



Dominar el Find

Índice

1. Filtros Avanzados
2. Arrays & Documentos Embebidos
3. Ordenar
4. Paginar y limitar
5. Más Cursores



1. Filtros Avanzados

Selectores

- **\$eq**. Igual al valor especificado
- **\$gt**. Mayor que valor especificado
- **\$gte**. Mayor o igual que valor especificado
- **\$in**. Cualquier valor de los especificados (array)
- **\$lt**. Menor que valor especificado
- **\$lte**. Menos o igual que valor especificado
- **\$ne**. Distinto de valor especificado
- **\$nin**. Cualquier valor que no esté especificado (array)



Lógicos

- **\$and**. Que cumpla todas las condiciones especificadas (array de filtros)
- **\$not**. Invierte las condición especificada (filtros)
- **\$nor**. NOT OR $((!expr1)AND(!expr2))$ (array de filtros)
- **\$or**. Que cumpla algunas de las condiciones especificadas (array de filtros)

<https://docs.mongodb.com/manual/reference/operator/query-logical/>

Elementos

- **\$exists**. Que contiene el campo o no (true/false)
- **\$type**. Filtra por el tipo

<https://docs.mongodb.com/manual/reference/operator/query-element/>

Evaluacion

- **\$expr**. Permite el uso de expresiones de agregación dentro de la query.
- **\$mod**. Realiza una operación de módulo en el valor de un campo y selecciona documentos con un resultado específico.
- **\$regex**. Permite expresiones regulares.
- **\$text**. Búsquedas de texto.
- **\$jsonSchema**. Búsquedas de texto.

<https://docs.mongodb.com/manual/reference/operator/query-evaluation/>



2. Arrays & Documentos Embebidos

Ejemplo:

```
{ item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" }
```

Documento Embebido

- Consulta **Exacta** de documento
`find({ size: {h: 14, w: 21, uom: "cm" } }`
- Consulta sobre **un elemento** del documento
`find({ "size.uom": "cm" })`
`find({ "size.h": { $lt: 15 } })`
- Consulta sobre **varios elementos** de un documento (and)
`find({ "size.h": { $lt: 15 }, "size.uom": "cm", status: "A" })`



Ejemplo:

```
{ item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [ 14, 21 ] }
```

Array

- Consulta sobre elementos de array en el orden, número de elementos y valor exacto
`find({ tags: ["blank", "red"] })`
- Consulta sobre elementos de array que incluya todos los valores sin importar el orden
`find({ tags: { $all: ["red", "blank"] } })`
- Consulta sobre elementos de array que satisface al menos un valor o condición
`find({ tags: "red" })`
`find({ dim_cm: { $gt: 25 } })`
- Consulta sobre elementos de array que satisface al menos un valor o condición de todas ellas (se deben cumplir todas ellas aunque no sea sobre un mismo elemento)
`find({ dim_cm: { $gt: 15, $lt: 20 } })`



Ejemplo:

```
{ item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [ 14, 21 ] }
```

Array

- Consulta sobre elementos de array que satisface todas las condiciones (al menos alguno de los elementos)

```
find( { dim_cm: { $elemMatch: { $gt: 22, $lt: 30 } } } )
```

- Consulta sobre un elementos del array

```
find( { "dim_cm.1": { $gt: 25 } } )
```

- Consulta sobre el número de elementos del array

```
find( { "tags": { $size: 3 } } )
```



Ejemplo:

```
{ item: "journal", instock: [ { warehouse: "A", qty: 5 }, { warehouse: "C", qty: 15 } ] }
```

Array de Documentos Embebidos

- Consulta sobre todos los documentos de un array, donde al menos un documento debe ser exactamente igual, valores, orden y número de campos
`find({ "instock": { warehouse: "A", qty: 5 } })`
- Consulta sobre todos los documentos de un array, donde al menos un documento debe tener el campo a filtrar sin especificar el orden
`find({ 'instock.qty': { $lte: 20 } })`
- Consulta sobre un documento del array, donde debe tener el campo a filtrar sin especificar el orden
`find({ 'instock.0.qty': { $lte: 20 } })`
- Consulta sobre todos los documentos de un array, donde al menos un documento satisface todas las condiciones
`find({ "instock": { $elemMatch: { qty: 5, warehouse: "A" } } })`
- Consulta sobre todos los documentos de un array, donde al menos un documento satisface alguna de las condiciones
`find({ "instock.qty": { $gt: 10, $lte: 20 } })`



Filtros

\$all. Encuentra arrays que contienen todos los elementos especificados en la consulta

\$elemMatch. Selecciona documentos si el elemento en el campo del array coincide con todas las condiciones especificadas de \$elemMatch

\$size. Selecciona si tiene el tamaño especificado

Proyecciones

\$. Selecciona el primer elemento del array

\$elemMatch. Muestra los campos si cumple criterios

\$slice. Controla el número de elementos de un array que devuelve una consulta

Modificadores

\$each . Para añadir varios items

\$position. Especifica la posición en el array

\$slice. Limita el tamaño del array

\$sort. Reordena el array

Update

\$. Actúa como un marcador de posición para actualizar el primer elemento que coincide con la condición de consulta

\$[]. Actúa como un marcador de posición para actualizar todos los elementos que coinciden con la condición de consulta

\$[<identifier>]. Actúa como un marcador de posición para actualizar todos los elementos que coinciden con la condición de consulta de

ArrayFilters.

\$addToSet. Agrega elementos a un array sólo si aún no existen en el conjunto.

\$pop. Elimina el primer o último elemento de un array

\$pull. Elimina todos los elementos de un array especificada en una query

\$push. Añade un item al array

\$pullAll. Elimina todos los elementos de un array



3. Ordenar

sort. Para ordenar por los campos

- Ascendente. 1
- Descendente. -1

```
SELECT *  
FROM people  
WHERE status = "A"  
ORDER BY user_id ASC
```

```
db.people.find( { status: "A" } ).sort( { user_id: 1 } )
```

```
SELECT *  
FROM people  
WHERE status = "A"  
ORDER BY user_id DESC
```

```
db.people.find( { status: "A" } ).sort( { user_id: -1 } )
```



4. Pagar y limitar

limit. Número máximo de documentos que devolverá el cursor

skip. Número de documentos a saltar

```
function printStudents(startValue, nPerPage) {  
  let endValue = null;  
  db.students.find( { _id: { $lt: startValue } } )  
    .sort( { _id: -1 } )  
    .limit( nPerPage )  
    .forEach( student => {  
      print( student.name );  
      endValue = student._id;  
    } );  
  
  return endValue;  
}
```

```
SELECT *  
FROM people  
LIMIT 1
```

db.people.findOne()

or

db.people.find().limit(1)

```
SELECT *  
FROM people  
LIMIT 5  
SKIP 10
```

db.people.find().limit(5).skip(10)



5. Más cursores

`find()` devuelve un cursor de documentos que se puede iterar. Las funciones principales son:

- `close`. Cierra el cursor.
- `count`. Cuenta número de documentos del cursor original (sin iterar)
- `forEach`. Aplicar una función a todos los documentos del cursor
- `hasNext`. Devuelve true o false si quedan documentos que iterar.
- `skip`. Salta número de documentos.
- `limit`. Limita número de documentos del cursor.
- `next`. Itera el siguiente documento.
- `pretty`. Mostrar JSON más visual.
- `size`. Cuenta número documentos después de realizar limit y/o skip.
- `sort`. Ordena la salida.
- `toArray`. Itera cursor y lo asigna a un array.

<https://docs.mongodb.com/manual/reference/method/js-cursor/>

<https://docs.mongodb.com/manual/reference/method/js-collection/>



