

Normalización en el Modelo Relacional

La normalización es el proceso de organizar los datos en una base de datos para reducir la redundancia y mejorar la integridad. A continuación, se describen las cinco formas normales:

1FN - Primera Forma Normal (1NF)

- Cada celda debe contener **un solo valor**.
- Cada fila debe ser **única**.
- **Uso:** Garantiza que cada atributo tenga valores atómicos, evitando datos compuestos o repetidos en una misma columna.
- **Cardinalidad:** 1:1 (Cada celda solo tiene un valor por atributo).

Ejemplo:

```
CREATE TABLE Persona (  
  ID SERIAL PRIMARY KEY,  
  Nombre VARCHAR(100) NOT NULL,  
  Edad INT NOT NULL  
);
```

2FN - Segunda Forma Normal (2NF)

- Cumple la 1FN.
- Todos los atributos no clave deben depender completamente de la **clave primaria**.
- **Uso:** Elimina dependencias parciales separando datos que dependen solo de una parte de la clave primaria.
- **Cardinalidad:** 1:N (Una persona puede tener uno o más registros de detalles).

Ejemplo:

```
CREATE TABLE Persona (  
  ID SERIAL PRIMARY KEY,  
  Nombre VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE DetallesPersona (  
  ID SERIAL PRIMARY KEY,  
  PersonaID INT REFERENCES Persona(ID),  
  Edad INT NOT NULL  
);
```

3FN - Tercera Forma Normal (3FN)

- Cumple la 2FN.
- Ningún atributo no clave debe depender de otro atributo no clave.
- **Uso:** Elimina dependencias transitivas organizando datos en tablas separadas.
- **Cardinalidad:** 1:N (Una persona puede tener varios contactos).

Ejemplo:

```
CREATE TABLE Persona (  
  ID SERIAL PRIMARY KEY,  
  Nombre VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE Contacto (  
  ID SERIAL PRIMARY KEY,  
  PersonaID INT REFERENCES Persona(ID),  
  Telefono VARCHAR(15) NOT NULL  
);
```

4FN - Cuarta Forma Normal (4NF)

- Cumple la 3FN.
- Las relaciones de múltiples valores deben separarse en tablas independientes.
- **Uso:** Separa datos con relaciones multivaluadas en tablas distintas para evitar redundancia.
- **Cardinalidad:** 1:N (Una persona puede tener múltiples aficiones o direcciones).

Ejemplo:

```
CREATE TABLE Persona (  
  ID SERIAL PRIMARY KEY,  
  Nombre VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Aficiones (  
    ID SERIAL PRIMARY KEY,  
    PersonaID INT REFERENCES Persona(ID),  
    Aficion VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Direcciones (  
    ID SERIAL PRIMARY KEY,  
    PersonaID INT REFERENCES Persona(ID),  
    Direccion VARCHAR(200) NOT NULL  
);
```


5FN - Quinta Forma Normal (5FN)

- Cumple la 4FN.
- Elimina dependencias de unión complejas.
- **Uso:** Separa relaciones complejas en tablas independientes para evitar redundancia y ambigüedades.
- **Cardinalidad:** N:M (Una persona puede estar asignada a varios proyectos y un proyecto puede tener varias personas asignadas).

Ejemplo:

```
CREATE TABLE Persona (  
  ID SERIAL PRIMARY KEY,  
  Nombre VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Proyecto (  
    ID SERIAL PRIMARY KEY,  
    Nombre VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Asignacion (  
    PersonaID INT REFERENCES Persona(ID),  
    ProyectoID INT REFERENCES Proyecto(ID),  
    PRIMARY KEY (PersonaID, ProyectoID)  
);
```