Explicación a los operadores y consultas

Miguel Reyes García

Administracion de redes y sistemas

IES Punta del verde

Explicación Consultas y operadores MusicaDB

1º db.musica.find({ $and: [{niños:true}, { nacimiento: {$lte: new Date("2010-01-01")}},
{ nacimiento: {$gte: new Date("1960-01-01")}}]},{genero:1, niños:1,nacimiento:1})

El $and coge el campo de niños, que el nacimiento del genero sea menor o igual a 2010 y que sea mayor o igual a 1960, mostrará solo el campor generos, niño y el año en que nació el genero del que queremos información.

2º db.musica.find({ $and: [{ genero: { $nin: ["Rock", "Blues"] },
{precioConcierto: {$lte: 250}}] },{genero:1, precioConcierto:1})

El $and coge el campo genero que con el $nin evita que coja generos que coincidan con Rock o Blues y el and cogerá los precios de concierto que sean igual o más bajos de 250 euros además el resultado mostrará el género y el precio.

3º db.musica.find({"exponentes.nuevaEra":{$eq: "Duki"}},{genero:1, "exponentes.nuevaEra":1})

Exponentes.nuevaEra deberá ser igual a Duki y que me muestre solo el género y el apartado de exponentes.nuevaEra en la que deberá de estar este cantante, nos mostrará el resultado el género que canta y el apartado donde el estará

He añadido una variante en el cual en vez de mostrar todos los cantantes de su apartado me muestre solo su nombre en el Array para ello añadiremos la variante $slice:-1 que al ser el cantante el ultimo de la lista mostrará su nombre si pusiese 1 sería el primero del array:

3.1 db.musica.find({"exponentes.nuevaEra":{$eq: "Duki"}},{"exponentes.nuevaEra":{$slice:-1} })

4º db.musica.find({ sonido: { $regex: /^Ritmo/ } },{genero:1, sonido:1})

Continuamos con un regex el cual nos mostrará las descripciones de sonido que comienzan con la primera palabra Ritmo el cual ^ quiere decir que comienza por, y el resultado mostrará el género y el sonido que es la descripción.

5º db.musica.find({"exponentes.nuevaEra":{$exists:false}},{genero:1, exponentes:1})

En este caso usamos el $Exist el cual hace que al poner la variante false busca los documentos que no contengan este apartado y el resultado mostrará el género y el campor de array exponentes.

6º db.musica.find({ notaExpertos : {$size : 3}},{genero:1,notaExpertos:1})

Con el operador $size quiero buscar entre los documentos uno que no contenga 3 notas en el campor notaExpertos, quiero que en el resultado me muestre el género y el campo notaExpertos.

7º db.musica.find({ $and: [{notaExpertos:{$eq:10} }, {paisOrigen:{$not:{$eq:["EEUU"]}} }}, {genero:1,notaExpertos:1, paisOrigen:1})

$and coge que la nota de los expertos sea $eq o igual a 10, que el pais de origen no sea igual que he mezclado $not y $eq a EEUU y quiero que en el resultado me salgan los campos genero, notaexpertos y su pais de origen.

8º db.musica.find({ $nor: [{ notaExpertos: {$lte: [4.99] }}, { precioConcierto: { $gt: 100 } }, { niños: false }] },{genero:1,notaExpertos:1, niños:1,precioConcierto:1})

$nor lo que consigue en este caso es que la nota de los expertos no contengan valores de menos de 4,99 o iguales y que el precio del concierto no sea mayor a 100 euros además de que el valor del campo niños no puede ser false. El resultado mostrará el genero, nota, niños y el precio del concierto.

9º db.musica.find({ $and: {"exponentes.nuevaEra":{ $in:["Travis Scott", "Drake"]}}, {paisOrigen: {$eq:["EEUU"]} }},{genero:1, exponentes:1 ,precioConcierto:1})

$and recoge que el cantante de nuevaera deberá ser travis o drake y que el pais de origen del genero sea $eq a EEUU. En el resultado debe mostrar el genero, exponentes y precio del concierto.

También podemos hacerlo con un $all que deberá coger la array que contenga todos eso valores que le proporcionamos en al orden en este caso Drake y Travis

9.1 db.musica.find({ $and: [{"exponentes.nuevaEra":{ $all:["Travis Scott", "Drake"]},{paisOrigen: {$eq:["EEUU"]} }] },{genero:1, exponentes:1 ,precioConcierto:1})

10º db.musica.find({ "epocaExito.año": { $elemMatch: { $gte: new Date("2001-01-01"), $lte: new Date("2016-01-01")}}} ,{genero:1, "epocaExito.año":1 ,})

$Elemmatch se asegura que la fecha mayor a 2001 o la fecha menor a 2016 sea uno de los resultados de la array a la que enfocamos la orden en este caso año de epoca de exito. En el resultado nos mostrará el género y el campor de año de exito

Y he añadido otra orden con valores de antes del 2001:

10.1 db.musica.find({ "epocaExito.año": { $elemMatch: {$lte: new Date("2001-01-01")}}} , {genero:1, "epocaExito.año":1 ,})

11º db.ventadiscos.find({ $or: [{artista:{$eq:"Michael Jackson"} }, {artista:{ $regex: /emine./i }}] },{disco:1,artista:1, album:1, ventas:1})

$or busca los que sean igual a Michael Jackson o que tenga el artista Eminem que en mi caso he hecho un regex con un punto de no saber por que letra acaba. El resultado me muestra nombre del disco, artista, album y ventas del album.

12º db.musica.find({ sonido: { $regex: /.*beat/i } },{genero:1, sonido:1})

Me hago un regex con la descripcion de sonido de un genero que tienen saltos de páginas /n en este caso busca las descripciones que despues del salto tengan la palabra beat y quiero que el resultado me muestre el genero y la descripción.

13º db.ventadiscos.find({ $and: [{orden:{$eq:1} }, {disco:{$ne:"Platino" }},{artista:{$not:{$eq:"Britney Spears"}} }] },{disco:1,artista:1, album:1, ventas:1})

El $and coge que la orden sea igual a 1 y que el disco no sea platino con el no igual o $ne y también que no sea igual a Britney Spears. Que el resultado muestre el disco, el artista, su album y las ventas.

14º db.musica.find({"exponentes.nuevaEra":{$exists: true, $eq:"Linkin Park"}},{genero:1, exponentes:1})

He querido implementar también un simple And implícito en el que nuevaEra debe cumplir 2 requisitos, existir el campo y que contenga el grupo Linkin Park. Que muestre el género y los exponentes

15º db.musica.find({ $and: [{ miOpinion: {$lt: 7.25 }}, {$or: [{nacimiento:{$lte:new Date("1990-01-01")}},{niños:{$eq: false}}]}]},{genero:1,miOpinion:1, niños:1,nacimiento:1})

Con el and cogemos la obligación de que mi nota sea de menos de 7,25 y cogemos el or que o nos mostrará fechas de nacimiento de generos antes del 1990 o que no sea para niños.

Mostrará el género, opinion, niños y nacimiento.

Propia investigación

1º db.musica.update({ genero:"Reggaeton"},{ $set: {niños: true}})

No es una consulta pero descubrí esta función en la que podemos cambiar un campo como por ejemplo aqui en el cual colocamos un identificador del documento en este caso el genero después el $set que se encargará de hacer el cambio y el cambio que queremos hacer.

También podemos hacerlo con varios documentos con updateMany.

2º db.musica.aggregate([{$project : {genero:1, _id:0}},{$sort : {genero : 1}}, {$skip: 2}])

Vamos a empezar a trabajar con los datos aportados:

Aquí usaremos aggregate que es capaz de compilar varios operadores de etapa como es el caso de $sort que ordena si le ponemos genero: 1 en este caso alfabeticamente los generos y el $skip saltaría los dos primeros en el orden.

$project basicamente es el operador donde podemos poner los valores que queremos que se muestren en el resultado final.

3º db.ventadiscos.aggregate({$project : {_id:0,disco:1,artista:1,UranioSioNo:{$cmp: ['$disco',"Uranio"]}}})

En este caso vamos a señalar con un 0 o en -1 en el caso de que un artista no tenga certificación Uranio, como podemos observar crea un campo llamado UranioSioNo y con el $cmp vamos a mostrar estos números, si la respuesta es Uranio saldrá un 0 si la respuesta es diamante será -1

Con el project mostrará el disco y artista además de UranioSioNo

4º db.musica.find({notaExpertos:{$eq:[1, 6, 9, 4.5]}},{notaExpertos:{$slice:3}})

Volvemos al find y las consultas, Aquí si las notas de los expertos es $eq a estos datos con $slice nos proporciona las 3 primeras notas no la cuarta si ponemos en vez de eso -3 nos mostrará las tres ultimas no la primera, he usado este comando en consultas básicas de arrays

5º db.musica.updateMany({},[{ $set:{ precioConcierto:{ $floor:{ $multiply: [{ $rand: {} }, 500] }}}}])

Volvemos con el updateMany en el cual vamos a cambiar con $set el preciodelconcierto, vamos a usar $floor para que los valores que se repartan no cambien a decimales y sean enteros, $rand se encargará de dar números aleatorios hasta 500 debido al $multiply ya que rand suele dar valores de 0 o 1