

# Resumen de bases de datos

**¿Que es una base de datos?**

R: administra, gestiona, actualiza, almacena o guarda una serie de datos información, de manera eficiente.

## ¿Que es una tabla?

R: esta creada con cfilas y columnas que almacenan datos. de manera organizada y estructurada, evitando la duplicidad de datos.

## ¿Que es una relación?

R: es un asocioacion entre dos tablas o mas el cual se relacionan por una clabe unica de los datos.

# Normalización de Bases de Datos

## ¿Qué es la Normalización?

La normalización es un proceso utilizado en el diseño de bases de datos para minimizar la redundancia y mejorar la integridad de los datos.

# Objetivos de la Normalización

- ✓ Eliminar redundancias.
- ✓ Mejorar la integridad de los datos.
- ✓ Facilitar la actualización y mantenimiento de la BD.
- ✓ Evitar anomalías de inserción, eliminación y actualización.

# Formas Normales (NF)

Existen varias formas normales, cada una con criterios específicos:

1. **Primera Forma Normal (1FN)**
2. **Segunda Forma Normal (2FN)**
3. **Tercera Forma Normal (3FN)**
4. **Forma Normal de Boyce-Codd (BCNF)**

# Ejemplo Base de Datos

Supongamos la siguiente tabla no normalizada:

ID_Cliente	Nombre	Dirección	Teléfono	Productos_Comprados
1	Juan	Calle 123	555-1234	Laptop, Mouse
2	María	Calle 456	555-5678	Tablet

✓ Esta tabla contiene datos repetidos y no cumple con las normas de normalización.



# Primera Forma Normal (1FN)

**Regla:** Cada celda debe contener un solo valor.

```
CREATE TABLE Cliente (  
    ID_Cliente SERIAL PRIMARY KEY,  
    Nombre VARCHAR(100),  
    Dirección TEXT,  
    Teléfono VARCHAR(20)  
);  
  
CREATE TABLE Producto_Cliente (  
    ID SERIAL PRIMARY KEY,  
    ID_Cliente INT REFERENCES Cliente(ID_Cliente),  
    Producto VARCHAR(100)  
);
```

✓ Ahora cada celda tiene un solo valor.

# Segunda Forma Normal (2FN)

**Regla:** Cumple con 1FN y todos los atributos dependen completamente de la clave primaria.

```
CREATE TABLE Producto (  
    ID_Producto SERIAL PRIMARY KEY,  
    Nombre VARCHAR(100)  
);  
  
CREATE TABLE Compra (  
    ID SERIAL PRIMARY KEY,  
    ID_Cliente INT REFERENCES Cliente(ID_Cliente),  
    ID_Producto INT REFERENCES Producto(ID_Producto)  
);
```

✓ Se eliminó la dependencia parcial dividiendo la tabla en entidades separadas.

# Tercera Forma Normal (3FN)

**Regla:** Cumple con 2FN y no hay dependencia transitiva.

**Ejemplo:** La **dirección** podría dividirse en una tabla separada.

```
CREATE TABLE Direccion (  
    ID_Direccion SERIAL PRIMARY KEY,  
    Calle TEXT,  
    Ciudad VARCHAR(100),  
    CodigoPostal VARCHAR(10)  
);  
  
ALTER TABLE Cliente ADD COLUMN ID_Direccion INT REFERENCES Direccion(ID_Direccion);
```

✓ Se evita la redundancia de direcciones en múltiples clientes.

# Beneficios de la Normalización

- ✓ Mejora la consistencia de los datos.
- ✓ Reduce el almacenamiento innecesario.
- ✓ Facilita la actualización de registros.
- ✓ Evita errores en las consultas.

# Conclusión

- 📌 La normalización es clave en bases de datos bien estructuradas.
- 📌 Se debe equilibrar la normalización con la eficiencia en consultas complejas.