

How can we group data in MongoDB and calculate the sum of a specific field?

Authored by
stats writer

June 30, 2024

RECOMMENDED CITATION

stats writer (2024). *How can we group data in MongoDB and calculate the sum of a specific field?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=162870>

MongoDB is a document-oriented database that allows for efficient storage and retrieval of large amounts of data. One of its key features is the ability to group data and perform calculations on specific fields within a collection. This is achieved through the use of the aggregation framework, which allows for the grouping of documents based on a specified criteria, such as a common field value. Once the data is grouped, the framework provides various functions, such as `$sum`, to calculate the total value of a specific field across all the grouped documents. This allows for quick and accurate analysis of data, making MongoDB a powerful tool for data management and analysis.

MongoDB: Group By and Sum

You can use the following syntax to group by and count in MongoDB:

```
db.collection.aggregate()
```

Note that `field_name1` is the field you'd like to group by and `field_name2` is the field you'd like to sum.

The following examples show how to use this syntax with a collection teams with the following documents:

```
db.teams.insertOne({team: "Mavs", position: "Guard",  
points: 31})db.teams.insertOne({team: "Spurs",  
position: "Forward", points:  
22})db.teams.insertOne({team: "Rockets", position:  
"Center", points: 19})db.teams.insertOne({team:  
"Warriors", position: "Forward", points:
```

```
26})db.teams.insertOne({team: "Cavs", position: "Guard", points: 33})
```

Example 1: Group By and Sum

We can use the following code to group by the 'position' field and count the sum of points for each position.

```
db.teams.aggregate()
```

This returns the following results:

```
{_id: 'Forward', count: 48 }  
{_id: 'Guard', count: 64 }  
{_id: 'Center', count: 19 }
```

This tells us:

The players with position 'Forward' have a total of 48 points. The players with position 'Guard' have a total of 64 points. The players with position 'Center' have a total of 19 points.

Example 2: Group By and Sum (Then Sort)

We can use the following code to find the sum of points

for each position and automatically sort the results in ascending order:

```
db.teams.aggregate()
```

This returns the following results:

```
{ _id: 'Center', count: 19 }  
{ _id: 'Forward', count: 48 }  
{ _id: 'Guard', count: 64 }
```

We can use -1 in the count argument to instead sort the results in descending order:

```
db.teams.aggregate()
```

This returns the following results:

```
{ _id: 'Guard', count: 64 }  
{ _id: 'Forward', count: 48 }  
{ _id: 'Center', count: 19 }
```

Notice that the results are sorted by points in descending order (largest to smallest).

Note: You can find the complete documentation for \$group .

Additional Resources

The following tutorials explain how to perform other common operations in MongoDB:

ARABPSYCHOLOGY.COM