

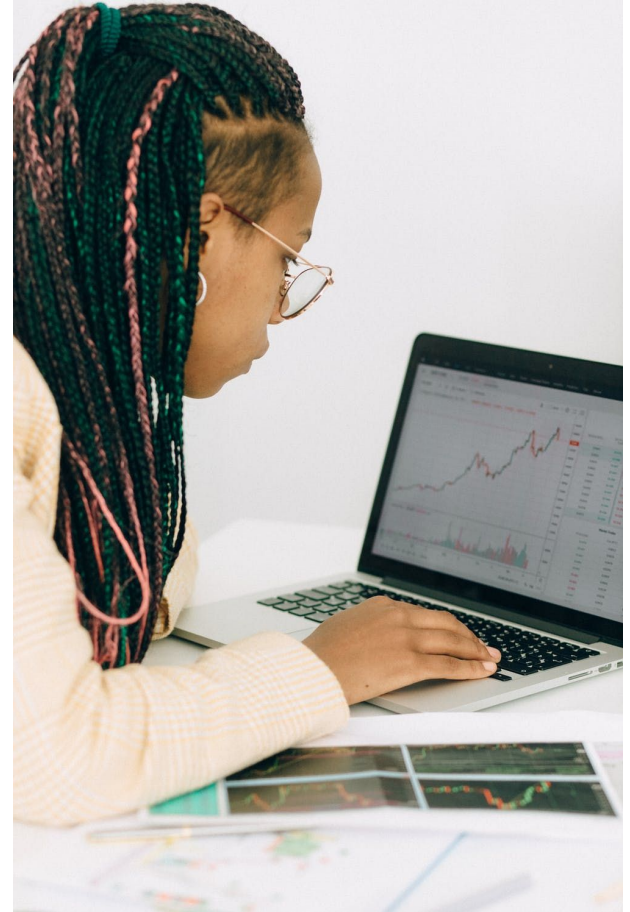
# Aggregation in MongoDB



# What is aggregation in the context of databases?

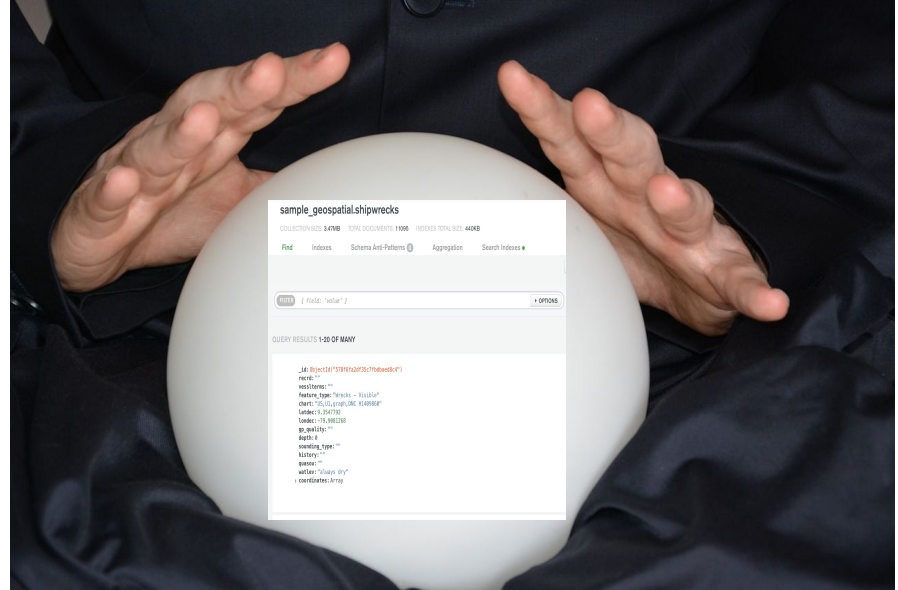
**Aggregation is filtering and sorting data in a database to find answers to our questions.**

MongoDB has developed an aggregation tool in order to allow data managers and developers to be able to filter and sort data at a raw-data, low-level.



## When might we use aggregation?

- When we have a MongoDB collection that we would like to know something about
- When we just need an easier way to find and look at certain documents
- When we want to know something from our data, but don't need to necessarily build it into an application
- When we quickly need a report on something in our data, but don't have a chart yet

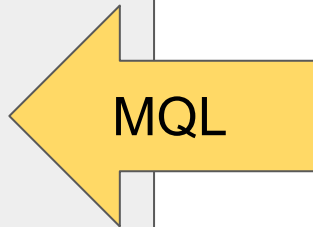


# What about MQL?

Aggregation actually uses Mongo Query Language (MQL) when you're writing what you want to be looking for in your data.

However, we don't have to rely wholly on MQL because Aggregation in MongoDB was designed to be a much easier way to create a series of queries that will return the information that you are looking for.

```
db.students.aggregate([
  { $match: { status: "enrolled" } },
  { $group: { _id: "$classroom",
              numberOfStudentsInEachClass: {
                $count: {}
              }
            }
  }
])
```



# What is the Aggregation Pipeline?

This is the series of steps that we take in order to find the information that we're looking for.

The first stage starts with all of the data in the collection, and then each following stage will only look at the documents that have met the criteria in the preceding stages.

Each stage of the pipeline is one main operation.



# What are the most common aggregation operations?

## **\$match**

Keep documents that match this criteria.  
Like SELECT \*  
WHERE in SQL

## **\$project**

For checking whether a value appears in an array in your data, and adding a way to reference that

## **\$sort**

Allows you to sort documents based on a value in them. Can be sub-sorted also!

## **\$group**

Allows us to group documents by index so we can do further operations, like find averages and sums

## **\$count**

Tells us how many documents there are at this stage in the pipeline

## **\$sum and \$avg**

Can be used with other stages to find sums and averages from the documents in the pipeline



**Let's see an example!**

