

# How do you use the AND (\$and) operator in MongoDB queries?

Authored by  
**stats writer**

June 30, 2024

## RECOMMENDED CITATION

stats writer (2024). *How do you use the AND (\$and) operator in MongoDB queries?*.  
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=162289>

The AND (\$and) operator in MongoDB queries is used to specify multiple conditions that must all be met in order for a document to be returned in the query results. It is typically used in conjunction with other operators, such as \$gt (greater than) or \$lt (less than), to create more specific search criteria. The \$and operator allows for more complex and precise filtering of data within a MongoDB database.

## MongoDB: Use the AND (\$and) Operator in Queries

You can use the \$and operator in MongoDB to query for documents that meet multiple criteria.

This operator uses the following basic syntax:

```
db.myCollection.find({  
  "$and":  
})
```

This particular example finds all documents in the collection titled myCollection where field1 is equal to "hello" and field2 has a value greater than or equal to 10.

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

```
db.teams.insertOne({team: "Mavs", points: 30,  
rebounds: 8})
```

```
db.teams.insertOne({team: "Mavs", points: 30, rebounds: 12})
```

```
db.teams.insertOne({team: "Spurs", points: 20, rebounds: 7})
```

```
db.teams.insertOne({team: "Spurs", points: 25, rebounds: 5})
```

```
db.teams.insertOne({team: "Spurs", points: 23, rebounds: 9})
```

#### Example 1: Use AND Operator with Two Fields

The following code shows how to find all documents in the teams collection where the "team" field is equal to "Spurs" and the value in the "points" field is greater than or equal to 22:

```
db.teams.find({  
  "$and":  
})
```

This query returns the following documents:

```
{ _id: ObjectId("6201824afd435937399d6b6c"),  
  team: 'Spurs',  
  points: 25,
```

```
rebounds: 5 }  
{ _id: ObjectId("6201824afd435937399d6b6d"),  
team: 'Spurs',  
points: 23,  
rebounds: 9 }
```

Notice that each document in the output contains "Spurs" in the team field and a value greater than or equal to 22 in the points field.

#### Example 2: Use AND Operator with More Than Two Fields

The following code shows how to find all documents in the teams collection where the "team" field is not equal to "Mavs" and the value in the "points" field is greater than or equal to 22 and the value in the "rebounds" field is less than 7:

```
db.teams.find(  
"$and":  
)
```

This query returns the following document:

```
{ _id: ObjectId("6201824afd435937399d6b6c"),
```

```
team: 'Spurs',  
points: 25,  
rebounds: 5 }
```

The "team" field is not equal to "Mavs" The "points" field has a value greater than or equal to 22 The "rebounds" field has a value less than 7

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

#### Additional Resources

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

**MongoDB: How to Query for "not null" in Specific Field**