

Propósito general de las Bases de Datos

Las **Bases de Datos** surgen de la idea de tener un espacio dónde poder almacenar de una forma mucho más eficiente toda la información de nuestros proyectos.



Anteriormente este almacenamiento era en papel, y aunque a la fecha algunas empresas por temas de regulación lo siguen haciendo en parte así, el tener una **Base de Datos** ha permitido tener mucho más control de la información.



Historia de las Bases de Datos

1950-1960: Maquinas tabuladoras, tarjetas perforadas y cintas magnéticas.

1960-1979: Modelos jerárquicos, discos duros, modelo de data relacional, transacciones en tiempo real.

Un disco duro tiene **información persistente**, o sea que perdura en el tiempo.

1970-1980: SQL, Sistemas SQL comerciales, bases de datos paralelas y distribuidas, bases de datos orientadas a objetos.

SQL es un **estándar**, la mayoría de los comandos básicos, en cualquier tipo de datos que sea SQL deben funcionar (MariaDB, MySQL, etc).

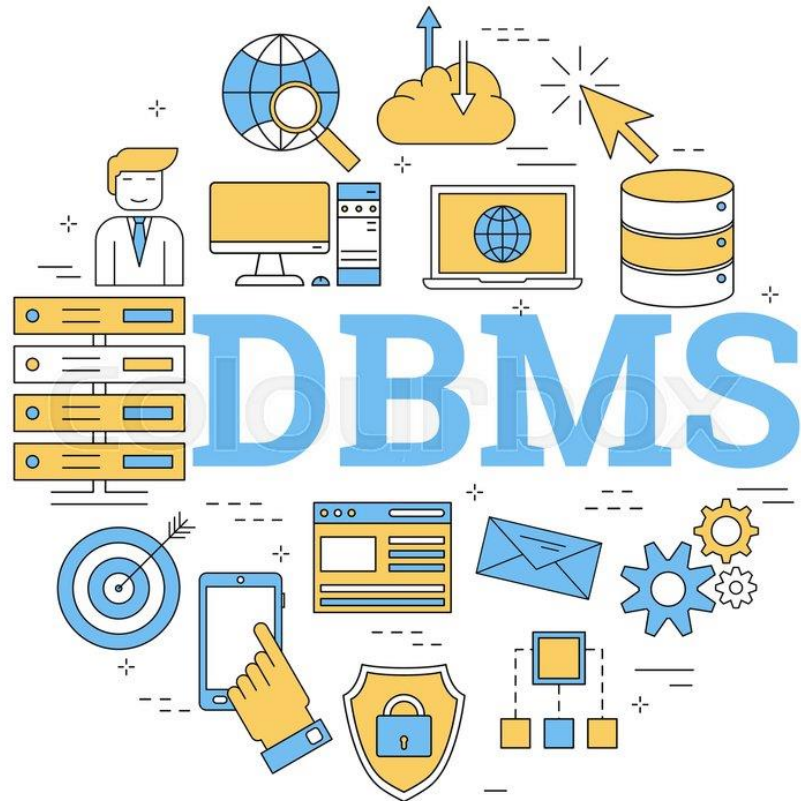
1980-1990: Data mining, data warehouse, e-commerce.

2000-Actualidad: XML, administración automatizada, analytics, big data, No SQL, InMemory, Scale Out, Systems of Engagement.

Los **datos no son información**. Solo en el momento que creamos un reporte que contenga ciertos datos, éstos se convierten en información.

DBMS = Data Base Management System ó

SGBD = Sistemas de Gestión de Bases de Datos.



Tipos de Bases de Datos y sus aplicaciones en la industria



Las **Bases de Datos** se pueden dividir en:

1. **Bases de Datos Relacionales**
2. **Bases de Datos no Relacionales**

Bases de Datos Relacionales *empresariales* (más importantes)

- **DB2**
- **SQL Server**
- **Oracle**



Algunas **Bases de Datos Relacionales** *comunes*:

- **MariaDB**: Es una distribución de Bases de Datos que deriva de MySQL.
- **PostgreSQL**: Esta es una Base de Datos comunitaria pero tiene una versión enterprise que tiene soporte.

Algunas **Bases de Datos No Relacionales** *comunes*:

- **Redis**: Una Base de Datos que en la actualidad se trabaja mucho.
- **neo4j**: Es una Base de Datos basada en nodos. Está centrada en grafos que nos va a permitir encontrar relaciones entre objetos. Muy común en eCommerce.
- **Cassandra**: Es una Base de Datos muy importante del proyecto Apache. Trabaja con grandes volúmenes de datos.
- **MongoDB**: Es una Base de Datos en NoSQL que se basa en trabajar en varias instancias.

Industria



Educación



Entidades Bancarias



Entidades con movimiento Financiero



Supermercados (Retail)



Logística



Recursos Humanos



En todas las industrias



¡¡¡TODAS!!!

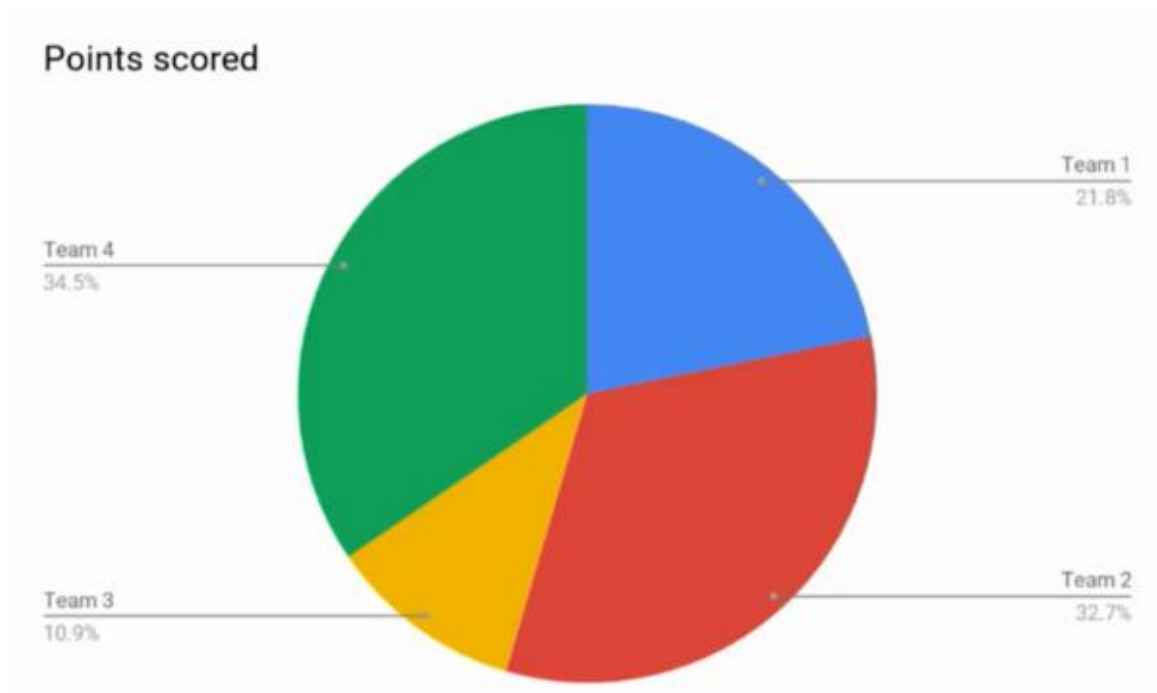
Visión general de los datos

¿Qué es un dato?

Un dato es algo que nos va a permitir describir un objeto. Ese objeto global lo vamos a poder llamar “Entidad”. Una entidad puede estar llena de datos.

Qué valor tienen los datos?

Los datos se convierten en información.

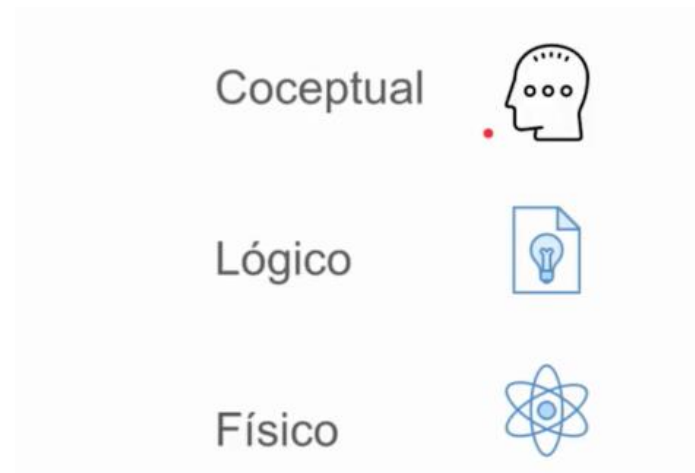


Existen **3 niveles de Abstracción** en las **Bases de Datos**:

Conceptual: Se tiene que empezar a modelar una Base de Datos dependiendo de lo que se quiere hacer basado en los conceptos de “entidad” y “relación”.

Lógico: El diagrama lógico nos va a resolver ciertas dudas de consistencia, para evitar crear loops o evitar que tenga cosas que no tengan sentido en nuestro proyecto.

Físico: Es finalmente cómo lo va a ver la Base de Datos.



Tipos de Datos

Igual que en cualquier lenguaje de programación, existen **variables** en las **Bases de Datos**:

Caracteres: Pueden ser desde letras hasta caracteres especiales.

Numérico: Del 0 al 9 pero con una longitud especial.

Varchar: Caracteres con un formato más variable.

Imagen

Fecha: Generalmente van acompañadas de una hora.

Moneda: esto facilita todo si se trabaja con diferentes denominaciones.

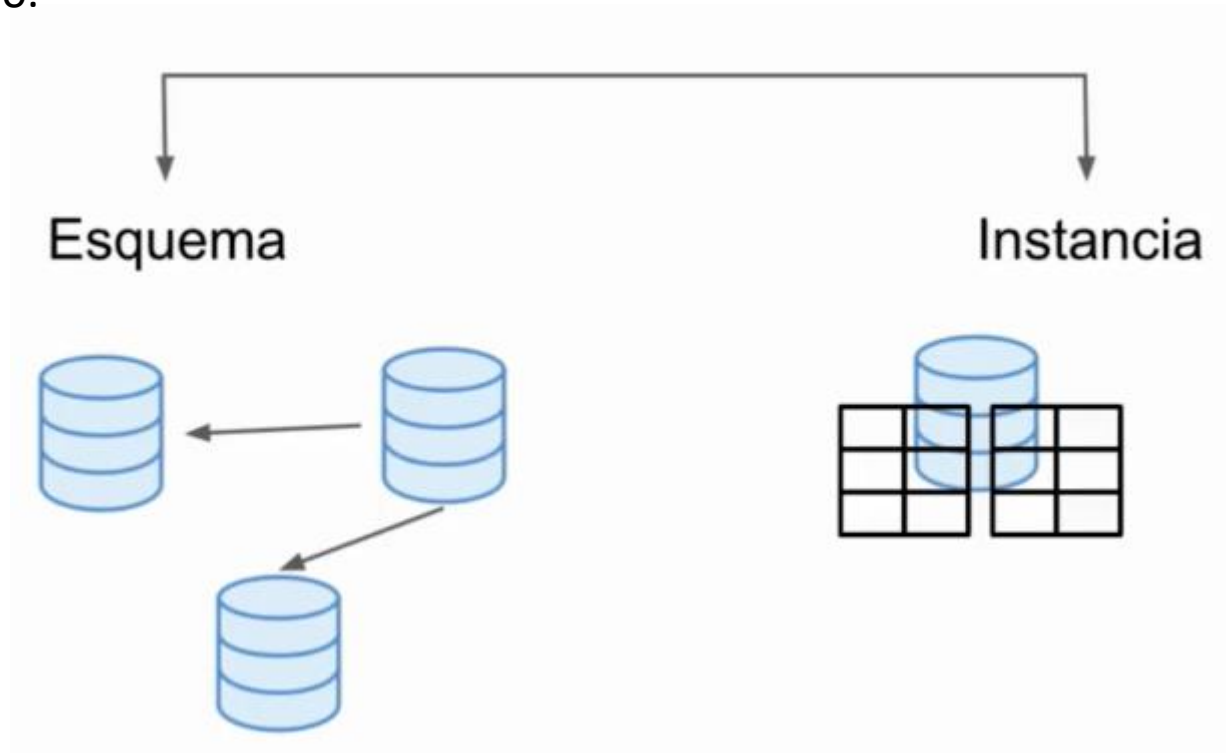
Texto: Variables que tienen mayor tamaño que un char o que un varchar.

Bit: Se puede trabajar con 1 y 0 o también con verdadero y falso.

Decimal

Esquema = Es la estructura lógica que va a tener una Base de Datos.

Instancia = Contenido de partículas que tiene una Base de Datos en un instante de tiempo.

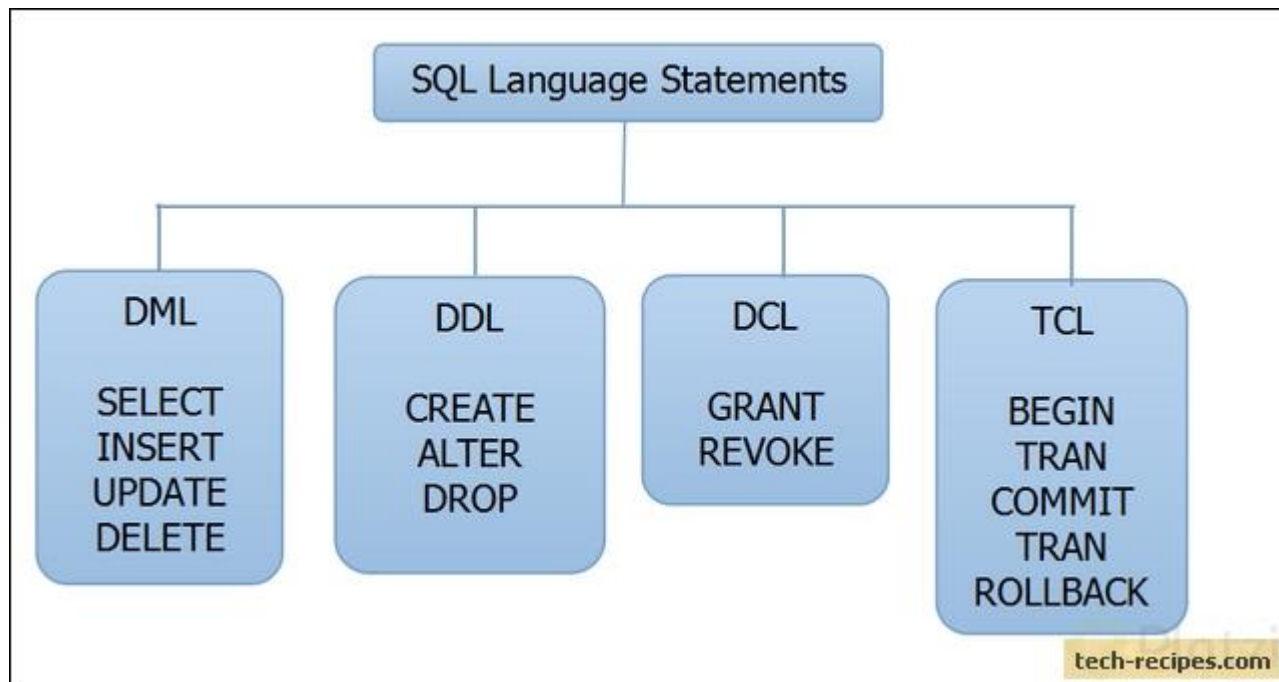


Existen 3 cosas para poder hacer la descripción de una **Base de Datos**:

DML = Data Manipulation Language o Lenguaje de Manipulación de Datos.

DDL = Data Definition Language o Lenguaje de Definición de Datos.

SQL = Structured Query Language o Lenguaje de Consulta Estructurada.



| ID_Cuenta | ID_Usuario | Saldo |
|-----------|------------|--------|
| 12 | 234 | 10.000 |
| 57 | 1 | 1.004 |

| ID_Usuario | Nombre | Direccion | Fecha_Nacimiento |
|------------|--------------|-----------------|------------------|
| 1 | Juan Gomez | Cra. 23 # 7 | 12/10/1975 |
| 234 | Maria Duarte | Avenida Reforma | 23/07/1998 |

El lenguaje para hacer consultas y manipular los datos es SQL

Otros tipos de Bases de Datos:

1. Bases de Datos Relacionales
2. Basadas en Objetos Relacionales
3. XML
4. NoSQL
5. In-Memory

Diferentes tipos de Bases de Datos

Características de Bases de Datos SQL:

- Es un lenguaje estructurado.
- Tiene un esquema de tablas.
- Tiene integración con otros tipos de archivos.
- Tiene indexación por medio de árboles.

Características de Bases de Datos NoSQL:

- Se puede trabajar con un lenguaje estructurado o con uno no estructurado.
- Tiene diferente tipo de indexación. Se utiliza normalmente Json.
- Tiene un crecimiento horizontal.

Características de Bases de Datos Analíticas y de Bigdata:

- Son de lenguaje no estructurado.
- Tiene integración de muchos sistemas.
- Tiene integración también a sistemas tradicionales y sistemas de engagement.
- Principio “divide y vencerás”
- Se basa en esquemas Scale Out.
- Crecimiento horizontal.

Características de Bases de Datos basadas en aceleración:

- Normalmente basadas in Memory.
- Uso de aceleradores como GPU, flash cards, FPGAs.
- Tienen estructuras diferentes, por ejemplo, basadas en nodos.
- Uso frecuente de ambientes empresariales productivos y de datawarehouse.

SQL

Lenguaje estructurado

Esquema de tablas

Integración con otros tipos de archivos

Indexación por medio de árboles



No SQL

Lenguaje estructurado

Lenguaje no estructurado

Diferente tipo de indexación, normalmente JSON

Crecimiento horizontal



mongoDB®



cassandra

Analíticas y de bigdata

Lenguaje no estructurado

Integración de muchos sistemas

Sistemas tradicionales y de engagement

Principio divide y vencerás

Basado en esquemas Scale Out



Basadas en aceleración

Normalmente basadas In Memory

Uso de aceleradores como GPU, Flash cards, FPGAs

Estructuras diferentes por ejemplo basadas en nodos

Uso frecuente en ambientes empresariales productivos y de datawarehouse



redis



neo4j

kinetica

Formas de usos en las bases de datos:

1. **On premise open source**, bases de datos de formato empresarial u opensource instalada en nuestra maquina sin una gran infraestructura.
2. Licenciamiento por **cores** o **sockets**, se paga dependiendo de ciertas características; como el hardware en el que va a correr.
3. Licenciamiento **modular**, se paga por funcionalidades o módulos para necesidades diferentes.
4. Pago por uso a través de **SAAS**(Software As A Service) o **PAAS**(Platform As A Service). Es como adquirir una renta y pagar por usar una base de datos.
5. Suscripción de **nodos de computo**, funciona para plataformas como Hadoop el cual no es centralizado y trabaja de forma distribuida, se paga por nodo utilizado.

Investigación

¿Hadoop y Blockchain podrían cambiar una Base de Datos tradicional?

Recordando para practicar.

- 1. Diagrama Entidad – Relación**
- 2. Entidades** (Fuertes y Débiles)
- 3. Atributos** (Simples, Compuestas, Derivadas y Multivaluados)
- 4. Relación**
- 5. Cardinalidad (1:1 – 1:M – M:M)**
- 6. Modelo Relacional**
- 7. Llaves en BD** (Primaria, Foránea y Candidata)
- 8. Normalización (1FN, 2FN, 3FN)**

Metodología básica de 9 pasos:

1. Identificar Cuáles son las Entidades Resuelven Nuestro Problema
2. Identificación de las Relaciones de las Entidades.
3. Entidades y Relaciones
4. Asignar Atributos a las Entidades.
5. Generar un Diagrama conceptual,
6. Generar un Diagrama lógico,
7. Identificar nuevos atributos que generan nuestras entidades débiles.
8. Construir el Diagrama del Modelo Físico
9. Pasar al estándar de la base de datos (SQL) – Código SQL

Caso Práctico

1. Identificar un problema a resolver
2. Realizar el diagrama de contexto
3. Identificar actores y roles
4. Definir los requerimientos funcionales y No funcionales
5. Realizar el diagrama Entidad - Relación
6. Realizar el modelo Conceptual
7. Realizar el modelo Lógico
8. Realizar el modelo Físico

