**Base de datos noSQL. Sebastian Carrondi 3º1 Seminario**

**Problemas propuestos (1.1)**

1. Insertar 2 documentos en la colección clientes con '\_id' no repetidos

2. Intentar insertar otro documento con clave repetida.

3. Mostrar todos los documentos de la colección libros

**Resultados:**

**2.**

bas1> use clientes

switched to db clientes

clientes> db.clientes.insertOne( {\_id:1, nombre: 'Jose Perez', domicilio:'Colon1000',provincia:'Neuquen'})

{ acknowledged: true, insertedId: 1 }

clientes> db.clientes.insertOne( {\_id:2, nombre: 'Juan Lopez', domicilio:'San Martin', provincia:'Mendoza'})

{ acknowledged: true, insertedId: 2 }

clientes> db.clientes.insertOne( {\_id:1, nombre: 'Santiago Gomez', domicilio:'Salta200', provincia:'San Juan'})

MongoServerError: E11000 duplicate key error collection: clientes.clientes index: \_id\_ dup key: { \_id: 1 }

clientes>

**3.**

clientes> use bas1

switched to db bas1

bas1> db.libros.find()

[

{

\_id: ObjectId('66e1f4b0fec171a7152710bc'),

codigo: 1,

nombre: 'El aleph',

autor: 'Borges',

editoriales: [ 'Planeta', 'Siglo XXI' ]

},

{

\_id: ObjectId('66e1f612fec171a7152710bd'),

codigo: 2,

nombre: 'Martin Fierro',

autor: 'Jose Hernandes',

editoriales: [ 'Planeta' ]

},

{

\_id: ObjectId('66e1f6ebfec171a7152710be'),

codigo: 3,

nombre: 'Aprenda PHP',

autor: 'Mario Molina',

editoriales: [ 'Planeta' ]

},

{

\_id: ObjectId('66e1f6ebfec171a7152710bf'),

codigo: 4,

nombre: 'Jaca en 10 minutos',

autor: 'Barros Sergio',

editoriales: [ 'Planeta', 'Siglo XXI' ]

}

]

**Problemas propuestos (1.2)**

1. Crear una base de datos llamada "blog".

2. Agregar una colección llamada "posts" e insertar 1 documento con una estructura a su elección.

3. Mostrar todas las bases de datos actuales.

4. Eliminar la colección "posts"

5. Eliminar la base de datos "blog" y mostrar las bases de datos existentes.

**Respuesta:**

clientes> use blog

switched to db blog

blog> db.posts.insertOne( {\_id:1, autor:'Sebastian Casas', tema:'Bases noSQL'})

{ acknowledged: true, insertedId: 1 }

blog> show dbs

Colegio 40.00 KiB

admin 40.00 KiB

bas1 72.00 KiB

blog 40.00 KiB

clientes 40.00 KiB

config 108.00 KiB

local 72.00 KiB

blog> db.posts.deleteMany({})

{ acknowledged: true, deletedCount: 1 }

blog> db.dropDatabase()

{ ok: 1, dropped: 'blog' }

blog> show dbs

Colegio 40.00 KiB

admin 40.00 KiB

bas1 72.00 KiB

clientes 40.00 KiB

config 108.00 KiB

local 72.00 KiB

**Problemas propuestos (1.3)**

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

2. Imprimir todos los documentos de la colección 'articulos'.

3. Imprimir todos los documentos de la colección 'articulos' que no son impresoras.

4. Imprimir todos los artículos que pertenecen al rubro de 'mouse'.

5. Imprimir todos los artículos con un precio mayor o igual a 5000.

6. Imprimir todas las impresoras que tienen un precio mayor o igual a 3500.

7. Imprimir todos los artículos cuyo stock se encuentra comprendido entre 0 y 4.

**Respuestas:**

**2:**

bas1> db.articulos.insertOne( {\_id: 1, nombre: 'MULTIFUNCION HP DESKJET 2675', rubro: 'impresora', precio: 3000, stock: 20})

{ acknowledged: true, insertedId: 1 }

bas1> db.articulos.insertOne( { \_id: 2, nombre: 'MULTIFUNCION EPSON EXPRESSION XP241', rubro: 'impresora', precio: 3700, stock: 5})

{ acknowledged: true, insertedId: 2 }

bas1> db.articulos.insertOne( { \_id: 3, nombre: 'LED 19 PHILIPS', rubro: 'monitor', precio: 4500, stock: 2})

{ acknowledged: true, insertedId: 3 }

bas1> db.articulos.insertOne( { \_id: 4, nombre: 'LED 22 PHILIPS', rubro: 'monitor', precio: 5700, stock: 4 })

{ acknowledged: true, insertedId: 4 }

bas1> db.articulos.insertOne( { \_id: 5, nombre: 'LED 27 PHILIPS', rubro: 'monitor', precio: 12000, stock: 1 })

{ acknowledged: true, insertedId: 5 }

bas1> db.articulos.insertOne( { \_id: 6, nombre: 'LOGITECH M90', rubro: 'mouse', precio: 300, stock: 4 })

{ acknowledged: true, insertedId: 6 }

bas1> db.articulos.find()

[

{

\_id: 1,

nombre: 'MULTIFUNCION HP DESKJET 2675',

rubro: 'impresora',

precio: 3000,

stock: 20

},

{

\_id: 2,

nombre: 'MULTIFUNCION EPSON EXPRESSION XP241',

rubro: 'impresora',

precio: 3700,

stock: 5

},

{

\_id: 3,

nombre: 'LED 19 PHILIPS',

rubro: 'monitor',

precio: 4500,

stock: 2

},

{

\_id: 4,

nombre: 'LED 22 PHILIPS',

rubro: 'monitor',

precio: 5700,

stock: 4

},

{

\_id: 5,

nombre: 'LED 27 PHILIPS',

rubro: 'monitor',

precio: 12000,

stock: 1

},

{

\_id: 6,

nombre: 'LOGITECH M90',

rubro: 'mouse',

precio: 300,

stock: 4

}

]

**3:**

bas1> db.articulos.find({rubro : {$ne: 'impresora'}})

[

{

\_id: 3,

nombre: 'LED 19 PHILIPS',

rubro: 'monitor',

precio: 4500,

stock: 2

},

{

\_id: 4,

nombre: 'LED 22 PHILIPS',

rubro: 'monitor',

precio: 5700,

stock: 4

},

{

\_id: 5,

nombre: 'LED 27 PHILIPS',

rubro: 'monitor',

precio: 12000,

stock: 1

},

{

\_id: 6,

nombre: 'LOGITECH M90',

rubro: 'mouse',

precio: 300,

stock: 4

}

]

**4:**

bas1> db.articulos.find({rubro :'mouse'})

[

{

\_id: 6,

nombre: 'LOGITECH M90',

rubro: 'mouse',

precio: 300,

stock: 4

}

]

**5:**

bas1> db.articulos.find({precio : {$gte: 5000 }})

[

{

\_id: 4,

nombre: 'LED 22 PHILIPS',

rubro: 'monitor',

precio: 5700,

stock: 4

},

{

\_id: 5,

nombre: 'LED 27 PHILIPS',

rubro: 'monitor',

precio: 12000,

stock: 1

}

]

**6:**

bas1> db.articulos.find({rubro: 'impresora',precio : {$gte: 3500}})

[

{

\_id: 2,

nombre: 'MULTIFUNCION EPSON EXPRESSION XP241',

rubro: 'impresora',

precio: 3700,

stock: 5

}

]

**7:**

bas1> db.articulos.find({stock : {$gte: 0, $lte: 4}})

[

{

\_id: 3,

nombre: 'LED 19 PHILIPS',

rubro: 'monitor',

precio: 4500,

stock: 2

},

{

\_id: 4,

nombre: 'LED 22 PHILIPS',

rubro: 'monitor',

precio: 5700,

stock: 4

},

{

\_id: 5,

nombre: 'LED 27 PHILIPS',

rubro: 'monitor',

precio: 12000,

stock: 1

},

{

\_id: 6,

nombre: 'LOGITECH M90',

rubro: 'mouse',

precio: 300,

stock: 4

}

]

Investigar y responder

1. ¿Qué es una Base de Datos No relacional?

2. ¿Las BADA No SQL permiten operaciones CRUD?

3. ¿Qué es un Replica Set?

4. ¿En BADA No SQL se escala verticalmente u horizontalmente?

5. ¿Cómo está compuesta una Base de Datos No SQL de documentos? ¿ Usa tablas, filas, registros?

6. ¿En qué casos conviene usar BADA Relacionales y BADA No SQL?

7. Crear la estructura de una Base de Datos de documentos con una colección para guardar datos de una veterinaria

1. Las bases de datos no relacionales o NoSQL utilizan una variedad de modelos de datos para acceder y administrar datos. Estos tipos de bases de datos están optimizados específicamente para aplicaciones que requieren modelos de datos flexibles, grandes volúmenes de datos y baja latencia, lo que se logra mediante la flexibilización de algunas de las restricciones de coherencia de datos en bases de datos relacionales. Además, no usa el esquema tabular de filas y columnas.
2. Sí, las bada noSQL permiten realizar operaciones CRUD, puede variar la manera en la que se implementan, pero es posible.
3. ReplicaSet es un proceso que ejecuta varias instancias de un pod y mantiene constante el número especificado de pods. Se asegura de que un conjunto estable de pods de réplica se ejecuta en un momento dado, lo que garantiza que haya un número especificado de pods idénticos disponibles. Un Pod es un grupo de uno o más contenedores.
4. La escalabilidad horizontal es una de las características más deseables en una base de datos. Es uno de los principales atributos de las bases de datos NoSQL. Esto debido a que el crecimiento es sumamente extenso y técnicamente no tiene límites.
5. Las BADA noSQL están compuestas de colecciones para reemplazar las tablas típicas de las BADA SQL, también usan documentos del tipo JSON que cuentan con estructuras anidadas permitiendo almacenar datos jerárquicos y complejos. Los campos forman parte de los documentos y representan pares clave-valor, básicamente son el nombre de los campos y los valores que estos toman en la base de datos y cuentan con índices que sirven para optimizar las consultas.
6. Conviene usar BADA relacionales cuando se busca tener un registro de datos que esté relacionado y se quiera mantener una estructura fija para el proyecto. Mientras tanto una BADA noSQL es mejor cuando se manejan grandes volúmenes de datos y no hay una necesidad de tenerlos relacionados.
7. db Veterinaria

Veterinaria>db.pacientes.insertOne ({ \_id: 1, nombre: 'Michifus', especie: 'Gato', raza: 'Siames', edad: 2, peso: 6, sexo: 'Macho', dueño: { nombre: 'Juan Perez', telefono: '+11 1111 1111'}})