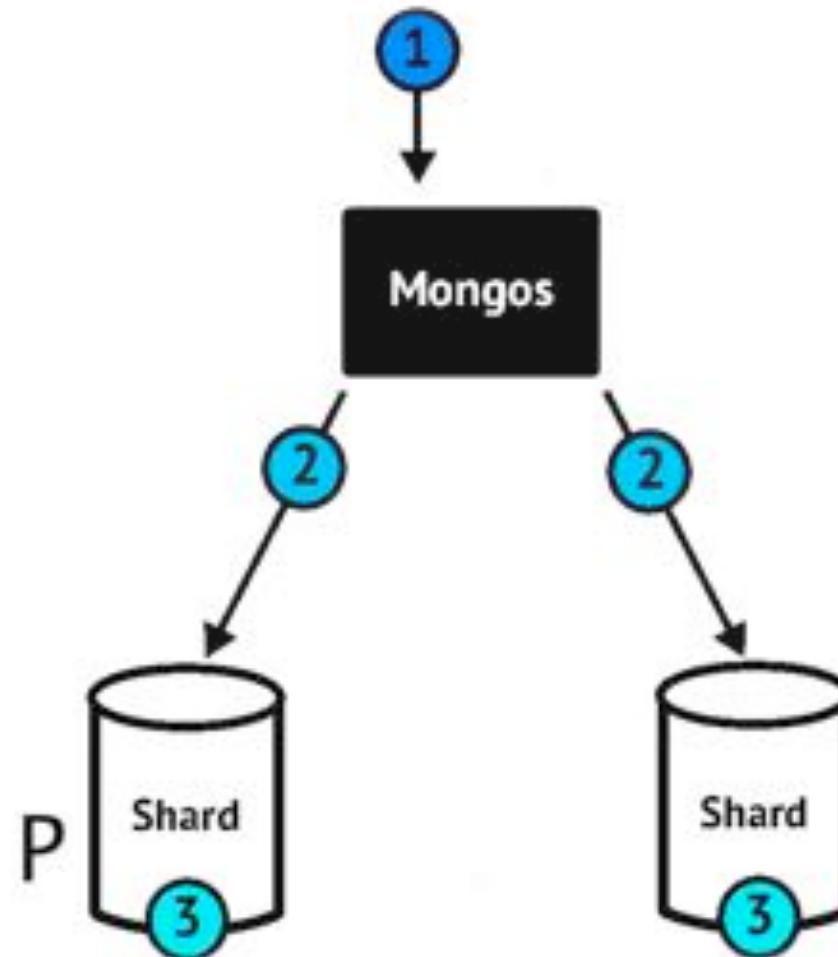
SHARDING



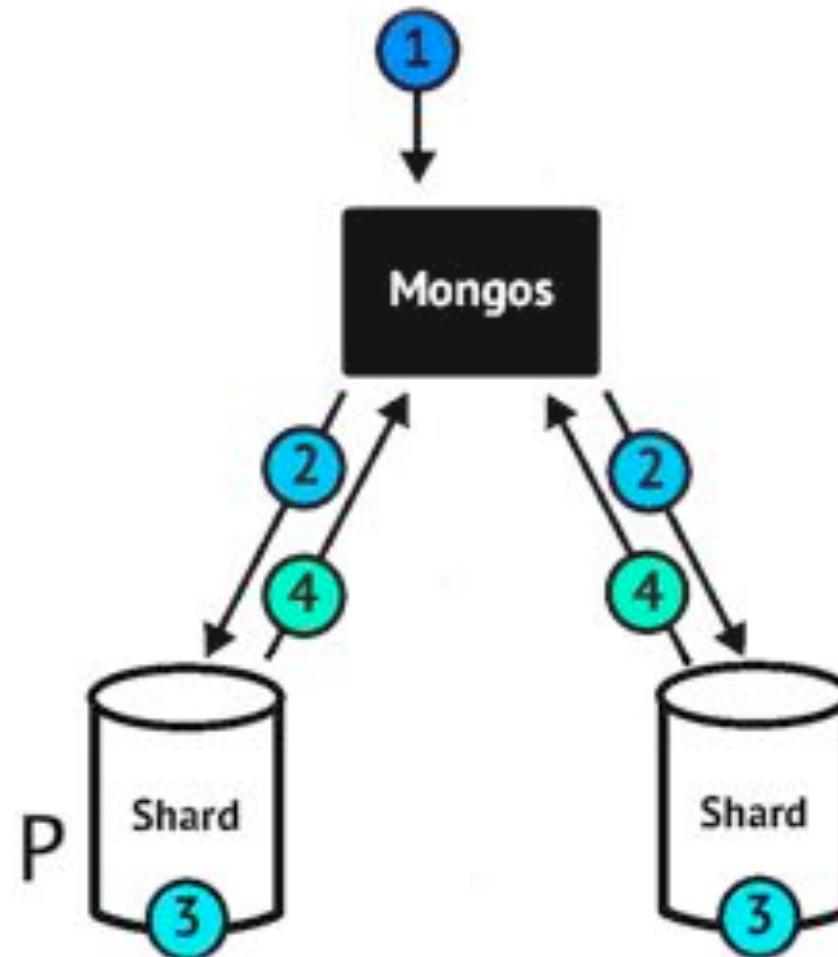
SHARDING



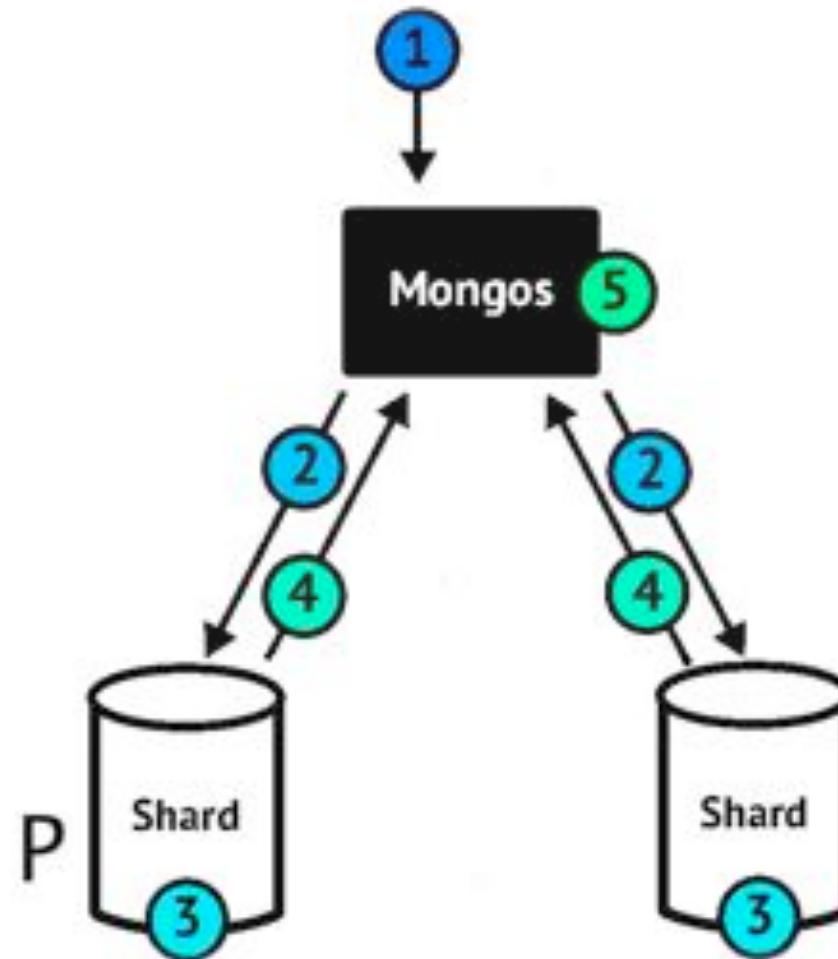
SHARDING



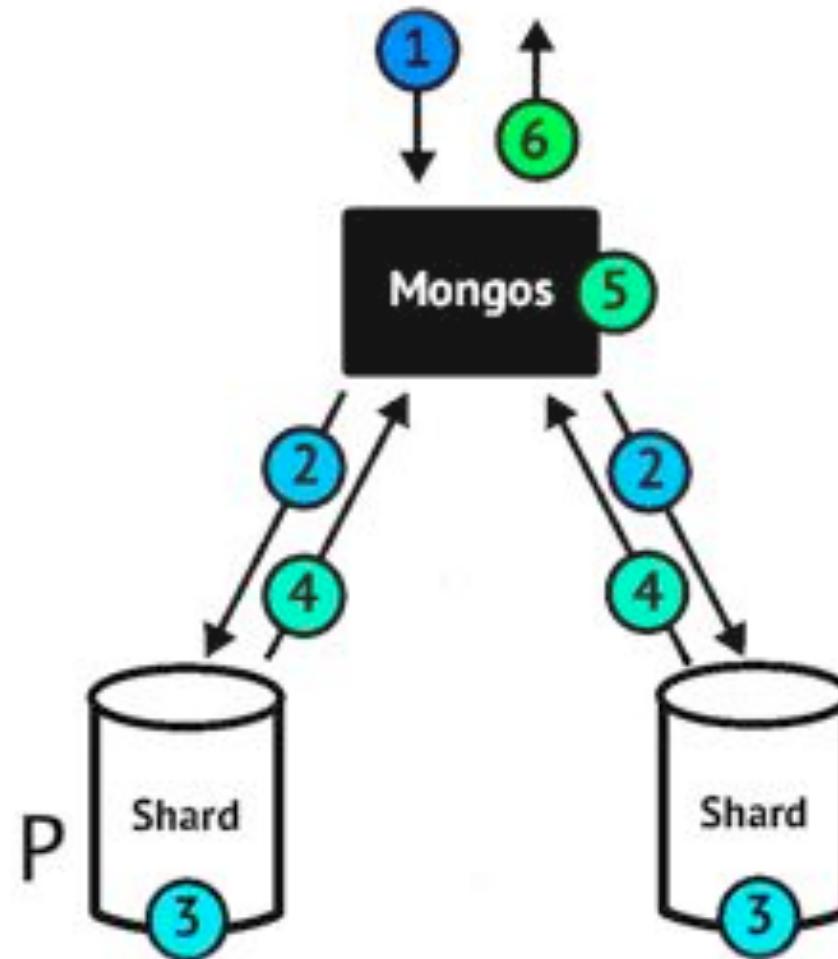
SHARDING



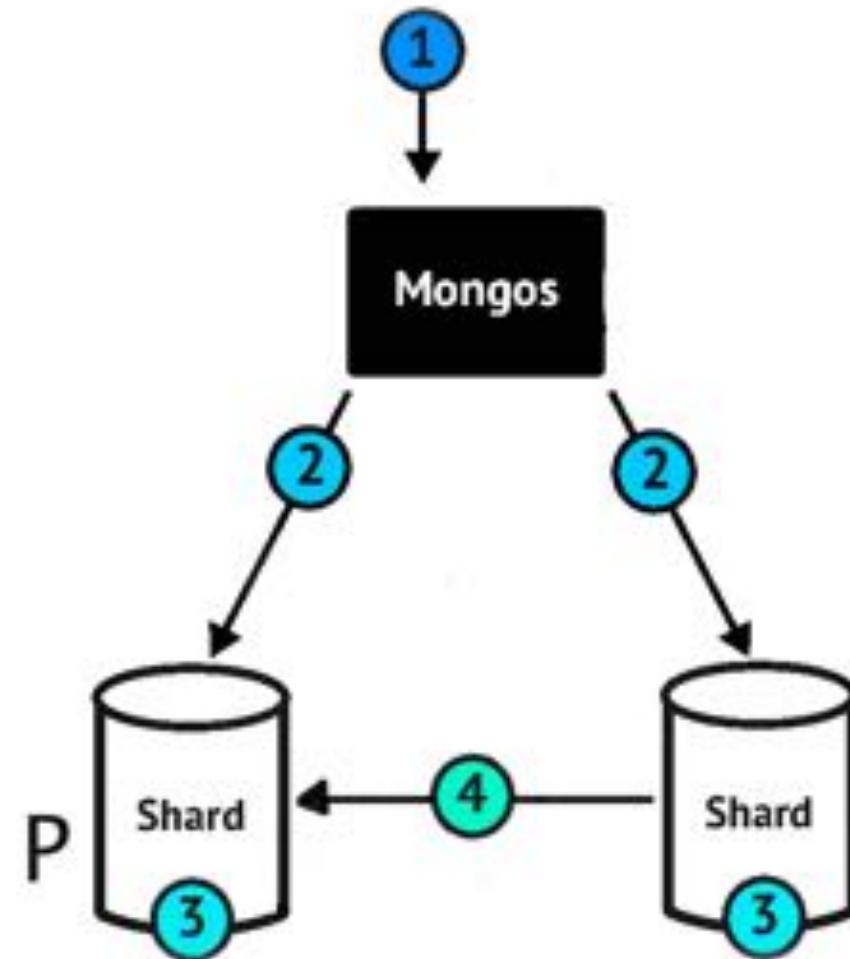
SHARDING



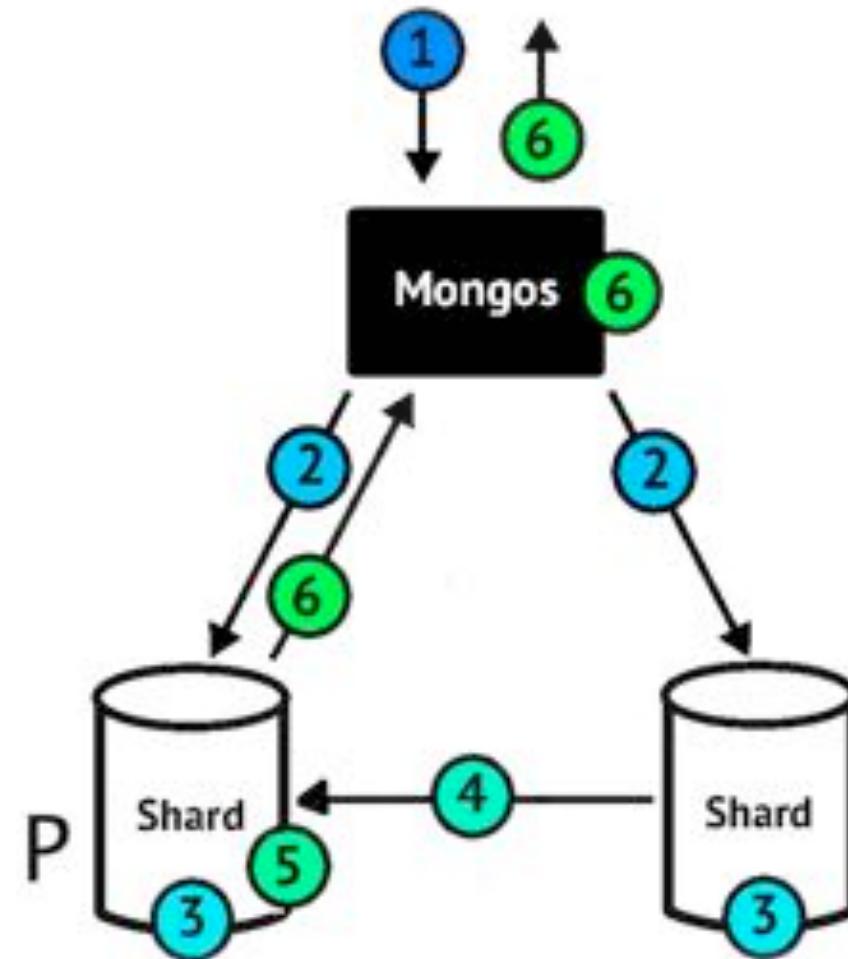
SHARDING



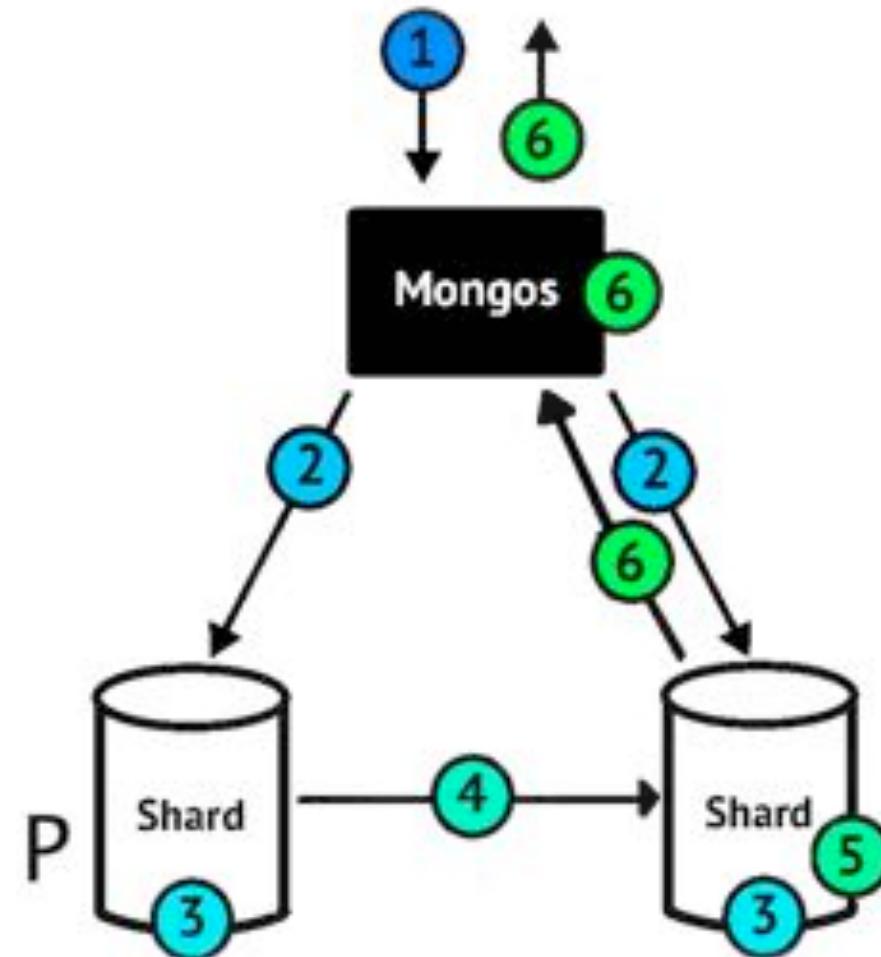
SHARDING AGGREGATION



SHARDING AGGREGATION



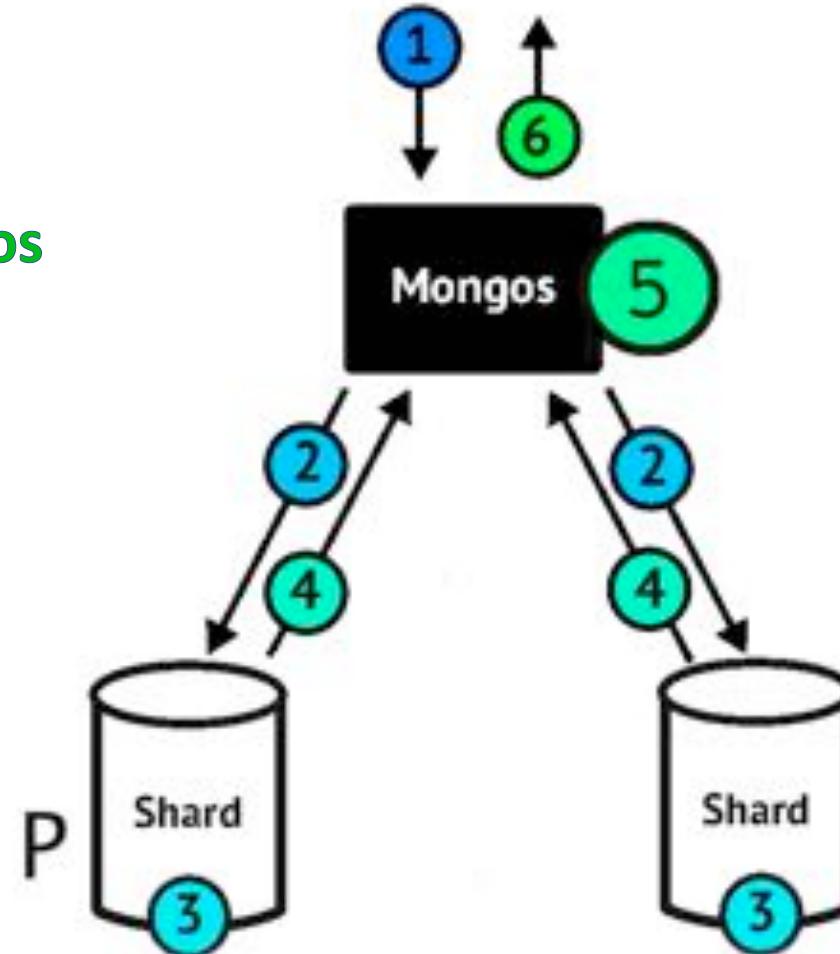
SHARDING AGGREGATION



SHARDING AGGREGATION

new in 3.6:

- merge back on mongos



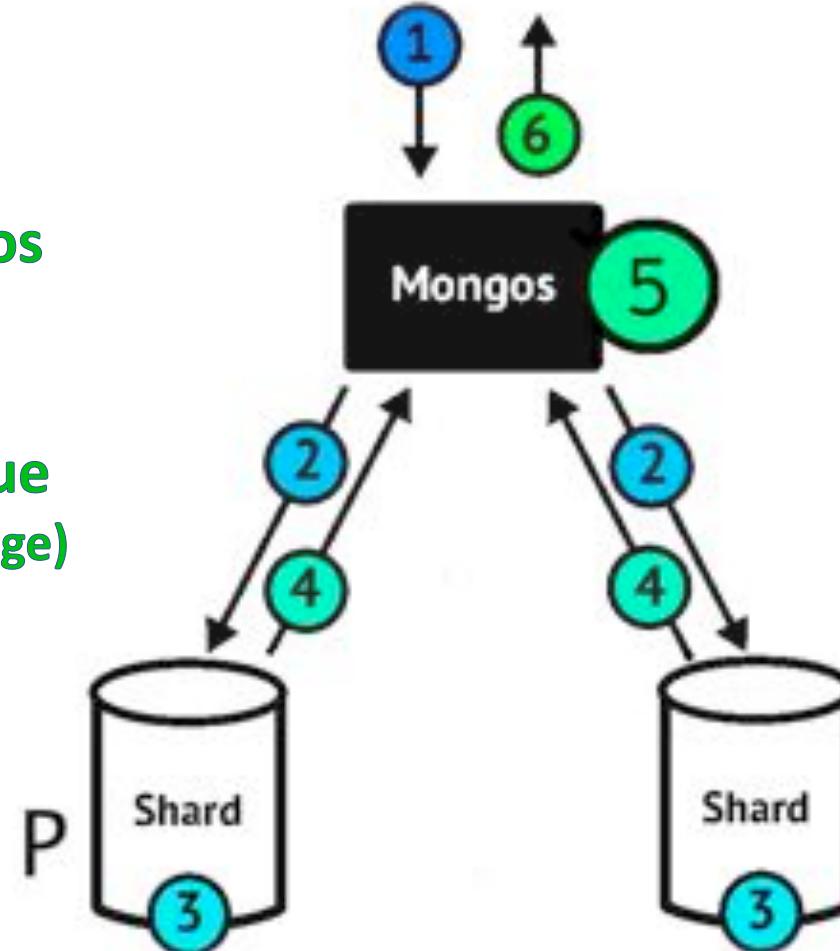
SHARDING AGGREGATION

new in 3.6:

- merge back on mongos
- exceptions:

\$out stage

allowDiskUse: true
(for blocking stage)



SHARDING AGGREGATION

Work starts on individual shards
only "targeted" shards

"Merging" for first "blocking" stage is split into two parts
shards part (each shard starts the merging work)
merge part (merging finishes at "merge" location)

Final merge can happen on primary shard, any shard or mongos
on mongos (starting in 3.6 only)

Merging with only streaming stages happens on 'mongos' same as "find"

I CAN EXPLAIN



MONGODB.local
SYDNEY

#MDBlocal

```
db.books.aggregate([
  {$unwind:"$subjects"},  

  {$match:{"language":"English","subjects":/^ [ABC] /},  

  {$group:{_id:"$subjects",count:{$sum:1}}},  

  {$sort:{count:-1}},  

  {$limit:3} ], {explain:true})
```

```
db.books.aggregate([
  {$unwind:"$subjects"},  

  {$match:{ "language": "English", "subjects": /[^ABC]/ }},  

  {$group:{_id:"$subjects", count:{ $sum:1 }}},  

  {$sort:{count:-1}},  

  {$limit:3} ], {explain:true})  

{  

  mergeType:  

  splitPipeline:  

  shards:  

}  

}
```

```
db.books.aggregate([
  {$unwind:"$subjects"},  
  {$match:{language:"English",subjects:/^ABC/}},  
  {$group:{_id:$subjects,count:{$sum:1}}},  
  {$sort:{count:-1}},  
  {$limit:3} ], {explain:true})  
{  
  mergeType:  
    splitPipeline:  
      shards:  
        "primaryShard"  
        "anyShard"  
        "mongos"  
  }  
}
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true})
```

```
{
```

```
  mergeType: "mongos",
```

```
  splitPipeline:
```

```
  shards:
```

```
}
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true, allowDiskUse: true })
```

```
{
  mergeType: "anyShard",
  splitPipeline:
  shards:
}
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3}, {$out: "output"} ], {explain:true})
```

```
{
  mergeType: "primaryShard",
  splitPipeline:
  shards:
}
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true})
{
  mergeType:
splitPipeline: { "shardsPart" : [
  shards:
    ],
  "mergerPart" : [
    ] },
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true})
{
  mergeType:
splitPipeline:  { "shardsPart" : [
    shards:
      {"$match" :{"language": {"$eq":"English"}},,
      {"$unwind": {"path": "$subjects"}},,
      {"$match" :{"subjects": {"$regex":^ABC"}},,
      {"$group" :{"_id":$subjects,"count":{$sum:1}}}
    ],
    "mergerPart" : [
    ] },,
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true})
{
  mergeType:
splitPipeline: { "shardsPart" : [
    {"$match" :{"language": {"$eq":"English"}}, "$unwind": {"path": "$subjects"}, {"$match" :{"subjects": {"$regex":^ABC"}}, {"$group" :{"_id":$subjects,"count":{$sum:1}}}
  ],
  "mergerPart" : [
    {"$group":{ "_id":$$ROOT.id,
      "count":{$sum:$$ROOT.count},
      "$doingMerge" : true}},
    {"$sort":{ "sortKey":{count: -1},
      "limit":NumberLong(3)}}
  ] },
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true})
{
  mergeType:
splitPipeline: { "shardsPart" : [
    {"$match" :{"language": {"$eq":"English"}},,
    {"$unwind": {"path": "$subjects"}},
    {"$match" :{"subjects": {"$regex":^ABC"}},,
    {"$group" :{"_id":$subjects,"count":{$sum:1}}}
  ],
  "mergerPart" : [
    {"$group":{
      "_id":$$ROOT._id",
      "count":{$sum:$$ROOT.count},
      "$doingMerge : true
    }},
    {"$sort":{
      "sortKey":{"count" : -1},
      "limit":NumberLong(3)}}
  ] },
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true})
```

```
{
  mergeType:
  splitPipeline:
```

```
shards: { "shard0000" : {
```

```
},
```

```
"shard0001" : {
```

```
}
```

```
}
```

```
}
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true})
{
  mergeType:
  splitPipeline:
  shards: { "shard0000" : {
    "host": <hostname>,
    "stages": [ ]
  },
  "shard0001" : {
    "host": <hostname>,
    "stages": [ ]
  }
}
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{"language":"English","subjects":/^ABC/}},
  {$group:{_id:"$subjects",count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true})
{
  mergeType:
  splitPipeline:
  shards: { "shard0000" : {
    "host": <hostname>,
    "stages": [
      {"$cursor": {
        "query": {"language": {"$eq": "English"}},
        "fields": {"subjects": 1, "_id": 0},
        "queryPlanner": {"plannerVersion": 1, "namespace": "t
      },
      {"$unwind": {"path": "$subjects"}},
      {"$match": {"subjects": {"$regex": "ABC"}}},
      {"$group": {"_id": "$subjects", "count": {"$sum": 1}}}
    ]
  }
}
```

```
db.books.aggregate([
{$unwind:"$subjects"},
{$match:{"language":"English","subjects":/^ABC/}},
{$group:{_id:"$subjects",count:{$sum:1}}},
{$sort:{count:-1}},
{$limit:3} ], {explain:true})
{
  mergeType:
  splitPipeline:
  shards: { "shard0000" : {
    "host": <hostname>,
    "stages": [
      {"$cursor": {
        "query": {"language": {"$eq": "English"}},
        "fields": {"subjects": 1, "_id": 0},
        "queryPlanner": {"plannerVersion": 1, "namespace": "test.books",
          "winningPlan": {
            "stage": "SHARDING_FILTER",
            "inputStage": {
              "stage": "FETCH",
              "inputStage": {
                "stage": "IXSCAN",
                "keyPattern": {"language": 1, "subjects": 1},
                "indexName": "language_1_subjects_1", "isMultiKey": false,
              }
            }
          }
        },
        {"$unwind": {"path": "$subjects"}},
        {"$match": {"subjects": {"$regex": "ABC"}}}
      }
    ]
  }
}
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true})
```

```
{
  mergeType:
  splitPipeline:
  shards:
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{language:"English",subjects:/^ABC/}},
  {$group:{_id:$subjects,count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true})
```

```
{
  mergeType:
  splitPipeline:
  shards: {
    "shard0000" : {
      ...
    }
  }
}
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{"language":"English","subjects":/^ABC/}},
  {$group:{_id:"$subjects",count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true})
```

```
{
```

```
  mergeType:
```

```
  splitPipeline: null,
```

```
  shards: {
```

```
    "shard0000" : {
```

```
      ...
```

```
    }
```

```
}
```

```
}
```

```
db.books.aggregate([
  {$unwind:"$subjects"},
  {$match:{"language":"English","subjects":/^ABC/}},
  {$group:{_id:"$subjects",count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:3} ], {explain:true})
```

```
{
```

```
  splitPipeline: null,
  shards: {
```

```
    "shard0000" : {
```

```
      ...
    }
```

```
}
```

```
}
```

```
db.orders.explain().aggregate( [  
    {$match:{status:"F"}},  
    {$sort:{totalprice:-1}},  
    {$limit:40},  
    {$group:{_id:1,avgprice:{$avg:"$totalprice"}}}])
```

```
db.orders.explain().aggregate( [  
    { $match: { status: "F" } },  
    { $sort: { totalprice: -1 } },  
    { $limit: 40 },  
    { $group: { _id: 1, avgprice: { $avg: "$totalprice" } } } ] )  
  
{ "stages" : [  
    { "$cursor" : { "query" : { "status" : "F" },  
        "sort" : { "totalprice" : -1 },  
        "limit" : NumberLong(40),  
        "fields" : { "totalprice" : 1, "_id" : 0 },  
        "queryPlanner" : {  
            "winningPlan" : {  
                "stage" : "PROJECTION",  
                "transformBy" : { "totalprice" : 1, "_id" : 0 },  
                "inputStage" : {  
                    "stage" : "IXSCAN",  
                    "indexName" : "status_1_totalprice_1",  
                    "direction" : "backward",  
                    "indexBounds" : {  
                        "status" : [ "[\"F\", \"F\"]" ],  
                        "totalprice" : "[MaxKey, MinKey]"  
                    } } } },  
    { } ,  
    { "$group" : { "_id" : 1, "avgprice" : { "$avg" : "$totalprice" } } }  
] }
```

```
db.orders.explain("executionStats").aggregate( [  
    {$match:{status:"F"}},  
    {$sort:{totalprice:-1}},  
    {$limit:40},  
    {$group:{_id:1,avgprice:{$avg:"$totalprice"}}}])  
{"stages": [  
    {"$cursor": { "query" : { "status" : "F"},  
        "sort" : { "totalprice" : -1},  
        "limit" : NumberLong(40),  
        "fields" : { "totalprice" : 1, "_id" : 0},  
        "queryPlanner" : {  
            "winningPlan" : {  
                "stage" : "PROJECTION",  
                "transformBy" : { "totalprice" : 1, "_id" : 0},  
                "inputStage" : {  
                    "stage" : "IXSCAN",  
                    "indexName" : "status_1_totalprice_1",  
                    "direction" : "backward",  
                    "indexBounds" : {  
                        "status" : [ "[\"F\", \"F\"]"],  
                        "totalprice" : "[MaxKey, MinKey]"  
                    } } } },  
    } },  
    {"$group" : {"_id" : 1, "avgprice" : {"$avg" : "$totalprice"} } }  
] }
```

```
db.orders.explain("executionStats").aggregate( [  
    {$match:{status:"F"}},  
    {$sort:{totalprice:-1}},  
    {$limit:40},  
    {$group:{_id:1,avgprice:{$avg:"$totalprice"}}}])  
{"stages": [  
    {"$cursor": { "query" : { "status" : "F"},  
        "sort" : { "totalprice" : -1},  
        "limit" : NumberLong(40),  
        "fields" : { "totalprice" : 1, "_id" : 0},  
        "queryPlanner" : {  
            "winningPlan" : {  
                "stage" : "PROJECTION",  
                "transformBy" : { "totalprice" : 1, "_id" : 0},  
                "inputStage" : {  
                    "stage" : "IXSCAN",  
                    "indexName" : "status_1_totalprice_1",  
                    "direction" : "backward",  
                    "indexBounds" : {  
                        "status" : [ "[\"F\", \"F\"]"],  
                        "totalprice" : "[MaxKey, MinKey]"  
                    } } } },  
    } },  
    {"$group" : {"_id" : 1, "avgprice" : {"$avg" : "$totalprice"} } }  
] }
```

```
db.orders.explain("executionStats").aggregate( [  
    {$match:{status:"F"}},  
    {$sort:{totalprice:-1}},  
    {$limit:40},  
    {$group:{_id:1,avgprice:{$avg:"$totalprice"}}}])  
{"stages": [  
    {"$cursor": { "query" : { "status" : "F"},  
        "sort" : { "totalprice" : -1},  
        "limit" : NumberLong(40),  
        "fields" : { "totalprice" : 1, "_id" : 0},  
        "queryPlanner" : {  
            "winningPlan" : {  
                "stage" : "PROJECTION",  
                "transformBy" : { "totalprice" : 1, "_id" : 0},  
                "inputStage" : {  
                    "stage" : "IXSCAN",  
                    "indexName" : "status_1_totalprice_1",  
                    "direction" : "backward",  
                    "indexBounds" : {  
                        "status" : [ "[\"F\", \"F\"]"],  
                        "totalprice" : "[MaxKey, MinKey]"  
                    } } } },  
        "executionStats" : {  
            "executionSuccess" : true,  
            "nReturned" : 85,  
            "executionTimeMillis" : 3,  
            "totalKeysExamined" : 30295,  
            "totalDocsExamined" : 0,  
            "executionStages" : { ... }  
        } } ],  
    {"$group" : {"_id" : 1, "avgprice" : {"$avg" : "$totalprice"} } }  
] }
```

NEW IN 3.6

executionStats option to explain
"hint" for aggregation

```
db.orders.explain().aggregate( [  
    {$match:{status:"F"}},  
    {$sort:{totalprice:-1}},  
    {$limit:40},  
    {$group:{_id:1,avgprice:{$avg:"$totalprice"}}}  
, {"hint":{"status":1, "totalprice":1}})
```

POWERFUL AGGREGATIONS

understand stages

- Best order for performance
- Avoid unnecessary "blocking"
 - keep "streaming"
- Maximize use of indexes
 - early stages get the index!
- Liberally check explain() output

understand expressions

- Schema manipulation
- Array transformation
- Use in find query filter, ...

use functions

- Readable, debug-able, reusable

THE FUTURE OF AGGREGATION

Better performance & optimizations

More stages & expressions

More options for output

Compass helper for aggregate

Unify different languages

<https://github.com/asya999/mdbw17>

AGGREGATE

is the new FIND

<https://github.com/asya999/mbw17>

#MDBlocal



THANK YOU!

<https://github.com/asya999/mbw17>