

## Normalização até a 3FN

- **Resumo Teórico**
- **Objetivo da normalização:**
  - ❖ *Evitar redundância e anomalias (inserção, exclusão, atualização) nos dados.*

### 1FN - Primeira Forma Normal:

#### O que exige?

- *Eliminar grupos repetitivos (valores múltiplos em uma mesma célula).*
- *Cada campo deve ter valor atômico (não dividido ou composto).*

#### Como fazer?

- *Criar linhas separadas para valores múltiplos.*

#### Exemplo ruim:

| Aluno | Telefones            |
|-------|----------------------|
| João  | 9999-1111, 9999-2222 |

#### Corrigido (1FN):

| Aluno | Telefone  |
|-------|-----------|
| João  | 9999-1111 |
| João  | 9999-2222 |

### 2FN - Segunda Forma Normal:

#### Pré-requisito: Estar na 1FN

#### O que exige?

- *Eliminar dependência parcial (campo que depende só de parte da chave primária, em tabelas com chave composta).*

#### Como fazer?

- *Separar os dados que dependem só de uma parte da chave em outra tabela.*

#### Exemplo:

- *Em uma tabela de Vendas com chave (CodCliente, CodProduto), o campo **NomeCliente** depende só de **CodCliente** → deve sair da tabela.*

### 3FN - Terceira Forma Normal:

#### Pré-requisito: Estar na 2FN

#### O que exige?

- *Eliminar dependência transitiva (campo depende de outro campo que não é a chave primária).*

#### Como fazer?

- *Criar nova tabela para os campos que dependem de outro campo não chave.*

#### Exemplo:

| CodCliente | Nome | Cidade | UF |

- *UF depende da Cidade, não diretamente do CodCliente → separar em uma tabela de cidades.*

### Resumo:

- ❖ 1FN = Sem repetições, tudo atômico.
- ❖ 2FN = Sem dependência parcial.
- ❖ 3FN = Sem dependência de não chave.

### Exemplo Inicial (Sem normalização)

Vamos considerar a seguinte tabela chamada Pedidos:

| PedidoID | ClienteNome | ClienteTelefone         | Produto          | Quantidade | PreçoUnit     | CidadeCliente | UF |
|----------|-------------|-------------------------|------------------|------------|---------------|---------------|----|
| 1        | João Silva  | 9999-1111,<br>9999-2222 | Caneta,<br>Lápis | 2, 3       | 2.00,<br>1.50 | Teresina      | PI |

### Problemas:

- Telefones e produtos estão em listas → violam a 1FN.
- Informações de cliente e cidade estão misturadas → redundância.

### Aplicando a 1FN (valores atômicos)

Separar os valores múltiplos (telefones, produtos, quantidades, preços):

Criamos uma linha para cada produto e telefone:

| PedidoID | ClienteNome | ClienteTelefone | Produto | Quantidade | PreçoUnit | CidadeCliente | UF |
|----------|-------------|-----------------|---------|------------|-----------|---------------|----|
| 1        | João Silva  | 9999-1111       | Caneta  | 2          | 2.00      | Teresina      | PI |
| 1        | João Silva  | 9999-2222       | Lápis   | 3          | 1.50      | Teresina      | PI |

### Aplicando a 2FN (eliminar dependência parcial)

Aqui a chave primária é composta: (PedidoID, Produto)

Mas ClienteNome, ClienteTelefone, CidadeCliente e UF dependem **só do Cliente**, e não do Produto → devem ir para uma nova tabela.

### Tabelas após 2FN:

#### Pedidos

| PedidoID | Produto | Quantidade | PreçoUnit |
|----------|---------|------------|-----------|
| 1        | Caneta  | 2          | 2.00      |
| 1        | Lápis   | 3          | 1.50      |

#### Clientes

| ClienteID | ClienteNome | ClienteTelefone | CidadeCliente | UF |
|-----------|-------------|-----------------|---------------|----|
| C1        | João Silva  | 9999-1111       | Teresina      | PI |
| C1        | João Silva  | 9999-2222       | Teresina      | PI |

PedidoCliente (relação entre pedido e cliente)

| PedidoID | ClienteID |
|----------|-----------|
| 1        | C1        |

### Aplicando a 3FN (eliminar dependência transitiva)

Agora observamos que UF depende de CidadeCliente, e não diretamente do cliente → criamos uma tabela para cidades:

### Tabelas finais após 3FN:

#### Pedidos

| PedidoID | Produto | Quantidade | PreçoUnit |
|----------|---------|------------|-----------|
| 1        | Caneta  | 2          | 2.00      |
| 1        | Lápis   | 3          | 1.50      |

#### Clientes

| ClienteID | ClienteNome |
|-----------|-------------|
| C1        | João Silva  |

#### TelefonesCliente

| ClienteID | Telefone  |
|-----------|-----------|
| C1        | 9999-1111 |
| C1        | 9999-2222 |

#### Cidades

| CidadeID | NomeCidade | UF |
|----------|------------|----|
| CT1      | Teresina   | PI |

#### ClienteCidade

| ClienteID | CidadeID |
|-----------|----------|
| C1        | CT1      |

#### PedidoCliente

| PedidoID | ClienteID |
|----------|-----------|
| 1        | C1        |

### Pronto! Agora os dados estão em 3FN:

- Sem repetição.
- Sem dependências parciais.
- Sem dependências transitivas.