

Bases de datos NoSQL

Las bases de datos no podían quedarse estancadas en un modelo que no respondía a problemas que se estaban presentando, el termino NoSQL hace referencia al modelo de bases de datos que busca alternativas al sistema de bases de datos relacionales, se destaca por priorizar el acceso rápido a la información por el sobre la integridad de datos.

Aparecen por los grandes volúmenes de información que están creciendo a niveles excepcionales, y la manera en la que se los administra se hace cada vez mas compleja. Los nuevos requerimientos se originan en base de los siguientes aspectos:

- Gran aumento de tamaño de los archivos, junto la cantidad de los mismos.
- La velocidad de respuesta ante las consultas simultaneas de millones de usuarios.
- Sistemas descentralizados.

Ventajas

Las principales ventajas que encontramos a la hora de trabajar con Bases de Datos NoSQL son las siguientes:

- Evitar la complejidad innecesaria.
- Alto rendimiento
- Escalabilidad horizontal y hardware básico
- No es necesario llevar a cabo el proceso de mapeado
- Comprometer la fiabilidad para lograr un mejor desempeño
- Facilidad para descentralizar la base de datos
- Consultas simples

Diferencias

El teorema de CAP dice que, en un sistema de bases de datos distribuida, puede tener como máximo solo dos de las tres características, en donde se puede tener consistencia, disponibilidad y tolerancia de peticiones.

- Consistencia
- Disponibilidad (Availability)
- Tolerancia a peticiones (Partition Tolerance)
- Atomicidad
- Consistencia
- Aislado
- Durabilidad

Tipos de BD NoSQL

- **CLAVE VALOR**
- **DOCUMENTALES**
- **GRAFO**
- **ORIENTADA A OBJETOS**
- **COLUMNAS**

MongoDB

Es una base de datos de documentos que ofrece una gran escalabilidad y flexibilidad, y un modelo de consulta e indexación avanzado. Almacena datos en documentos flexibles similares a JSON, por lo que los campos pueden variar entre documentos y la estructura de datos puede cambiarse con el tiempo.

Un registro en MongoDB es un documento, un documentos es una estructura de datos compuesta de pares de campos y valores.

La ventaja de utilizar documentos son:

- Los documentos corresponden a tipos de datos nativos en muchos lenguajes de programación.
- Los documentos y matrices embebidos reducen la necesidad de realizar joins, los cuales son muy pesados.
- Los esquemas dinámicos admiten polimorfismo con total fluidez

MongoDB almacena documentos en colecciones. Las colecciones son análogas a las tablas de las bases de datos relacionales. Las claves son cadenas, pero existen un conjunto de restricciones, debe ser única y no puede contener un carácter nulo.

Se debe evitar el uso de puntos y el símbolo dólar, ya que dentro del motor estos son caracteres especiales.

Los elementos de MongoDB:

- Nulo
- Boolean
- Numero (hay enteros y decimales)
- Cadena de caracteres
- Arrays
- Documentos
- Fechas
- Expresiones regulares
- Identificadores de objetos
- Datos binarios
- Código de JavaScript

Una colección es un conjunto de documentos, en donde las colecciones son análogas a las tablas en las bases de datos relacionales. Es una buena práctica que una colección agrupe documentos muy similares, en donde cada uno de ellos contenga cierta relación. Las colecciones se agrupan en una base de datos y una base de datos es un conjunto de colecciones.

Creacion de una base de datos

No es necesario crear la base de datos ya que esta se crea únicamente cuando se inserta un documento en ella.

Inserción de datos

Si el documento no especifica un campo `_id`, MongoDB agrega el campo `_id` con un valor ObjectId al nuevo documento.

MongoDB brinda 2 metodos para la inserción de datos:

- **db.colleccion.insertOne()** inserta un solo documento.

```
>>> db.alumnos.insertOne({"nombre": "matias", "legajo": 123})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("604d2639bbe6c6160939d7e4")
}
```

- **db.collection.insertMany()**
para insertar mas de un documento

```
>>> db.alumnos.insertMany([{"alumnos": "Matias", "legajo": 123},
{"alumnos": "Nicolas", "legajo": 1234}])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("604d2a8a759006f14714a54c"),
    ObjectId("604d2a8a759006f14714a54d")
  ]
}
```

Operaciones de lectura

Las operaciones de lectura recuperan documentos de una colección. MongoDB proporciona el método **db.<collection>.find()** para leer los documentos de una colección. Dento del método se pueden especificcar diferentes filtros de consulta.

```
>>> db.alumnos.find({})
{ "_id" : ObjectId("604d2df5b7cf5b788008f227"), "nombre" : "Matias", "legajo" : 123 }
{ "_id" : ObjectId("604d2e2cb7cf5b788008f228"), "nombre" : "Martin", "legajo" : 1234 }
{ "_id" : ObjectId("604d2e2cb7cf5b788008f229"), "nombre" : "Nicolas", "legajo" : 12345 }
}
```

Si se busca obtener un documento en particular, especificamos dentro del método el campo y el valor del mismo.

```
>>> db.alumnos.find({"legajo": 123})
{ "_id" : ObjectId("604d2df5b7cf5b788008f227"), "nombre" : "Matias", "legajo" : 123 }
```

Para modificar el formato del retorno, agregue **".pretty()"**

```
>>> db.alumnos.find({}).pretty()
{
  "_id" : ObjectId("604d2df5b7cf5b788008f227"),
  "nombre" : "Matias",
  "legajo" : 123
}
{
  "_id" : ObjectId("604d2e2cb7cf5b788008f228"),
  "nombre" : "Martin",
  "legajo" : 1234
}
{
  "_id" : ObjectId("604d2e2cb7cf5b788008f229"),
  "nombre" : "Nicolas",
  "legajo" : 12345
}
```

Operaciones de actualización

Se utiliza para modificar documentos existentes , MongoDB proporciona los siguientes métodos.

- **db.collection.updateOne()** actualiza un solo documento

```
>>> db.alumnos.updateOne({ "legajo": 1234 }, { $set: { "nombre": "Lucas" } })
```
- **db.collection.updateMany()** se utiliza para actualizar mas de un documento que cumpla cierta condición

```
db.users.updateMany(  
  { age: { $lt: 18 } },  
  { $set: { status: "reject" } }  
)
```



- **db.collection.replaceOne()** a reemplazar todo el contenido de un documento excepto el campo _id,

Las operaciones de actualización tienen como objetivo una sola colección. Puede especificar criterios o filtros que identifican los documentos a actualizar. Estos filtros utilizan sintaxis que la misma sintaxis que las operaciones de lectura.

MongoDb proporciona los siguientes operadores de actualización.

Nombre	Descripción
\$currentDate	Establece el valor de un campo a la fecha actual, ya sea como Fecha o Marca de tiempo.
\$inc	Incrementa el valor del campo en la cantidad especificada.
\$min	Solo actualiza el campo si el valor especificado es menor que el valor del campo existente.
\$max	Solo actualiza el campo si el valor especificado es mayor que el valor del campo existente.
\$mul	Multiplca el valor del campo por la cantidad especificada.
\$rename	Cambia el nombre de un campo.
\$set	Establece el valor de un campo en un documento.

Nombre	Descripción
\$setOnInsert	Establece el valor de un campo si una actualización da como resultado la inserción de un documento. No tiene ningún efecto sobre las operaciones de actualización que modifican documentos existentes.
\$unset	Elimina el campo especificado de un documento.

Array

Nombre	Descripción
\$	Elimina todos los valores coincidentes de una matriz.
\$[]	Actúa como un marcador de posición para actualizar todos los elementos de una matriz para los documentos que coinciden con la condición de la consulta.
\$[<identifier>]	Actúa como un marcador de posición para actualizar todos los elementos que coinciden con la condición arrayFilters para los documentos que coinciden con la condición de la consulta.
\$addToSet	Agrega elementos a una matriz sólo si aún no existen en el conjunto.
\$pop	Elimina el primer o último elemento de una matriz.
\$pull	Elimina todos los elementos de la matriz que coinciden con una consulta específica.
\$push	Agrega un elemento a una matriz.
\$pullAll	Elimina todos los valores coincidentes de una matriz.

Eliminar documentos

MongoDB proporciona los siguientes métodos y tiene como objetivo eliminar una sola colección:

- **db.collection.deleteOne()** elimina un solo documento que coincida con un filtro especificado.
- **db.collection.deleteMany()** elimina todos los documentos de una colección si el filtro esta vacío, si se agrega un filtro es para limitar esta acción a un grupo específico.

```
>>> db.alumnos.deleteMany({})
```

```
>>> db.alumnos.deleteMany({"legajo":123})  
{ "acknowledged" : true, "deletedCount" : 1 }
```