**MONGODB**

[**1.** **$match:** 2](#_Toc88236181)

[**2.** **$group:** 2](#_Toc88236182)

[**3.** **$count** 3](#_Toc88236183)

[**4.** **$sort** 4](#_Toc88236184)

[**5.** **$project** 4](#_Toc88236185)

[**6.** **$limit** 6](#_Toc88236186)

[**7.** **$sum** 6](#_Toc88236187)

[**8.** **$skip** 6](#_Toc88236188)

[**9.** **Tip to update records using forEach** 6](#_Toc88236189)

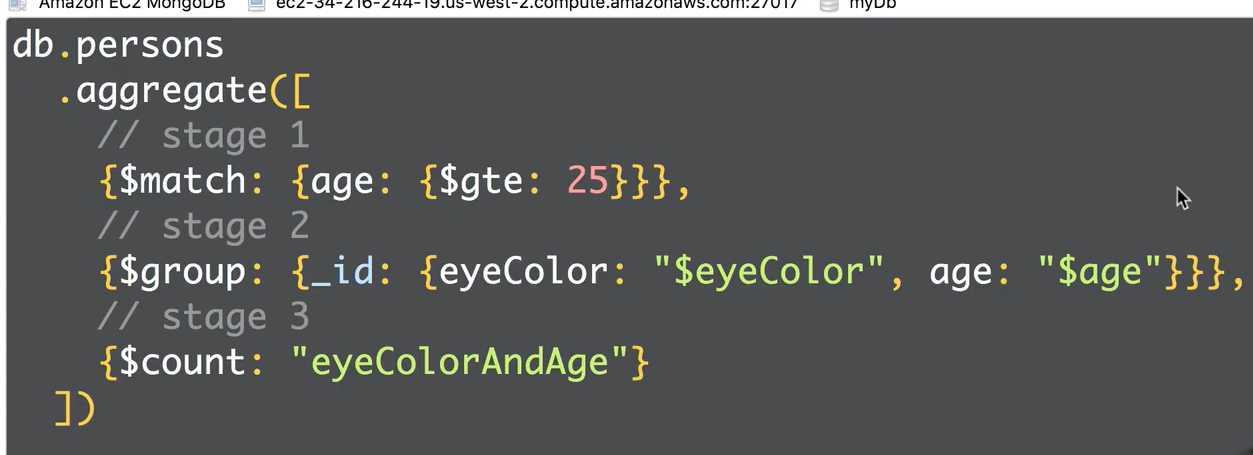
db.collection.aggregate( [ { <stage> }, ... ] )

note: aggregrate nhận **vào 1 array**. Trong array gồm các objects. Các objects(gồm <stage>: <field path>) được gọi là stages: { $group: “...”}.

db.collection.aggregate([

{ $group: {\_id: <expression>, <field>: {<accumulator>: <expression>}}},

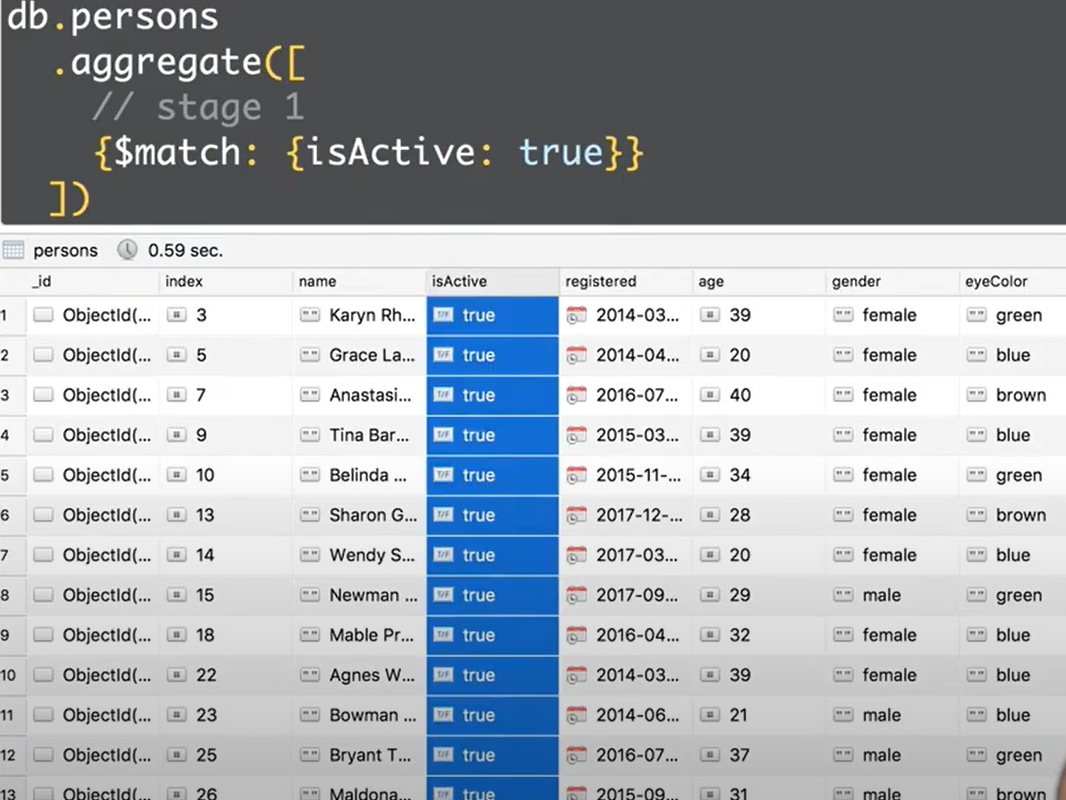
])



Các <stage>: $group, $merge, $unwind, $project, $match,$count,$limit,...

Các <accumulator> : $sum, $avg, $max,$min

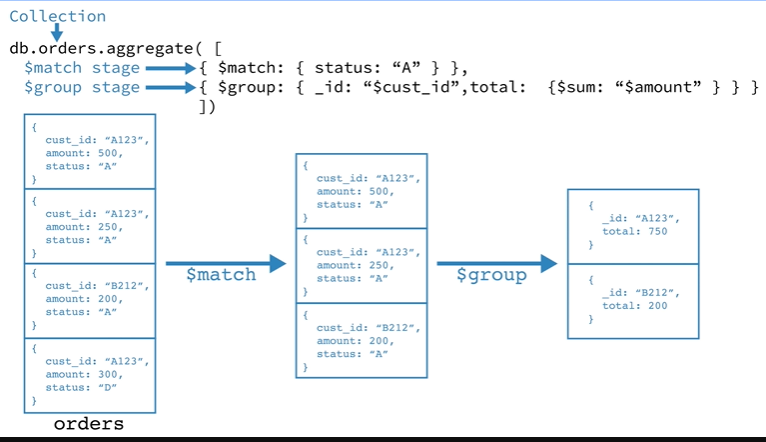
1. **$match:**

****

Nhận vào các fields theo như các fields trong objects. (giống như filter trong js)

1. **$group:**

luôn luôn có \_id. Đây là prop bắt buộc, để $group dựa vào **các fields trong \_id** để nhóm theo điều kiện



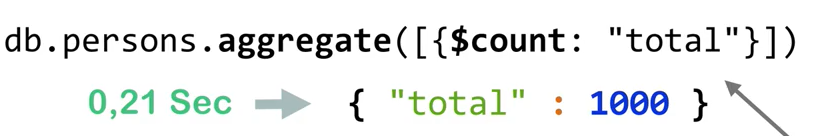
* Có thể group với multi fields. Tất cả fields đưa vào \_id:



\_id trên là object chứa 2 fields “eyeColor” và “favoriteFruit”.

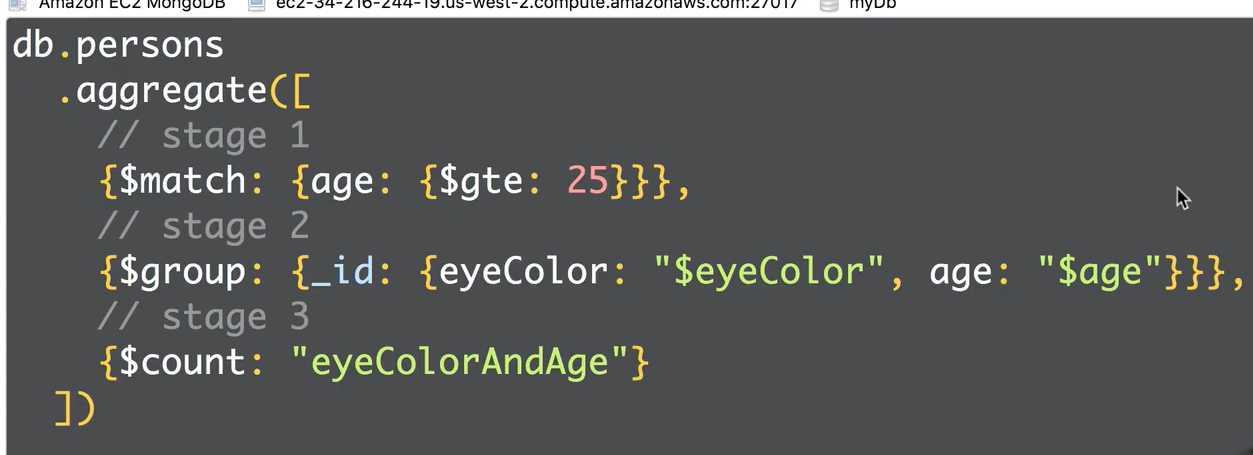
1. **$count**

Đếm số lượng documents. Giống Array.length js



$count: “fieldName”

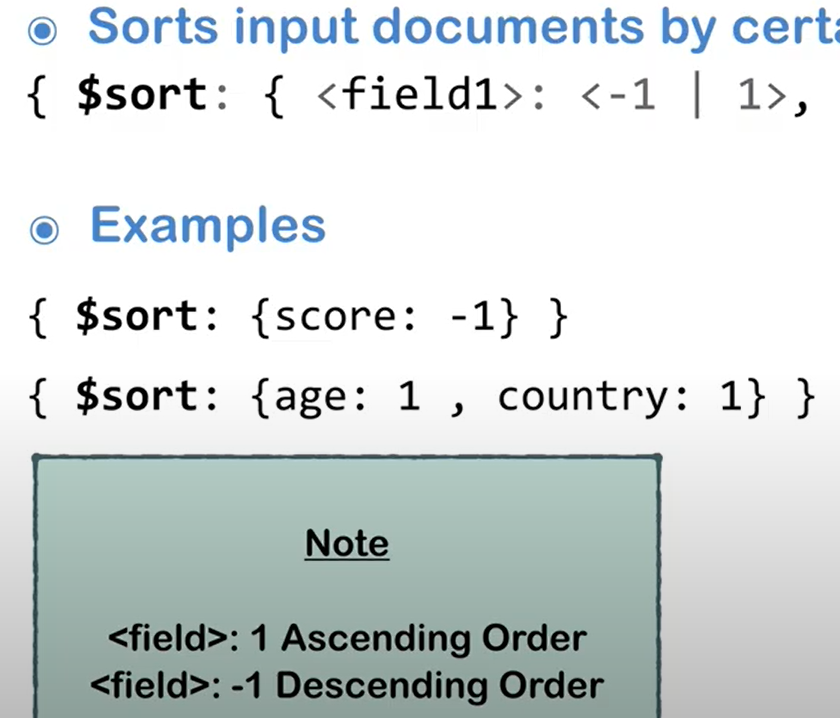
Có thể kết hợp với các Stage khác



1. **$sort**

Sắp xếp các documents

Thường đứng sau $match, $group



Giá trị các fields trong sort là 1 (tăng dần) or -1 (giảm dần)

1. **$project**



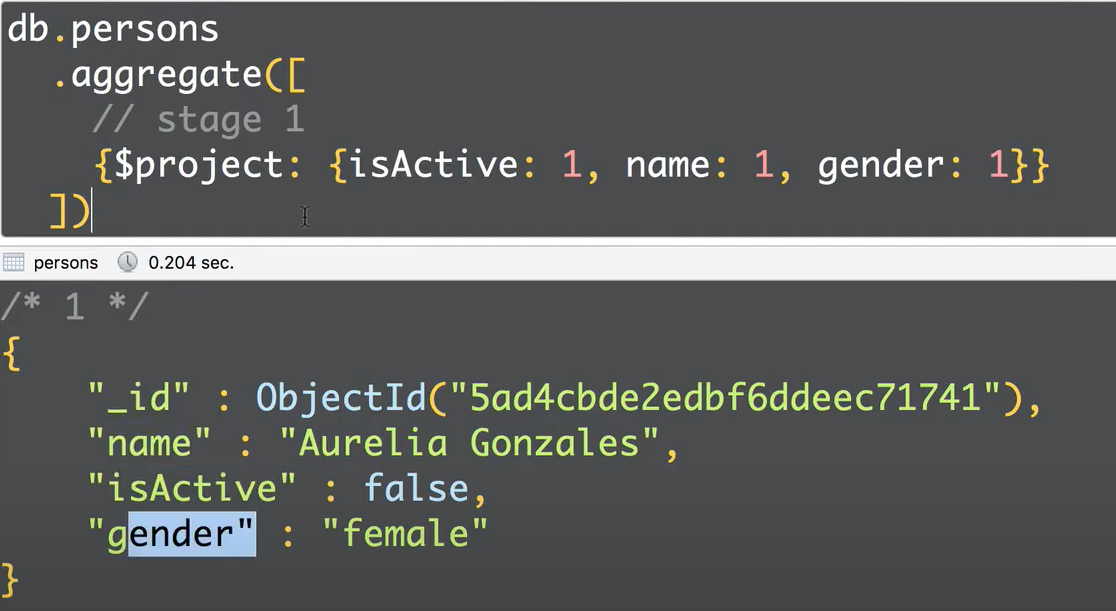
Value: <1 or true> => includes

Value: <0 or false> => excludes

<newField1>: <expression> remane field thành <newField1> (<expression>: “$field\_path”)

Mặc định \_id luôn có. Nhưng có thể exclude ra khi khai báo value: <0>

 nếu define các fields là 0 hết thì các fields còn lại là 1



Khi có field: <1> thì sẽ chỉ trả về các field có giá trị là 1.

Rename fields:

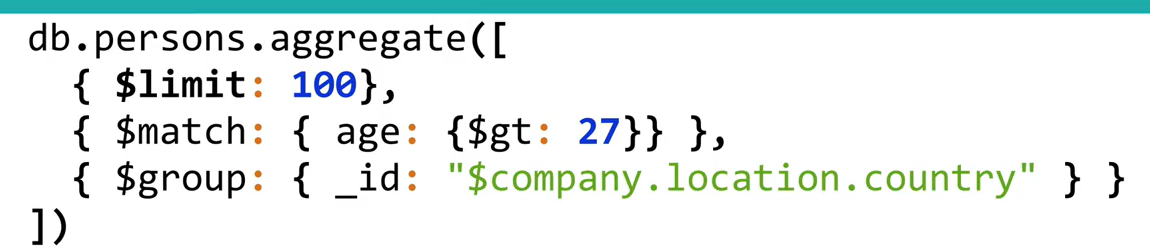
Dùng để reStructure documents cho dễ đọc nếu có các fields lồng nhau nhiều cấp



1. **$limit**

Dùng để phân trang

Cơ thể để cái stage này lên trước để giới hạn lại các documents



1. **$sum**

Tính tổng. Nếu {$sum: 1} thì sẽ tính tổng các records

Nếu {$sum: <expression>} thì sẽ tính tổng theo value của <expression> đó

Vd: {$sum: "$Amount"} thì sẽ tính tổng theo field Amount.

1. **$skip**

Được dùng với limit để phân trang

Như tên thì nó sẽ skip/bỏ đi các records dựa trên value của $skip ($skip: 21 bỏ 21 lấy record thử 22 trở đi)

Tip: skip bắt đầu từ 0 nên khi kết hợp $limit sẽ có expression: $skip: (page-1)\*pageSize

1. **Tip to update records using forEach**

db.getCollection**(**"GradeTenScoreModel"**)**.find**({**subjectId**:** "a616e8776465c3dad9"**})**.forEach**(**function**(**doc**)** **{**

db.getCollection**(**"GradeTenScoreModel"**)**.update**({**\_id**:** doc.\_id**},** **{**

$set**:** **{** score**:** **[**3**,**4**,**5**][**Math.floor**(**Math.random**()** \* 3**)]** **}**

**})**

**})**