

# Análisis comparativo de bases de datos relacionales y no relacionales

Jonathan Martín Valera

28 de marzo de 2019

# 1. Introducción

Las bases de datos son el método preferido para el almacenamiento de datos.

Desde las grandes aplicaciones multiusuario, hasta los teléfonos móviles y las agendas electrónicas utilizan tecnología de bases de datos para asegurar la integridad de los datos.



## 2. Fundamentos de base de datos

### ¿Qué es una base de datos?

Entendemos como Base de Datos un conjunto de datos estructurado y almacenado de forma sistemática con objeto de facilitar su posterior utilización. Una base de datos puede, por tanto, constituirse con cualquier tipo de datos.

### Ventajas del almacenamiento estructurado

- Mayor independencia.

## 2. Fundamentos de base de datos

### ¿Qué es una base de datos?

Entendemos como Base de Datos un conjunto de datos estructurado y almacenado de forma sistemática con objeto de facilitar su posterior utilización. Una base de datos puede, por tanto, constituirse con cualquier tipo de datos.

### Ventajas del almacenamiento estructurado

- Mayor independencia.
- Mayor disponibilidad.

## 2. Fundamentos de base de datos

### ¿Qué es una base de datos?

Entendemos como Base de Datos un conjunto de datos estructurado y almacenado de forma sistemática con objeto de facilitar su posterior utilización. Una base de datos puede, por tanto, constituirse con cualquier tipo de datos.

### Ventajas del almacenamiento estructurado

- Mayor independencia.
- Mayor disponibilidad.
- Mayor seguridad (protección de los datos).

## 2. Fundamentos de base de datos

### ¿Qué es una base de datos?

Entendemos como Base de Datos un conjunto de datos estructurado y almacenado de forma sistemática con objeto de facilitar su posterior utilización. Una base de datos puede, por tanto, constituirse con cualquier tipo de datos.

### Ventajas del almacenamiento estructurado

- Mayor independencia.
- Mayor disponibilidad.
- Mayor seguridad (protección de los datos).
- **Menor redundancia.**

## 2. Fundamentos de base de datos

### ¿Qué es una base de datos?

Entendemos como Base de Datos un conjunto de datos estructurado y almacenado de forma sistemática con objeto de facilitar su posterior utilización. Una base de datos puede, por tanto, constituirse con cualquier tipo de datos.

### Ventajas del almacenamiento estructurado

- Mayor independencia.
- Mayor disponibilidad.
- Mayor seguridad (protección de los datos).
- Menor redundancia.
- Mayor eficiencia en la captura, codificación y entrada de datos.

## 2. Fundamentos de base de datos

### ¿Qué es una base de datos?

Entendemos como Base de Datos un conjunto de datos estructurado y almacenado de forma sistemática con objeto de facilitar su posterior utilización. Una base de datos puede, por tanto, constituirse con cualquier tipo de datos.

### Resultados de la explotación de la base de datos

- Mayor coherencia.



## 2. Fundamentos de base de datos

### ¿Qué es una base de datos?

Entendemos como Base de Datos un conjunto de datos estructurado y almacenado de forma sistemática con objeto de facilitar su posterior utilización. Una base de datos puede, por tanto, constituirse con cualquier tipo de datos.

### Resultados de la explotación de la base de datos

- Mayor coherencia.
- Mayor eficiencia.

## 2. Fundamentos de base de datos

### ¿Qué es una base de datos?

Entendemos como Base de Datos un conjunto de datos estructurado y almacenado de forma sistemática con objeto de facilitar su posterior utilización. Una base de datos puede, por tanto, constituirse con cualquier tipo de datos.

### Resultados de la explotación de la base de datos

- Mayor coherencia.
- Mayor eficiencia.
- Mayor valor informativo.

## 2.1 Tipos de bases de datos

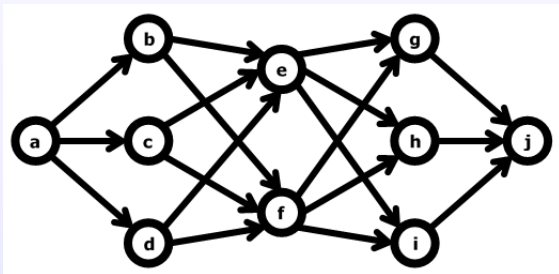
### 1 Modelo de base de datos plana

Código Postal
9063635
9345452
6345469
6654760
7754742



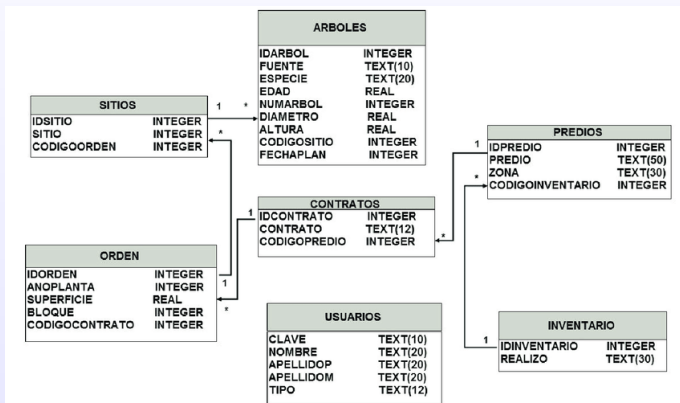
## 2.1 Tipos de bases de datos

### 3 Modelo de red



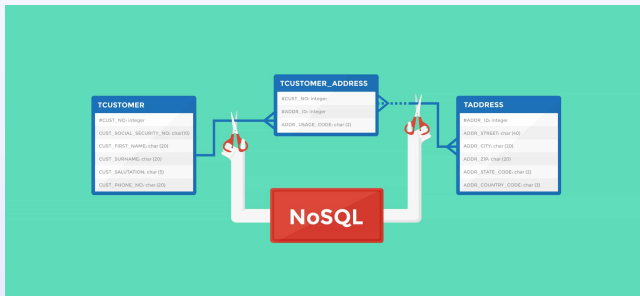
# 2.1 Tipos de bases de datos

## Modelo de bases de datos relacionales



## 2.1 Tipos de bases de datos

### 5 Modelo de bases de datos NoSQL



## 2.2 Sistema gestor de base de datos

### SGBD

Un sistema gestor de bases de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener una base de datos, y proporciona acceso controlado a la misma

- Definición de la base de datos.
- Inserción, actualización, eliminación y consulta de datos.
- Seguridad.
- Integridad.
- Concurrencia.
- Recuperación.
- Catálogos.



### 3. Bases de datos en la actualidad

Hoy en día podemos hablar sobre dos principales modelos de bases de datos: el modelo SQL y NoSQL.



### 3. Bases de datos en la actualidad

Hoy en día podemos hablar sobre dos principales modelos de bases de datos: el modelo SQL y NoSQL.



## 4. Bases de datos relacionales. SQL

En el modelo relacional las dos capas de diseño conceptual y lógico, se parecen mucho. Generalmente se implementan mediante diagramas de Entidad/Relación (modelo conceptual) y tablas y relaciones entre éstas (modelo lógico).

Se rige por algunas normas sencillas

- Todos los datos se representan en forma de tablas.

## 4. Bases de datos relacionales. SQL

En el modelo relacional las dos capas de diseño conceptual y lógico, se parecen mucho. Generalmente se implementan mediante diagramas de Entidad/Relación (modelo conceptual) y tablas y relaciones entre éstas (modelo lógico).

Se rige por algunas normas sencillas

- Todos los datos se representan en forma de tablas.
- Las tablas están compuestas por filas (o registros) y columnas (o campos) que almacenan cada uno de los registros

## 4. Bases de datos relacionales. SQL

En el modelo relacional las dos capas de diseño conceptual y lógico, se parecen mucho. Generalmente se implementan mediante diagramas de Entidad/Relación (modelo conceptual) y tablas y relaciones entre éstas (modelo lógico).

Se rige por algunas normas sencillas

- Todos los datos se representan en forma de tablas.
- Las tablas están compuestas por filas (o registros) y columnas (o campos) que almacenan cada uno de los registros
- Cada tabla debe poseer una clave primaria. Identificador único.

## 4. Bases de datos relacionales. SQL

En el modelo relacional las dos capas de diseño conceptual y lógico, se parecen mucho. Generalmente se implementan mediante diagramas de Entidad/Relación (modelo conceptual) y tablas y relaciones entre éstas (modelo lógico).

Se rige por algunas normas sencillas

- Todos los datos se representan en forma de tablas.
- Las tablas están compuestas por filas (o registros) y columnas (o campos) que almacenan cada uno de los registros
- Cada tabla debe poseer una clave primaria. Identificador único.
- **Clave externa.**

## 4. Bases de datos relacionales. SQL

Sus principales **ventajas** son:

- Mayor soporte y herramientas.
- Es una tecnología ampliamente conocida.
- Transaccionalidad entre tablas.
- Los datos deben cumplir con el tipo de dato definido en su estructura.

## 4. Bases de datos relacionales. SQL

Sus principales **inconvenientes** son:

- Mayor soporte y herramientas.
- Necesita más procesamiento.
- La escalabilidad es reducida.



## 4. Bases de datos relacionales. SQL

Algunas tecnologías conocidas son:



## 5. Bases de datos no relacionales. NoSQL

### Bases de datos clave-valor

Los datos están formados en dos partes, una cadena que representa la clave y los datos reales a los que se hace referencia como valor, creando así un par “clave-valor”, que es una clave única con la que se identifica cada elemento.

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

## 5. Bases de datos no relacionales. NoSQL

### Bases de datos documentales

Las bases de datos documentales almacenan sus datos en forma de documentos, generalmente utilizando como estructura JSON o XML.

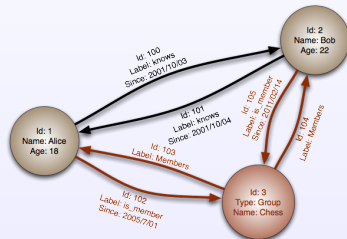


**Document data model**

# 5. Bases de datos no relacionales. NoSQL

## Bases de datos orientadas a grafos

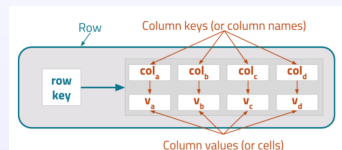
Las bases de datos orientadas a grafos representan las entidades como nodos de un grafo y las relaciones como las aristas entre ellos. Utiliza una técnica llamada adyacencia libre de índice (index-free adjacency) la cual consiste en que cada nodo tiene un puntero que apunta al nodo adyacente.



## 5. Bases de datos no relacionales. NoSQL

### Bases de datos orientadas a columnas

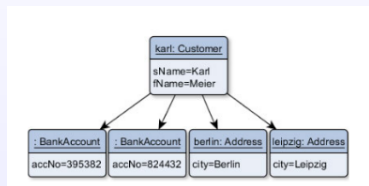
Son bases de datos similares a las bases de datos relacionales, aunque comparten el concepto de almacenamiento columna a columna de las bases de datos basadas en filas, los almacenes de columnas no almacenan los datos en tablas sino en arquitecturas distribuidas masivamente.



## 5. Bases de datos no relacionales. NoSQL

### Bases de datos orientadas a objetos

Son bases de datos en las cuales los datos o la información a almacenar se representan como un objeto (similar al de programación orientada a objetos).



## 5. Bases de datos no relacionales. NoSQL

Las principales **ventajas** de este modelo son:

- La **escalabilidad** y su carácter descentralizado.
- Suelen ser bases de datos mucho más **abiertas y flexibles**.
- Se pueden **hacer cambios de los esquemas** sin tener que parar bases de datos.
- **Escalabilidad horizontal**: son capaces de crecer en número de máquinas, en lugar de tener que residir en grandes máquinas.
- Se pueden ejecutar en **máquinas con pocos recursos**.
- **Optimización de consultas** en base de datos para grandes cantidades de datos.

## 5. Bases de datos no relacionales. NoSQL

Los principales **inconvenientes** de este modelo son:

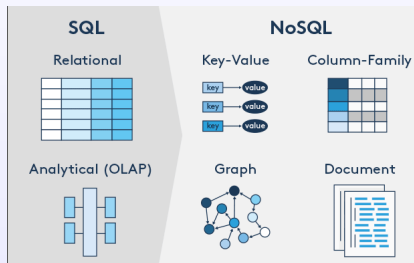
- No todas las bases de datos NoSQL contemplan la **atomicidad** de las instrucciones y la **integridad de los datos**.
- Problemas de **compatibilidad entre instrucciones SQL**.
- Falta de **estandarización**.
- Soporte **multiplataforma**.
- **herramientas** de administración **no muy usables**.



## 6. Comparativa SQL vs NoSQL

¿Es NoSQL el sustituto directo para evolucionar de las antiguas bases de datos relacionales?

Esto no es cierto para todos los casos, sino que se tiene que tener en cuenta el diseño del problema para poder saber a qué modelo de base de datos se puede ajustar mejor.



## 6. Comparativa SQL vs NoSQL

### Integridad de datos

**SQL:** Las tablas tienen estructuras rígidas, donde cada dato tiene un tipo definido, no podemos almacenar datos de otro tipo diferente, y no se vale más de un dato en un mismo campo.

**NoSQL:** Hay varios tipos de base de datos NoSQL, pero en general, ninguna te exige que definas el tipo de datos que vas a almacenar.

## 6. Comparativa SQL vs NoSQL

### Operaciones atómicas

**SQL:** Las bases de datos relacionales tienen atomicidad gracias a que sus tablas están conectadas y pueden “ponerse de acuerdo” para no aceptar cambios nuevos hasta que termine una transacción.

**NoSQL:** Datos no relacionales = no hay relaciones sobre las que hacer una transacción atómica. Simplemente, cuando quieres hacer cambios en 5 entidades diferentes, de frente o detrás de cámaras habrá 5 llamadas diferentes a la base de datos una detrás de otra..

## 6. Comparativa SQL vs NoSQL

### Escalabilidad

**SQL:** La verdad es que la mayoría de soluciones SQL tienen buen soporte para escalar verticalmente.

**NoSQL:** Cuando no tienes la consistencia de datos como prioridad, distribuir y replicar tu base de datos en múltiples máquinas es trivial, y por eso se considera que el NoSQL es excelente para bases de datos necesitan escalar horizontalmente.

## 6. Comparativa SQL vs NoSQL

### Velocidad

**SQL:** Las garantías que te dan las relaciones conllevan un precio. Esto es más evidente cuando empezamos a hacer consultas con “joins” (que involucran múltiples entidades) y de repente una búsqueda puede tardar minutos y hasta horas debido a la gran cantidad de datos que está revisando.

**NoSQL:** Las bases de datos no relacionales suelen contar con mecanismos de búsqueda sumamente rápida para conseguir un dato específico entre millones.

## 6. Comparativa SQL vs NoSQL

### Consistencia vs redundancia

**SQL:** La consistencia de datos es asegurarse de que un único dato este una única vez en toda la base de datos; y se suele lograr con el proceso de “Normalización”.

**NoSQL:** La redundancia es repetir adrede los datos a conveniencia en varias partes de la BD.

## 6. Comparativa SQL vs NoSQL

### Comodidad para el desarrollador

**SQL:** La comunidad SQL lleva décadas madurando, y esto se traduce no solo en mejores herramientas administrativas, sino en estándares mejores definidos, mayor documentación.

**NoSQL:** Aquí el punto fuerte es la conveniencia: factores como que los datos no necesiten tipos o que puedas aprovechar la redundancia, hacen más flexible el desarrollar con NoSQL.

## 6. Comparativa SQL vs NoSQL

Característica	NoSQL	SQL
Rendimiento	Alto	Bajo
Confiabilidad	Baja	Buena
Disponibilidad	Buena	Buena
Consistencia	Baja	Buena
Almacenamiento	Enormes cantidades	Cantidades medio/grande
Escalabilidad	Alta	Alta pero más cara

Tabla: Resumen de las características NoSQL y SQL



## 7. Conclusiones

Muchas personas piensan que las tecnologías NoSQL son lo “nuevo” y por lo tanto todo debe migrar a este modelo, pero es un grave error. NoSQL no es un reemplazo, es simplemente un modelo diferente que ofrece ventajas y soluciones a problemas que poseen las bases de datos relacionales.

### SQL

- Respeto por la integridad de los datos.
- Si se necesita información con estructura consistente.
- Si la jerarquía de datos es importante. . .

### NoSQL

- Si la escalabilidad es importante.
- Si no es necesario respetar la integridad de los datos.
- Desarrollo Big data.

# Gracias por su atención

Para más información se puede consultar el repositorio en github.



[https://github.com/jmv74211/LaTeX/tree/master/ejercicios/Entrega\\_final](https://github.com/jmv74211/LaTeX/tree/master/ejercicios/Entrega_final)

