

Todo lo que necesitas saber para hacer consultas múltiples son los OPERADORES LÓGICOS y RELACIONALES.

Estos operadores son \$and, \$or y \$not

Una consulta múltiple en sí, de forma básica, si quieres recuperar por ejemplo los cuadros que tengan un precio de 50€ y que tengas 10 unidades. Sería;

```
db.cuadros.find({precio : 50, cantidad : 10 })
```

esto devolverá los cuadros que tienen un precio de 50 euros y además la cantidad exacta en bbdd de estos cuadros sean 10. esto limita la respuesta ya que se tienen que cumplir las dos condiciones para sacar ese cuadro.

Si queremos ampliar el abanico ya entra en juego los operadores lógicos.

En el caso de \$and la sintaxis para el find es:

```
db.cuadros.find({$and : [{precio:50}, {cantidad:20}] })
```

El valor para el operador \$and es un arreglo con cada una de las condiciones que debe cumplir.

Para los operadores \$or y \$not no hay una forma de disponer una sintaxis implícita.

un ejemplo del empleo de los operadores \$or y \$not:

Para recuperar los cuadros que tienen un precio mayor o igual a 50 o la cantidad es 1:

```
db.cuadros.find({$or: [{precio:{$gte:50}}, {cantidad:1}] })
```

Puedes ver que \$or lo que hace es decir que muestre cualquier dato que cumpla algunas de las condiciones que tu metas despues, osea, transformado a lenguaje humano esta diciendo “busca los cuadros que tengan el precio mayor o igual a 50 **Ó** que la cantidad sea 1”.

El caso de \$not es igual pero al revés. Por ejemplo:

```
db.cuadros.find({precio: {$not: {$gte:50}} })
```

esta consulta esta diciendo “busca los cuadros que NO tenga un precio igual o mayor a 50. o lo que es lo mismo, “busca los cuadros que su precio sea menor de 50”

Has visto que en algunas consultas anteriores te he puesto algunos operadores como \$gte. Esto son operadores relacionales que son en total unos cuantos:

son \$eq, \$gt, \$gte, \$lt, \$lte, \$in, \$nin y \$ne

Al final, estos operadores seleccionan dentro de una consulta aquellos elemento que cumplan la condicion que le metamos.
Por ejemplo.

En la consulta que devolvía aquellos cuadros que valgan mas o igual de 50 o que la cantidad fuera 1 veíamos que dentro de precio había un \$gte.

```
db.cuadros.find({$or: [{precio:{$gte:50}}, {cantidad:1} ]})
```

Este \$gte es igual a poner “>=” o lo que es lo mismo “mayor o igual que” de esta forma le decimos a una consulta, que nos devuelva un articulo que el precio sea mayor o igual que 50. en ese caso. Para que tengas un ejemplo rapido de todos estos operadores, esta logica se aplica dependiendo de lo que quieras hacer:

- \$eq - equal - igual
- \$lt - low than - menor que
- \$lte - low than equal - menor o igual que
- \$gt - greater than - mayor que
- \$gte - greater than equal - mayor o igual que
- \$ne - not equal - distinto
- \$in - in - dentro de
- \$nin - not in - no dentro de

Recuperar todos los cuadros que tienen un precio mayor a 40:

```
db.cuadros.find({ precio: { $gt:40 } })
```

Recuperar todos los cuadros que en el campo cantidad tiene 50 o más:

```
db.cuadros.find( { cantidad: { $gte : 50 } })
```

Recuperar todos los cuadros que en el campo cantidad hay un valor distinto a 50:

```
db.cuadros.find( { cantidad: { $ne : 50 } })
```

Recuperar todos los cuadros cuyo precio estén comprendidos entre 20 y 45:

```
db.cuadros.find( { precio: { $gte : 20 , $lte : 45 } })
```

Recuperar todos los cuadros del pintor 'Picasso':

```
db.cuadros.find( { pintor: { $in : ['Picasso'] } })
```

Recuperar todos los cuadros que no pertenezcan al pintor 'Picasso':

```
db.cuadros.find( { pintor: { $nin : ['Picasso'] } })
```