



Normalização

Professor: Elton Sarmanho¹
E-mail: eltonss@ufpa.br



¹Faculdade de Sistemas de Informação - UFPA/CUTINS

14 de janeiro de 2024

Roteiro

Introdução

Conceito

Tipos

Referências Bibliográficas



└ Introdução

└ Conceito

Conceito de Normalização

Definição

A **normalização** é um processo sistemático em bancos de dados para reduzir a redundância de dados e melhorar a integridade dos dados. Ela envolve a organização das colunas (atributos) e das tabelas (relações) de um banco de dados para minimizar a dependência e a complexidade dos dados.

- ▶ **Objetivo:** Evitar anomalias de inserção, atualização e exclusão.
- ▶ **Método:** Dividir uma grande tabela em tabelas menores e relacioná-las usando chaves estrangeiras.
- ▶ **Resultado:** Aumento da consistência e qualidade dos dados, facilitando a manutenção.



Primeira Forma Normal (1FN)

- ▶ Cada coluna deve conter valores atômicos, sem grupos de dados.
 - ▶ Atributos compostos e multivalorados.
- ▶ Cada registro deve ser único.
- ▶ Importância: Elimina grupos de dados, simplificando o acesso e a manipulação dos dados.



Primeira Forma Normal (1FN)

Definição

A tabela está na **Primeira Forma Normal (1FN)** se:

- ▶ Cada coluna contém valores atômicos (indivisíveis).
- ▶ Cada registro (linha) é único.

Objetivo: Evitar a duplicidade e a complexidade na manipulação dos dados.



└ Introdução
└ Tipos

Exemplo Prático de 1FN

Antes de Aplicar 1FN:

Aluno	Cursos
Ana	Matemática, Física
João	História
Maria	Literatura, Matemática

Após Aplicar 1FN:

Aluno	Curso
Ana	Matemática
Ana	Física
João	História
Maria	Literatura
Maria	Matemática

Nota: Cada curso é agora listado em uma linha separada, garantindo valores atômicos.



Atributos Multivalorados

Antes de Aplicar 1FN:

Aluno	Telefones
Pedro	(11) 1234-5678, (11) 8765-4321
Ana	(21) 1234-5678
Carlos	(31) 9876-5432, (31) 5432-9876, (31) 1122-3344

Após Aplicar 1FN:

Aluno	Telefone
Pedro	(11) 1234-5678
Pedro	(11) 8765-4321
Ana	(21) 1234-5678
Carlos	(31) 9876-5432
Carlos	(31) 5432-9876
Carlos	(31) 1122-3344



└ Introdução
└ Tipos

Atributos Compostos

Antes de Aplicar 1FN:

Professor	Endereço
Maria	Rua A, 123, São Paulo, SP
João	Av. B, 456, Rio de Janeiro, RJ

Após Aplicar 1FN:

Professor	Rua	Número	Cidade	Estado
Maria	Rua A	123	São Paulo	SP
João	Av. B	456	Rio de Janeiro	RJ



Segunda Forma Normal (2FN)

- ▶ Aplica-se a tabelas com uma chave primária composta.
- ▶ Um registro está na 2FN se estiver na 1FN e todos os atributos não-chave forem dependentes de toda a chave primária.
- ▶ Importância: Reduz a redundância de dados e **dependência parcial**.



Dependências Parciais na 2FN

- ▶ Dependências parciais são um conceito-chave na normalização de dados, especialmente na Segunda Forma Normal (2FN).
- ▶ Ocorrem quando uma coluna em uma tabela depende funcionalmente apenas de parte da chave primária, em vez de depender de toda a chave primária.

Exemplo: Suponha uma tabela de pedidos de vendas:

- ▶ Chave Primária: (Pedidoid, Produtoid)
- ▶ Outras colunas: ProdutoNome, Quantidade
- ▶ Dependência Parcial: ProdutoNome depende apenas de Produtoid, não de Pedidoid.
- ▶ Isso indica que o nome do produto está associado apenas ao Produtoid, não ao Pedidoid.

Para resolver dependências parciais, é comum dividir a tabela em partes, garantindo que cada coluna dependa completamente da chave primária da tabela à qual pertence.



Segunda Forma Normal (2FN)

Definição

Uma tabela está na **Segunda Forma Normal (2FN)** se:

- ▶ Está na Primeira Forma Normal (1FN).
- ▶ Todos os atributos não-chave são totalmente dependentes da chave primária.

Objetivo: Eliminar a redundância e dependência parcial dos dados.



Exemplo Prático de 2FN

Antes de Aplicar 2FN:

ProfessorID	NomeProfessor	Departamento
101	João Silva	Matemática
102	Ana Costa	Física
101	João Silva	Computação

Após Aplicar 2FN:

ProfessorID	NomeProfessor
101	João Silva
102	Ana Costa

ProfessorID	Departamento
101	Matemática
101	Computação
102	Física

Nota: Os dados foram divididos em duas tabelas para garantir que todos os atributos não-chave dependam totalmente da chave primária.

Chave Primária Composta: Conceito

Breve explicação sobre chaves primárias compostas:

- ▶ Chaves formadas pela combinação de duas ou mais colunas.
- ▶ Utilizadas para identificar de forma única cada registro em uma tabela.
- ▶ Importantes em relações muitos-para-muitos e na normalização de dados.



└ Introdução
└ Tipos

Exemplo - Antes da 2FN

Tabela **OrdensDeCompra**:

- ▶ OrdemID, ProdutoID, ClientelD, ProdutoNome, ClienteNome
- ▶ Observação: Redundância de dados e dependência parcial.



└ Introdução
└ Tipos

Exemplo Antes da 2FN

Tabela **OrdensDeCompra**:

OrdemID	ProdutoID	ClienteID	ProdutoNome	ClienteNome
1	101	501	Caneta	Ana Silva
1	102	501	Caderno	Ana Silva
2	103	502	Lápis	Bruno Dias

Tabela: Não normalizada com redundâncias



└ Introdução
└ Tipos

Aplicando a 2FN

- ▶ Separação das informações em diferentes tabelas.
- ▶ Redução de redundâncias e eliminação de dependências parciais.



Aplicando a 2FN

Processo de normalização:

1. Identificar e separar as dependências parciais.
2. Criar tabelas separadas para entidades independentes.
3. Estabelecer relações com chaves primárias compostas.



Exemplo Após a 2FN

Tabelas Normalizadas:

- ▶ Tabela **Ordens** (OrdemID, ClienteID)
- ▶ Tabela **Produtos** (ProdutoID, ProdutoNome)
- ▶ Tabela **Clientes** (ClienteID, ClienteNome)
- ▶ Tabela **OrdensProdutos** (OrdemID, ProdutoID) - Chave Primária Composta



Exemplo Após a 2FN

Tabelas Normalizadas:

Tabela Ordens

OrdemID	ClienteID
1	501
2	502

Tabela Produtos

ProdutoID	ProdutoNome
101	Caneta
102	Caderno
103	Lápis

Tabela Clientes

ClienteID	ClienteNome
501	Ana Silva
502	Bruno Dias

Tabela OrdensProdutos

OrdemID	ProdutoID
1	101
1	102
2	103



Consulta SQL com Chave Primária Composta

```
SELECT O.OrdemID, P.ProdutoNome, C.ClienteNome  
FROM Ordens O  
JOIN OrdensProdutos OP ON O.OrdemID = OP.OrdemID  
JOIN Produtos P ON OP.ProdutoID = P.ProdutoID  
JOIN Clientes C ON O.ClienteID = C.ClienteID;
```

Esta consulta exemplifica o uso das tabelas normalizadas e a relação definida pela chave primária composta.



Terceira Forma Normal (3FN)

- ▶ Um registro está na 3FN se estiver na 2FN e todos os seus atributos não-chave forem dependentes apenas da chave primária.
- ▶ Elimina **dependências transitivas**.
- ▶ Importância: Aumenta a eficiência na atualização de dados e garante maior precisão e consistência.



Dependências Transitivas

- ▶ Elas ocorrem quando a relação entre três atributos implica uma dependência funcional entre dois deles por meio do terceiro.
- ▶ Por exemplo, se A depende funcionalmente de B e B depende funcionalmente de C , então A depende transitivamente de C .

Exemplo: Suponha uma tabela de alunos com os atributos:

- ▶ ID_Aluno (PK)
- ▶ $Nome_Aluno$
- ▶ $ID_Professor$
- ▶ $Nome_Professor$
- ▶ ID_Aluno determina $Nome_Aluno$.
- ▶ $ID_Professor$ determina $Nome_Professor$.
- ▶ Dependência Transitiva: $ID_Aluno \rightarrow ID_Professor \rightarrow Nome_Professor$.

Dependências transitivas podem levar a redundância de dados e são eliminadas ou reduzidas na normalização de dados.

Exemplo Antes da 3FN

Tabela **FuncionariosProjetos**:

FuncID	NomeFunc	DeptID	NomeDept	ProjetoID	NomeProjeto
1	Ana Silva	D01	Marketing	P123	Campanha A
2	Bruno Dias	D02	Vendas	P124	Estratégia B

Tabela: Não normalizada com dependências transitivas



Aplicando a 3FN

Processo de normalização para a 3FN:

1. Identificação e remoção de dependências transitivas.
2. Criação de tabelas separadas para cada entidade.



└ Introdução
└ Tipos

Exemplo Após a 3FN

Tabela Funcionarios

FuncID	NomeFunc	DeptID
1	Ana	D01
2	Bruno	D02

Tabela Projetos

ProjID	NomeProjeto
P123	Campanha A
P124	Estratégia B

Tabela Departamentos

DeptID	NomeDept
D01	Marketing
D02	Vendas

Tabela FuncionariosProjetos

FuncID	ProjID
1	P123
2	P124



Resumo da Aula

- ▶ A normalização é um processo essencial em bancos de dados relacionais.
- ▶ As formas normais, incluindo 1FN, 2FN e 3FN, ajudam a eliminar a redundância e garantir a integridade dos dados.
- ▶ A 1FN exige valores atômicos e registros únicos.
- ▶ A 2FN vai além, garantindo que todos os atributos não-chave sejam totalmente dependentes da chave primária.
- ▶ A 3FN elimina dependências transitivas.



Aplicações Práticas

- ▶ A normalização é amplamente utilizada na indústria para projetar bancos de dados eficientes e confiáveis.
- ▶ Reduz a necessidade de atualizações e manutenção complexa.
- ▶ Contribui para um melhor desempenho e escalabilidade de sistemas de gerenciamento de dados.



Considerações Finais

- ▶ Lembre-se de aplicar esses conceitos ao projetar sistemas de banco de dados.
- ▶ Pratique a normalização em projetos de bancos de dados para aprimorar suas habilidades.



Referências I

- 📄 HEUSER, C. A. Projeto de Banco de Dados. 2^a edição. Porto Alegre: Sagra-Luzzato, 1999.
- 📄 SILBERSCHATZ, A.; KORTH, H.; SUDARSHAN, S. Sistema de Banco de Dados. 3^a edição. São Paulo: Markon Books, 1999.
- 📄 DATE, C. J. Introdução a Sistemas de Banco de Dados. 8^a ed. Tradução de Daniel Vieira. Rio de Janeiro, Editora Campus, 2003.





Normalização

Professor: Elton Sarmanho¹
E-mail: eltonss@ufpa.br



¹Faculdade de Sistemas de Informação - UFPA/CUTINS

14 de janeiro de 2024