

# Base de datos

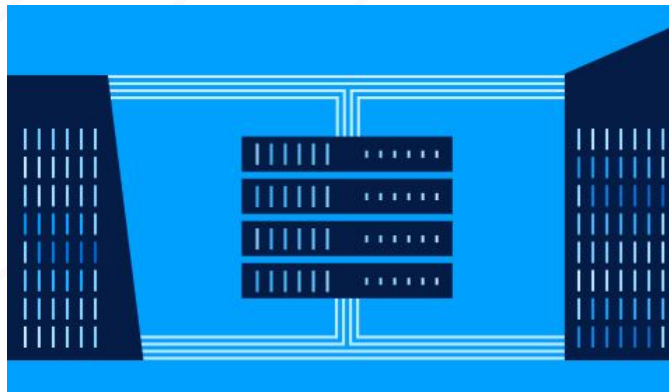
**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

# ¿Qué es una base de datos?

Una **base de datos** es una colección organizada de información o datos que se almacena de manera sistemática para un fácil acceso, gestión y actualización.

En términos más simples, se puede considerar como un **almacén de datos**. Los datos en una base de datos se **organizan** y **almacenan** de manera que se puedan **recuperar** y **manipular** de **manera eficiente**.



# Propiedades

Las bases de datos tienen varias propiedades que ayudan a garantizar la coherencia, la integridad y la confiabilidad de los datos que contienen.

- Persistencia e Independencia de datos
- Redundancia mínima.
- Concurrencia.
- Seguridad de los datos y Auditoría.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Respaldo y recuperación.

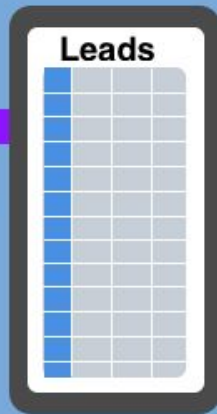
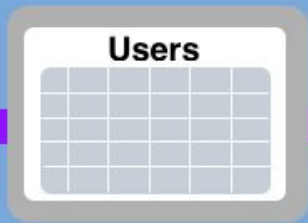
# Tipos de Bases de Datos

SQL & NoSQL

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

# SQL vs. NoSQL



# Bases de Datos SQL

**DEV.FX**  
DESARROLLAMOS(PERSONAS);

dev

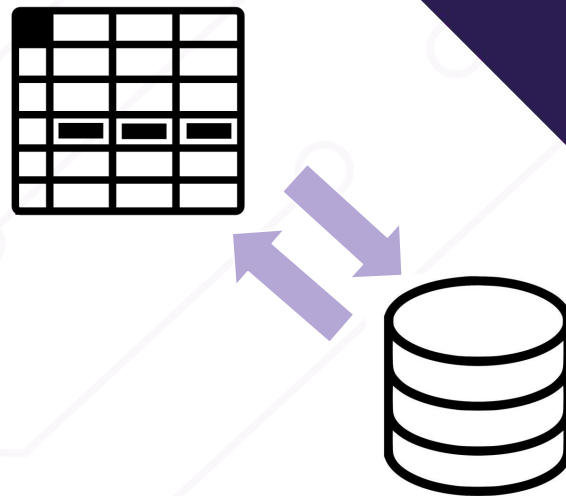
# Bases de Datos SQL

*SQL = Structured Query Language*

Comúnmente se les llama **bases de datos relacionales** (y/o **Tabulares**).

**Nota:** la realidad es que en BD NoSQL (el otro tipo de bases de datos) también existen relaciones.

Una base de datos SQL se compone de una o más **tablas**. Cada tabla está compuesta de **columnas** (atributos) y **filas** (registros).



# Propiedades - Bases de Datos Relacionales (SQL)

- Las bases de datos SQL exigen una **estructura de datos consistente**: la estructura de cada tabla debe definirse antes de almacenar los datos. Esto garantiza una cierta coherencia en los datos almacenados.
- Las bases de datos SQL **se basan en el modelo relacional**, una forma intuitiva y directa de representar **datos** en **tablas**.
- Funcionan con un **lenguaje estructurado de consulta (SQL)**, el cual es potente, flexible y se utiliza para interactuar con la base de datos, permitiendo variedad de operaciones.
- Están diseñadas para seguir el modelo **ACID**, el cual es un conjunto de propiedades (**Atomicidad, Consistencia, Aislamiento, Durabilidad**) para garantizar la seguridad de las transacciones.



# Forma Común de una tabla

- Cada **fila** en esta tabla representaría un **registro** individual, ex: Un Estudiante.
- Cada registro tiene un **ID** único llamado **clave** o **key**
- Las **columnas** de la tabla contienen **atributos** del registro.
- Cada registro generalmente tiene un valor para cada atributo.

The diagram illustrates a database table structure. It features three columns labeled 'Atributo1', 'Atributo2', and 'Atributo3' at the top, each with a bracket indicating its span. The table itself is represented by a grid of colored cells. The first row is a dark blue header. Below it are several rows of lighter blue cells. To the left of the table, three green arrows point to specific rows, labeled 'Tupla 1', 'Tupla 2', and 'Tupla n' respectively. The entire grid is labeled 'Relación' at the bottom center.

	Atributo1	Atributo2	Atributo3
Tupla 1			
Tupla 2			
Tupla n			

Relación

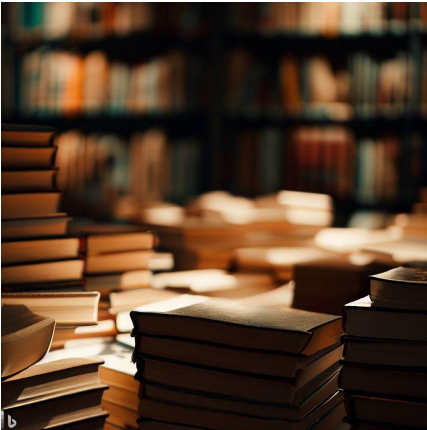
# Ejemplo (Empleados)

ID	nombre	apellidos	email	telefono	salario
1	Laura	Rodriguez	laura@devf.mx	55555555	500
2	Victor	Dominguez	victor@devf.mx	53555555	
3	Helge	Aburto	helge@devf.mx	54555555	400

# Entidades

En las bases de datos, una entidad es cualquier objeto en el sistema que queremos modelar y almacenar información sobre él.

Por ejemplo, en una base de datos de una **librería**, las entidades podrían ser los **Libros**, los **Autores**, los **Empleados**, etc. Cada entidad se representa como una **tabla** en la base de datos, y las instancias individuales de una **entidad** se representan como **filas** en la tabla.



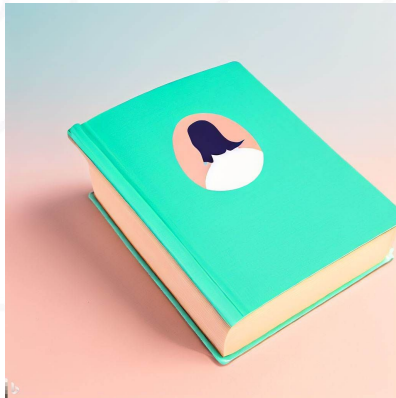
# Relaciones

Las relaciones son las asociaciones que existen entre las entidades. Por ejemplo, un **Autor escribe un Libro**. Aquí, "**escribe**" es la relación entre las entidades Autor y Libro. En una base de datos relacional, estas relaciones se pueden representar a través de **claves primarias** y **claves foráneas**.



# Cardinalidad

La **cardinalidad** es el término utilizado para describir la cantidad de elementos en un conjunto, y en el caso de las bases de datos, **describe la relación entre los registros de dos tablas**. Específicamente, se refiere al número máximo de veces que una instancia en una entidad puede estar asociada con instancias en la entidad relacionada. Por ejemplo, la cardinalidad entre **Autores** y **Libros** podría ser "**uno a muchos**", lo que significa que un **Autor** puede escribir muchos Libros, pero cada **Libro** ha sido escrito por un único Autor.



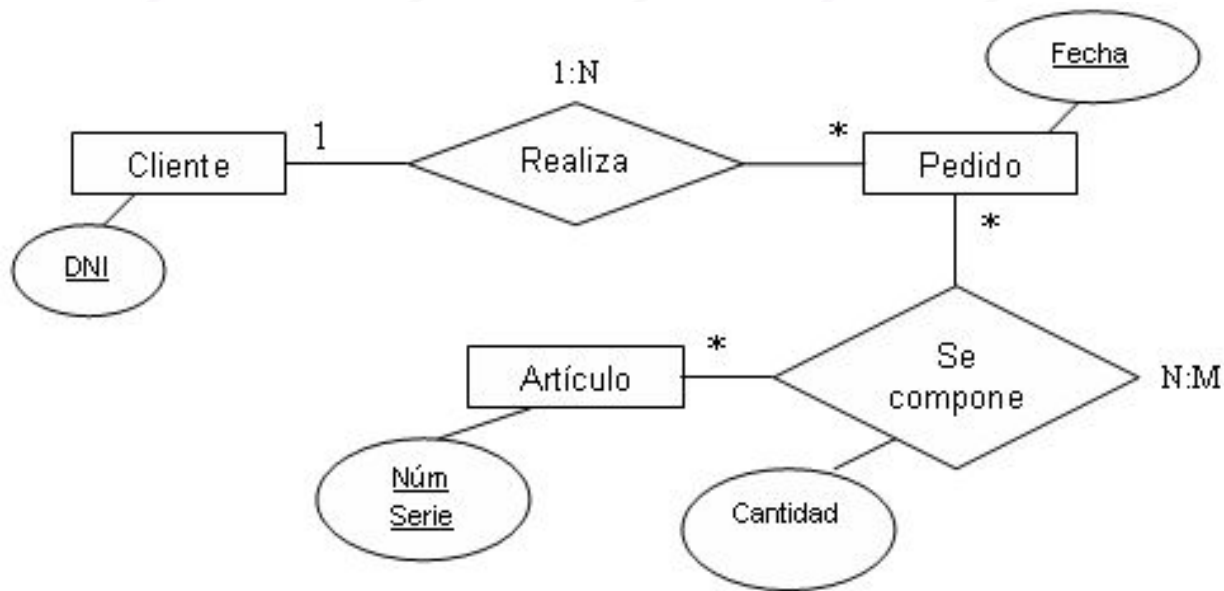
# Cardinalidad

Los tipos más comunes son:

- Uno a uno 1:1
- Uno a muchos 1:N
- Muchos a uno N:1
- Muchos a Muchos N:M

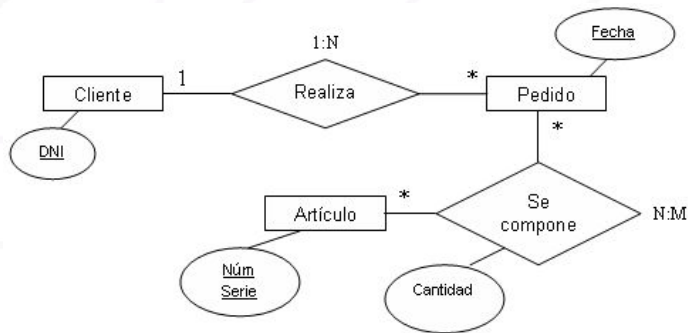
# Modelo Relacional | Entidad-Relación (ER)

El modelo ER es una herramienta conceptual que se utiliza en el diseño de bases de datos, donde las entidades se representan como **nodos** y las relaciones se representan como arcos entre los nodos. Este modelo se utiliza para abstraer y representar los datos en un nivel alto y comprensible para el usuario.



# Modelo Relacional | Entidad-Relación (ER)

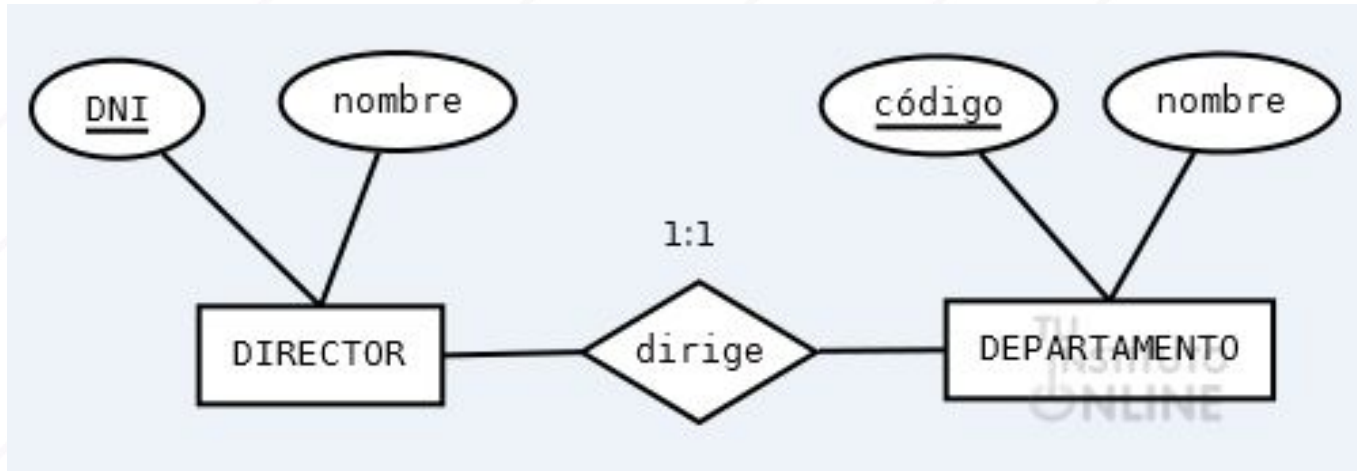
- ❑ Un **cliente** puede realizar varios **pedidos**, pero cada **pedido** es hecho por un único **cliente**. Esto es una relación uno a muchos (**1:N**). Aquí, la entidad "Cliente" es la entidad principal y "Pedido" es la entidad dependiente.
- ❑ Un pedido puede contener varios artículos, y un artículo puede estar en varios pedidos. Esto es una relación muchos a muchos (**N:M**). Para manejar este tipo de relación en una base de datos relacional, **normalmente se necesitaría una tabla de unión o intermedia**. Esta tabla podría llamarse "DetallePedido" y tendría referencias tanto al pedido como al artículo (es decir, incluiría las **claves foráneas** de ambas tablas), además de cualquier otra información relevante para la relación, como la **cantidad de cada artículo** en el pedido.





## Relacion 1 : 1

Es cuando el registro de una tabla está asociado única y exclusivamente al registro de otra tabla



# Relacion 1 : N

Es cuando el registro de una tabla está asociado con varios registros de otra tabla

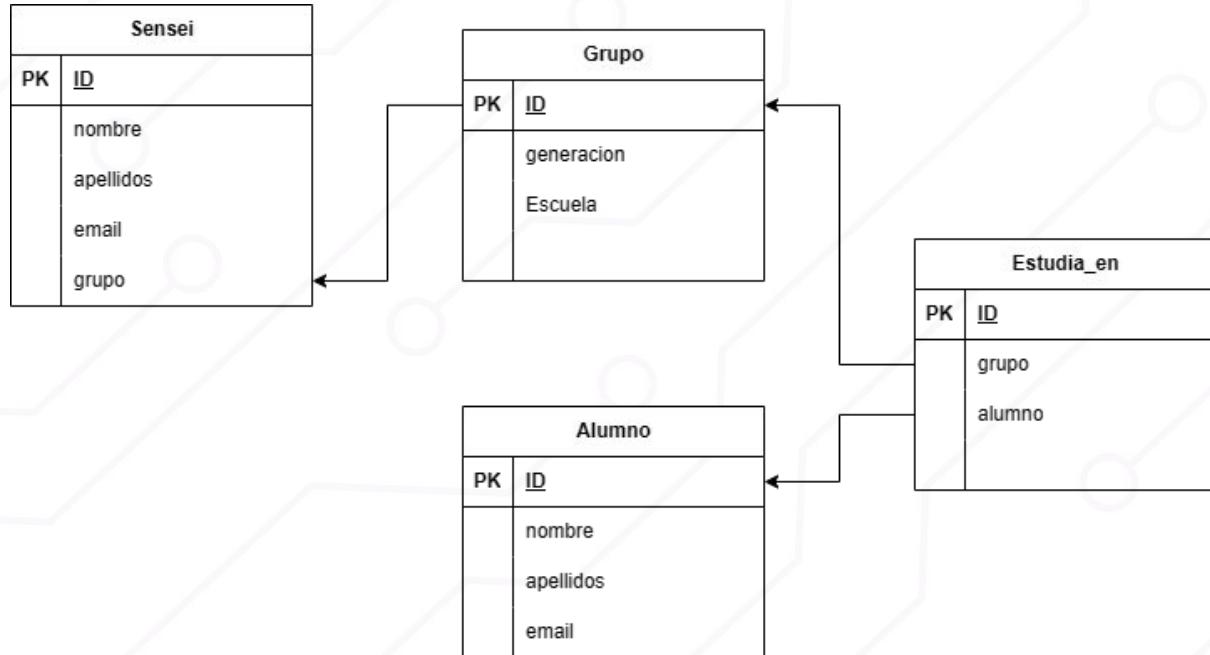


# Relacion N : M

Es cuando uno o más registro de una tabla está asociado con varios registros de otra tabla, por lo regular se usa una tabla intermedia para asociarlos.



# Modelo entidad - relación



**PK:** Primary key o llave primaria es el identificador único de cada registro en una tabla.

**FK:** Foreign Key o Llave foránea, indica que hay una relación con otra tabla a través del primary key

# Tipos de base de datos (relacionales) SQL



Microsoft®  
**SQL Server®**



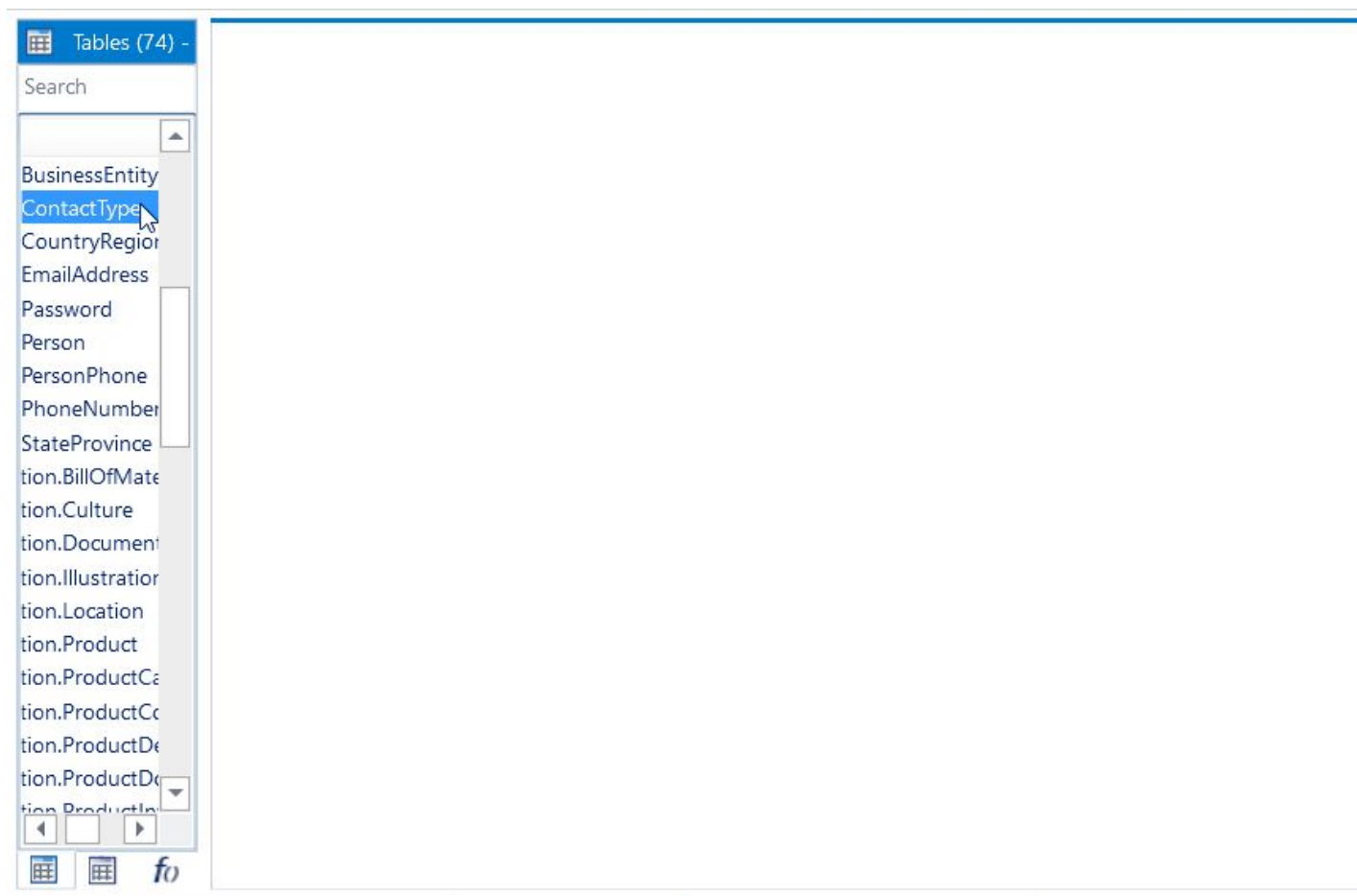
PostgreSQL



# Sistema de Gestión de Base de Datos (SGBD)

Los Sistemas de Gestión de Base de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.





# Ejercicio: Veterinaria

Una veterinaria, requiere un sistema que le ayude a controlar la entrada de las mascotas a la veterinaria.

La veterinaria requiere almacenar el nombre de la mascota, el tipo de mascota, raza y edad.

Del dueño de la mascota se necesitan datos personales, cada mascota puede ser atendida por uno o varios veterinarios de los cuales necesitamos saber cédula profesional, nombre , edad, años de experiencia.

**Para este negocio, es necesario crear:**

- 1. Diagrama Relacional**
- 2. Diagrama Entidad Relación**