

Nociones teóricas ejercicio002

(Proyecto 1002_QUERY_DOCUMENTS)

Antonio Ávila Delgado

1º ASIR

DESCRIPCIÓN DEL PROYECTO

En este proyecto vamos a poblar con documentos una colección llamada INVENTARIO con información, de material de oficina (ej: producto, cantidad, medidas, etc). 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 INVENTARIO):

```
>>> db.inventario.insertMany([
...   { producto: "diario", cantidad: 25, medidas: { alto: 14, ancho: 21, unidad_medida: "cm" }, estado: "disponible" },
...   { producto: "cuaderno", cantidad: 50, medidas: { alto: 8.5, ancho: 11, unidad_medida: "pulgada" }, estado: "disponible" },
...   { producto: "papel", cantidad: 100, medidas: { alto: 8.5, ancho: 11, unidad_medida: "pulgada" }, estado: "pedido" },
...   { producto: "carpeta", cantidad: 75, medidas: { alto: 22.85, ancho: 39, unidad_medida: "cm" }, estado: "pedido" },
...   { producto: "carjeta", cantidad: 45, medidas: { alto: 10, ancho: 15.25, unidad_medida: "cm" }, estado: "obsoleto" },
...   { producto: "archivador", cantidad: 95, medidas: { alto: 8, ancho: 8.25, unidad_medida: "pulgada" }, estado: "obsoleto" }
... ]);
```

REALIZACIÓN DE CONSULTAS

Consulta 1: Vamos a usar el operador **\$eq** (igual a), para buscar todos documentos cuyo estado sea igual a "disponible":

```
db.inventario.find({ estado: { $eq: "disponible" } })
```

Y nos devuelve el resultado esperado por lo que funciona correctamente:

```
{ "_id" : ObjectId("5f99994ff66ffb3c2f662c7c"), "producto" : "diario", "cantidad" : 25, "medidas" : { "alto" : 14, "ancho" : 21, "unidad_medida" : "cm" }, "estado" : "disponible" }
{ "_id" : ObjectId("5f99994ff66ffb3c2f662c7d"), "producto" : "cuaderno", "cantidad" : 50, "medidas" : { "alto" : 8.5, "ancho" : 11, "unidad_medida" : "pulgada" }, "estado" : "disponible" }
{ "_id" : ObjectId("5f9999a16f66ffb3c2f662c82"), "producto" : "diario", "cantidad" : 25, "medidas" : { "alto" : 14, "ancho" : 21, "unidad_medida" : "cm" }, "estado" : "disponible" }
{ "_id" : ObjectId("5f9999a16f66ffb3c2f662c83"), "producto" : "cuaderno", "cantidad" : 50, "medidas" : { "alto" : 8.5, "ancho" : 11, "unidad_medida" : "pulgada" }, "estado" : "disponible" }
```

Consulta 2: Vamos a usar el operador **\$gt** (mayor que), para buscar todos documentos cuya cantidad sea mayor de 50. Nótese que al ser un valor numérico, no necesita comillas:

```
db.inventario.find({ cantidad: { $gt: 50 } })
```

Y nos devuelve el resultado esperado por lo que funciona correctamente:

```
{ "_id" : ObjectId("5f99d63476260b4309846cca"), "producto" : "papel", "cantidad" : 100, "medidas" : { "alto" : 8.5, "ancho" : 11, "unidad_medida" : "pulgada" }, "estado" : "pedido" }
{ "_id" : ObjectId("5f99d63476260b4309846ccb"), "producto" : "carpeta", "cantidad" : 75, "medidas" : { "alto" : 22.85, "ancho" : 39, "unidad_medida" : "cm" }, "estado" : "pedido" }
{ "_id" : ObjectId("5f99d63476260b4309846ccd"), "producto" : "archivador", "cantidad" : 95, "medidas" : { "alto" : 8, "ancho" : 8.25, "unidad_medida" : "pulgada" }, "estado" : "obsoleto" }
>>>
```

NOTA: los documentos con cantidad 50 no aparecen, ya que no cumplen la consulta.

Consulta 3: Vamos a usar el operador **\$gte** (*mayor o igual que*), para buscar todos documentos cuya cantidad sea mayor o igual que 50. Nótese que al ser también un valor numérico, no necesita comillas:

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

Y nos devuelve el resultado esperado por lo que funciona correctamente:

```
{ "_id" : ObjectId("5f99d63476260b4309846cc9"), "producto" : "cuaderno", "cantidad" : 50, "medidas" : { "alto" : 8.5, "ancho" : 11, "unidad_medida" : "pulgada" },
  "estado" : "disponible" }
{ "_id" : ObjectId("5f99d63476260b4309846cca"), "producto" : "papel", "cantidad" : 100, "medidas" : { "alto" : 8.5, "ancho" : 11, "unidad_medida" : "pulgada" },
  "estado" : "pedido" }
{ "_id" : ObjectId("5f99d63476260b4309846ccb"), "producto" : "carpeta", "cantidad" : 75, "medidas" : { "alto" : 22.85, "ancho" : 39, "unidad_medida" : "cm" }, "estado" : "pedido" }
{ "_id" : ObjectId("5f99d63476260b4309846ccd"), "producto" : "archivador", "cantidad" : 95, "medidas" : { "alto" : 8, "ancho" : 8.25, "unidad_medida" : "pulgada" },
  "estado" : "obsoleto" }
>>>
```

NOTA: los documentos con cantidad 50 sí aparecen esta vez, ya que cumplen la consulta.

Consulta 4: Vamos a usar el operador **\$lt** (*menor que*), para buscar todos documentos cuya cantidad sea menor de 45. Nótese que al ser un valor numérico, no necesita comillas:

```
db.inventario.find( { cantidad: { $lt: 45 } } )
```

Y nos devuelve el resultado esperado por lo que funciona correctamente:

```
{ "_id" : ObjectId("5f99db337e1c9c51b78af572"), "producto" : "diario", "cantidad" : 25, "medidas" : { "alto" : 14, "ancho" : 21, "unidad_medida" : "cm" }, "estado" : "disponible" }
>>>
```

NOTA: los documentos con cantidad 45 no aparecen, ya que no cumplen la consulta.

Consulta 5: Vamos a usar el operador **\$lte** (*menor o igual que*), para buscar todos documentos cuya cantidad sea menor o igual que 45. Nótese que al ser también un valor numérico, no necesita comillas:

```
db.inventario.find( { cantidad: { $lte: 45 } } )
```

Y nos devuelve el resultado esperado por lo que funciona correctamente:

```
{ "_id" : ObjectId("5f99db337e1c9c51b78af572"), "producto" : "diario", "cantidad" : 25, "medidas" : { "alto" : 14, "ancho" : 21, "unidad_medida" : "cm" }, "estado" : "disponible" }
{ "_id" : ObjectId("5f99db337e1c9c51b78af576"), "producto" : "tarjeta", "cantidad" : 45, "medidas" : { "alto" : 19, "ancho" : 15.25, "unidad_medida" : "cm" }, "estado" : "obsoleto" }
>>>
```

NOTA: los documentos con cantidad 45 sí aparecen esta vez, ya que cumplen la consulta.

Consulta 6: Vamos a usar el operador **\$ne** (*no igual a*), para buscar todos documentos cuyo estado no sea “obsoleto”. Nótese que al ser un valor alfanumérico, sí necesita comillas:

```
db.inventario.find( { estado: { $ne: "obsoleto" } } )
```

Y nos devuelve el resultado esperado por lo que funciona correctamente:

```
{ "_id": ObjectId("5f99db337e1c9c51b78af572"), "producto": "diario", "cantidad": 25, "medidas": { "alto": 14, "ancho": 21, "unidad_medida": "cm" }, "estado": "disponible" }
{ "_id": ObjectId("5f99db337e1c9c51b78af573"), "producto": "cuaderno", "cantidad": 50, "medidas": { "alto": 8.5, "ancho": 11, "unidad_medida": "pulgada" }, "estado": "disponible" }
{ "_id": ObjectId("5f99db337e1c9c51b78af574"), "producto": "papel", "cantidad": 100, "medidas": { "alto": 8.5, "ancho": 11, "unidad_medida": "pulgada" }, "estado": "pedido" }
{ "_id": ObjectId("5f99db337e1c9c51b78af575"), "producto": "carpeta", "cantidad": 75, "medidas": { "alto": 22.85, "ancho": 39, "unidad_medida": "cm" }, "estado": "pedido" }
>>>
```

Consulta 7: Vamos a usar el operador **\$and** para buscar todos documentos cuyo cantidad sea mayor o igual que 50 y además el estado es “obsoleto”. Nótese que el estado, al ser valor alfanumérico, sí necesita comillas, pero la cantidad no, al ser valor numérico.

```
db.inventario.find( { $and: [ { cantidad: { $gte: 50 } }, { estado: { $eq: "obsoleto" } } ] } )
```

Y nos devuelve el resultado esperado por lo que funciona correctamente:

```
{ "_id": ObjectId("5f99db337e1c9c51b78af577"), "producto": "archivador", "cantidad": 95, "medidas": { "alto": 8, "ancho": 8.25, "unidad_medida": "pulgada" }, "estado": "obsoleto" }
```

Consulta 8: Vamos a usar el operador **\$or** para buscar todos documentos cuyo cantidad sea mayor o igual que 50 o el estado sea “obsoleto”. Nótese que el estado, al ser valor alfanumérico, sí necesita comillas, pero la cantidad no, al ser valor numérico.

```
db.inventario.find( { $or: [ { cantidad: { $gte: 50 } }, { estado: { $eq: "obsoleto" } } ] } )
```

Y nos devuelve el resultado esperado por lo que funciona correctamente:

```
{ "_id": ObjectId("5f99db337e1c9c51b78af573"), "producto": "cuaderno", "cantidad": 50, "medidas": { "alto": 8.5, "ancho": 11, "unidad_medida": "pulgada" }, "estado": "disponible" }
{ "_id": ObjectId("5f99db337e1c9c51b78af574"), "producto": "papel", "cantidad": 100, "medidas": { "alto": 8.5, "ancho": 11, "unidad_medida": "pulgada" }, "estado": "pedido" }
{ "_id": ObjectId("5f99db337e1c9c51b78af575"), "producto": "carpeta", "cantidad": 75, "medidas": { "alto": 22.85, "ancho": 39, "unidad_medida": "cm" }, "estado": "pedido" }
{ "_id": ObjectId("5f99db337e1c9c51b78af576"), "producto": "tarjeta", "cantidad": 45, "medidas": { "alto": 10, "ancho": 15.25, "unidad_medida": "cm" }, "estado": "obsoleto" }
{ "_id": ObjectId("5f99db337e1c9c51b78af577"), "producto": "archivador", "cantidad": 95, "medidas": { "alto": 8, "ancho": 8.25, "unidad_medida": "pulgada" }, "estado": "obsoleto" }
>>>
```

Consulta 9: Vamos a usar el operador **\$nor** para buscar todos documentos cuyo cantidad ni sea mayor o igual que 50 o ni el estado sea “obsoleto”. Nótese que el estado, al ser valor alfanumérico, sí necesita comillas, pero la cantidad no, al ser valor numérico.

```
db.inventario.find( [ { $nor: [ { cantidad: { $gte: 50 } }, { estado: { $eq: "obsoleto" } } ] } ] )
```

Y nos devuelve el resultado esperado por lo que funciona correctamente:

```
{ "_id" : ObjectId("5f99db337a1c9c51b78af972"), "producto" : "diario", "cantidad" : 25, "medidas" : { "alto" : 14, "ancho" : 21, "unidad_medida" : "cm" }, "estado" : "disponible" }
>>>
```