NIELIT Gorakhpur

Course name: A level **SUBJECT: DATABASE TECHNOLOGIES**

Topic: MongoDB **DATE: 21/5/2020**

Aggregation framework cont'd

$skip stage operator

$skip stage operator skips the specified number of documents that pass into the stage and passes the remaining documents to the next stage in the pipeline. $skip has no effect on the content of the documents it passes along the pipeline.

Syntax

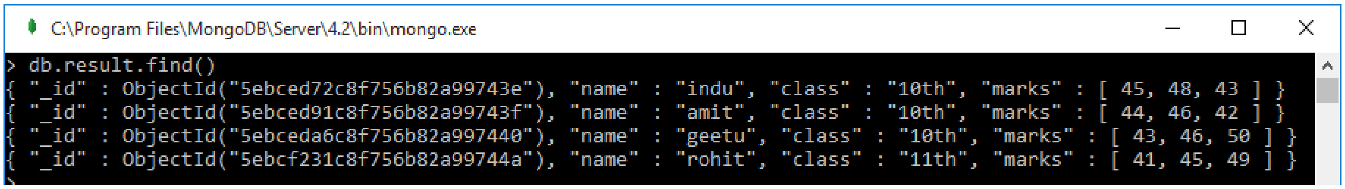
{ $skip: <positive integer> }

Where,

positive integer specifies the maximum number of documents to skip to be passed with $skip stage operator.

Example

Consider the result collection having 4 documents:

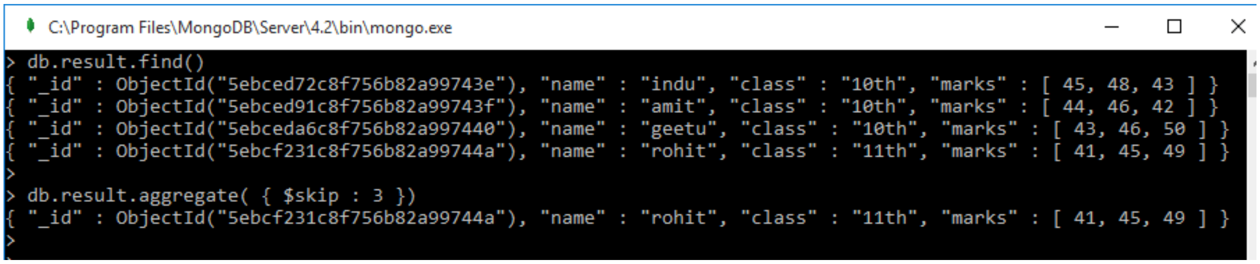


```
> db.result.find()
{ "_id" : ObjectId("5ebced72c8f756b82a99743e"), "name" : "indu", "class" : "10th", "marks" : [ 45, 48, 43 ] }
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [ 44, 46, 42 ] }
{ "_id" : ObjectId("5ebceda6c8f756b82a997440"), "name" : "geetu", "class" : "10th", "marks" : [ 43, 46, 50 ] }
{ "_id" : ObjectId("5ebcf231c8f756b82a99744a"), "name" : "rohit", "class" : "11th", "marks" : [ 41, 45, 49 ] }
```

- If we want to skip first 3 records of result collection, then the command will be

db.result.aggregate({ $skip : 3 })

This operation skips the first 3 documents passed to it by the pipeline :

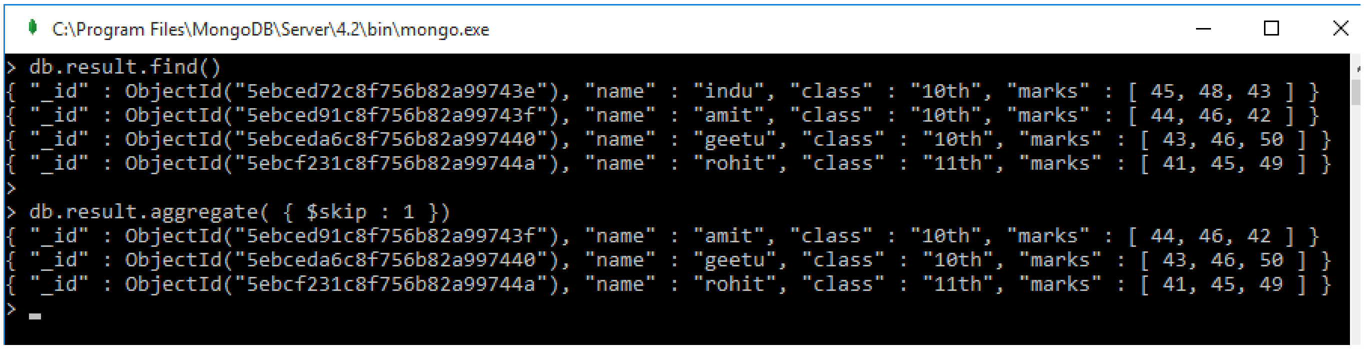


```
> db.result.find()
{ "_id" : ObjectId("5ebced72c8f756b82a99743e"), "name" : "indu", "class" : "10th", "marks" : [ 45, 48, 43 ] }
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [ 44, 46, 42 ] }
{ "_id" : ObjectId("5ebceda6c8f756b82a997440"), "name" : "geetu", "class" : "10th", "marks" : [ 43, 46, 50 ] }
{ "_id" : ObjectId("5ebcf231c8f756b82a99744a"), "name" : "rohit", "class" : "11th", "marks" : [ 41, 45, 49 ] }
>
> db.result.aggregate( { $skip : 3 })
{ "_id" : ObjectId("5ebcf231c8f756b82a99744a"), "name" : "rohit", "class" : "11th", "marks" : [ 41, 45, 49 ] }
>
```

- If we want to skip only the first records of result collection, then the command will be

db.result.aggregate({ $skip : 1 })

This operation skips the first documents passed to it by the pipeline :



```
> db.result.find()
{ "_id" : ObjectId("5ebced72c8f756b82a99743e"), "name" : "indu", "class" : "10th", "marks" : [ 45, 48, 43 ] }
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [ 44, 46, 42 ] }
{ "_id" : ObjectId("5ebceda6c8f756b82a997440"), "name" : "geetu", "class" : "10th", "marks" : [ 43, 46, 50 ] }
{ "_id" : ObjectId("5ebcf231c8f756b82a99744a"), "name" : "rohit", "class" : "11th", "marks" : [ 41, 45, 49 ] }
>
> db.result.aggregate( { $skip : 1 })
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [ 44, 46, 42 ] }
{ "_id" : ObjectId("5ebceda6c8f756b82a997440"), "name" : "geetu", "class" : "10th", "marks" : [ 43, 46, 50 ] }
{ "_id" : ObjectId("5ebcf231c8f756b82a99744a"), "name" : "rohit", "class" : "11th", "marks" : [ 41, 45, 49 ] }
> _
```

$limit stage operator

$limit stage operator returns the first N documents, where N is the specified limit and these output documents may be passed to the next stage in the pipeline.

Syntax

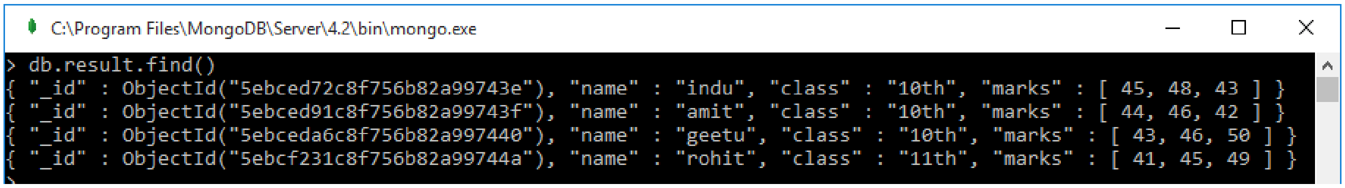
{ $limit: <positive integer> }

Where,

positive integer specifies the maximum number of documents to be passed, i.e. no of documents for next stage in the pipeline.

Example

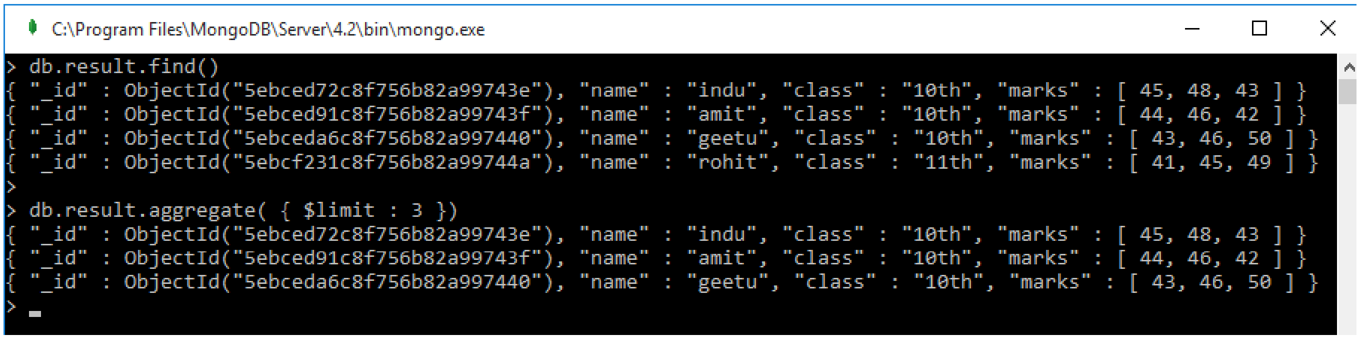
Consider the result collection having 4 documents:



```
> db.result.find()
{ "_id" : ObjectId("5ebced72c8f756b82a99743e"), "name" : "indu", "class" : "10th", "marks" : [ 45, 48, 43 ] }
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [ 44, 46, 42 ] }
{ "_id" : ObjectId("5ebceda6c8f756b82a997440"), "name" : "geetu", "class" : "10th", "marks" : [ 43, 46, 50 ] }
{ "_id" : ObjectId("5ebcf231c8f756b82a99744a"), "name" : "rohit", "class" : "11th", "marks" : [ 41, 45, 49 ] }
>
```

- If we want to pass only first 3 records of the result collection, then the command will be

db.result.aggregate({ $limit : 3 })

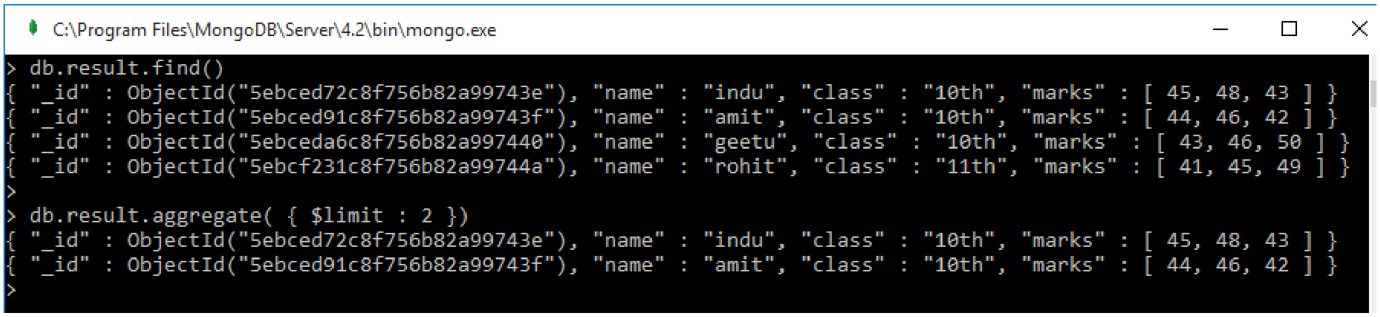


```
> db.result.find()
{ "_id" : ObjectId("5ebced72c8f756b82a99743e"), "name" : "indu", "class" : "10th", "marks" : [ 45, 48, 43 ] }
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [ 44, 46, 42 ] }
{ "_id" : ObjectId("5ebceda6c8f756b82a997440"), "name" : "geetu", "class" : "10th", "marks" : [ 43, 46, 50 ] }
{ "_id" : ObjectId("5ebcf231c8f756b82a99744a"), "name" : "rohit", "class" : "11th", "marks" : [ 41, 45, 49 ] }
>
> db.result.aggregate( { $limit : 3 })
{ "_id" : ObjectId("5ebced72c8f756b82a99743e"), "name" : "indu", "class" : "10th", "marks" : [ 45, 48, 43 ] }
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [ 44, 46, 42 ] }
{ "_id" : ObjectId("5ebceda6c8f756b82a997440"), "name" : "geetu", "class" : "10th", "marks" : [ 43, 46, 50 ] }
> _
```

The above command will pass only top 3 documents.

- If we want to pass only first 2 records of the result collection, then the command will be

db.result.aggregate({ $limit : 2 })

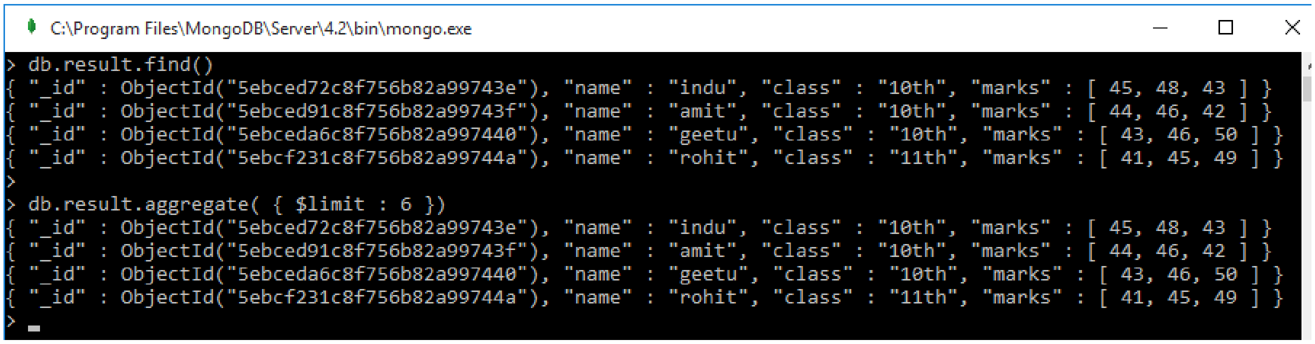


```
> db.result.find()
{ "_id" : ObjectId("5ebced72c8f756b82a99743e"), "name" : "indu", "class" : "10th", "marks" : [ 45, 48, 43 ] }
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [ 44, 46, 42 ] }
{ "_id" : ObjectId("5ebceda6c8f756b82a997440"), "name" : "geetu", "class" : "10th", "marks" : [ 43, 46, 50 ] }
{ "_id" : ObjectId("5ebcf231c8f756b82a99744a"), "name" : "rohit", "class" : "11th", "marks" : [ 41, 45, 49 ] }
>
> db.result.aggregate( { $limit : 2 })
{ "_id" : ObjectId("5ebced72c8f756b82a99743e"), "name" : "indu", "class" : "10th", "marks" : [ 45, 48, 43 ] }
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [ 44, 46, 42 ] }
>
```

- If we want to pass only first 6 documents of the result collection, then the command will be

db.result.aggregate({ $limit : 6 })

but if the collection is not having the specified documents, then all the documents available in the collection will be passed, no Error will be displayed. Here there are 4 documents which are passed:



```
> db.result.find()
{ "_id" : ObjectId("5ebced72c8f756b82a99743e"), "name" : "indu", "class" : "10th", "marks" : [ 45, 48, 43 ] }
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [ 44, 46, 42 ] }
{ "_id" : ObjectId("5ebceda6c8f756b82a997440"), "name" : "geetu", "class" : "10th", "marks" : [ 43, 46, 50 ] }
{ "_id" : ObjectId("5ebcf231c8f756b82a99744a"), "name" : "rohit", "class" : "11th", "marks" : [ 41, 45, 49 ] }
>
> db.result.aggregate( { $limit : 6 })
{ "_id" : ObjectId("5ebced72c8f756b82a99743e"), "name" : "indu", "class" : "10th", "marks" : [ 45, 48, 43 ] }
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [ 44, 46, 42 ] }
{ "_id" : ObjectId("5ebceda6c8f756b82a997440"), "name" : "geetu", "class" : "10th", "marks" : [ 43, 46, 50 ] }
{ "_id" : ObjectId("5ebcf231c8f756b82a99744a"), "name" : "rohit", "class" : "11th", "marks" : [ 41, 45, 49 ] }
> _
```

Assignments

1. What do you mean by Skip stage operator? Explain.
2. How limit stage operator is used in aggregation?