Base de Datos No SQL

Preguntas y problemas propuestos

**Investigar y responder**

1. ¿Qué es una Base de Datos No relacional?
2. ¿Las BADA No SQL permiten operaciones CRUD?
3. ¿Qué es una 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.
8. Una Base de Datos No relacional es aquella que no usa el esquema de filas y columnas como las tradicionales (Base de datos Relacionales). Usa un modelo de almacenamiento mejor optimizado para los datos que almacena, donde no importa la relación entre ellos. Dando una mejor disponibilidad, velocidad y volumen de almacenamiento para los datos.
9. Las Bases de Datos No SQL permiten operaciones CRUD, proporcionan un lenguaje de consulta o API que les permite a los desarrolladores:

* **Create:** crear documentos con identificadores únicos (claves).
* **Read:** leer los datos de los documentos mediante consultas ya sea de valores de campos o las claves.
* **Update:** actualizar los documentos existentes.
* **Delete:** eliminar documentos por completo de la base de datos.

1. Replica Set son un conjunto de servidores de Base de Datos, un primario y varios secundarios. Esto sirve por si el servidor primario se cae, o en caso de que se baje para hacer un Backup, se activa un proceso donde seguirá trabajando el servidor secundario.
2. En BADA no SQL se puede escalar tanto verticalmente como horizontalmente, pudiendo elegir si quiero tener un servidor más grande o varios servidores.

* Verticalmente al servidor se le puede agregar más RAM, otra CPU, más memoria o discos con mayores velocidades en la transferencia de datos y una mejor red para manejar las peticiones de los usuarios.
* Horizontalmente se puede agregar otro Replica Set y así comenzar a trabajar con 2 servidores en conjunto en una misma base de datos.

1. Una base de datos No SQL de documentos está compuesta principalmente por documentos, por ejemplo, JSON, donde se almacenan los datos semiestructurados de diversas formas, pudiendo ser, ‘clave-valor’ con un solo valor de cualquier tipo para la clave, ‘matriz’ donde hay una colección ordenada de valores para una clave, y ‘objetos’ son un conjunto de pares clave-valor. La agrupación de estos documentos relacionados se llama colección.  
   No usa tablas, filas o registros. Si se quiere comparar, los documentos serian filas o registros y las colecciones las tablas que guardan esas filas y registros, pero con mayor flexibilidad.
2. Ambas tienen sus pro y contras al momento de elegir, hay que tener en cuenta que requerimientos va a tener la aplicación a realizar.  
   Si vamos a trabajar con muchos datos donde no importa relacionarlos, no debemos seguir una estructura de los datos a guardar y necesitamos disponibilidad y velocidad la mejor opción sería una Base de Datos No SQL, pero hay que tener en cuenta el presupuesto dedicado al almacenamiento, que es una tecnología nueva por lo que le falta madurez y desarrolladores que sepan gestionarla y donde no hay compatibilidad entre los distintos gestores de bases de datos NoSQL.  
   En otro caso, si vamos a trabajar con datos organizados y relacionados, donde se deban manejar con más comodidad la mejor opción sería una Base de Datos relacional (SQL), ahora hay que tener en cuenta que no vamos a contar con el inmenso volumen o la flexibilidad para manejar y almacenar datos que ofrecen las bases de datos NoSQL.
3. Veterinaria

use veterinaria

db.mascotas.insertOne({ \_id: 1, nombre: 'Toby', especie: 'Canino', raza: 'Beagle', genero: 'Macho', pelaje: 'Tricolor', propietario: 'Ramon Rodriguez', telefono: ['1176284920']})

**MongoDB:** { acknowledged: true, insertedId: 1 }

db.mascotas.insertOne({\_id: 2, nombre: 'Kaia', especie: 'Felino', raza: 'Siames', genero: 'Hembra', pelaje: 'Seal Point', propietario: 'Camila Perez', telefono: ['1123859375']})

**MongoDB:** { acknowledged: true, insertedId: 2 }

**Problemas Propuestos:**

**(1.1)**

**1.** Insertar 2 documentos en la colección clientes con ‘\_id’ no repetidos.

db.clientes.insertMany(

[

{

\_id: 2,

nombre: 'Perez Gabriel',

domicilio: 'Catamarca 564',

provincia: 'Buenos Aires',

telefono: [ '1123653275' ]

},

{

\_id: 3,

nombre: 'Rodriguez Florencia',

domicilio: 'Almafuerte 870',

provincia: 'Santa Fe',

telefono: [ '1195724876', '1109235039']

}

]

)

**MongoDB:** { acknowledged: true, insertedIds: { '0': 2, '1': 3 } }

**2.** Intentar insertar otro documento con clave repetida.

db.clientes.insertOne( {\_id: 3, nombre: 'Lima Sofia', domicilio: 'Arenales 1020', provincia: 'Buenos Aires', telefono: [ '1146889673' ]} )

**MongoDB:** MongoServerError: E11000 duplicate key error collection: base1.clientes index: \_id\_ dup key: { \_id: 3 }

**3.** Mostrar todos los documentos de la colección libros.

db.libros.find()

[

{

\_id: ObjectId('6709c80c8f7d1d003b86b01d'),

codigo: 1,

nombre: 'El aleph',

autor: 'Borges',

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

},

{

\_id: ObjectId('6709c9048f7d1d003b86b01e'),

codigo: 2,

nombre: 'Martin Fierro',

autor: 'Jose Hernandez',

editoriales: [ 'Planeta' ]

},

{

\_id: ObjectId('6709ccd9227c5936fd86b01d'),

codigo: 3,

nombre: 'Aprenda PHP',

autor: 'Mario Molina',

editoriales: [ 'Planeta' ]

},

{

\_id: ObjectId('6709ccd9227c5936fd86b01e'),

codigo: 4,

nombre: 'Java en 10 minutos',

autor: 'Barros Sergio',

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

}

]

**(1.2)**

**1.** Crear una base de datos llamada “blog”.

use blog

**MongoDB:** switched to db blog

**2.** Agregar una colección llamada “posts” e insertar 1 documento con una estructura a elección.

db.posts.insertOne(

{

titulo: 'Luchando contra el Phishing',

cuerpo: ' Estos servicios son utilizados para enviar correos falsos a los objetivos del phishing, pudiendo incluso suplantar la identidad del remitente. Esta vía de ingeniería social es muy utilizada por los *phishers*, ya que a día de hoy todavía hay muchos usuarios de confían que un correo es legítimo porque el remitente es una persona conocida, y además así lo indica el *correo remitente*.',

autor: 'Oscar Ramos',

etiquetas: [ 'Phishing', 'Hacking', 'Spam']

}

)

**MongoDB:**

{

acknowledged: true,

insertedId: ObjectId('6709edc2ad46d4468986b01e')

}

**3.** Mostrar todas las bases de datos actuales.

show dbs

Colegio 72.00 KiB

admin 40.00 KiB

blog 40.00 KiB

config 96.00 KiB

local 40.00 KiB

**4.** Eliminar la colección “posts”

db.posts.drop()

**MongoDB:** true

**5.** Eliminar la base de datos “blog” y mostrar las bases de datos existentes.

db.dropDatabase()

**MongoDB:** { ok: 1, dropped: 'blog' }

show dbs

Colegio 72.00 KiB

admin 40.00 KiB

config 96.00 KiB

local 40.00 KiB

**(1.3)**

**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

}

)

**MongoDB:** { acknowledged: true, insertedId: 1 }

db.articulos.insertOne(

{

\_id: 2,

nombre: 'MULTIFUNCION EPSON EXPRESSION XP241',

rubro: 'impresora',

precio: 3700,

stock: 5

}

)

**MongoDB:** { acknowledged: true, insertedId: 2 }

db.articulos.insertOne(

{

\_id: 3,

nombre: 'LED 19 PHILIPS',

rubro: 'monitor',

precio: 4500,

stock: 2

}

)**MongoDB:** { acknowledged: true, insertedId: 3 }

db.articulos.insertOne(

{

\_id: 4,

nombre: 'LED 22 PHILIPS',

rubro: 'monitor',

precio: 5700,

stock: 4

}

)**MongoDB:** { acknowledged: true, insertedId: 4 }

db.articulos.insertOne(

{

\_id: 5,

nombre: 'LED 27 PHILIPS',

rubro: 'monitor',

precio: 12000,

stock: 1

}

)

**MongoDB:** { acknowledged: true, insertedId: 5 }

db.articulos.insertOne(

{

\_id: 6,

nombre: 'LOGITECH M90',

rubro: 'mouse',

precio: 300,

stock: 4

}

)**MongoDB:** { acknowledged: true, insertedId: 6 }

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

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

db.articulos.find({ rubro: { $nin: ['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.** Imprimir todos los articulos que pertenecen al rubro de 'mouse'.

db.articulos.find({rubro: { $in: ['mouse']}})

[

{

\_id: 6,

nombre: 'LOGITECH M90',

rubro: 'mouse',

precio: 300,

stock: 4

}

]

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

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

db.articulos.find({ rubro: {$in: ['impresora']}, precio: {$gte: 3500}})

[

{

\_id: 2,

nombre: 'MULTIFUNCION EPSON EXPRESSION XP241',

rubro: 'impresora',

precio: 3700,

stock: 5

}

]

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

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

}

]