

¿CÓMO HACER CONSULTAS EN MONGODB? OPERADORES

Categoría

MongoDB (/es/taxonomy/term/90)

Descripción

OPERADORES DE COMPARACIÓN

Los operadores de comparación se usan en filtros que, en vez de pasar únicamente un valor, pasan un nuevo documento que contiene las comparaciones a realizar. Por ejemplo, en vez de expresar `{ "edad": 19 }` para obtener las personas cuya edad sea exactamente 19, indico un objeto `{ "edad" : { $gt: 19 } }` para expresar las personas cuya edad sea mayor que 19

- `$eq` : igual que (**e**qual)
- `$ne` : distinto de (**n**ot **e**qual)
- `$gt` : valores mayores que (**g**reater **t**han)
- `$gte` : valores mayores o iguales que (**g**reater **t**han or **e**qual)
- `$lt` : valores menores que (**l**ower **t**han)
- `$lte` : valores menores o iguales que (**l**ower **t**han or **e**qual)
- `$in` : sus valores se encuentran dentro del array especificado. Por ejemplo: `{ "pais": { $in: ["España", "Francia"] } }` mostrará todos los documentos cuyo campo `pais` sea España o Francia
- `$nin` : operación contraria a `$in`, obtiene los documentos cuyo valor NO esté en el array

OPERADORES DE ELEMENTO

Los operadores de elemento se utilizan para comprobar si cierto campo existe o no, y su tipo

- `$exists` : comprueba que el campo existe. Por ejemplo: `{ "nacionalidad" : { $exists: true } }` muestra los documentos que contienen un campo llamado `nacionalidad`. Fijarse en que no es lo mismo que filtrar por un campo con valor a `null`, puesto que en ese caso devolvería tanto los documentos que contienen el campo seteado a `null`, como aquellos que no contienen ese campo
- `$type` : comprueba el tipo del campo. Por ejemplo: `{ "longitud": { $type: "int" } }` obtendrá los documentos cuyo campo `longitud` tenga un valor entero de 32 bit

OPERADORES LÓGICOS

Los operadores lógicos permiten realizar filtrados atendiendo a varias condiciones:

- `$or` : filtra los documentos que cumplan alguna de las condiciones del array. Por ejemplo: `{ $or: [{ "edad": { $gte: 60 } }, { "edad": { $lte: 10 } }] }` filtra los documentos cuyo campo `edad` tenga valor mayor o igual que 60 o valor menor o igual que 10
- `$and` : similar a `$or`, pero en este caso los documentos deben cumplir todas las condiciones del array. Su uso puede resultar superfluo teniendo en cuenta que en un filtro normal, las condiciones indicadas se juntan mediante operadores lógicos `Y`. Normalmente se usa en situaciones en las que se deben aplicar varias condiciones sobre un mismo campo. En una consulta normal, los campos deben ser únicos.

OPERADORES SOBRE ARRAYS

Los operadores sobre arrays permiten filtrar documentos con campos de tipo array que cumplan ciertas condiciones:

- `$all` : el array debe contener todos los valores indicados en el parámetro, que a su vez es un array. Los elementos pueden estar en cualquier orden. Por ejemplo: `{ "ciudades" : { $all: ["Madrid", "Barcelona"] } }` El array `ciudades` debe tener los datos Madrid y Barcelona.
- `$size` : especifica el tamaño del array. Por ejemplo: `{ "ciudades": { $size: 1 } }` El array `ciudades` deberá ser de tamaño 1
- `$elemMatch` : cuando los documentos principales contienen arrays que a su vez contienen documentos, `$elemMatch` sirve para filtrar los documentos principales cuyo campo array contiene documentos que cumplen todos los criterios especificados en el documento pasado como parámetro. Por ejemplo, supongamos que tenemos una colección de clases de un colegio. Cada clase contiene un campo que es de tipo array con los nombres de los alumnos de esa clase y la edad. Imaginemos que queremos buscar las clases de una facultad que contienen alumnos cuyo nombre sea Maria y su edad sea 23. Si usásemos `{ "alumno.nombre" : "Maria", "alumno.edad" : 23 }` nos recuperará todas las clases que contengan alumnos con nombre Maria o alumnos con edad 23. Si usamos `{ "alumno": { $elemMatch: { "nombre" : "Maria", "edad": 23 } } }`, entonces nos mostrará las clases

cuyo array de alumnos contengan documentos cuyo campo nombre sea Maria **Y** su campo edad sea 23

TWITTER ([HTTPS://TWITTER.COM/RCARRASCOMU](https://twitter.com/rcarrasco))

LINKEDIN ([HTTPS://WWW.LINKEDIN.COM/IN/RAMON-CARRASCO-MUNOZ](https://www.linkedin.com/in/ramon-carrasco-munoz))

GITHUB ([HTTPS://GITHUB.COM/RAMONCARRASCOM](https://github.com/ramoncarrasco))

COPYRIGHT © 2019-2020 RAMONCARRASCO.ES

COSAS LEGALES (/COSASLEGALES) · CRÉDITOS (/CREDITOS) · POLÍTICA DE COOKIES (/COOKIES) · TÉRMINOS Y CONDICIONES (/LEGAL) · POLÍTICA DE PRIVACIDAD (/POLITICA-PRIVACIDAD)

TODOS LOS ELEMENTOS AQUÍ PUBLICADOS FORMAN PARTE DE MI PROPIO CONOCIMIENTO, Y DESARROLLAN LOS TEMAS HASTA DONDE ALCANZA MI CONOCIMIENTO. SIEMPRE INTENTARÉ MANTENER LA INFORMACIÓN MÁS ACTUAL Y FIEL A LA REALIDAD POSIBLE. NO OBSTANTE, PUEDE SER QUE ÉSTA NO SEA TOTALMENTE FIDEDIGNA, Y PUEDA SER MÁS AMPLIA O ESPECÍFICA. POR LO TANTO, NO ME HAGO RESPONSABLE DEL USO QUE SE HAGA DE LOS CONTENIDOS EXPUESTOS EN ESTA WEB, YA SEA TEORÍA, O CÓDIGO DE PROGRAMACIÓN.