

## Aggregation Framework cont'd

### Project Function

In MongoDB Aggregation, \$project function is passed along the documents with only the specified fields to the next stage in the pipeline. This field can be either the existing fields from the input documents or newly computed fields.

#### Syntax

```
{ $project: { <specifications> } }
```

Where, the specification contain

- the inclusion of fields,
- the suppression of the \_id field,
- the addition of new fields, and
- the resetting the values of existing fields.

#### Parameters:

Specification	Description
<field>: <1 or true>	Specify the inclusion of a field.
_id: <0 or false>	Specify the suppression of the _id field. By default _id field is included in the output document.
<field>: <expression>	Add a new field or reset the value of an existing field.

#### Further,

- A field that does not exist in the document are going to include, the \$project ignores that field inclusion;
- A new field can be added and value of an existing field can be reset by specifying the field name and set its value to some expression.

**For Example 1 & 2 below,** Lets have a school collection as follows

```
> db.school.find()
{ "_id" : ObjectId("5ec233f75ddaf5609c6cb180"), "stu_id" : "A01", "stu_name" : "vinod", "class" : "2nd", "marks" : 60 }
{ "_id" : ObjectId("5ec233f75ddaf5609c6cb181"), "stu_id" : "A02", "stu_name" : "amit", "class" : "2nd", "marks" : 55 }
{ "_id" : ObjectId("5ec233f75ddaf5609c6cb182"), "stu_name" : "arun" }
{ "_id" : ObjectId("5ec236255ddaf5609c6cb183"), "stu_id" : "A04", "stu_name" : "bipin", "class" : "2nd", "marks" : 60 }
{ "_id" : ObjectId("5ec236255ddaf5609c6cb184"), "stu_id" : "A05", "stu_name" : "anil", "class" : "3rd", "marks" : 55 }
{ "_id" : ObjectId("5ec236255ddaf5609c6cb185"), "stu_id" : "A06", "stu_name" : "ajay", "class" : "3rd", "marks" : 70 }
```

**Example-1:** To include some Specific Fields in the Output Documents

```
db.school.aggregate([ { $project : { Stu_id: 1 , stu_name : 1 } } ] )
```

```
> db.school.aggregate([{$project:{stu_id:1,stu_name:1}}])
{ "_id" : ObjectId("5ec233f75ddaf5609c6cb180"), "stu_id" : "A01", "stu_name" : "vinod" }
{ "_id" : ObjectId("5ec233f75ddaf5609c6cb181"), "stu_id" : "A02", "stu_name" : "amit" }
{ "_id" : ObjectId("5ec233f75ddaf5609c6cb182") }
{ "_id" : ObjectId("5ec236255ddaf5609c6cb183"), "stu_id" : "A04", "stu_name" : "bipin" }
{ "_id" : ObjectId("5ec236255ddaf5609c6cb184"), "stu_id" : "A05", "stu_name" : "anil" }
{ "_id" : ObjectId("5ec236255ddaf5609c6cb185"), "stu_id" : "A06", "stu_name" : "ajay" }
```

The command includes only the `_id`, `std_id`, and `stu_name` fields in its output documents as shown above. `_id` field is included by default.

**Example-2 :** To Suppress `_id` Field in the Output Documents

```
db.school.aggregate([ { $project : { _id: 0, stu_id : 1 ,stu_name : 1 } } ] )
```

```
> db.school.aggregate([{$project:{_id:0,stu_id:1,stu_name:1}}])
{ "stu_id" : "A01", "stu_name" : "vinod" }
{ "stu_id" : "A02", "stu_name" : "amit" }
{ }
{ "stu_id" : "A04", "stu_name" : "bipin" }
{ "stu_id" : "A05", "stu_name" : "anil" }
{ "stu_id" : "A06", "stu_name" : "ajay" }
```

The above command suppresses i.e. excludes the `_id` field but includes the `std_id`, and `stu_name` fields in its output documents as we have given value as 0 for `_id` field.

**Example:** To Include Specific Fields from Embedded Documents

Lets 1<sup>st</sup> consider the School collection, the same has been updated and added an embedded document fees having tution, admission and books keys. Now the collection looks like:

```
> db.school.find()
{ "_id" : ObjectId("5ec233f75ddaf5609c6cb180"), "stu_name" : "vinod", "class" : "2nd", "marks" : 60, "fees" : { "tution" : 6000, "admission" : 600, "books" : 700 } }
{ "_id" : ObjectId("5ec233f75ddaf5609c6cb181"), "stu_name" : "amit", "class" : "2nd", "marks" : 55, "fees" : { "tution" : 4000, "admission" : 300, "books" : 300 } }
{ "_id" : ObjectId("5ec233f75ddaf5609c6cb182"), "Stu_name" : "arun" }
{ "_id" : ObjectId("5ec236255ddaf5609c6cb183"), "stu_name" : "bipin", "class" : "2nd", "marks" : 45, "fees" : { "tution" : 2000, "admission" : 500, "books" : 700 } }
{ "_id" : ObjectId("5ec236255ddaf5609c6cb184"), "stu_name" : "anil", "class" : "3rd", "marks" : 50, "fees" : { "tution" : 5000, "admission" : 300, "books" : 900 } }
{ "_id" : ObjectId("5ec236255ddaf5609c6cb185"), "stu_name" : "ajay", "class" : "3rd", "marks" : 50, "fees" : { "tution" : 4500, "admission" : 600, "books" : 200 } }
```

We wish to include only tuition fee component from the embedded document. We may use dot (`.`) notation for taking only the tution from fee `{fee.tution:1}` or we may write it like `{fee:{tution:1}}`.

The command for this will be:

```
db.school.aggregate([ { $project: { "fee.tuition": 1 } } ])
```

or we may also write it as

```
db.schools.aggregate([ { $project: { fees: { tuition: 1 } } } ] );
```

Once executed, it will show the output as

```
> db.school.aggregate([{$project:{"fees.tuition":1}}])
{ "_id" : ObjectId("5ec233f75ddaf5609c6cb180"), "fees" : { "tuition" : 6000 } }
{ "_id" : ObjectId("5ec233f75ddaf5609c6cb181"), "fees" : { "tuition" : 4000 } }
{ "_id" : ObjectId("5ec233f75ddaf5609c6cb182") }
{ "_id" : ObjectId("5ec236255ddaf5609c6cb183"), "fees" : { "tuition" : 2000 } }
{ "_id" : ObjectId("5ec236255ddaf5609c6cb184"), "fees" : { "tuition" : 5000 } }
{ "_id" : ObjectId("5ec236255ddaf5609c6cb185"), "fees" : { "tuition" : 4500 } }
>
```

### Assignments:

1. What is the use of Project function? Explain.
2. How we can suppress `_id` field in output document? Give example.
3. How fields from embedded documents are added in output document?