Proyecto final del 1º trimestre MONGO

Consolación Begines Roldán 1ºDAM

El proyecto consta de una base de datos que sería de una empresa de arte donde venden los cuadros originales de artistas históricos como por ejemplo Diego Velázquez llamada “Concepto arte” un nombre ideal, convincente y memorable para empresas que ofrecen una amplia gama de productos creativos e importantes.
La base de datos se compone de dos colecciones: “cuadros” y “pedidos”, con 30 y 15 registros respectivamente. Además, consta de 18 consultas a realizar, de menor a mayor complejidad, para experimentar y comprobar la efectividad de la base de datos de esta empresa.

Dentro de este proyecto, se pone a prueba el objetivo de la asignatura, la creación original de una base de datos funcional y efectiva usando tanto todos los operadores aprendidos en clase, como algunos nuevos que han sido aprendidos por cuenta ajena. Veremos a continuación, la diferencia de los operadores lógicos y relacionales, sus usos y una breve descripción de cada uno. Además, se mostrarán varios ejemplos de consultas utilizadas en la base de datos, desde lo más simple hasta lo más complejo para ofrecer una gran facilidad de atención y entendimiento.

Operadores lógicos y relacionales

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. Seria;

```
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.

Comenzamos con los operadores $and, $or y $not.

-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} ]})
```

Se busca ver que $or lo que hace es decir que muestre cualquier dato que cumpla algunas de las condiciones que tu metas después, es decir, transformado a lenguaje humano está 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 tengan un precio igual o mayor a 50, o lo que es lo mismo, “busca los cuadros que su precio sea menor de 50”

Dentro de los operadores relacionales nos podemos encontrar con: $eq, $gt, $gte, $lt, $lte, $in, $nin y $ne

Al final, estos operadores seleccionan dentro de una consulta aquellos elementos que cumplan la condición que le metamos.

Por ejemplo:

En la consulta que devuelve aquellos cuadros que valgan más o igual de 50 o que la cantidad fuera 1 vemos que dentro del 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 artículo que el precio sea mayor o igual que 50, en ese caso. Para que tengas un ejemplo rápido de todos estos operadores, esta lógica 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

Veamos ahora, más ejemplos seguidos de las consultas utilizadas donde estén compuestas por estos operadores, para que se entienda desde un punto de vista más práctico y funcional:

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'] } })
```