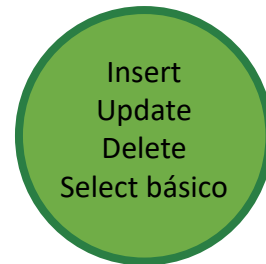




Operaciones básicas en mongoDB



- **MongoDB** es una base de datos **noSQL** de tipo documental.
- Su nombre viene de la palabra inglesa "humongous" que significa enorme
- Almacena la información en documentos tipo BSON (JSON Binario).
- Carece de esquema predefinido.

En mongoDB una **base de datos** es una **colección de documentos**.

Estos documentos se componen de campos/características ... de distintos tipos de datos.

```
{
  Nombre      : "Departamento de Lenguajes y Sistemas Informáticos",
  Empleados: 70,
  Asignaturas: [ "Diseño de Bases de Datos", "Fundamentos de Bases de Datos", "Programación", "Gestión de la Información" ],
  Dirección: {
    calle: "Carretera de San Vicente",
    número: 2,
    codigopostal: 03690
  },
  Teléfono: {
    centralUA: { número: 965903400,
                  extensión: 3972
                },
    directo: [965903466, 965943211]
  }
}
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

MongoDB, a través de JSON, puede utilizar los siguientes tipos de datos:

- String: guardados en UTF-8. Van siempre entre comillas dobles.
- Number: números. Al guardarse en BSON pueden ser de tipo byte, int32, int64 o double.
- Boolean: con valor true o false.
- Array: van entre corchetes [] y pueden contener de 1 a N elementos, que pueden ser de cualquiera de los otros tipos.
- Documentos: un documento en formato JSON puede contener otros documentos embebidos que incluyan más documentos o cualquiera de los tipos anteriormente descritos.
- Null.

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

Trabajaremos de modo local. Para empezar a trabajar



1.- Iniciamos el servidor
mongod



2.- Abrimos un terminal
mongo

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

```
> show dbs
admin      0.000GB
comercio   0.000GB
config     0.000GB
local      0.000GB
profesores 0.000GB
practicasyDBD 0.000GB
```

Muestra las bases de datos creadas

```
> use comercio
switched to db comercio
```

Indica la base de datos a utilizar.

```
> Show collections
clientes
```

Muestra las colecciones creadas.

```
> db
comercio
```

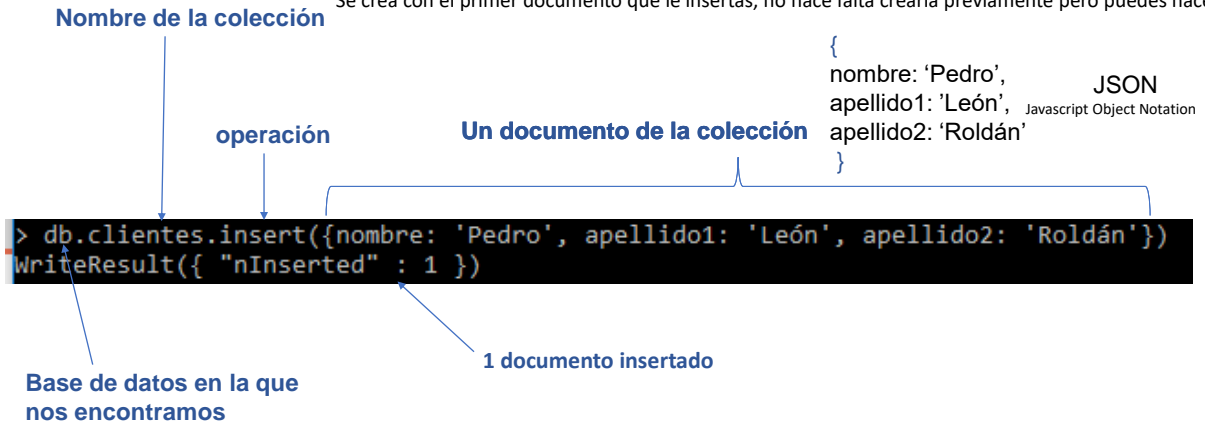
Muestra la base de datos en la que te encuentras.

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

INSERT**.insert**

La inserción de datos se realiza a través de colecciones.

Se crea con el primer documento que le insertas, no hace falta crearla previamente pero puedes hacerlo



DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

SELECT**.find**

Para ver la información almacenada en una colección

```
> db.clientes.find()
{ "_id" : ObjectId("5c0275f8f87ed69a71545d14"), "nombre" : "Pedro", "apellido1" : "León", "apellido2" : "Roldán" }
```

_id: Identificador único asignado por MongoDB si no se lo asignamos

Para ver la información almacenada en una colección en un formato más legible

```
> db.clientes.find().pretty()
{
  "_id" : ObjectId("5c0275f8f87ed69a71545d14"),
  "nombre" : "Pedro",
  "apellido1" : "León",
  "apellido2" : "Roldán"
}
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

INSERT**.insert**

Para insertar **varios** clientes de golpe, se utiliza un **array** de clientes, []

```
> db.clientes.insert([
  {nombre: 'Laura', apellido1: 'Rodríguez', apellido2: 'Sanz'},
  {nombre: 'Andrea', apellido1: 'Lara', apellido2: 'Sempere'},
  {nombre: 'Miguel', apellido1: 'Cobos', apellido2: 'Pascual'},
  {nombre: 'Manuel', apellido1: 'Beltrán', apellido2: 'Sanz'}
])

BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 4,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
```

Si quisiésemos asignar el
identificador(único)
> db.clientes.insert(
{_id:1, nombre: 'Laura', apellido1:
'Rodríguez', apellido2: 'Sanz'},

→ 4 inserciones

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

SELECT**.find**

Vemos los clientes insertados

```
> db.clientes.find()
{ "_id" : ObjectId("5c0275f8f87ed69a71545d14"), "nombre" : "Pedro", "apellido1" : "León", "apellido2" : "Roldán" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Laura", "apellido1" : "Rodríguez", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec859"), "nombre" : "Andrea", "apellido1" : "Lara", "apellido2" : "Sempere" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85a"), "nombre" : "Miguel", "apellido1" : "Cobos", "apellido2" : "Pascual" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85b"), "nombre" : "Manuel", "apellido1" : "Beltrán", "apellido2" : "Sanz" }
>
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

SELECT

.find

```
> db.clientes.find().pretty()
{
  "_id" : ObjectId("5c0275f8f87ed69a71545d14"),
  "nombre" : "Pedro",
  "apellido1" : "León",
  "apellido2" : "Roldán"
}
{
  "_id" : ObjectId("5c02c396735bd96b3e9ec858"),
  "nombre" : "Laura",
  "apellido1" : "Rodríguez",
  "apellido2" : "Sanz"
}
{
  "_id" : ObjectId("5c02c396735bd96b3e9ec859"),
  "nombre" : "Andrea",
  "apellido1" : "Lara",
  "apellido2" : "Sempere"
}
{
  "_id" : ObjectId("5c02c396735bd96b3e9ec85a"),
  "nombre" : "Miguel",
  "apellido1" : "Cobos",
  "apellido2" : "Pascual"
}
{
  "_id" : ObjectId("5c02c396735bd96b3e9ec85b"),
  "nombre" : "Manuel",
  "apellido1" : "Beltrán",
  "apellido2" : "Sanz"
}
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

INSERT

.insert

No es necesario que todos los clientes tengan los mismos campos.

```
> db.clientes.insert({nombre: 'Rosa', apellido1: 'Rodríguez', apellido2: 'Sanz', sexo:'mujer'});
WriteResult({ "nInserted" : 1 })
```

```
> db.clientes.find()
{ "_id" : ObjectId("5c0275f8f87ed69a71545d14"), "nombre" : "Pedro", "apellido1" : "León", "apellido2" : "Roldán" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Laura", "apellido1" : "Rodríguez", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec859"), "nombre" : "Andrea", "apellido1" : "Lara", "apellido2" : "Sempere" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85a"), "nombre" : "Miguel", "apellido1" : "Cobos", "apellido2" : "Pascual" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85b"), "nombre" : "Manuel", "apellido1" : "Beltrán", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez", "apellido2" : "Sanz", "sexo" : "mujer" }
>
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

SELECT

.find

Si queremos buscar los clientes que tengan un valor en una de las características `find({característica:valor})`

```
> db.clientes.find({apellido2:'Sanz'})
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Laura", "apellido1" : "Rodríguez", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85b"), "nombre" : "Manuel", "apellido1" : "Beltrán", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez", "apellido2" : "Sanz",
"sexo" : "mujer" }
```

¿Y si nos equivocamos al poner el nombre de una característica al insertar? `apellido2 apellido2`

```
> db.clientes.insert({nombre: 'Sergio', apellido1: 'Valiente', apellido2: 'Sanz'});
WriteResult({ "nInserted" : 1 })
```

No aparece ya que `apellido2 <> apellido2`

```
> db.clientes.find({apellido2:'Sanz'})
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Laura", "apellido1" : "Rodríguez", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85b"), "nombre" : "Manuel", "apellido1" : "Beltrán", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez", "apellido2" : "Sanz",
"sexo" : "mujer" }
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

SELECT

.find

```
> db.clientes.find()
{ "_id" : ObjectId("5c0275f8f87ed69a71545d14"), "nombre" : "Pedro", "apellido1" : "León", "apellido2" : "Roldán" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Laura", "apellido1" : "Rodríguez", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec859"), "nombre" : "Andrea", "apellido1" : "Lara", "apellido2" : "Sempere" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85a"), "nombre" : "Miguel", "apellido1" : "Cobos", "apellido2" : "Pascual" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85b"), "nombre" : "Manuel", "apellido1" : "Beltrán", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez", "apellido2" : "Sanz", "sexo" : "mujer" }
>
```

Búsqueda por identificador

```
> db.clientes.find({_id:'5c0275f8f87ed69a71545d14'})
>
```

Hay que utilizar la función `ObjectId()`

```
> db.clientes.find({_id:ObjectId('5c0275f8f87ed69a71545d14')})
{ "_id" : ObjectId("5c0275f8f87ed69a71545d14"), "nombre" : "Pedro", "apellido1" : "León", "apellido2" : "Roldán" }
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

UPDATE

.update

```

> db.clientes.update(
... {apellido2:'Sanz'},
... {
... nombre:'Sergio',
... apellido1:'Valiente',
... apellido2:'Sanz'
... }
... )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

```

Características que debe cumplir el documento a modificar

Hemos cambiado el nombre de la característica `apellido2` por `apellido2`

Nuevo contenido del documento

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

UPDATE

.update

```

> db.clientes.find()
{ "_id" : ObjectId("5c0275f8f87ed69a71545d14"), "nombre" : "Pedro", "apellido1" : "León", "apellido2" : "Roldán" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Laura", "apellido1" : "Rodríguez", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec859"), "nombre" : "Andrea", "apellido1" : "Lara", "apellido2" : "Sempere" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85a"), "nombre" : "Miguel", "apellido1" : "Cobos", "apellido2" : "Pascual" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85b"), "nombre" : "Manuel", "apellido1" : "Beltrán", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez", "apellido2" : "Sanz", "sexo" : "mujer" }
>

```

```

> db.clientes.update({apellido1:'Rodríguez', apellido2:'Sanz'},
{nombre:'Lara', apellido1:'Rodríguez', apellido2:'Sanz', sexo:'mujer'})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

```

O bien

```

> db.clientes.update({_id: ObjectId('5c02c396735bd96b3e9ec858')},
{nombre:'Lara', apellido1:'Rodríguez', apellido2:'Sanz', sexo:'mujer'})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

UPDATE

.update

Cuidado

```
> db.clientes.insert({nombre: 'Mario', apellido1: 'Valiente', apellido2: 'xxx'});
WriteResult({ "nInserted" : 1 })
```

```
> db.clientes.update( {apellido2:'xxx'}, {apellido2:'Oncina'} )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
>db.clientes.find()
{ "_id" : ObjectId("5c0275f8f87ed69a71545d14"), "nombre" : "Pedro", "apellido1" : "León", "apellido2" : "Roldán" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Laura", "apellido1" : "Rodríguez", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec859"), "nombre" : "Andrea", "apellido1" : "Lara", "apellido2" : "Sempere" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85a"), "nombre" : "Miguel", "apellido1" : "Cobos", "apellido2" : "Pascual" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85b"), "nombre" : "Manuel", "apellido1" : "Beltrán", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez", "apellido2" : "Sanz", "sexo" : "mujer" }
{ "_id" : ObjectId("5c02cbd7735bd96b3e9ec85d"), "nombre" : "Sergio", "apellido1" : "Valiente", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02d0e1735bd96b3e9ec85e"), "apellido2" : "Oncina" }
>
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

UPDATE

.update

¿Hay que repetir la información que no se modifica? NO

\$set

```
> db.clientes.update(
... { _id:ObjectId("5c02c396735bd96b3e9ec858")},
... {
... $set:{edad:30}
... }
... )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

```
> db.clientes.find()
{ "_id" : ObjectId("5c0275f8f87ed69a71545d14"), "nombre" : "Pedro", "apellido1" : "León", "apellido2" : "Roldán" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Laura", "apellido1" : "Rodríguez", "apellido2" : "Sanz", "sexo" : "mujer", "edad" : 30 }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec859"), "nombre" : "Andrea", "apellido1" : "Lara", "apellido2" : "Sempere" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85a"), "nombre" : "Miguel", "apellido1" : "Cobos", "apellido2" : "Pascual" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85b"), "nombre" : "Manuel", "apellido1" : "Beltrán", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez", "apellido2" : "Sanz", "sexo" : "mujer" }
{ "_id" : ObjectId("5c02cbd7735bd96b3e9ec85d"), "nombre" : "Sergio", "apellido1" : "Valiente", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02d0e1735bd96b3e9ec85e"), "apellido2" : "Oncina" }
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

UPDATE

.update

¿Hay que repetir la información que no se modifica? NO

\$set

```
> db.clientes.insert({nombre: 'Pilar', apellido1: 'Valiente', apellido2: 'xxx'});
WriteResult({ "nInserted" : 1 })
```

```
> db.clientes.find()
{ "_id" : ObjectId("5c0275f8f87ed69a71545d14"), "nombre" : "Pedro", "apellido1" : "León", "apellido2" : "Roldán" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Lara", "apellido1" : "Rodríguez", "apellido2" : "Sanz", "sexo" : "mujer", "edad" : 30 }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec859"), "nombre" : "Andrea", "apellido1" : "Lara", "apellido2" : "Sempere" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85a"), "nombre" : "Miguel", "apellido1" : "Cobos", "apellido2" : "Pascual" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85b"), "nombre" : "Manuel", "apellido1" : "Beltrán", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez", "apellido2" : "Sanz", "sexo" : "mujer" }
{ "_id" : ObjectId("5c02cbd7735bd96b3e9ec85d"), "nombre" : "Sergio", "apellido1" : "Valiente", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02d0e1735bd96b3e9ec85e"), "apellido2" : "Oncina" }
{ "_id" : ObjectId("5c02e27a735bd96b3e9ec85f"), "nombre" : "Pilar", "apellido1" : "Valiente", "apellido2" : "xxx" }
```

```
> db.clientes.update(
... { _id:ObjectId('5c02e27a735bd96b3e9ec85f')},
... {
... $set:{apellido2:'Ruiz',edad:30}
... }
... )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.clientes.find({_id:ObjectId('5c02e27a735bd96b3e9ec85f')})
{ "_id" : ObjectId("5c02e27a735bd96b3e9ec85f"), "nombre" : "Pilar", "apellido1" : "Valiente", "apellido2" :
"Ruiz", "edad" : 30 }
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

UPDATE

.update

```
> db.clientes.find({apellido1:'Rodríguez'})
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Lara", "apellido1" : "Rodríguez",
"apellido2" : "Sanz", "sexo" : "mujer", "edad" : 30 }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez",
"apellido2" : "Sanz", "sexo" : "mujer" }
>
```

```
> db.clientes.update(
... {apellido1:'Rodríguez'},
... {
... $inc:{edad:7}
... }
... )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.clientes.find({apellido1:'Rodríguez'})
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Lara", "apellido1" : "Rodríguez",
"apellido2" : "Sanz", "sexo" : "mujer", "edad" : 37 }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez",
"apellido2" : "Sanz", "sexo" : "mujer" }
>
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

```
> db.clientes.update(
... {apellido1:'Rodríguez'},
... {
...   $inc:{edad:-2}
... }
... )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.clientes.find({apellido1:'Rodríguez'}).pretty()
{
  "_id" : ObjectId("5c02c396735bd96b3e9ec858"),
  "nombre" : "Lara",
  "apellido1" : "Rodríguez",
  "apellido2" : "Sanz",
  "sexo" : "mujer",
  "edad" : 35
}
{
  "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"),
  "nombre" : "Rosa",
  "apellido1" : "Rodríguez",
  "apellido2" : "Sanz",
  "sexo" : "mujer"
}
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

```
> db.clientes.update(
{nombre:'Rosa',apellido1:'Rodríguez'},
{
  $unset:{sexo:0}
}
)
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.clientes.find({apellido1:'Rodríguez'}).pretty()
{
  "_id" : ObjectId("5c02c396735bd96b3e9ec858"),
  "nombre" : "Lara",
  "apellido1" : "Rodríguez",
  "apellido2" : "Sanz",
  "sexo" : "mujer",
  "edad" : 35
}
{
  "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"),
  "nombre" : "Rosa",
  "apellido1" : "Rodríguez",
  "apellido2" : "Sanz"
}
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

UPDATE

.update

```
> db.clientes.update(
... {nombre:'Lara'},
... {
... $set:{apellido1:'Ramírez'},
... $unset:{apellido2:0},
... $inc:{edad:3}
... }
... )
```

```
> db.clientes.find({nombre:'Lara'}).pretty()
{
  "_id" : ObjectId("5c02c396735bd96b3e9ec858"),
  "nombre" : "Lara",
  "apellido1" : "Ramírez",
  "sexo" : "mujer",
  "edad" : 38
}
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

```
> db.cliente.update(
... { nombre:'Samuel'},
... {
... nombre:'Samuel',
... apellido1:'Carrasco',
... apellido2:'Carratalá'
... }
... )
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
```

```
> db.cliente.update(
... {nombre:'Samuel'},
... {
... nombre:'Samuel',
... apellido1:'Carrasco',
... apellido2:'Carratalá'
... },
... {upsert: true}
... )
WriteResult({
  "nMatched" : 0,
  "nUpserted" : 1,
  "nModified" : 0,
  "_id" : ObjectId("5c02f5c9b61a0cb0edb5781c")
})
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

UPDATE

.update

UPDATE

.update

```
> db.clientes.update( {}, { $rename:{"sexo":"genero"} } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
```

```
> db.clientes.update( {nombre:"Lara"}, { $rename:{"genero":"sexo","apellido1":"apel1"} } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.clientes.find()
{ "_id" : ObjectId("5c0275f8f87ed69a71545d14"), "nombre" : "Pedro", "apellido1" : "León", "apellido2" : "Roldán" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Lara", "edad" : 38, "apel1" : "Ramírez", "sexo" : "mujer" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec859"), "nombre" : "Andrea", "apellido1" : "Lara", "apellido2" : "Sempere" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85a"), "nombre" : "Miguel", "apellido1" : "Cobos", "apellido2" : "Pascual" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85b"), "nombre" : "Manuel", "apellido1" : "Beltrán", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02cbd7735bd96b3e9ec85d"), "nombre" : "Sergio", "apellido1" : "Valiente", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02d0e1735bd96b3e9ec85e"), "apellido2" : "Oncina" }
{ "_id" : ObjectId("5c02e27a735bd96b3e9ec85f"), "nombre" : "Pilar", "apellido1" : "Valiente", "apellido2" : "Ruiz", "edad" : 30 }
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

DELETE

.remove

```
> db.clientes.find()
{ "_id" : ObjectId("5c0275f8f87ed69a71545d14"), "nombre" : "Pedro", "apellido1" : "León", "apellido2" : "Roldán" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Lara", "edad" : 38, "apel1" : "Ramírez", "sexo" : "mujer" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec859"), "nombre" : "Andrea", "apellido1" : "Lara", "apellido2" : "Sempere" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85a"), "nombre" : "Miguel", "apellido1" : "Cobos", "apellido2" : "Pascual" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85b"), "nombre" : "Manuel", "apellido1" : "Beltrán", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02cbd7735bd96b3e9ec85d"), "nombre" : "Sergio", "apellido1" : "Valiente", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02d0e1735bd96b3e9ec85e"), "apellido2" : "Oncina" }
{ "_id" : ObjectId("5c02e27a735bd96b3e9ec85f"), "nombre" : "Pilar", "apellido1" : "Valiente", "apellido2" : "Ruiz", "edad" : 30 }
```

```
> db.clientes.remove({apellido2:'Oncina'})
WriteResult({ "nRemoved" : 1 })
```

```
> db.clientes.find()
{ "_id" : ObjectId("5c0275f8f87ed69a71545d14"), "nombre" : "Pedro", "apellido1" : "León", "apellido2" : "Roldán" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec858"), "nombre" : "Lara", "edad" : 38, "sexo" : "mujer", "apellido1" : "Ramírez" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec859"), "nombre" : "Andrea", "apellido1" : "Lara", "apellido2" : "Sempere" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85a"), "nombre" : "Miguel", "apellido1" : "Cobos", "apellido2" : "Pascual" }
{ "_id" : ObjectId("5c02c396735bd96b3e9ec85b"), "nombre" : "Manuel", "apellido1" : "Beltrán", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02c7cd735bd96b3e9ec85c"), "nombre" : "Rosa", "apellido1" : "Rodríguez", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02cbd7735bd96b3e9ec85d"), "nombre" : "Sergio", "apellido1" : "Valiente", "apellido2" : "Sanz" }
{ "_id" : ObjectId("5c02e27a735bd96b3e9ec85f"), "nombre" : "Pilar", "apellido1" : "Valiente", "apellido2" : "Ruiz", "edad" : 30 }
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

DELETE

.remove

```
> db.clientes.remove(
... {apellido2:'Sanz'}
... )
WriteResult({ "nRemoved" : 3 })
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

SELECT

.find

db.nombre_colección.find(criterios_búsqueda) → muestra toda la información almacenada de los documentos que responden a los criterios de búsqueda.

Criterios de búsqueda

Ejemplo

Condición propiedad=valor	:valor	db.clientes.find({nombre:'Juan'})
Varias condiciones con AND	{cond1, cond2,...}	db.clientes.find({nombre:'Juan', edad:45})
Varias condiciones con OR	{ \$or: [...] }	db.clientes.find({ \$or:[{nombre:'Andrea'}, {nombre:'Miguel'}] })
Condición propiedad > valor	{ \$gt: <i>valor</i> } greater than	db.clientes.find({edad :{ \$gt: 20 } })
Condición propiedad >= valor	{ \$gte: <i>valor</i> } greater than equal	db.clientes.find({edad :{ \$gte: 20 } })
Condición propiedad < valor	{ \$lt: <i>valor</i> } lower than	db.clientes.find({edad :{ \$lt: 30 } })
Condición propiedad <= valor	{ \$lte: <i>valor</i> } lower than equal	db.clientes.find({edad :{ \$lte: 30 } })

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

SELECT

.find

Criterios de búsqueda

Ejemplo

Condición valor1<=propiedad >= valor2	<code>{ \$gte: valor1 }, { \$lte: valor2 }</code>	<code>db.clientes.find({ edad: { \$gte: 20, \$lte: 30 } })</code>
Condición encontrar una subcadena	<code>{ \$regex: 'expresión' }</code>	<code>db.clientes.find({ nombre: { regex: 'a' } })</code>
Condición EXISTS	<code>{ propiedad: { \$exists: true } }</code>	<code>db.clientes.find({ edad: { \$exists: true } })</code>
Condición NOT	<code>{ propiedad: { \$not: { condición } } }</code>	<code>db.clientes.find({ edad: { \$not: { \$gte: 30 } } })</code> <code>db.clientes.find({ edad: { \$not: { \$exists: true } } })</code>

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

SELECT

.find

`db.nombre_colección.find(criterios_búsqueda, propiedades a mostrar)`**→ muestra las propiedades que se indiquen de los documentos que responden a los criterios de búsqueda. (0 es no mostrar)**`db.clientes.find({ nombre: { $regex: 'a' }, { edad: 1 } })` → De cada documento muestra únicamente el identificador y la edad.`db.clientes.find({ nombre: { $regex: 'a' }, { edad: 0 } })` → De cada documento muestra todas las propiedades salvo la edad.

Aspectos sobre la respuesta

Ejemplo

Mejor presentación	<code>pretty()</code>	<code>db.clientes.find().pretty()</code>
Ordenación ascendente por una propiedad	<code>sort({ propiedad: 1 })</code>	<code>db.clientes.find({ nombre: { \$regex: 'a' } }).sort({ nombre: 1 })</code>
Ordenación descendente por una propiedad	<code>sort({ propiedad: -1 })</code>	<code>db.clientes.find({ nombre: { \$regex: 'a' } }).sort({ nombre: -1 })</code>
limitar el número de documentos mostrados	<code>limit</code>	<code>db.clientes.find({ nombre: { \$regex: 'a' }, { edad: 1 }).sort({ nombre: -1 }).limit(1)</code>

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

SELECT

.find

Agregación (group by)

```
db.micoleccion.aggregate(
  {$match:{condiciones}},
  {$group:{
    _id: "$groupbyfield",
    nombre_resultado: {función_agregada} } } )
```

- \$sum: suma o incrementa
- \$avg : calcula la media del grupo
- \$min: mínimo de los valores del grupo
- \$max: máximo del grupo
- \$first: obtiene el primer elemento del grupo
- \$last: obtiene el último elemento del grupo

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

Ejemplo: ¿Cuántos clientes hay de cada edad?

```
db.clientes.aggregate({$match:{}},{ $group:{_id:"$edad", cuenta:{$sum:1}}})
```

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

DROP DATABASE**.dropDatabase**

Para borrar una base de datos:

- Te situas en la base de datos `>use nombre_bd`
- TE puedes asegurar si está en la base de datos correcta `>bd`
- Ejecutas `>bd.dropDatabase()`

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA

<https://docs.mongodb.com>

DISEÑO de BASES de DATOS - CURSO 2018-19 - UA