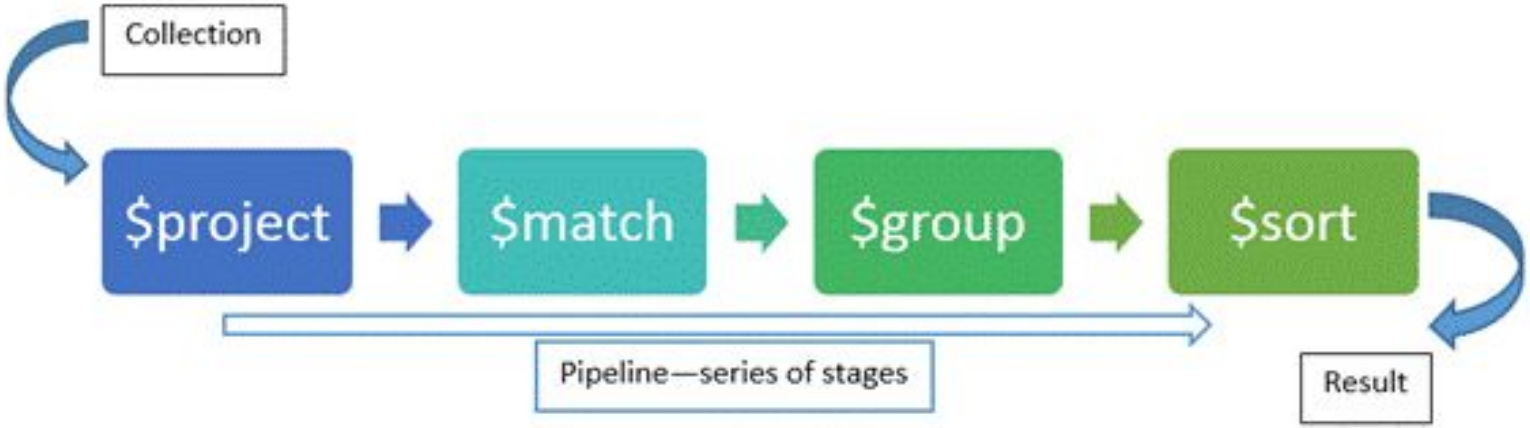
MongoDB & NoSQL Analytics

The Aggregation Pipeline Framework

What is the Aggregation Framework?

Set of analytics tools within MongoDB that allows you to run various reports or analysis on one or more MongoDB collections.



Aggregation Pipeline Mapping to SQL functions

SQL Terms	`MongoDB Agg`	Explanation	Example
WHERE	$match	Filter documents	
GROUP BY	$group	Group documents by value, summarize documents. Applies accumulator expression to each group	
HAVING	$match	Filters documents with respect to specific criteria that are passed on to next stage of pipeline	
SELECT	$project	Reshape documents, include exclude fields, create new fields	
ORDER BY	$sort	Reorder document with respect to specific sort key	
SUM()	$sum	Returns sum of each group. Ignores non-numeric values	
COUNT()	$sum	See above	
join	$lookup	Performs left outer join	
N/A	$unwind	Deconstructs an array field and returns a document for each array element	At Twitter you want to figure out who included the most user mentions in their tweet. In this case, user mentions is an array within a tweet

Question 1

$project, $group, $sum

How many total users are there?
Only return the total.

How many total users are there? Only return the total.

In aggregation, the total number of documents collection or individual inputs is a **$sum**

How many total users are there?

Only return the total.

Similar to the SELECT in SQL,
in aggregation, this indicates
what values should be
returned via **$project**

How many total users are there?
Only return the total.

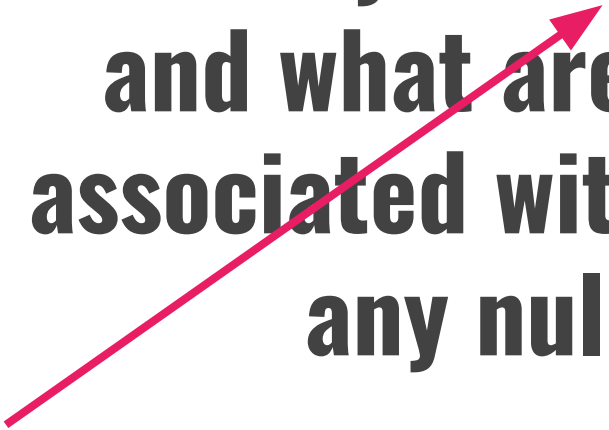
```
// Number of users per category
//filtering out nulls and empty values
   db.users.aggregate([
       { $group: {
           _id: null, count: {$sum: 1}
// _id refers to which fields to return,
//and since we are just looking for the total number of
//documents, we can just make it 'null'
           }
        },
        { $project: {
 // now we must use project in order to
 //only return the count and not the _id
            _id:0, count:1
            }
        }
    ])

//result:
{ "count" : 450 }
```

Question 2

$group, $match, $push, $sum

How many total users are there per offer and what are the respective names associated with those offers ? Remove any null or empty values.

How many total users are there per offer and what are the respective names associated with those offers ? Remove any null or empty values.



Total/per combination should instantly make you think I need a **$group** of items

How many total users are there per offer and what are the respective names associated with those offers ? Remove any null or empty values.

Total/per combination should instantly make you think **$addToSet or $push**

How many total users are there per offer and what are the respective names associated with those offers ? Remove any null or empty values.

‘Remove’, ‘filter out’ is another way of saying un-“**$match**” these documents from the output

How many total users are there per offer and what are the respective names associated with those offers ? Remove any null or empty values.

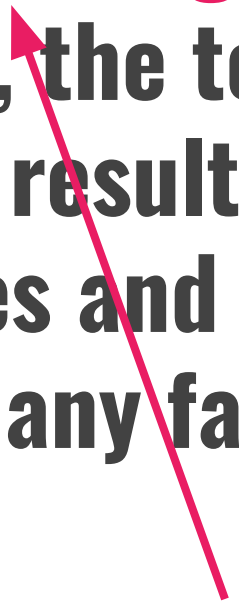
```
db.users.aggregate([
    { $match:
// similar to the WHERE clause in SQL, or, in the case of an aggregate, the HAVING clause
        { $and: [{offer: {$ne: ""}},
                {offer: {$ne: null} }]
        }
    },
    { $group:
        {
        _id:'$offer',
        namesArray: {$push: '$profile.name'},
        count: {$sum: 1}
        }
    },
    { $sort: {
        count: -1 }
    }
])
{ "_id" : "Angel/Seed",
"namesArray" : [ "Sonya Sepahban", "Sally Kang", .... ],
"count" : 33 }
{ "_id" : "Other",
"namesArray" : [ "Sydney Spraggins",  .... ],
 "count" : 24 }
```

Question 3

`$group, $unwind, $push, $sum`

Count total number of events per category, list the respective events by title, the total number of attendees, and list the results by the most profitable categories and respective total profits. Remove any false values.

Count total number of events per category, list the respective events by title, the total number of attendees, and list the results by the most profitable categories and respective total profits. Remove any false values.



Again, the **per category** should make you think **$group** and, maybe, **$unwind** if the category field is an array

Count **total number** of events per category, list the respective events by title, the **total number** of attendees, and list the results by the most profitable categories and **respective total profits**. Remove any false values.

Total should make you think **$sum** or **$size** depending on the field type

Count total number of events per category, list the respective events by title, the total number of attendees, and list the results by the most profitable categories and respective total profits. Remove any false values.

To **list** results in any fashion, should indicate to you some type of **$sort**

Count total number of events per category, list the respective events by title, the total number of attendees, and list the results by the most profitable categories and respective total profits. Remove any false values.

```
db.listings.aggregate([
// unwind events by categories field to create a copy of the event for each categories array value
    { $unwind : "$categories"},
// then we must group those results by category
//and $sum and $multiply in order to evaluate
//the number of events,
// attendees and profitability
    { $group : {
        _id: "$categories",
        events_array: { $push : "$title"},
        numberOfEvents: { $sum: 1 },
        numberOfAttendees:{
            $sum: {$size:'$listing_users'}
            },
        profitability: { $sum: {
            $multiply: [  {$size: "$paid_users"}, "$price" ] }
            }
        }
    },
// sort by created field profitability
//to determine most profitable categories, descending
    { $sort: {profitability : -1}}
])

// results
{ "_id" : "General Business",
"events_array" : [ "Build Your Dream team", "Early Stage Startup Success Factors at Pepperdine (West LA)"
 "numberOfEvents" : 30,
 "numberOfAttendees" : 197,
 "profitability" : 790 }

{ "_id" : "Angel/Seed",
"events_array" : [ "Startegies for Building Your Dream Team and Fundraising", ... ],
 "numberOfEvents" : 10,
 "numberOfAttendees" : 77,
 "profitability" : 250 }
```

Question 4

$group, $match, $push, $lookup

Calculate the percentage distribution of Pin categories in the Post document titled “Build a Dream Team”.

Calculate the percentage distribution of Pin categories in the Post document titled “Build a Dream Team”.

Calculate should indicate some sort of mathematical operator, such as **$sum, $multiply, and/or $divide**

Calculate the percentage distribution of Pin categories in the Post document titled “Build a Dream Team”.

This indicates a specific Post document we should find or **$match**

```
"pin_array" : [ "57e41d788277a00300a7b02e",
ObjectId("57eaa3dfeeff760300103c3d"),
ObjectId("57ebff744a353b030029d781"),
ObjectId("5838d70699812a04008b0203"),
ObjectId("583a158becd6db040040407d"),
"5895d0b5df391f000388779b",
ObjectId("58ae816c72ef240003ffea97"),
ObjectId("58af768472ef240003ffeaca") ],
```

$lookup

```
db.posts.aggregate([
    {$match : {"_id" : ObjectId("57e413f200223203000d62d9")}},
    {$unwind : "$pin_array"},
// return a copy of post document for each
//pin_array. At this point,
// each element, if found in
    {$lookup : {
         from: "pins", localField: "pin_array",
         foreignField: "_id", as: "pin_docs"}},
// remove unmatched Pin array elements
    {$match: {"pin_docs": {$ne: []}} },
// unwind out of array i.e. flatten it. Though
//the array contains only one Pin document,
//you must flatten the array to
//return the right results
    {$unwind : "$pin_docs"},
    {$unwind : "$pin_docs.pin_categories"},
// group all pins by category and normalize
//categories to lower case, in case there are differences. Count number of categories present
```

$lookup results

```
{ "_id" : ObjectId("57e413f200223203000d62d9"),
"pin_id" : [ ObjectId("57eaa3dfeeff760300103c3d") ],
"pin_title" : [ "Recruiting Strategies for Startups" ],
"pin_cat" : [ [ "Human Resources" ] ] }

{ "_id" : ObjectId("57e413f200223203000d62d9"),
"pin_id" : [ ObjectId("57ebff744a353b030029d781") ],
"pin_title" :
[ "The 5 Key Dynamics That Make A Great Team" ],
"pin_cat" : [[ "human resources", "Human Resources" ]] }
```

$group Part I

```
// group all pins by category and normalize
//categories to lower case, in case there are differences. Count number of categories present
    {$group :{
        _id: { pin_cat: {
                $toLower: "$pin_docs.pin_categories"}},
        records: { $push : "$pin_docs.title"},
        count: { $sum: 1 }}
    },
// project values from the group so that
// we can easily collect the group into one document.
// create temporary variable to do this
    { $project: {
        tmp: {
            _id: '$_id',
            records: '$records',
            count: '$count'
        }
    }},
```

$group Part I results


```
{ "_id" : { "pin_cat" : "other" }, "tmp" : { "_id" : {
"pin_cat" : "other" }, "records" : [ "Teams deck" ],
"count" : 1 } }

{ "_id" : { "pin_cat" : "human resources" },
"tmp" : { "_id" : { "pin_cat" : "human resources" },
"records" : [ "Recruiting Strategies for Startups", "The
5 Key Dynamics That Make A Great Team", "The 5 Key
Dynamics That Make A Great Team", "4 Traits to Look for
When Hiring Remote Workers (UpWork)", "Federal Court
Blocks New Overtime Rule (By Littler, 11/23/16)",
"2-23-17" ], "count" : 6 } }
```

$group Part II

```
// now group all the inputs into one input to
// get the the total number of inputs,
// where the pin_category_group array
// represents a grouping of pins
// based on category
    {$group: {
        _id: null,
        total:{$sum: "$tmp.count"},
        pin_category_group: {$push: "$tmp"}}
    },
// unwind the pin_category_group group
// to do the individual math that each
// category requires to discover the
// distribution
    {$unwind : "$pin_category_group"},
```

$group Part II results


```
{ "_id" : null, "total" : 7,

"pin_category_group" : [
    {"_id" : { "pin_cat" : "other" },
    "records" : [ "Teams deck" ], "count" : 1 },

    { "_id" : { "pin_cat" : "human resources" },
    "records" : [ "Recruiting Strategies for Startups",
    "The 5 Key Dynamics That Make A Great Team", "The 5
    Key Dynamics That Make A Great Team", "4 Traits to
    Look for When Hiring Remote Workers (UpWork)",
    "Federal Court Blocks New Overtime Rule (By Littler,
    11/23/16)", "2-23-17" ],
     "count" : 6 } ] }
```

$unwind & $project

```
// unwind the pin_category_group group
// to do the individual math that each
// category requires to discover the
// distribution
    {$unwind : "$pin_category_group"},
    {$project : {
        _id: "$pin_category_group._id",
        records: "$pin_category_group.records",
        count: "$pin_category_group.count",
        total: 1,
        percentage: {
            $multiply: [
            { $divide:
                [ "$pin_category_group.count",
                "$total"]
            }, 100]
            }
        }
    }
```

$unwind & $project results


```
{ "_id" : { "pin_cat" : "other" },
 "total" : 7, "records" : [ "Teams deck" ],
  "count" : 1,
  "percentage" : 14.285714285714285 }

{ "_id" : { "pin_cat" : "human resources" },
"total" : 7,
"records" : [ "Recruiting Strategies for Startups", "The
5 Key Dynamics That Make A Great Team",
"The 5 Key Dynamics That Make A Great Team", "4 Traits to
Look for When Hiring Remote Workers (UpWork)",
"Federal Court Blocks New Overtime Rule (By Littler,
11/23/16)", "2-23-17" ],
 "count" : 6, "percentage" : 85.71428571428571 }
```

Calculate the percentage distribution of Pin categories in a Post document.

```
db.posts.aggregate([
    {$match : {"_id" : ObjectId("57e413f200223203000d62d9")}},
    {$unwind : "$pin_array"},
    {$lookup : {
         from: "pins", localField: "pin_array",
         foreignField: "_id", as: "pin_docs"}},
    {$match: {"pin_docs": {$ne: []}} },
    {$unwind : "$pin_docs"},
    {$unwind : "$pin_docs.pin_categories"},
    {$group :{
        _id: { pin_cat: {
                $toLower: "$pin_docs.pin_categories"}},
        records: { $push : "$pin_docs.title"},
        count: { $sum: 1 }}
    },
    { $project: {
        tmp: { _id: '$_id',
        records: '$records', count: '$count'}}
    },
    {$group: {
        _id: null,
        total:{$sum: "$tmp.count"},
        data: {$push: "$tmp"}}
    },
    {$unwind : "$data"},
    {$project : {
        _id: "$data._id", records: "$data.records", count: "$data.count", total: 1,
        percentage: {
            $multiply: [
                { $divide: [ "$data.count", "$total"] }, 100]
                }
        }
    }
])

// results
{ "_id" : { "pin_cat" : "other" },
 "total" : 7, "records" : [ "Teams deck" ],
  "count" : 1,
  "percentage" : 14.285714285714285 }
```