Experiment:5

Aggregation Operators

Aggregation operators are functions that process data and return a single result. They are used within the aggregation pipeline stages in MongoDB to transform and manipulate data.

- 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" }

Types of Aggregation Operators

MongoDB provides a rich set of aggregation operators categorized into different groups:

Accumulator Operators
Used within the `$group` stage to calculate accumulated values.

- `$avg`, `$sum`, `$min`, `$max`, `$first`, `$last`, `$push`, `$addToSet`, `$sum`, `$avg`, `$stdDevPop`, `$stdDevSamp`

Arithmetic Operators

- Perform mathematical calculations on numeric values.
  - `$add`, `$subtract`, `$multiply`, `$divide`, `$mod`, `$abs`, `$ceil`, `$floor`, `$round`, `$sqrt`, `$pow`

Comparison Operators

- Compare values and return boolean results.
  - `$eq`, `$gt`, `$gte`, `$lt`, `$lte`, `$ne`, `$cmp`

Conditional Operators

- Perform conditional logic.
  - `$cond`, `$ifNull`, `$switch`, `$case`

String Operators

- Manipulate strings.
  - `$concat`, `$substr`, `$toLower`, `$toUpper`, `$trim`, `$split`, `$indexOfCP`, `$indexOfBytes`, `$strLenCP`, `$strLenBytes`

Array Operators

- Process and manipulate arrays.
  - `$arrayElemAt`, `$concatArrays`, `$filter`, `$isArray`, `$map`, `$push`, `$pop`, `$pull`, `$addToSet`, `$size`, `$slice`, `$reverseArray`, `$reduce`

Object Operators

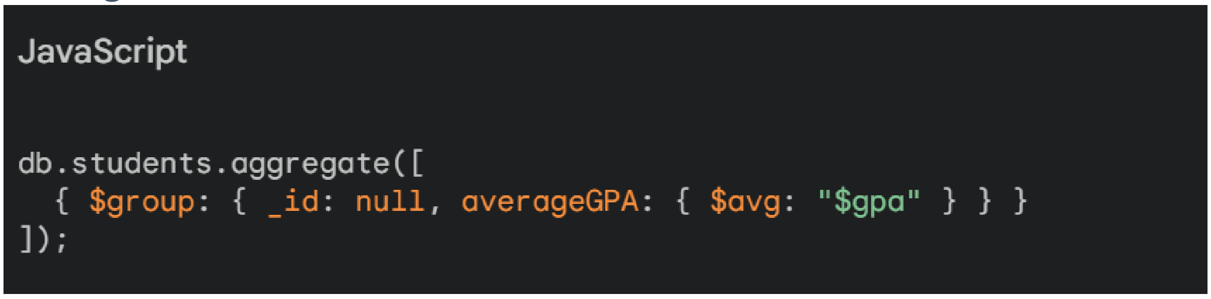
- Work with objects.
  - `$objectToArray`, `$mergeObjects`, `$concatObjects`, `$objId`, `$literal`

Date Operators

- Manipulate date and time values.
  - `$dateFromString`, `$dateToParts`, `$dateToString`, `$dayOfMonth`, `$dayOfWeek`, `$dayOfYear`, `$hour`, `$minute`, `$second`, `$millisecond`, `$week`, `$month`, `$year`, `$addToDate`, `$subtractFromTime`

Lets see some examples:

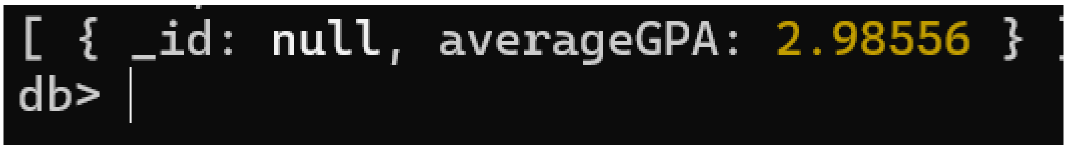
Average GPA of All Students:



```javascript
JavaScript

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

Output:

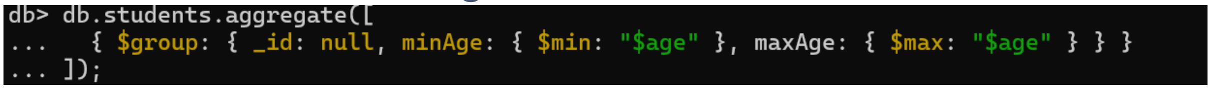


```
[ { _id: null, averageGPA: 2.98556 }
db>
```

Explanation:

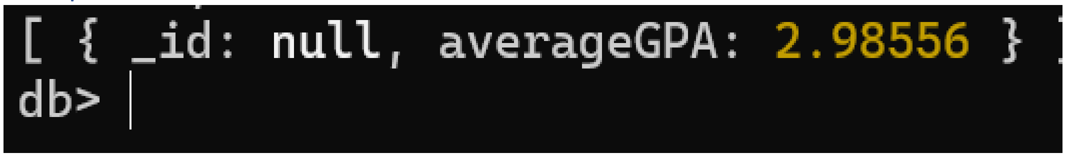
- `$group`: Groups all documents together.
  - `_id: null`: Sets the group identifier to null (optional, as there's only one group in this case).
  - `averageGPA`: Calculates the average value of the "gpa" field using the `$avg` operator.

Minimum and Maximum Age:



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

Output:

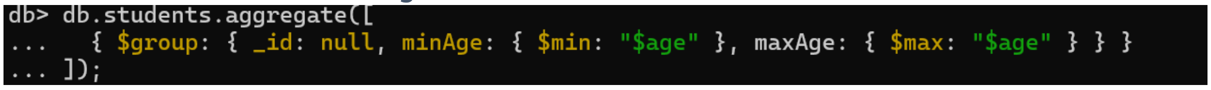


```
[ { _id: null, averageGPA: 2.98556 }
db>
```

Explanation:

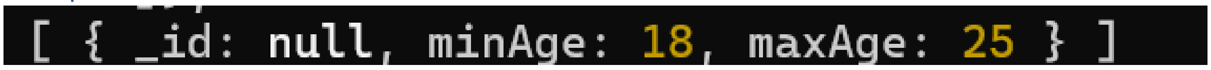
- Similar to the previous example, it uses `$group` to group all documents.
- `minAge`: Uses the `$min` operator to find the minimum value in the "age" field.
- `maxAge`: Uses the `$max` operator to find the maximum value in the "age" field.

Minimum and Maximum Age:



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

Output:

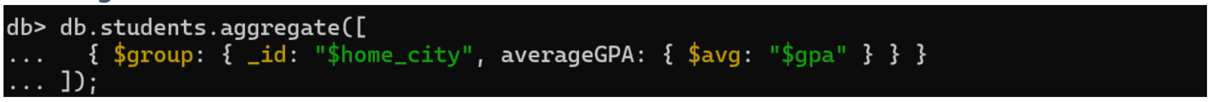


```
[ { _id: null, minAge: 18, maxAge: 25 } ]
```

Explanation:

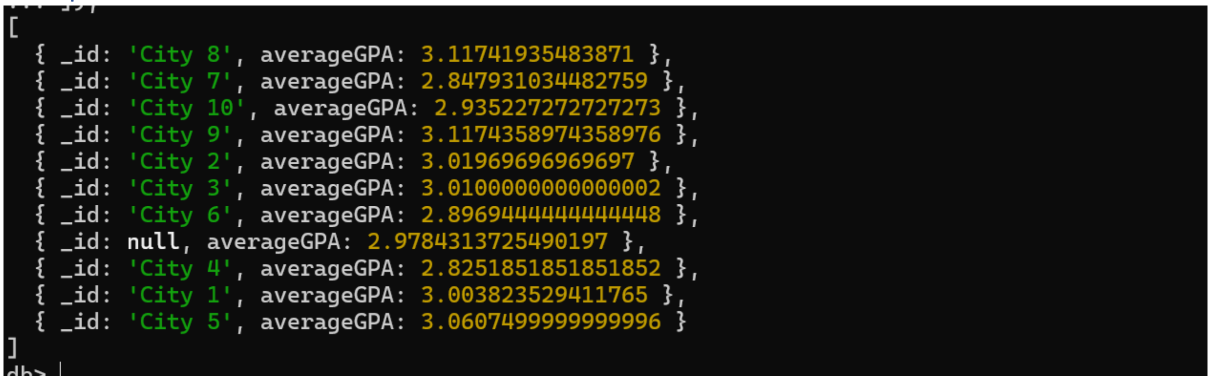
- Similar to the previous example, it uses `$group` to group all documents.
- `minAge`: Uses the `$min` operator to find the minimum value in the "age" field.
- `maxAge`: Uses the `$max` operator to find the maximum value in the "age" field.

Average GPA for all home cities?



```
db> db.students.aggregate([
...   { $group: { _id: "$home_city", averageGPA: { $avg: "$gpa" } } }
... ]);
```

Output:



```
[
  { _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 }
]
db>
```