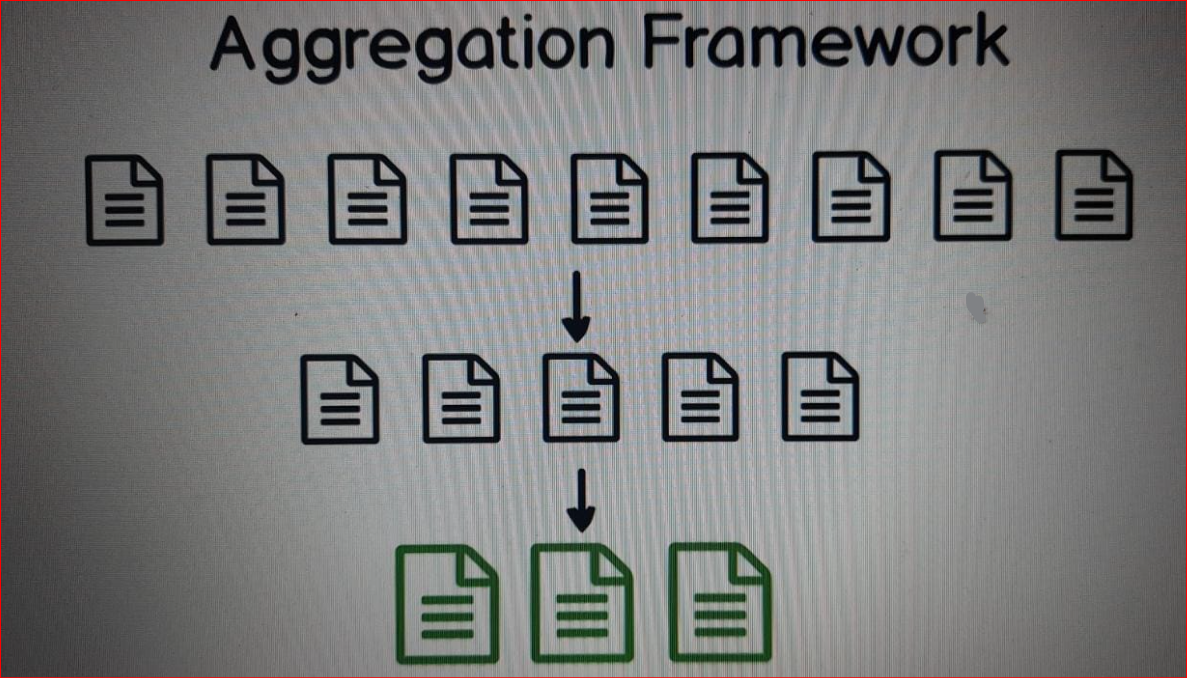
**34. MongoDB Aggregation Framework**

In this section we will talk about some advanced MongoDB features and first topic is **Aggregation Framework.**



* **What is the purpose of this Aggregation Framework?**

Using Aggregation Framework you can take documents in your collection,

**you can find a subset of the documents and then produce brand new documents at the end.**

For example using Aggregation Framework

you can find all distinct values of the tags in "tags" array in all documents

you can find all values of the "comments" field

or you can find all titles in the documents and so on.

This framework is very powerful

I'll show you a couple of examples of the usage of Aggregation Framework.

**35. PRACTICE - Aggregating Documents**

Let's go back to Robo 3T (Mongo 3T). I'll close all tabs and open new shell. Here I'll use aggregate syntax.

"db.posts.aggregate()" and aggregate method needs an array, array of objects.

// Aggregate Syntax

**// {$group: {\_id: "$author.name"}}**

**// let's find all different author names in the documents. eg: "author" : { "name" : "Mike Forester",**

**// use $group operator: Curly braces and here $group. Then colon and another object "\_id", colon then in double quotes "$author.name"**

db.posts.aggregate([

{$group: {\_id: "$author.name"}}

])

**// OutPut/ Result:**

//{

// "\_id" : "Emily Watson"

//}

//{

// "\_id" : "Mike Forester"

//}

//{

// "\_id" : "Bob Hutchinson"

//}

**//{$group: {\_id: "$author.nickname"}}**

**// Let's change "author.name" to "author.nickname" and run this.**

db.posts.aggregate([

{$group: {\_id: "$author.nickname"}}

])

**// OutPut/ Result:**

//{

// "\_id" : "mikef"

//}

//{

// "\_id" : "bob1995"

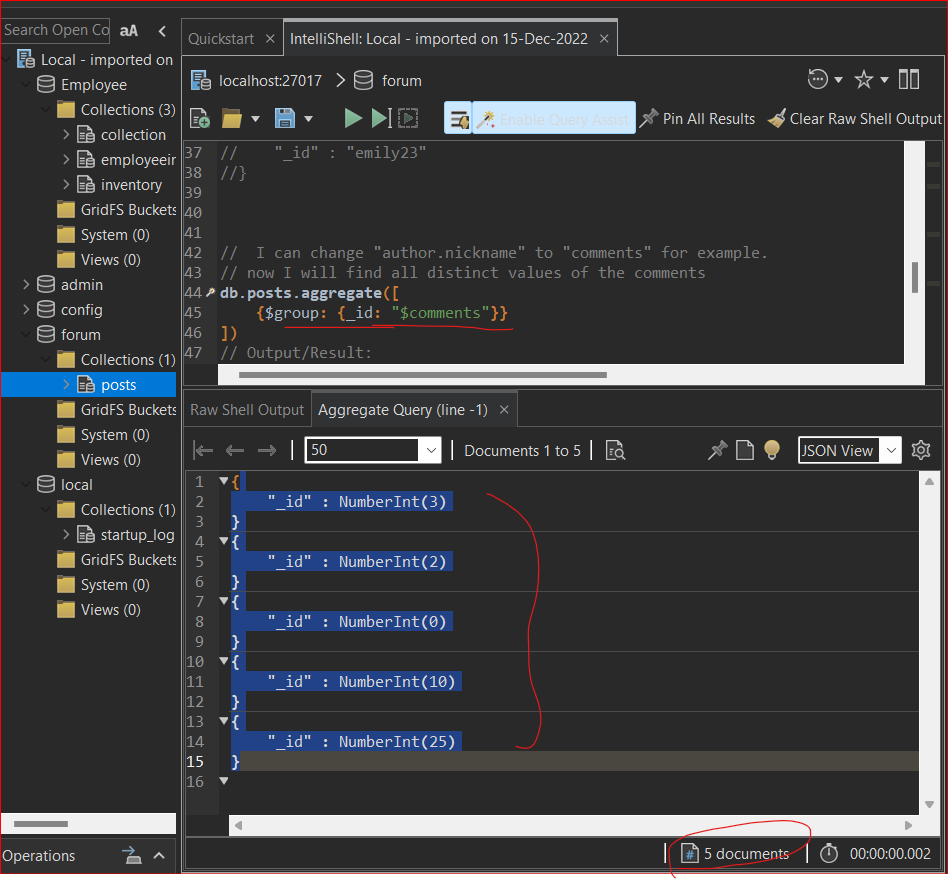
//}

//{

// "\_id" : "emily23"

//}

**//{$group: {\_id: "$comments"}}**

****

**// I can change "author.nickname" to "comments" for example.**

**// now I will find all distinct values of the comments**

db.posts.aggregate([

{$group: {\_id: "$comments"}}

])

**// Output/Result:**

// "\_id" : NumberInt(3)

//}

//{

// "\_id" : NumberInt(2)

//}

//{

// "\_id" : NumberInt(0)

//}

//{

// "\_id" : NumberInt(10)

//}

//{

// "\_id" : NumberInt(25)

//}

**let's move to the next section where we will talk about MongoDB Indexes**