

# MongoDB

## **Tercera Práctica:**

- 1) Explicación General de la práctica.
- 2) Recursos utilizados en la práctica.
- 3) Capturas de pantalla mostrando que las Querys han sido bien realizadas.

### **1) Explicación General**

En esta práctica hemos profundizado en las Querys de MongoDB, especialmente en los operadores lógicos y de comparación.

Para realizarla, hemos importado a nuestra base de datos una serie de documentos que vienen en el siguiente enlace (A los cuales nosotros mismos hemos añadido más documentos).

[Enlace donde están los documentos de la base de datos 'inventory'](#)

Tras haber añadido los documentos hemos realizado una serie de Querys utilizando operadores lógicos y de comparación como exigía el enunciado. En los siguientes links está toda la información acerca de su uso de forma general.

[Enlace a los operadores de comparación](#)

[Enlace a los operadores lógicos](#)

## 2) Recursos utilizados en la práctica:

En primera instancia para importar los documentos hemos utilizado la siguiente función:

- **insertMany()**: Nos permite importar una serie de documentos a la vez.

Para las Querys hemos utilizado tanto operadores lógicos como de comparación:

- **Operadores de Comparación:**
  - **\$lt**: Selecciona los documentos para los cuales el campo es menor al valor indicado.
  - **\$gte**: Selecciona los documentos para los cuales el campo es mayor al valor indicado.
  - **\$in**: Selecciona los documentos para los cuales el campo es igual a los valores indicados.
  - **\$nin**: Selecciona los documentos para los cuales el campo no es igual al valor indicado.
  - **Los no empleados**: No hemos utilizado operadores como **\$el** que selecciona los documentos para los que el campo es igual al valor, **\$gt** el cual es exactamente igual que el **\$gte** solo que no funciona en caso de igualdad, **\$gt** el cual es exactamente igual que el **\$lt** solo que funciona también en caso de igualdad, y el **\$ne** el cual es como el **\$nin** solo que funciona para valores únicos, no para listas.
- **Operadores Lógicos:**
  - **\$and**: Devuelve todos los documentos **que cumplan todas** las condiciones que se establezcan.
  - **\$not**: Devuelve todos los documentos **que no cumplan todas** las condiciones que se establezcan.
  - **\$or**: Devuelve todos los documentos **que cumplan alguna** de las condiciones que se establezcan.
  - **\$nor**: Devuelve todos los documentos **que no cumplan alguna** de las condiciones que se establezcan.
- **Otros recursos:**
  - **And implícito**: Cuando realizamos una query no es obligatorio utilizar un operador **\$and** ya que automáticamente al utilizar una coma se nos buscarán los documentos que obligatoriamente cumplan con todas las condiciones introducidas.
  - **Búsqueda de documentos anidados**: Cuando hemos introducido un documento el cual un campo viene dado por una serie de valores en forma de query (por ejemplo el campo "size" de nuestra base de datos), se tendrá que utilizar una sintaxis especial para realizar la Query como viene en la captura de la Tercera Query.
  - **Búsqueda de documento mediante la búsqueda de un campo tipo string a partir de un carácter**: Para seleccionar una serie de documentos mediante la búsqueda por caracteres de un campo tipo string deberemos utilizar una sintaxis especial como viene en la captura de la Quinta query.

### 3) Capturas de pantalla mostrando las queries:

Primera Query (uso de \$or, \$lt, \$in):

```
> db.inventory.find( { $or: [ { qty: { $lt: 100 } }, { status: { $in: ["B","C"] } } ] } )
{ "_id" : ObjectId("5f9b15e433449235952fbbbc"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5f9b15e433449235952fbbbd"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5f9b15e433449235952fbbbf"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5f9b15e433449235952fbbc0"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5f9b172533449235952fbbc1"), "item" : "pencil", "qty" : 250, "size" : { "h" : 1, "w" : 4, "uom" : "cm" }, "status" : "B" }
{ "_id" : ObjectId("5f9b172533449235952fbbc2"), "item" : "carpet", "qty" : 5, "size" : { "h" : 10, "w" : 12, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5f9b172533449235952fbbc3"), "item" : "rubber", "qty" : 500, "size" : { "h" : 3, "w" : 2, "uom" : "cm" }, "status" : "C" }
{ "_id" : ObjectId("5f9b172533449235952fbbc4"), "item" : "tipex", "qty" : 12, "size" : { "h" : 1, "w" : 4, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5f9b172533449235952fbbc5"), "item" : "keyboard", "qty" : 5, "size" : { "h" : 5, "w" : 40, "uom" : "cm" }, "status" : "B" }
> []
```

Segunda Query (uso de \$and, \$gte):

```
> db.inventory.find( { $and: [ { qty: { $gte: 50 } }, { status: "A" } ] } )
{ "_id" : ObjectId("5f9b15e433449235952fbbbd"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
> []
```

Tercera Query (Uso de \$nin y búsqueda de documentos anidados):

```
> db.inventory.find( { $and: [ { qty: { $nin: [250,100] } }, { 'size.h': { $lt: 5 } } ] } )
{ "_id" : ObjectId("5f9b172533449235952fbbc3"), "item" : "rubber", "qty" : 500, "size" : { "h" : 3, "w" : 2, "uom" : "cm" }, "status" : "C" }
{ "_id" : ObjectId("5f9b172533449235952fbbc4"), "item" : "tipex", "qty" : 12, "size" : { "h" : 1, "w" : 4, "uom" : "cm" }, "status" : "A" }
> []
```

Cuarta Query (Uso del and implícito):

```
> db.inventory.find({status:"B", qty:250})
{ "_id" : ObjectId("5f9b172533449235952fbbc1"), "item" : "pencil", "qty" : 250, "size" : { "h" : 1, "w" : 4, "uom" : "cm" }, "status" : "B" }
> []
```

Quinta query (Búsqueda de documento mediante la búsqueda de un campo tipo string y uso de \$not):

```
> db.inventory.find({ item: { $not: /^p.*/ } })
{ "_id" : ObjectId("5f9b15e433449235952fbbbc"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5f9b15e433449235952fbbbd"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5f9b172533449235952fbbc2"), "item" : "carpet", "qty" : 5, "size" : { "h" : 10, "w" : 12, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5f9b172533449235952fbbc3"), "item" : "rubber", "qty" : 500, "size" : { "h" : 3, "w" : 2, "uom" : "cm" }, "status" : "C" }
{ "_id" : ObjectId("5f9b172533449235952fbbc4"), "item" : "tipex", "qty" : 12, "size" : { "h" : 1, "w" : 4, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5f9b172533449235952fbbc5"), "item" : "keyboard", "qty" : 5, "size" : { "h" : 5, "w" : 40, "uom" : "cm" }, "status" : "B" }
> []
```

Sexta Query (Uso del \$not)

```
> db.inventory.find({ item: { $not: /^p.*/ } })
{ "_id" : ObjectId("5f9b15e433449235952fbbbc"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5f9b15e433449235952fbbbd"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5f9b172533449235952fbbc2"), "item" : "carpet", "qty" : 5, "size" : { "h" : 10, "w" : 12, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5f9b172533449235952fbbc3"), "item" : "rubber", "qty" : 500, "size" : { "h" : 3, "w" : 2, "uom" : "cm" }, "status" : "C" }
{ "_id" : ObjectId("5f9b172533449235952fbbc4"), "item" : "tipex", "qty" : 12, "size" : { "h" : 1, "w" : 4, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5f9b172533449235952fbbc5"), "item" : "keyboard", "qty" : 5, "size" : { "h" : 5, "w" : 40, "uom" : "cm" }, "status" : "B" }
> []
```