

# Modelo Lógico Relacional

A Estrutura Formal do Banco de Dados

Capítulo 3: Transformando conceitos abstratos em estruturas concretas e funcionais para sistemas de banco de dados robustos.



# Transformação do Modelo Conceitual em Relacional

01

## Modelo Conceitual

Representa entidades e relacionamentos de forma abstrata, focando no "o que" armazenar sem se preocupar com implementação técnica.

02

## Conversão Estrutural

Entidades tornam-se tabelas e atributos transformam-se em colunas com tipos de dados específicos.

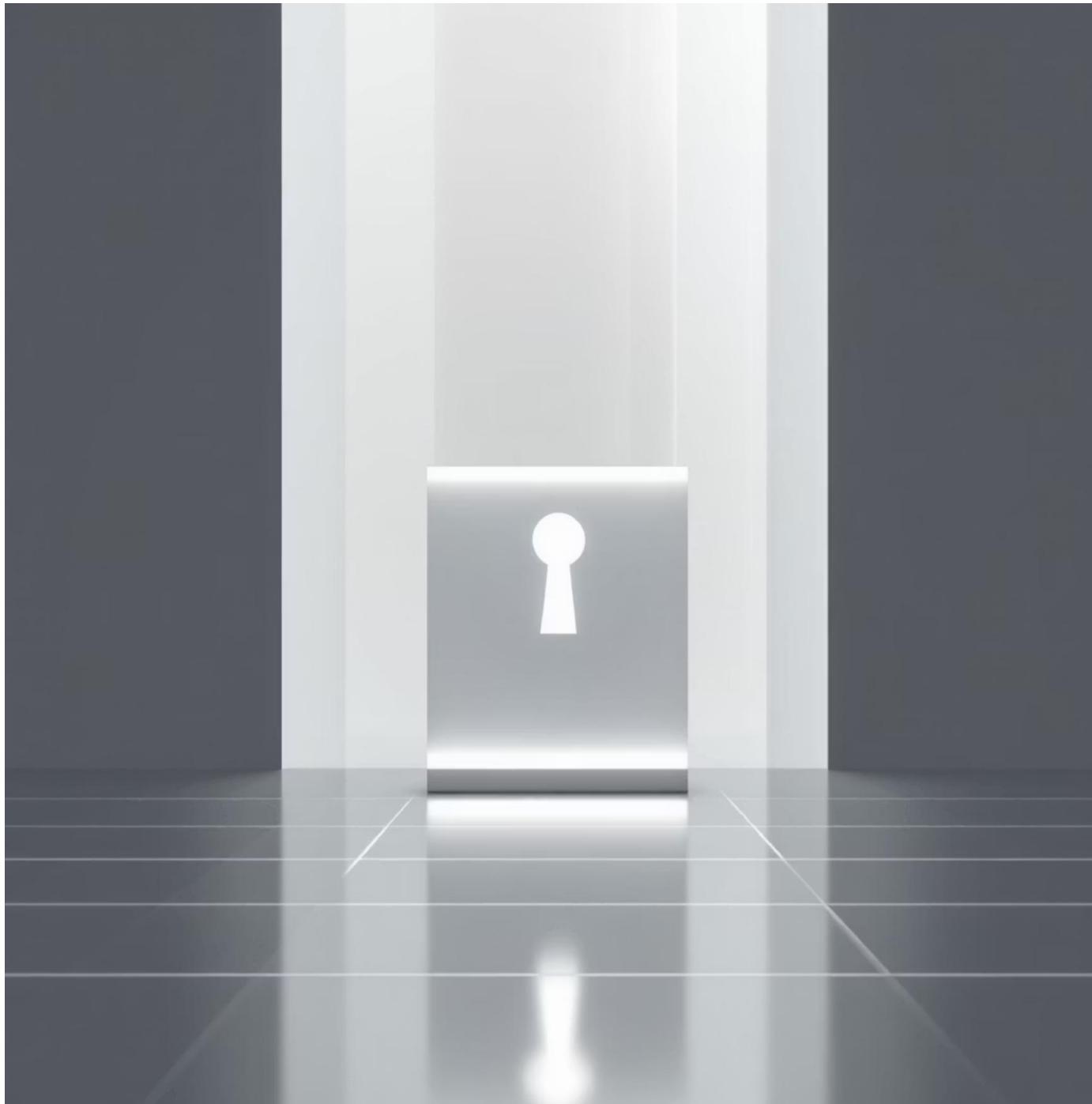
03

## Mapeamento de Relacionamentos

Relacionamentos são implementados através de chaves estrangeiras que conectam as tabelas entre si.

**Exemplo prático:** A entidade Cliente vira a tabela CLIENTE, com atributos como nome e CPF convertidos em colunas estruturadas.

# Chaves Primárias, Estrangeiras e Integridade Referencial



## Chave Primária (PK)

Identifica unicamente cada registro na tabela. Exemplo: código\_cliente garante que não existam clientes duplicados.



## Chave Estrangeira (FK)

Referencia a chave primária de outra tabela para estabelecer vínculos. Exemplo: código\_cliente na tabela PEDIDO.



## Integridade Referencial

Garante que toda chave estrangeira sempre aponte para um registro válido e existente, prevenindo inconsistências.

# Normalização Detalhada

Primeira, Segunda e Terceira Formas Normais



## Primeira Forma Normal (1FN)

Elimina grupos repetitivos e garante que cada campo contenha apenas valores atômicos indivisíveis.

- Cada célula possui um único valor
- Remove colunas multivaloradas



## Segunda Forma Normal (2FN)

Remove dependências parciais, garantindo que todos os atributos dependam da chave primária completa.

- Atende 1FN primeiro
- Elimina dependências parciais da PK



## Terceira Forma Normal (3FN)

Elimina dependências transitivas, assegurando que atributos não-chave dependam exclusivamente da chave primária.

- Atende 2FN primeiro
- Remove dependências transitivas

**Benefícios:** Evita redundância de dados, previne inconsistências e melhora significativamente a manutenibilidade do banco.

# Visualizando o Processo de Normalização



Tabela Desnormalizada

Dados repetitivos e redundantes



1FN

Valores atômicos



2FN

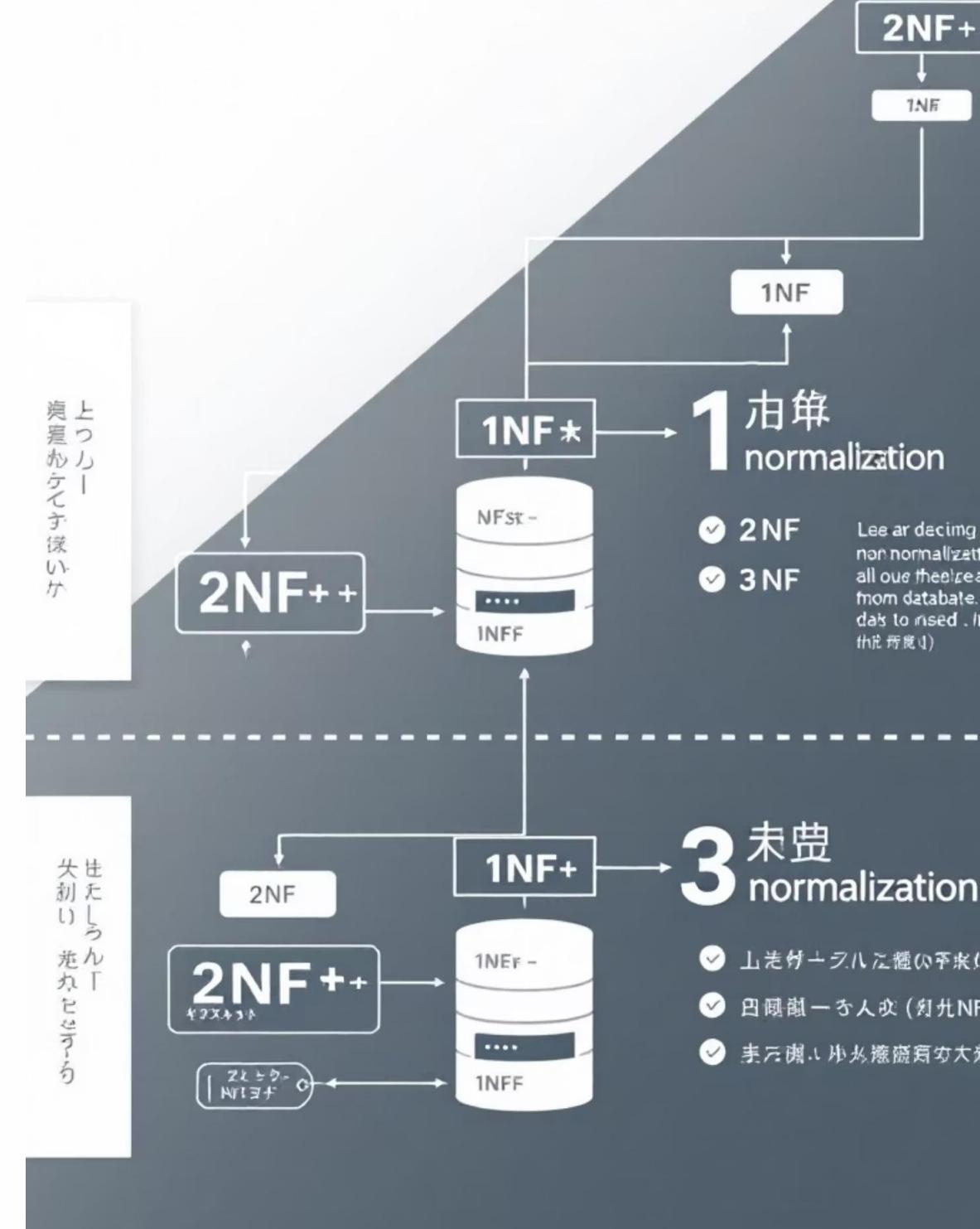
Sem dependências parciais

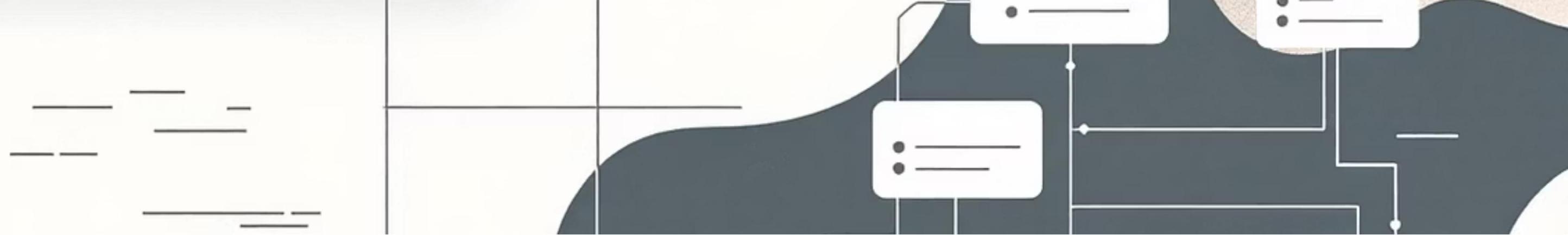


3FN

Sem dependências transitivas

Cada forma normal representa um nível superior de organização, resultando em estruturas mais eficientes e confiáveis.





# Regras de Mapeamento e Convenções

1

Entidades → Tabelas

Cada entidade do modelo conceitual torna-se uma tabela no modelo relacional, com atributos convertidos em colunas.

2

Relacionamentos 1:N

Adiciona-se chave estrangeira na tabela do lado N do relacionamento para estabelecer o vínculo.

3

Relacionamentos N:M

Cria-se tabela associativa com chaves compostas, conectando as duas entidades relacionadas.

4

Convenções de Nomes

Utilize nomes claros, consistentes e padronizados para tabelas e colunas, facilitando compreensão e manutenção.

- **Exemplo:** Tabela PEDIDO\_PRODUTO implementa relacionamento N:M entre PEDIDO e PRODUTO.

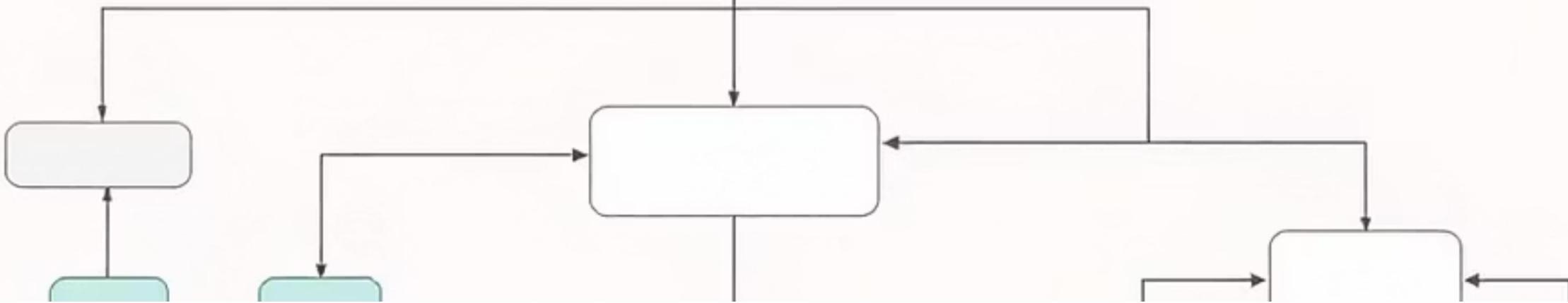
# Exportando Código DDL SQL no brModelo

## Recursos Principais

- Geração automática de scripts DDL completos
- Definição de chaves primárias e estrangeiras
- Especificação de tipos de dados
- Inclusão de restrições de integridade
- Redução de erros manuais

O brModelo transforma visualmente seu modelo lógico em código SQL executável, pronto para implementação direta no SGBD.





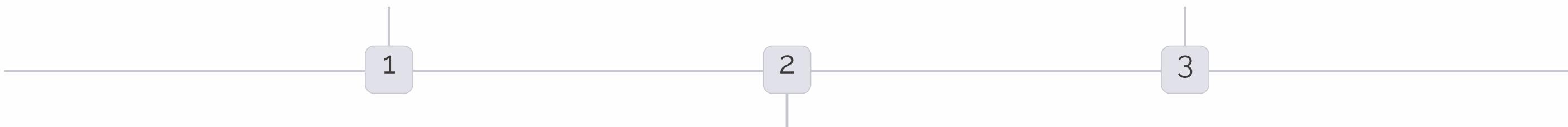
# Caso Prático: Do Conceitual ao Código DDL

Modelo Conceitual

Cliente com relacionamento 1:N para Telefone. Pedido com relacionamento N:M para Produto.

Código DDL

Scripts SQL com CREATE TABLE, PRIMARY KEY, FOREIGN KEY e constraints de integridade.



Modelo Lógico

Tabela TELEFONE com FK para CLIENTE. Tabela ITEM\_PEDIDO conectando PEDIDO e PRODUTO.

**Estrutura resultante:** Tabela TELEFONE vinculada a CLIENTE por FK, tabela ITEM\_PEDIDO com chaves compostas unindo PEDIDO e PRODUTO, garantindo integridade referencial completa.

# Impacto da Modelagem Lógica Relacional



## Estrutura Organizada

Garante organização eficiente dos dados, facilitando operações de consulta e atualização.



## Manutenção Simplificada

Facilita atualizações e correções, reduzindo tempo e custos de manutenção do sistema.



## Escalabilidade

Cria base sólida para sistemas robustos que crescem com as necessidades do negócio.



## Consistência de Dados

Reduz drasticamente redundâncias e inconsistências através de normalização adequada.



# Domine a Modelagem Lógica Relacional

## Ponte Essencial

O modelo lógico conecta o mundo conceitual abstrato à implementação física concreta do banco de dados.

## Conhecimento Fundamental

Compreender chaves, normalização e regras de mapeamento é essencial para qualquer profissional de dados.

## Ferramentas Poderosas

O brModelo e outras ferramentas agilizam o processo e garantem qualidade na criação do banco.

**Próximos passos:** Pratique a transformação de modelos conceituais, domine as formas normais e construa bancos de dados confiáveis e eficientes.