

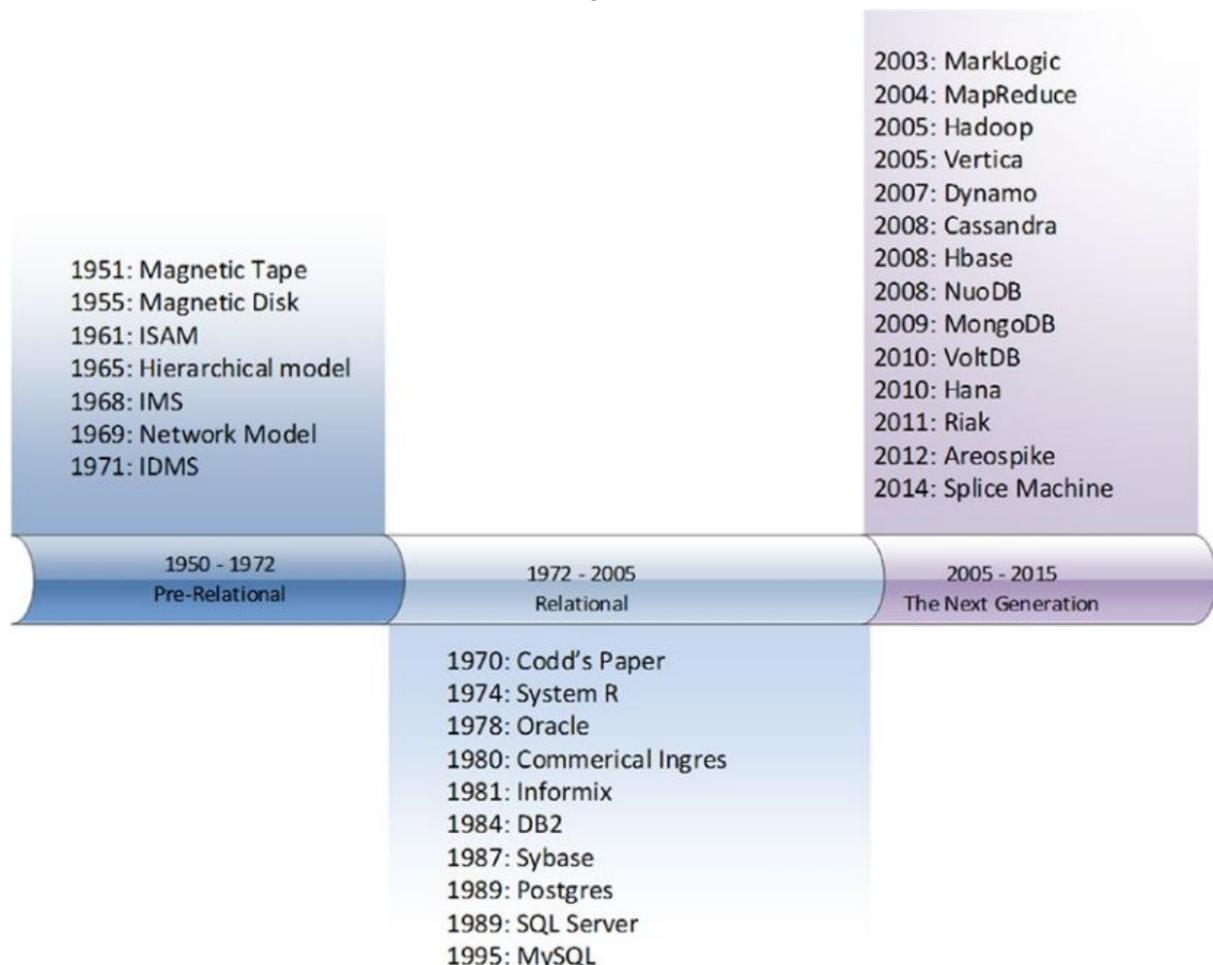
Bases de Datos NoSQL

Definición e historia.

La definición más simple de las Bases de datos NoSql es que aquella que no utiliza un modelo relacional y SQL como lenguaje de consulta/modificación. Esta definición más que explicar que es el modelo NoSql explica que **no es**. Esto se debe a la basta cantidad de subdivisiones que se pueden realizar a las BD NoSQL, una tan distinta a la otra como lo es al modelo relacional.

Aunque primordialmente nos enfocaremos en las bases de datos mas recientes y populares para entender qué son las Base de datos NoSQL es conveniente hacer un repaso por su historia, entender cuándo y por que surgieron, para de esta forma ver sus usos y utilidades.

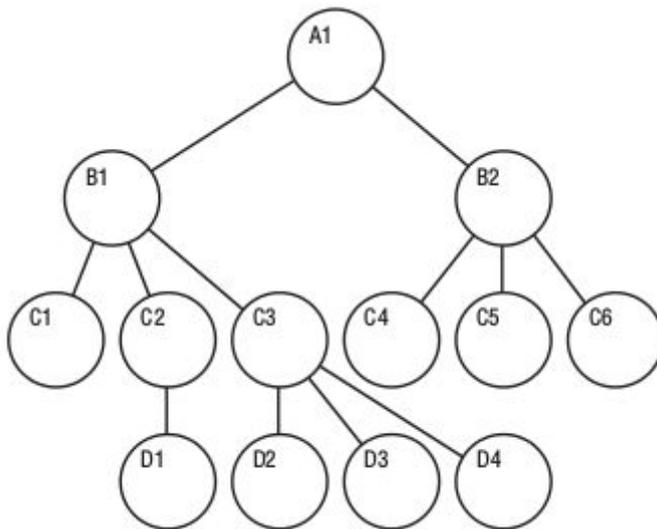
Guy Harrison¹ especifica una línea de tiempo de las grandes lanzamientos en innovaciones en el mundo de base de datos, definiendo 3 grandes subdivisiones.



Pre-relacional

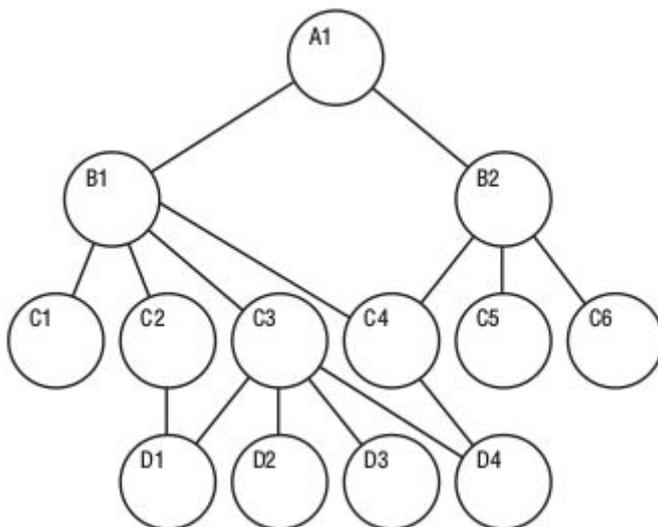
Dentro de la Subdivisión **pre-relacional** podemos encontrar 2 grandes modelos de bases de datos, estos es los puede denominar modelos *navegaciones* ya que en los mismos para llegar a un objeto se debía recorrer un camino hecho con punteros o links. Estos modelos son:

- **Jerárquico**, este se asemejaba a una estructura de árbol con relaciones padre-hijo en la cual un padre podía tener varios hijos pero un hijo solo un padre (similar a como están organizados los filesystems). El modelo representaba muy bien las relaciones 1 a N, relativamente bien las 1 a 1 pero tenía grandes problemas en representar relaciones N a N, a su vez presentaba una gran rigidez ya que agregar/modificar una relación podía implicar un cambio total en la estructura del árbol.



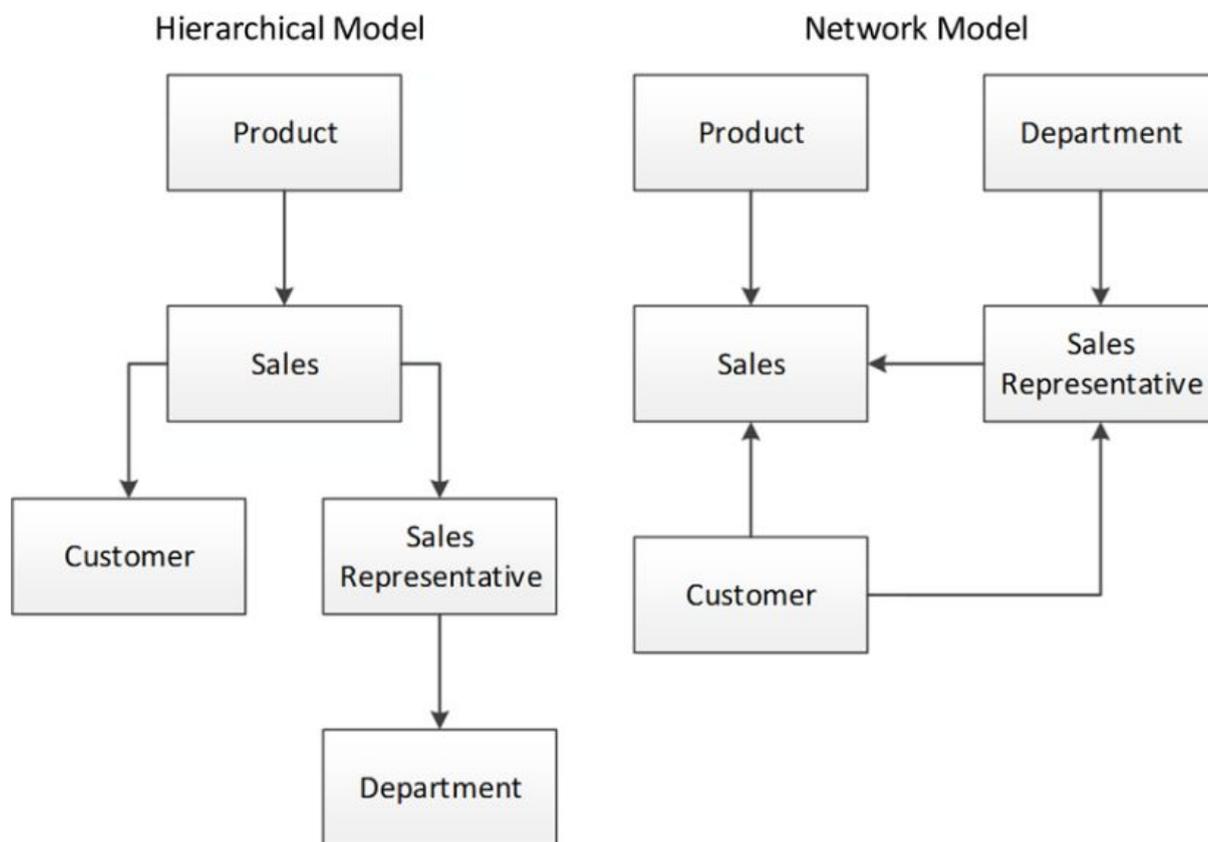
2

- **Red**, surgió como una mejora al modelo jerárquico permitiendo a cada hijo tener varios padres, por lo tanto permitiendo relaciones N a N. Pese a esta mejora el modelo seguía manteniendo poca flexibilidad y relativa complejidad para el software que lo use.



3

Estos modelos dominaron el mercado hasta finales de los 70, eran utilizados en grandes mainframes y su enfoque era en operaciones CRUD (Create, Read, Update and Delete).



Comparación de modelos por Guy Harrison

Relacional

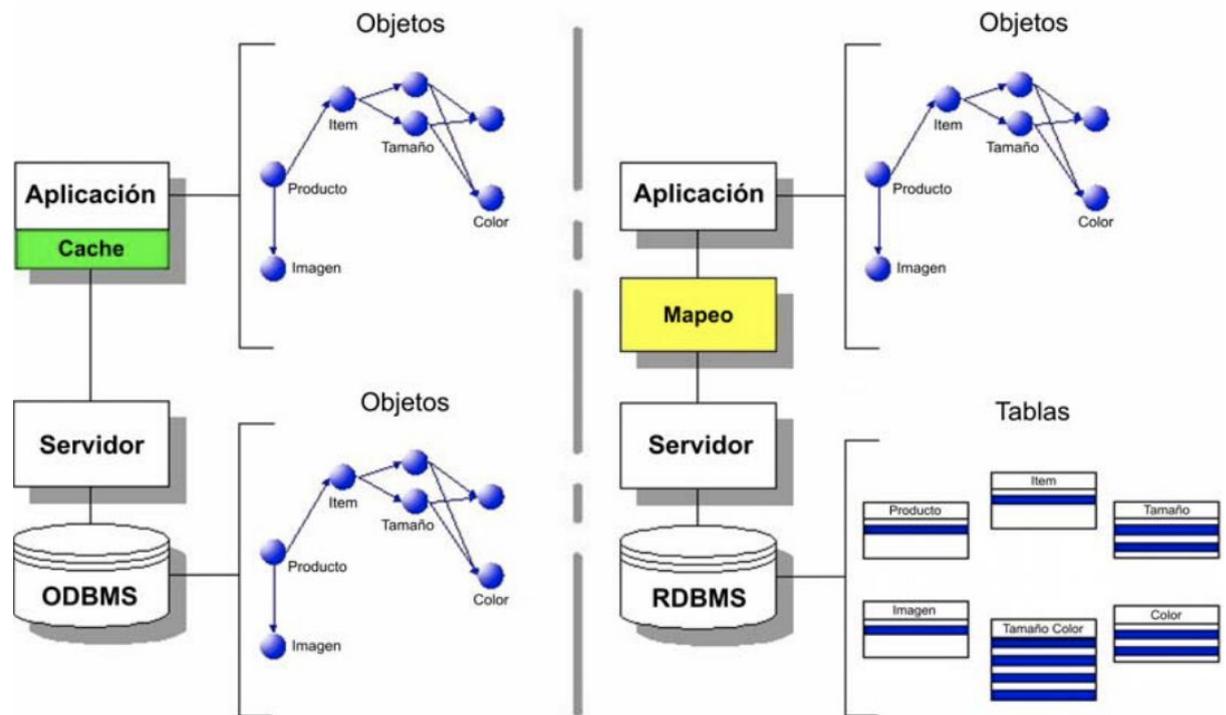
En 1970 Edgar Codd presento su paper "Relational Model of Data for Large Shared Data Banks."⁴ sentando las bases del modelo relacional e iniciando la etapa definida con este nombre por Guy Harrison.

Dado que el modelo relacional fue visto y perfeccionado durante Base de Datos 1 no se profundizará sobre el mismo.

A mediados de los 80 un paradigma empieza a hacerse fuerte en el mundo de la programación, el paradigma orientado a objetos. Este paradigma no resulta totalmente compatible con el modelo relacional, derivando en diferencias llamadas **impedance mismatch** siendo un problema constante a resolver por parte de los programadores. A su vez, en áreas donde se manejan grandes volúmenes de datos complejos resultaba un desafío representar esa información en el modelo relacional. Frente a estos problemas aparece un nuevo tipo de bases de datos, las Bases de Datos Orientadas a Objetos.

- **Bases de Datos Orientadas a Objetos(BDOO):** Es una BD la cual persiste la información como objetos los cuales tiene atributos y están relacionados entre ellos.

El acceso y navegación entre los mismo se realiza mediante punteros, lo cual provee una alta velocidad ya que no es necesario hacer 'joins'. Como ventaja frente al modelo relacional es que brinda persistencia transparente, esto significa que los objetos utilizados en la base de datos se manejan en el lenguaje de programación directamente, sin la necesidad de conversiones/traduccion. A su vez las BDOO soportan las características del mundo OO como: Agregación, Encapsulamiento, Herencia y Polimorfismo



Comparacion de modelos por German Viscuso

Next Generation

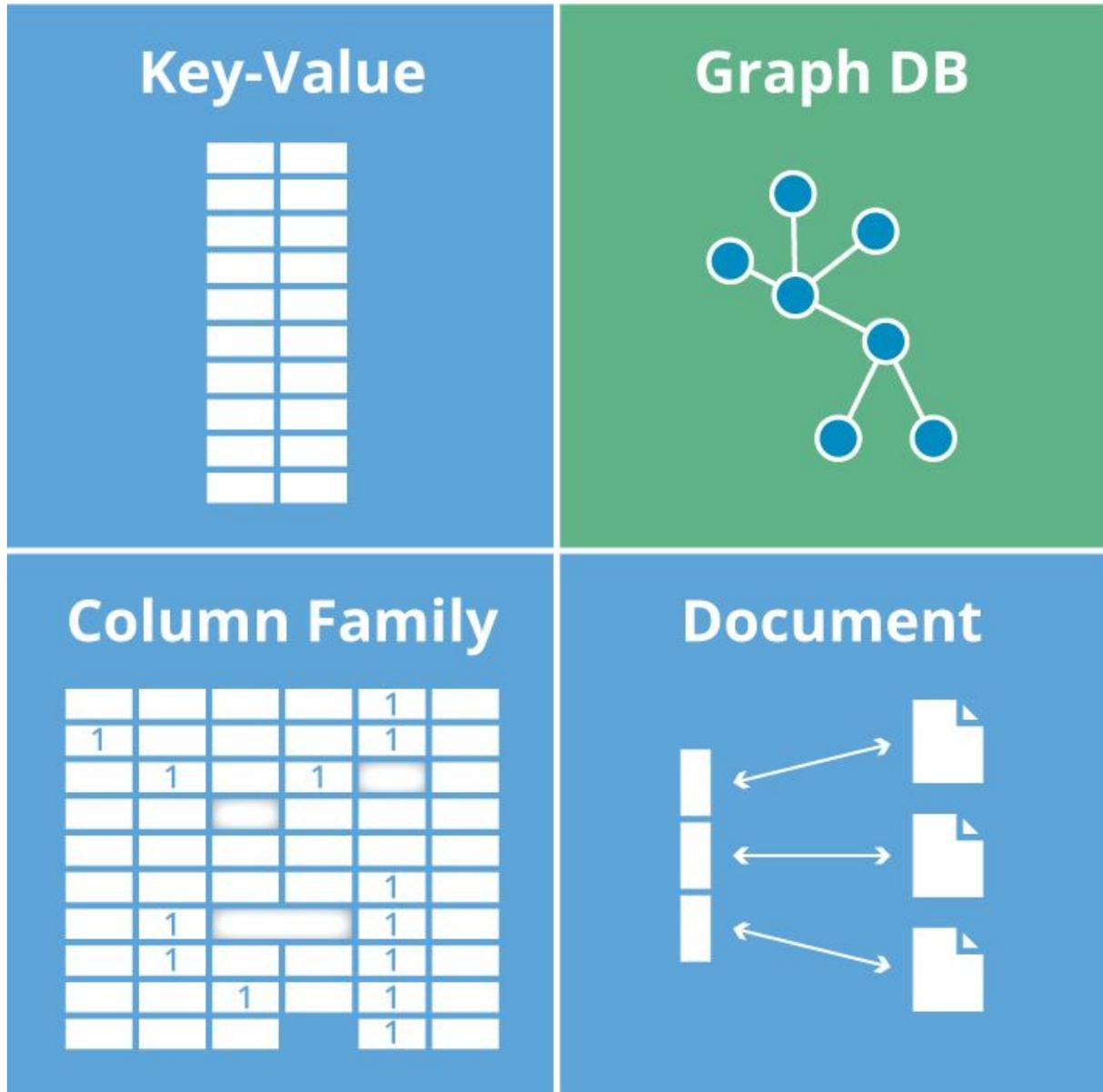
A partir del 2000 con el rápido crecimiento de internet y la web, la evolución y aparición de nuevos lenguajes de programación, uso de los smartphones, explosión de datos de usuarios y cambios en las necesidades de las empresas surgieron nuevos paradigmas de bases de datos.

Una de las primeras empresas en enfrentarse al problema de grandes volúmenes de información fue Google, para esto la empresa en 2003 publicó un paper acerca de GFS, su file system distribuido, en 2004 de MapReduce un algoritmo de procesamiento distribuido y en 2006 de su base de datos distribuida **BigTable**, perteneciente al grupo de las **bases de datos columnares**.

Empresas de e-commerce como Amazon también tienen dificultades a la hora de brindar un servicio que soporte miles de transacciones por minuto y fácil de escalar, por lo que en 2008 sacan a la luz a **DynamoDB** una BD de tipo "key-value".

Desde 2004 varios sitios web empezaron a utilizar un estilo de programación llamado AJAX, el cual permite brindar mayor interactividad enviando mensajes al servidor de manera asíncrona en formato XML o JSON, fue solo una cuestión de tiempo para que surgieran BD específicas para estos formatos, estas BD se llaman BD **documentales** y dos ejemplos son: MongoDB y CouchBase.

En 2007 Neo4J es lanzado y con él resurgen las **Base de datos de grafos**, las cuales se utilizan en escenarios muy específicos y en casos en los que la relación entre los datos es más importante que los datos en sí.



Estas 4 caracterizaciones de Bases de Datos NoSQL no son las únicas que existen, además de existir híbridas capaces de manejar más de un modelo, también existen bases de datos geoespaciales, biométricas, textuales, multivaluadas, en memoria, etc. Sin embargo estas 4 serán las tratadas en la materia.

Database	Column oriented	Document store	Key value store	Graph
Amazon SimpleDB	No	No	Yes	No
BaseX	No	Yes	No	No
Cassandra	Yes	Yes	No	No
CouchDB	No	Yes	No	No
Google Datastore	Yes	No	No	No
HBase	Yes	No	No	No
MemcacheDB	No	No	Yes	No
MongoDB	No	Yes	No	No
Neo4j	No	No	No	Yes
Redis	No	Yes	Yes	No

Caracterización de bases de datos. Fuente 5. Pag: 61

Casos de uso de Bases de datos NoSQL

Base de Datos	Empresa
Casandra	Facebook, Netflix,
CouchBase	Paypal, Ebay
DynamoDB	Amazon
Hbase	Pinterest, Yahoo, Verizon
MongoDB	SEGA, Coinbase, Adobe
Neo4J	Comcast, Monsanto, NASA, LinkedIn
Redis	Twitter, Github, Snapchat

Tabla armando usando fuentes: 1º Apendice A, 6, 7, 8, 9 y 10

Theorema CAP y BASE

El Teorema CAP formulado por Eric Brewer en 1999 explica que para un sistema de almacenamiento distribuido es imposible ofrecer 2 de las siguiente 3 características en todo momento:

- Consistency: Todos los nodos ven la misma información y esta sigue las reglas establecidas.
- Availability: Toda petición que se hace al sistema tiene una respuesta
- Partition Tolerance: Ante la caída de un nodo el sistema sigue funcionando

El teorema indica que en un ambiente distribuido, la caída de un nodo es un hecho probable y frente a ese hecho el sistema deberá elegir entre:

- Dar una respuesta al cliente pero que no garantiza mantener la consistencia
- No dar una respuesta al cliente, bajando la disponibilidad pero manteniendo consistencia.

Frente a este problema, y dado que el surgimiento de las Bases de Datos NoSql se dio en ambientes distribuidos donde mantener una alta disponibilidad es esencial, muchas bases de datos dejaron el modelo ACID y optaron por otro, el modelo BASE:

- Basic availability: Toda petición tiene una respuesta garantizada que puede ser correcta o un fallo por que la información esta en un estado inconsistente o cambiante.
- Soft state: El estado del sistema puede cambiar, aun cuando no haya inputs al mismo.
- Eventual consistency: El sistema podrá ser inconsistente por momentos pero eventualmente sera consistente.

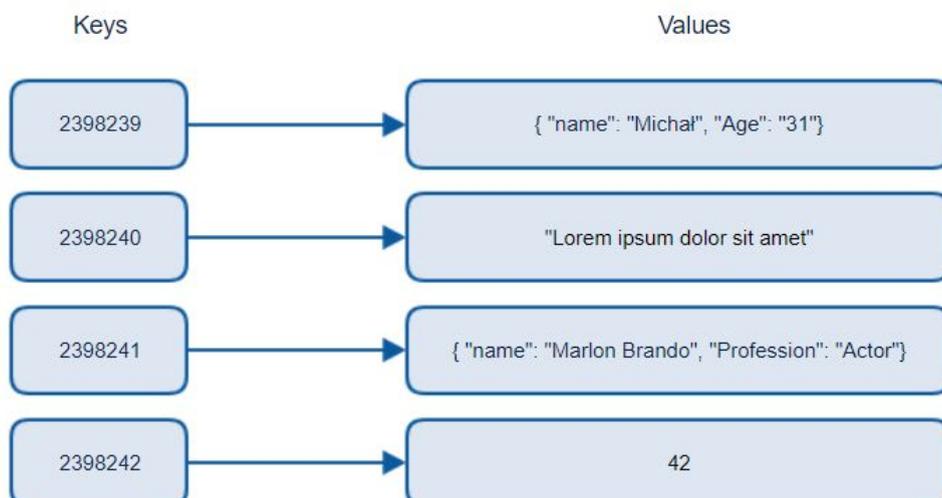
Esto no implica que todas las BD NoSql no sean ACID, es más, algunas hasta permiten elegir entre ambos modelos o usan distintas variantes de los mismos!

Bases de datos key-value

Las bases de datos Key-Value guardan la información en un arreglo asociativo de Clave-Valor. La clave debe ser única y tanto esta como el valor asociado pueden ser de cualquier tipo de dato, por lo tanto se dice que la base de datos no tiene esquema, por lo tanto todos los chequeos de integridad deben realizarse en la aplicación.

Una BD key-value permite operaciones las operaciones básicas de CRUD con la diferencia de que al Recuperar una clave, si esta no existe retorna un error.

En este tipo de BD es fundamental un buen diseño de las claves.



Bases de datos documentales - Document store/oriented

Las bases de datos documentales permiten la inserción, obtención y manipulación de información en un formato semi-estructurado(ej: JSON, XML) o estructurado (XSD) llamado documento.

Las bases de datos documentales son flexibles y no están restringidas a un esquema, esto significa que la base de datos no tiene que saber de antemano la estructura del dato o sus tipos de valores.

Esto significa que en la base de datos podríamos tener un documento con la siguiente información:

```
{
  "EmployeeID": "SM1",
  "FirstName" : "Anuj",
  "LastName" : "Sharma",
  "Age" : 45,
  "Salary" : 10000000
}
```

Y un segundo documento con la siguiente información

```
{
  "EmployeeID": "MM2",
  "FirstName" : "Anand",
  "Age": 34,
  "Salary": 5000000,
  "Address" : {
    "Line1" : "123, 4th Street",
    "City" : "Bangalore",
    "State" : "Karnataka"
  },
  "Projects" : [
    "nosql-migration",
    "Top-secret-007"
  ]
}
```

En otras palabras, la estructura de los documentos es flexible, permitiendo que cada uno contenga distintos atributos de ser necesario.

Algunas ventajas que presentan las bases de datos documentales son:

- Dada la estructura jerárquica, no es necesario hacer JOINS reduciendo el tiempo de búsqueda
- Rápida evolución frente a los cambios gracias a su naturaleza desestructurada.
- Reduce tiempos de desarrollo cuando se usa formato JSON en la BD y aplicación. (ver MEAN y MERN)

Document 1

```
{
  "id": "1",
  "name": "John Smith",
  "isActive": true,
  "dob": "1964-30-08"
}
```

Document 2

```
{
  "id": "2",
  "fullName": "Sarah Jones",
  "isActive": false,
  "dob": "2002-02-18"
}
```

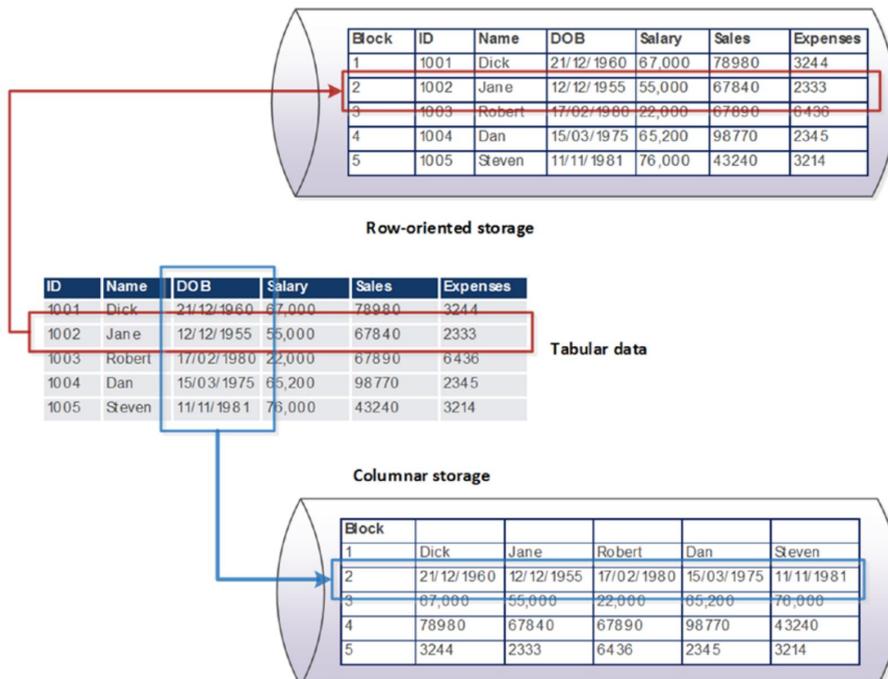
Document 3

```
{
  "id": "3",
  "fullName": {
    "first": "Adam",
    "last": "Stark"
  },
  "isActive": true,
  "dob": "2015-04-19"
}
```

Bases de datos Columnares

En las bases de datos relacionales, una tabla es un conjunto de filas con una única estructura, una fila es un conjunto desordenado de columnas donde cada una guarda un valor dependiendo un único dominio al que pertenece. En el File System usualmente se sigue el mismo modelo, siendo una tabla un archivo, una fila un registro y campos por valor de una columna.

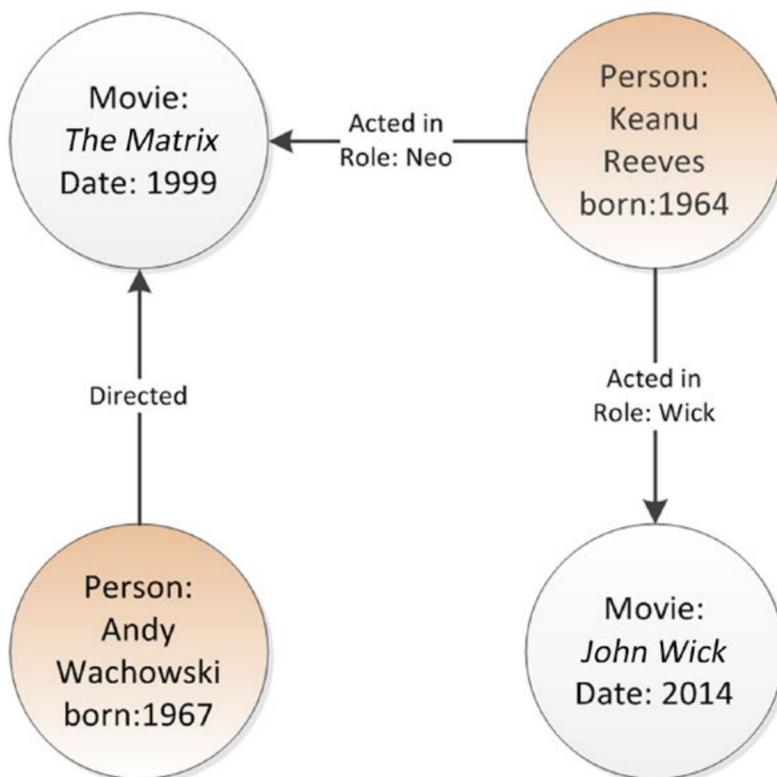
El modelo columnar busca guardar cada columna en una estructura distinta, y a partir de la unión de estas armar las filas y tablas. Dado que cada columna es una estructura distinta, se pueden usar algoritmos de compresión específicos para ese tipo de dato como también optimizar queries como SUM(), AVG() entre otras.



Bases de datos Orientadas a Grafos - Graph Database

Las bases de datos orientadas a grafos se enfocan en modelar las relaciones más que estructura de los datos, no se utilizan para hacer análisis aritméticos o estadísticos sino para ver la relación entre los datos.

Estas bases de datos tienen su fundamento en la **teoría de grafos**. Un grafo tiene 2 tipos de cosas: Nodos y Arcos. Los Nodos son abstracciones y se podría considerar que modelan lo que es una entidad en el modelo relacional. Los Arcos son los encargados de conectar los nodos, un arco puede ser dirigido o no dirigido, contener información y peso.



Bibliografía:

1. Next generation databases. Nosql, newsql and big data. Apress. ISBN:978-1-4842-1330-8
2. <https://mariadb.com/kb/en/library/understanding-the-hierarchical-database-model/>
3. <https://mariadb.com/kb/en/library/understanding-the-network-database-model/>
4. <https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>
5. Getting Started with NoSQL. Pack Publishing. ISBN: 978-1-84969-4-988
6. <https://stackshare.io/hbase>
7. <https://www.mongodb.com/who-uses-mongodb>
8. <https://www.couchbase.com/>
9. <https://neo4j.com/customers/>
10. <https://redis.io/topics/whos-using-redis>
11. <https://www.mongodb.com/document-databases>
12. Joe Celko's Complete Guide to NoSQL. Elsevier. ISBN: 978-0-12-407192-6