



MongoDB

CORA PEREZ MANIVESA

NoSQL

MongoDB

Tabla de contenido

INTRODUCCIÓN	3
HISTORIA DEL TERMINO	4
Importancia y uso de las bases de datos NoSQL	5
Qué son las bases de datos NoSQL?	5
COMO FUNCIONAN?	5
TIPOS DE BASES DE DATOS NO SQL	5
PORQUE USAR UNA BASE DE DATOS NoSQL?	8
TIPOS DE ESCALABILIDAD EN LAS BASES DE DATOS	9
VENTAJAS Y DESVENTAJAS	11
CASOS DE USO	12
IV. Comparación entre bases de datos NoSQL y bases de datos relacionales.	14
CONSULTAS	20
ESTUDIO COMPARATIVO STATE –OF- THE- ART	23
MONGODB	24
ARQUITECTURA	26
VENTAJAS Y DESVENTAJAS	27
CASOS DE USO	28
VISION UNIFICADA	28
REQUISITOS DEL SISTEMA	29
DESCARGA E INSTALCIÓN DE MONGODB	31
EXPORTAR INFORMACIÓN DESDE MONGODB	42

NoSQL

MongoDB

```
xubu@xubu:~$ sudo mongoimport --db newdb --collection ciudades --drop --file /home/xubu/Descargas/zips.json
2023-03-21T13:37:07.013+0100    connected to: mongodb://localhost/
2023-03-21T13:37:07.014+0100    dropping: newdb.ciudades
2023-03-21T13:37:07.038+0100    15 document(s) imported successfully. 0 document(s) failed to import.
```

..... 43

```
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
newdb    0.012GB
> use newdb
switched to db newdb
> show collections
ciudades
restaurants
> show ciudades
uncaught exception: Error: don't know how to show [ciudades] :
shellHelper.show@src/mongo/shell/utils.js:1191:11
shellHelper@src/mongo/shell/utils.js:819:15
@(shellhelp2):1:1
> show ciudades
uncaught exception: Error: don't know how to show [ciudades] :
shellHelper.show@src/mongo/shell/utils.js:1191:11
shellHelper@src/mongo/shell/utils.js:819:15
@(shellhelp2):1:1
> db.colecciones.find()
> db.ciudades.find()
{ "_id" : "01001", "city" : "AGAWAM", "loc" : [ -72.622739, 42.070206 ], "pop" : 15338, "state" : "MA" }
{ "_id" : "01002", "city" : "CUSHMAN", "loc" : [ -72.51565, 42.377017 ], "pop" : 36963, "state" : "MA" }
{ "_id" : "01005", "city" : "BARRE", "loc" : [ -72.108354, 42.409698 ], "pop" : 4546, "state" : "MA" }
{ "_id" : "01007", "city" : "BELCHERTOWN", "loc" : [ -72.410953, 42.275103 ], "pop" : 10579, "state" : "MA" }
{ "_id" : "01008", "city" : "BLANDFORD", "loc" : [ -72.936114, 42.182949 ], "pop" : 1240, "state" : "MA" }
{ "_id" : "01010", "city" : "BRIMFIELD", "loc" : [ -72.188455, 42.116543 ], "pop" : 3706, "state" : "MA" }
{ "_id" : "01011", "city" : "CHESTER", "loc" : [ -72.988761, 42.279421 ], "pop" : 1688, "state" : "MA" }
{ "_id" : "01012", "city" : "CHESTERFIELD", "loc" : [ -72.833309, 42.38167 ], "pop" : 177, "state" : "MA" }
{ "_id" : "01013", "city" : "CHICOPEE", "loc" : [ -72.607962, 42.162046 ], "pop" : 23396, "state" : "MA" }
{ "_id" : "01020", "city" : "CHICOPEE", "loc" : [ -72.576142, 42.176443 ], "pop" : 31495, "state" : "MA" }
{ "_id" : "01022", "city" : "WESTOVER AFB", "loc" : [ -72.558657, 42.196672 ], "pop" : 1764, "state" : "MA" }
{ "_id" : "01026", "city" : "CUMMINGTON", "loc" : [ -72.905767, 42.435296 ], "pop" : 1484, "state" : "MA" }
{ "_id" : "01027", "city" : "MOUNT TOM", "loc" : [ -72.679921, 42.264319 ], "pop" : 16864, "state" : "MA" }
{ "_id" : "01028", "city" : "EAST LONGMEADOW", "loc" : [ -72.505565, 42.067203 ], "pop" : 13367, "state" : "MA" }
{ "_id" : "01030", "city" : "FEEDING HILLS", "loc" : [ -72.675077, 42.07182 ], "pop" : 11985, "state" : "MA" }
>
```

..... 43

CONSULTAS DE LA BASE DE DATOS USANDO EL SHELL DE MONGODB..... 44

CONCLUSIONES 47

RESUMEN VENTAJAS Y DESVENTAJAS MONGODB 47

FUTURO DE LAS BASES DE DATOS NoSQL y MongoDB..... 49

REFERENCIAS 51

NoSQL

MongoDB

INTRODUCCIÓN

Particularmente a parte de las clases bases de datos SQL(RDBMS), aparecen y van tomando fuerza nuevos tipos de bases de datos. Podemos destacar que su mayor ventaja es que son mucho más rápidas. Dependiendo de su tipo, cada una sigue una estrategia diferente para persistir la información. Debemos destacar que no suelen sustituir a las bases de datos clásicas SQL, aparecen por otras necesidades. Una necesidad de rendimiento extremo. Dependiendo de la arquitectura del sistema podemos decidir si usamos solo un tipo o los dos. Algunas de ellas pueden ser accedidas mediante SQL, pero normalmente no será así, puesto que cada una tendrá una API exclusiva.

NoSQL

MongoDB

HISTORIA DEL TERMINO

Carlo Strozzi usó el término NoSQL en 1998 para referirse a su base de datos. Era una base de datos open-source, ligera que no ofrecía un interface SQL, pero si seguía el modelo relacional. El actual movimiento NoSql "Se sale completamente del modelo relacional, debería, por tanto, llamado "NoRel".

Eric Evans, un empleado de Rackspace, reintrodujo el término NoSQL cuando Johan Oskarsson de Last.fm quiso organizar un evento para discutir bases de datos distribuidas de código abierto. El nombre intentaba recoger el número creciente de bases de datos no relacionales y distribuidas que no garantizaban ACID, atributo clave en las SGBDR clásicas.

NoSQL

MongoDB

Importancia y uso de las bases de datos NoSQL

Qué son las bases de datos NoSQL?

Podemos definir las bases de datos NoSQL son una categoría de bases de datos que se han desarrollado para manejar grandes volúmenes de datos no estructurados. A diferencia de las bases de datos relacionales, que utilizan tablas para almacenar datos, las bases de datos NoSQL utilizan otros modelos de datos, como documentos, grafos o pares clave-valor.

Una de las principales ventajas de las bases de datos NoSQL es su capacidad para escalar horizontalmente, lo que significa que pueden manejar grandes cantidades de datos sin sacrificar rendimiento. Además las bases de datos NoSQL son más flexibles que las bases de datos relacionales, ya que permiten agregar y modificar datos sin tener que cambiar el esquema de la base de datos.

COMO FUNCIONAN?

Aunque varían según su tipo, podemos decir que el razonamiento general es NO vamos a guardar la información en tablas.

TIPOS DE BASES DE DATOS NO SQL

La evolución de las tecnologías NoSQL para responder a distintos problemas, entre ellas tienen muchos aspectos en común, siembargo son muy diferentes entre si. Debido a la diversidad de tecnologías NoSQL habitualmente se clasifican en 4 grupos por como ordenan los datos:

NoSQL

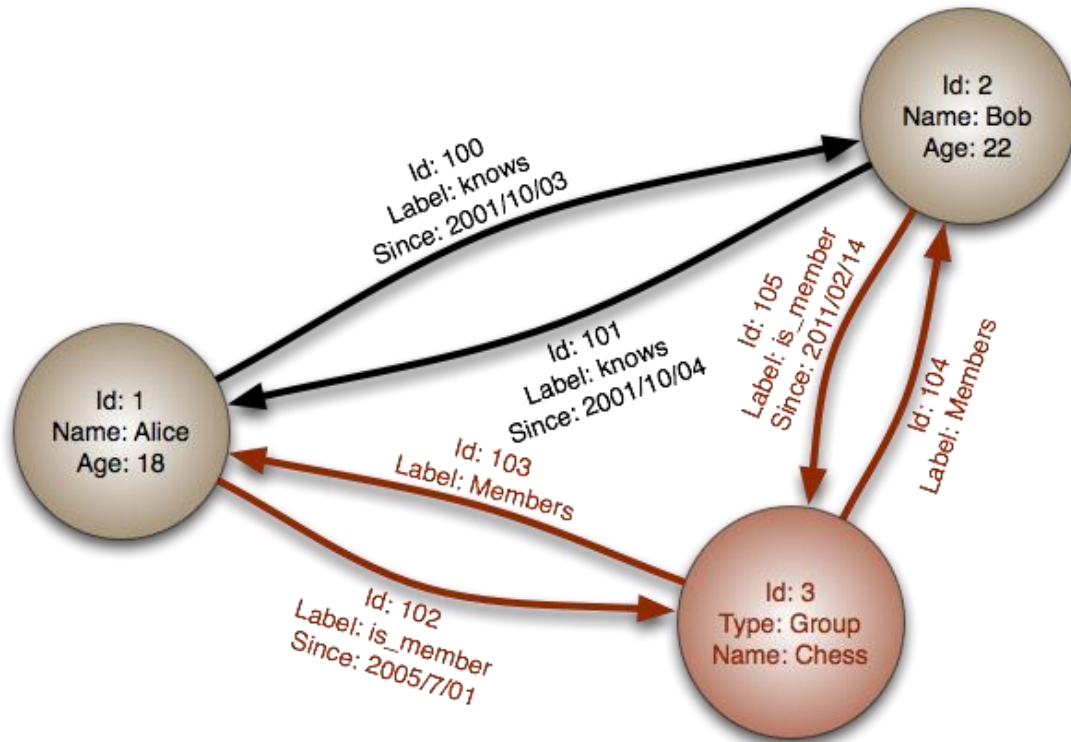
MongoDB

Bases de datos de documentos o documentales: estas bases de datos almacenan datos en documentos, qué pueden ser archivos JSON,XML o BSON.Ejemplos populares de bases de datos de documentos incluyen MongoDB y CouchBase .Las bases de datos de documentos son altamente escalables y flexibles, y son particularmente útiles para aplicaciones que manejan datos semiestructurados o no estructurados.



Bases de datos de Grafos: estas bases de datos almacenan datos en forma de nodos y relaciones, lo que las hace ideales para aplicaciones que necesitan modelar relaciones complejas entre datos. Ejemplos mas populares de bases de datos de grafos incluyen Neo4j y ArangoDB .Las bases de datos de grafos son altamente eficientes para consultas de datos complejas, pero pueden ser más difíciles de escalar que las otras bases de datos NoSQL.

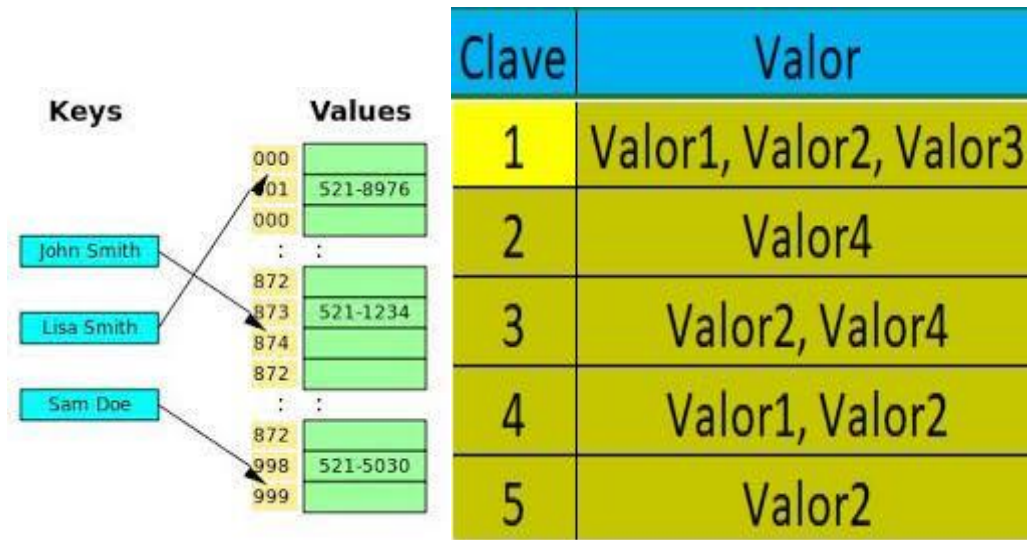
NoSQL MongoDB



Bases de datos Clave-Valor: estas bases de datos almacenan datos de forma de pares clave-valor incluyen Redis y Amazon DynamoDB. Las bases de datos clave-valor son altamente escalables y eficientes para operaciones de lectura y escritura, pero pueden ser menos adecuadas para aplicaciones que necesitan realizar consultas complejas.

NoSQL

MongoDB



En conclusión las bases de datos NoSQL ofrecen una alternativa a las bases de datos relacionales para manejar grandes volúmenes de datos no estructurados. La elección de la base de datos NoSQL adecuada dependerá de las necesidades específicas de cada aplicación.

PORQUE USAR UNA BASE DE DATOS NoSQL?

Las bases de datos NoSQL se adaptan perfectamente a muchas aplicaciones modernas, como dispositivos móviles, web y juegos, que requieren bases de datos flexibles, escalables de alto rendimiento y altamente funcionales para proporcionar excelentes experiencias de usuario.

Flexibilidad: las bases de datos NoSQL generalmente ofrecen esquemas flexibles que permiten un desarrollo más rápido y más interactivo. El modelo de datos flexible hace que las bases de datos NoSQL sean ideales para datos semiestructurados y no estructurados.

Escalabilidad: las bases de datos NoSQL generalmente están diseñadas para escalar usando clústers distribuidos de hardware en lugar de escalar añadiendo servidores caros y sólidos. Algunos proveedores de la nube manejan estas operaciones en segundo plano, como un servicio completamente administrado.

NoSQL

MongoDB

Alto rendimiento: la base de datos NoSQL esta optimizada para modelos de datos específicos y patrones de acceso que permiten un mayor rendimiento que el intento de lograr una funcionalidad similar con bases de datos relacionales.

Altamente funcional: las bases de datos NoSQL proporcionan API altamente funcionales y tipos de datos que están diseñados específicamente para cada uno de sus respectivos modelos de datos.

TIPOS DE ESCALABILIDAD EN LAS BASES DE DATOS

Existen distintas maneras de hacer que un software sea escalable. Pueden combinarse estructuras técnicas de software y hardware e incluso arquitecturas alternas de red. Pero existe una clasificación base de la actual debemos partir para analizar el potencial escalable de una base de datos y de cualquier aplicación.

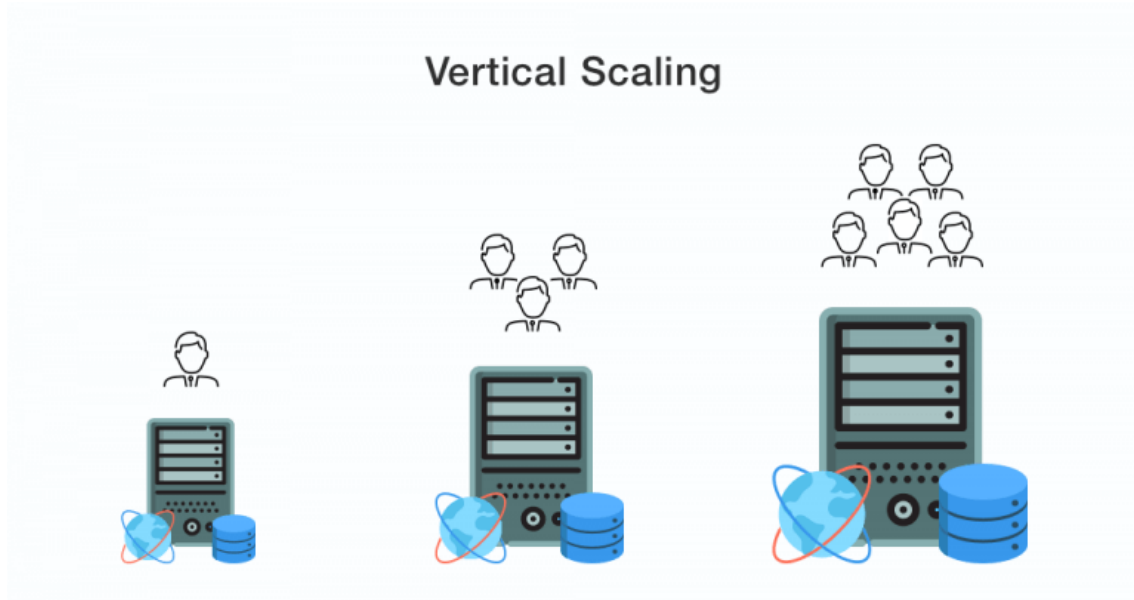
Estas clasificaciones son las que conocemos como escalabilidad horizontal, vertical y híbrida. Cada cuenta con un conjunto de características, ventajas y desventajas que debemos tomar en cuenta para el diseño correcto del proyecto que queramos realizar. Veamos sus diferencias y ventajas.

TIPO VERTICAL

La escalabilidad también conocida como escalabilidad hacia arriba es un modelo de crecimiento. Significa básicamente que el crecimiento o expansión de la cantidad de nodos de almacenamiento depende de la estructura tecnológica física. Su crecimiento depende de la ampliación o cambio del hardware por una versión adicional o más potente.

NoSQL

MongoDB



El esfuerzo de este crecimiento es relativamente bajo. No tiene mayor incidencia en el rendimiento ya que con el tiempo, el almacenamiento y otras capacidades vuelven a demandar ampliación y renovación.

Es importante entender que este escalamiento no es malo. Es necesario que sea realizado cada cierto tiempo para ir pasos delante de los límites lógicos que impone el hardware. También sirve para ir incorporando tecnología nueva a nuestros procesos.

ESCALABILIDAD HORIZONTAL

El escalado horizontal es un método muy poderoso y representa un desafío mayor. Este modelo implica tener múltiples servidores conocidos como nodos que actúan como un todo. Se organizarán en forma de redes de servidores denominadas *clústeres*. Su propósito es dividir eficientemente las demandas de trabajo entre todos los nodos que componen una red de servidores,

Como el rendimiento del clúster se ve afectado por el aumento de usuarios u operaciones, los nodos escalan para equilibrar la demanda de trabajo en cada nodo. Para lograr un funcionamiento óptimo de esta metodología, debe existir un servidor principal en el que se ejecuta la administración del clúster.

NoSQL

MongoDB

VENTAJAS Y DESVENTAJAS

El escalonamiento tiene una serie de beneficios a considerar. No es traumático para las aplicaciones ya que el cambio se produce en el hardware. Es fácil de implementar con un plan de migración ordenado. Se puede hacer rápidamente en el tiempo.

También tiene una serie de desventajas importantes que deberían darnos que pensar. Pero lo más importantes que debería hacernos evaluar completamente el modelo de escalabilidad que queremos para nuestro proyecto y no solo depender de un método, El hardware limita el crecimiento de esta técnica de escalabilidad. La falla de un servidor significa una interrupción completa de las aplicaciones y ciertamente puede significar altos costos de inversión en equipos.

La escalabilidad horizontal es una de las características más deseadas de una base de datos, Este es uno de los atributos clave de las bases de datos NoSQL. Esto se debe a que el crecimiento es tan extenso y técnicamente ilimitado. Permite agregar tantos servidores como desee.

Además, este modelo de escalabilidad es totalmente compatible con el escalado vertical. Y si falla un nodo, otros nodos en el clúster continúan ejecutándose, por lo que su software y aplicaciones no dejan de ejecutarse.

Pero este método no es perfecto. La escalabilidad horizontal requiere de un mantenimiento exhaustivo. Configurarlos tiende a ser complicado. Es posible que se requiera una infraestructura técnica más grande. Además si su aplicación no se diseñó originalmente para funcionar en un modelo, tendrá que estructurar significativamente su código para adaptarse.

CASOS DE USO

La implementación de modelos escalables es muy eficiente para grandes proyectos de procesamiento y búsqueda de datos, Su distribución horizontal permite una distribución eficiente de los datos para el acceso a consultas simultáneas.

La mayoría de las bases de datos NoSQL tienen esta importante propiedad.

TIPO HIBRIDA

Scale Diagonally buscar ayuda a combinar elementos de escalabilidad vertical y horizontal. Este método facilita la reducción de los recursos de almacenamiento cuando disminuye la demanda de Cloud - Hybrid

Este modelo ofrece flexibilidad para las cargas de trabajo aumentando su disponibilidad para cualquier momento. Este modelo introduce interesantes variables a nivel de coste rentabilidad de la estructura tecnológica. La inversión de la empresa en esta área está directamente relacionada con el tiempo de uso ,es decir, con la necesidad real de implementación, lo que en la mayoría de los casos representa un ahorro importante.

VENTAJAS

Dado que es una arquitectura semidescentralizada flexible , tiene una posibilidad de ampliarse para atender un tráfico de carga de datos de gran volumen.

Eso significa que es una opción bastante eficiente para proyectos que comienzan con un presupuesto modesto.

CASOS DE USO

MongoDB es una base de datos NoSQL que ha ganado popularidad en los últimos años debido a su capacidad de almacenar grandes cantidades de datos no estructurados o semi-estructurados .A continuación, se describen algunos de los casos de uso más comunes de MongoDB.

1.Aplicaciones web: es un excelente opción para las aplicaciones web que manejan grandes cantidades de datos. Su capacidad para manejar datos no estructurados, su escalabilidad y su capacidad para manejar datos no estructurados, su escalabilidad y su capacidad para procesar grandes cantidades de datos a tiempo real hacen que MongoDB una opción atractiva para las aplicaciones web modernas.

2.IoT (Internet de las cosas): La creciente cantidad de dispositivos conectados a la red ha generado una gran cantidad de datos que deben ser almacenados y

NoSQL

MongoDB

procesados. MongoDB es una opción para las aplicaciones IoT a la capacidad para manejar grandes cantidades de datos a tiempo real.

3.Big Data: El creciente volumen de datos que se generan en la actualidad requiere de soluciones de almacenamiento de datos que sean escalables y flexibles. MongoDB es una buena opción para almacenar y procesar grandes cantidades de datos a tiempo real.

4.Análisis de datos: MongoDB es una excelente opción para el análisis de datos debido a su capacidad para manejar datos no estructurados y semi-estructurados. También es compatible con herramientas de análisis de datos populares como Hadoop y Spark.

5.Publicidad en línea: La publicidad en línea requiere procesamiento de grandes cantidades de datos en tiempo real. MongoDB es una opción popular de publicidad en línea debido a su capacidad para manejar grandes volúmenes de datos en tiempo real y su escalabilidad.

6.Juegos: Los juegos en línea requieren de procesamiento de grandes cantidades de datos a tiempo real. MongoDB es una buena opción para juegos en línea debido a su capacidad para manejar grandes volúmenes de datos en tiempo real y su escalabilidad.

En conclusión , MongoDB es una base de datos NoSQL popular debido a su capacidad para manejar grandes volúmenes de datos a tiempo real, su escalabilidad y su capacidad para procesar datos no estructurados y semi-estructurados. Es una excelente opción para una variedad de casos de uso, incluyendo aplicaciones web , IoT, big data, análisis de datos , publicidad en línea y juegos en línea.

IV. Comparación entre bases de datos NoSQL y bases de datos relacionales

A continuación, se presenta una tabla comparativa de las ventajas y desventajas de las bases de datos NoSQL y las bases de datos relacionales:

NoSQL MongoDB

Ventajas		Desventajas
NoSQL	- Escalabilidad y disponibilidad	- Falta de estandarización en diseño y lenguajes de consulta
	- Flexibilidad en la estructura de los datos	- Posible redundancia de datos
	- Capacidad para manejar diferentes tipos de datos	
	- Velocidad en el manejo de grandes cantidades de datos	
Relacionales	- Estándares en el diseño y lenguajes de consulta	- Rigidez en la estructura de los datos
	- Evita redundancia de datos y asegura	- Dificultad para escalar y adaptar a cambios en la estructura de los datos

NoSQL

MongoDB

	la integridad de los mismos	
--	-----------------------------	--

ESCALABILIDAD

Hoy en día, la escalabilidad es un factor clave de éxito para cualquier aplicación o sistema. Este ensayo discutirá las ventajas y desventajas de las bases de datos MySQL y NoSQL en términos de escalabilidad.

MySQL es una base de datos relacional que ha sido ampliamente utilizada en aplicaciones y sistemas durante muchos años. En términos de escalabilidad, MySQL puede manejar grandes cantidades de datos y admitir múltiples conexiones simultáneas. Además, MySQL es compatible con muchos sistemas operativos y plataformas de hardware, lo que facilita su implementación en una variedad de entornos. Otra ventaja de MySQL es que es compatible con muchas herramientas de análisis y visualización de datos, lo que facilita el procesamiento de grandes cantidades de datos.

Sin embargo, una de las principales desventajas de MySQL en términos de escalabilidad es que puede tener problemas de rendimiento al procesar grandes cantidades de datos o muchas conexiones simultáneas. Además, MySQL es más difícil de escalar, lo que significa que tendrá que agregar más recursos a un servidor existente en lugar de agregar nuevos servidores. Esto puede limitar su capacidad para procesar grandes cantidades de datos en el futuro.

Por otro lado, las bases de datos NoSQL como MongoDB y Cassandra se han convertido en una alternativa a MySQL en términos de escalabilidad. Las bases de datos NoSQL se diseñaron específicamente para manejar grandes cantidades de datos y brindar alta disponibilidad. Estas bases de datos escalan bien horizontalmente, lo que significa que puede agregar nuevos servidores para manejar grandes cantidades de datos sin comprometer el rendimiento.

Otra ventaja de las bases de datos NoSQL es que pueden manejar varios tipos de datos, como datos no estructurados y big data. Además, las bases de datos NoSQL son muy flexibles en términos de estructura de datos, lo que significa que pueden adaptarse fácilmente a los cambios en la estructura de datos.

Sin embargo, una desventaja de las bases de datos NoSQL es que pueden ser más difíciles de administrar y configurar en comparación con MySQL. Además, sin usar un

NoSQL

MongoDB

modelo tabular, puede ser más difícil ejecutar consultas complejas y, en algunos casos, puede ser necesario un lenguaje de consulta personalizado.

En conclusión, tanto las bases de datos MySQL como las bases de datos NoSQL tienen sus ventajas y desventajas cuando se trata de escalabilidad. MySQL tiene la ventaja de ser compatible con muchos sistemas operativos y herramientas de análisis de datos, pero puede tener problemas de rendimiento y es más difícil de escalar. Por otro lado, las bases de datos NoSQL escalan bien horizontalmente y pueden manejar diferentes tipos de datos, pero pueden ser más complejas de administrar y ajustar. En última instancia, la elección entre MySQL y una base de datos NoSQL dependerá de las necesidades específicas de la aplicación o sistema en cuestión.

FLEXIBILIDAD DE ESQUEMA

La flexibilidad del esquema es uno de los factores más importantes a considerar al elegir una aplicación o sistema de base de datos. Este artículo discutirá las ventajas y desventajas de las bases de datos MySQL y NoSQL en términos de evolución del esquema.

MySQL es una base de datos relacional basada en un esquema fijo de tablas y columnas. Esto significa que el esquema debe definirse de antemano y deben seguirse las reglas para crear y organizar tablas y columnas. En términos de flexibilidad de esquema, MySQL no proporciona mucha libertad para agregar o eliminar campos o cambiar la estructura de la tabla. Sin embargo, MySQL le permite crear tablas temporales, lo que facilita la realización de ciertas operaciones en los datos.

Por otro lado, las bases de datos NoSQL como MongoDB y Cassandra son más flexibles en términos de estructura de datos. Estos programas no usan tablas o columnas, sino que usan un documento o estructura de importancia clave. Esto significa que los datos se pueden agregar o eliminar fácilmente sin tener que seguir

NoSQL

MongoDB

una estructura fija de tablas y columnas. Además, las bases de datos NoSQL le permiten crear esquemas dinámicos, lo que significa que se pueden cambiar en tiempo real sin detener la base de datos.

Una de las principales ventajas de las bases de datos NoSQL en cuanto a la flexibilidad del esquema es que pueden manejar diferentes tipos de datos sin tener que seguir una estructura fija. Por ejemplo, se pueden agregar datos no estructurados como imágenes o archivos de audio sin tener que seguir una estructura específica de tablas y columnas. Además, la flexibilidad del esquema permite a los desarrolladores realizar cambios en la estructura de la base de datos sin tener que detener la aplicación o sistema.

Sin embargo, una de las desventajas de las bases de datos NoSQL en cuanto a la flexibilidad del esquema es que pueden ser más difíciles de consultar en comparación con MySQL. Al no seguir una estructura fija de tablas y columnas, las bases de datos NoSQL pueden requerir un lenguaje de consulta especializado para realizar consultas complejas. Además, la falta de un esquema fijo puede hacer que sea más difícil garantizar la integridad de los datos y evitar la redundancia.

En conclusión, tanto MySQL como las bases de datos NoSQL tienen ventajas y desventajas en cuanto a la flexibilidad del esquema. MySQL tiene la ventaja de tener un esquema fijo y definido de tablas y columnas, lo que facilita la consulta de datos complejos y la integridad de los datos. Por otro lado, las bases de datos NoSQL son altamente flexibles en cuanto a la estructura de los datos, lo que permite una mayor libertad para agregar y eliminar datos, pero puede requerir un lenguaje de consulta especializado y puede ser más difícil de mantener la integridad de los datos. En última instancia, la elección entre MySQL y una base de datos NoSQL dependerá de las necesidades específicas de la aplicación o sistema en cuestión.

CONSISTENCIA DE LOS DATOS

La consistencia de los datos es uno de los aspectos más importantes a considerar al elegir una base de datos para un sistema o aplicación. En este ensayo, se discutirán las ventajas y desventajas de las bases de datos MySQL y NoSQL en términos de consistencia de datos.

NoSQL

MongoDB

MySQL es una base de datos relacional basada en transacciones ACID (Atomismo, Consistencia, Aislamiento y Resistencia), lo que significa que todas las transacciones se manejan de forma segura y se garantiza la integridad de los datos. Esto garantiza que todos los cambios en la base de datos se realicen de forma coherente y que los datos sean precisos y fiables. MySQL también tiene mecanismos de comprobación de la integridad de los datos, como restricciones de clave externa, que ayudan a garantizar la coherencia de los datos. Por otro lado, las bases de datos NoSQL, como MongoDB y Cassandra, ofrecen diferentes enfoques para garantizar la coherencia de los datos. Las bases de datos NoSQL a menudo se basan en un modelo de coherencia eventual, lo que significa que las actualizaciones de datos pueden tardar en propagarse a través de la base de datos. Esto se debe a que las bases de datos NoSQL a menudo transmiten datos a través de varios servidores, lo que puede afectar la coherencia de los datos. Sin embargo, algunas bases de datos NoSQL brindan modelos sólidos de consistencia para garantizar la integridad de los datos en todo momento.

Una de las principales ventajas de MySQL en términos de consistencia de datos es que siempre asegura la integridad de los datos. De hecho, se basa en transacciones ACID y tiene mecanismos para verificar la integridad de los datos. Esto facilita garantizar la precisión y confiabilidad de los datos en aplicaciones y sistemas de misión crítica.

Por otro lado, una de las ventajas de las bases de datos NoSQL en términos de consistencia de datos es que son altamente escalables y pueden manejar grandes volúmenes de datos sin comprometer la disponibilidad o el rendimiento. Además, algunas bases de datos NoSQL proporcionan modelos sólidos de coherencia para garantizar la coherencia de los datos en todo momento. Esto hace que las bases de datos NoSQL sean una opción atractiva para aplicaciones y sistemas que manejan grandes volúmenes de datos y necesitan un alto grado de escalabilidad. Sin embargo, una de las desventajas de las bases de datos NoSQL en términos de consistencia de datos es que a menudo se basan en un modelo de consistencia final, lo que significa que puede haber algún retraso antes de que se completen las actualizaciones de datos transmitidas a través de la base de datos. Esto puede afectar la consistencia de los datos y dificultar la garantía de la precisión y confiabilidad de los datos en aplicaciones y sistemas de misión crítica.

En resumen, las bases de datos MySQL y NoSQL tienen sus ventajas y desventajas cuando se trata de la consistencia de los datos. MySQL proporciona un alto nivel de consistencia de datos a través del modelo de transacción ACID y el mecanismo de verificación de integridad de datos. Por otro lado, las bases de datos NoSQL son

NoSQL

MongoDB

altamente escalables y pueden manejar grandes volúmenes de datos, pero pueden ser lentas para propagar actualizaciones en todos los ámbitos.

CONSULTAS

Las consultas son una parte fundamental de cualquier base de datos, ya que permiten a los usuarios recuperar información relevante. En este ensayo, se discutirán las ventajas y desventajas de las bases de datos MySQL y NoSQL en términos de consultas.

MySQL es una base de datos relacional que utiliza SQL (lenguaje de consulta estructurado) para ejecutar consultas. SQL es un lenguaje poderoso y flexible que permite a los usuarios recuperar información de manera eficiente. Además, MySQL cuenta con una gran cantidad de herramientas y aplicaciones que permiten a los usuarios crear y ejecutar consultas complejas. Por otro lado, las bases de datos NoSQL, como MongoDB y Cassandra, utilizan su propio lenguaje de consulta para recuperar información. Estos lenguajes de consulta son generalmente más simples que SQL y están diseñados para trabajar con datos no estructurados. Además, las bases de datos NoSQL a menudo ofrecen opciones avanzadas de búsqueda y análisis de texto completo.

NoSQL

MongoDB

Una de las principales ventajas de las bases de datos NoSQL en términos de consultas es su capacidad para manejar grandes volúmenes de datos de manera eficiente. Las bases de datos NoSQL utilizan búsquedas de texto completo e índices de valores clave para acelerar las consultas. Además, las bases de datos NoSQL pueden escalar horizontalmente para manejar grandes volúmenes de datos y usuarios simultáneos.

Sin embargo, las bases de datos NoSQL también tienen algunas desventajas cuando se trata de consultas. Primero, debido a que no usan SQL, puede ser difícil para los usuarios migrar de una base de datos relacional a una base de datos NoSQL. Además, las bases de datos NoSQL no brindan una funcionalidad similar a SQL, como la capacidad de realizar operaciones complejas, como uniones y agregados.

En resumen, las bases de datos MySQL y NoSQL tienen ventajas y desventajas cuando se trata de consultas. MySQL proporciona un lenguaje de consulta flexible y potente que permite a los usuarios realizar operaciones complejas, mientras que las bases de datos NoSQL ofrecen una gran escalabilidad y la capacidad de manejar grandes volúmenes de datos. La elección entre bases de datos MySQL y NoSQL dependerá de las necesidades específicas de cada proyecto y de las preferencias individuales del usuario.

Para ilustrar las ventajas y desventajas de las bases de datos MySQL y NoSQL en términos de consultas, podemos considerar los siguientes casos:

mysql:

Supongamos que tenemos una base de datos de ventas en línea y queremos consultar para obtener las ventas totales de cada mes del año anterior. Usando SQL, podemos crear una consulta como esta:

SELECCIONE MES(fecha), SUMA(ligado) DE ventas DONDE AÑO(fecha) = 2021 GRUPO POR MES(día)

NoSQL

MongoDB

Esta consulta utiliza funciones SQL para obtener las ventas totales de cada mes del año anterior y agruparlas por mes. MySQL nos permite realizar operaciones complejas como esta de manera eficiente.

Bases de datos NoSQL:

Supongamos que tenemos una base de datos de reseñas de productos y queremos encontrar todas las reseñas que contienen la palabra "excelente". En una base de datos NoSQL como MongoDB, podemos usar un índice de texto completo para acelerar la consulta. La consulta podría verse así:

```
db.reviews.find({ $texto:  
{ $buscar:  
"excelente" } })
```

Esta consulta utiliza la búsqueda de texto completo para encontrar todas las reseñas que contienen la palabra "excelente". Las bases de datos NoSQL como MongoDB brindan excelentes capacidades de búsqueda y análisis de texto completo.

En resumen, MySQL es una buena opción para proyectos que requieren muchas operaciones complejas y un lenguaje de consulta poderoso como SQL. Por otro lado, una base de datos NoSQL como MongoDB es una buena opción para proyectos que requieren una alta escalabilidad y la capacidad de manejar grandes volúmenes de datos no estructurados.

NoSQL

MongoDB

ESTUDIO COMPARATIVO STATE –OF- THE- ART

Las bases de datos NoSQL son un tipo de bases de datos diseñadas para manejar grandes cantidades de datos no estructurados o semiestructurados, como documentos ,gráficos ,datos de sensores , etc.A diferencia de las bases de datos relacionales tradicionales, las bases de datos NoSQL no requieren un esquema fijo definido de antemano ,lo que hace más flexibles y escalables.

Existen varios tipos de datos NoSQL, incluyendo:

1.Bases de datos de documentos: Las bases de datos de documentos almacenan datos en formato de documentos como JSON o BSON. Ejemplos populares de bases de datos documentales incluyen MongoDB ,CouchDB y RavenDB.

2.Bases de datos de grafos: Las bases de datos de grafos almacenan los datos en forma de nodos y relaciones. Ejemplos populares de bases de datos de grafos incluyen Neo4j,ArangoDB y OrienteDB.

3.Bases de datos de clave –valor: Las bases de datos clave-valor almacenan datos en forma de pares de clave-valor.Ejemplos populares de bases de datos clave-valor incluyen Redis, Riak y Amazon DynamoDB .

4.Bases de datos de columnas: Las bases de datos de columnas almacenan los datos en forma de columnas en lugar de filas. Ejemplos populares de bases de datos de columnas incluyen Apache Cassandra y HBase.

ANÁLISIS COMPARATIVO DE LAS DIFERENTES BASES DE DATOS NoSQL

1 Escalabilidad: Todas las bases de datos NoSQL están diseñadas para ser escalables horizontalmente, lo que significa que se pueden agregar más nodos al sistema para aumentar la capacidad de almacenamiento y procesamiento de datos. Sin embargo, las bases de datos de clave-valor y de columnas suelen ser más escalables que las bases de datos de documentos y grafos.

2.Flexibilidad:Las bases de datos NoSQL son más flexibles que las bases de datos relacionales tradicionales, ya que no requieren un esquema definido de antemano .Sin embargo ,las bases de datos de documentos son más flexibles que las bases de datos de grafos y de columnas, ya que permiten almacenar datos no estructurados o semiestructurados.

3.Rendimiento:La bases de datos NoSQL suelen tener mejor rendimiento que las bases de datos relacionales tradicionales en aplicaciones que requieren una gran cantidad de lecturas y escrituras de datos a tiempo real. Las bases de datos clave-

NoSQL

MongoDB

valor suelen ser las más rápidas, seguidas de las bases de datos de columnas, de documentos y de grafos.

4.Consistencia: Las bases de datos NoSQL suelen ser menos consistentes que las bases de datos relacionales tradicionales, ya que no garantizan la consistencia de los datos en tiempo real. Sin embargo algunas bases de datos como MongoDB y CouchDB , ofrecen opciones de consistencia que permiten ajustar el nivel de consistencia según las necesidades de la aplicación.

MONGODB

HISTORIA

MongoDB es una base de datos NoSQL que fue desarrollada por la compañía MongoDB Inc. y lanzada al mercado en 2009. Desde entonces, ha experimentado una evolución constante, convirtiéndose en una de las bases de datos más populares del mundo.

La historia de MongoDB comienza en el año 2007, cuando Dwight Merriman y Eliot Horowitz, dos ingenieros de DoubleClick (una compañía de publicidad en línea), empezaron a trabajar en un proyecto interno para simplificar la administración de bases de datos. Su objetivo era crear una base de datos escalable y flexible que pudiera manejar grandes volúmenes de datos.

Inicialmente, el proyecto se llamó "10gen" y se centró en la creación de una base de datos orientada a objetos. Sin embargo, después de varios intentos fallidos, Merriman y Horowitz cambiaron de enfoque y empezaron a trabajar en una base de datos NoSQL.

NoSQL

MongoDB

En 2009, 10gen lanzó MongoDB 1.0, la primera versión de su base de datos NoSQL. Desde entonces, la compañía ha lanzado varias versiones más, con mejoras significativas en la funcionalidad y la escalabilidad.

Uno de los mayores beneficios de MongoDB es su flexibilidad. A diferencia de las bases de datos relacionales tradicionales, MongoDB no tiene un esquema fijo y puede manejar datos estructurados y no estructurados. Además, MongoDB puede escalar horizontalmente, lo que significa que puede manejar grandes volúmenes de datos distribuyéndolos en múltiples servidores.

En los primeros años de MongoDB, la mayoría de los usuarios eran startups y empresas en crecimiento que buscaban una alternativa más escalable y flexible a las bases de datos relacionales tradicionales. Sin embargo, con el tiempo, MongoDB se ha vuelto popular entre una amplia variedad de organizaciones, incluyendo empresas Fortune 500, gobiernos y organizaciones sin fines de lucro.

En 2013, MongoDB Inc. se separó de 10gen y se convirtió en una compañía independiente. Desde entonces, ha continuado mejorando la funcionalidad de MongoDB y ha lanzado varias herramientas y servicios adicionales, como MongoDB Atlas (un servicio de base de datos en la nube) y MongoDB Stitch (una plataforma de backend como servicio).

En 2017, MongoDB Inc. hizo una oferta pública inicial (IPO) y comenzó a cotizar en la bolsa de Nueva York. Desde entonces, la compañía ha continuado expandiéndose, abriendo oficinas en todo el mundo y adquiriendo otras compañías para mejorar su oferta de productos.

En la actualidad, MongoDB es una de las bases de datos NoSQL más populares del mundo, con millones de usuarios en todo el mundo. La compañía continúa innovando y mejorando su oferta de productos, y se espera que siga siendo una fuerza importante en el mundo de las bases de datos en los próximos años.

NoSQL

MongoDB

ARQUITECTURA

MongoDB es una base de datos NoSQL orientada a documentos. En lugar de guardar los datos de registros, MongoDB guarda los datos en documentos almacenados en BSON, que es una representación binaria de JSON. MongoDB es un sistema de código abierto escrito en C++.

MongoDB tiene varias características que lo hacen atractivo para los desarrolladores. Una de ellas es la capacidad de realizar consultas ad hoc, lo que significa que se pueden realizar todo tipo de consultas, como búsqueda por campos, consulta de rangos y expresiones regulares. MongoDB también tiene la capacidad de ejecutarse de manera simultánea en múltiples servidores, ofreciendo un balanceo de carga de servicio de replicación de datos,

La arquitectura de MongoDB está diseñada para satisfacer las demandas de las aplicaciones modernas. MongoDB también tiene la capacidad de ejecutarse de manera simultánea en múltiples servidores, ofreciendo un balanceo de carga o servicio de replicación de datos.

La arquitectura de MongoDB está diseñada para satisfacer las demandas de las aplicaciones modernas, MongoDB tiene un modelo de datos de documentos que presenta la mejor manera de trabajar con datos. También tiene un diseño de sistemas distribuidos que le permite colocar inteligentemente los datos. También tiene un diseño de sistemas distribuidos que le permiten colocar inteligentemente los datos donde necesita, ofrece una experiencia unificada que le da libertad de ejecutar en cualquier lugar, lo que permite proteger su trabajo y eliminar el bloqueo del proveedor.

NoSQL

MongoDB

En resumen, MongoDB es una base de datos NoSQL orientada a documentos que utiliza BSON para almacenar datos. Tiene características atractivas para los desarrolladores, como consultas ad hoc, balaceo de carga. La arquitectura de MongoDB está diseñada para satisfacer las demandas de las aplicaciones modernas y ofrece una experiencia unificada que las da libertad de ejecutar cualquier lugar.

VENTAJAS Y DESVENTAJAS

Las principales ventajas de esta base de datos son:

Ideal para entornos con pocos recursos de computación

Cualquier servidor o cualquier ordenador personal sirve para montar MongoDB y tener un servidor para nuestros proyectos.

Es una herramienta con coste bajo

Al ser una herramienta de código abierto se paga licencia, lo único que se paga es el soporte, en caso de necesitarlo.

Tiene gran documentación

Posee una documentación muy buena, muy amplia y detallada en comparación con otras bases de datos NoSQL.

Es un complemento perfecto para JavaScript S

Si eres desarrollador de aplicaciones utilizando este lenguaje, podrás utilizar toda la potencia de sus funciones y operadores en MongoDB.

DESVENTAJAS DE MONGODB

NoSQL

MongoDB

Algunas desventajas que presenta MongoDB son:

No es una base de datos adecuada para aplicaciones con transacciones completas

Para este tipo de aplicaciones, las bases de datos relacionales son más idóneas.

Es una tecnología joven

A pesar de estas ampliamente usada en la actualidad, sigue siendo tecnología joven.

No permite Joins para consultas

Esta es una de las grandes desventajas de MongoDB, y es que no permite hacer Joins para consultas en las que se combinan o relacionan diferentes tablas. La forma de ejecutar ese tipo de consultas en MongoDB se hace de otra manera.

CASOS DE USO

VISION UNIFICADA

La primera de las categorías incluye a la empresa MetLife. El gigante de los seguros trabaja con MongoDB. La aplicación consiste en la creación de un repositorio central que ofrece una visión a partir de muchas fuentes de datos que provienen de repositorio o sistemas.

INTERNET DE LAS COSAS

En el mundo de internet, todos los dispositivos están conectados entre sí, generando y compartiendo información. Así la empresa Bosh está poniendo a prueba una aplicación que es capaz de capturar datos del vehículo como sistema de frenado, la dirección asistida, los limpiaparabrisas, etc. Con todos estos datos capturados, se pueden hacer diagnósticos de necesidad de mantenimiento preventivo.

TECNOLOGÍAS MÓVILES

The Weather Channel utiliza MongoDB para sus aplicaciones móviles que disfrutan de cerca de 40 millones de personas en todo el mundo, La idea de que MongoDB se convierta en la parte backend del mundo de los smartphones y tablets toma cada vez mas fuerza.

ANALÍTICA EN TIEMPO REAL

Cuando hablamos de analítica en tiempo real nos referimos a la necesidad de conseguir resultados de manera inmediata. La ciudad de Chicago ha desarrollado una aplicación llamada WindyGrid que esta basada en MongoDB, y recoge datos

NoSQL

MongoDB

de policía, transporte e incendios. La aplicación además notifica alertas por obras en carretera, retrasos en recolección de basura, quejas por ruido o tweets públicos, entre otros.

ALMACENAMIENTO DE CONTENIDO

MongoDB es adecuado para almacenar datos multimedia es adecuado datos multimedia como imágenes, videos y archivos de audio. Pueden manejar grandes archivos y proporciona una fácil integración con otros sistemas de almacenamiento de archivos como Amazon S3.

APLICACIONES MÓVILES

MongoDB es una opción para almacenar datos de aplicaciones móviles, especialmente en aplicaciones con grandes volúmenes de datos y alta escalabilidad.

En general, MongoDB es adecuado para cualquier caso de uso que involucre que grandes cantidades de datos, alta escalabilidad y la necesidad de un rendimiento rápido y un alta disponibilidad.

REQUISITOS DEL SISTEMA

Los requisitos del sistema para MongoDB dependen de varios factores, como la cantidad de datos que se desean almacenar, la cantidad de usuarios que accederán a la base de datos y el tipo de consultas que se realizarán.

En general, los requisitos mínimos para MongoDB son los siguientes:

Sistema operativo: MongoDB es compatible con una amplia gama de sistemas operativos, incluidos Windows, macOS, Linux y varias distribuciones de Unix.

Procesador: MongoDB se ejecuta en sistemas de 64 bits y requiere un procesador de 64 bits.

Memoria RAM: se recomienda una memoria RAM mínima de 4 GB, aunque la cantidad de memoria necesaria dependerá de la cantidad de datos que se desean almacenar y de la complejidad de las consultas.

NoSQL

MongoDB

Espacio en disco: MongoDB requiere un espacio en disco suficiente para almacenar los datos. El tamaño de la base de datos dependerá de la cantidad de datos que se deseen almacenar.

Red: MongoDB utiliza el protocolo TCP/IP para la comunicación entre los diferentes componentes del sistema. Se recomienda una conexión de red rápida y confiable.

Además de estos requisitos mínimos, es importante considerar la escalabilidad del sistema a medida que la cantidad de datos y usuarios aumenta. MongoDB está diseñado para ser escalable horizontalmente, lo que significa que se pueden agregar nodos adicionales para manejar la carga de trabajo adicional. Por lo tanto, es importante planificar la arquitectura del sistema desde el principio para permitir la escalabilidad futura.

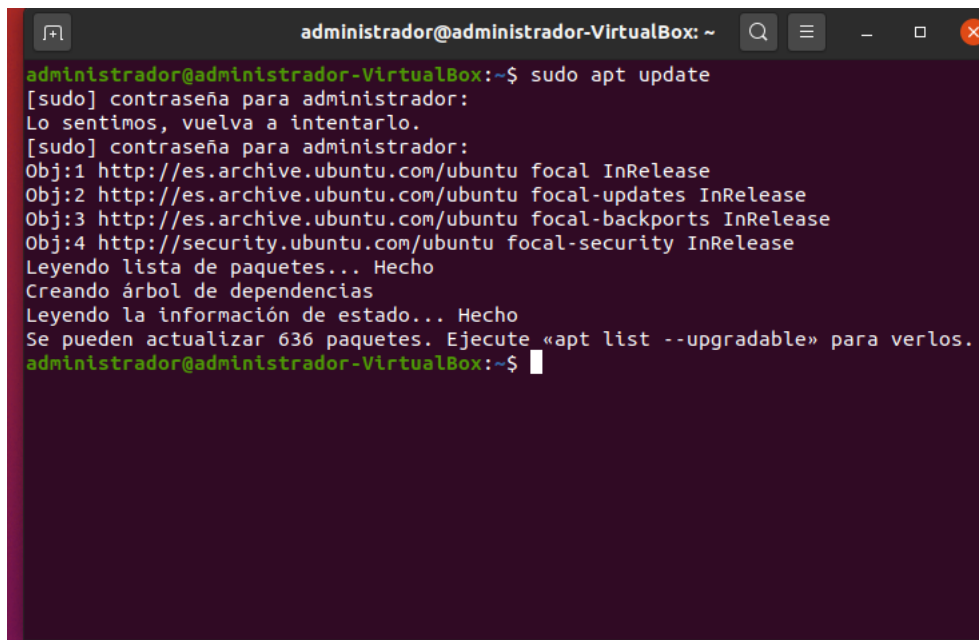
NoSQL

MongoDB

DESCARGA E INSTALCIÓN DE MONGODB

Descarga e instalación de MongoDB en Linux

Antes de instalar MongoDB debemos asegurarnos de que están los paquetes del sistema actualizados.

A terminal window titled 'administrador@administrador-VirtualBox: ~' with standard window controls. The terminal shows the command 'sudo apt update' being executed. The output includes a password prompt, a message about package lists, a list of four objects (http://es.archive.ubuntu.com/ubuntu focal InRelease, http://es.archive.ubuntu.com/ubuntu focal-updates InRelease, http://es.archive.ubuntu.com/ubuntu focal-backports InRelease, and http://security.ubuntu.com/ubuntu focal-security InRelease), and a confirmation that 636 packages can be updated. The prompt returns to 'administrador@administrador-VirtualBox:~\$' with a cursor.

```
administrador@administrador-VirtualBox:~$ sudo apt update
[sudo] contraseña para administrador:
Lo sentimos, vuelva a intentarlo.
[sudo] contraseña para administrador:
Obj:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease
Obj:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 636 paquetes. Ejecute «apt list --upgradable» para verlos.
administrador@administrador-VirtualBox:~$
```

Instalamos el paquete de MongoDB

NoSQL

MongoDB

```
administrador@administrador-VirtualBox: ~  
administrador@administrador-Virtua... x administrador@administrador-Virtua... x  
administrador@administrador-VirtualBox:~$ sudo apt install mongodb  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.  
libfprint-2-tod1 libfwupdplugin1 libllvm10 libxmlb1 linux-headers-5.4.0-42  
linux-headers-5.4.0-42-generic linux-image-5.4.0-42-generic  
linux-modules-5.4.0-42-generic linux-modules-extra-5.4.0-42-generic  
ubuntu-advantage-desktop-daemon  
Utilice «sudo apt autoremove» para eliminarlos.  
Se instalarán los siguientes paquetes adicionales:  
libboost-program-options1.71.0 libgoogle-perftools4 libpcrecpp0v5  
libtcmalloc-minimal4 libyaml-cpp0.6 mongo-tools mongodb-mongoclient  
mongodb-server mongodb-server-core  
Se instalarán los siguientes paquetes NUEVOS:  
libboost-program-options1.71.0 libgoogle-perftools4 libpcrecpp0v5  
libtcmalloc-minimal4 libyaml-cpp0.6 mongo-tools mongodb-mongoclient  
mongodb-server mongodb-server-core  
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 3 no actualizados.  
Se necesita descargar 56,3 MB de archivos.  
Se utilizarán 229 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n] s  
Des:1 http://es.archive.ubuntu.com/ubuntu focal/main amd64 libboost-program-opti
```

```
administrador@administrador-VirtualBox: ~  
administrador@administrador-VirtualBox:~$ sudo systemctl status mongod  
● mongod.service - An object/document-oriented database  
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)  
   Active: active (running) since Wed 2023-03-22 23:51:24 CET; 2min 59s ago  
     Docs: man:mongod(1)  
  Main PID: 4544 (mongod)  
    Tasks: 23 (limit: 2319)  
   Memory: 42.7M  
    CGroup: /system.slice/mongod.service  
            └─4544 /usr/bin/mongod --unixSocketPrefix=/run/mongod --config />  
  
mar 22 23:51:24 administrador-VirtualBox systemd[1]: Started An object/document>  
línea 1-11/11 (END)
```

2º Comprobamos el servicio usando el siguiente comando

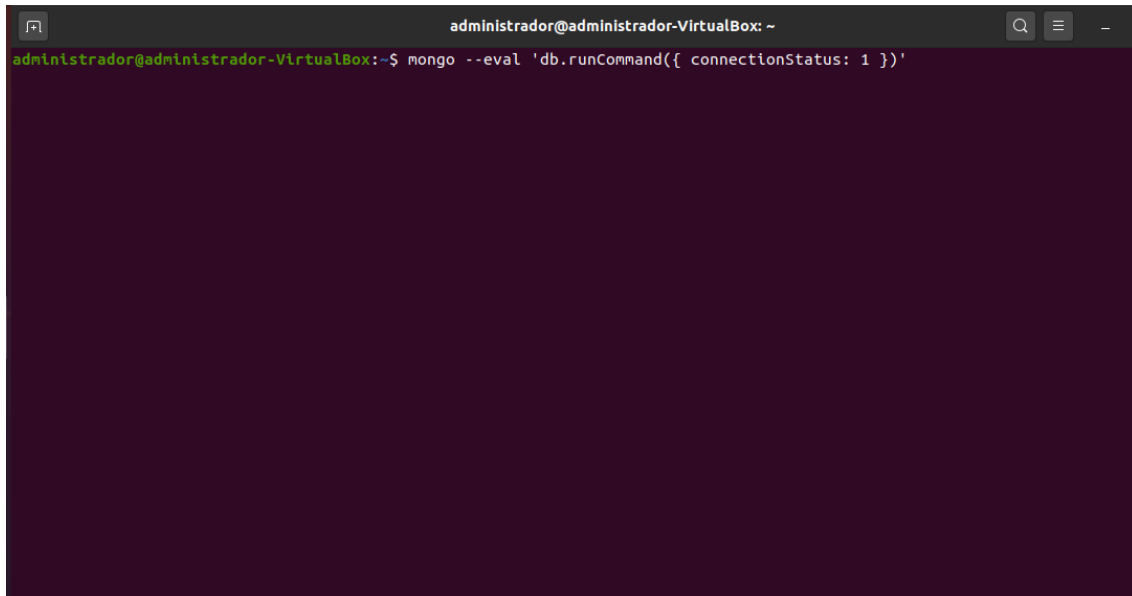
3º Podemos verificar este servicio en profundidad estableciendo conexión con el servidor de la base de datos y ejecutando el siguiente comando de diagnóstico. Con

NoSQL

MongoDB

esto nos mostrará la versión actual de base de datos, la dirección y el puerto del servidor, y el resultado del comando status:

mongo --eval 'db.runCommand({ connectionStatus: 1 })'



```
administrador@administrador-VirtualBox: ~  
administrador@administrador-VirtualBox:~$ mongo --eval 'db.runCommand({ connectionStatus: 1 })'
```



```
administrador@administrador-VirtualBox:~$ mongo --eval 'db.runCommand({ connectionStatus: 1 })'  
MongoDB shell version v3.6.8  
connecting to: mongodb://127.0.0.1:27017  
explicit session: session { "id" : UUID("6c031738-dee7-46ac-b127-8cc6da22eSec") }  
MongoDB server version: 3.6.8  
  
  "authInfo" : {  
    "authenticatedUsers" : [ ],  
    "authenticatedUserRoles" : [ ]  
  },  
  "ok" : 1  
administrador@administrador-VirtualBox:~$
```

Un valor 1 para el campo ok en respuesta indica que el servidor funciona correctamente.

NoSQL

MongoDB

GESTION DEL SERVICIO

En esta apartado vamos a configuraremos MongoDB como servicio de **system**, lo que significa que se puede administrar utilizazndo comandos de **systemctl** estándares,junto con todos los demás servicios de sistema Ubuntu.

Para verificar el servidor ,escribimos lo siguiente:

```
sudo systemctl status mongodb
```

Podemos detener el servidor en cualquier momento escribiendo lo siguiente:

```
sudo systemctl stop mongodb
```

Para reanudar el servicio cuando esta parado ,escribimos los siguiente:

```
sudo systemctl start mongodb
```

Tambien podemos reanudar el servidor con el siguiente comando:

```
sudo systemctl restart mongodb
```

Por defecto,MongoDB se configura para iniciarse de forma automática con el servidor.Si desea desactivar el inicio automático,escribimos lo siguiente:

```
sudo systemctl disable mongodb
```

Puede volver a habilitar el inicio automático en cualquier momento con el siguiente:

```
sudo systemctl enable mongodb
```

CREACION DE UNA BASE DE DATOS Y UNA COLECCIÓN

IMPORTAR UNA COLECCIÓN A MONGODB

NoSQL

MongoDB

Para exportar una base de datos podemos usar una base de datos de muestra en este caso sobre restaurantes. En este caso está en formato.json y se puede descargar con **wget** de esta manera:

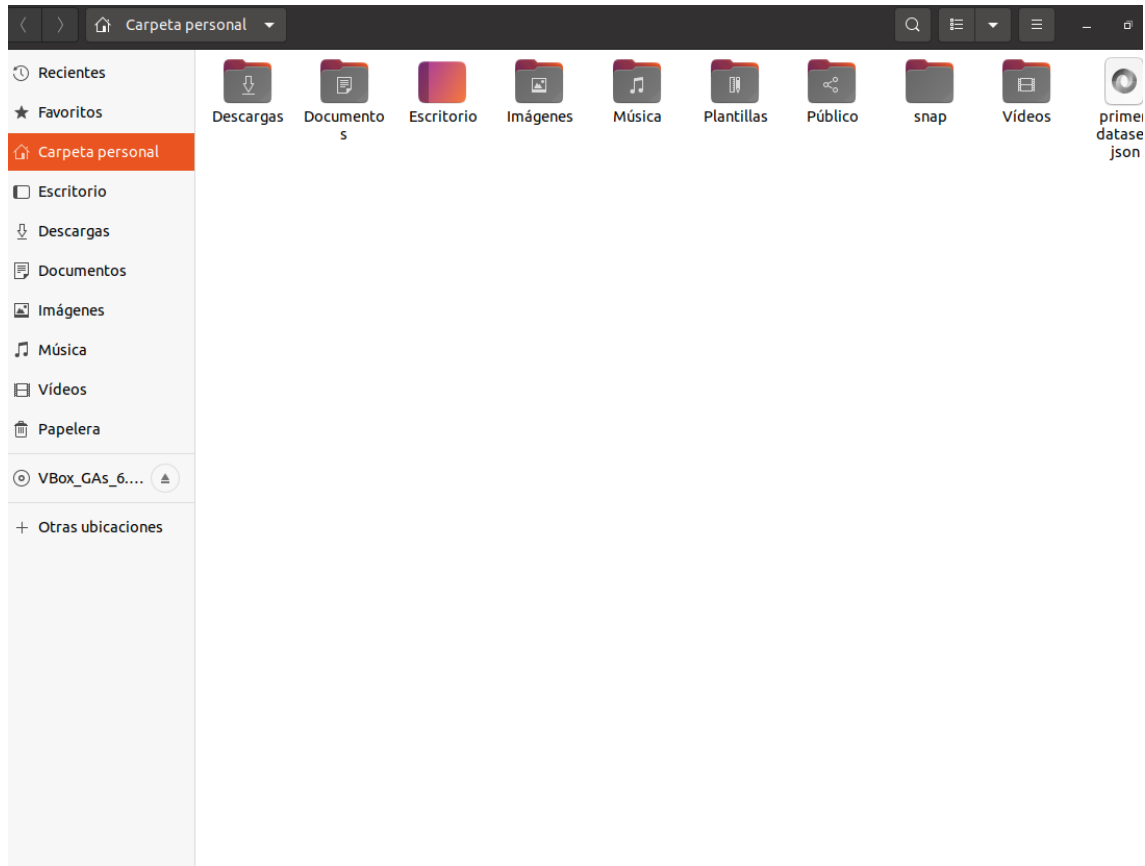
wget <https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json>

A terminal window titled 'administrador@administrador-VirtualBox: ~' showing the execution of a wget command. The command is 'wget https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json'. The output shows the file being resolved, connected to, and downloaded successfully. The file size is 11,32M and it was downloaded at 8,04MB/s in 1,4s. The terminal text is as follows:

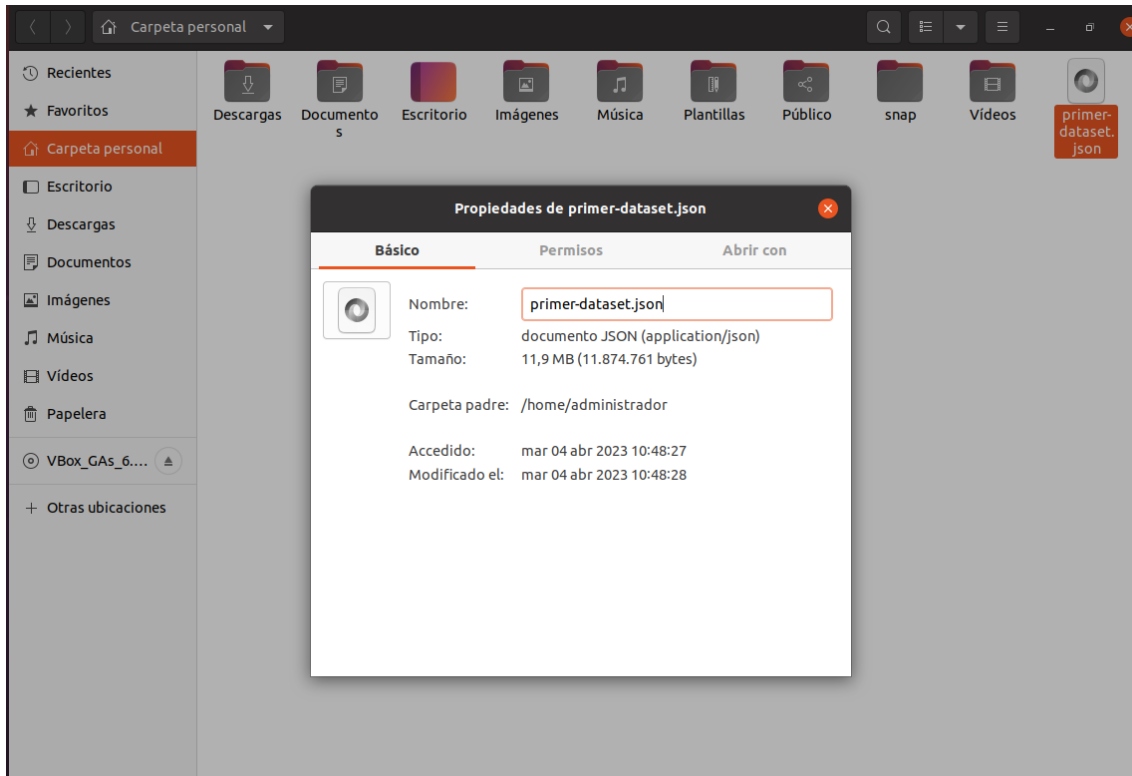
```
administrador@administrador-VirtualBox: ~  
administrador@administrador-VirtualBox:~$ wget https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json  
--2023-04-04 10:48:26-- https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json  
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.108.133, 185.199.109.133, ...  
Conectando con raw.githubusercontent.com (raw.githubusercontent.com)[185.199.110.133]:443... conectado.  
Petición HTTP enviada, esperando respuesta... 200 OK  
Longitud: 11874761 (11M) [text/plain]  
Guardando como: "primer-dataset.json"  
  
primer-dataset.json 100%[=====>] 11,32M 8,04MB/s en 1,4s  
  
2023-04-04 10:48:28 (8,04 MB/s) - "primer-dataset.json" guardado [11874761/11874761]  
  
administrador@administrador-VirtualBox:~$
```

NoSQL

MongoDB



NoSQL MongoDB



Una vez completada la descarga, deberíamos tener un archivo llamado `primer-dataset.json` (12MB) en el directorio actual, importaremos los datos de este archivo a una vez de este archivo a una nueva base de datos llamado `newdb` y una colección llamada `restaurants`.

Utilizamos el comando `mongoimport` de esta manera:

```
sudo mongoimport --db newdb --collection restaurants --file primer-dataset.json
```

```
primer-dataset.json 100%[=====] 11,32M 8,04MB/s en 1,4s
2023-04-04 10:48:28 (8,04 MB/s) - "primer-dataset.json" guardado [11874761/11874761]
administrador@administrador-VirtualBox:~$ sudo mongoimport --db newdb --collection restaurants --file primer-dataset.json
[sudo] contraseña para administrador:
2023-04-04T11:07:04.716+0200 connected to: localhost
2023-04-04T11:07:05.578+0200 imported 25359 documents
administrador@administrador-VirtualBox:~$
```

Como podemos ver en la imagen con el comando anterior se han importado 25359 documentos. Dado que no teníamos una base de datos llamada `newdb`, MongoDB la creó de forma automática.

Con este paso vamos a verificar la importación.

NoSQL

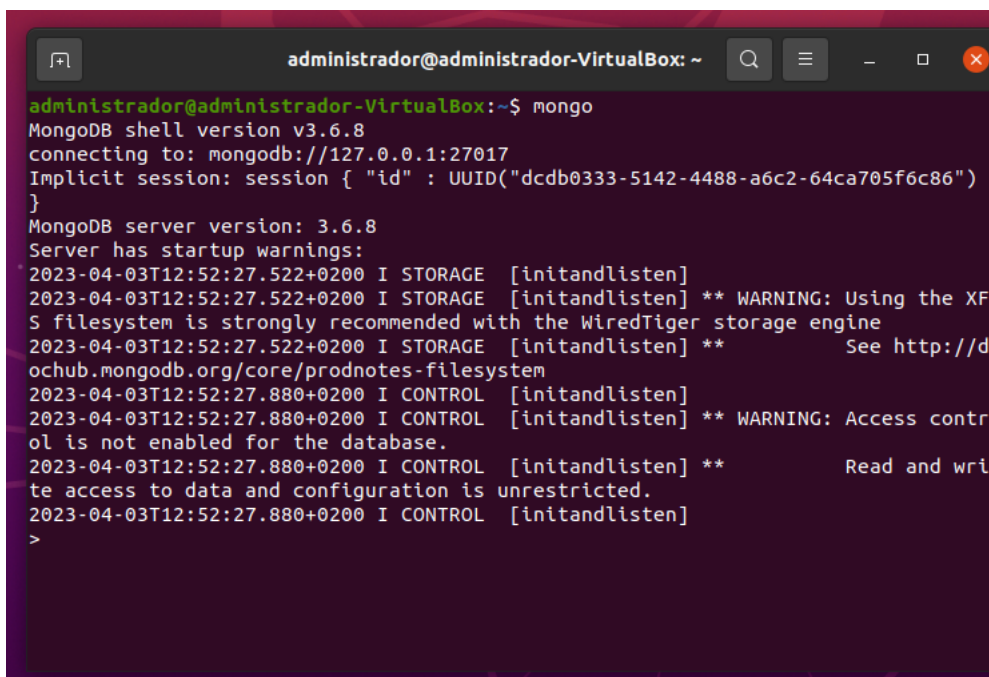
MongoDB

Nos conectamos a la base de datos [newdb](#) que recientemente hemos creado:

Para crear una base de datos y una colección en MongoDB, sigue estos pasos:

Inicia el servidor de MongoDB.

Abre la terminal o consola de comandos y accede a la interfaz de MongoDB escribiendo el siguiente comando:

A screenshot of a terminal window titled 'administrador@administrador-VirtualBox: ~'. The terminal shows the command 'mongo' being executed, which opens the MongoDB shell. The output includes the shell version (v3.6.8), the connection string (mongodb://127.0.0.1:27017), the implicit session ID, and the MongoDB server version (3.6.8). It also displays several startup warnings from the log, such as 'WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine' and 'WARNING: Access control is not enabled for the database.' The prompt '>' is visible at the bottom of the terminal.

```
administrador@administrador-VirtualBox: ~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("dcd03333-5142-4488-a6c2-64ca705f6c86") }
MongoDB server version: 3.6.8
Server has startup warnings:
2023-04-03T12:52:27.522+0200 I STORAGE [initandlisten]
2023-04-03T12:52:27.522+0200 I STORAGE [initandlisten] ** WARNING: Using the XFS
filesystem is strongly recommended with the WiredTiger storage engine
2023-04-03T12:52:27.522+0200 I STORAGE [initandlisten] ** See http://d
ochub.mongodb.org/core/prodnotes-filesystem
2023-04-03T12:52:27.880+0200 I CONTROL [initandlisten]
2023-04-03T12:52:27.880+0200 I CONTROL [initandlisten] ** WARNING: Access contr
ol is not enabled for the database.
2023-04-03T12:52:27.880+0200 I CONTROL [initandlisten] ** Read and wri
te access to data and configuration is unrestricted.
2023-04-03T12:52:27.880+0200 I CONTROL [initandlisten]
>
```

Para saber las bases de datos que tenemos debemos poner el comando [show dbs](#)

NoSQL

MongoDB

```
administrador@administrador-VirtualBox: ~  
administrador@administrador-VirtualBox:~$ mongo  
MongoDB shell version v3.6.8  
connecting to: mongodb://127.0.0.1:27017  
Implicit session: session { "id" : UUID("26a89a0f-f7db-44e5-bd86-173e304e87f7")  
}  
MongoDB server version: 3.6.8  
Server has startup warnings:  
2023-04-03T12:52:27.522+0200 I STORAGE [initandlisten]  
2023-04-03T12:52:27.522+0200 I STORAGE [initandlisten] ** WARNING: Using the XFS  
filesystem is strongly recommended with the WiredTiger storage engine  
2023-04-03T12:52:27.522+0200 I STORAGE [initandlisten] ** See http://d  
ochub.mongodb.org/core/prodnotes-filesystem  
2023-04-03T12:52:27.880+0200 I CONTROL [initandlisten]  
2023-04-03T12:52:27.880+0200 I CONTROL [initandlisten] ** WARNING: Access contr  
ol is not enabled for the database.  
2023-04-03T12:52:27.880+0200 I CONTROL [initandlisten] ** Read and wri  
te access to data and configuration is unrestricted.  
2023-04-03T12:52:27.880+0200 I CONTROL [initandlisten]  
> show dbs  
admin 0.000GB  
config 0.000GB  
local 0.000GB  
>
```

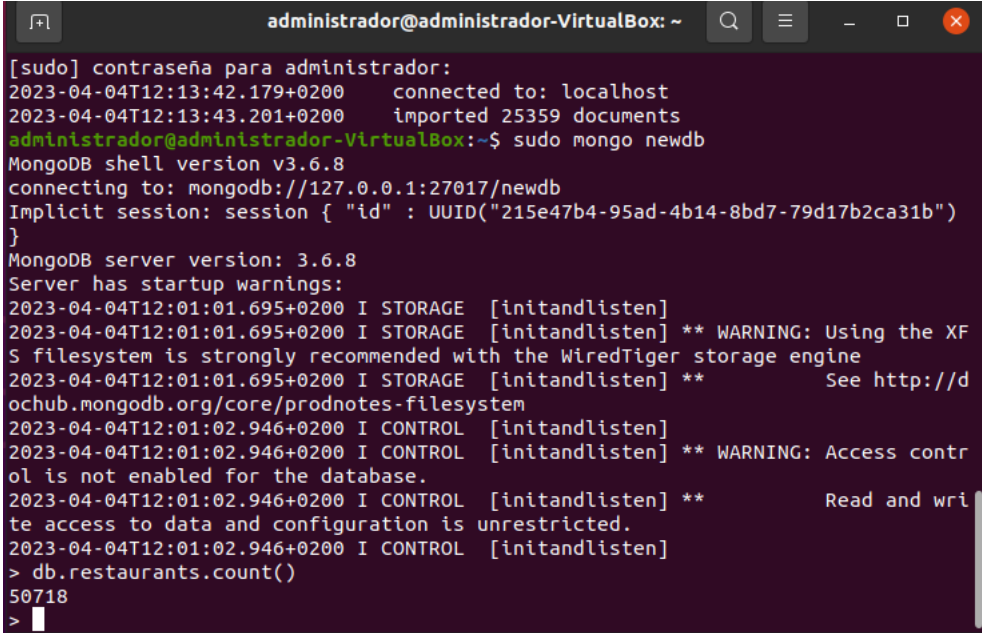
```
bye  
administrador@administrador-VirtualBox:~$ sudo mongo newdb  
MongoDB shell version v3.6.8  
connecting to: mongodb://127.0.0.1:27017/newdb  
Implicit session: session { "id" : UUID("6249edcd-73a2-49d7-a0b7-8b0bc9aa0e00")  
}  
MongoDB server version: 3.6.8  
Server has startup warnings:  
2023-04-04T10:46:38.107+0200 I STORAGE [initandlisten]  
2023-04-04T10:46:38.107+0200 I STORAGE [initandlisten] ** WARNING: Using the XFS  
filesystem is strongly recommended with the WiredTiger storage engine  
2023-04-04T10:46:38.107+0200 I STORAGE [initandlisten] ** See http://d  
ochub.mongodb.org/core/prodnotes-filesystem  
2023-04-04T10:46:39.323+0200 I CONTROL [initandlisten]  
2023-04-04T10:46:39.323+0200 I CONTROL [initandlisten] ** WARNING: Access contr  
ol is not enabled for the database.  
2023-04-04T10:46:39.323+0200 I CONTROL [initandlisten] ** Read and wri  
te access to data and configuration is unrestricted.  
2023-04-04T10:46:39.323+0200 I CONTROL [initandlisten]  
>
```


NoSQL

MongoDB

Vamos a contar los documentos de la colección de restaurantes con el comando:

db.restaurants.count()



```
administrador@administrador-VirtualBox: ~  
[sudo] contraseña para administrador:  
2023-04-04T12:13:42.179+0200    connected to: localhost  
2023-04-04T12:13:43.201+0200    imported 25359 documents  
administrador@administrador-VirtualBox:~$ sudo mongo newdb  
MongoDB shell version v3.6.8  
connecting to: mongodb://127.0.0.1:27017/newdb  
Implicit session: session { "id" : UUID("215e47b4-95ad-4b14-8bd7-79d17b2ca31b")  
}  
MongoDB server version: 3.6.8  
Server has startup warnings:  
2023-04-04T12:01:01.695+0200 I STORAGE [initandlisten]  
2023-04-04T12:01:01.695+0200 I STORAGE [initandlisten] ** WARNING: Using the XFS  
filesystem is strongly recommended with the WiredTiger storage engine  
2023-04-04T12:01:01.695+0200 I STORAGE [initandlisten] ** See http://d  
ochub.mongodb.org/core/prodnotes-filesystem  
2023-04-04T12:01:02.946+0200 I CONTROL [initandlisten]  
2023-04-04T12:01:02.946+0200 I CONTROL [initandlisten] ** WARNING: Access contr  
ol is not enabled for the database.  
2023-04-04T12:01:02.946+0200 I CONTROL [initandlisten] ** Read and wri  
te access to data and configuration is unrestricted.  
2023-04-04T12:01:02.946+0200 I CONTROL [initandlisten]  
> db.restaurants.count()  
50718  
>
```

El resultado mostrará 25359, que es número de documentos importados. Para obtener una comprobación aún mejor, puede seleccionar el primer documento de la colección de restaurantes de esta manera:

db.restaurants.findOne()

NoSQL

MongoDB

```
> db.restaurants.findOne()
{
  "_id" : ObjectId("642be8b87f09e0928009483d"),
  "address" : {
    "building" : "1007",
    "coord" : [
      -73.856077,
      40.848447
    ],
    "street" : "Morris Park Ave",
    "zipcode" : "10462"
  },
  "borough" : "Bronx",
  "cuisine" : "Bakery",
  "grades" : [
    {
      "date" : ISODate("2014-03-03T00:00:00Z"),
      "grade" : "A",
      "score" : 2
    },
    {
      "date" : ISODate("2013-09-11T00:00:00Z"),
      "grade" : "A",
      "score" : 6
    },
    {
      "date" : ISODate("2013-01-24T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    },
    {
      "date" : ISODate("2011-11-23T00:00:00Z"),
      "grade" : "A",
      "score" : 9
    },
    {
      "date" : ISODate("2011-03-10T00:00:00Z"),
      "grade" : "B",
      "score" : 14
    }
  ],
  "name" : "Morris Park Bake Shop",
  "restaurant_id" : "30075445"
}
> █
```

Para salir de la consulta escribimos el comando exit.

NoSQL

MongoDB

EXPORTAR INFORMACIÓN DESDE MONGODB

Para exportar información desde MongoDB utilizamos el comando **mongoexport** .

Un ejemplo sencillo de **mongoexport** sería exportar la colección de restaurantes desde la base de datos **newdb** que hemos importado anteriormente. Podemos hacerlo de la siguiente manera.

En el comando anterior, se utilizó **-db** para especificar la base de datos, **-c** para la colección y **..out** ,para el archivo en que se guardan los datos.

Output

2020-11-11T19:39:57.595+0000 connected to: mongodb://localhost/

2020-11-11T19:39:58.619+0000 [#####.....]
newdb.restaurants 16000/25359 (63.1%)

2020-11-11T19:39:58.871+0000
[#####] newdb.restaurants
25359/25359 (100.0%)

2020-11-11T19:39:58.871+0000 exporte

```
xubu@xubu:~$ sudo mongoimport --db newdb --collection ciudades --drop --file /home/xubu/Descargas/zips.json
2023-03-21T13:37:07.013+0100 connected to: mongodb://localhost/
2023-03-21T13:37:07.014+0100 dropping: newdb.ciudades
2023-03-21T13:37:07.038+0100 15 document(s) imported successfully. 0 document(s) failed to import.
```

d 25359 records

NoSQL

MongoDB

```
xubu@xubu:~$ sudo mongoimport --db newdb --collection ciudades --drop --file /home/xubu/Descargas/zip.js.json
2023-03-21T13:37:07.013+0100   connected to: mongod://localhost/
2023-03-21T13:37:07.014+0100   dropping: newdb.ciudades
2023-03-21T13:37:07.038+0100   15 document(s) imported successfully. 0 document(s) failed to import.
```

```
> show dbs
admin      0.000GB
config     0.000GB
local      0.000GB
newdb      0.012GB
> use newdb
switched to db newdb
> show collections
ciudades
restaurants
> show ciudades
uncaught exception: Error: don't know how to show [ciudades] :
shellHelper.show@src/mongo/shell/utils.js:1191:11
shellHelper@src/mongo/shell/utils.js:819:15
@(shellhelp2):1:1
> show ciudades
uncaught exception: Error: don't know how to show [ciudades] :
shellHelper.show@src/mongo/shell/utils.js:1191:11
shellHelper@src/mongo/shell/utils.js:819:15
@(shellhelp2):1:1
> db.collections.find()
> db.ciudades.find()
{ "_id" : "01001", "city" : "AGAWAM", "loc" : [ -72.622739, 42.070206 ], "pop" : 15338, "state" : "MA" }
{ "_id" : "01002", "city" : "CUSHMAN", "loc" : [ -72.51565, 42.377017 ], "pop" : 36963, "state" : "MA" }
{ "_id" : "01005", "city" : "BARRE", "loc" : [ -72.108354, 42.409698 ], "pop" : 4546, "state" : "MA" }
{ "_id" : "01007", "city" : "BELCHERTOWN", "loc" : [ -72.410953, 42.275103 ], "pop" : 10579, "state" : "MA" }
{ "_id" : "01008", "city" : "BLANDFORD", "loc" : [ -72.936114, 42.182949 ], "pop" : 1240, "state" : "MA" }
{ "_id" : "01010", "city" : "BRIMFIELD", "loc" : [ -72.188455, 42.116543 ], "pop" : 3706, "state" : "MA" }
{ "_id" : "01011", "city" : "CHESTER", "loc" : [ -72.988761, 42.279421 ], "pop" : 1688, "state" : "MA" }
{ "_id" : "01012", "city" : "CHESTERFIELD", "loc" : [ -72.833309, 42.38167 ], "pop" : 177, "state" : "MA" }
{ "_id" : "01013", "city" : "CHICOPEE", "loc" : [ -72.607962, 42.162046 ], "pop" : 23396, "state" : "MA" }
{ "_id" : "01020", "city" : "CHICOPEE", "loc" : [ -72.576142, 42.176443 ], "pop" : 31495, "state" : "MA" }
{ "_id" : "01022", "city" : "WESTOVER AFB", "loc" : [ -72.558657, 42.196672 ], "pop" : 1764, "state" : "MA" }
{ "_id" : "01026", "city" : "CUMMINGTON", "loc" : [ -72.905767, 42.435296 ], "pop" : 1484, "state" : "MA" }
{ "_id" : "01027", "city" : "MOUNT TOM", "loc" : [ -72.679921, 42.264319 ], "pop" : 16864, "state" : "MA" }
{ "_id" : "01028", "city" : "EAST LONGMEADOW", "loc" : [ -72.505565, 42.067203 ], "pop" : 13367, "state" : "MA" }
{ "_id" : "01030", "city" : "FEEDING HILLS", "loc" : [ -72.675077, 42.07182 ], "pop" : 11985, "state" : "MA" }
>
```

```
> db.ciudades.find()
{ "_id" : "01001", "city" : "AGAWAM", "loc" : [ -72.622739, 42.070206 ], "pop" : 15338, "state" : "MA" }
{ "_id" : "01002", "city" : "CUSHMAN", "loc" : [ -72.51565, 42.377017 ], "pop" : 36963, "state" : "MA" }
{ "_id" : "01005", "city" : "BARRE", "loc" : [ -72.108354, 42.409698 ], "pop" : 4546, "state" : "MA" }
{ "_id" : "01007", "city" : "BELCHERTOWN", "loc" : [ -72.410953, 42.275103 ], "pop" : 10579, "state" : "MA" }
{ "_id" : "01008", "city" : "BLANDFORD", "loc" : [ -72.936114, 42.182949 ], "pop" : 1240, "state" : "MA" }
{ "_id" : "01010", "city" : "BRIMFIELD", "loc" : [ -72.188455, 42.116543 ], "pop" : 3706, "state" : "MA" }
{ "_id" : "01011", "city" : "CHESTER", "loc" : [ -72.988761, 42.279421 ], "pop" : 1688, "state" : "MA" }
{ "_id" : "01012", "city" : "CHESTERFIELD", "loc" : [ -72.833309, 42.38167 ], "pop" : 177, "state" : "MA" }
{ "_id" : "01013", "city" : "CHICOPEE", "loc" : [ -72.607962, 42.162046 ], "pop" : 23396, "state" : "MA" }
{ "_id" : "01020", "city" : "CHICOPEE", "loc" : [ -72.576142, 42.176443 ], "pop" : 31495, "state" : "MA" }
{ "_id" : "01022", "city" : "WESTOVER AFB", "loc" : [ -72.558657, 42.196672 ], "pop" : 1764, "state" : "MA" }
{ "_id" : "01026", "city" : "CUMMINGTON", "loc" : [ -72.905767, 42.435296 ], "pop" : 1484, "state" : "MA" }
{ "_id" : "01027", "city" : "MOUNT TOM", "loc" : [ -72.679921, 42.264319 ], "pop" : 16864, "state" : "MA" }
{ "_id" : "01028", "city" : "EAST LONGMEADOW", "loc" : [ -72.505565, 42.067203 ], "pop" : 13367, "state" : "MA" }
{ "_id" : "01030", "city" : "FEEDING HILLS", "loc" : [ -72.675077, 42.07182 ], "pop" : 11985, "state" : "MA" }
>
```

NoSQL

MongoDB

CONSULTAS DE LA BASE DE DATOS USANDO EL SHELL DE MONGODB

En ejemplo hemos utilizado algunas ciudades españolas incluyendo su nombre, población, ubicación geográfica y algunos de sus puntos de interés turísticos.

```
{  
  "ciudades": [  
    {  
      "nombre": "Madrid",
```

NoSQL
MongoDB

```

    "poblacion": 3.3,
    "latitud": 40.4168,
    "longitud": -3.7038,
    "atracciones": [
      "Museo del Prado",
      "Parque del Retiro",
      "Plaza Mayor",
      "Palacio Real de Madrid"
    ]
  },
  {
    "nombre": "Barcelona",
    "poblacion": 1.6,
    "latitud": 41.3851,
    "longitud": 2.1734,
    "atracciones": [
      "Basílica de la Sagrada Familia",
      "Parque Güell",
      "La Rambla",
      "Catedral de Barcelona"
    ]
  },
  {
    "nombre": "Sevilla",
    "poblacion": 0.7,
    "latitud": 37.3891,
    "longitud": -5.9845,
    "atracciones": [

```

NoSQL

MongoDB

```
{
  "ciudades": [
    {
      "nombre": "Sevilla",
      "poblacion": 0.7,
      "latitud": 37.3886,
      "longitud": -5.9823,
      "atracciones": [
        "Catedral de Sevilla",
        "La Giralda",
        "Alcázar de Sevilla",
        "Barrio de Santa Cruz"
      ]
    },
    {
      "nombre": "Valencia",
      "poblacion": 0.8,
      "latitud": 39.4699,
      "longitud": -0.3763,
      "atracciones": [
        "Ciudad de las Artes y las Ciencias",
        "Catedral de Valencia",
        "Mercado Central de Valencia",
        "Playa de la Malvarrosa"
      ]
    }
  ]
}
```

1. Mostrar todas las ciudades y sus atracciones turísticas:

```
db.ciudades.find({}, {nombre: 1, atracciones: 1, _id: 0})
```

2. Mostrar todas las ciudades que tienen una población mayor a 1 millón:

NoSQL

MongoDB

`db.ciudades.find({poblacion: {$gt: 1}})`

3. Mostrar todas las ciudades que tienen una población mayor a 1 millón

`db.ciudades.find({latitud: {$gt: 40.4168}}, {nombre: 1, latitud: 1, longitud: 1, _id: 0})`

4. Mostrar todas las ciudades que tienen la palabra "Bar" en su nombre:

`db.ciudades.find({nombre: /Bar/})`

5. Mostrar todas las ciudades que tienen el Parque Guell como atracción turística:

`db.ciudades.find({atracciones: "Parque Güell"})`

CONCLUSIONES

RESUMEN VENTAJAS Y DESVENTAJAS MONGODB

Mongo es una popular base de datos orientada a documentos NoSQL que utiliza documentos NoSQL que utiliza documentos similares a JSON con esquemas dinámicos, lo que ofrece una amplia gama de ventajas sobre las bases de datos relacionales tradicionales.

Una de las principales ventajas de MongoDB es su flexibilidad al tratar con estructuras de datos complejas. A diferencia de las bases de datos relacionales, que requieren un esquema fijo, MongoDB crea dinámicamente colecciones y documentos a medida

NoSQL

MongoDB

que se agregan datos, lo que permite actualizaciones y modificaciones sencillas a medida que cambia la estructura de los datos. Esta flexibilidad hace de MongoDB una opción ideal para manejar aplicaciones de big data donde los datos evolucionan y cambian constantemente..

Otra ventaja de MongoDB es su escalabilidad. A medida que crecen los volúmenes de datos, las bases de datos relacionales tradicionales a menudo luchan por adaptarse a la expansión, lo que genera problemas como velocidades de escritura lentas, problemas de escalabilidad y recuperación de datos lenta. MongoDB, por otro lado, ofrece funciones de fragmentación integradas que le permiten escalar horizontalmente en múltiples instancias de manera rápida y eficiente.

MongoDB también proporciona un mejor rendimiento que las bases de datos relacionales tradicionales. Dado que MongoDB almacena todos los datos en un solo documento, elimina la necesidad de operaciones JOIN complejas, que consumen mucho tiempo y pueden afectar negativamente el rendimiento. Además, MongoDB ofrece capacidades de indexación integradas que pueden ayudar a optimizar el rendimiento de las consultas, haciéndolas más rápidas que las bases de datos tradicionales.

Otro beneficio importante de MongoDB para los desarrolladores es su facilidad de uso. MongoDB es fácil de configurar y el uso de documentos hace que el acceso a los datos sea rápido y directo. Los desarrolladores también pueden usar MongoDB para almacenar cualquier tipo de datos, como texto, imágenes, videos o cualquier otro formato de archivo. Esta flexibilidad permite a los desarrolladores crear aplicaciones que son más robustas y también permite el desarrollo paralelo de varias funciones de una aplicación.

Sin embargo, MongoDB no es perfecto y existen diferencias notables entre MongoDB y las bases de datos relacionales. Una importante es que no admite las garantías de consistencia transaccional que normalmente se esperan de las bases de datos relacionales. El enfoque de MongoDB enfatiza la disponibilidad y el rendimiento, compensando la consistencia para lograr estos objetivos.

Otra diferencia es que MongoDB no tiene un soporte integrado para las operaciones JOIN que se encuentran en las bases de datos relacionales tradicionales. Si bien esto es intencional, obliga a los desarrolladores a diseñar sus patrones de acceso a los datos en torno a las restricciones de la estructura del documento de MongoDB.

NoSQL

MongoDB

FUTURO DE LAS BASES DE DATOS NoSQL y MongoDB

Las bases de datos NoSQL son un conjunto de tecnologías que se han desarrollado en los últimos años con el objetivo de resolver algunos de los problemas que presentan las bases de datos relacionales. Estas bases de datos se caracterizan por no utilizar el modelo relacional y por ser altamente escalables y flexibles, lo que las hace ideales para aplicaciones web y móviles que requieren una gran cantidad de datos y una alta disponibilidad. Entre las bases de datos NoSQL más populares se encuentra MongoDB, una base de datos documental que ha ganado gran popularidad en los últimos años y que se espera tenga un gran futuro en el mundo de las bases de datos.

El futuro de las bases de datos NoSQL es muy prometedor debido a que las aplicaciones web y móviles están creciendo a un ritmo acelerado, y estas bases de datos se adaptan perfectamente a este entorno. Las bases de datos NoSQL pueden manejar grandes cantidades de datos de manera eficiente y rápida, lo que las hace ideales para aplicaciones que requieren de un alto rendimiento y una gran capacidad de almacenamiento. Además, estas bases de datos son altamente escalables, lo que significa que se pueden añadir nodos adicionales a medida que la carga de trabajo aumenta, sin necesidad de detener el servicio. Esto las hace ideales para aplicaciones que deben soportar un gran número de usuarios y que deben estar disponibles las 24 horas del día, los 7 días de la semana.

MongoDB es una de las bases de datos NoSQL más populares y se espera que tenga un gran futuro debido a sus características y su facilidad de uso. MongoDB es una base de datos documental, lo que significa que almacena la información en forma de documentos, en lugar de utilizar el modelo relacional de tablas utilizado por las bases de datos relacionales. Esto hace que sea muy fácil trabajar con la información, ya que los documentos pueden contener toda la información relacionada en un solo lugar. Además, MongoDB tiene una sintaxis similar a la de JavaScript, lo que la hace muy fácil de aprender para los desarrolladores que trabajan con aplicaciones web y móviles.

Otra de las ventajas de MongoDB es su alta escalabilidad. MongoDB se puede escalar horizontalmente, lo que significa que se pueden añadir nodos adicionales a medida que la carga de trabajo aumenta. Esto hace que sea muy fácil escalar una aplicación a medida que crece el número de usuarios o la cantidad de información que se debe almacenar. Además, MongoDB es altamente disponible, lo que significa que puede estar disponible las 24 horas del día, los 7 días de la semana, lo que la hace ideal para aplicaciones críticas que no pueden permitirse estar fuera de servicio.

NoSQL

MongoDB

El futuro de MongoDB se ve muy prometedor debido a la creciente demanda de aplicaciones web y móviles. La popularidad de las aplicaciones móviles ha crecido exponencialmente en los últimos años y se espera que continúe creciendo en el futuro. Esto significa que se necesitarán bases de datos que puedan manejar grandes cantidades de información y que puedan estar disponibles las 24 horas del día, los 7 días de la semana. MongoDB es una de las bases de datos NoSQL que mejor se adapta a este entorno, por lo que se espera que continúe ganando popularidad.

Además, MongoDB está en constante evolución, lo que significa que se están añadiendo nuevas características y funcionalidades a medida que se descubren nuevas necesidades. Esto la hace muy flexible y adaptable, lo que la convierte en una opción muy atractiva para los desarrolladores que necesitan una base de datos que pueda crecer con ellos.

Una de las últimas novedades de MongoDB es Atlas, su servicio de base de datos en la nube. Atlas ofrece una serie de características muy interesantes para los desarrolladores que necesitan una base de datos escalable y disponible en la nube. Atlas ofrece la posibilidad de escalar la base de datos en tiempo real, lo que significa que se pueden añadir nodos adicionales a medida que la carga de trabajo aumenta. Además, Atlas ofrece la posibilidad de configurar la base de datos en diferentes regiones geográficas, lo que la hace ideal para aplicaciones que deben estar disponibles en diferentes partes del mundo.

Otra de las características interesantes de Atlas es la integración con otros servicios de la nube, como Amazon Web Services o Google Cloud Platform. Esto significa que se puede integrar fácilmente la base de datos con otros servicios de la nube, como almacenamiento de archivos o procesamiento de datos, lo que la hace muy versátil.

En resumen, el futuro de las bases de datos NoSQL y MongoDB es muy prometedor debido a la creciente demanda de aplicaciones web y móviles que requieren de una base de datos escalable, disponible y flexible. MongoDB es una de las bases de datos NoSQL más populares y se espera que continúe ganando popularidad debido a sus características y facilidad de uso. Además, la evolución constante de MongoDB y la introducción de servicios como Atlas la hacen muy atractiva para los desarrolladores que necesitan una base de datos que pueda crecer con ellos y adaptarse a sus necesidades. En conclusión, MongoDB es una excelente opción para aquellos que buscan una base de datos NoSQL escalable, disponible y flexible, que les permita adaptarse a las necesidades del futuro.

NoSQL

MongoDB

REFERENCIAS

<https://en.wikipedia.org/wiki/MongoDB>

<https://www.digitalocean.com/community/tutorials/how-to-import-and-export-a-mongodb-database-on-ubuntu-20-04-es>