



# BD NoSQL

4 Grandes grupos



# NoSQL - NOSQL (No SQL / Not Only SQL)

Some characteristics are common amongst these databases, but none are definitional:

- Not using the relational model (nor the SQL language)
- Open source
- Designed to run on large clusters
- Based on the needs of 21st century web properties
- No schema, allowing fields to be added to any record without controls



# Polyglot Persistence

- Polyglot persistence is about using different data storage technologies to handle varying data storage needs.
- Polyglot persistence can apply across an enterprise or within a single application.
- Encapsulating data access into services reduces the impact of data storage choices on other parts of a system.
- Adding more data storage technologies increases complexity in programming and operations, so the advantages of a good data storage fit need to be weighed against this complexity.

## Speculative Retailers Web Application

User sessions

Redis

Financial Data

RDBMS

Shopping Cart

Riak

Recommendations

Neo4J

Product Catalog

MongoDB

Reporting

RDBMS

Analytics

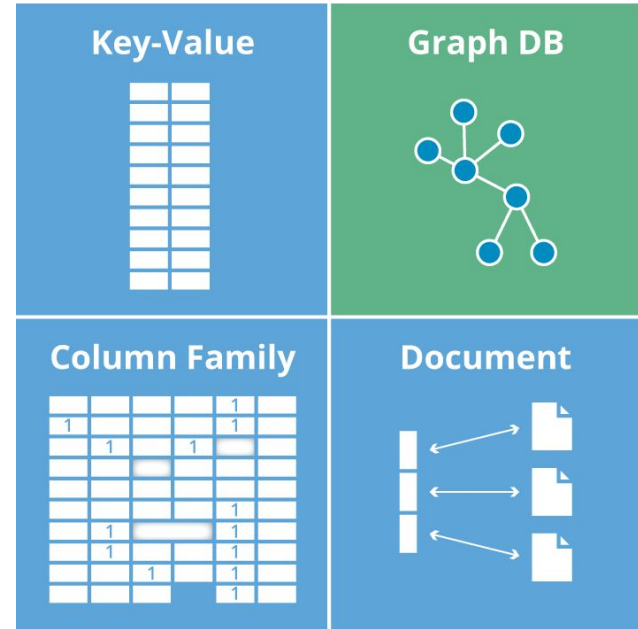
Cassandra

User activity logs

Cassandra

# Grandes grupos:

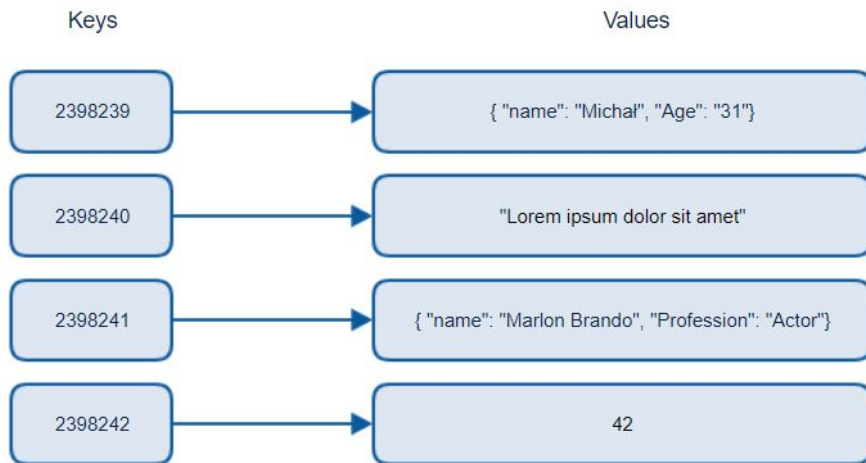
- Key-value
- Document Oriented/Store
- Columnar databases
- Graph



# Key-Value

Las Bases de datos Key-Value guardan la información como pares clave-valor, en estos pares se debe definir al mismo tiempo la clave y el valor al realizar un “insert” o “update”.

Dada su estructura este tipo de BD no tiene esquema, permitiendo agregar para cualquier clave cualquier tipo de dato.





## Key Value

Dado que en la BD se puede guardar cualquier tipo de dato, cualquier restricción de integridad deberá chequearse en la aplicación que la use.

En este tipo de BD no puede existir dos clave iguales. En caso de que suceda la BD podrá indicar un error o pisar el dato original.

```
> set materia 11078
```

```
OK
```

```
> set materia "BD2"
```

```
OK
```

```
> get materia
```

```
"BD2"
```

Ejemplo en BD Redis



## Key Value

El acceso a la información siempre se realiza mediante la clave.

La obtención del valor dada la clave tiene una complejidad de  $O(1)$ .

Es fundamental un buen algoritmo de hashing para la generación de las claves.







# Key value

Usos:

- Base de datos embebida o en memoria
- Manejo de gran cantidad de datos con poca complejidad
- Necesidad de respuesta extremadamente rápida
- Rápida escalabilidad



# Vamos a ver Redis en particular

Redis es una completa Base de Datos key-value que soporta varias estructuras de datos.

Funciona en **memoria** y también posee varios mecanismos de persistencia a disco. Soporta estructuras como String, numeros, listas, sets, hashes, geoespaciales, etc. Cuenta con replicación, manejo de transacciones, Lua scripting y un conjunto de módulos que permiten expandir su funcionalidad.

<https://redis.io/>

<https://try.redis.io/>



# Document Oriented

Permiten la insercion, obtención y manipulación de información en formato semi estructurado llamado **Documento**.

Basicamente un documento es un conjunto de claves-valor, las cuales a su vez, pueden contener otros documentos.



**Document data model**



# Document Oriented

Usualmente el documento está en alguno de los siguientes formatos:

- XML
- JSON
- YAML
- XSD

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

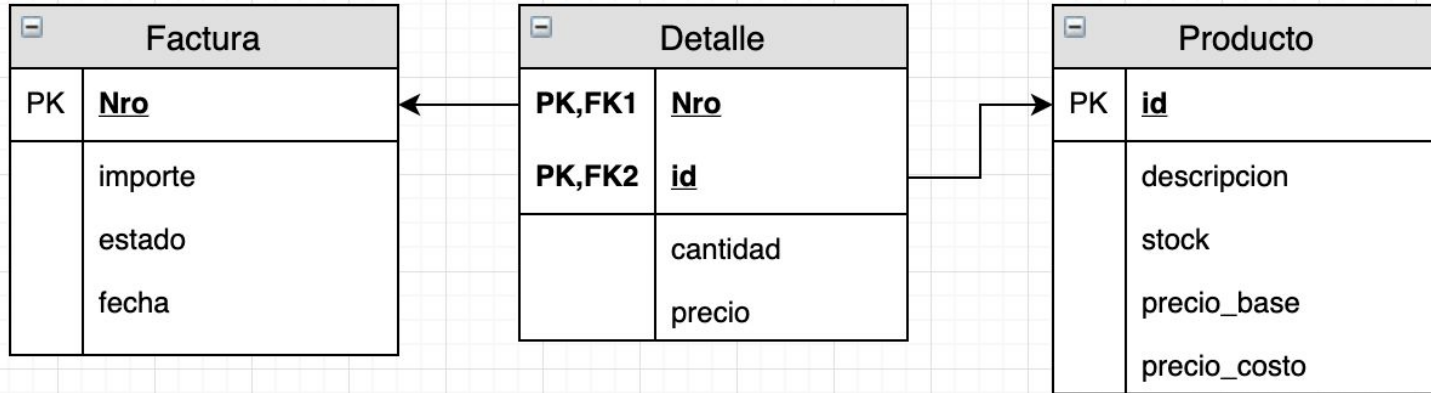
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

Ejemplo de XSD

# Document Oriented - Ejemplo factura

Esta es una posible representación del ejemplo de factura en una BD relacional



# Document Oriented -- XML - JSON - YAML

```
<?xml version="1.0" encoding="UTF-8" ?>
<factura>
  <nro>123</nro>
  <importe>100</importe>
  <estado>iniciada</estado>
  <fecha>10/03/2019</fecha>
  <detalles>
    <detalle>
      <cantidad>2</cantidad>
      <precio>35.5</precio>
      <subtotal>71</subtotal>
      <producto>
        <id>456</id>
        <descripcion>algo</descripcion>
      </producto>
    </detalle>
    <detalle>
      <precio>29</precio>
      <cantidad>1</cantidad>
      <producto>algo diferente</producto>
    </detalle>
  </detalles>
</factura>
```

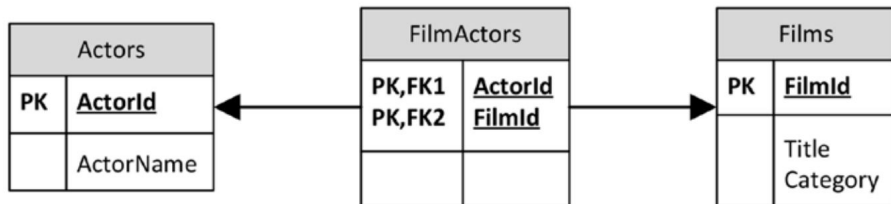
```
{
  "nro":123,
  "importe":100,
  "estado":"iniciada",
  "fecha":"10/03/2019",
  "detalles":[
    {
      "cantidad":2,
      "precio":35.5,
      "subtotal":71,
      "producto":{
        "id":456,
        "descripcion":"algo"
      }
    },
    {
      "precio":29,
      "cantidad":1,
      "producto":"algo diferente"
    }
  ]
}
```

```
---
nro: 123
importe: 100
estado: iniciada
fecha: 10/03/2019
detalles:
  - cantidad: 2
    precio: 35.5
    subtotal: 71
    producto:
      id: 456
      descripcion: algo
  - precio: 29
    cantidad: 1
    producto: algo diferente
```

# Document Oriented

No tienen un esquema, esto significa que los documentos y sus 'campos' pueden variar de documento a documento tanto en tipo de dato como estructura.

Esto permite gran flexibilidad al programador, rápida respuesta a cambios y un desarrollo orientado a prototipos.



```
{ "_id": 115, "Title": "CAMPUS REMEMBER",
  "Category": "Action",
  "Actors" :
    [
      { "actorId": 8, "Name": "MATTHEW JOHANSSON" },
      { "actorId": 45, "Name": "REESE KILMER" },
      { "actorId": 168, "Name": "WILL WILSON" }
    ]
}
```

Actor document



# Document Oriented

Se alienta hacer un diseño desnormalizado donde los documentos contengan otros documentos embebidos, esto es así ya que en una BD orientada a documento se intenta no realizar ningún tipo de 'Join', incrementando significativamente la velocidad de consultas y recuperación de datos.

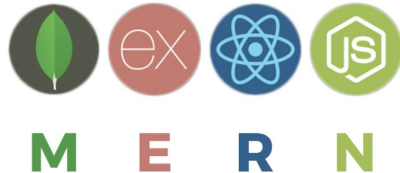
La Base de datos asigna un Identificador a cada documento, aunque también ofrecen potentes lenguajes de consulta para resolución de complejas queries.



# Document Oriented

Si se utiliza formato JSON en la Base de Datos y javascript en el servidor, no se necesario hacer ningún tipo de transformación de los objetos, resultando en una persistencia totalmente transparente.

Esto hace que las BD Documentales están tomando mucho auge en el desarrollo web.





# Document Oriented

A relational database is like a garage that forces you to take your car apart and store the pieces in little drawers

- Object-oriented data community, mid-1990s

An Object database is like a closet which requires that you hang up your suit with tie, underwear, belt, socks and shoes all attached.

—David Ensor, same period

# Columnar Database

La mayor diferencia entre las BD columnares y el modelo relacional es en cómo se guarda la información.

SSN	Name	Age	Addr	City	St
101259797	SMITH	88	899 FIRST ST	JUNO	AL
892375862	CHIN	37	16137 MAIN ST	POMONA	CA
318370701	HANDU	12	42 JUNE ST	CHICAGO	IL

Muchas de las mismas estructuran la información como en el modelo relacional y hasta soportan SQL.



# Columnar Database

Mientras que en una BD Relacional cada TUPLA se escribe una a continuación de la otra en el filesystem, el modelo columnar escribe cada columna en un archivo separado.

SSN	Name	Age	Addr	City	St
101259797	SMITH	88	899 FIRST ST	JUNO	AL
892375862	CHIN	37	16137 MAIN ST	POMONA	CA
318370701	HANDU	12	42 JUNE ST	CHICAGO	IL

Estos archivos después son unidos para recuperar lo que en el modelo relacional es una tupla.

101259797   892375862   318370701   468248180   378568310   231346875   317346551   770336528   277332171   455124598   735885647   387586301
---

Block 1



# Columnar Database

Al mantener cada archivo con un mismo dominio de datos se puede utilizar algoritmos específicos para compresión y/o cálculos como SUM, AVG.

Como cada columna es archivo, este puede ser distribuido entre distintos nodos o filesystems como RAID.

## Product Columns

ID	Name	Color	Unit Price
1	Camcorder	Red	112.25
2	Camera	Red	97.50
3	Smartphone	White	100.00
4	Console	Black	112.25
5	TV	Blue	1,240.85
6	CD	Red	39.99
7	Touch screen	Blue	45.12
8	PDA	Black	120.25
9	Keyboard	Black	120.50



## Columnar Database

En una BD Relacional agregar/eliminar una columna significa reorganizar las filas. Un valor nulo ocupa el mismo espacio que existiera.

En una BD columnar agregar/eliminar una columna es una operación independiente. Los valores nulos pueden no cargarse.



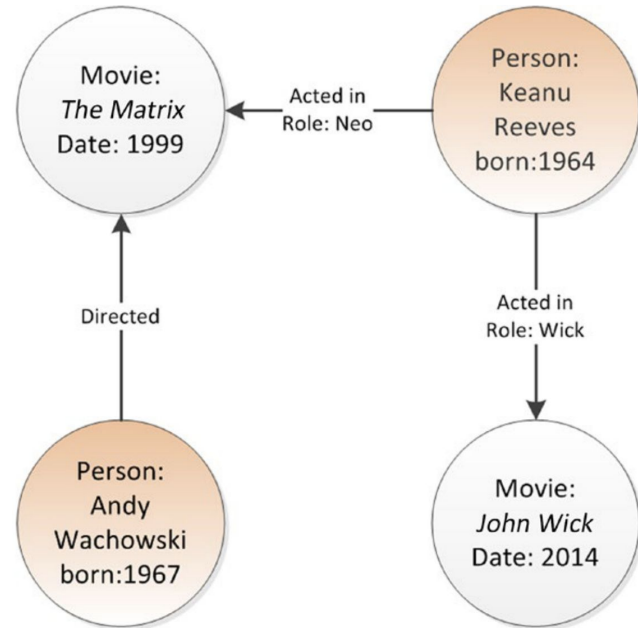
Amazon **Redshift**

# Graph Database

Se enfocan en modelar las relaciones entre los datos más que su estructura y restricciones.

Tiene su fundamento en la teoría de grafos.

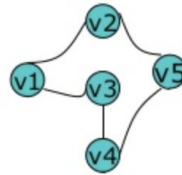
Define nodos y arcos. Los nodos son abstracciones y serían las entidades del modelo relacional. Los arcos(aristas) conectan los nodos representando una relación.



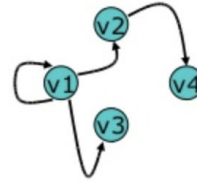
# Graph Database

Los nodos pueden tener cualquier tipo de dato y los arcos pueden tener una o más propiedades.

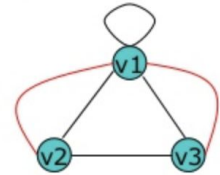
Se pueden representar grafos dirigidos, no dirigidos, completos, partidos, etc.



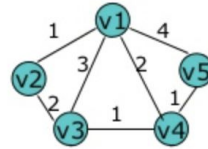
No Dirigido



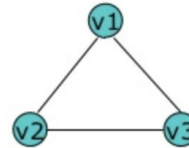
Dirigido



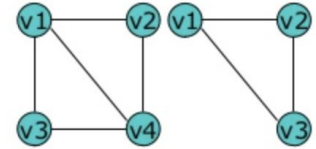
No Simple



Ponderado



Completo



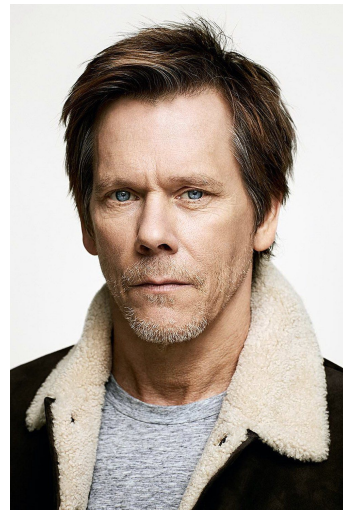
De Similitud



# Graph Database

Los arcos pueden tener un peso o probabilidad asociada, permitiendo realizar complejos queries o consultas para obtener nueva información.

Ciertos modelos son más fácilmente representables como grafos.



<http://oracleofbacon.org/>

Kevin Spacey has a Bacon number of 2.

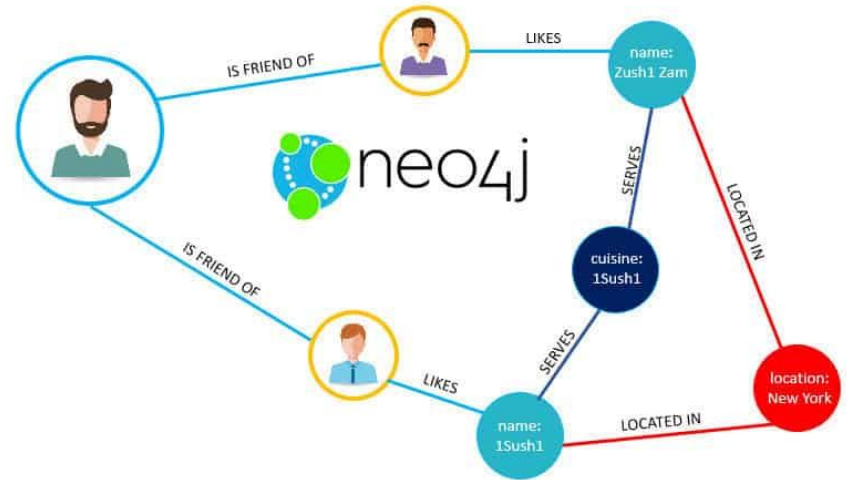


Kevin Bacon to Kevin Spacey Find link

# Graph Database

Sus aplicaciones van desde Inteligencia Artificial, modelos de inferencia, redes sociales, recomendaciones, networking, aplicaciones químicas entre otras.

Distintas implementaciones ofrecen distintos lenguajes como: Gremlin, Sparkql, RDF, etc





# Referencias

- Joe Celko's Complete Guide to NoSQL. Elsevier. ISBN: 978-0-12-407192-6
- Getting Started with NoSQL. Pack Publishing. ISBN: 978-1-84969-4-988
- Next generation databases. Nosql, newsql and big data. Apress. ISBN:978-1-4842-1330-8
- <https://redis.io/>
- <https://try.redis.io/>
- <https://neo4j.com/>
- <https://www.mongodb.com/document-databases>
- [https://docs.aws.amazon.com/redshift/latest/dg/c\\_columnar\\_storage\\_disk\\_mem\\_mgmnt.html](https://docs.aws.amazon.com/redshift/latest/dg/c_columnar_storage_disk_mem_mgmnt.html)