

Nociones teóricas ejercicio001

(Proyecto 1002_QUERY_DOCUMENTS)

Antonio Ávila Delgado

1º ASIR

DESCRIPCIÓN DEL PROYECTO

En este proyecto vamos a poblar con documentos una colección llamada INVENTORY con información, en inglés, de material de oficina. La poblaremos con valores tanto alfanuméricos como numéricos (y dentro de estos, usaremos decimales, para mayor variedad). Después haremos distintas operaciones de búsqueda/consultas sobre dichos datos.

INSERCIÓN DE DOCUMENTOS

Para la inserción de documentos, como son varios, en vez de usar el comando **insertOne** (que los inserta de uno en uno), vamos a usar el comando **insertMany**, que es más efectivo en esta situación.

La sintaxis es (usando como ejemplo la colección INVENTORY):

```
db.inventory.insertMany([
  { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "Available" },
  { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "Available" },
  { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "Unavailable" },
  { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "Unavailable" },
  { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "Obsolete" }
]);
```

REALIZACIÓN DE CONSULTAS

Consulta 1: Comenzamos primero con una sencilla consulta, que nos tiene que devolver todos los documentos de la colección, para ello usamos la sintaxis:

```
db.inventory.find( {} )
```

Y nos devuelve el resultado esperado (todos los documentos) por lo que funciona correctamente:

```
{ "_id" : ObjectId("5f988b4a937b572817b95f58"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "Available" }
{ "_id" : ObjectId("5f988b4a937b572817b95f59"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "Available" }
{ "_id" : ObjectId("5f988b4a937b572817b95f5a"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "Unavailable" }
{ "_id" : ObjectId("5f988b4a937b572817b95f5b"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "Unavailable" }
{ "_id" : ObjectId("5f988b4a937b572817b95f5c"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "Obsolete" }
>>>
```

Consulta 2: Seguidamente realizamos una consulta que nos devuelva todos los documentos cuyo estado sea "available", para ello usamos la sintaxis:

```
db.inventory.find( { status: "Available" } )
```

Y nos devuelve el resultado esperado (solo hay 2 documentos con dicho estado):

```
{ "_id" : ObjectId("5f988b4a937b572817b95f58"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "Available" }
{ "_id" : ObjectId("5f988b4a937b572817b95f59"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "Available" }
>>>
```

Consulta 3: Ahora realizamos una consulta que nos devuelva todos los documentos cuyo estado sea “unavailable” o “obsolete”. Por tanto nos devolverá todos los documentos cuyo estado, de un modo u otro, indiquen que no está disponible (“available”) en el inventario. Para ello usamos la sintaxis:

```
db.inventory.find({ status: { $in: [ "Unavailable", "Obsolete" ] } })
```

Y nos devuelve el resultado esperado (solo hay 3 documentos que no estén “available”):

```
{ "_id" : ObjectId("5f988b4a937b572817b95f5a"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "Unavailable" }
{ "_id" : ObjectId("5f988b4a937b572817b95f5b"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "Unavailable" }
{ "_id" : ObjectId("5f988b4a937b572817b95f5c"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "Obsolete" }
>>> |
```

NOTA: en este caso usamos **\$in** (en vez de **\$or**), ya que estamos haciendo una consulta en el mismo campo (“status”)

Consulta 4: Ahora realizamos una consulta que nos devuelva todos los documentos cuyo estado sea “available” o la cantidad sea “menos que” (less than) 50. En este caso estamos haciendo una consulta en el distintos campos (“status”, “qty”), por eso usamos **\$or**. La sintaxis es:

```
db.inventory.find({ $or: [ { status: "Available" }, { qty: { $lt: 50 } } ] })
```

Y nos devuelve el resultado esperado (solo hay 3 documentos que cumplan esas condiciones):

```
>>> db.inventory.find({ $or: [ { status: "Available" }, { qty: { $lt: 50 } } ] })
{ "_id" : ObjectId("5f98947813786ca0099308d2"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "Available" }
{ "_id" : ObjectId("5f98947813786ca0099308d3"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "Available" }
{ "_id" : ObjectId("5f98947813786ca0099308d6"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "Obsolete" }
>>> |
```

Consulta 5: Realizamos una consulta que nos devuelva todos los documentos cuyo estado sea “available” y la cantidad sea “menos que” (less than) 50. Usamos esta sintaxis:

```
db.inventory.find({ status: "Available", qty: { $lt: 50 } })
```

Y nos devuelve el resultado esperado (solo hay 1 documento que cumpla esas condiciones):

```
{ "_id" : ObjectId("5f98947813786ca0099308d2"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "Available" }
>>> |
```

Consulta 6: Realizamos una consulta que nos devuelva todos los documentos cuyo estado sea “available” y, o bien la cantidad es “menos que” (less than) 30 o el nombre empieza por “p”. La sintaxis:

```
db.inventory.find({ status: "Available", $or: [ { qty: { $lt: 30 } }, { item: /^p/ } ] })
```

Y nos devuelve el resultado esperado (solo hay 1 documento que cumpla esas condiciones):

```
{ "_id" : ObjectId("5f98947813786ca0099308d2"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "Available" }
```