

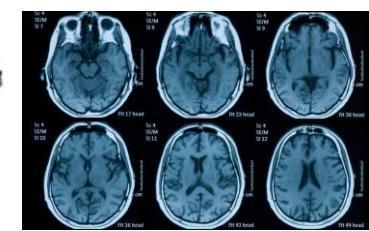
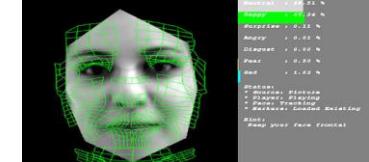
NoSQL

NoSQL			
Definición	NoSQL vs. SQL	Bases de datos NoSQL	
<p>El movimiento NoSQL incluye todas las bases de datos con arquitectura distinta a la utilizada en sistemas relacionales tradicionales.</p>	<p>NoSQL:</p> <ul style="list-style-type: none">- Es una tecnología que se lleva utilizando desde los años 60, aunque el nombre fue acuñado en 2009.- Las bases de datos NoSQL se caracterizan por su escalabilidad horizontal, sencillez, evitan cuellos de botella, permiten manejar grandes volúmenes de datos y su puesta en funcionamiento es más asequible.	<p>Cada base de datos a continuación corresponde a un tipo de NoSQL</p>	
<p>Tipos</p> <p>Las bases de datos NoSQL pueden categorizarse en cuatro grupos:</p> <ul style="list-style-type: none">- Clave-valor simples- Clave-valor sofisticadas- Basadas en documentos- Basadas en grafos	<p>Desventajas:</p> <ul style="list-style-type: none">- La falta de madurez, en algunos casos, es un problema porque lleva consigo escasa documentación y comunidades de usuarios muy reducidas.- Problemas de compatibilidad con otras herramientas, aunque esto es un problema que tiende a desaparecer debido a la apuesta que hacen las grandes compañías al integrar este tipo de productos en sus servicios.	<p>Apache Cassandra</p> <p>Es una base de datos distribuida de código abierto escrita en Java. Todos sus nodos actúan por igual, agrupándose en anillos. Permite sistemas de réplicas y el acceso a los datos se hace a través de CQL.</p>	<p>Neo4J</p> <p>Es una base de datos basada en grafos, compatible con ACID. Es accesible desde software escrito en otros lenguajes usando Cypher Query Language, a través de un punto HTTP transaccional.</p>
<p>Teorema CAP</p> <p>También llamado teorema de Brewer, señala que todo sistema distribuido no puede garantizar la vez que haya consistencia, disponibilidad y tolerancia a particiones (Consistency-Availability-Partition Tolerance).</p>	<p>Patrones de diseño:</p> <p>MongoDB es un buen ejemplo de base de datos NoSQL que aplica relaciones entre documentos para aprovechar la utilidad de los modelos relacionales, sin llegar a ser un relacional en toda regla.</p> <p>Relaciones que se crean en MongoDB:</p> <ul style="list-style-type: none">- Relaciones entre documentos uno a uno.- Relación uno a muchos con documentos embebidos.- Relación uno a muchos con documentos referidos.	<p>MongoDB</p> <p>Es una base de datos basada en documentos, la cual almacena los datos en esta estructura. El conjunto de documentos se denomina colecciones y el conjunto de estos conforman la base de datos.</p> <p>El formato de almacenamiento es BSON.</p>	

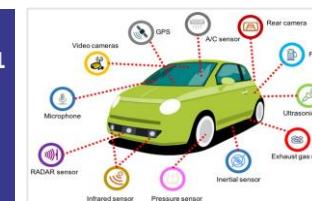
Datos Estructurados y No Estructurados

- ▶ Los datos en forma tabular son más comprensibles (están estructurados)
- ▶ **Semiestructurados y No estructurados (85%)**
- ▶
 - ❖ No se ajustan al modelo de bases de datos relacionales
 - ❖ No tienen un esquema
 - ❖ No utilizan SQL ni permiten los joins
 - ❖ No garantizan la propiedad ACID
 - ❖ Escalan horizontalmente

Fecha	Vendedor	Comunidad	Zona	Servicio	Precio
02/01/2015	Ramon Garrido Esteban	La Rioja	Norte	Formación	13.737,00 €
03/01/2015	Alberto Rivadesella	Canarias	Islas	Formación	9.895,00 €
04/01/2015	Manuela Gonzalez	Ceuta y Melilla	Sur	Consultoría	4.904,00 €
06/01/2015	Jaimie Montalbán de la Torre	Andalucía	Sur	Consultoría	18.807,00 €
07/01/2015	Soledad Gutierrez Jimenez	Asturias	Oeste	Formación	17.744,00 €
08/01/2015	Carmen Sobejano Garrido	Cataluña	Este	Formación	7.786,00 €
09/01/2015	Pedro Garcia de Urrea	Galicia	Oeste	Formación	7.595,00 €



```
order_id: order_0001
created_at: 2019/02/01
products:
  - product_01
  - product_02
  - product_03
```



Tipo de bases de datos

- ▶ Clave-valor simples
- ▶ Clave-valor sofisticadas
- ▶ Documentos
- ▶ Grafos

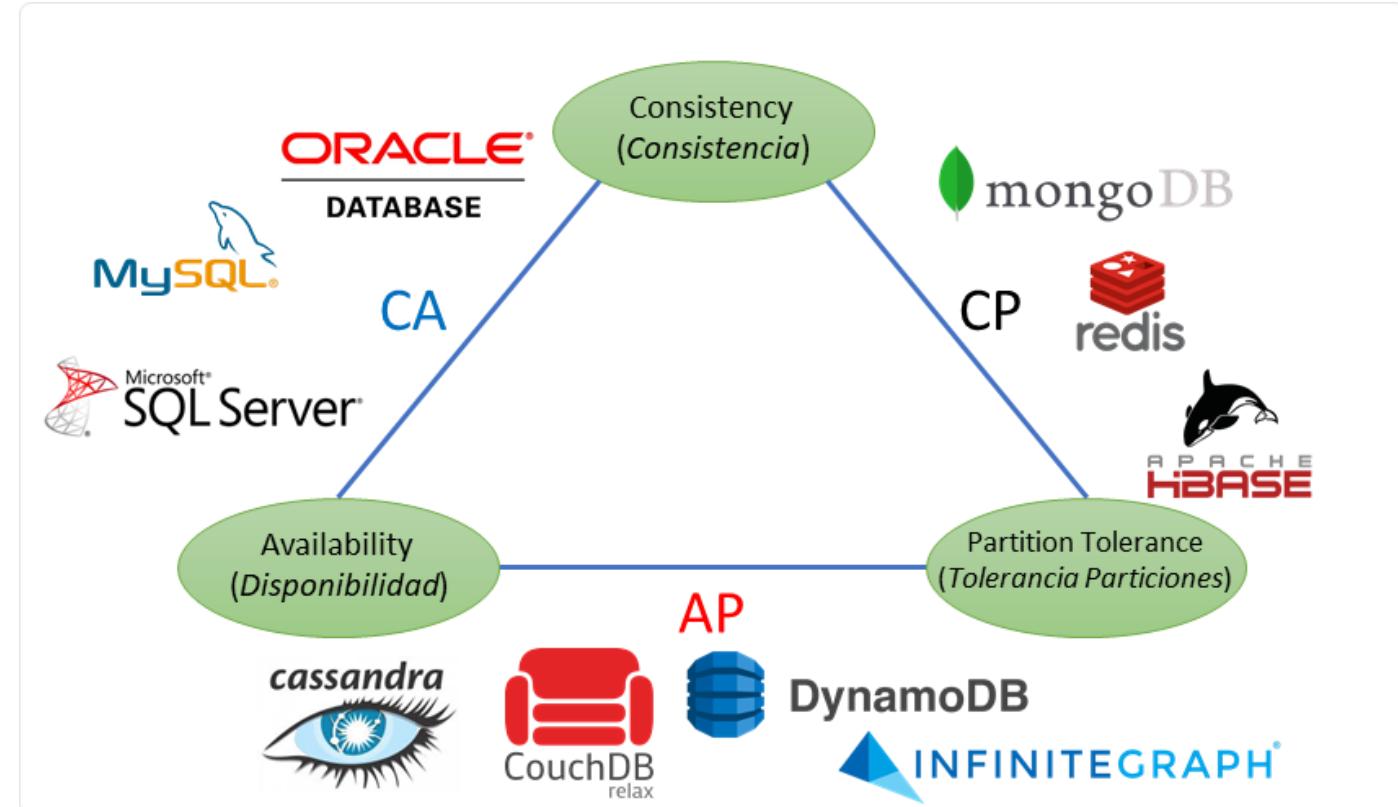


{ name: mongo, type: DB }



Teorema CAP

- ▶ Consistencia
- ▶ Disponibilidad
- ▶ Tolerancia a Particiones



SQL

- ▶ Interpretación de datos
- ▶ **Usos:**
 - ❖ Almacenar datos
 - ❖ Consultar datos
 - ❖ Consistencia
 - ❖ Normalización (3N)
- ▶ **Limitaciones:**
 - ❖ Capacidad del proceso
 - ❖ Necesidad de disco

OLTP (On-Line Transactional Processing):

- ❖ Es un sistema que gestiona aplicaciones orientadas a transacciones online.
- ❖ Es un sistema de modificación de base de datos online.

OLAP (On-Line Analytical Processing):

- ❖ Es un sistema en línea que reporta a consultas analíticas multidimensionales como informes, pronósticos, etc.
- ❖ Es un sistema de respuesta de consulta de base de datos online.

SQL - OLTP

- ▶ Bases de datos orientadas al procesamiento de transacciones. Una transacción genera un proceso atómico (que debe ser validado con un *commit*, o invalidado con un *rollback*).
- ▶ Puede involucrar operaciones de inserción, modificación y borrado de datos **CRUD**.
- ▶ El proceso transaccional es típico de las bases de datos operacionales.
 - El acceso a los datos está optimizado para tareas frecuentes de lectura y escritura. (Por ejemplo, la enorme cantidad de transacciones que tienen que soportar las BD de bancos o hipermercados diariamente).
 - Los datos se estructuran según el nivel aplicación (programa de gestión a medida, ERP o CRM implantado, sistema de información departamental...).
 - Los formatos de los datos no son necesariamente uniformes en los diferentes departamentos (es común la falta de compatibilidad y la existencia de islas de datos).
 - El historial de datos suele limitarse a los datos actuales o recientes.

SQL - OLAP

- ▶ Bases de datos orientadas al procesamiento analítico. Este análisis suele implicar la lectura de grandes cantidades de datos para llegar a extraer algún tipo de información útil: tendencias de ventas, patrones de comportamiento de los consumidores, elaboración de informes complejos... etc.
- ▶ Este sistema es típico de los *datamarts*.
 - El acceso a los datos suele ser de sólo lectura. La acción más común es la consulta, con **muy pocas inserciones, actualizaciones o eliminaciones**.
 - Los datos se estructuran según las áreas de negocio, y los formatos de los datos están integrados de manera uniforme en toda la organización.
 - El **historial de datos** es a largo plazo, normalmente de dos a cinco años.
 - Se suelen alimentar de información procedente de los sistemas operacionales existentes, mediante un proceso de extracción, transformación y carga (**ETL**).

SQL

- ▶ **Normalización:** eliminar redundancia de datos
- ▶ **SQL:** lenguaje de consulta estructurado, para recuperar la información
- ▶ **Modelo Entidad Relación** para diseñar
- ▶ **Formas normales:** son aplicadas a las tablas de una base de datos.
 - ❖ **1FN:** atributos atómicos (es atómico si los elementos son simples e indivisibles).
 - ✓ No debe existir variación en el número de columnas.
 - ✓ Esta forma normal elimina los valores repetidos dentro de una base de datos.
 - ❖ **2FN:** cada fila debe ser asociada a una única clave irrepetible.
 - ❖ **3FN:** sin dependencias transitivas dentro de una tabla.

SQL

1FN

Items				
Item	Size	Price	ManufacturerId	ManufacturerName
Just Do It T-Shirt	Women's Medium	29.99	1	Nike
Just Do It T-Shirt	Women's Large	34.99	1	Nike
Nebraska hat	One size fits all	9.99	2	Adidas

Items				
Item	Size	Price	ManufacturerId	ManufacturerName
Just Do It T-Shirt	Women's Medium	29.99	1	Nike

2FN PK

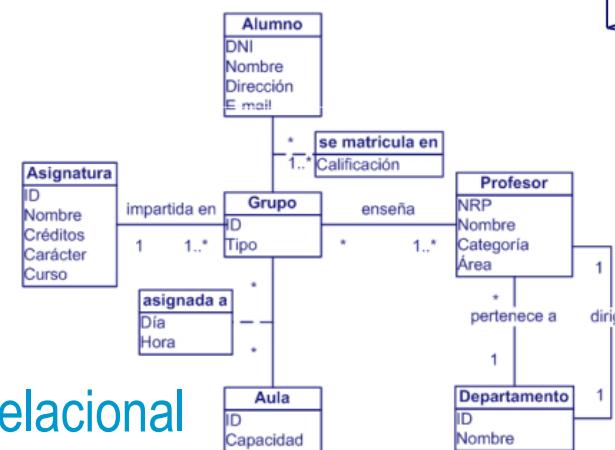
Items		
Item	ManufacturerId	ManufacturerName
Just Do It T-Shirt	1	Nike
Nebraska hat	2	Adidas

3FN

ItemPrices		
Item	Size	Price
Just Do It T-Shirt	Women's Medium	29.99
Just Do It T-Shirt	Women's Large	34.99
Nebraska hat	One size fits all	9.99

ItemsCategories		
Item	Size	CategoryId
Just Do It T-Shirt	Women's Medium	1
Just Do It T-Shirt	Women's Medium	2
Just Do It T-Shirt	Women's Medium	3
Just Do It T-Shirt	Women's Large	1
Just Do It T-Shirt	Womens' Large	2
Just Do It T-Shirt	Women's Large	3
Nebraska hat	One size fits all	1
Nebraska hat	One size fits all	2

Categories	
CategoryId	Category
1	Womens
2	T-shirt
3	Sportswear
4	Nebraska
5	Hats



Modelo Relacional

SQL

- ▶ Algebra relacional
- ▶ **RDBMS: Atómica, Consistente, Aislada, Durable**



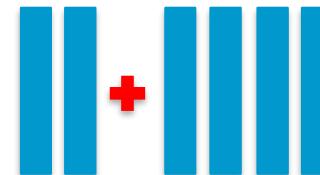
Proyección



Filtros



Group By/Aggregate



Join



Atomicity requires that each transaction be "all or nothing": if one part of the transaction fails, then the entire transaction fails, and the database state is left unchanged.

The consistency property ensures that any transaction will bring the database from one valid state to another.

The isolation property ensures that the concurrent execution of transactions results in a system state that would be obtained if transactions were executed serially, i.e., one after the other.

The durability property ensures that once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors.

SQL Joins Cheatsheet



INNER JOIN

```
SELECT <columns>
FROM A
[INNER] JOIN B
ON A.Key = B.Key;
```



LEFT JOIN

```
SELECT <columns>
FROM A
LEFT [OUTER] JOIN B
ON A.Key = B.Key;
```



RIGHT JOIN

```
SELECT <columns>
FROM A
RIGHT [OUTER] JOIN B
ON A.Key = B.Key;
```



FULL JOIN

```
SELECT <columns>
FROM A
FULL [OUTER] JOIN B
ON A.Key = B.Key;
```



© Satish Chandra Gupta, Creative Commons BY-NC-ND 4.0 International License.

ml4devs.com/sql-joins



Key	X	Y
1	x1	y1
2	x2	y2

Key	X	Y
1	x1	y1
2	x2	y2
NULL		

Key	X	Y
1	x1	y1
2	x2	y2
NULL		

Key	X	Y
1	x1	y1
2	x2	y2
3	x3	NULL
4	NULL	y3

scgupta

linkedin.com/in/scgupta

NoSQL

- ▶ No se ajustan al modelo de bases de datos relacionales.
- ▶ No tienen un esquema.
- ▶ No utilizan SQL ni permiten los joins.
- ▶ No garantizan la propiedad ACID.
- ▶ Escalan horizontalmente.
- ▶ Usan la memoria principal del computador.
- ▶ Resuelven el problema de los altos volúmenes de información y la inmensa cantidad de consultas y transacciones diarias, pero, **no son relacionales**.

Componentes de BD – Métodos ETL y ELT

Database Components

> Interface Language / API (definition, manipulation, query, control)

- ① Parses and validates a request vis-à-vis metadata.
- ② Using indexes, creates an efficient execution plan.
- ③ Reads or updates storage.
- ④ Updates metadata and indexes.
- ⑤ Computes and returns results.

Metadata



Query Processor



Storage

ml4devs.com/datastores



In-built **operations** in the query language or API determine **ease-of-use**.

Index design determine **performance** of operations.



Indexes

© Satish Chandra Gupta

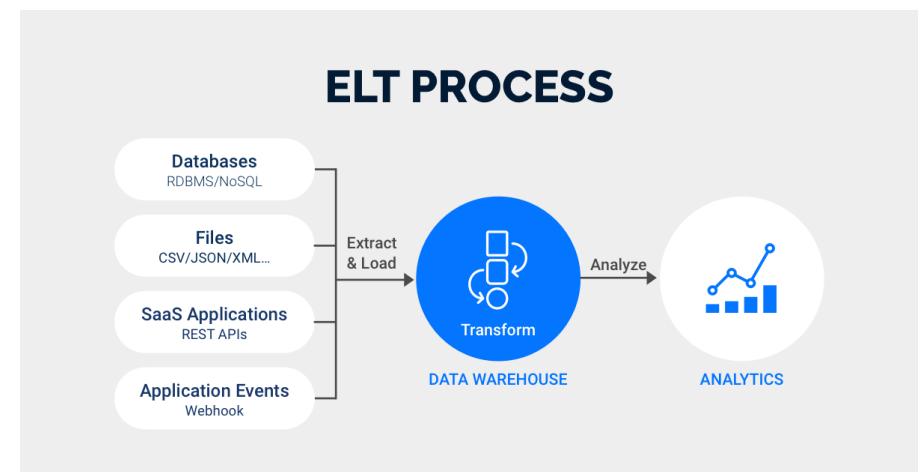
CC BY-NC-ND 4.0 International License
creativecommons.org/licenses/by-nc-nd/4.0/

scgupta.me
twitter.com/scgupta
linkedin.com/in/scgupta

ETL PROCESS

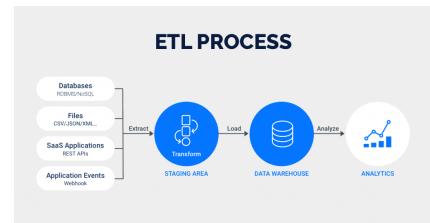


ELT PROCESS



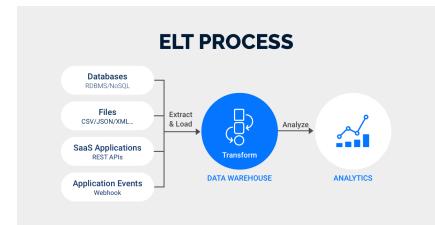
Método ETL

- ▶ Extraer, transformar y cargar (ETL) es una **metodología de integración de datos** que extrae datos sin procesar de fuentes, transforma los datos en un servidor de procesamiento secundario y luego carga los datos en una base de datos de destino.
- ▶ ETL se utiliza cuando los datos deben transformarse para ajustarse al régimen de datos de una base de datos de destino. El método surgió en la década de 1970 y sigue prevaleciendo entre las bases de datos locales que poseen memoria y capacidad de procesamiento limitadas.
- ▶ Considere un ejemplo de ETL en acción. Los almacenes de datos de procesamiento analítico en línea (OLAP) solo aceptan estructuras de datos relacionales basadas en SQL.
- ▶ Con este tipo de almacén de datos, un protocolo como ETL garantiza el cumplimiento al enrutar los datos extraídos a un servidor de procesamiento y luego transformar los datos no conformes en datos basados en SQL.
- ▶ Los datos extraídos solo se mueven del servidor de procesamiento al almacén de datos una vez que se han transformado con éxito.

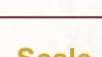
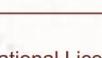


Método ELT

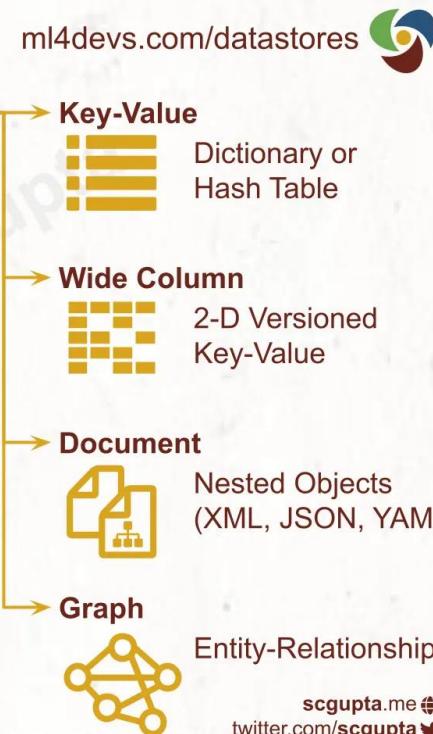
- ▶ A diferencia de ETL, extraer, cargar y transformar (ELT) no requiere que se realicen transformaciones de datos antes del proceso de carga. ELT carga datos sin procesar directamente en un almacén de datos de destino, en lugar de moverlos a un servidor de procesamiento para su transformación.
- ▶ Con la canalización de datos ELT, la limpieza, el enriquecimiento y la transformación de datos ocurren dentro del propio almacén de datos. Los datos sin procesar se almacenan indefinidamente en el almacén de datos, lo que permite múltiples transformaciones.
- ▶ ELT es un desarrollo relativamente nuevo, gracias en parte a los almacenes de datos escalables basados en la nube.
- ▶ Los almacenes de datos en la nube como **Snowflake**, **Amazon Redshift**, **Google BigQuery** y **Microsoft Azure** tienen la infraestructura digital, en términos de capacidad de almacenamiento y procesamiento, para facilitar los repositorios de datos sin procesar y las transformaciones en la aplicación.
- ▶ Aunque la canalización de datos ELT no se usa universalmente, el método se está volviendo más popular a medida que las empresas adoptan la infraestructura en la nube.



SQL vs. NoSQL: Comparison

SQL		NoSQL	
	Relational		Non-relational
Structured tables		Semi-structured	
Strict schema		Dynamic schema	
ACID		Mostly BASE, few ACID	
Strong		Eventual to Strong	
Consistency prioritized		Basic Availability	
Vertically by upgrading hardware		Horizontally by data partitioning	

© Satish Chandra Gupta

 CC BY-NC-ND 4.0 International License
creativecommons.org/licenses/by-nc-nd/4.0/


SQL vs. NoSQL: Cheatsheet for AWS, Azure, and Google Cloud

ml4devs.com/datastores	AWS	Azure	Cloud	Cloud Agnostic
Relational	RDS, Aurora	Azure SQL Database	Cloud SQL, Cloud Spanner	SQL Server, Oracle, DB2, MySQL, PostgreSQL
Columnar	RedShift	Azure Synapse	BigQuery	Snowflake, ClickHouse, Druid, Pinot, Databricks
Key-Value	DynamoDB	Cosmos DB	BigTable	Redis, ScyllaDB, Ignite
In-memory	ElastiCache	Azure Cache for Redis	Memory-store	Redis, Memcached, Hazelcast, Ignite
Wide Column	Keyspaces	Cosmos DB	BigTable	HBase, Cassandra, ScyllaDB
Time Series	Timestamp	Cosmos DB	BigTable, BigQuery	TimescaleDB, OpenTSDB, InfluxDB, ScyllaDB
Immutable Ledger	Quantum Ledger Database (QLDB)	Azure SQL Database Ledger	X	Hyperledger Fabric
Geospatial	Keyspaces	Cosmos DB	BigTable, BigQuery	Solr, PostGIS, MongoDB (GeoJSON)
Graph	Neptune	Cosmos DB	JanusGraph + BigTable	OrientDB, Neo4J, Giraph
Document	Document DB	Cosmos DB	Firestore	MongoDB, Couchbase, Solr
Text Search	Open-Search, Cloud-Search	Cognitive Search	Search APIs on Datastores	Elastic-Search, Solr, Elassandra
Blob	S3	Blob Storage	Cloud Storage	HDFS, MinIO

© Satish Chandra Gupta
 CC BY-NC-ND 4.0 International License
creativecommons.org/licenses/by-nc-nd/4.0/

scgupta.me
[@scgupta](https://twitter.com/scgupta)
[LinkedIn](https://linkedin.com/in/scgupta)

Tipología BD NoSQL



{ name: mongo, type: DB }

Documentos

Suelen almacenar documentos JSON, XML y BSON. Son similares a las BBDD de valores clave, pero en este caso, un valor es un solo documento que almacena todos los datos relacionados con una clave específica. Los campos populares del documento pueden indexarse para proporcionar una recuperación rápida sin conocer la clave. Cada documento puede tener la misma estructura o una estructura diferente.

Wide Column

Almacenan datos en tablas con filas y columnas similares a las BBDD relacionalles, pero los nombres y los formatos de las columnas pueden variar de fila a fila en la tabla. Las columnas de columnas anchas agrupan columnas de datos relacionados juntos. Una consulta puede recuperar datos relacionados en una sola operación porque sólo se recuperan las columnas asociadas con la consulta.



Grafos

Utiliza estructuras de grafos para almacenar, correlacionar y consultar relaciones. Proporcionan una adyacencia libre de índice, de modo que los elementos adyacentes se unen entre sí sin usar un índice.



Clave Valor



Hacen hincapié en la simplicidad y son útiles para acelerar una aplicación que admite el procesamiento de alta velocidad de lectura y escritura de datos no transaccionales.

Los valores almacenados pueden ser cualquier tipo de objeto binario (texto, video, documento JSON, etc.) y se accede a través de una clave. La aplicación tiene control total sobre lo que se almacena en el valor, convirtiéndolo en el modelo NoSQL más flexible.

Los datos se comparten y se replican en un clúster para obtener escalabilidad y disponibilidad. Por esta razón, las bases de datos de valores clave a menudo no admiten transacciones. Sin embargo, son muy eficaces en aplicaciones de escala que se ocupan de datos de alta velocidad y no transaccionales.

En una BBDD, los datos estarían en diferentes filas almacenadas en diferentes lugares del disco, requiriendo múltiples operaciones de disco para su recuperación.

Cassandra o HBase, permiten realizar consultas en grandes conjuntos de datos y almacenan los datos en columnas, en lugar de filas.

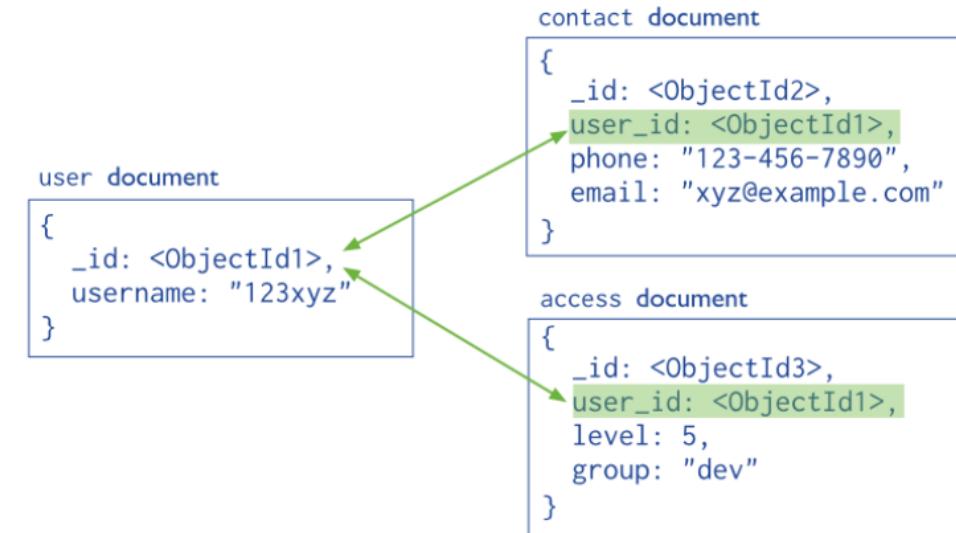
MongoDB



- ▶ Fue creado por **10gen** en el 2009 (ahora mongodb).
- ▶ Almacena documentos semiestructurados tipo JSON.
- ▶ Orientado a desarrollo web (JSON con JavaScript).
- ▶ Filosofía de almacenar los datos de manera cercana al paradigma OOP (**no es relacional**).
- ▶ Optimizada para operaciones OLTP sobre un solo documento.
- ▶ Esquema “Flexible” y semi estructurado, facilita las listas y anidaciones.
- ▶ Escalable a billones de documentos, tamaño óptimo de los documentos 16MB.
- ▶ Open source.
- ▶ Los despliegues en vertical y en modo clúster son eficientes.
- ▶ Gestión variada: **Atlas, On-Premise**.
- ▶ Entorno visual: **MongoDB Compass, MongoDB Studio, Robot3T, otros...**

Modelado con MongoDB

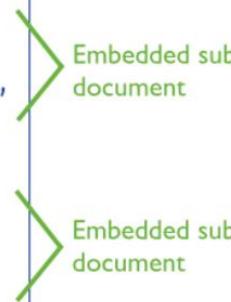
- ▶ Modelo normalizado
- ▶ No tiene operación Join
- ▶ Actualización de datos eficiente
- ▶ Consulta de datos algo ineficiente



Modelado con MongoDB

- ▶ Desnormalizado
- ▶ Embebido
- ▶ Actualización de datos ineficiente
- ▶ Consulta de datos eficiente

```
{  
  _id: <ObjectId1>,  
  username: "123xyz",  
  contact: {  
    phone: "123-456-7890",  
    email: "xyz@example.com"  
  },  
  access: {  
    level: 5,  
    group: "dev"  
  }  
}
```



Embedded sub-document

Embedded sub-document

Preparando el entorno de MongoDB



MongoDB Atlas

Cloud-hosted MongoDB service on AWS, Azure and Google Cloud. Deploy, operate, and scale a MongoDB database in just a few clicks

[Sign up with Google](#)
or
[Start Free](#)

Nominate a project for the MongoDB Innovation Awards today!

Choose which type of deployment is best for you

Cloud MongoDB as a service

On-premises MongoDB locally

Tools Boost productivity

Mobile & Edge Realm Datastore

MongoDB Enterprise Server

MongoDB Community Server

Preparando el entorno de MongoDB



PER1825-LAB-MCB > PER1825-LABMCB

Clusters

Find a cluster...

SANDBOX

Cluster0
Version 4.4.4

CONNECT **METRICS** **COLLECTIONS** **...**

CLUSTER TIER
M0 Sandbox (General)

REGION
AWS / Ireland (eu-west-1)

TYPE
Replica Set - 3 nodes

LINKED REALM APP
None Linked

Operations R: 0 W: 0
100.0/s
Last 6 Hours

Logical Size 0.0 B
512.0 MB max
0.0 B
Last 6 Hours

Connections 0
500 max
0
Last 6 Hours

Enhance Your Experience
For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!

Upgrade

A screenshot of the MongoDB Cloud interface. The main header shows the path 'PER1825-LAB-MCB > PER1825-LABMCB'. Below this, a large heading says 'Clusters'. A search bar is present. On the left, there's a sidebar with cluster details: 'Cluster0' (version 4.4.4), 'CONNECT', 'METRICS', 'COLLECTIONS', and a three-dot menu. Below this are sections for 'CLUSTER TIER' (M0 Sandbox (General)), 'REGION' (AWS / Ireland (eu-west-1)), 'TYPE' (Replica Set - 3 nodes), and 'LINKED REALM APP' (None Linked). The main area contains four cards: 'Operations R: 0 W: 0' (100.0/s, Last 6 Hours), 'Logical Size 0.0 B' (512.0 MB max, 0.0 B, Last 6 Hours), 'Connections 0' (500 max, 0, Last 6 Hours), and an 'Enhance Your Experience' section with a 'Upgrade' button.

Muchas gracias por tu
atención