

Construcción de software y toma de decisiones

TC2005B

Dr. Esteban Castillo Juarez

ITESM, Campus Santa Fe



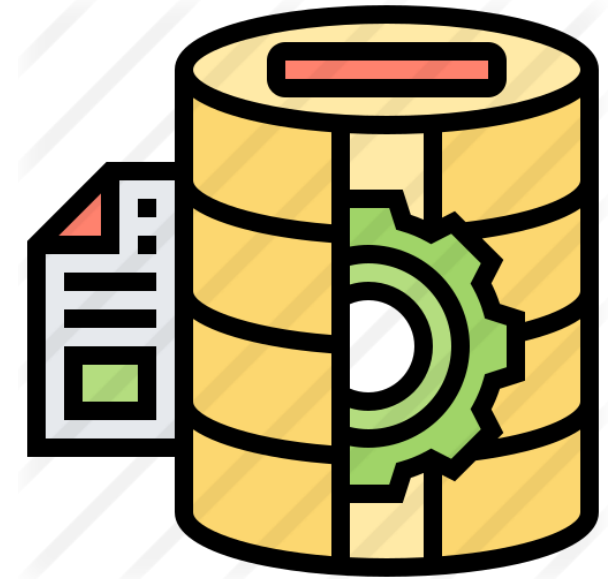
esteban.castillojz@tec.mx

Agenda

- Base de datos relacionales y no-relacionales
- Introducción a MongoDB
- Conceptos básicos de MongoDB
- Conceptos básicos de MongoDB (interfaz grafica)

Base de datos relacionales y no-relacionales

- Una base de datos es una plataforma para almacenar datos de una manera segura, confiable y fácilmente disponible.
- Hay dos tipos de bases de datos que se utilizan en general: bases de datos relacionales (SQL) y bases de datos no relacionales.
- Las bases de datos no relacionales a menudo se denominan bases de datos NoSQL.



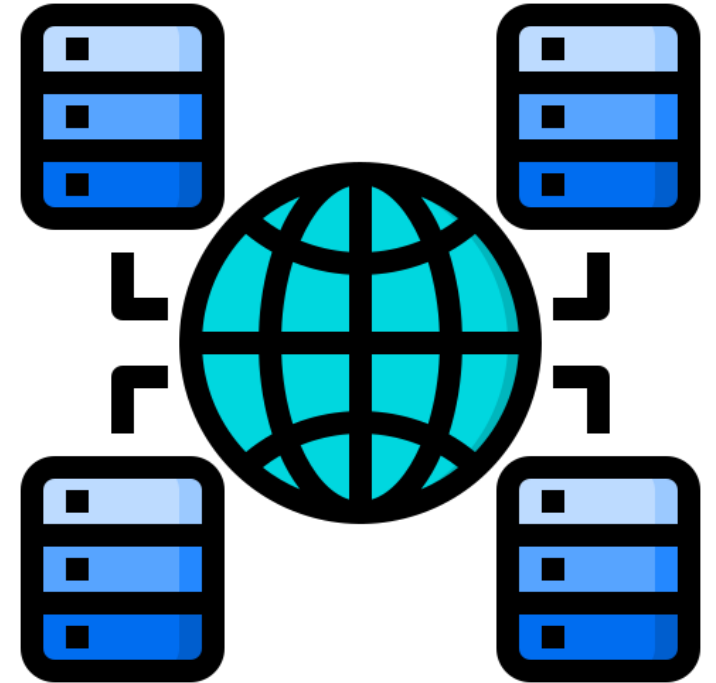
Base de datos relacionales y no-relacionales

- Una base de datos NoSQL se utiliza para almacenar grandes cantidades de datos complejos y diversos, como catálogos de productos, registros, interacciones de usuarios, análisis y más.
- MongoDB es una de las bases de datos NoSQL más establecidas, con funciones como agregación de datos, transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), escalado horizontal y Gráficos.



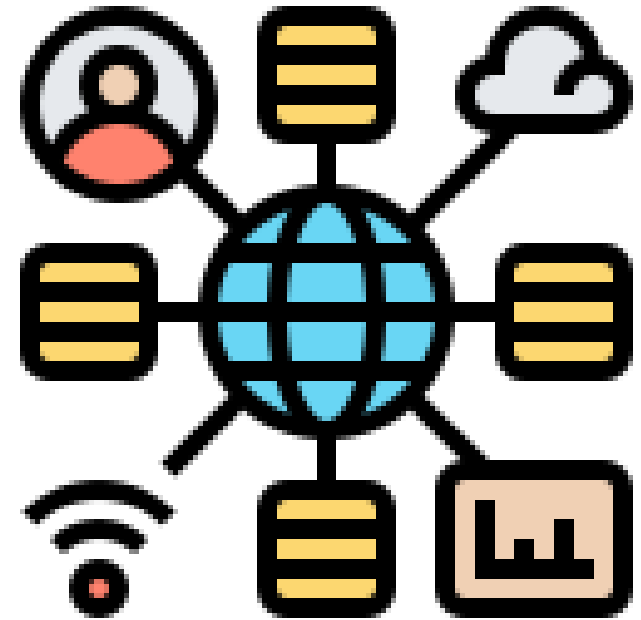
Base de datos relacionales y no-relacionales

- MongoDB viene en diferentes variantes y se puede utilizar tanto para aplicaciones experimentales como del mundo real.
- Es mas fácil de configurar y administrar que la mayoría de las otras bases de datos debido a su sintaxis intuitiva para consultas y comandos.
- MongoDB está disponible para que cualquiera lo instale en su(s) propia(s) máquina(s) o para usarlo en la nube como un servicio administrado.



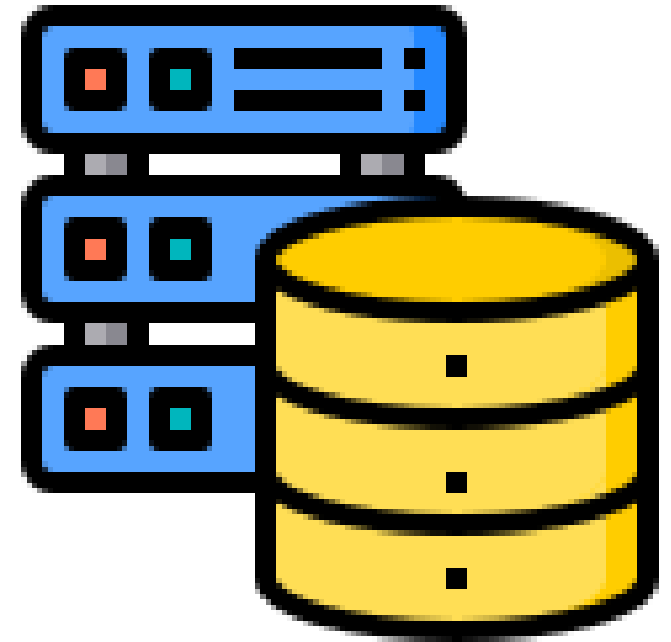
Base de datos relacionales y no-relacionales

- Las bases de datos NoSQL se inventaron para resolver el problema del almacenamiento de datos no estructurados y semiestructurados.
- Las bases de datos relacionales imponen que la estructura de los datos se **defina** antes de que se puedan almacenar información.
- Esta definición de estructura de base de datos a menudo se denomina esquema, que se refiere a las entidades de datos, es decir, sus atributos y tipos.



Base de datos relacionales y no-relacionales

- Las aplicaciones de cliente con un esquema relacional están estrechamente relacionadas con el esquema.
- Es difícil modificar el esquema sin afectar a los clientes.
- Por el contrario, las bases de datos NoSQL permiten almacenar los datos sin un esquema y también admiten un esquema dinámico, que desacopla a los clientes de una estructura rígida y, a menudo, es necesario para aplicaciones modernas y **experimentales**.



Base de datos relacionales y no-relacionales

- Los datos almacenados en una base de datos NoSQL varían según el proveedor, pero generalmente, **los datos se almacenan como documentos en lugar de tablas.**
- Un ejemplo de esto serían las bases de datos para la gestión de inventarios, donde diferentes productos pueden tener diferentes atributos y, por lo tanto, requieren una estructura flexible.
- De manera similar, una base de datos de análisis que almacene datos de diferentes fuentes en diferentes estructuras también necesitaría una estructura flexible.

Base de datos relacionales y no-relacionales

Comparemos las bases de datos NoSQL y SQL según los siguientes factores.

Feature	Relational Database	NoSQL Database
Schema	A relational database follows a rigid schema. The database tables should have a definition of all the desired columns and their types. Any data manipulation that deviates from the schema generates an error.	A NoSQL database does not impose a rigid schema and allows you to store the unstructured data with dynamic structures. This allows an evolving database structure.
Data model/ Storage Structure	The data is stored in tables. Each record is stored as a row that contains information about all the columns. Changing a table can affect the other tables and applications.	The data is stored in different formats, depending on the provider. The standard storage structures are documents, graphs, key-values, and wide columns. There is no alteration required as the database adapts to the dynamic nature of data and the application works seamlessly.
Normalization	Normalization is the process used to remove duplicate data and avoid data anomalies. A relational database prevents data anomalies, using normalization. It requires the data to be stored in different tables and to create relationships among them.	NoSQL databases focus more on fast data retrieval, and the data can be denormalized.
Scaling	Scaling is the ability of a database to grow or reduce in size, depending on the need. Relational databases are hard to scale and are generally scaled vertically, which means increasing the machine compute and storage.	NoSQL databases provide both vertical and horizontal scaling. In horizontal scaling, the data can be distributed across different machines/ clusters.

Introducción a MongoDB

- MongoDB es una base de datos NoSQL popular que puede almacenar datos estructurados y no estructurados.
- Su diseño basado en documentos hace que sea fácil de entender y usar.
- Está diseñado para ser utilizado tanto para aplicaciones experimentales del mundo real y es más fácil de configurar y más simple de administrar que la mayoría de las otras bases de datos NoSQL.



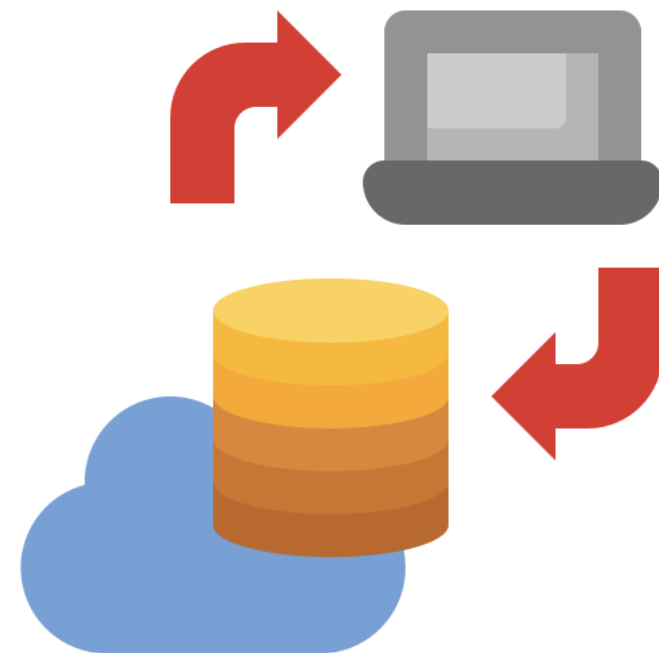
Introducción a MongoDB

La siguiente lista explora las características principales de MongoDB:

Esquema flexible y dinámico: MongoDB permite un esquema flexible para su base de datos.

Un esquema flexible permite la variación de campos en diferentes documentos.

En términos simples, cada registro en la base de datos puede o no tener la misma cantidad de atributos. Aborda la necesidad de almacenar datos en evolución sin realizar ningún cambio en el esquema en sí.



Introducción a MongoDB

La siguiente lista explora las características principales de MongoDB:

Lenguaje de consulta enriquecido: MongoDB admite un lenguaje de consulta intuitivo y enriquecido, lo que significa consultas simples pero potentes.

Viene con un rico marco de agregación que le permite agrupar y filtrar datos según sea necesario.

También tiene soporte incorporado para búsqueda de texto de propósito general y propósitos específicos como búsquedas geoespaciales.



Introducción a MongoDB

La siguiente lista explora las características principales de MongoDB:

Transacciones ACID de varios documentos: La atomicidad, la consistencia, la integridad y la durabilidad (ACID) son características que permiten que sus datos se almacenen y actualicen para mantener su precisión.

Las transacciones se utilizan para combinar operaciones que deben ejecutarse juntas. MongoDB admite ACID en un solo documento y transacciones de varios documentos.



Introducción a MongoDB

La siguiente lista explora las características principales de MongoDB:

Alto rendimiento: MongoDB proporciona un alto rendimiento utilizando modelos de datos integrados para reducir el uso de E/S del disco.

Además, la amplia compatibilidad con la indexación de diferentes tipos de datos hace que las consultas sean más rápidas.

La indexación es un mecanismo para mantener punteros de datos relevantes en un índice como un índice en un libro.

Introducción a MongoDB

La siguiente lista explora las características principales de MongoDB:

Alta disponibilidad: MongoDB admite clústeres distribuidos con un mínimo de tres nodos.

Un clúster hace referencia a una implementación de base de datos que utiliza múltiples nodos/máquinas para el almacenamiento y la recuperación de datos.

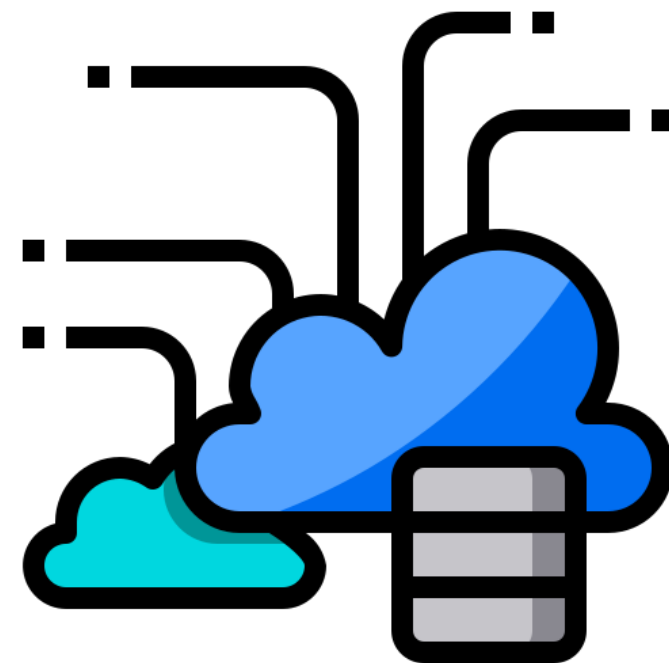
Las conmutaciones por error son automáticas y los datos se replican en nodos secundarios de forma asíncrona.



Introducción a MongoDB

La siguiente lista explora las características principales de MongoDB:

Escalabilidad: MongoDB proporciona una forma de escalar sus bases de datos horizontalmente en cientos de nodos. Entonces, para todas sus necesidades de big data, MongoDB es la solución perfecta.



Conceptos básicos de MongoDB

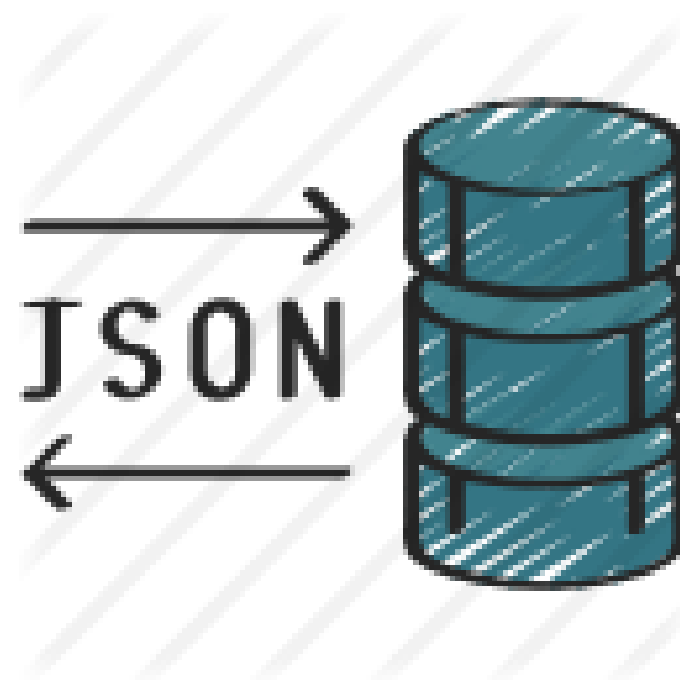
- Ahora, profundicemos en algunos elementos muy básicos de MongoDB, como bases de datos, colecciones y documentos.
- Las bases de datos en MongoDB son básicamente agregaciones de colecciones, que a su vez están formadas por documentos.
- Un documento es el bloque de construcción básico en MongoDB y contiene información sobre los diversos campos en un formato clave-valor.



Conceptos básicos de MongoDB

Documentos en MongoDB

- MongoDB almacena registros de datos en documentos.
- Un documento es una colección de nombres y valores de campos, estructurados en un formato similar a la notación de objetos de JavaScript (JSON).
- JSON es un formato de par clave-valor fácil de entender para describir datos.



Conceptos básicos de MongoDB

Documentos en MongoDB

- Los documentos en MongoDB se almacenan como una extensión del tipo JSON, que se denomina BSON (Binary JSON).
- Es una serialización codificada en binario de documentos similares a JSON.
- BSON está diseñado para ser más eficiente en el espacio que JSON estándar.
- BSON también contiene extensiones que permiten la representación de tipos de datos que no se pueden representar en JSON.

Conceptos básicos de MongoDB

Documentos en MongoDB

Los documentos de MongoDB contienen pares de campo y valor y siguen una estructura básica, de la siguiente manera:

```
{  
  "firstName": firstNameValue,  
  "secondFieldName": secondFieldValue,  
  ...  
  "nthFieldName": nthFieldValue  
}
```

Conceptos básicos de MongoDB

Documentos en MongoDB

El siguiente es un ejemplo de un documento que contiene detalles sobre una persona:

```
{
  "_id": ObjectId("5da26111139a21bbe11f9e89"),
  "name": "Anita P",
  "placeOfBirth": "Koszalin",
  "profession": "Nursing"
}
```

Conceptos básicos de MongoDB

Documentos en MongoDB

El siguiente es otro ejemplo con algunos campos y tipos de fecha de BSON:

```
{
  "_id" :
  ObjectId("5da26553fb4ef99de45a6139"),
  "name" : "Roxana",
  "dateOfBirth" : new Date("Dec 25,
2007"),
  "placeOfBirth" : "Brisbane",
  "profession" : "Student"
}
```

Conceptos básicos de MongoDB

Documentos en MongoDB

- El siguiente ejemplo de un documento contiene una matriz y un subdocumento.
- Una arreglo es un conjunto de valores y se puede usar cuando necesita almacenar varios valores para una clave, como los pasatiempos.
- Los subdocumentos permiten envolver atributos relacionados en un documento con una clave, como los es una dirección en el siguiente ejemplo:

Conceptos básicos de MongoDB

Documentos en MongoDB

```
"_id" :  
ObjectId("5da2685bfb4ef99de45a613a"),  
  "name" : "Helen",  
  "dateOfBirth" : new Date("Dec 25,  
2007"),  
  "placeOfBirth" : "Brisbane",  
  "profession" : "Student",  
  "hobbies" : [  
    "painting",  
    "football",  
    "singing",  
    "story-writing"],  
  "address" : {  
    "city" : "Sydney",  
    "country" : "Australia",  
    "postcode" : 2161  
  }  
}
```

El campo `_id` que se muestra en el fragmento anterior lo genera automáticamente MongoDB y se utiliza como un identificador único para el documento.

Conceptos básicos de MongoDB

Documentos en MongoDB

- En MongoDB, los documentos se almacenan en colecciones.
- Las colecciones son análogas a las tablas en las bases de datos relacionales.
- Se debe usar el nombre de la colección en las consultas para operaciones como insertar, recuperar, eliminar, etc.



Conceptos básicos de MongoDB

Arquitectura de MongoDB

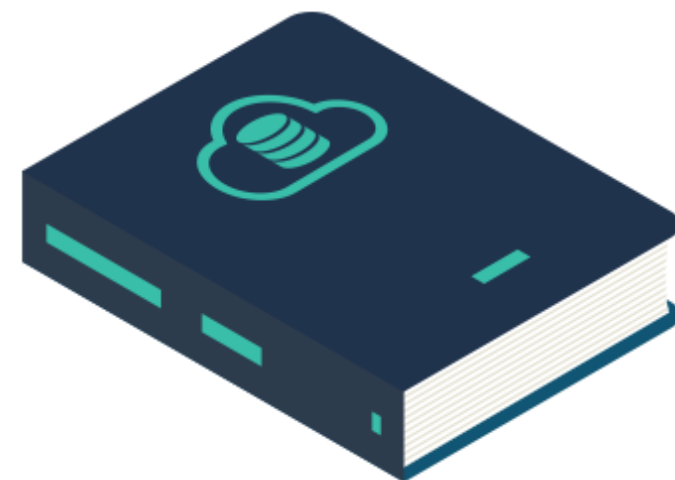
- Una base de datos es un contenedor de colecciones agrupadas.
- Cada esquema de MongoDB tiene varios ficheros que contienen metadatos y los “datos” reales almacenados.
- MongoDB permite tener múltiples bases de datos, y cada una de estas bases de datos puede tener varias colecciones.



Conceptos básicos de MongoDB

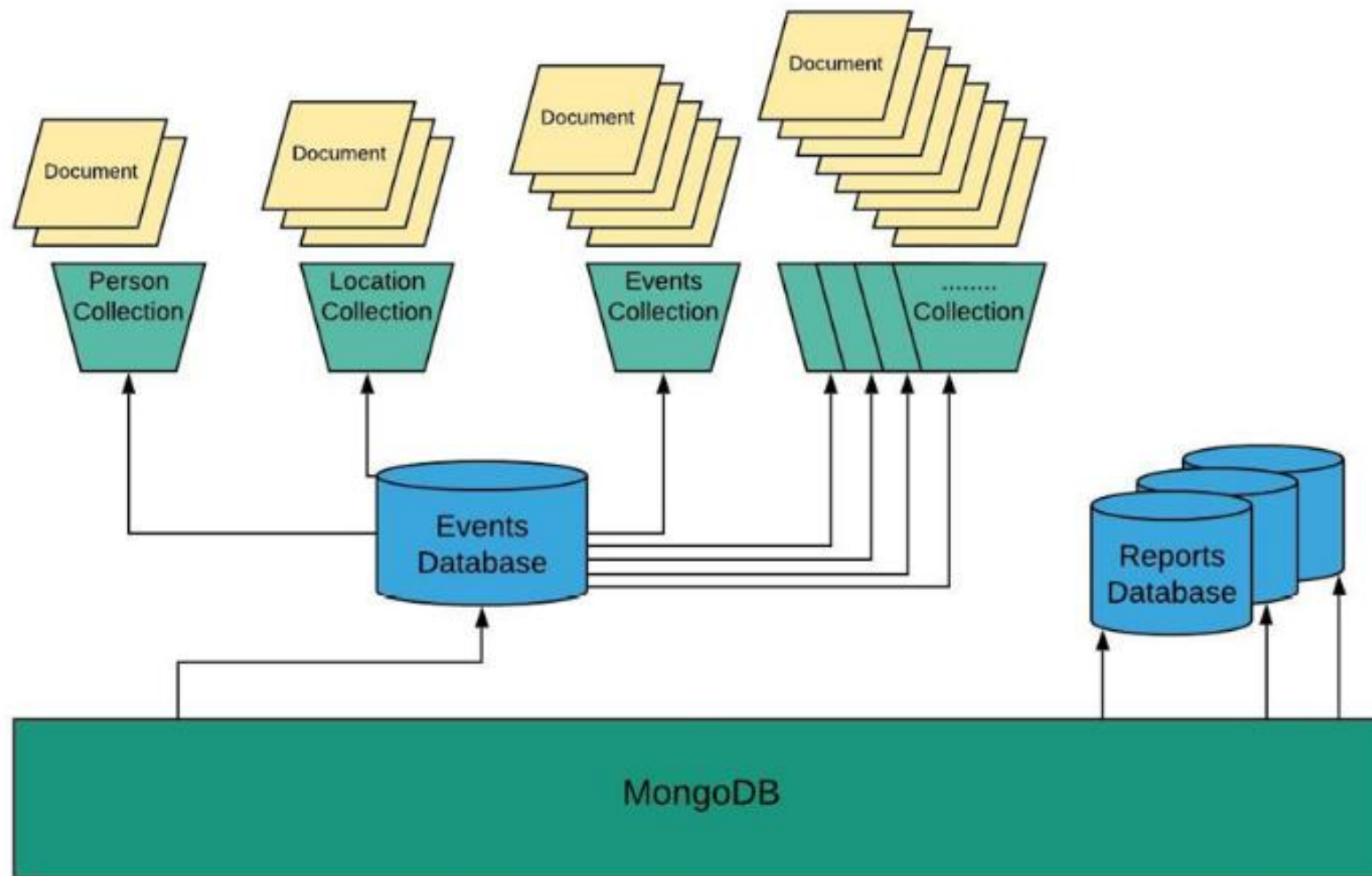
Arquitectura de MongoDB

- A su vez, cada una de las colecciones antes mencionadas puede tener numerosos documentos.
- Esto se ilustra en la siguiente figura, que muestra una base de datos que contiene colecciones para diferentes campos relacionados con eventos, como Persona o Ubicación; estos, a su vez, contienen varios documentos con todos los datos.



Conceptos básicos de MongoDB

Arquitectura de MongoDB



Conceptos básicos de MongoDB

Creación de una base de datos

Primero, debemos tener abierto el servidor de MongoDB (mongod).

```
C:\Program Files\MongoDB\Server\5.0\bin>mongod.exe

stamp monitor starting"}
{"t":{"$date":"2022-05-14T12:56:14.126-05:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2022-05-14T12:56:14.126-05:00"},"s":"W", "c":"CONTROL", "id":22140, "ctx":"initandlisten","msg":"This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning","tags":["startupWarnings"]}
{"t":{"$date":"2022-05-14T12:56:14.139-05:00"},"s":"I", "c":"NETWORK", "id":4915702, "ctx":"initandlisten","msg":"Updated wire specification","attr":{"oldSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":true},"newSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":13,"maxWireVersion":13},"outgoing":{"minWireVersion":13,"maxWireVersion":13},"isInternalClient":true}}}
{"t":{"$date":"2022-05-14T12:56:14.140-05:00"},"s":"I", "c":"STORAGE", "id":5071100, "ctx":"initandlisten","msg":"Clearing temp directory"}
{"t":{"$date":"2022-05-14T12:56:14.141-05:00"},"s":"I", "c":"CONTROL", "id":20536, "ctx":"initandlisten","msg":"Flow Control is enabled on this deployment"}
{"t":{"$date":"2022-05-14T12:56:14.424-05:00"},"s":"W", "c":"FTDC", "id":23718, "ctx":"initandlisten","msg":"Failed to initialize Performance Counters for FTDC","attr":{"error":{"code":179,"codeName":"WindowsPdhError","errmsg":"PdhAddEnglishCounterW failed with 'El objeto especificado no se encontró en el equipo.'"}}}
{"t":{"$date":"2022-05-14T12:56:14.425-05:00"},"s":"I", "c":"FTDC", "id":20625, "ctx":"initandlisten","msg":"Initializing full-time diagnostic data capture","attr":{"dataDirectory":"C:/data/db/diagnostic.data"}}
{"t":{"$date":"2022-05-14T12:56:14.435-05:00"},"s":"I", "c":"REPL", "id":6015317, "ctx":"initandlisten","msg":"Setting new configuration state","attr":{"newState":"ConfigReplicationDisabled","oldState":"ConfigPreStart"}}
{"t":{"$date":"2022-05-14T12:56:14.439-05:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Listening on","attr":{"address":"127.0.0.1"}}
{"t":{"$date":"2022-05-14T12:56:14.439-05:00"},"s":"I", "c":"NETWORK", "id":23016, "ctx":"listener","msg":"Waiting for connections","attr":{"port":27017,"ssl":"off"}}
```

Conceptos básicos de MongoDB

Creación de una base de datos

- Crear una base de datos en MongoDB es muy sencillo.
- Ejecute el comando `use` en el shell mongo de la siguiente manera, reemplazando `nombreBaseDatos` con su propia elección de nombre de base de datos:

```
use nombreBaseDatos
```

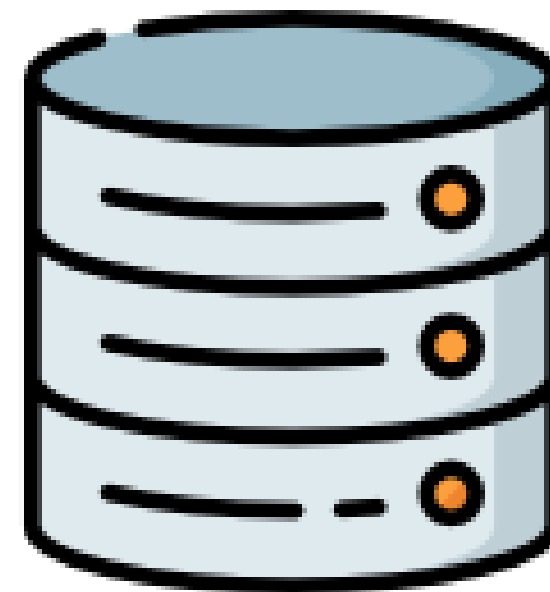


Conceptos básicos de MongoDB

Creación de una base de datos

- Si la base de datos no existe, MongoDB creará una nueva y cambiará la base de datos actual a la nueva base de datos.
- Si la base de datos existe, MongoDB se referirá a la base de datos existente. Aquí está la salida del último comando:

```
switched to db nombreBaseDatos
```

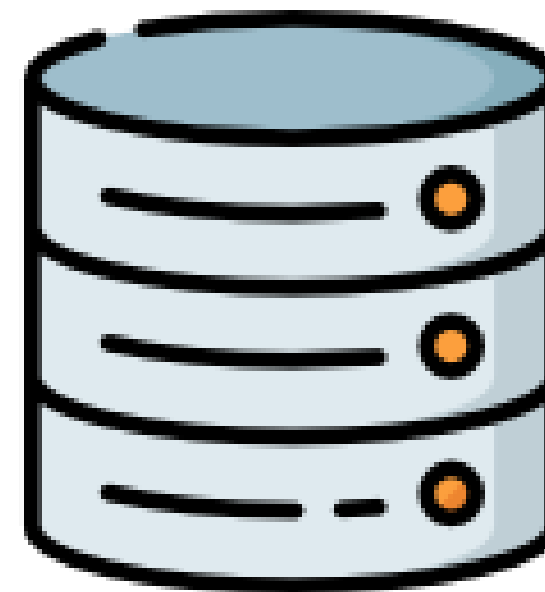


Conceptos básicos de MongoDB

Crear una colección

- Si la base de datos no existe, MongoDB creará una nueva y cambiará la base de datos actual a la nueva base de datos.
- Si la base de datos existe, MongoDB se referirá a la base de datos existente. Aquí está la salida del último comando:

```
switched to db nombreBaseDatos
```



Conceptos básicos de MongoDB

Crear una colección

- Puede usar el comando `createCollection` para crear una colección.
- Este comando permite utilizar diferentes opciones para la creación de una colección, como una colección limitada, validación, intercalación de datos, etc.
- Otra forma de crear una colección es simplemente insertando un documento en una colección inexistente.



Conceptos básicos de MongoDB

Crear una colección

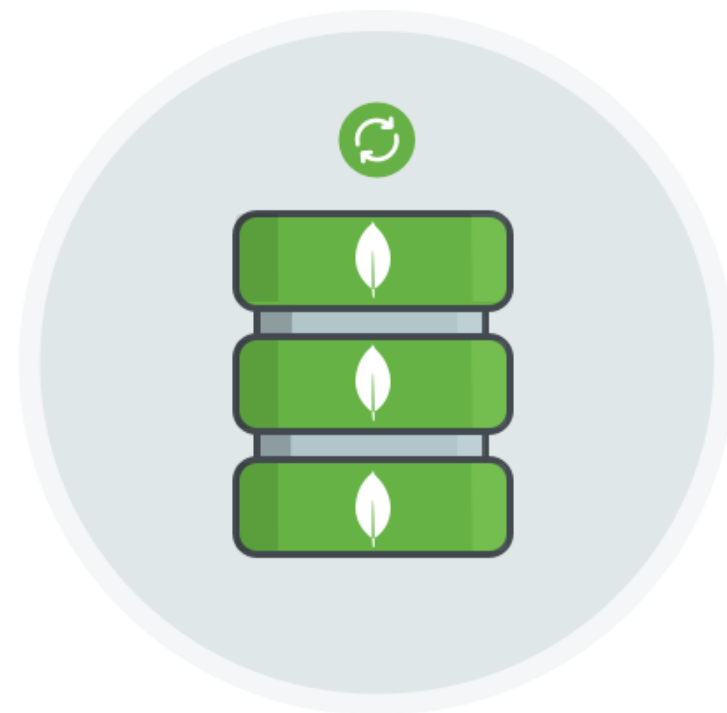
- Considerando el caso anterior, MongoDB verifica si la colección existe y, de no ser así, creará la colección antes de insertar los documentos dados al sistema.
- Para crear una colección explícitamente, se puede utilizar la operación `createCollection` de la siguiente manera:

```
db.createCollection( '<collectionName>',  
{  
    capped: <boolean>,  
    autoIndexId: <boolean>,  
    size: <number>,  
    max: <number>,  
    storageEngine: <document>,  
    validator: <document>,  
    validationLevel: <string>,  
    validationAction: <string>,  
    indexOptionDefaults: <document>,  
    viewOn: <string>,  
    pipeline: <pipeline>,  
    collation: <document>,  
    writeConcern: <document>  
})
```

Conceptos básicos de MongoDB

Crear una colección

- En la figura siguiente, estamos creando una colección finita (capped collection) con un máximo de 5 documentos, y cada documento tiene un límite de tamaño de 256 bytes.
- La colección limitada funciona como una cola circular, lo que significa que los documentos más antiguos saldrán para hacer espacio para las últimas inserciones cuando se alcance el tamaño máximo:



Conceptos básicos de MongoDB

Crear una colección

Aquí está el resultado del comando createCollection:

```
db.createCollection('myCappedCollection',  
{  
  capped: true,  
  size: 256,  
  max: 5  
})
```



```
{  
  «ok» : 1,  
  «$clusterTime» : {  
    «clusterTime» :  
      Timestamp(1592064731, 1),  
    «signature» : {  
      «hash» :  
        BinData(0,»XJ2D0zjAagUkftFkLQIT  
        9W2rKjc="),  
      «keyId» :  
        NumberLong(«6834058563036381187»)   
    }  
  },  
  «operationTime» :  
    Timestamp(1592064731, 1)  
}
```

Conceptos básicos de MongoDB

Crear una colección

- No debemos de preocuparnos por las opciones anteriores ya que ninguna de ellas es obligatoria.
- Si no necesita configurar ninguno de estos, entonces el comando `createCollection` se puede simplificar de la siguiente manera:

```
db.createCollection('myFirstCollection')
```



Conceptos básicos de MongoDB

Crear una colección

```
db.createCollection('myFirstCollection')
```

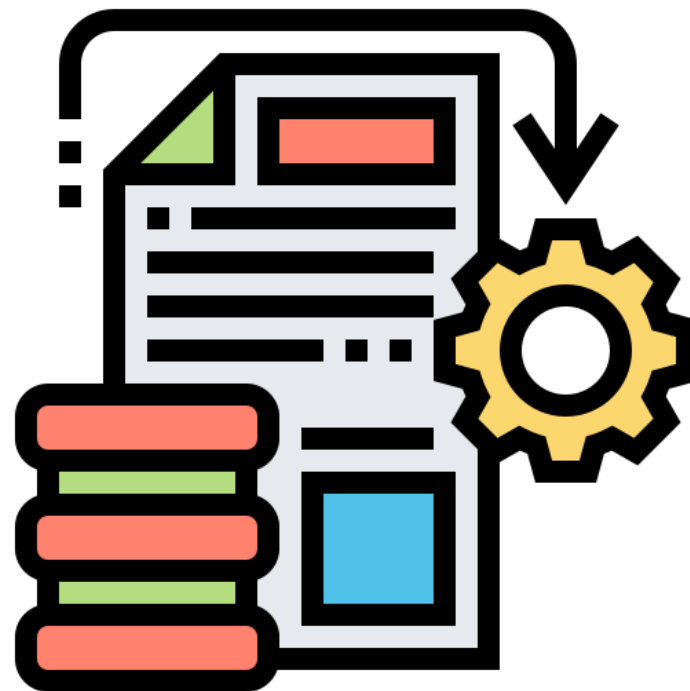


```
{
  «ok» : 1,
  «$clusterTime» : {
    «clusterTime» :
      Timestamp(1597230876, 1),
    «signature» : {
      «hash» :
        BinData(0,»Y08Flg5AglrxCV3XqEuZG
        aaLzZc="),
      «keyId» :
        NumberLong(«6853300587753111555»)
    }
  },
  «operationTime» :
    Timestamp(1597230876, 1)
}
```

Conceptos básicos de MongoDB

Creación de una colección mediante la inserción de documentos

- No se necesita crear una colección antes de insertar documentos.
- MongoDB crea una colección si no existe en la primera inserción del documento.
- Este método funciona de la siguiente manera:

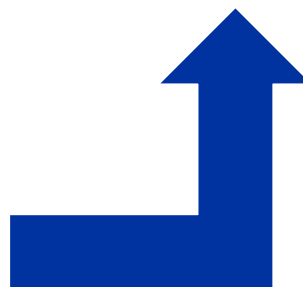


Conceptos básicos de MongoDB

Creación de una colección mediante la inserción de documentos

```
use yourDatabaseName;  
db.myCollectionName.insert(  
  {  
    "name" : "Yahya A", "company" :  
    "Sony"}  
);
```

```
WriteResult({ "nInserted" : 1 })
```



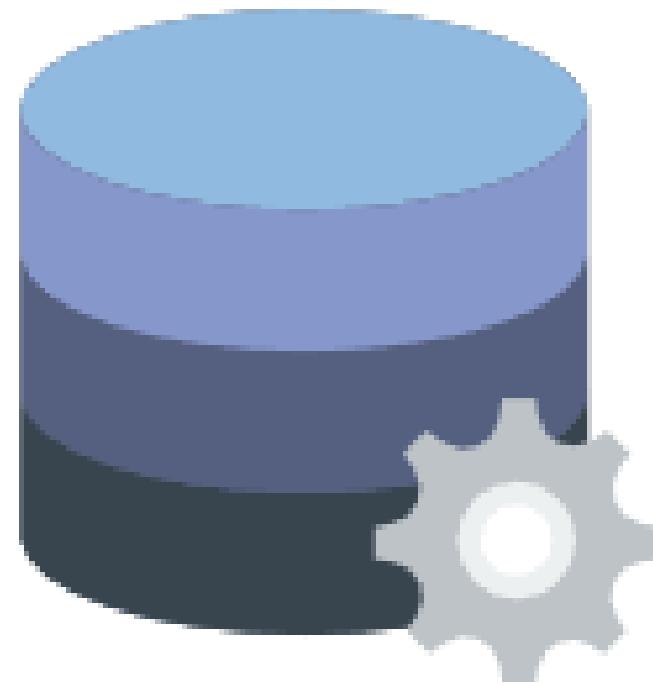
El resultado anterior devuelve el número de documentos insertados en la colección.

Conceptos básicos de MongoDB

Creación de una colección mediante la inserción de documentos

- Como se ha insertado un documento en una colección inexistente, MongoDB debe haber creado la colección para nosotros antes de insertar un documento.
- Para confirmar lo anterior, se muestra la lista de colecciones usando el siguiente comando:

```
show collections;
```



Conceptos básicos de MongoDB

Creación de documentos

- Usamos el comando `insert` para poner un documento en una colección.
- El comando `insertOne` se usa para insertar un documento a la vez, como en el siguiente ejemplo:

```
db.blogs.insertOne(  
  { username: "Zakariya", noOfBlogs: 100,  
    tags: ["science", "fiction"]  
})
```

La operación `insertOne` devuelve el valor `_id` del documento recién insertado.

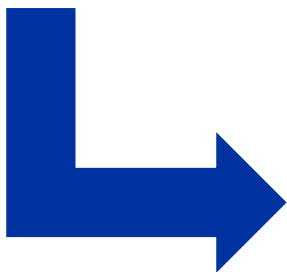
Conceptos básicos de MongoDB

Creación de documentos

El comando `insertOne` se usa para insertar un documento a la vez, como en el siguiente ejemplo:

```
db.blogs.insertOne(  
  { username: "Zakariya", noOfBlogs: 100,  
    tags: ["science", "fiction"]  
  })
```

`insertedId` es el ID exclusivo del documento que se inserta y no será el mismo para cada usuario que ejecuta la instrucción.



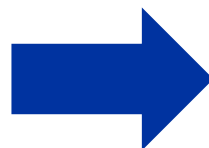
```
{  
  "acknowledged" : true,  
  "insertedId" :  
  ObjectId("5ea3a1561df5c3fd4f752636")  
}
```

Conceptos básicos de MongoDB

Creación de documentos

- El comando `insertMany` agrega varios documentos a la vez.
- Puede pasar una arreglo de documentos como se menciona en la siguiente figura:

```
db.blogs.insertMany(  
  [  
    { username: "Thaha", noOfBlogs:  
      200, tags: ["science", "robotics"]},  
    { username: "Thayebbah", noOfBlogs:  
      500, tags: ["cooking", "general  
knowledge"]},  
    { username: "Thaherah", noOfBlogs:  
      50, tags: ["beauty", "arts"]}]  
)
```



```
{  
  «acknowledged» : true,  
  «insertedIds» : [  
    ObjectId(«5f33cf74592962df72246ae8»),  
    ObjectId(«5f33cf74592962df72246ae9»),  
    ObjectId(«5f33cf74592962df72246aea»)  
  ]  
}
```

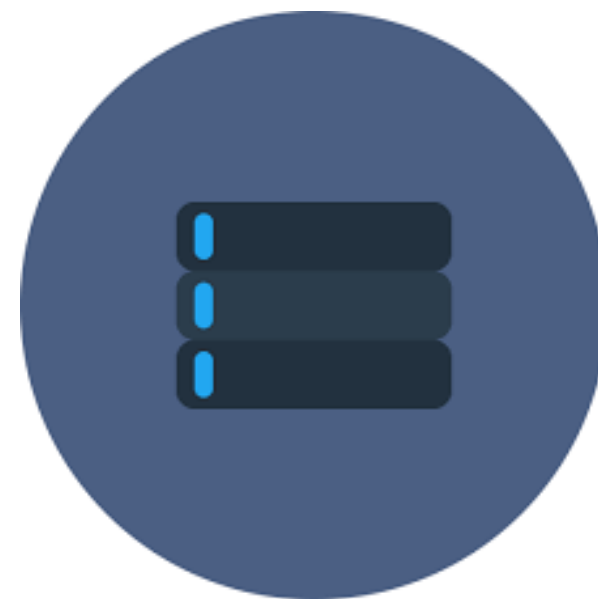
La salida devuelve los valores `_id` de todos los documentos recién insertados.

Conceptos básicos de MongoDB

Obtener documentos de MongoDB

- MongoDB proporciona el comando `find` para obtener documentos de una colección.
- Este comando es útil para verificar si las inserciones previas están realmente guardadas en las colecciones. Aquí está la sintaxis para el comando de búsqueda:

```
db.collection.find(query, projection)
```



Conceptos básicos de MongoDB

Obtener documentos de MongoDB

- El comando toma dos parámetros opcionales: `query` y `projection`.
- El parámetro `query` permite pasar un documento para aplicar filtros durante la operación de búsqueda.
- El parámetro de `projection` permite elegir los atributos deseados de los documentos devueltos en lugar de todos los atributos.
- Cuando no se pasa ningún parámetro en el comando de búsqueda, se devuelven todos los documentos.

Conceptos básicos de MongoDB

Obtener documentos de MongoDB

- Cuando no se pasa ningún parámetro en el comando de búsqueda, se devuelven todos los documentos.

```
> db.records.find()
{ "_id" : ObjectId("628049513d6bad3b35b4470c"), "Name" : "Robert A", "City" : "Sydney" }
{ "_id" : ObjectId("628049513d6bad3b35b4470d"), "Name" : "Lucas B", "City" : "New Delhi" }
```

- El parámetro query permite pasar un documento para aplicar filtros durante la operación de búsqueda.

```
> db.records.find()
{ "_id" : ObjectId("628049513d6bad3b35b4470c"), "Name" : "Robert A", "City" : "Sydney" }
{ "_id" : ObjectId("628049513d6bad3b35b4470d"), "Name" : "Lucas B", "City" : "New Delhi" }
> db.records.find({City: "Sydney"})
{ "_id" : ObjectId("628049513d6bad3b35b4470c"), "Name" : "Robert A", "City" : "Sydney" }
```

Conceptos básicos de MongoDB

Obtener documentos de MongoDB usando aggregate()

Filtra los documentos para pasar solo aquellos que coinciden con una condición especificada a una posterior etapa de canalización/análisis.

```
> db.records.find()
{ "_id" : ObjectId("628049513d6bad3b35b4470c"), "Name" : "Robert A", "City" : "Sydney" }
{ "_id" : ObjectId("628049513d6bad3b35b4470d"), "Name" : "Lucas B", "City" : "New Delhi" }
> db.records.aggregate([ { $match : { City : "Sydney" } } ]);
{ "_id" : ObjectId("628049513d6bad3b35b4470c"), "Name" : "Robert A", "City" : "Sydney" }
```


Conceptos básicos de MongoDB

Formateo de la salida de búsqueda usando el método `pretty()`

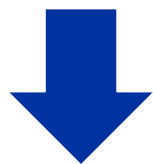
- Cuando el comando de búsqueda devuelve varios registros, a veces es difícil leerlos porque no tienen el formato adecuado.
- MongoDB proporciona el método `pretty()` al final del comando de búsqueda para obtener los registros devueltos de forma formateada.
- Para verlo en acción, podemos insertar un par de elementos en una colección llamada registros.



Conceptos básicos de MongoDB

Formateo de la salida de búsqueda usando el método pretty()

```
> use records
switched to db records
> db.records.insertMany( [ { Name: "Robert A", City: "Sydney"}, { Name: "Lucas B", City: "New Delhi"} ] )
```



```
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("628049513d6bad3b35b4470c"),
    ObjectId("628049513d6bad3b35b4470d")
  ]
}
```

Debería generar una salida de la siguiente manera:

Conceptos básicos de MongoDB

Formateo de la salida de búsqueda usando el método pretty()

- Primero, obtenga estos registros usando el comando de búsqueda sin el método pretty: `db.records.find()`
- Debería devolver una salida como se muestra aquí:

```
> pretty: db.records.find()
{ "_id" : ObjectId("628047bd3d6bad3b35b44707"), "Name" : "Naseem A", "City" : "New Delhi" }
{ "_id" : ObjectId("628047df3d6bad3b35b44709"), "Name" : "Lucas B", "City" : "New Delhi" }
{ "_id" : ObjectId("628048c43d6bad3b35b4470a"), "Name" : "Robert A", "City" : "Sydney" }
{ "_id" : ObjectId("628048c43d6bad3b35b4470b"), "Name" : "Lucas B", "City" : "New Delhi" }
```

Conceptos básicos de MongoDB

Eliminar documentos usando el método deleteOne()

Para eliminar como máximo un único documento que coincida con un filtro específico (aunque varios documentos puedan coincidir con el filtro especificado), utilice el método `db.collection.deleteOne()`.

```
> db.records.find()
{ "_id" : ObjectId("628044323d6bad3b35b44704"), "Name" : "Aaliya A", "City" : "Sydney" }
{ "_id" : ObjectId("628044323d6bad3b35b44705"), "Name" : "Naseem A", "City" : "New Delhi" }
> db.records.deleteOne( { City: "Sydney" } )
{ "acknowledged" : true, "deletedCount" : 1 }
> db.records.find()
{ "_id" : ObjectId("628044323d6bad3b35b44705"), "Name" : "Naseem A", "City" : "New Delhi" }
```

Conceptos básicos de MongoDB

Eliminar todos los documentos usando el método deleteMany()

Puede especificar criterios o filtros que identifiquen los documentos que desea eliminar. Los filtros utilizan la misma sintaxis que las operaciones de lectura. Para hacer lo anterior), utilice el método `db.collection.deleteMany()`.

```
> db.records.find()
{ "_id" : ObjectId("628044323d6bad3b35b44705"), "Name" : "Naseem A", "City" : "New Delhi" }
{ "_id" : ObjectId("628047bd3d6bad3b35b44706"), "Name" : "Aaliya A", "City" : "Sydney" }
{ "_id" : ObjectId("628047bd3d6bad3b35b44707"), "Name" : "Naseem A", "City" : "New Delhi" }
{ "_id" : ObjectId("628047df3d6bad3b35b44708"), "Name" : "Robert A", "City" : "Sydney" }
{ "_id" : ObjectId("628047df3d6bad3b35b44709"), "Name" : "Lucas B", "City" : "New Delhi" }
> db.records.deleteOne( { Name: "Naseem A" } )
{ "acknowledged" : true, "deletedCount" : 1 }
> db.records.find()
{ "_id" : ObjectId("628047bd3d6bad3b35b44706"), "Name" : "Aaliya A", "City" : "Sydney" }
{ "_id" : ObjectId("628047bd3d6bad3b35b44707"), "Name" : "Naseem A", "City" : "New Delhi" }
{ "_id" : ObjectId("628047df3d6bad3b35b44708"), "Name" : "Robert A", "City" : "Sydney" }
{ "_id" : ObjectId("628047df3d6bad3b35b44709"), "Name" : "Lucas B", "City" : "New Delhi" }
> db.records.deleteMany({ City : "Sydney" })
{ "acknowledged" : true, "deletedCount" : 2 }
> db.records.find()
{ "_id" : ObjectId("628047bd3d6bad3b35b44707"), "Name" : "Naseem A", "City" : "New Delhi" }
{ "_id" : ObjectId("628047df3d6bad3b35b44709"), "Name" : "Lucas B", "City" : "New Delhi" }
```

Conceptos básicos de MongoDB

Crear un índice utilizando el método `createIndex()`

Se pueden crear columnas optimizadas para realizar consultas mas rápidas. Esto a través del método `db.collection.createIndex()` (veremos como crear estos mas adelante).

```
> db.records.find()
{ "_id" : ObjectId("628049513d6bad3b35b4470c"), "Name" : "Robert A", "City" : "Sydney" }
{ "_id" : ObjectId("628049513d6bad3b35b4470d"), "Name" : "Lucas B", "City" : "New Delhi" }
{ "_id" : ObjectId("62804fb63d6bad3b35b4470e"), "Name" : "Robert A", "City" : "Sydney" }
{ "_id" : ObjectId("62804fb63d6bad3b35b4470f"), "Name" : "Lucas B", "City" : "New Delhi" }
> db.records.aggregate([ { $match : { City : "Sydney" } } ]);
{ "_id" : ObjectId("628049513d6bad3b35b4470c"), "Name" : "Robert A", "City" : "Sydney" }
{ "_id" : ObjectId("62804fb63d6bad3b35b4470e"), "Name" : "Robert A", "City" : "Sydney" }
> db.records.createIndex( { City: "text" } )
{
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "createdCollectionAutomatically" : false,
  "ok" : 1
}
```

Conceptos básicos de MongoDB (interfaz grafica)

Creación de una base de datos en MongoDB compass

Nuevamente, debemos revisar si tenemos abierto el servidor de MongoDB (mongod).

```
C:\Program Files\MongoDB\Server\5.0\bin\mongod.exe
stamp monitor starting"}
{"t":{"$date":"2022-05-14T12:56:14.126-05:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2022-05-14T12:56:14.126-05:00"},"s":"W", "c":"CONTROL", "id":22140, "ctx":"initandlisten","msg":"This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning","tags":["startupWarnings"]}
{"t":{"$date":"2022-05-14T12:56:14.139-05:00"},"s":"I", "c":"NETWORK", "id":4915702, "ctx":"initandlisten","msg":"Updated wire specification","attr":{"oldSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":true},"newSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":13,"maxWireVersion":13},"outgoing":{"minWireVersion":13,"maxWireVersion":13},"isInternalClient":true}}}
{"t":{"$date":"2022-05-14T12:56:14.140-05:00"},"s":"I", "c":"STORAGE", "id":5071100, "ctx":"initandlisten","msg":"Clearing temp directory"}
{"t":{"$date":"2022-05-14T12:56:14.141-05:00"},"s":"I", "c":"CONTROL", "id":20536, "ctx":"initandlisten","msg":"Flow Control is enabled on this deployment"}
{"t":{"$date":"2022-05-14T12:56:14.424-05:00"},"s":"W", "c":"FTDC", "id":23718, "ctx":"initandlisten","msg":"Failed to initialize Performance Counters for FTDC","attr":{"error":{"code":179,"codeName":"WindowsPdhError","errmsg":"PdhAddEnglishCounterW failed with 'El objeto especificado no se encontró en el equipo.'"}}}
{"t":{"$date":"2022-05-14T12:56:14.425-05:00"},"s":"I", "c":"FTDC", "id":20625, "ctx":"initandlisten","msg":"Initializing full-time diagnostic data capture","attr":{"dataDirectory":"C:/data/db/diagnostic.data"}}
{"t":{"$date":"2022-05-14T12:56:14.435-05:00"},"s":"I", "c":"REPL", "id":6015317, "ctx":"initandlisten","msg":"Setting new configuration state","attr":{"newState":"ConfigReplicationDisabled","oldState":"ConfigPreStart"}}
{"t":{"$date":"2022-05-14T12:56:14.439-05:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Listening on","attr":{"address":"127.0.0.1"}}
{"t":{"$date":"2022-05-14T12:56:14.439-05:00"},"s":"I", "c":"NETWORK", "id":23016, "ctx":"listener","msg":"Waiting for connections","attr":{"port":27017,"ssl":"off"}}
```


Conceptos básicos de MongoDB (interfaz grafica)

Creación de una base de datos en MongoDB compass

Si no esta abierto el servidor, MongoDB Compass marcara un error de conexión.

New Connection

Connect to a MongoDB deployment



FAVORITE

URI ⓘ

Edit Connection String ☒

mongodb://localhost:27017

➤ Advanced Connection Options



connect ECONNREFUSED 127.0.0.1:27017

Connect



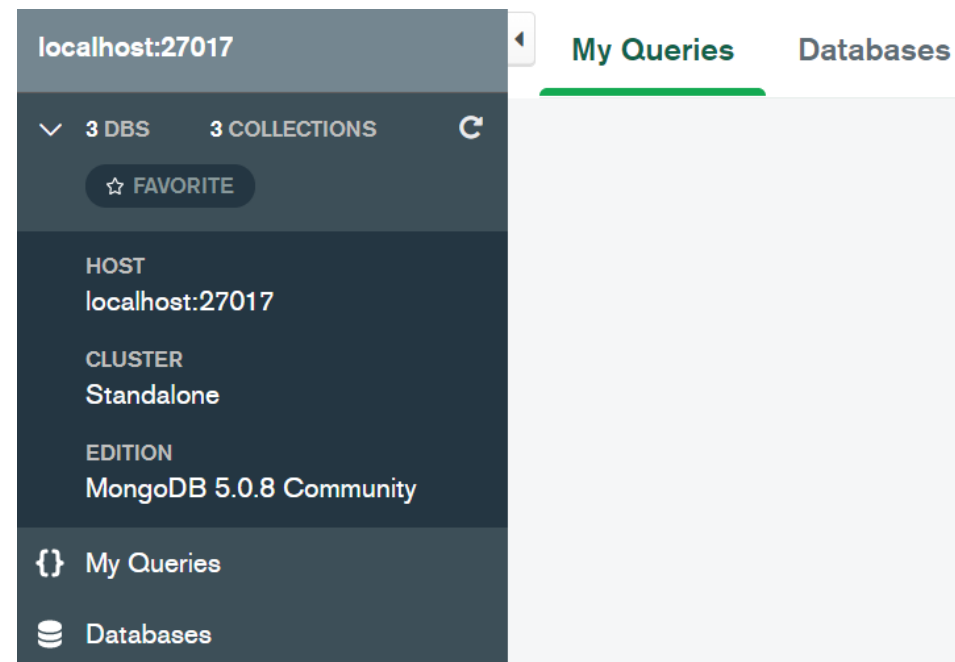
Tecnológico
de Monterrey

Conceptos básicos de MongoDB (interfaz grafica)

Creación de una base de datos en MongoDB compass

Si no esta abierto el servidor, MongoDB Compass marcara un error de conexión.

```
C:\Program Files\MongoDB\Server\5.0\bin>mongod.exe
stamp monitor starting
{"t":{"$date":"2022-05-14T12:56:14.126-05:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2022-05-14T12:56:14.126-05:00"},"s":"W", "c":"CONTROL", "id":22140, "ctx":"initandlisten","msg":"This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning","tags":["startupWarnings"]}
{"t":{"$date":"2022-05-14T12:56:14.139-05:00"},"s":"I", "c":"NETWORK", "id":4915702, "ctx":"initandlisten","msg":"Updated wire specification","attr":{"oldSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":true},"newSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":13,"maxWireVersion":13},"outgoing":{"minWireVersion":13,"maxWireVersion":13},"isInternalClient":true}}}
{"t":{"$date":"2022-05-14T12:56:14.140-05:00"},"s":"I", "c":"STORAGE", "id":5071100, "ctx":"initandlisten","msg":"Clearing temp directory"}
{"t":{"$date":"2022-05-14T12:56:14.141-05:00"},"s":"I", "c":"CONTROL", "id":20536, "ctx":"initandlisten","msg":"Flow control is enabled on this deployment"}
{"t":{"$date":"2022-05-14T12:56:14.424-05:00"},"s":"W", "c":"FTDC", "id":23718, "ctx":"initandlisten","msg":"Failed to initialize Performance Counters for FTDC","attr":{"error":{"code":179,"codeName":"WindowsPdhError","errmsg":"PdhAddEnglishCounterW failed with 'El objeto especificado no se encontró en el equipo.'"}}}
{"t":{"$date":"2022-05-14T12:56:14.425-05:00"},"s":"I", "c":"FTDC", "id":20625, "ctx":"initandlisten","msg":"Initializing full-time diagnostic data capture","attr":{"dataDirectory":"C:/data/db/diagnostic.data"}}
{"t":{"$date":"2022-05-14T12:56:14.435-05:00"},"s":"I", "c":"REPL", "id":6015317, "ctx":"initandlisten","msg":"Setting new configuration state","attr":{"newState":"ConfigReplicationDisabled","oldState":"ConfigPreStart"}}
{"t":{"$date":"2022-05-14T12:56:14.439-05:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Listening on","attr":{"address":"127.0.0.1"}}
{"t":{"$date":"2022-05-14T12:56:14.439-05:00"},"s":"I", "c":"NETWORK", "id":23016, "ctx":"listener","msg":"Waiting for connections","attr":{"port":27017,"ssl":"off"}}
```



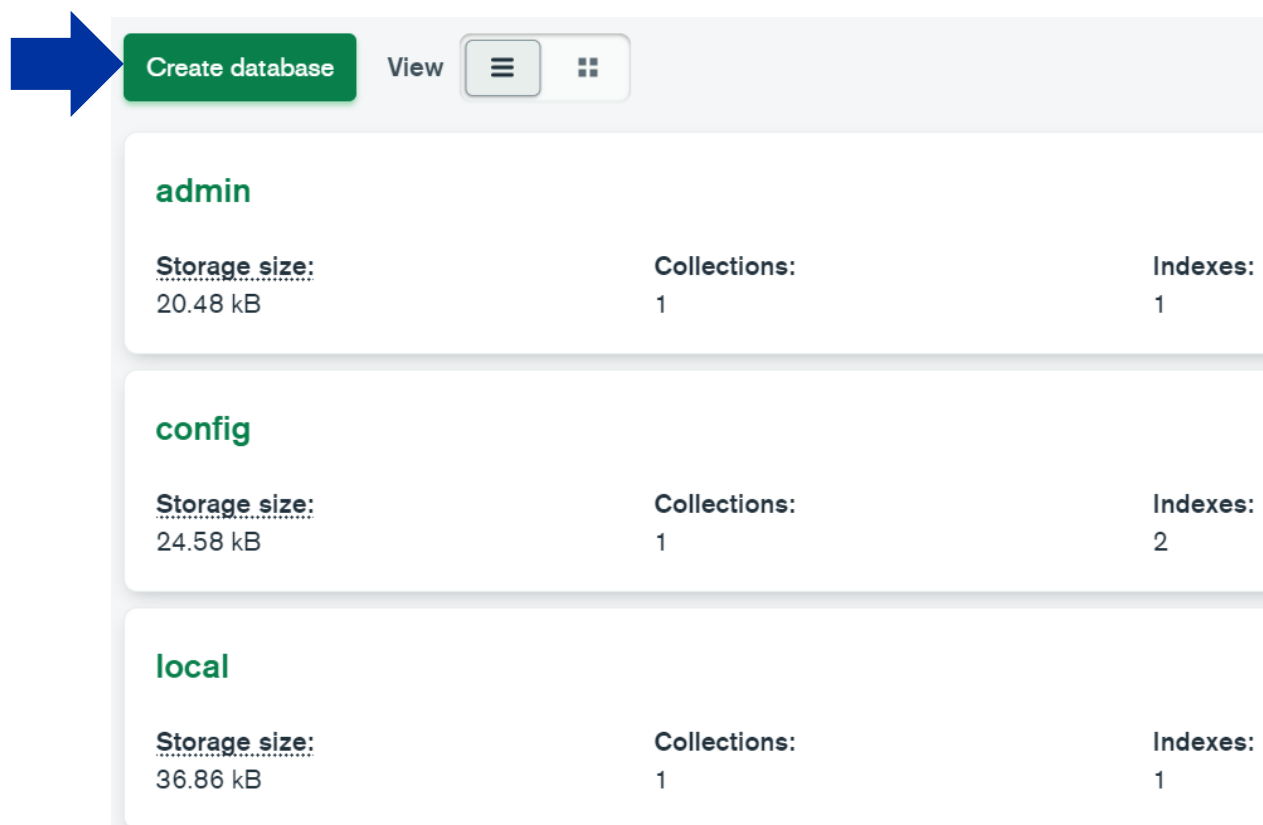
Servidor mongod

MongoDB compass

Conceptos básicos de MongoDB (interfaz grafica)

Creación de una base de datos en MongoDB compass

En principio tenemos tres bases de datos por default. Podemos crear una nueva seleccionando la opción “Create Database”.



Conceptos básicos de MongoDB (interfaz grafica)

Creación de una base de datos en MongoDB compass

Create Database

Database Name

Store

Collection Name

Clients

☐ Capped Collection

Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. [Learn More](#)

☐ Use Custom Collation

Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. [Learn More](#)

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

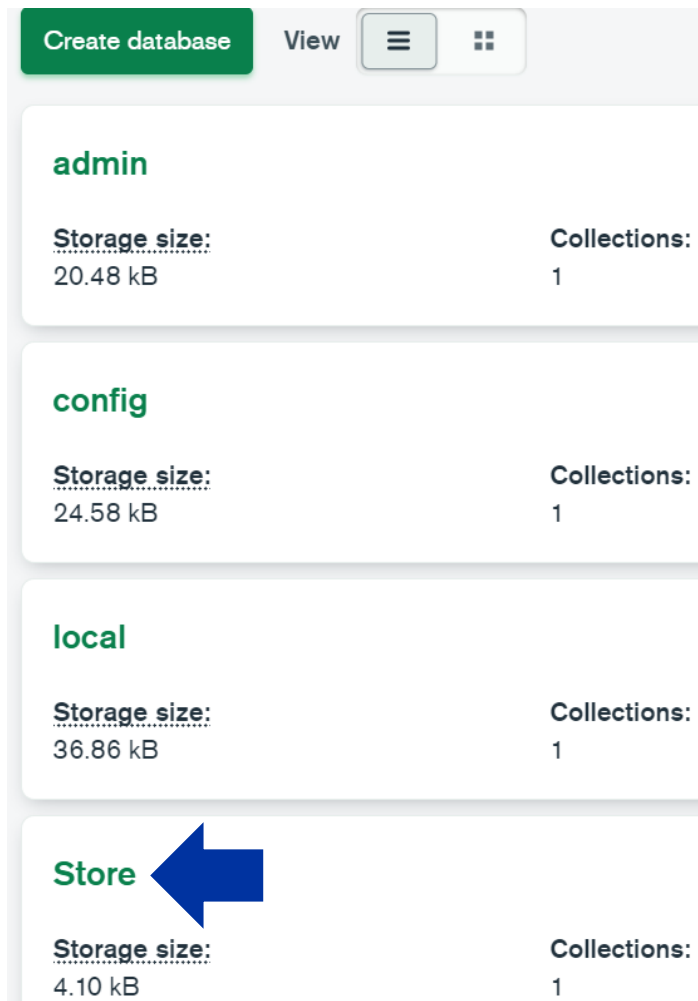
Cancel

Create Database

Creamos una nueva base de datos llamada “Store” con una primera colección/tabla llamada “Clients”. Paso siguiente, seleccionamos la opción “Create Database”.

Conceptos básicos de MongoDB (interfaz grafica)

Creación de una base de datos en MongoDB compass

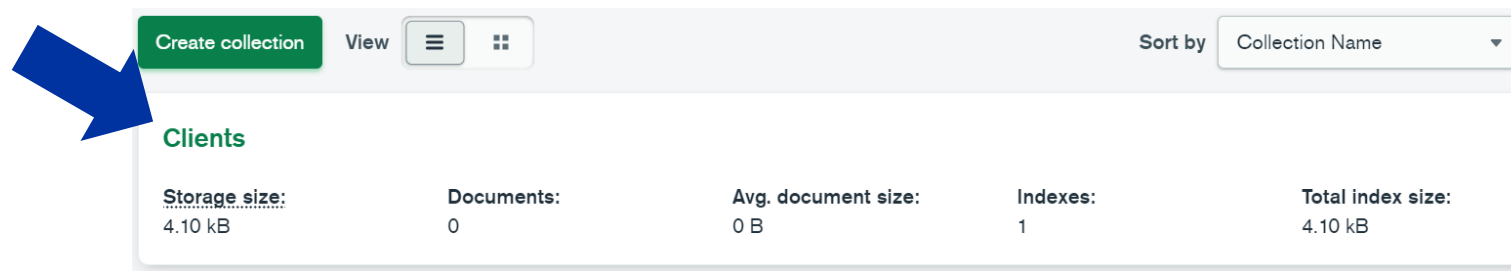


This screenshot shows the MongoDB Compass interface with a list of databases. The 'admin' database is selected. The interface includes a 'Create database' button and a 'View' toggle. The list shows the following details for each database:

Database Name	Storage size	Collections
admin	20.48 kB	1
config	24.58 kB	1
local	36.86 kB	1
Store	4.10 kB	1

Blue arrows point from the 'Store' database entry to the 'Clients' collection entry in the adjacent screenshot, and from the 'Collections' label to the '1' value.

Al crear la base de datos, esta aparecerá en la lista de elementos creados. Paso siguiente, la seleccionaremos para ver su contenido.



This screenshot shows the details for the 'Clients' collection. A blue arrow points from the 'admin' database in the previous screenshot to this collection. The interface includes a 'Create collection' button and a 'View' toggle. The collection details are as follows:

Collection Name	Storage size	Documents	Avg. document size	Indexes	Total index size
Clients	4.10 kB	0	0 B	1	4.10 kB

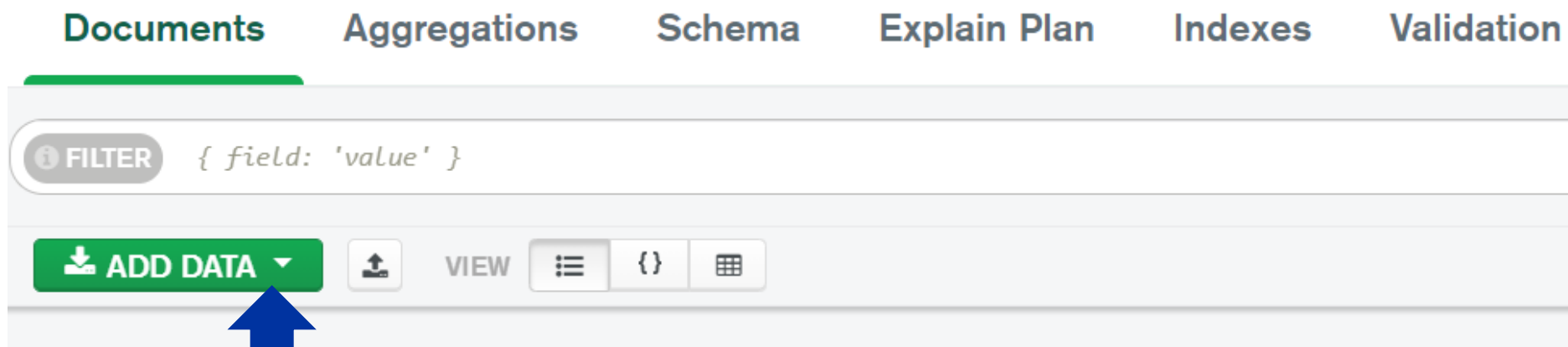
Blue arrows point from the 'Clients' collection entry to the 'Store' database entry in the adjacent screenshot, and from the 'Collections' label to the '1' value.

Conceptos básicos de MongoDB (interfaz grafica)

Agregar documentos en MongoDB compass

Lo siguiente seria explorar la colección/tabla “Clients”, por lo que la seleccionamos y después escogemos la opción “ADD DATA”, la cual nos permitirá agregar un documento (si, en MongoDB nos referimos a cada “fila” de una “tabla” como documento).

Store.Clients

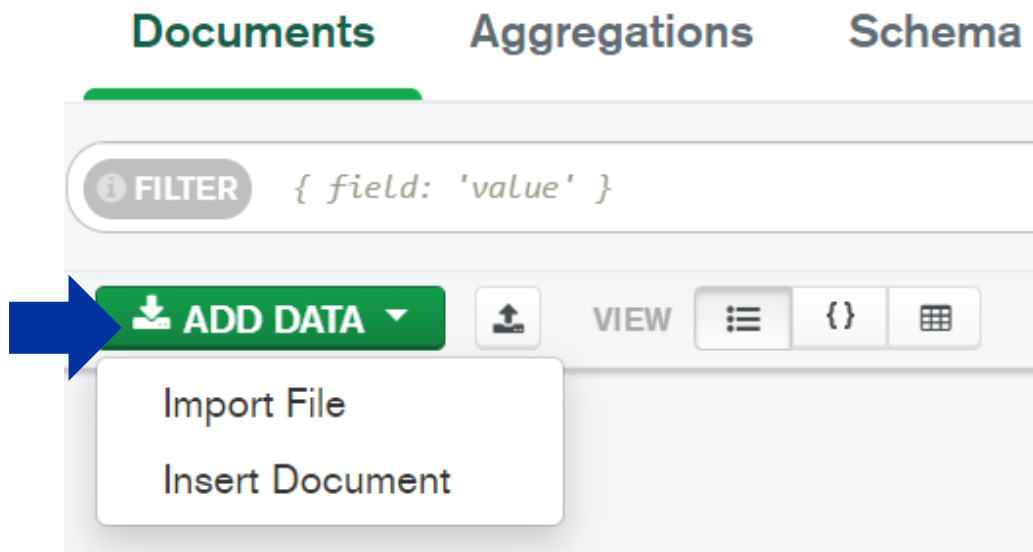


Conceptos básicos de MongoDB (interfaz grafica)

Agregar documentos en MongoDB compass

La opción “ADD DATA” nos permite añadir un documento de manera manual o importando un archivo, el cual normalmente es de extensión JSON. Este formato, como bien saben es el mas usado para la transferencia de información en la WEB.

Store.Clients




Conceptos básicos de MongoDB (interfaz grafica)

Agregar documentos en MongoDB compass

Insert to Collection Store.Clients

VIEW {} ≡

```
1 {  
2   "ClientName": "David",  
3   "ClientLastName": "Jones",  
4   "PhoneNumber": "555-432-567",  
5   "City": "NY",  
6   "Date": "09/12/2021"  
7 }
```



De momento, agregaremos información de manera manual, es decir, sin importar algún documento, por lo que seleccionamos la opción “Insert Document”.

Importante mencionar, que al insertar un documento, MongoDB agrega automáticamente un `_id` para diferenciarlo.

```
ClientName: "David"  
ClientLastName: "Jones"  
PhoneNumber: "555-432-567"  
City: "NY"  
Date: "09/12/2021"  
_id: ObjectId('627ff8ac9cc4ab79281a4052')
```

Conceptos básicos de MongoDB (interfaz grafica)

Agregar documentos en MongoDB compass

Insert to Collection Store.Clients

VIEW

```
1 [{
2   "ClientName":"Patricia",
3   "ClientLastName":"Robert",
4   "PhoneNumber":"927-887-991",
5   "City":"AZ",
6   "Date":"02/07/2019"
7 },
8
9 {
10  "ClientName":"John",
11  "ClientLastName":"Doe",
12  "PhoneNumber":"435-213-987",
13  "City":"CA",
14  "Date":"08/11/2020"
15 }]
```

También podemos agregar múltiples documentos usando la opción de arreglo en JSON (no olviden usar los corchetes para separar elementos al insertar).

```
_id: ObjectId('627ff8ac9cc4ab79281a4052')
ClientName: "David"
ClientLastName: "Jones"
PhoneNumber: "555-432-567"
City: "NY"
Date: "09/12/2021"
```

```
_id: ObjectId('627ffa7d9cc4ab79281a4054')
ClientName: "Patricia"
ClientLastName: "Robert"
PhoneNumber: "927-887-991"
City: "AZ"
Date: "02/07/2019"
```

```
_id: ObjectId('627ffa7d9cc4ab79281a4055')
ClientName: "John"
ClientLastName: "Doe"
PhoneNumber: "435-213-987"
```

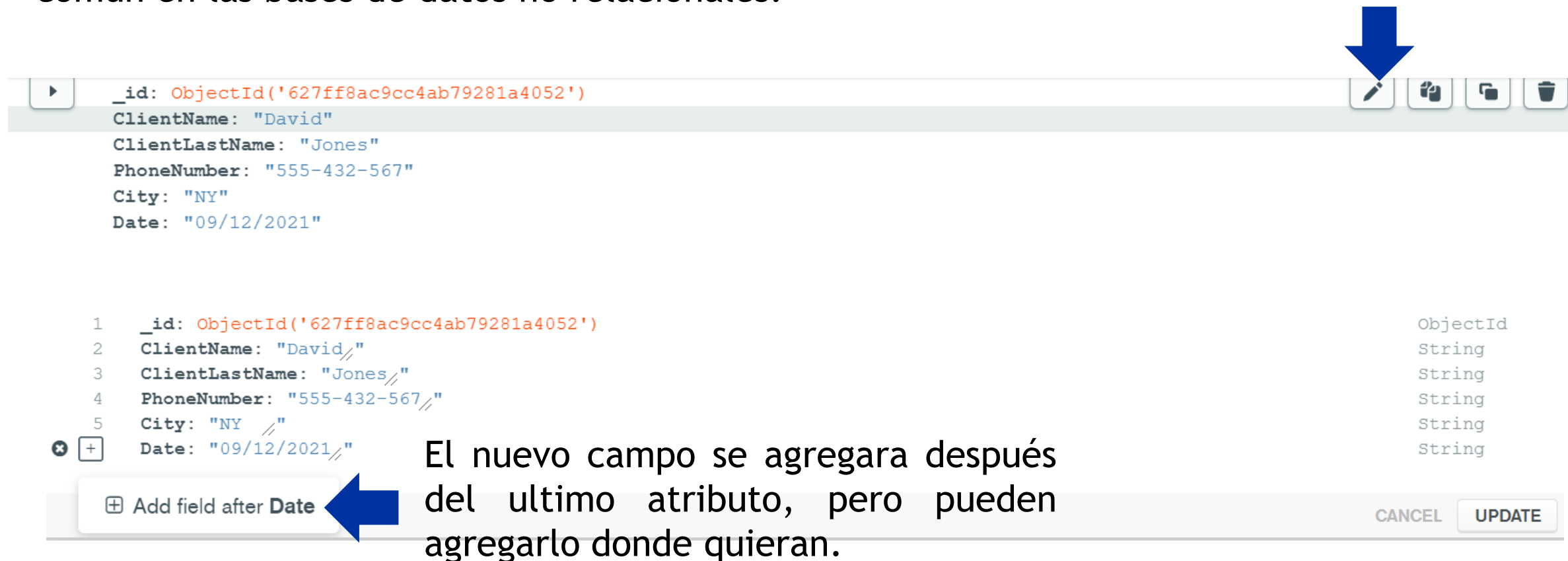
Cancel

Insert

Conceptos básicos de MongoDB (interfaz grafica)

Actualizar documentos en MongoDB compass

Podemos editar un documento insertado y agregar mas campos, lo cual es una característica común en las bases de datos no relacionales.



The screenshot illustrates the MongoDB Compass interface for editing a document. The document being edited is:

```
{
  "_id": ObjectId('627ff8ac9cc4ab79281a4052'),
  "ClientName": "David",
  "ClientLastName": "Jones",
  "PhoneNumber": "555-432-567",
  "City": "NY",
  "Date": "09/12/2021"
}
```

A blue arrow points to the edit icon (pencil) in the top right of the document view. Below the document view, a JSON representation of the document is shown with line numbers 1 through 5. At the bottom, a modal dialog "Add field after Date" is open, with a blue arrow pointing to the "Add field after Date" button. The modal also shows "CANCEL" and "UPDATE" buttons.

Conceptos básicos de MongoDB (interfaz grafica)

Actualizar documentos en MongoDB compass

Podemos editar un documento insertado y agregar mas campos, lo cual es una característica común en las bases de datos no relacionales.

```
1  _id: ObjectId('627ff8ac9cc4ab79281a4052')
2  ClientName: "David/"
3  ClientLastName: "Jones/"
4  PhoneNumber: "555-432-567/"
5  City: "NY /"
6  Date: "09/12/2021/"
+ Age: 35
```

Document Modified.

CANCEL

UPDATE

ObjectId
String
String
String
String
String
Int32

Puede agregar nuevos elementos, cambiar el tipo y después seleccionamos la opción “UPDATE”.

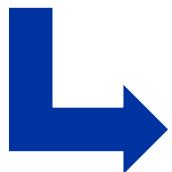
Conceptos básicos de MongoDB (interfaz grafica)

Actualizar documentos en MongoDB compass

Podemos editar un documento insertado y agregar mas campos, lo cual es una característica común de las bases de datos no relacionales.

1	<code>_id: ObjectId('627ff8ac9cc4ab79281a4052')</code>	ObjectId
2	<code>ClientName: "David"</code>	String
3	<code>ClientLastName: "Jones"</code>	String
4	<code>PhoneNumber: "555-432-567"</code>	String
5	<code>City: "NY"</code>	String
6	<code>Date: "09/12/2021"</code>	String
✕ +	<code>Age: 35</code>	Int32 ▼

Document Modified. CANCEL UPDATE



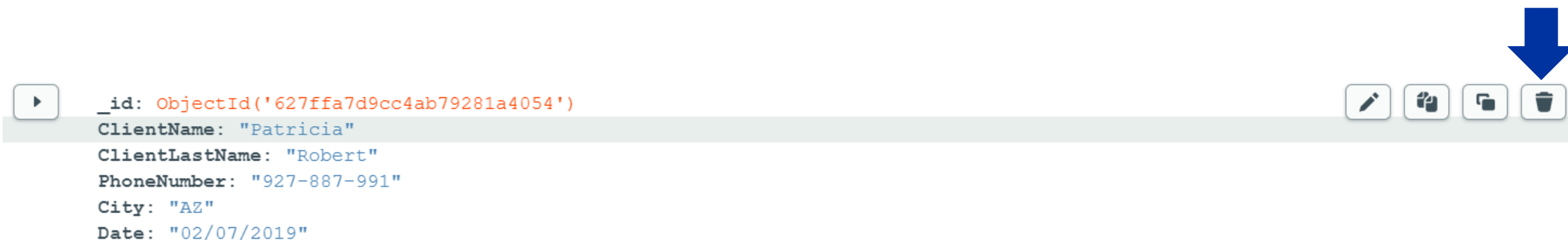
```
_id: ObjectId('627ff8ac9cc4ab79281a4052')
ClientName: "David"
ClientLastName: "Jones"
PhoneNumber: "555-432-567"
City: "NY"
Date: "09/12/2021"
Age: 35
```

Elemento modificado, con mas atributos que los demás en la colección.

Conceptos básicos de MongoDB (interfaz grafica)

Eliminar documentos en MongoDB compass

De igual manera a como podemos insertar y actualizar, podemos eliminar algún documento de la colección en MongoDB.



A screenshot of the MongoDB Compass interface. On the left, a document is displayed with the following fields: `_id: ObjectId('627ffa7d9cc4ab79281a4054')`, `ClientName: "Patricia"`, `ClientLastName: "Robert"`, `PhoneNumber: "927-887-991"`, `City: "AZ"`, and `Date: "02/07/2019"`. On the right, there is a toolbar with four icons: a pencil (edit), a document with a plus sign (insert), a document (refresh), and a trash can (delete). A large blue arrow points down to the trash can icon.

```
_id: ObjectId('627ffa7d9cc4ab79281a4054')
ClientName: "Patricia"
ClientLastName: "Robert"
PhoneNumber: "927-887-991"
City: "AZ"
Date: "02/07/2019"
```

Lo que resta hacer es seleccionar la opción
“DELETE”.

Document Flagged For Deletion.

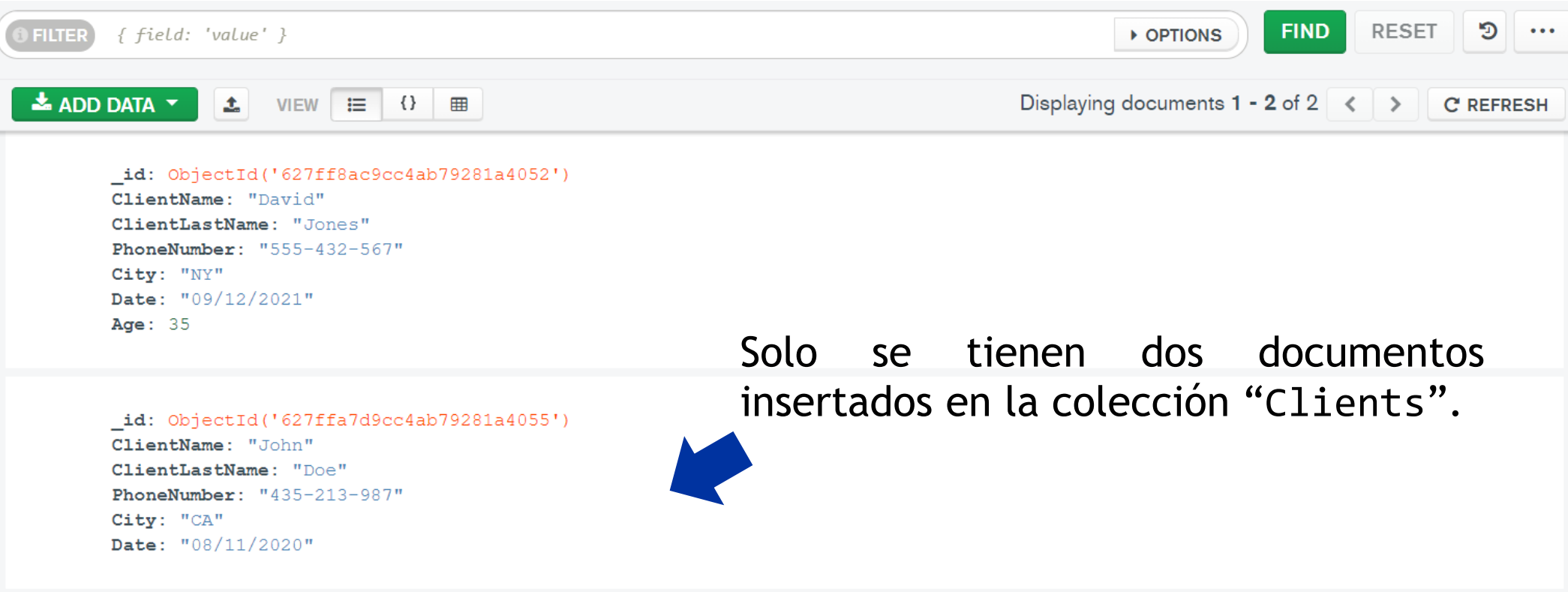
CANCEL

DELETE

Conceptos básicos de MongoDB (interfaz grafica)

Eliminar documentos en MongoDB compass

Después de eliminar un documento, podremos notar que la lista de elementos se actualiza con el estado actual de la colección.



The screenshot shows the MongoDB Compass interface. At the top, there is a filter bar with the text "{ field: 'value' }". To the right of the filter bar are buttons for "OPTIONS", "FIND", "RESET", and a refresh icon. Below the filter bar is a toolbar with a green "ADD DATA" button, a "VIEW" button, and icons for list, JSON, and grid views. To the right of the toolbar, it says "Displaying documents 1 - 2 of 2" with navigation arrows and a "REFRESH" button. The main area displays two documents in a list view. The first document is for "David Jones" and the second is for "John Doe". A blue arrow points to the second document.

```
_id: ObjectId('627ff8ac9cc4ab79281a4052')
ClientName: "David"
ClientLastName: "Jones"
PhoneNumber: "555-432-567"
City: "NY"
Date: "09/12/2021"
Age: 35
```

```
_id: ObjectId('627ffa7d9cc4ab79281a4055')
ClientName: "John"
ClientLastName: "Doe"
PhoneNumber: "435-213-987"
City: "CA"
Date: "08/11/2020"
```

Solo se tienen dos documentos insertados en la colección "Clients".

Conceptos básicos de MongoDB (interfaz grafica)

Buscar documentos en MongoDB compass

Para buscar un documento, debemos especificar un filtro en el formato {llave:“valor”}. Por ejemplo, busquemos a un cliente que sea del estado de Nueva York (NY). Pero notamos que agregamos el atributo “City” en vez de “State” por lo que procedemos a actualizar los documentos insertados.

```
_id: ObjectId('627ff8ac9cc4ab79281a4052')
ClientName: "David"
ClientLastName: "Jones"
PhoneNumber: "555-432-567"
Date: "09/12/2021"
Age: 35
State: "NY"
```

```
_id: ObjectId('627ffa7d9cc4ab79281a4055')
ClientName: "John"
ClientLastName: "Doe"
PhoneNumber: "435-213-987"
Date: "08/11/2020"
State: "CA"
```

Conceptos básicos de MongoDB (interfaz grafica)

Buscar documentos en MongoDB compass

Después procedemos a realizar la búsqueda antes mencionada usando el patrón `{llave:"valor"}`.

FILTER `{State:"NY"}` **OPTIONS** **FIND**

Elemento a buscar



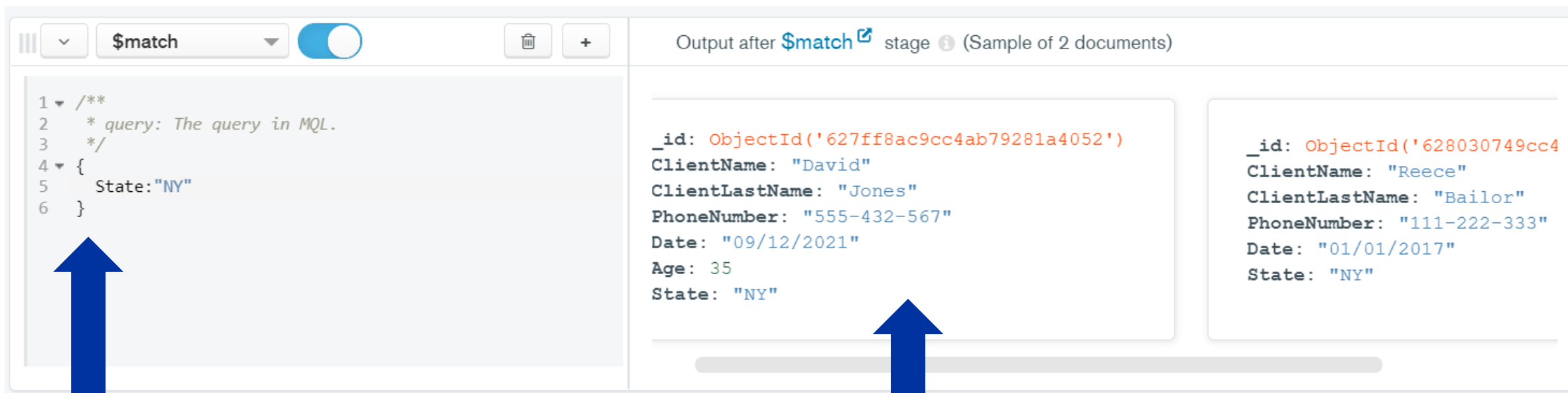
```
_id: ObjectId('627ff8ac9cc4ab79281a4052')
ClientName: "David"
ClientLastName: "Jones"
PhoneNumber: "555-432-567"
Date: "09/12/2021"
Age: 35
State: "NY"
```

Este seria el único documento recuperado.

Conceptos básicos de MongoDB (interfaz grafica)

Buscar documentos en MongoDB compass

Otra forma de buscar documentos en MongoDB es a través de agregaciones (consultas avanzadas). En este caso, usaremos una agregación de tipo “MATCH”, donde nuevamente buscaremos a las personas que sean del estado de NY: {State: "NY"}



The screenshot shows the MongoDB Compass interface. On the left, the query editor displays a \$match query: `{ State: "NY" }`. A blue arrow points from the label 'Consulta' to this query. On the right, the 'Output after \$match stage' panel shows two sample documents. A blue arrow points from the label 'Resultado' to this panel.

Output after `$match` stage (Sample of 2 documents)

```
{
  "_id": ObjectId('627ff8ac9cc4ab79281a4052'),
  "ClientName": "David",
  "ClientLastName": "Jones",
  "PhoneNumber": "555-432-567",
  "Date": "09/12/2021",
  "Age": 35,
  "State": "NY"
}
```

```
{
  "_id": ObjectId('628030749cc4'),
  "ClientName": "Reece",
  "ClientLastName": "Bailor",
  "PhoneNumber": "111-222-333",
  "Date": "01/01/2017",
  "State": "NY"
}
```

Consulta

Resultado

Conceptos básicos de MongoDB (interfaz grafica)

Crear índices en documentos usando MongoDB compass

- Si queremos agilizar el tiempo de ejecución de una consulta sobre un documento en MongoDB, podemos crear un índice.
- En específico, los índices de MongoDB son similares a los de otras bases de datos no relacionales como DynamoDB, ya que en ambos, un índice se aplica sobre atributos en los cuales se hacen muchas consultas.
- Podemos crear un índice sobre un solo atributo o sobre múltiples, aunque entre mas atributos, será mas costoso en términos de consumo de memoria RAM.

Conceptos básicos de MongoDB (interfaz grafica)

Crear índices en documentos usando MongoDB compass

Para crear un índice en MongoDB, debemos dirigirnos a la sección “Indexes”, en la cual, podemos implementar los índices que deseemos (mientras tengamos RAM Y ROM suficiente).



Conceptos básicos de MongoDB (interfaz grafica)

Crear índices en documentos usando MongoDB compass

Dentro de la sección de índices, seleccionamos la opción “CREATE INDEX”. Importante notar que cada documento tiene un índice por default de tipo único (unique), el cual funciona como una llave primaria en una base de datos relacional.

Store.Clients

4 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan **Indexes** Validation

CREATE INDEX

Name and Definition ^	Type	Size	Usage	Properties
<code>_id_</code> _id ↑	REGULAR ⓘ	4.1 KB	1 since Sat May 14 2022	UNIQUE ⓘ

Conceptos básicos de MongoDB (interfaz grafica)

Crear índices en documentos usando MongoDB compass

Dentro de la sección de índices, seleccionamos la opción “CREATE INDEX”. Importante notar que cada documento tiene un índice por default de tipo único (unique), el cual funciona como una llave primaria en una base de datos relacional.

Column	Description
Name and Definition	The name of the index and keys.
Type	Regular, text, geospatial, or hashed index.
Size	How large the index is.
Usages	The number of times the index has been used
Properties	Any special properties of the index.

Descripción de las propiedades asociadas a un índice.



Conceptos básicos de MongoDB (interfaz grafica)

Crear índices en documentos usando MongoDB compass

Create Index

Choose an index name

StateIndex

Configure the index definition

State

ADD ANOTHER

text

1 (asc)

-1 (desc)

2dsphere

text

Options

- ☐ Build index in the background ⓘ
- ☐ Create unique index ⓘ
- ☐ Create TTL ⓘ
- ☐ Partial Filter Expression ⓘ
- ☐ Use Custom Collation ⓘ
- ☐ Wildcard Projection ⓘ

La opción “Type”, nos permite decidir si tenemos un índice de tipo alfanumérico (text) o numérico (ascendente-asc; descendente).

Conceptos básicos de MongoDB (interfaz grafica)

Crear índices en documentos usando MongoDB compass

Create Index

Choose an index name

StateIndex

Configure the index definition

State

text

ADD ANOTHER FIELD

Options

- ☒ Build index in the background ⓘ
- ☐ Create unique index ⓘ
- ☐ Create TTL ⓘ
- ☐ Partial Filter Expression ⓘ
- ☐ Use Custom Collation ⓘ
- ☐ Wildcard Projection ⓘ

CANCEL

CREATE INDEX

Documents

Aggregations

Schema

Explain Plan

Indexes

CREATE INDEX

Name and Definition ^

Type

Size

StateIndex

_fts

text

_ftsx

↑

TEXT ⓘ

20.5 KB

id

_id

↑

REGULAR ⓘ

36.9 KB

Referencias

- Sommerville, I., Software Engineering, 10th Edition, Pearson, 2016, IN, 1292096144, 9781292096148.
- Connolly Thomas M, Database systems : a practical approach to design, implementation and management, 5thed., London : Addison-Wesley, 2010, 9780321523068.
- Perez, C., MySQL para windows y Linux, España, Alfaomega, 2004.
- <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>

Gracias!
Preguntas...



Dr. Esteban Castillo Juarez

Google academics:

<https://scholar.google.com/citations?user=JfZpVO8AAAAJ&hl=en>

<https://dblp.uni-trier.de/pers/hd/c/Castillo:Esteban>