

BIG DATA & IA

BOOTCAMP

# BBDD y Modelado



# Sistema gestor de BBDD (DBMS)

Software especializado que permite crear, gestionar y manipular bases de datos de manera eficiente. Actúa como un administrador que organiza y facilita el acceso a la información almacenada. Algunos de los más usados son:

- MySQL
- Microsoft SQL Server
- Oracle
- PostgreSQL



# Bases de datos

Una base de datos es un conjunto estructurado de piezas de información, que permite acceder, actualizar y transformar los datos de forma eficiente y segura.

- Organiza los datos siempre siguiendo la misma estructura
- Facilita la búsqueda y análisis rápido
- Permite que varias persona o sistemas accedan de forma segura a la información sin desorden

En una base de datos relacional (SQL), la información se organiza en tablas, como si fuera una hoja de cálculo.

Veamos cómo sería la base de datos de una biblioteca si organizamos la información relacionada a libros y autores.



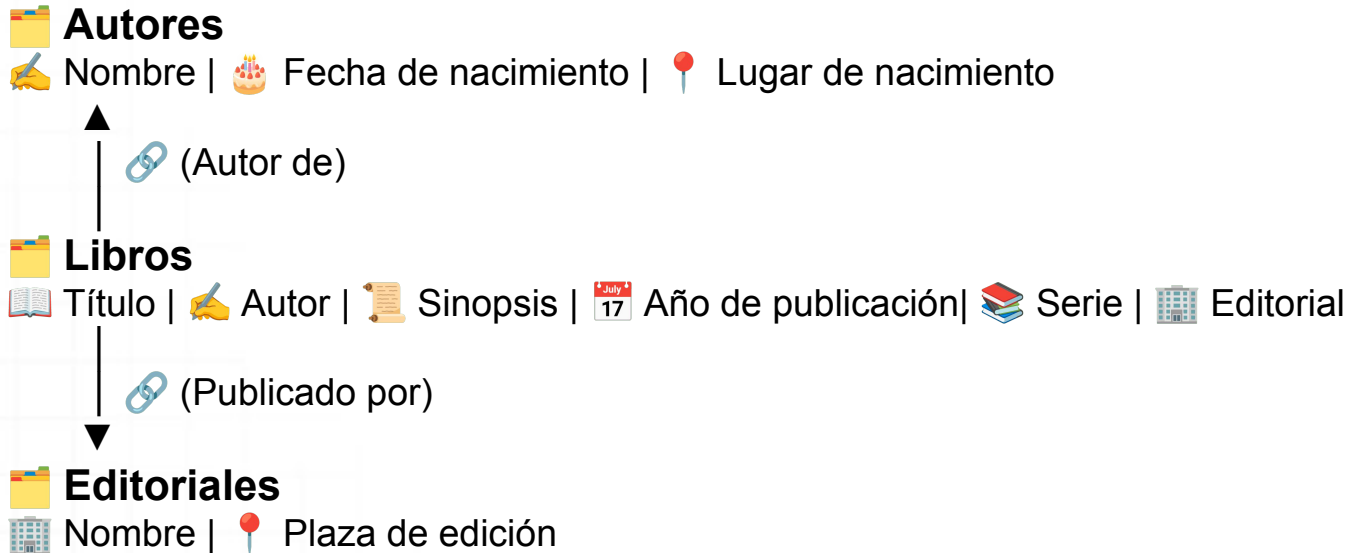
# Bases de datos

Las tablas de una base de datos se relacionan entre sí, lo que permite almacenar la información de cada entidad por separado y complementarla con datos de otras entidades solo cuando sea necesario.

Este aislamiento mejora el rendimiento, ya que evita almacenar datos repetidos, optimiza el uso del espacio y reduce el tiempo de respuesta en las consultas. A su vez, esto se traduce en un menor coste tanto en procesamiento como en almacenamiento.



# Modelo Entidad - Relación





# Modelo Entidad - Relación

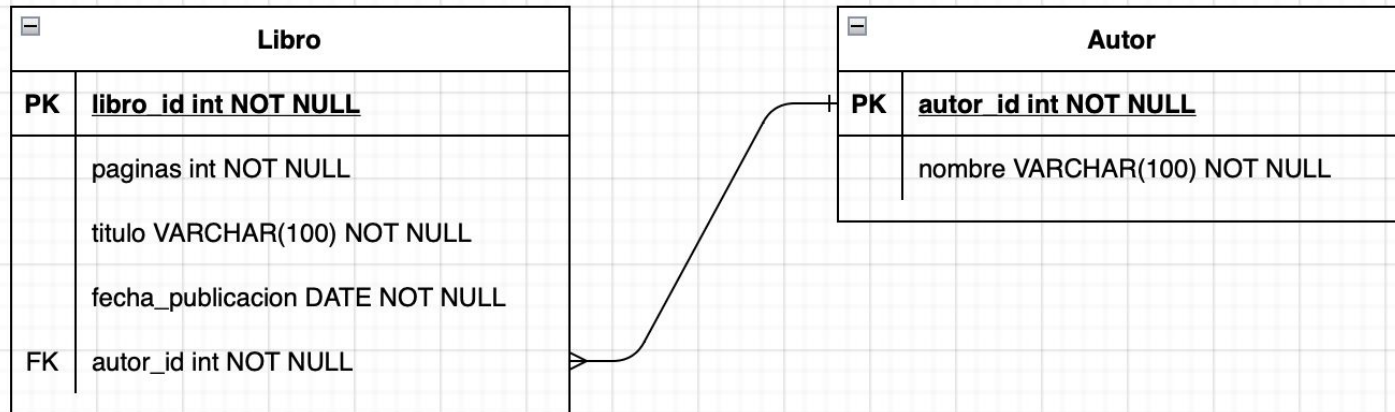
Un Modelo Entidad - Relación (ER) es una representación gráfica y conceptual de la estructura de una base de datos. Este modelo utiliza entidades (objetos o conceptos) y sus relaciones para describir la información y cómo se relaciona dentro de un sistema.

Se compone de:

- Entidades: Representan objetos o conceptos del mundo real, como "Cliente" o "Producto".
- Atributos: Características o propiedades de las entidades, como "Nombre" o "Edad".
- Relaciones: Conexiones entre entidades que muestran cómo se relacionan entre sí. Por ejemplo, un "Cliente" realiza "Compras".



# Modelo Entidad - Relación





# Modelo Entidad - Relación

1. **Entidad** 📁  
Objeto o concepto del mundo real sobre el que se almacena información.  
*Ejemplo:* Libro, Autor, Editorial.
2. **Atributo** 🏷️  
Propiedad o característica que describe a una entidad.  
*Ejemplo:* “Título” y “Año de publicación” son atributos de la entidad Libro.
3. **Relación** 🔗  
Asociación entre dos o más entidades.  
*Ejemplo:* “Un autor escribe un libro” o “Una editorial publica un libro”.
4. **Clave primaria (PK)** 🔑  
Atributo que identifica de manera única cada registro dentro de una entidad.  
*Ejemplo:* id\_libro en la entidad Libro.





# Modelo Entidad - Relación

## 4. Clave foránea (FK)

Atributo en una tabla que hace referencia a la clave primaria de otra, estableciendo la relación entre ambas.

*Ejemplo:* `id_autor` en la entidad Libro, que enlaza con la PK de Autor.

## 5. Cardinalidad

Indica el número de elementos que pueden participar en una relación:

- **1:1** → Una entidad se relaciona con una sola de la otra.
- **1:N** → Una entidad se relaciona con varias de la otra.
- **N:M** → Varias entidades de un tipo se relacionan con varias del otro.

## 6. Normalización

Proceso de organizar los datos para eliminar redundancias y mejorar la integridad.

*Ejemplo:* Dividir una entidad “Libros y Autores” en dos entidades separadas “Libros” y “Autores” relacionadas por una clave foránea.

## 7. Integridad referencial

Garantiza que las relaciones entre tablas sean coherentes, de forma que las claves foráneas siempre apunten a registros válidos de la tabla referenciada.



# Formas normales

**La normalización de bases de datos** es un conjunto de reglas y principios de diseño que permiten organizar los datos de manera eficiente, eliminando redundancias y mejorando la **integridad** y consistencia de la información.

Existen 3 reglas o formas normales



# Primera forma normal (1FN)

Cada registro debe ser único y se puede identificar con su clave primaria.

El valor de un campo es indivisible (atómico).

No debe haber duplicados

user_id	name	country	province	town	street	number
1	Alex López	España	Madrid	Getafe	Dulcinea	2
2	Celia Pérez	España	Alicante	Elche	Goya	5
3	Juan Martín	España	Sevilla	Utrera	Gran Vía	4
4	Laura García	España	Toledo	Yuncos	Mozart	45



user_id	name	surname	country	province	town	street	number
1	Alex	López	España	Madrid	Getafe	Dulcinea	2
2	Celia	Pérez	España	Alicante	Elche	Goya	5
3	Juan	Martín	España	Sevilla	Utrera	Gran Vía	4
4	Laura	García	España	Toledo	Yuncos	Mozart	45



# Segunda forma normal (2FN)

Debe cumplir con la primera forma normal 1FN

Elimina dependencias parciales de la clave primaria. Todos los atributos no claves, depende de la clave primaria

user_id	name	surname	country	province	town	street	number	reservation_date
1	Alex	López	España	Madrid	Getafe	Dulcinea	2	2023-05-19
2	Celia	Pérez	España	Alicante	Elche	Goya	5	2022-08-15
3	Juan	Martín	España	Sevilla	Utrera	Gran Vía	4	2021-12-06
4	Laura	García	España	Toledo	Yuncos	Mozart	45	2023-10-12



user_id	name	surname	country	province	town	street	number
1	Alex	López	España	Madrid	Getafe	Dulcinea	2
2	Celia	Pérez	España	Alicante	Elche	Goya	5
3	Juan	Martín	España	Sevilla	Utrera	Gran Vía	4
4	Laura	García	España	Toledo	Yuncos	Mozart	45

reservation_id	reservation_date	user_id
1001	2023-05-19	1
1002	2022-08-15	2
1003	2021-12-06	3
1004	2023-10-12	4



# Tercera forma normal (3FN)

Debe cumplir con la segunda forma normal 2FN

No tiene dependencia parcial transitiva, es decir, los atributos no tienen dependencia entre sí

user_id	name	surname	country	province	town	street	number
1	Alex	López	España	Madrid	Getafe	Dulcinea	2
2	Celia	Pérez	España	Alicante	Elche	Goya	5
3	Juan	Martín	España	Sevilla	Utrera	Gran Vía	4
4	Laura	García	España	Toledo	Yuncos	Mozart	45

user_id	name	street_id	number
1	Alex López	2001	2
2	Celia Pérez	2002	5
3	Juan Martín	2003	4
4	Laura García	2004	45

country_id	name
1	España
2	Francia

province_id	name	country_id
1	Madrid	1
2	Alicante	1
3	Sevilla	1
4	Toledo	1

town_id	name	province_id
101	Getafe	1
102	Elche	2
103	Utrera	3
104	Yuncos	4

street_id	name	town_id
2001	Dulcinea	101
2002	Goya	102
2003	Gran Vía	103
2004	Mozart	104



# Integridad referencial

La integridad referencial **mantiene la consistencia de los datos** entre diferentes tablas de una base de datos. Evita la pérdida de conexiones lógicas y garantiza que las relaciones entre tablas sean coherentes y válidas.

En el contexto de **Libros y Autores**:

- Garantiza que **no se pueda hacer referencia a un autor que no existe** en la tabla de Libros.
- Si intentamos **eliminar un autor** que tiene libros asociados, la integridad referencial:
  - **Impide la eliminación**, o
  - Obliga a **manejar los datos de forma consistente**, por ejemplo, eliminando también los libros asociados o reasignándolos.

# keep coding

