

# Aggregation

- To actually run this in MongoDB, pass each operation to the `aggregate()` function:

```
db.books.aggregate(  
  {$project: {"authors": 1}},  
  {$unwind: "$authors"},  
  {$group: {"_id": "$authors", "count": {$sum: 1}}},  
  {$sort: {"count": -1}},  
  {$limit: 5}  
)
```

```
> db.books.aggregate(  
...   {$project: {"authors": 1}},  
...   {$unwind: "$authors"},  
...   {$group: {"_id": "$authors", "count": {$sum: 1}}},  
...   {$sort: {"count": -1}},  
...   {$limit: 5}  
... )  
{ "_id" : "", "count" : 59 }  
{ "_id" : "Vikram Goyal", "count" : 12 }  
{ "_id" : "Richard Siddaway", "count" : 6 }  
{ "_id" : "Don Jones", "count" : 6 }  
{ "_id" : "Christian Bauer", "count" : 5 }
```

## \$match

- ▶ \$match filters documents so that the aggregation runs on a subset of documents
- ▶ For example, some of the books contain empty strings in the authors array, e.g.

```
{ "_id" : 175, "authors" : [ "Dierk König", "Guillaume Laforge", "Paul King", "Cédric Champeau", "Hamlet D'Arcy", "Erik Pragt", "", "Jon Skeet" ] }
```

- ▶ To disregard these authors, we can add a \$match expression after \$unwind:

```
db.books.aggregate(  
  {$project: {"authors": 1}},  
  {$unwind: "$authors"},  
  {$match: {"authors": {$ne: ""}}},  
  {$group: {"_id": "$authors", "count": {$sum: 1}}},  
  {$sort: {"count": -1}},  
  {$limit: 5}  
)
```

```
> db.books.aggregate(  
...   {$project: {"authors": 1}},  
...   {$unwind: "$authors"},  
...   {$match: {"authors": {$ne: ""}}},  
...   {$group: {"_id": "$authors", "count": {$sum: 1}}},  
...   {$sort: {"count": -1}},  
...   {$limit: 5}  
... )  
{ "_id" : "Vikram Goyal", "count" : 12 }  
{ "_id" : "Richard Siddaway", "count" : 6 }  
{ "_id" : "Don Jones", "count" : 6 }  
{ "_id" : "Christian Bauer", "count" : 5 }  
{ "_id" : "Gavin King", "count" : 5 }
```

- ▶ \$match can use all of the usual query operators \$gt, \$lt, \$in, etc.
- ▶ A good practice is to put "\$match" expressions as early as possible in the pipeline

## Array Aggregation Operators

- For example, the following display the author's first name of each book:

```
db.books.aggregate(  
  {$project: {"authors": 1}},  
  {$unwind: "$authors"},  
  {$project: {"authors": {$split: ["$authors", " "]}},  
  {$project: {"firstName": {$arrayElemAt: ["$authors", 1]}},  
  {$limit: 5}  
)
```

```
> db.books.aggregate(  
...   {$project: {"authors": 1}},  
...   {$unwind: "$authors"},  
...   {$project: {"authors": {$split: ["$authors", " "]}},  
...   {$project: {"firstName": {$arrayElemAt: ["$authors", 1]}},  
...   {$limit: 5}  
... )  
{ "_id" : 4, "firstName" : "Ahmed" }  
{ "_id" : 4, "firstName" : "Abid" }  
{ "_id" : 2, "firstName" : "Frank" }  
{ "_id" : 2, "firstName" : "Sen" }  
{ "_id" : 6, "firstName" : "Alag" }
```