

Guia de Normalização de Bancos de Dados: Do Caos à Estrutura

Material de Estudo para Aula

1. Introdução à Normalização

A **Normalização** é um processo sistemático utilizado na modelagem de bancos de dados relacionais para organizar os dados de forma eficiente. O conceito foi introduzido por **Edgar F. Codd** em 1970, visando fundamentar os bancos de dados em princípios matemáticos (teoria dos conjuntos) para garantir a integridade e facilitar a manutenção.

Objetivos Principais

- **Eliminar redundância:** Evitar que a mesma informação seja armazenada em múltiplos locais.
 - **Prevenir anomalias:** Garantir que operações de inserção, atualização e exclusão não gerem inconsistências.
 - **Facilitar a manutenção:** Mudanças em um dado devem ser feitas em apenas um lugar.
 - **Otimizar o armazenamento:** Reduzir o espaço ocupado por dados repetidos.
-

2. Anomalias: O Perigo da Falta de Normalização

Quando um banco de dados não está normalizado, ele sofre de **Anomalias**, que são comportamentos inesperados que corrompem a integridade dos dados.

Tipos de Anomalias

Tipo	Descrição	Exemplo Prático
Inserção	Impossibilidade de adicionar um dado sem a existência de outro.	Não conseguir cadastrar um novo <i>Curso</i> se ainda não houver nenhum <i>Aluno</i> matriculado nele.
Exclusão	Perda indesejada de informações ao remover um registro.	Ao excluir o único <i>Aluno</i> de um curso, as informações do <i>Professor</i> e do <i>Curso</i> também desaparecem.
Atualização	Inconsistência gerada quando um dado repetido é alterado em apenas alguns locais.	Alterar o nome de um <i>Professor</i> em uma linha, mas esquecer de alterar nas outras 50 onde ele aparece.

Exemplo de Tabela Não Normalizada:

Código Aluno	Nome Aluno	Curso	Professor
1	João	Informática	Carlos
2	Maria	Informática	Carlos
3	Pedro	Redes	Ana

Problema: O nome “Carlos” está redundante. Se ele mudar de nome, o risco de erro na atualização é alto.

3. Conceitos Fundamentais

Antes de aplicar as Formas Normais, é essencial entender as dependências:

- 1. **Chave Primária (PK):** Atributo que identifica unicamente uma linha.
- 2. **Chave Composta:** Quando a PK é formada por dois ou mais atributos (ex: `AlunoID` + `DisciplinaID`).
- 3. **Dependência Funcional:** Quando o valor de A determina o valor de B ($A \rightarrow B$).
- 4. **Dependência Parcial:** Quando um atributo depende de apenas *parte* de uma chave composta.

5. **Dependência Transitiva:** Quando um atributo depende de outro atributo que não é a chave ($A \rightarrow B \rightarrow C$).

4. As Formas Normais (FN)

As Formas Normais são níveis de conformidade. Para uma tabela estar na 3FN, ela deve obrigatoriamente estar na 1FN e 2FN.

4.1. Primeira Forma Normal (1FN) - Atomicidade

Regra: Não devem existir grupos repetitivos nem atributos multivalorados. Cada campo deve conter um único valor indivisível (**atômico**).

- **Violação:** Uma coluna “Telefones” com “9999-1111, 9888-2222”.
- **Solução:** Criar uma nova tabela ou novas linhas para cada valor.

4.2. Segunda Forma Normal (2FN) - Dependência Total

Regra: Deve estar na 1FN e todos os atributos não-chave devem depender da **chave primária completa**. *Atenção: Só é relevante para tabelas com Chaves Compostas.*

- **Violação:** Na tabela `Matrícula(AlunoID, DisciplinaID, NomeAluno, Nota)`, o `NomeAluno` depende apenas de `AlunoID` (parte da chave).
- **Solução:** Separar `NomeAluno` para a tabela `Aluno`.

4.3. Terceira Forma Normal (3FN) - Independência de Atributos

Regra: Deve estar na 2FN e não possuir **dependências transitivas**. Atributos não-chave devem depender apenas da chave, e de nada mais.

- **Violação:** Na tabela `Funcionário(ID, Nome, DeptoID, NomeDepto)`, o `NomeDepto` depende de `DeptoID`, que não é a chave da tabela.
 - **Solução:** Criar a tabela `Departamento(DeptoID, NomeDepto)`.
-

5. Além da 3FN: Forma Normal de Boyce-Codd (BCNF)

Embora a maioria dos bancos comerciais pare na 3FN, a **BCNF** (ou 3.5FN) é um refinamento necessário quando existem múltiplas chaves candidatas que se sobrepõem.

Regra de Ouro: “Todo determinante deve ser uma chave candidata.”

Se você tem uma dependência $X \rightarrow Y$, então X deve ser uma superchave. Isso elimina redundâncias sutis que a 3FN pode deixar passar em casos complexos de chaves compostas sobrepostas.

6. Resumo e Macete para Provas

Forma Normal	O que eliminar?	Pergunta Chave
1FN	Multivalorados / Listas	Existe mais de um valor na mesma célula?
2FN	Dependências Parciais	O dado depende de apenas parte da chave composta?
3FN	Dependências Transitivas	O dado depende de outro campo que não é a chave ?
BCNF	Determinantes não-chave	O campo que determina outro é uma chave candidata?

7. Exercício Resolvido

Cenário: Tabela de Pedidos PedidoID (PK), ProdutoID (PK), NomeProduto, ClienteID, NomeCliente, Quantidade

- Análise 1FN:** Ok (assumindo valores atômicos).
- Análise 2FN: Falha.** NomeProduto depende apenas de ProdutoID. NomeCliente depende apenas de ClienteID.
- Correção (Normalização):**
 - PRODUTO (ProdutoID, NomeProduto)
 - CLIENTE (ClienteID, NomeCliente)
 - ITEM_PEDIDO (PedidoID, ProdutoID, ClienteID, Quantidade)

8. Conclusão para o Aluno

Normalizar não é apenas “separar tabelas”. É garantir que cada fato seja armazenado em **apenas um lugar**. Isso torna o banco de dados robusto, escalável e imune a erros bizarros de atualização que podem custar caro em sistemas reais.