

How can values in MongoDB be rounded to a specific number of decimal places?

Authored by
stats writer

June 30, 2024

RECOMMENDED CITATION

stats writer (2024). *How can values in MongoDB be rounded to a specific number of decimal places?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=162336>

MongoDB values can be rounded to a specific number of decimal places by using the \$round operator in an aggregation pipeline. This operator takes in two parameters - the field name and the desired number of decimal places. It then rounds the values in the specified field to the specified number of decimal places and returns the result in a new field. This allows for precise control over the rounding of values in MongoDB and ensures consistency in data presentation.

MongoDB: Round Values to Decimal Places

You can use the \$round operator in MongoDB to round numeric values to a certain number of decimal places.

This operator uses the following basic syntax:

```
db.myCollection.aggregate( }} } ])
```

This particular example rounds the values in the field "value" to one decimal place.

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

```
db.teams.insertOne({team: "Mavs", points: 31.345})
```

```
db.teams.insertOne({team: "Spurs", points: 22.88})
```

```
db.teams.insertOne({team: "Rockets", points: 19.91})
```

```
db.teams.insertOne({team: "Warriors", points: 26.5})
```

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

Example 1: Round Values to One Decimal Place

We can use the following code to round the values in the "points" field to one decimal place:

```
db.teams.aggregate( {} } ])
```

Here is the output:

```
{ _id: ObjectId("6203d3b11e95a9885e1e763b"),  
rounded_points: 31.3 }  
{ _id: ObjectId("6203d3b11e95a9885e1e763c"),  
rounded_points: 22.9 }  
{ _id: ObjectId("6203d3b11e95a9885e1e763d"),  
rounded_points: 19.9 }  
{ _id: ObjectId("6203d3b11e95a9885e1e763e"),  
rounded_points: 26.5 }  
{ _id: ObjectId("6203d3b11e95a9885e1e763f"),  
rounded_points: 33 }
```

Notice that each value in the "points" field has been rounded to one decimal place.

Example 2: Round Values to Zero Decimal Places

Note that if you don't specify a number of decimal

places, MongoDB will automatically round to zero decimal places.

For example, suppose we use the \$round operator with the following syntax:

```
db.teams.aggregate( {} } ])
```

Here is the output:

```
{ _id: ObjectId("6203d3b11e95a9885e1e763b"),  
rounded_points: 31 }  
{ _id: ObjectId("6203d3b11e95a9885e1e763c"),  
rounded_points: 23 }  
{ _id: ObjectId("6203d3b11e95a9885e1e763d"),  
rounded_points: 20 }  
{ _id: ObjectId("6203d3b11e95a9885e1e763e"),  
rounded_points: 26 }  
{ _id: ObjectId("6203d3b11e95a9885e1e763f"),  
rounded_points: 33 }
```

Notice that each points value is rounded to zero decimal places.

Example 3: Round Values & Display Other Fields in Output

We can use a value of 0 (don't display) or 1 (display) to also show other fields in the output.

For example, the following code shows how to round the points values to two decimal places and also display the "team" field and the "points" field in the output:

```
db.teams.aggregate(  
  }  
  }  
  }  
  ]  
  )
```

Here is the output:

```
{ team: 'Mavs', points: 31.345, rounded_points: 31.34 }  
{ team: 'Spurs', points: 22.88, rounded_points: 22.88 }  
{ team: 'Rockets', points: 19.91, rounded_points: 19.91 }  
{ team: 'Warriors', points: 26.5, rounded_points: 26.5 }  
{ team: 'Cavs', points: 33, rounded_points: 33 }
```

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

Additional Resources

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

ARABPSYCHOLOGY.COM