# Array Operators

Array operators allow you to query documents based on conditions that apply to array fields. They help you filter documents by checking the contents, size, or elements of arrays.

| Operator | Description | Example Usage |
|---|---|---|
| **$all** | Matches arrays that contain all specified elements | { tags: { $all: ["accessories", "wireless"] } } |
| **$size** | Matches arrays with a specified number of elements | { tags: { $size: 3 } } |
| **$elemMatch** | Matches arrays containing at least one element that meets multiple conditions | { reviews: { $elemMatch: { rating: { $gte: 4 }, user: "Bob" } } } |

## 1. $all

- Matches documents where the array contains **all specified values** (anywhere in the array).
- Works with arrays of primitive values like strings or numbers.
- Order of elements doesn’t matter.
- Does **not** require these values to be in the same array element (since array elements are primitives).

**Example 1: Find products tagged with both "accessories" and "gaming"**

```
db.products.find({
 tags: { $all: ["accessories", "gaming"] }
});
```

- *Finds products whose tags include both "accessories" and "gaming".*

**Example 2: Find users with both "admin" and "editor" roles**

```
db.users.find({
 roles: { $all: ["admin", "editor"] }
});
```

- *Finds users that have both "admin" and "editor" roles.*

## 2. $size

- Matches documents where the array contains **exactly the specified number of elements**.
- Useful for filtering based on array length.

**Example 1: Find products that have exactly 3 tags**

```
db.products.find({
 tags: { $size: 3 }
});
```

- *Finds products with exactly 3 tags.*

**Example 2: Find posts with no comments**

```
db.posts.find({
 comments: { $size: 0 }
});
```

- *Finds posts that have an empty comments array.*

## 3. $elemMatch

- Matches documents where the array contains **at least one element** that satisfies **all the specified conditions**.
- Works on arrays of embedded documents (objects).
- Ensures that multiple conditions apply to the **same element** in the array, avoiding false positives where different conditions are met by different elements.

**Example 1: Find products where at least one review has rating ≥ 4 and is by user "Bob"**

```
db.products.find({
 reviews: {
   $elemMatch: {
     rating: { $gte: 4 },
     user: "Bob"
   }
 }
});
```

- *Finds products with at least one review matching both conditions.*

**Example 2: Find orders where at least one item has quantity > 2 and price < 100**

```
db.orders.find({
 items: {
   $elemMatch: {
     quantity: { $gt: 2 },
     price: { $lt: 100 }
   }
 }
});
```

- *Finds orders with at least one item meeting both quantity and price conditions.*