

How can I group data in MongoDB by date?

Authored by
stats writer

June 30, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I group data in MongoDB by date?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=162878>

Grouping data in MongoDB by date allows you to organize and analyze your data based on specific dates or time periods. This can be done by using the MongoDB aggregation framework, which allows you to group documents by a specific date field and perform operations on the grouped data. This feature is useful for tasks such as generating reports, tracking trends, or identifying patterns in your data. By grouping data by date, you can easily extract meaningful insights and make informed decisions based on time-based data.

MongoDB: Group by Date

You can use the following syntax to group documents by date in MongoDB:

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

Note that `day` is the name of the date field that we'd like to group by in this example.

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

```
db.sales.insertOne({day: new Date("2020-01-20"),  
amount: 40})
```

```
db.sales.insertOne({day: new Date("2020-01-21"),  
amount: 32})
```

```
db.sales.insertOne({day: new Date("2020-01-22"),  
amount: 19})
```

```
db.sales.insertOne({day: new Date("2020-01-23"),
```

```
amount: 29})
```

```
db.sales.insertOne({day: new Date("2020-01-24"),  
amount: 35})
```

Example 1: Group by Date & Count

We can use the following code to count the number of documents, grouped by date:

```
db.sales.aggregate()
```

This query returns the following results:

```
{_id: '2020-01-20', count: 2 }  
{_id: '2020-01-22', count: 1 }  
{_id: '2020-01-21', count: 2 }
```

From the results we can see:

The date 2020-01-20 occurs 2 times. The date 2020-01-22 occurs 1 time. The date 2020-01-21 occurs 2 times.

Example 2: Group by Date & Count (Then Sort)

We can use the following code to count the number of

documents, grouped by date, and sort the results based on count ascending:

```
db.sales.aggregate()
```

This query returns the following results:

```
{ _id: '2020-01-22', count: 1 }  
{ _id: '2020-01-20', count: 2 }  
{ _id: '2020-01-21', count: 2 }
```

```
db.sales.aggregate()
```

This query returns the following results:

```
{ _id: '2020-01-20', count: 2 }  
{ _id: '2020-01-21', count: 2 }  
{ _id: '2020-01-22', count: 1 }
```

Note: You can find the complete documentation for the **sort** function .

Additional Resources

The following tutorials explain how to perform other

common operations in MongoDB:

ARABPSYCHOLOGY.COM