

AGGREGATION OPERATORS

Execute Aggregation operations (\$avg, \$min, \$max, \$push, \$addToSet etc.). students encourage to execute several queries to demonstrate various aggregation operators).

Aggregation means grouping together

For example sum, avg, min, max.

Syntax

Db.collection.aggregate(<AGGREGATE OPERATION>

Types

Expression Type	Description	Syntax
Accumulators	Perform calculations on entire groups of documents	
* \$sum	Calculates the sum of all values in a numeric field within a group.	"\$fieldName": { \$sum: "\$fieldName" }
* \$avg	Calculates the average of all values in a numeric field within a group.	"\$fieldName": { \$avg: "\$fieldName" }
* \$min	Finds the minimum value in a field within a group.	"\$fieldName": { \$min: "\$fieldName" }
* \$max	Finds the maximum value in a field within a group.	"\$fieldName": { \$max: "\$fieldName" }
* \$push	Creates an array containing all unique or duplicate values from a field	"\$arrayName": { \$push: "\$fieldName" }
* \$addToSet	Creates an array containing only unique values from a field within a group.	"\$arrayName": { \$addToSet: "\$fieldName" }
* \$first	Returns the first value in a field within a group (or entire collection).	"\$fieldName": { \$first: "\$fieldName" }
* \$last	Returns the last value in a field within a group (or entire collection).	"\$fieldName": { \$last: "\$fieldName" }

Average GPA of All Students:

```
db.students.aggregate([
  { $group: { _id: null, overageGPA: "$gpa" } }
]);
```

Output:

```
[ { _id: null, averageGPA: 2.98556 } ]
db> |
```

\$group: Groups all documents together

_id: null: Sets the group identifier to null (optional as there's only one group in this case).

averageGPA: Calculates the average value of the "gpa" field using the \$avg operator.

Minimum and Maximum Age:

```
db.students.aggregate([
  { $group: { _id: null, minAge: { $min: "$age" }, maxAge: { $max: "$age" } } }
]);
```

Output:

```
[ { _id: null, minAge: 18, maxAge: 25 } ]
```

Similar to the previous example, it uses \$group to group all documents.

minAge: Uses the \$min operator to find the minimum value in the "age" field.

maxAge: Uses the \$max operator to find the minimum value in the "age" field.

Average GPA for all home cities?

```
db> db.students.aggregate([
...   { $group: { _id: "$home_city", averageGPA: { $avg: "$gpa" } } }
... ]);
[
  { _id: 'City 8', averageGPA: 3.11741935483871 },
  { _id: 'City 7', averageGPA: 2.847931034482759 },
  { _id: 'City 10', averageGPA: 2.935227272727273 },
  { _id: 'City 9', averageGPA: 3.1174358974358976 },
  { _id: 'City 2', averageGPA: 3.01969696969697 },
  { _id: 'City 3', averageGPA: 3.0100000000000002 },
  { _id: 'City 6', averageGPA: 2.8969444444444448 },
  { _id: null, averageGPA: 2.9784313725490197 },
  { _id: 'City 4', averageGPA: 2.8251851851851852 },
  { _id: 'City 1', averageGPA: 3.003823529411765 },
  { _id: 'City 5', averageGPA: 3.0607499999999996 }
]
```

Pushing All Courses into a Single Array:

```
db.students.aggregate([
  { $project: { _id: 0, allCourses: { $push: "$courses" } } }
]);
```

\$project: Transforms the input documents

_id:0: Excludes the _id field from the output documents.

allCourses: Uses the \$push operator to create an array. It pushes all elements from the "courses" field of each student document into the allCourses array.

Result:

This will return a list of documents, each with an allCourses array containing all unique courses offered (assuming courses might be duplicated across students).

BUT:

```
db> db.students.aggregate([
...   { $project: { _id: 0, allCourses: { $push: "$courses" } } }
... ]);
MongoServerError[Location31325]: Invalid $project :: caused by :: Unknown expression $push
db> |
```

This is because our Array is incorrect 😞

Collect Unique Courses Offered (Using \$addToSet):

```
db.candidates.aggregate([
  { $unwind: "$courses" },
  { $group: { _id: null, uniqueCourses: { $addToSet: "$courses" } } }
]);
```

What does it do?

```
db> db.candidates.aggregate([
...   { $unwind: "$courses" }, // Deconstruct courses array
...   { $group: { _id: null, uniqueCourses: { $addToSet: "$courses" } } }
que courses
... ]);
[
  {
    _id: null,
    uniqueCourses: [
      'Sociology',
      'Literature',
      'Ecology',
      'Physics',
      'Mathematics',
      'Marine Science',
      'Artificial Intelligence',
      'Art History',
      'Creative Writing',
      'Robotics',
      'Environmental Science',
      'Biology',
      'Statistics',
      'Music History',
      'Philosophy',
      'Film Studies',
      'Engineering',
      'Computer Science',
      'English',
      'Psychology',
      'Chemistry',
      'Political Science',
```