

MongoDB. Prácticas

Cómo ejecutar MongoDB en los laboratorios:

- Arranca el servidor. Está en 'Todos los programas/BBDD/MongoDB/Servidor MongoDB'.
- Arranca el cliente. Está en 'C:/software/bbdd/MongoDB/bin/mongo'.

Dos instrucciones útiles:

- `use mi_bbdd`: crea o abre *mi_bbdd*.
- `db.createCollection('mi_colección')`: crea *mi_colección*.

Práctica 1. Diseña una base de datos en MongoDB para representar la información de un nuevo sistema solar. Cuestiones:

1. Crea la base de datos **astronomia** y la colección **planetas**. Inserta, además, la siguiente información:

Id	Nombre	Masa	Volumen	Composición	Perihelio	Afelio
1	P111	1.9	3	H	0.5	
2	P222	3.1	4.5	O	0.07	1.7
3	S111	0.2				
4	S222	0.3	1.1			
5	S333		0.02			
6	C111	0.05		H	0.01	25
7	P333	3.8	4.9	He		

S111, *S222* y *S333* son satélites de *P222*. Añade al documento que contenga la información de *P222* una clave, llamada *satelites*, cuyo valor sea un array con la información de cada uno de los satélites.

2. Muestra por pantalla únicamente los nombres de aquellos objetos que tengan en la composición *H* o *N* o bien su masa sea mayor que 3.2. La consulta, que solo se realizará una vez, debe ser eficiente.
3. Actualiza el nombre de *C111* a *Z111*.

Nota: *perihelio* y *afelio* son la distancia más corta y más larga en una órbita, respectivamente, a una estrella.

Soluciones:

```
use astronomia
db.createCollection("planetas")
db.planetas.insert({"nombre":"p111","masa":1.9,"volumen":3,
  composicion:"H","perihelio":0.5})
db.planetas.insert({"nombre":"p222","masa":3.1,"volumen":3.5,
  composicion:"O","perihelio":0.07,"afelio":1.7})
db.planetas.insert({"nombre":"c111","masa":0.05,"composicion":"H","perihelio":0.5,"afelio":25})
db.planetas.update({"nombre":"p222"},
  {"$push":{"satelites":{"nombre":"s111","masa":0.2}}})
db.planetas.update({"nombre":"p222"},
  {"$push":{"satelites":{"nombre":"s222","masa":0.3,"volumen":1.1}}})
db.planetas.update({"nombre":"p222"},
  {"$push":{"satelites":{"nombre":"s333","volumen":0.02}}})
db.planetas.insert({"nombre":"p333","masa":3.8,"volumen":3.9,
  composicion:"He","perihelio":0.07,"afelio":1.7})

db.planetas.find({"$or":[{"composicion":{"$in":["H","N"]}],{"masa":{"$gt":3.2}}},
  {"nombre":1,"_id":0})

db.planetas.update({"nombre":"c111"},{"$set":{"nombre":"z111"}})
```

Práctica 2. Diseña una base de datos en MongoDB que gestione la siguiente información sobre restaurantes: nombre, tipo de comida, número de visitas a la web y valoraciones. Diseña las sentencias que permiten implementar el siguiente comportamiento:

1. Crea una base de datos llamada *restaurantes* y una colección llamada *catalogo* con los siguientes valores para los campos que se especifican:

codigo	nombre	tipo	visitas
1	El meson	mediterranea	250
2	La patata loca	mediterranea	50
3	Yakuza	japones	150
4	Todo pizza	italiano	350

2. Muestra todos los restaurantes (solo nombre y tipo) cuyas visitas sean más de 200.
3. Incrementa las visitas del restaurante con valor de código igual a 1 en 25 unidades.
4. Añade a los restaurantes de tipo 'italiano' el campo especialidad con valor 'pizza romana'.
5. Añade las siguientes valoraciones según su código:

Código	Valoraciones
1	3
2	3,4,5
3	4,5,5
4	1,5

Solamente se puede realizar una instrucción por código de restaurante, es decir, se tienen que añadir todas las valoraciones en una única instrucción.

6. Con una única instrucción elimina todas las valoraciones iguales a 5 del restaurante con código igual a 3.

Solución:

```
use restaurantes
db.createCollection("catalogo")
db.catalogo.insert({"codigo":1,"nombre":"El_meson","tipo":"mediterranea","visitas":250})
db.catalogo.insert({"codigo":2,"nombre":"La_patata_loca","tipo":"mediterranea","visitas":50})
db.catalogo.insert({"codigo":3,"nombre":"Yakuza","tipo":"japones","visitas":150})
db.catalogo.insert({"codigo":4,"nombre":"Todo_pizza","tipo":"italiano","visitas":350})
```

```
db.catalogo.find({"visitas":{"$gt":200}},{"nombre":1,"tipo":1,"_id":0})
```

```
db.catalogo.update({"codigo":1},{"$inc":{"visitas":25}})
```

```
db.catalogo.update({"tipo":"italiano"},{$set:{"especialidad":"pizza_romana"}})
```

```
db.catalogo.update({"codigo":1},{"$push":{"valoraciones":3}})
db.catalogo.update({"codigo":2},{"$push":{"valoraciones":{"$each":[3,4,5]}}})
db.catalogo.update({"codigo":3},{"$push":{"valoraciones":{"$each":[4,5,5]}}})
db.catalogo.update({"codigo":4},{"$push":{"valoraciones":{"$each":[1,5]}}})
```

```
db.catalogo.update({"codigo":3},{ "$pull":{"valoraciones":5}})
```

Práctica 3. Para la BD adjunta, referida a los Pedidos a un distribuidor de electrónica y componentes informáticos, realiza las siguientes operaciones.

- Indica qué operaciones es necesario realizar para crear una base de datos y un colección (llamada *Pedidos*) que contenga exactamente los datos de la BD adjunta.
- Diseña una consulta en mongoDB que devuelva por orden alfabético los nombres de los clientes que han realizado un pedido que tenga componentes de los años 2019, 2020 o 2021, estando el importe total del pedido entre 400 y 500 euros. El importe total también se ha de mostrar.
- Modifica los valores de detalles del segundo componente del ensamblado para el cliente "PCfine", tomando ahora los nuevos: fabricante: "AMD", país: "USA", año: "2022".
- Para el cliente "PCfine" y con una sola instrucción, elimina completamente el primero de los componentes listados.
- Para el cliente "Info2000" y con una sola instrucción, añade dos nuevos componentes con todos sus elementos y con los siguientes valores:

```
"componente" : "USBHub",
"detalles"   : {"fabricante":"INTEL","país":"UK","año":2022},
"precio"     : 50

"componente" : "HDMI",
"detalles"   : {"fabricante":"Connect","país":"Finland","año":2018},
"precio"     : 35
```

Base de datos adjunta:

```
* 1 */
{
  "ID_Pedido" : 1,
  "Nombre_Cliente" : "PCfine",
  "Importe_Total" : 405,
  "Ensamblado" : [
    {
      "componente" : "pb_2000",
      "detalles" : {
        "fabricante" : "AJT",
        "país" : "Taiwan",
        "año" : 2019
      },
      "precio" : 250
    },
    {
      "componente" : "cpu",
      "detalles" : {
        "fabricante" : "MicroL",
        "país" : "USA",
        "año" : 2021
      },
      "precio" : 155
    }
  ]
}
```

```

    ]
}

/* 2 */
{
    "ID_Pedido" : 2,
    "Nombre_Cliente" : "SistemasTotal",
    "Importe_Total" : 195,
    "Ensamblado" : [
        {
            "componente" : "hdd",
            "detalles" : {
                "fabricante" : "SingaHD",
                "país" : "Singapur",
                "año" : 2021
            },
            "precio" : 195
        }
    ]
}

/* 3 */
{
    "ID_Pedido" : 3,
    "Nombre_Cliente" : "Info2000",
    "Importe_Total" : 400,
    "Ensamblado" : [
        {
            "componente" : "GPU",
            "detalles" : {
                "fabricante" : "Nvidia",
                "país" : "USA",
                "año" : 2020
            },
            "precio" : 400
        }
    ]
}

```

Solución:

```

db.getCollection("Pedidos").find({"Ensamblado.detalles.año":
{"$in":[2020,2019,2021]}, "Importe_Total":
{"$gte": 400, "$lte":500}}, {"Nombre_Cliente":1, "Importe_Total": 1, _id:0}).sort({"Nombre_Cliente":1})

```

```

db.getCollection("Pedidos").update({"Nombre_Cliente":"PCfine"},
{"$set": {"Ensamblado.1":{"componente":"cpu", "detalles":
{"fabricante":"AMD", "pais":"USA", "año":2022}, "precio":"155"}}}
)

```

```

db.getCollection("Pedidos").update({"Nombre_Cliente":"PCfine"},
{"$pop": {"Ensamblado":-1}})

```

```
db.getCollection("Pedidos").update({"Nombre_Cliente":"Info2000"},
{"$push":{"Ensamblado":{"$each":[{"componente":"USBHub",
"detalles":{"fabricante":"INTEL","país":"UK","año":2022},
"precio":50},
{"componente":"HDMI",
"detalles":{"fabricante":"Connect","país":"Finland","año":2018},
"precio":35}
]}
})
```