

## EJERCICIO 1

Crear la colección 'articulos' en la base de datos 'base1' (eliminar la colección previamente), cargar luego 6 documentos:

```
use base1
```

```
db.articulos.drop()
```

```
db.articulos.insertOne(
```

```
{
```

```
  _id: 1,
```

```
  nombre: 'MULTIFUNCION HP DESKJET 2675',
```

```
  rubro: 'impresora',
```

```
  precio: 3000,
```

```
  stock: 20
```

```
}
```

```
)
```

```
db.articulos.insertOne(
```

```
{
```

```
  _id: 2,
```

```
  nombre: 'MULTIFUNCION EPSON EXPRESSION XP241',
```

```
  rubro: 'impresora',
```

```
  precio: 3700,
```

```
  stock: 5
```

```
}
```

```
)
```

```
db.articulos.insertOne(
```

```
{
```

```
  _id: 3,
```

```
  nombre: 'LED 19 PHILIPS',
```

```
        rubro: 'monitor',
        precio: 4500,
        stock: 2
    }
)
db.articulos.insertOne(
    {
        _id: 4,
        nombre: 'LED 22 PHILIPS',
        rubro: 'monitor',
        precio: 5700,
        stock: 4
    }
)
db.articulos.insertOne(
    {
        _id: 5,
        nombre: 'LED 27 PHILIPS',
        rubro: 'monitor',
        precio: 12000,
        stock: 1
    }
)

db.articulos.insertOne(
    {
        _id: 6,
        nombre: 'LOGITECH M90',
        rubro: 'mouse',
```

```
    precio: 300,  
    stock: 4  
  }  
)
```

1. Imprimir todos los documentos de la colección 'articulos'.
2. Imprimir todos los documentos de la colección 'articulos' que no son impresoras.
3. Imprimir todos los artículos que pertenecen al rubro de 'mouse'.
4. Imprimir todos los artículos con un precio mayor o igual a 5000.
5. Imprimir todas las impresoras que tienen un precio mayor o igual a 3500.
6. Imprimir todos los artículos cuyo stock se encuentra comprendido entre 0 y 4.

## EJERCICIO 2

Crear la colección 'articulos' en la base de datos 'base1' (eliminar la colección previamente), cargar luego 6 documentos:

```
use base1  
db.articulos.drop()  
  
db.articulos.insertOne(  
  {  
    _id: 1,  
    nombre: 'MULTIFUNCION HP DESKJET 2675',  
    rubro: 'impresora',  
    precio: 3000,  
    stock: 20
```

```
}  
)  
db.articulos.insertOne(  
  {  
    _id: 2,  
    nombre: 'MULTIFUNCION EPSON EXPRESSION XP241',  
    rubro: 'impresora',  
    precio: 3700,  
    stock: 5  
  }  
)  
db.articulos.insertOne(  
  {  
    _id: 3,  
    nombre: 'LED 19 PHILIPS',  
    rubro: 'monitor',  
    precio: 4500,  
    stock: 2  
  }  
)  
db.articulos.insertOne(  
  {  
    _id: 4,  
    nombre: 'LED 22 PHILIPS',  
    rubro: 'monitor',  
    precio: 5700,  
    stock: 4  
  }  
)
```

```
db.articulos.insertOne(  
  {  
    _id: 5,  
    nombre: 'LED 27 PHILIPS',  
    rubro: 'monitor',  
    precio: 12000,  
    stock: 1  
  }  
)
```

```
db.articulos.insertOne(  
  {  
    _id: 6,  
    nombre: 'LOGITECH M90',  
    rubro: 'mouse',  
    precio: 300,  
    stock: 4  
  }  
)
```

1. Imprimir todos los documentos de la colección 'articulos'.
2. Borrar los documentos de la colección 'articulos' cuyo rubro son impresoras, utilizar las dos sintaxis que permite MongoDB.
3. Borrar todos los artículos que tienen un \_id mayor o igual a 5.

### **EJERCICIO 3**

Crear la colección 'articulos' en la base de datos 'base1' (eliminar la colección previamente), cargar luego 6 documentos:

```
use base1
```

```
db.articulos.drop()
```

```
db.articulos.insertOne(
```

```
{
```

```
  _id: 1,
```

```
  nombre: 'MULTIFUNCION HP DESKJET 2675',
```

```
  rubro: 'impresora',
```

```
  precio: 3000,
```

```
  stock: 20
```

```
}
```

```
)
```

```
db.articulos.insertOne(
```

```
{
```

```
  _id: 2,
```

```
  nombre: 'MULTIFUNCION EPSON EXPRESSION XP241',
```

```
  rubro: 'impresora',
```

```
  precio: 3700,
```

```
  stock: 5
```

```
}
```

```
)
```

```
db.articulos.insertOne(
```

```
{
```

```
  _id: 3,
```

```
  nombre: 'LED 19 PHILIPS',
```

```
  rubro: 'monitor',
```

```
    precio: 4500,  
    stock: 2  
  }  
)  
db.articulos.insertOne(  
  {  
    _id: 4,  
    nombre: 'LED 22 PHILIPS',  
    rubro: 'monitor',  
    precio: 5700,  
    stock: 4  
  }  
)  
db.articulos.insertOne(  
  {  
    _id: 5,  
    nombre: 'LED 27 PHILIPS',  
    rubro: 'monitor',  
    precio: 12000,  
    stock: 1  
  }  
)  
  
db.articulos.insertOne(  
  {  
    _id: 6,  
    nombre: 'LOGITECH M90',  
    rubro: 'mouse',  
    precio: 300,
```

```
    stock: 4
  }
)
```

1. Imprimir todos los documentos de la colección 'articulos'.
2. Modificar el precio del mouse 'LOGITECH M90'.
3. Fijar el stock en 0 del artículo cuyo \_id es 6.
4. Agregar el campo proveedores con el array ['Martinez','Gutierrez'] para el artículo cuyo \_id es 6.
5. Eliminar el campo proveedores para el artículo cuyo \_id es 6.

## EJERCICIO 4

Crear la colección 'articulos' en la base de datos 'base1' (eliminar la colección previamente), cargar luego 6 documentos:

```
use base1
db.articulos.drop()

db.articulos.insertOne(
  {
    _id: 1,
    nombre: 'MULTIFUNCION HP DESKJET 2675',
    rubro: 'impresora',
    precio: 3000,
    stock: 20
  }
)
db.articulos.insertOne(
  {
    _id: 2,
```



```
    nombre: 'MULTIFUNCION EPSON EXPRESSION XP241',
    rubro: 'impresora',
    precio: 3700,
    stock: 5
  }
)
db.articulos.insertOne(
  {
    _id: 3,
    nombre: 'LED 19 PHILIPS',
    rubro: 'monitor',
    precio: 4500,
    stock: 2
  }
)
db.articulos.insertOne(
  {
    _id: 4,
    nombre: 'LED 22 PHILIPS',
    rubro: 'monitor',
    precio: 5700,
    stock: 4
  }
)
db.articulos.insertOne(
  {
    _id: 5,
    nombre: 'LED 27 PHILIPS',
    rubro: 'monitor',
```

```

    precio: 12000,
    stock: 1
  }
)

db.articulos.insertOne(
{
  _id: 6,
  nombre: 'LOGITECH M90',
  rubro: 'mouse',
  precio: 300,
  stock: 4
}
)

```

1. Imprimir todos los documentos de la colección 'articulos'.
2. Fijar el stock en cero para todos los artículos del rubro monitor.
3. Agregar un campo llamado 'pedir' con el valor true para todos los artículos que tienen el campo stock en 0.
4. Eliminar el campo 'pedir' de todos los documentos

## EJERCICIO 5

Crear la colección 'medicamentos' en la base de datos 'base1' (eliminar la colección previamente), cargar luego 6 documentos:

```

use base1
db.medicamentos.drop()

db.medicamentos.insertOne(

```

```
{
  _id: 1,
  nombre: 'Sertal',
  laboratorio: 'Roche',
  precio: 5.2,
  cantidad: 100
}
)
db.medicamentos.insertOne(
{
  _id: 2,
  nombre: 'Buscapina',
  laboratorio: 'Roche',
  precio: 4.10,
  cantidad: 200
}
)
db.medicamentos.insertOne(
{
  _id: 3,
  nombre: 'Amoxidal 500',
  laboratorio: 'Bayer',
  precio: 15.60,
  cantidad: 100
}
)
db.medicamentos.insertOne(
{
  _id: 4,
```

```

    nombre: 'Paracetamol 500',
    laboratorio: 'Bago',
    precio: 1.90,
    cantidad: 200
  }
)
db.medicamentos.insertOne(
  {
    _id: 5,
    nombre: 'Bayaspirina',
    laboratorio: 'Bayer',
    precio: 2.10,
    cantidad: 150
  }
)
db.medicamentos.insertOne(
  {
    _id: 6,
    nombre: 'Amoxidal jarabe',
    laboratorio: 'Bayer',
    precio: 5.10,
    cantidad: 50
  }
)

```

1. Imprimir todos los documentos de la colección 'medicamentos'.
2. Recupere los medicamentos cuyo laboratorio sea 'Roche' y cuyo precio sea menor a 5.

3. Recupere los medicamentos cuyo laboratorio sea 'Roche' o cuyo precio sea menor a 5.
4. Muestre todos los medicamentos cuyo laboratorio NO sea "Bayer"
5. Muestre todos los medicamentos cuyo laboratorio sea "Bayer" y cuya cantidad NO sea=100
6. Elimine todos los documentos de la colección medicamentos cuyo laboratorio sea igual a "Bayer" y su precio sea mayor a 10
7. Cambie la cantidad por 200, a todos los medicamentos de "Roche" cuyo precio sea mayor a 5
8. Borre los medicamentos cuyo laboratorio sea "Bayer" o cuyo precio sea menor a 3

## **EJERCICIO 6**

Crear la colección 'articulos' en la base de datos 'base1' (eliminar la colección previamente), cargar luego 6 documentos:

```
use base1
```

```
db.articulos.drop()
```

```
db.articulos.insertOne(
```

```
{
```

```
  _id: 1,
```

```
  nombre: 'MULTIFUNCION HP DESKJET 2675',
```

```
  rubro: 'impresora',
```

```
  precio: 3000,
```

```
  stock: 20
```

```
}
```

```
)
```

```
db.articulos.insertOne(
```

```
{
```

```
  _id: 2,
```

```
nombre: 'MULTIFUNCION EPSON EXPRESSION XP241',
rubro: 'impresora',
precio: 3700,
stock: 5
}
)
db.articulos.insertOne(
{
  _id: 3,
  nombre: 'LED 19 PHILIPS',
  rubro: 'monitor',
  precio: 4500,
  stock: 2
}
)
db.articulos.insertOne(
{
  _id: 4,
  nombre: 'LED 22 PHILIPS',
  rubro: 'monitor',
  precio: 5700,
  stock: 4
}
)
db.articulos.insertOne(
{
  _id: 5,
  nombre: 'LED 27 PHILIPS',
  rubro: 'monitor',
```

```

    precio: 12000,
    stock: 1
  }
)

db.articulos.insertOne(
{
  _id: 6,
  nombre: 'LOGITECH M90',
  rubro: 'mouse',
  precio: 300,
  stock: 4
}
)

```

1. Imprimir todos los documentos de la colección 'articulos', mostrar solo los campos \_id y nombre.
2. Imprimir todos los documentos de la colección 'articulos' que son impresoras, mostrar solo los campos nombre y precio.
3. Imprimir todas las impresoras que tienen un precio mayor o igual a 3500. Solo mostrar los campos \_id, nombre, precio y stock
4. Imprimir todos los documentos de la colección 'articulos' que son monitor, mostrar solo los campos nombre y precio ordenados de menor a mayor.

## EJERCICIO 7

Crear la colección 'libros' en la base de datos 'base1' (eliminar la colección previamente), cargar luego 4 documentos:

```
use base1
```

```
db.libros.drop()
```

```
db.libros.insertOne(  
  {  
    _id: 1,  
    titulo: 'El aleph',  
    autor: {  
      nombre: 'Borges',  
      nacionalidad: 'Argentina'  
    },  
    editorial: ['Siglo XXI', 'Planeta'],  
    precio: 20,  
    cantidad: 50  
  }  
)
```

```
db.libros.insertOne(  
  {  
    _id: 2,  
    titulo: 'Martin Fierro',  
    autor: {  
      nombre: 'Jose Hernandez',  
      nacionalidad: 'Argentina'  
    },  
    editorial: ['Siglo XXI'],  
    precio: 50,  
    cantidad: 12  
  }  
)
```

```
db.libros.insertOne(  
  {
```



```

{
  _id: 3,
  titulo: 'Aprenda PHP',
  autor: {
    nombre: 'Mario Molina',
    nacionalidad: 'Española'
  },
  editorial: ['Siglo XXI', 'Planeta'],
  precio: 50,
  cantidad: 20
}
)
db.libros.insertOne(
{
  _id: 4,
  autor: {
    nombre: 'Java en 10 minutos',
    nacionalidad: 'Española'
  },
  editorial: ['Siglo XXI'],
  precio: 45,
  cantidad: 1
}
)

```

1. Imprimir todos los documentos de la colección 'libros'.
2. Imprimir todos los libros de autores de nacionalidad 'Argentina'.
3. Imprimir los libros de 'Borges'.

4. Imprimir todos los libros de nacionalidad 'Española' que cuestan 50 o más.

## EJERCICIO 8

Crear la colección 'alumnos' en la base de datos 'base1' (eliminar la colección previamente), cargar luego 3 documentos:

use base1

```
db.alumnos.drop()
```

```
db.alumnos.insertOne(  
  {  
    _id: 20456123,  
    apellido: 'Gonzalez',  
    nombre: 'Ana',  
    domicilio: 'Colon 123',  
    fechanacimiento: new Date(1990,7,15)  
  }  
)
```

```
db.alumnos.insertOne(  
  {  
    _id: 45123845,  
    apellido: 'Juarez',  
    nombre: 'Bernardo',  
    domicilio: 'Sucre 456',  
    fechanacimiento: new Date(1964,0,1)  
  }  
)
```

```
db.alumnos.insertOne(  
  {  
    _id: 12345678,  
    apellido: 'Perez',  
    nombre: 'Carlos',  
    domicilio: 'Avda. 123',  
    fechanacimiento: new Date(1985,12,1)  
  }  
)
```

```
{  
  _id: 16567512,  
  apellido: 'Perez',  
  nombre: 'Laura',  
  domicilio: '21 de Septiembre 3233',  
  fechanacimiento: new Date(1972,3,2)  
}  
)
```

```
db.alumnos.find().pretty()
```

1. Imprimir todos los documentos de la colección alumnos.
2. Imprimir solo el apellido y la fecha de nacimiento.
3. Imprimir todos los datos ordenados por la fecha de nacimiento de mayor a menor.
4. Imprimir todos los alumnos que nacieron a partir de 1970.

## **EJERCICIO 9**

¿Cómo obtenemos el número total de estudiantes que han pertenecido alguna vez a cada una de las universidades ordenando los resultados de forma descendente por el número de estudiantes?

## **EJERCICIO 10**

Trabajando con el dataset de ZIPS (carpeta de datos, importarlo mediante mongoimport). Responde a las siguientes preguntas:

1. Cuales son los estados con una población superior a 10 millones
2. Cual es la población media de cada uno de los estados
3. Cual es la ciudad mas grande y mas pequeña de cada estado

## EJERCICIO 11

Crea una colección con esquema de validación llamada posts. Define los campos: titulo, cuerpo, categoria, likes, tags, fecha. Titulo y cuerpo tienen que ser de tipo string y obligatorios. Categoria debe ser de tipo string y opcional. Likes de tipo entero y opcional. Tags un array de strings opcional. Date una fecha y es opcional.

## EJERCICIO 12

Crea una base de datos y agrega en ella tres tipos de usuarios:

1. Lector, tan solo podrá escribir en la BBDD
2. Escritor, tan solo podrá leer de la BBDD
3. Administrador, podrá leer y escribir de la BBDD

Prueba los tres usuarios, insertando, leyendo y haciendo ambas acciones sobre la BBDD respectivamente

Una vez compruebes que todo funciona correctamente, vamos a ponernos en el siguiente supuesto:

4. El administrador deja la empresa. Debemos de eliminar su usuario. Busca en la documentación como realizar esta acción.
5. El escritor pasa a ser administrador, elevale los permisos añadiéndole otro rol. Utiliza el comando

```
db.grantRolesToUser(  
    '<userName>',  
    [ { role : '<roleName>', db : '<dbName>' }, '<roleName>', ... ]  
)
```

6. El lector pasa a ser escritor, habilítale los permisos de escritura y elimínale los de lectura con el siguiente comando

```
db.revokeRolesFromUser(  
  '<userName>',  
  [ { role : '<roleName>', db : '<dbname>' } | '<roleName>' ]  
)
```

Prueba de nuevo que funciona todo correctamente.