```
PIPELINE OUTPUT

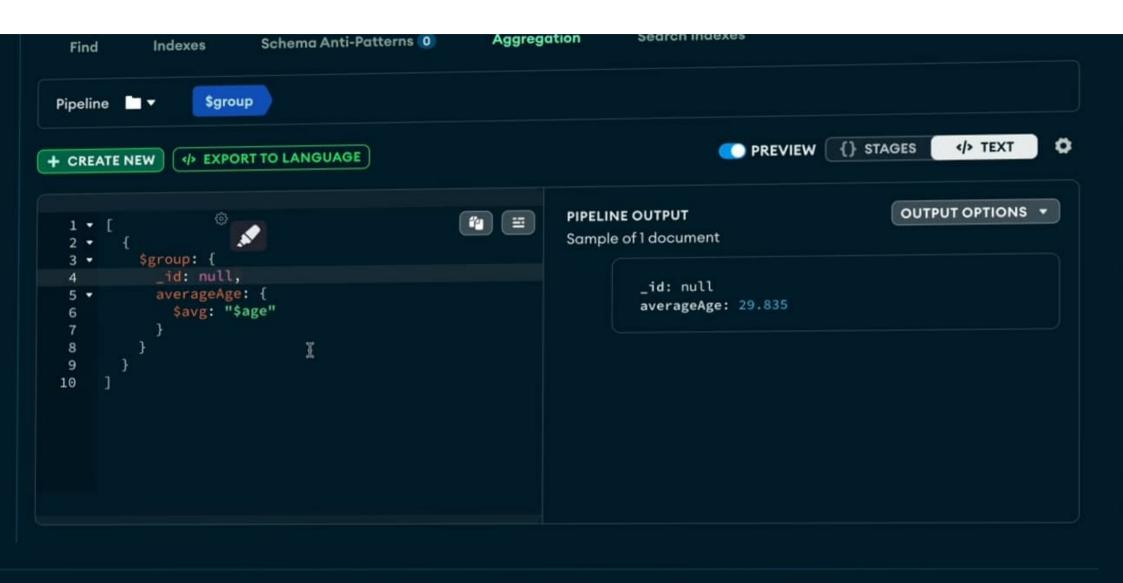
1 * [
2 * {
3 * Smatch: {
4     isActive: true,
5     },
6     },
7 * {
8     Scount: "activeUsers",
9     },
10 ]

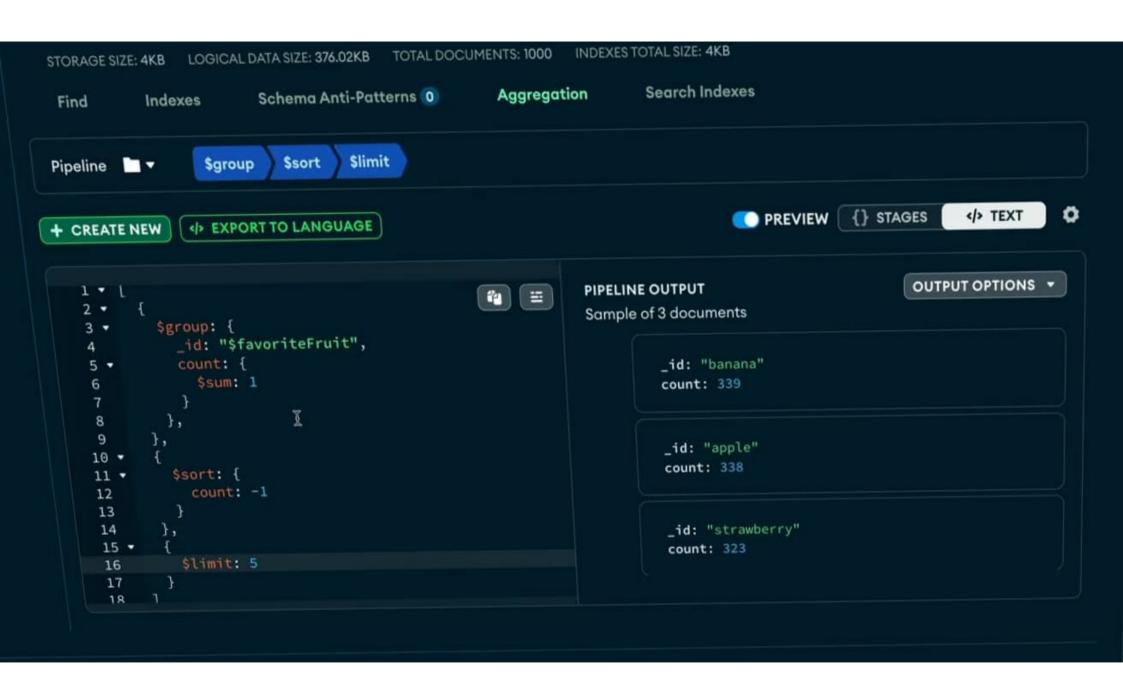
PIPELINE OUTPUT

Sample of 1 document

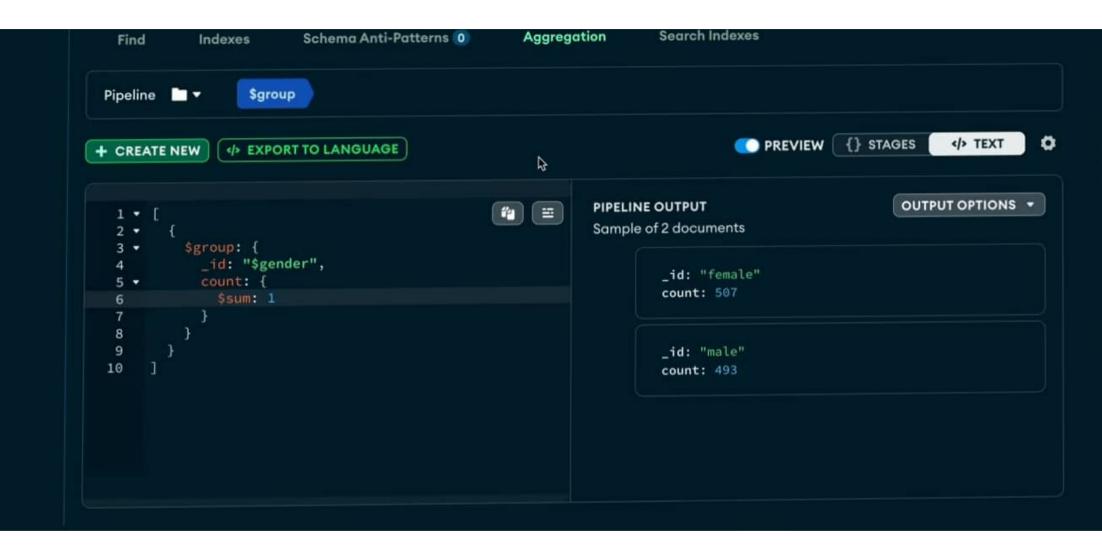
activeUsers: 516
```

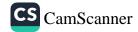




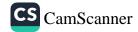










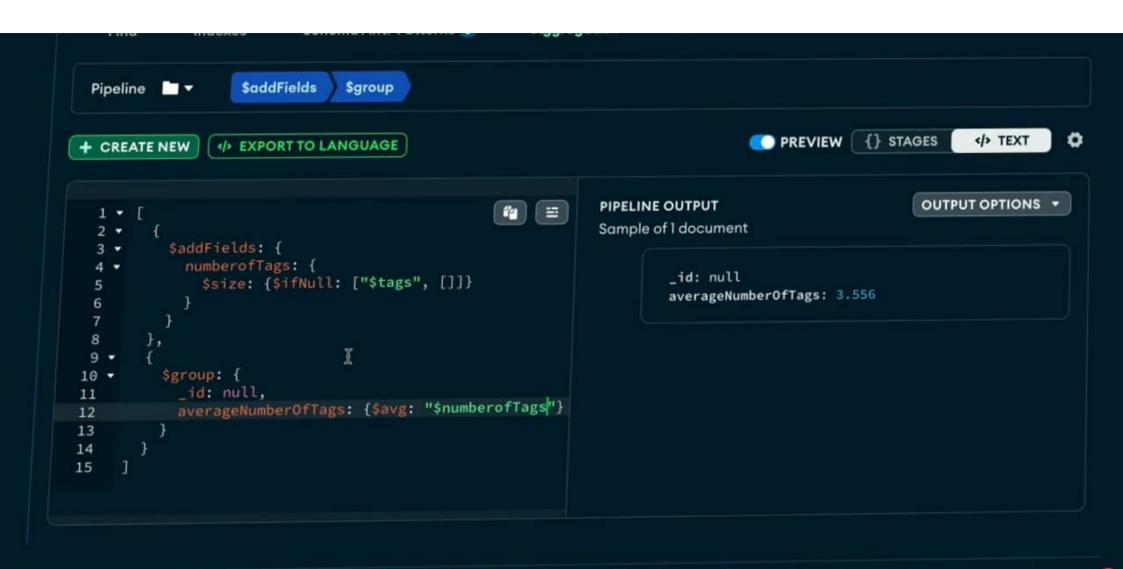




s Privacy Atlas Blog Contact Sales

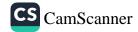


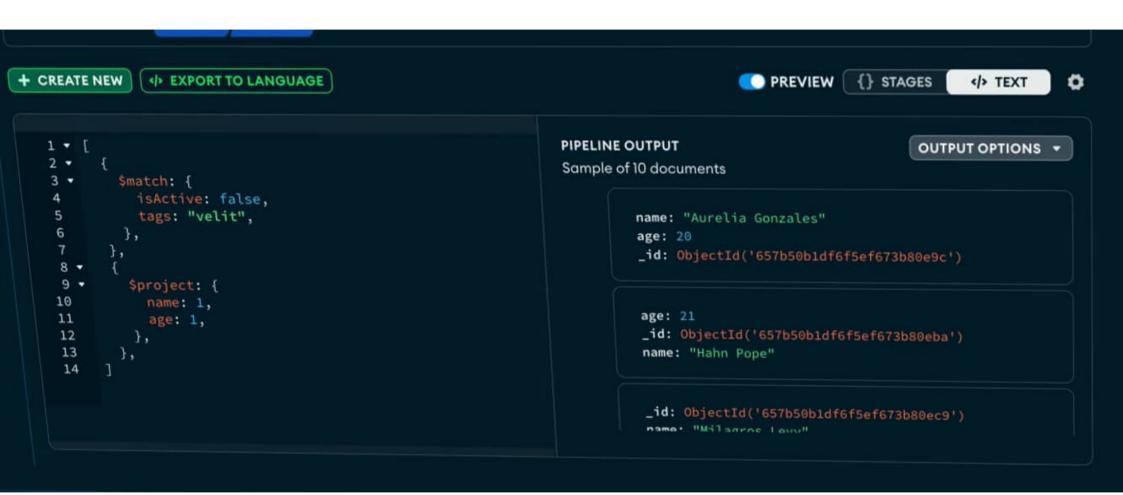




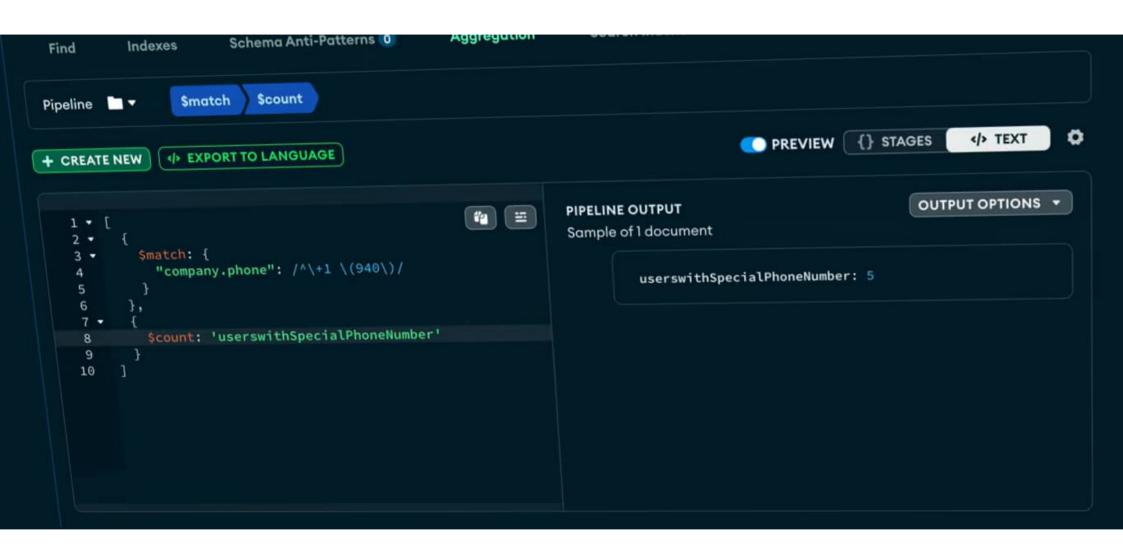






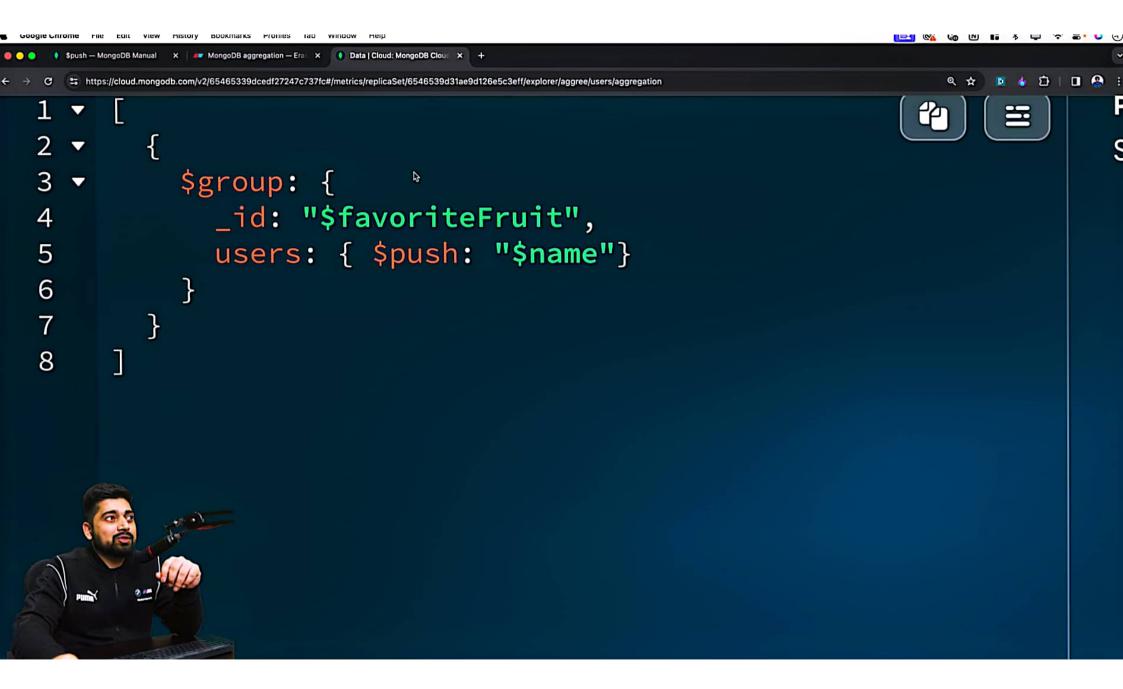


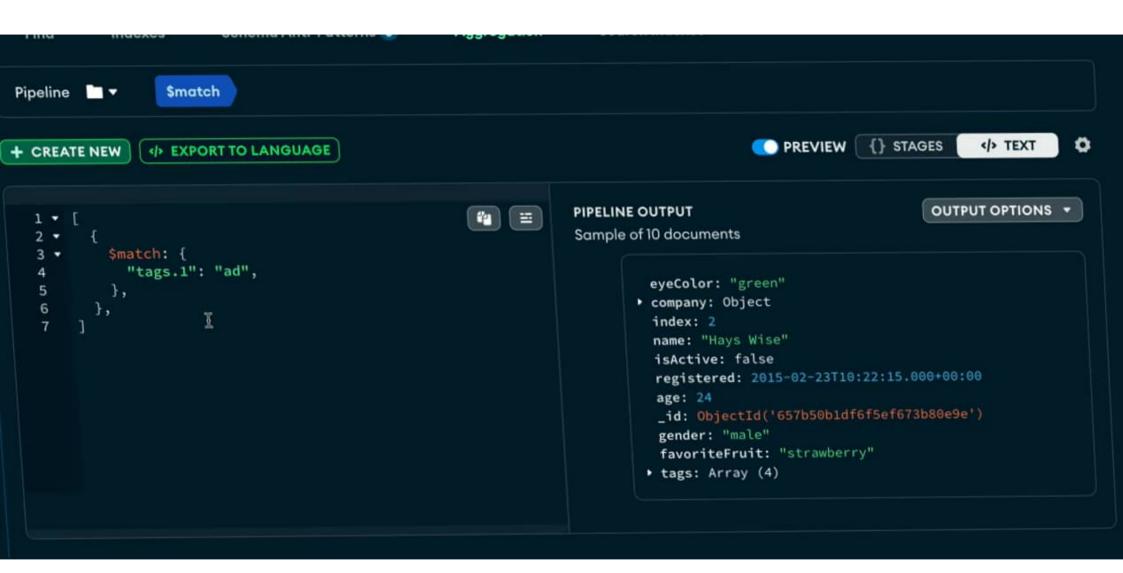




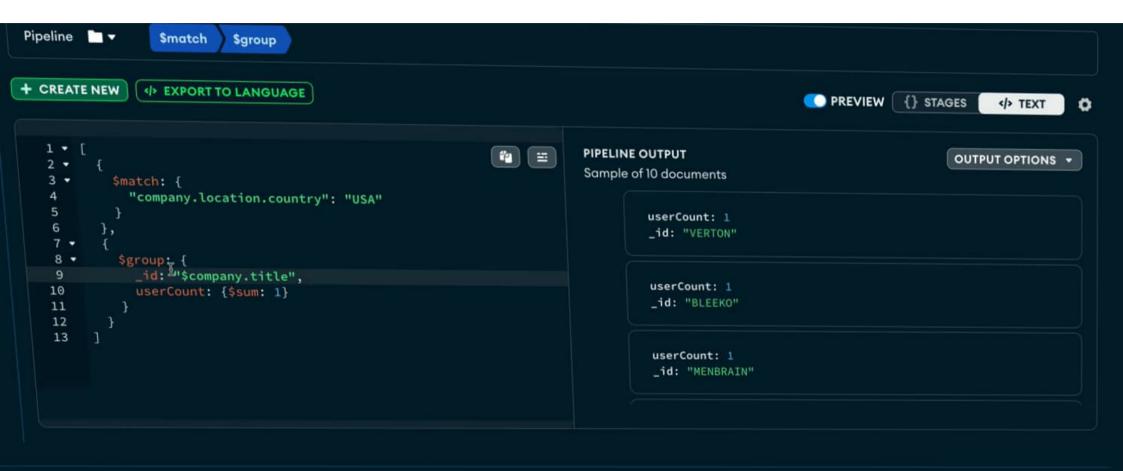
```
3
         $sort: {
           registered: -1
5
 6
       },
       {$limit: 4},
8 🕶
 9
         $project: {
10
            name: 1,
            registered: 1,
11
            favoriteFruit: 1
```







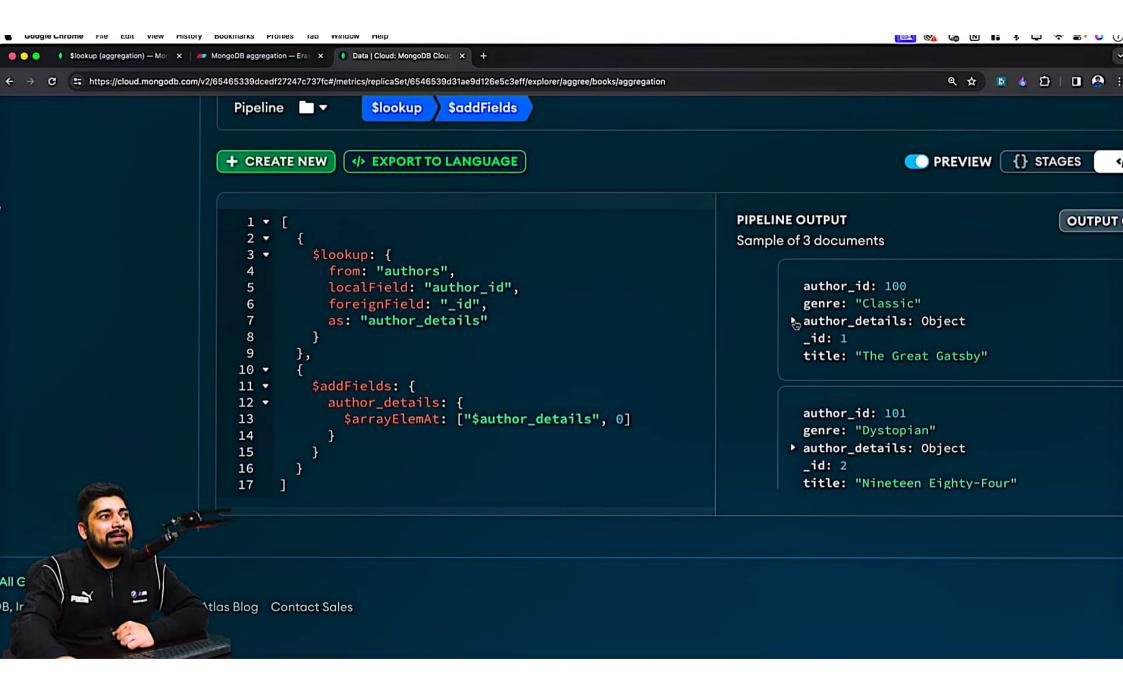














@tarunbidaliya9438 4 months ago

Some additional tips

- Always think TWICE when you are using operators like "\$unwind" because it is very costly operator because it creates multiple copies of a same documents. Try to think some other possible solution. Some costly operators are: "\$lookup", "\$unwind", "\$facet", etc.
- 2) It is a best practice to use \$project after the \$group stage in the pipeline because "by default, mongodb return a BSON object and not a JSON object" and BSON object have more properties than a JSON object. So using the \$project, you can get the JSON object You can study this point for more knowledge.
- 3) If the size of your document is big, then always use ".allowDiscUse(true)" in your aggregation pipeline.
- 4) When dealing with larger chunks of data, use "cursor" and "batchSize" approach.

Show less

占 20 **分 Reply** 

a --- line

