



EDUARDO AKIRA DA ROSA MIYAMOTO

MODELAGEM DE DADOS:
NORMALIZAÇÃO DE DADOS

TOYOKAWA - JAPÃO
2025

EDUARDO AKIRA DA ROSA MIYAMOTO - RA2025*****

MODELAGEM DE DADOS

NORMALIZAÇÃO DE DADOS - UNIDADE 4

Professor da Matéria: Kariston Stevan Luiz

Curso: CyberSecurity

Relatório apresentado à disciplina de
MODELAGEM DE DADOS, do curso de
Cibersecurity da Faculdade Anhanguera
polo Guarulhos, como requisito parcial
para aprovação na disciplina da Unidade
4.

Toyokawa - Japão
2025

INTRODUÇÃO

A modelagem de dados é uma etapa fundamental no desenvolvimento de sistemas de informação, pois visa organizar os dados de maneira eficiente, garantindo sua consistência, integridade e facilidade de manutenção. No contexto de bancos de dados relacionais, a normalização é uma técnica essencial para reduzir redundâncias, evitar inconsistências e facilitar atualizações futuras.

O presente trabalho tem como objetivo aplicar os conceitos de normalização de dados, especificamente a transformação da Primeira Forma Normal (1FN) para a Segunda Forma Normal (2FN), utilizando como cenário uma pequena livraria online. As tabelas fornecidas – Pedidos, Produtos e Clientes – já se encontram em 1FN, ou seja, todos os atributos possuem valores atômicos. A partir dessa base, o trabalho consiste em identificar e eliminar dependências parciais, garantindo que cada atributo não-chave seja totalmente dependente da chave primária, conforme os princípios da 2FN.

Aula 2: Transformação 1FN - 2FN

Nome do Projeto: Livraria Online

Linguagem usada: SQL

Programa usado: MySQL Workbench

ATIVIDADE PROPOSTA:

Você recebeu um conjunto de dados de uma pequena livraria online. O objetivo é normalizar esses dados para garantir que estejam em uma forma mais eficiente e consistente.

1. Identificar as Tabelas Inicialmente

As tabelas "Pedidos", "Produtos" e "Clientes" já estão em 1FN, pois todos os atributos contêm valores atômicos.

2. Identificar Dependências Parciais

Para atingir a 2FN, precisamos eliminar dependências parciais. Isso significa que todo atributo não-chave deve ser totalmente funcionalmente dependente da chave primária.

3. Criar Novas Tabelas para Eliminar Dependências Parciais

Dessa maneira você terá normalizado esse modelo para a 2FN. A normalização facilita a manutenção e atualização dos dados, reduzindo redundâncias e melhorando a integridade dos dados.

Todo o desenvolvimento deste projeto será realizado usando o MySQL Workbench, o que seria essa ferramenta ?

O MySQL Workbench é uma ferramenta visual unificada para arquitetos de banco de dados, desenvolvedores e administradores de banco de dados. O MySQL Workbench oferece modelagem de dados, desenvolvimento em SQL e ferramentas de administração abrangentes para configuração de servidores, administração de usuários, backup e muito mais. O MySQL Workbench está disponível para Windows, Linux e Mac OS X.

É uma ferramenta visual que nos permite modelar de forma mais fácil o banco de dados.

Você pode encontrar mais informações diretamente no site oficial:

“ <https://www.mysql.com/products/workbench/> ”

Aqui você encontra o link para o download do MySQL Workbench:

“ <https://dev.mysql.com/downloads/workbench/> ”

Todos os código e arquivos aqui disponibilizados estarão também em nosso github:

“ <https://github.com/akiramiyamoto-dev> ”

DEFINIÇÕES

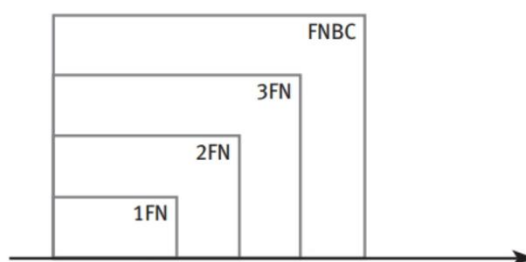
O que é Normalização de Dados?

A normalização é um processo matemático aplicado ao modelo relacional de bancos de dados, baseado na teoria dos conjuntos, que tem como objetivo melhorar a qualidade das tabelas. Esse procedimento analisa e impõe restrições a uma tabela para reduzir redundâncias, corrigir anomalias de inserção, exclusão e atualização, e alcançar propriedades desejáveis de organização.

Durante o processo, as tabelas são avaliadas segundo regras chamadas formas normais, que devem ser seguidas de maneira sequencial: uma forma normal inferior é pré-requisito para a seguinte. Assim, uma tabela na Terceira Forma Normal (3FN) automaticamente também respeita a Segunda (2FN) e a Primeira (1FN). Em níveis mais avançados, como a Forma Normal de Boyce-Codd (FNBC), também se incluem as anteriores.

Quando uma tabela não está bem projetada, ela é decomposta em outras menores e mais eficientes, sem perda de informação e preservando os relacionamentos e a semântica dos dados. Dessa forma, garante-se a consistência lógica e a criação de esquemas de banco de dados claros, aplicáveis e de alta qualidade.

Veja a imagem a seguir que mostra os níveis de Normalização:



Definição do que é a 1FN (Primeira Forma Normal)

A **Primeira Forma Normal (1FN)** é a base da normalização e exige que:

- Uma tabela só estará em 1FN se todas as colunas possuírem um único valor (atômico), ou seja, o valor não pode ser subdividido nem conter múltiplos valores.
- Não devem existir grupos repetitivos (colunas) em uma linha, nem atributos compostos.

Aplicar a 1FN consiste em retirar da estrutura os elementos repetitivos, ou seja, aqueles dados que poderiam compor uma estrutura do tipo vetor, e organizá-los em grupos lógicos (novas tabelas, se necessário). Cada tabela em 1FN deve ter uma chave primária única que identifica cada linha.

Definição do que é a 2FN (Segunda Forma Normal)

A **Segunda Forma Normal (2FN)** só pode ser aplicada após a 1FN ter sido realizada. O seu princípio fundamental é **eliminar a Dependência Parcial**.

- **Requisito Principal:** Todo atributo que não for chave deve ser **totalmente funcionalmente** dependente de **toda a Chave Primária**.

Isso significa que se uma tabela tiver uma **chave primária composta** (formada por várias colunas), cada atributo não-chave deverá estar diretamente relacionado com **todas** as colunas da chave primária.

Se um atributo estiver relacionado somente com **parte** da chave primária (o que chamamos de Dependência Parcial), ele deverá ser removido da tabela original e colocado em uma **nova tabela separada**, juntamente com a parte da chave da qual ele realmente depende. A 2FN garante que a informação só exista uma única vez no banco de dados.

Desenvolvimento

1 - Verificação da Primeira Forma Normal (1FN)

O Primeiro passo da normalização é verificar se todas as tabelas estão na Primeira Forma Normal(1FN).

Tabela Pedidos:

PedidoID	ClientelD	NomeCliente	ProdutoID	NomeProduto	Quantidade	PreçoUnitário	EndereçoEntrega
1	101	João Silva	201	Livro A	2	20	Rua A, 123
2	102	Maria Souza	202	Livro B	1	30	Rua B, 456
3	103	Ana Lima	203	Livro C	3	25	Rua C, 789

Tabela Produtos:

ProdutoID	NomeProduto	Categoria	PreçoUnitário
201	Livro A	Ficção	20
202	Livro B	História	30
203	Livro C	Ciência	25

Tabela Clientes:

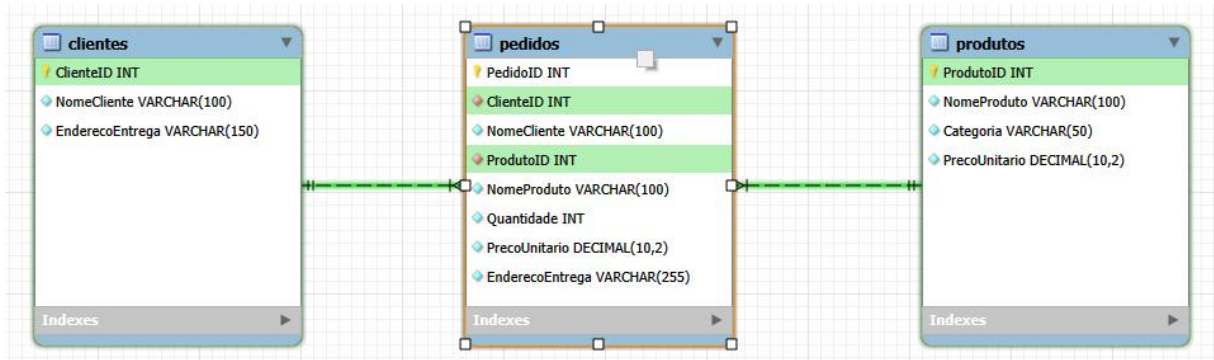
ClientelD	NomeCliente	EndereçoEntrega
101	João Silva	Rua A, 123
102	Maria Souza	Rua B, 456
103	Ana Lima	Rua C, 789

As tabelas originais(Pedidos, Produtos e Clientes) estão em 1FN. Isso ocorre porque:

- **Atomicidade:** Todos os atributos(colunas) contêm apenas valores atômicos, ou seja, não há grupos repetitivos de dados, nem colunas com múltiplos valores armazenados. Por exemplo, o NomeCliente armazena apenas um nome e o Quantidade armazena apenas um número.
- **Ausência de Grupos Repetitivos:** Não há repetição de colunas(como Produto1, Produto2, etc.) na mesma linha.

Diagrama Entidade-Relacionamento (DER) Inicial (1FN)

O diagrama a seguir representa a estrutura em 1FN, antes da normalização, mostrando a tabela Pedidos centralizada e redundante.



2 - Análise para a Segunda Forma Normal (2FN)

Atingir a Segunda Forma Normal (2FN) exige que a tabela esteja em 1FN e que não possua Dependências Parciais.

2.1 Estaremos a partir daqui analisando se há redundâncias nas tabelas:

A violação da 2FN é encontrada na tabela Pedidos, cuja Chave Primária é PedidoID. A dependência parcial ocorre quando um atributo não-chave depende apenas de uma parte da informação chave.

■ Dependência Parcial de Clientes

Os atributos NomeCliente e EnderecoEntrega são funcionalmente dependentes apenas do ClienteID, e não do PedidoID.

O que acontece nesse caso é que os dados do cliente são repetidos em cada linha de pedido, violando a atomicidade estrutural e a regra de dependência total. Dessa forma esses atributos na tabela Pedidos gera redundância e anomalias de atualização.

Tabela Pedidos(1FN)

PedidoID	ClienteID	NomeCliente	ProdutoID	NomeProduto	Quantidade	PreçoUnitário	EndereçoEntrega
1	101	João Silva	201	Livro A	2	20	Rua A, 123
2	102	Maria Souza	202	Livro B	1	30	Rua B, 456
3	103	Ana Lima	203	Livro C	3	25	Rua C, 789



■ Dependência Parcial de Produtos

Os atributos NomeProduto e PreçoUnitario são funcionalmente dependentes apenas do ProdutoID. Logo esses atributos, que deveriam estar na tabela Produtos, são repetidos na tabela Pedidos para cada item vendido.

Tabela Pedidos:

PedidoID	ClienteID	NomeCliente	ProdutoID	NomeProduto	Quantidade	PreçoUnitário	EndereçoEntrega
1	101	João Silva	201	Livro A	2	20	Rua A, 123
2	102	Maria Souza	202	Livro B	1	30	Rua B, 456
3	103	Ana Lima	203	Livro C	3	25	Rua C, 789

Estes atributos, que deveriam estar exclusivamente na tabela Produtos, são repetidos na tabela Pedidos para cada item vendido.

3 - Solução para transformar em 2FN: Decomposição das colunas

A solução para eliminar a Dependência Parcial e atingir a 2FN é a Decomposição. O Processo envolve:

- **Remover** os atributos que são parcialmente dependentes (NomeCliente, ProdNomeProduto, PreçoUnitario, EndereçoEntrega)
- **Criar** uma nova tabela associativa, PedidoDetalhes, com uma Chave Primária Composta formada por (PedidoID, ProdutoID).

Dessa forma, os atributos restantes na nova tabela PedidoDetalhes(Como Quantidade e PreçoVenda) passam a depender totalmente da chave composta, validando a Segunda Forma Normal.

Tabela Pedidos (com a nova formação):

PedidoID(PK)	ClienteID(FK)
1	101
2	102
3	103

PedidoDetalhes(Entidade Criada):

PedidoID	ProdutoID	Quantidade	PreçoUnitário
1	201	2	20
2	202	1	30
3	203	3	25

As outras duas tabelas que já havíamos falado continuaram iguais, pois já respeitavam as regras da 1FN e também se enquadravam na 2FN :

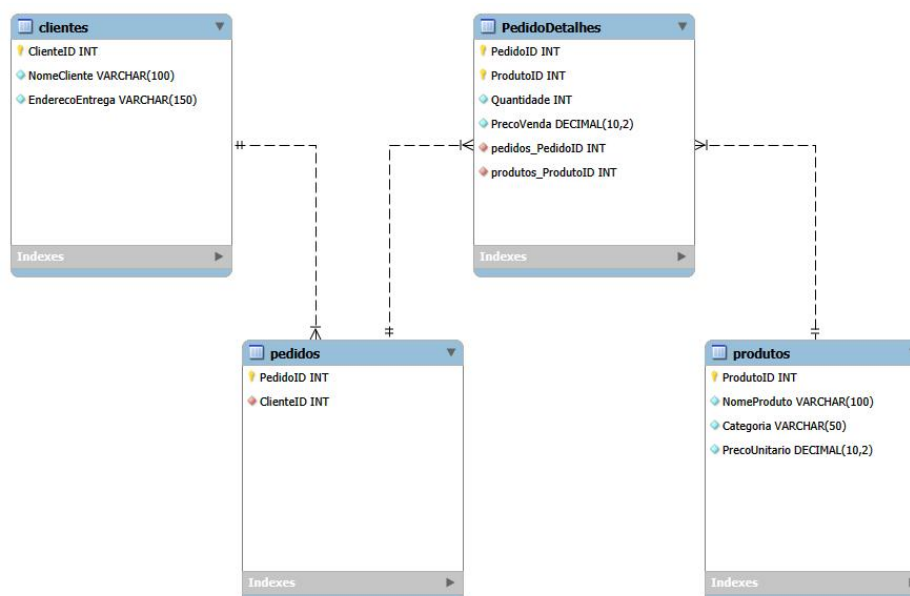
Tabela Produtos:

ProdutoID	NomeProduto	Categoria	PreçoUnitário
201	Livro A	Ficção	20
202	Livro B	História	30
203	Livro C	Ciência	25

Tabela Clientes:

ClienteID	NomeCliente	EndereçoEntrega
101	João Silva	Rua A, 123
102	Maria Souza	Rua B, 456
103	Ana Lima	Rua C, 789

Veja como ficou o nosso Diagrama Entidade-Relacionamento (DER):



Na tabela Pedidos, inicialmente havia informações de produtos, o que gerava dependência parcial, pois atributos como NomeProduto e PreçoUnitario dependiam apenas de ProdutoID. Para atender à 2FN, criamos a tabela PedidoDetalhes, que relaciona Pedidos e Produtos, eliminando as dependências parciais. Agora, todos os atributos não-chave são totalmente dependentes da chave primária.

RESUMO

Este trabalho teve como objetivo aplicar os conceitos de normalização de dados no contexto de uma pequena livraria online, realizando a transformação da Primeira Forma Normal (1FN) para a Segunda Forma Normal (2FN).

Inicialmente, as tabelas Pedidos, Produtos e Clientes estavam em 1FN, com atributos atômicos, mas apresentavam redundâncias e dependências parciais. Na tabela Pedidos, os atributos referentes aos clientes (NomeCliente, EndereçoEntrega) dependiam apenas de ClienteID, e os atributos dos produtos (NomeProduto, PreçoUnitário) dependiam apenas de ProdutoID, o que violava a 2FN.

Para corrigir essas dependências, foi realizada a decomposição da tabela Pedidos, criando-se a tabela associativa PedidoDetalhes, com chave primária composta por (PedidoID, ProdutoID). Essa estrutura eliminou as redundâncias e garantiu que cada atributo não-chave dependesse totalmente da chave primária.

Com isso, o modelo final passou a ser formado por quatro tabelas: Clientes, Produtos, Pedidos e PedidoDetalhes. Essa reorganização assegurou maior consistência, facilidade de manutenção e integridade dos dados, atendendo plenamente aos princípios da Segunda Forma Normal (2FN).

REFERÊNCIAS

MACHADO, Felipe Nery Rodrigues. Banco de Dados: projeto e implementação. 4. ed. Rio de Janeiro: Érica, 2020. E-book. ISBN 978-85-365-3270-7. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/9788536532707/pageid/176>. Acesso em: 03 out. 2025.

LEAL, Gislaine Camila Lapasini. *Linguagem, programação e banco de dados: guia prático de aprendizagem*. 1. ed. Rio de Janeiro: InterSaberes, 2015. E-book. ISBN 978-85-4430-258-3. Disponível em: “<https://plataforma.bvirtual.com.br/Leitor/Publicacao/30495/epub/136?code=...>”. Acesso em: 03 out. 2025.

BACONLINE. O que é a normalização de bases de dados e como fazê-la. E-BAC Online, [S. l.], 1 jun. 2023. Disponível em: <https://ebaonline.com.br/blog/normalizacao-de-bases-de-dados>. Acesso em: 03 out. 2025.