

Bases de Datos NoSQL














DEV.F
DESARROLLAMOS(PERSONAS);

dev

Las Bases de Datos NoSQL (“Not Only SQL”) pertenecen al modelo no relacional. Las principales características y ventajas de este tipo son:

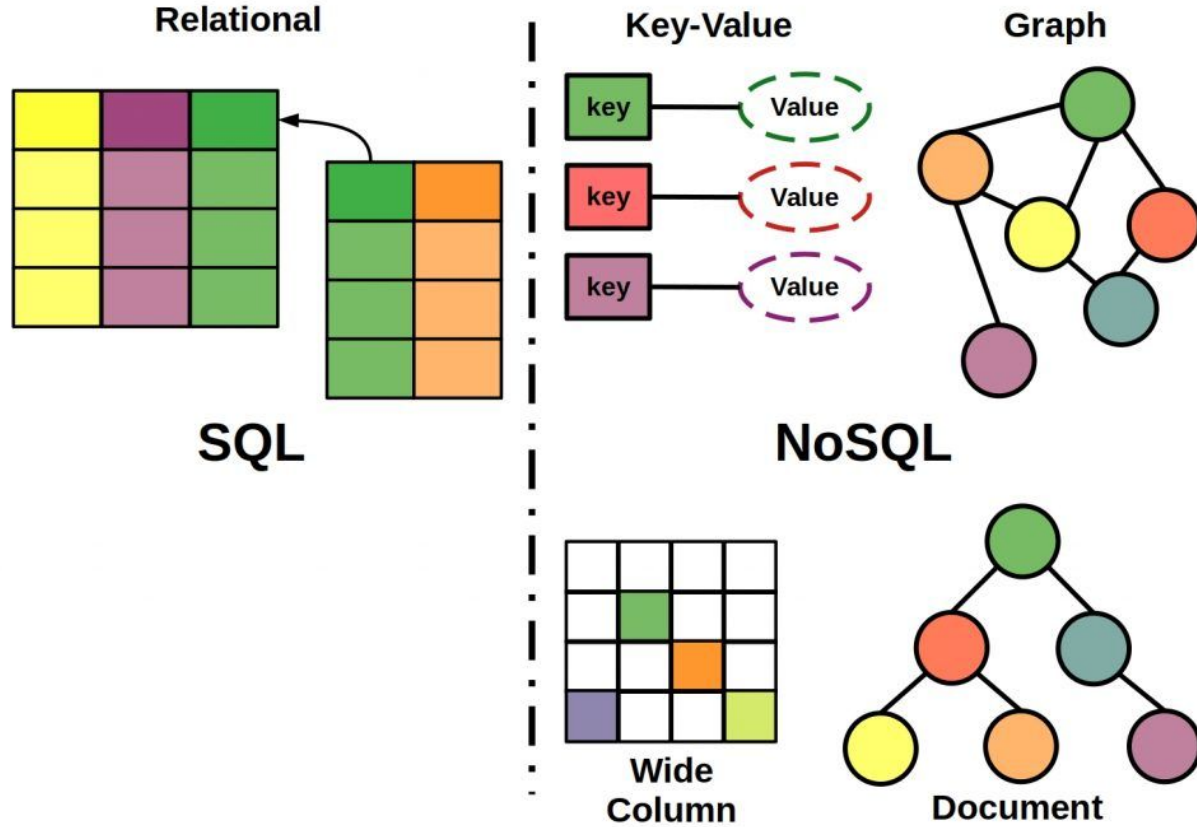
- SQL no es el lenguaje de consulta/modificación de datos principal, aunque sí lo soportan, de ahí el nombre **No Solo SQL (No Solo Relacional)**.
- Los datos no tienen que almacenarse en tablas.
- Generalmente, su arquitectura es distribuida almacenando la información en más de una máquina del sistema. Por lo tanto, los sistemas que las soportan tienen una **mayor escalabilidad horizontal** (a mayor número de nodos mayor rendimiento) y también mayor tolerancia ante fallos en los distintos nodos.

Tipos de Base de Datos NoSQL

Document Database	Graph Databases
 Couchbase  MarkLogic  mongoDB	 Neo4j  InfiniteGraph The Distributed Graph Database
Wide Column Stores	Key-Value Databases
 redis  amazon DynamoDB  AEROSPIKE  riak	 accumulo  HYPERTABLE INC  Cassandra  APACHE HBASE Amazon SimpleDB

SQL vs NoSQL

	SQL	NoSQL
Database Type	Relational Databases	Non-relational Databases / Distributed Databases
Structure	Table-based	<ul style="list-style-type: none">• Key-value pairs• Document-based• Graph databases• Wide-column stores
Scalability	Designed for scaling up vertically by upgrading one expensive custom-built hardware	Designed for scaling out horizontally by using shards to distribute load across multiple commodity (inexpensive) hardware
Strength	<ul style="list-style-type: none">• Great for highly structured data and don't anticipate changes to the database structure• Working with complex queries and reports	<ul style="list-style-type: none">• Pairs well with fast paced, agile development teams• Data consistency and integrity is not top priority• Expecting high transaction load



Intro a mongoDB®

DEV.F
DESARROLLAMOS(PERSONAS);

dev

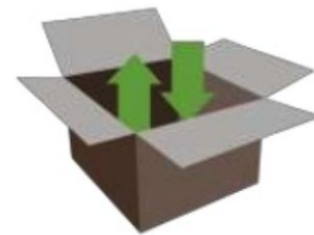
MongoDB: El lider Bases de Datos NoSQL



- Lleno de herramientas
- Alto Performance
- Escalable

```
{  
  name: "John Smith",  
  pfxs: ["Dr.", "Mr."],  
  address: "10 3rd St.",  
  phones: [  
    { number: "555-1212",  
      type: "land" },  
    { number: "444-1212",  
      type: "mobile" }  
  ]  
}
```

Document Data Model



Open - Source



500+ employees



2000+ customers



Offices in NY & Palo Alto and
across EMEA, and APAC

NEA

SEQUOIA CAPITAL
THE ENTREPRENEURS BEHIND THE ENTREPRENEURS

intel
Capital

union
square
ventures

T.Rowe Price
INVEST WITH CONFIDENCE

Fidelity
INVESTMENTS

FLYBRIDGE
CAPITAL PARTNERS

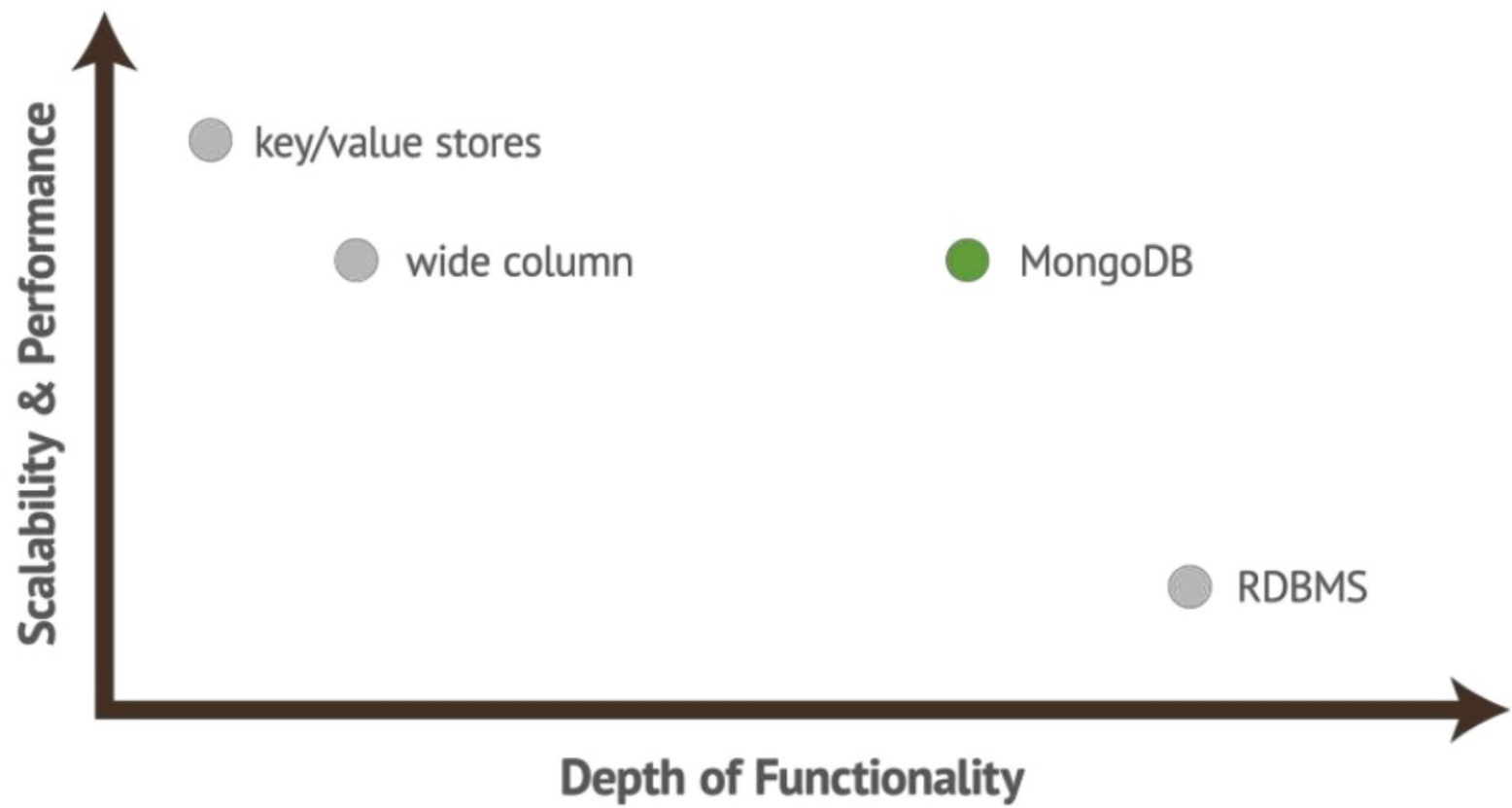
salesforce

redhat

ALTIMETER

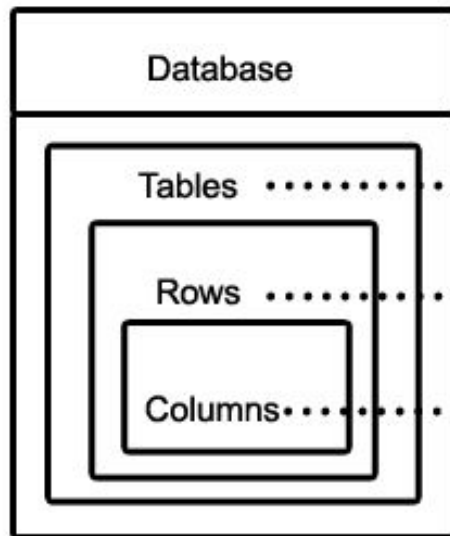
IQT
INTEL

Over **\$311 million** in funding

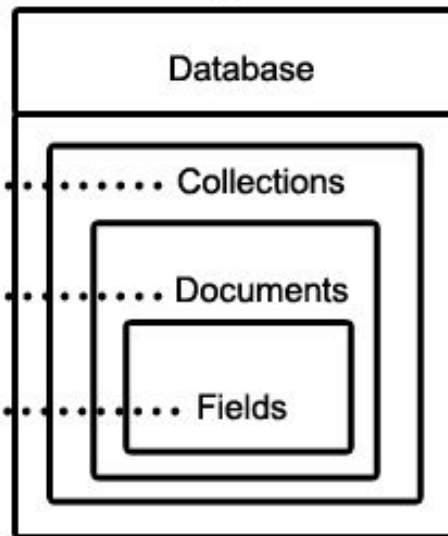


Tabular/RDMBS	Document/MongoDB
Entity	Entity
Relationship	Relationship
Database	Database
Table	Collection
Attribute/Column	Field
Record/Row	Document
Join	Embed or Link

RDBMS



MongoDB



Tables

Rows

Columns.....

..... Collections

..... Documents

..... Fields

```
{
  _id: "123",
  title: "MongoDB: The Definitive Guide",
  authors: [
    { _id: "kchodorow", name: "Kristina Chodorow" },
    { _id: "mdirold", name: "Mike Dirolf" }
  ],
  published_date: ISODate("2010-09-24"),
  pages: 216,
  language: "English",
  thumbnail: BinData(0,"AREhMQ=="),
  publisher: {
    name: "O'Reilly Media",
    founded: 1980,
    locations: ["CA", "NY" ]
  }
}
```

MongoDB es Schemaless

- No tener schema significa que **no hay una estructura fija que le dé forma a los documentos que almacenamos**. Estos pueden tener diferentes esquemas y pueden variar sus propiedades.
- Si bien esto es una gran ventaja, tampoco hay que abusar de ella porque nuestra base deja de ser eficiente al momento de realizar búsquedas.















Un escenario NoSQL

Resuelven necesidades habituales, tanto aplicaciones web, apps móviles o en el Internet de las Cosas. Pero no se puede decir que se adapten a todo tipo de proyectos.

En general, podemos encontrar de utilidad una base de datos NoSQL cuando nuestras necesidades son las siguientes, conocidas como las “3V”.

- Velocidad
- Volumen
- Variabilidad

Ejemplos de uso de MongoDB

Big Data	Product & Asset Catalogs	Security & Fraud	Internet of Things	Database-as-a-Service	Complex Data Management
   	   	  Top Investment and Retail Banks Intelligence Agencies	Top Global Shipping Company   Top Industrial Equipment Manufacturer	  Top Media Company Top Investment and Retail Banks	Cushman & Wakefield Top Investment and Retail Banks
Mobile Apps	Customer Data Management	Single View	Social & Collaboration	Content Management	Embedded / ISV
   	   	   	   	   	 

Intro a Mongoose

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Objetivo: El objetivo de esta clase es que los alumnos entiendan cómo es la estructura para conectar un servidor de Node.js usando express con una base de datos de Mongo.

Artefacto: Crear un archivo de Node.js en el que se hará la conexión con la base de datos de Mongo Atlas que se creó durante la sesión pasada.

The logo consists of the text 'DEV.F.' in a bold, white, sans-serif font. The 'F' is stylized with a grid of small squares at its right end. The logo is centered within a dark blue diamond shape.

DEV.F.

Ejercicios

Algunos casos sencillos para practicar

Ejercicio 1: Películas

Un coleccionista de películas necesita organizar su colección, por lo que decide tener una app para hacerlo, necesita guardar la información básica de las películas como, nombre, género, actores y descripción.

Tu eres el encargado de diseñar la base de datos para esta app.

Ejercicio 2: Universidad

Una universidad requiere llevar un registro de todos los exalumnos que estuvieron en la universidad.

Para ello requieren conocer el empleo con el que cuentan los egresados, lugar donde trabajan los egresados, así como su información básica como estudiantes (facultad y carrera donde estuvieron).

Ejercicio 3: Veterinaria

Una veterinaria, requiere un sistema que le ayude a controlar la entrada de las mascotas a la veterinaria.

La veterinaria requiere almacenar el nombre de la mascota, el tipo de mascota, y el dueño de la mascota.

Ejercicio 4: Aeropuerto

Un aeropuerto busca controlar los vuelos que llegan al lugar, desea conocer los vuelos que existen, a qué aerolínea pertenecen, las características del avión y el lugar de procedencia.

Ayuda al aeropuerto a solucionar su problema.