

Referência > Operadores > Operadores de consulta e projeção > Operadores de consulta de comparação > \$ in

\$ em

Nesta página

- Exemplos
- Use o \$inoperador para combinar valores
- Use o \$inoperador para combinar valores em uma matriz
- Use o \$inoperador com uma expressão regular

\$in

O \$inoperador seleciona os documentos em que o valor de um campo é igual a qualquer valor na matriz especificada. Para especificar uma \$inexpressão, use o seguinte protótipo:

Para comparação de diferentes valores do tipo BSON, consulte a ordem de comparação do BSON especificada .

```
{ campo : { $ in : [ < valor1 > , < valor2 > , ... < valorN > ] } }}
```

Se o `field` contém uma matriz, em seguida, o \$inoperador selecciona os documentos cujos `field` contém uma matriz que contém, pelo menos, um elemento que coincide com um valor na gama especificada (por exemplo `<value1>`, `<value2>`, etc.)

Alterado na versão 2.6: O MongoDB 2.6 remove o limite combinatório para o \$in operador que existe para versões anteriores [↗](#) do operador.

Exemplos

Use o \$inoperador para combinar valores

Considere o seguinte exemplo:

```
db . inventário . find ( { qty : { $ in : [ 5 , 15 ] } } )
```

Esta consulta seleciona todos os documentos (

Search Documentation

15. Embora você possa expressar essa consult

\$or operador ao executar verificações de igualdade no mesmo campo.

Use o \$in operador para combinar valores em uma matriz

A coleção `inventory` contém documentos que incluem o campo `tags`, como a seguir:

```
{ _id: 1, item: "abc", qty: 10, tags: [ "school", "clothing" ], sale: false }
```

Then, the following `update()` operation will set the `sale` field value to `true` where the `tags` field holds an array with at least one element matching either `"appliances"` or `"school"`.

```
db.inventory.update(
  { tags: { $in: ["appliances", "school"] } },
  { $set: { sale: true } }
)
```

For additional examples in querying arrays, see:

- Query an Array
- Query an Array of Embedded Documents

For additional examples in querying, see:

- Query Documents

Use the \$in Operator with a Regular Expression

The `$in` operator can specify matching values using regular expressions of the form `/pattern/`. You *cannot* use `$regex` operator expressions inside an `$in`.

Consider the following example:

```
db.inventory.find( { tags: { $in: [ /^be/, /^st/ ] } } )
```

This query selects all documents in the `inventory` collection that start with `be` or `st` or an array with at least one

[Search Documentation](#)

SEE ALSO:

`find()`, `update()`, `$or`, `$set`, `$elemMatch`.