

INGENIERÍA MECATRÓNICA



DI_CERO

DIEGO CERVANTES RODRÍGUEZ

PROGRAMACIÓN: DESARROLLO BACKEND

SQL

Bases de Datos No Relacionales
Basadas en Documentos: MongoDB

Contenido

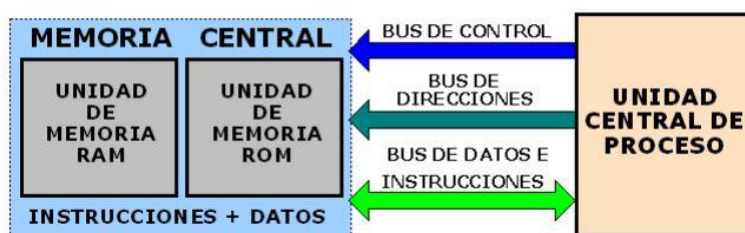
Introducción a las Bases de Datos	2
Tipos de Bases de Datos.....	2
Representación de las Bases de Datos: Nomenclatura de Chen	3
Bases de Datos No Relacionales	3
Jerarquía de Datos de las Bases de Datos No Relacionales basadas en Documentos.....	5
Tipos de Escalamiento: Horizontal y Vertical	7
Réplicas o Sharding	8
Instalación de MongoDB	8
Referencias.....	9



Introducción a las Bases de Datos

Las **bases de datos** ayudan a complementar la **arquitectura de Von Neumann**, que es utilizada en ordenadores, que a diferencia de la **arquitectura Harvard** usada en microcontroladores, utiliza una sola memoria para realizar sus funciones y guardar sus datos. La necesidad de extender la capacidad de la memoria central es la de conservar los datos más allá de la memoria RAM o ROM, ya que en la **arquitectura Von Neumann** si se contempla el procesamiento de datos, pero no el almacenamiento de datos persistentes, por lo que es de suma importancia la utilización de las **databases (DB)**.

ARQUITECTURA VON NEUMANN



Para resolver esta situación, donde se busca que de una forma fácil se puedan **guardar y extraer datos** de información, se obtuvieron dos soluciones:

- **Bases de datos basadas en archivos:** Este **método de almacenamiento de datos persistentes** consiste en **guardar información en un archivo de texto plano**, hojas de cálculo, etc. usualmente separados por comas o de alguna otra forma ordenada.
- **Bases de datos basadas en documentos:** En este tipo de **base de datos**, la unidad básica de almacenamiento es el **documento**, que puede contener datos en forma de texto, números, listas, objetos JSON (JavaScript Object Notation) y a veces incluso otros documentos anidados.

Tipos de Bases de Datos

Los diferentes tipos de bases de datos existentes son los siguientes:

- **Relacionales o RDB:** Son bases de datos basadas en documentos y están gobernadas por las 12 reglas de Edgar Codd, que dan como resultado el **álgebra relacional**, a través de las cuales se indican las reglas con las que los **datos** de las **RDB (Relational Databases)** se pueden relacionar.
 - **Privadas:** Microsoft SQL Server, **Oracle**, etc.
 - **Open Source:** **PostgreSQL**, MySQL, MariaDB, etc.

Ejemplos de bases de datos relacionales



- **No relacionales o NRDB:** Hay varios tipos de **bases de datos no relacionales**, todas ellas pueden ser muy distintas unas de otras, pero se engloban dentro de la misma categoría de **NRDB (Non Relational Databases)** porque utilizan lenguajes NoSQL (Not Only SQL) para sus consultas. Los diferentes tipos de bases de datos no relacionales a grandes rasgos son:
 - Basadas en Clave-Valor, en Documentos, en Grafos, en Memoria, Optimizadas para Búsquedas, etc. Algunos ejemplos de ellas son:
 - Memcached, Cassandra (Facebook), DynamoDB, ElasticSearch, BigQuery, Neo4j (GraphQL), **MongoDB**, Firestore (Firebase).

Bases de datos no relacionales



Representación de las Bases de Datos: Nomenclatura de Chen

- **Entidad:** Se refiere a una **tabla** que almacena datos sobre un tipo de objeto o elemento del mundo real.
 - Cada **fila** en la **tabla** representa una **instancia individual** de esa **entidad**.
 - Cada **columna** en la **tabla** representa un **atributo o característica** de esa **entidad**.
- **Atributo:** Son las **columnas de una tabla** que representan las **características o propiedades** de la **entidad** que está siendo modelada, todas ellas tienen un **nombre y tipo de dato asociado**.
- **Registro:** Representa una **fila perteneciente a una tabla**. También es conocido como "**tupla**" y **contiene los valores** de los **atributos** correspondientes a una **instancia** específica de una **entidad**.

Bases de Datos No Relacionales

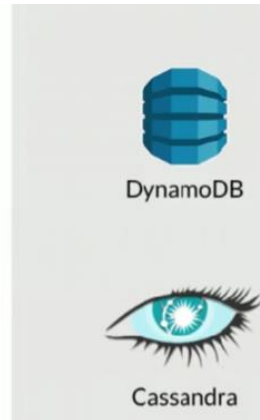
Como ya se había mencionado previamente, las **bases de datos no relacionales** o **NRDB (Non Relational Data Bases)** **no se conforman de un solo tipo de bases de datos, sino de varios**, y aunque puedan ser muy distintas unas de otras, todas se engloban dentro de la misma categoría de **base de datos no relacional**, ya que utilizan lenguajes NoSQL (Not Only SQL). Los diferentes tipos de **databases no relacionales** son:

- **NRDB Basadas en Clave-Valor:** Estas **bases de datos no relacionales** están diseñadas para almacenar y recuperar **información de manera rápida** mediante el uso de una **clave**. Cada **clave** está asociada a un **valor**, que puede ser un **dato simple**, un **objeto** o un **conjunto de datos**. Las consultas se realizan utilizando estas **claves únicas** y su **principal característica es su alta velocidad y eficiencia en operaciones de lectura y escritura de datos**.

- Algunos ejemplos de estas bases de datos no relacionales son **DynamoDB de AWS**, Cassandra de Facebook y Redis.

Clave - valor

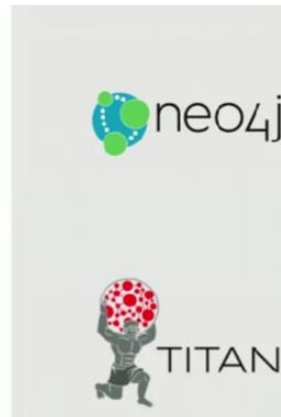
Son ideales para almacenar y extraer datos con una clave única. Manejan los diccionarios de manera excepcional.



- **NRDB Basadas en Grafos:** Los grafos se componen de **nodos** o **entidades (tablas)** que tienen **relaciones** muy complejas unas con otras y generalmente se **conectan** entre sí, creando redes de **datos**, se usan para crear **inteligencias artificiales (redes neuronales)** o **redes sociales**.
 - Algunos ejemplos de estas **bases de datos no relacionales** son Titan y Neo4j.

Basadas en grafos

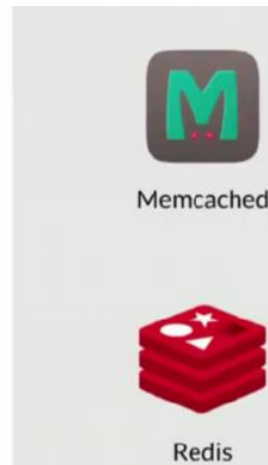
Basadas en teoría de grafos sirven para entidades que se encuentran interconectadas por múltiples relaciones. Ideales para almacenar relaciones complejas.



- **NRDB Basadas en Memoria:** Este tipo de **base de datos** son sumamente rápidas, pero tienen la gran desventaja de que son volátiles, osea que **su memoria no es duradera**.

En memoria

Pueden ser de estructura variada, pero su ventaja radica en la velocidad, ya que al vivir en memoria la extracción de datos es casi inmediata.



- **NRDB Optimizadas para Búsquedas:** Este tipo de **base de datos** pueden ejecutar Queries muy complejas de forma muy rápida y a grandes **repositorios de datos históricos** que almacenan un gran volumen de información, son muy utilizadas en aplicaciones de **business intelligence** y **machine learning**.
 - Algunos ejemplos de estas bases de datos no relacionales son **BigQuery de Google** y Elasticsearch.

Optimizadas para búsqueda

Pueden ser de diversas estructuras, su ventaja radica en que se pueden hacer queries y búsquedas complejas de manera sencilla.



BigQuery



Elasticsearch

- **NRDB Basadas en Documentos:** En estas **bases de datos no relacionales** se empareja cada clave con una estructura de datos llamada **Documento** que es mayormente utilizado para referirnos a **archivos de tipo JSON (JavaScript Object Notation) o XML**.
 - Algunos ejemplos de estas bases de datos no relacionales son **MongoDB**, **Firestore de Google**, Couchbase, etc.

Basados en documentos

Son una implementación de clave valor que varía en la forma semiestructurada en que se trata la información. Ideal para almacenar datos JSON y XML.



MongoDB



Firestore

Jerarquía de Datos de las Bases de Datos No Relacionales basadas en Documentos

En las bases de **datos no relacionales basadas en documentos**, en vez de que se cuente con **tablas (entidades)**, **atributos (columnas)**, **relaciones (conexiones)**, etc. su estructura se basa en **colecciones de**

datos, que son el equivalente a las **entidades**, las cuales clasifican los distintos **documentos** que contienen estructuras JSON que asocian cada **valor** con una **clave**.



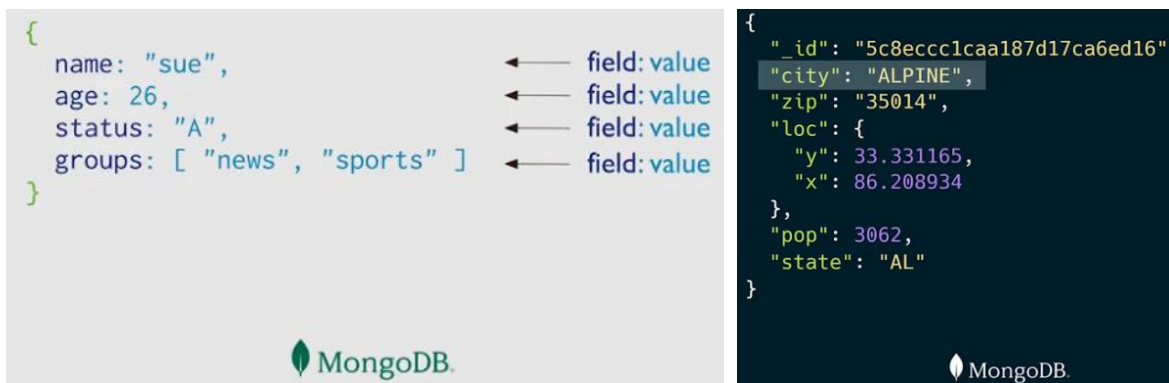
Cuando trabajamos con bases de datos basadas en documentos como **MongoDB** o **Firestore**, cambiaremos el concepto de las **tablas** por las **colecciones** y las **tuplas (filas)** por los **documentos**.

- **Entidad o Tabla** → **Colección**.
- **Tuplas o Filas** → **Documento**, que almacena sus **datos** en forma de **clave:valor**, donde cada **clave** corresponde a las **columnas** de la **tabla** y su valor a cada **instancia** específica.

Además, en el contexto de las **colecciones**, identificamos dos categorías principales:

- Las **"Top level collections"** o **Colecciones de nivel superior**.
- Y las **"subcollections"** o **subcolecciones**, que se incorporan dentro de otra **colección**, así como en las estructuras JSON, podemos meter un JSON dentro de otro, de igual forma se puede introducir un **documento** dentro de otro o una **colección** dentro de otra.

No existe una regla estricta para determinar si la **colección** debe ser de **nivel superior** o una **subcolección** al crear una **base de datos basada en documentos**; más bien, esta decisión depende del caso de uso específico.



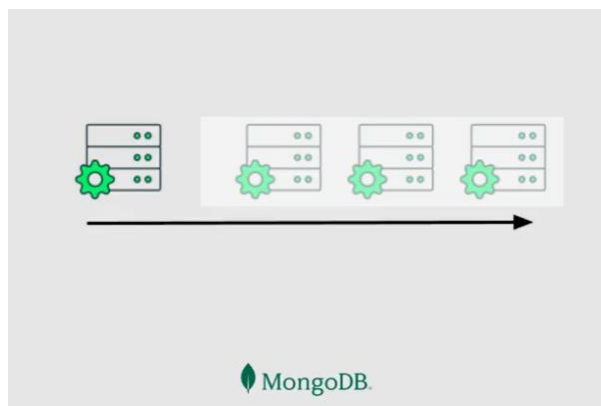
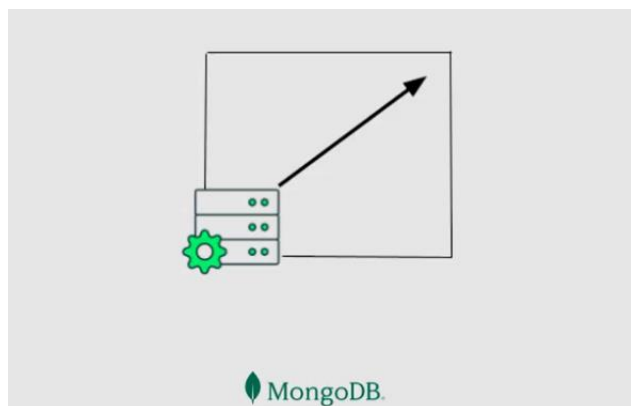
Una consideración clave al diseñar la **database no relacional** es anticipar cómo se **extraerán los datos**. En el contexto de una aplicación, es útil pensar en términos de las vistas que se mostrarán en un momento específico. En otras palabras, al estructurar la **DB**, **debemos asegurarnos de que refleje o contenga, al menos, todos los datos necesarios para satisfacer los requisitos visuales de nuestra aplicación** en un momento dado.

Esta regla se aplica salvo algunas excepciones, como cuando se tiene una entidad que necesita existir y modificarse de manera constante e independiente de otras colecciones.

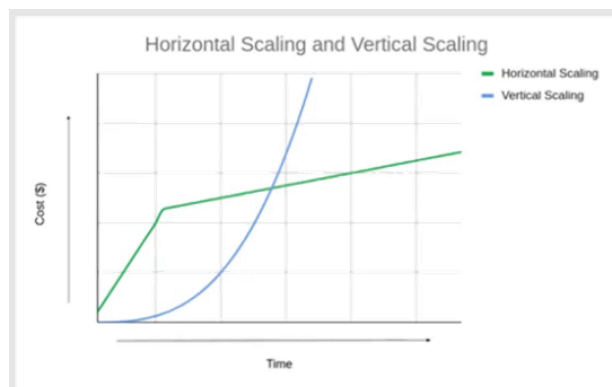
Tipos de Escalamiento: Horizontal y Vertical

El **escalamiento** se refiere a la habilidad de nuestras **bases de datos** o sus **servidores (ordenadores donde se encuentran almacenadas las DB)** de adaptarse a las necesidades que requiera la aplicación o el usuario. Existen dos tipos de escalamiento:

- **Escalamiento vertical:** Este escalamiento se refiere a **aumentar las capacidades del servidor donde se encuentra montada la base de datos**, incrementando los límites de sus recursos como su procesador, memoria RAM, espacio de almacenamiento, etc.
- **Escalamiento horizontal:** Este escalamiento se refiere a crear varias **réplicas (o nodos)** de las **bases de datos** en **diferentes servidores**, lo cual hace que, **con recursos limitados de los servidores, nos aseguremos que la DB posea alta disponibilidad de datos, copias de seguridad, un sistema en conjunto que responda de forma simultánea, etc.** sin necesidad de aumentar los recursos de las máquinas.



Las **bases de datos no relacionales** son más propensas a permitir el **escalamiento horizontal** y aunque el **escalamiento vertical** en un inicio es más fácil implementarlo, a la larga es más costoso, mientras que el escalamiento horizontal es más costoso en un inicio, pero si aumentan los requerimientos del sistema, su costo se mantiene.



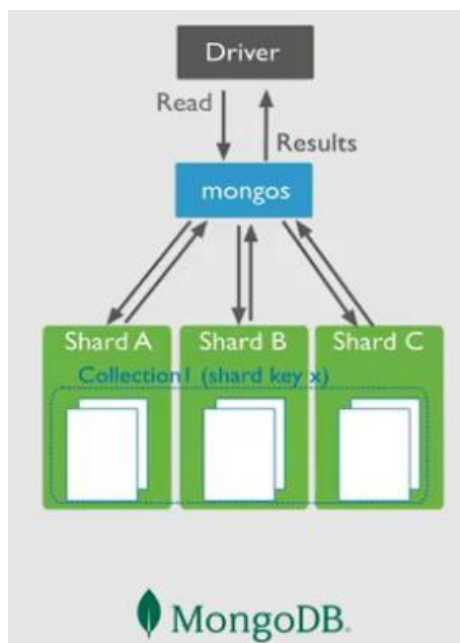
Réplicas o Sharding

Réplica o Shard se refiere al proceso de **copiar y mantener actualizada una copia de la base de datos en un servidor diferente**.

Esta técnica es utilizada para evitar problemas de **entrada y salida de datos** en los sistemas operativos, ya que, si nuestra aplicación realiza **peticiones de lectura y escritura de datos** de forma exponencial; como **no se puede leer y escribir datos** en una **tabla** al mismo tiempo, esta se bloquea durante dicho proceso, pero cuando se tienen manejo de datos múltiples, existen límites electrónicos o físicos, que restringen la capacidad de procesamiento del CPU, por lo que una **petición podría tardar minutos en ejecutarse** y esto es catastrófico.

Por eso es tan importante el uso de **réplicas**, donde al menos se cuenta con **dos servidores distintos (aunque pueden ser más de 2)**, uno como **master** y el otro es la **réplica**:

- Se tiene un **servidor** con una **database principal**, donde solo se realizan las **entradas o modificaciones de datos**.
- Y otro **servidor** con una **base de datos secundaria** (que es la **réplica**), donde solo se realiza la **lectura de datos**.



Instalación de MongoDB



Referencias

Platzi, Israel Vázquez, “Curso de Fundamentos de Bases de Datos”, 2018 [Online], Available: [https://platzi.com/new-home/clases/1566-bd/19781-bienvenida-conceptos-basicos-y-contexto-historico-/-/](https://platzi.com/new-home/clases/1566-bd/19781-bienvenida-conceptos-basicos-y-contexto-historico-/)

Platzi, Nicolás Molina, “Curso de Introducción a MongoDB”, 2023 [Online], Available: <https://platzi.com/cursos/mongodb/>

