



1FN

2FN

3FN

Normalização de Banco de Dados

Eliminando Redundância e Garantindo
Integridade dos Dados

Aula 02 - UC3

Objetivos da Aula



Compreender o Problema da Redundância

Entender por que a redundância de dados é prejudicial em bancos de dados, identificando desperdício de espaço, dificuldade de manutenção e riscos de inconsistência.



Dominar as Três Formas Normais

Aprender as regras da Primeira Forma Normal (1FN), Segunda Forma Normal (2FN) e Terceira Forma Normal (3FN), compreendendo quando e como aplicar cada uma.



Identificar Anomalias

Reconhecer as três anomalias causadas por esquemas desnormalizados: anomalia de inserção, anomalia de atualização e anomalia de exclusão.



Aplicar Normalização na Prática

Desenvolver habilidade prática de normalizar esquemas desnormalizados, aplicando sistematicamente as três formas normais em cenários reais.

Por que Normalizar?



Por que não colocar todos os dados em uma única tabela?

✗ O Problema da Redundância

Quando colocamos todos os dados em uma única tabela, enfrentamos sérios problemas:

Desperdício de espaço: Mesmos dados repetidos múltiplas vezes

Dificuldade de atualização: Preciso modificar várias linhas para um único dado

Risco de inconsistência: Dados conflitantes sobre a mesma entidade

Anomalias operacionais: Problemas ao inserir, atualizar ou excluir dados

✓ A Solução: Normalização

A normalização é um processo sistemático que organiza os dados em tabelas menores e interligadas:

Elimina redundância: Cada dado armazenado uma única vez

Garante integridade: Consistência automática dos dados

Facilita manutenção: Atualizações em um único local

Previne anomalias: Operações seguras e previsíveis

Normalização é o processo de organizar os dados de um banco de dados para **minimizar redundância e dependências indesejadas**, aplicando um conjunto de regras chamadas **Formas Normais**.

O Problema do Esquema Desnormalizado

Tabela: PedidosGeral (Exemplo de Loja Online)

ID_Pedido	Cliente_Nome	Cliente_CPF	Cliente_Endereco	Produto_Nome	Produto_Preco
100	Aha Silva	12345678901	Rua X, 123	Camiseta	50.00
100	Aha Silva	12345678901	Rua X, 123	Calça Jeans	100.00
101	Bruno Costa	98765432109	Av. Y, 456	Camiseta	50.00
102	Aha Silva	12345678901	Rua X, 123	Tênis	150.00
103	Carlos Lima	11122233344	Rua Z, 789	Calça Jeans	100.00

Dados do Cliente Repetidos

Dados do Produto Repetidos

Problemas Identificados:

1 Redundância de Dados

Os dados de Ana Silva aparecem **3 vezes**. O produto "Camiseta" e seu preço aparecem **2 vezes**. Quanto mais pedidos, mais repetição.

2 Desperdício de Espaço

Informações que poderiam estar em tabelas separadas são duplicadas desnecessariamente, ocupando mais espaço no banco de dados.

3 Dificuldade de Atualização

Se Ana Silva mudar de endereço, é preciso atualizar **3 linhas diferentes**. Se o preço da Camiseta mudar, múltiplas linhas precisam ser atualizadas.

4 Risco de Inconsistência

Ao atualizar dados repetidos, podemos esquecer alguma linha, gerando **informações conflitantes** sobre a mesma entidade.

Anomalias Causadas pela Redundância

Primeira Forma Normal (1FN)

Regra da 1FN

Uma tabela está na Primeira Forma Normal quando todos os atributos contêm apenas VALORES ATÔMICOS (indivisíveis) e não há grupos repetitivos.

Conceitos-Chave:

- Valor Atômico:** Cada célula contém apenas UM valor, não uma lista ou conjunto de valores
- Grupos Repetitivos:** Campos que contêm múltiplos valores separados por vírgula, ponto-e-vírgula ou outros delimitadores
- Solução:** Separar cada valor em uma linha distinta, criando linhas adicionais quando necessário

Benefício: Facilita consultas SQL, permite filtros eficientes e garante que cada dado possa ser acessado individualmente.

✖ ANTES da 1FN (Violação)

ID_Pedido	Cliente	Produtos
100	Ana Silva	Camiseta, Calça Jeans
101	Bruno Costa	Camiseta
102	Ana Silva	Tênis

⚠ Campo "Produtos" contém múltiplos valores!

↓ Aplicando 1FN

✓ DEPOIS da 1FN (Normalizado)

ID_Pedido	Cliente	Produto
100	Ana Silva	Camiseta
100	Ana Silva	Calça Jeans
101	Bruno Costa	Camiseta
102	Ana Silva	Tênis

✓ Cada célula contém apenas UM valor!

Segunda Forma Normal (2FN)

Conceito-Chave: Dependência Parcial

Ocorre quando um atributo depende apenas de **parte** da chave primária composta, e não da chave completa. Só acontece em tabelas com chave primária formada por múltiplos campos.

☰ Regras da 2FN

A tabela deve estar na **1FN** (valores atômicos)

NÃO pode ter dependências parciais da chave primária

Todos os atributos não-chave devem depender da **chave primária COMPLETA**

💡 Como Aplicar

Passo 1: Identificar a chave primária composta

Passo 2: Analisar cada atributo: depende da chave completa ou apenas de parte?

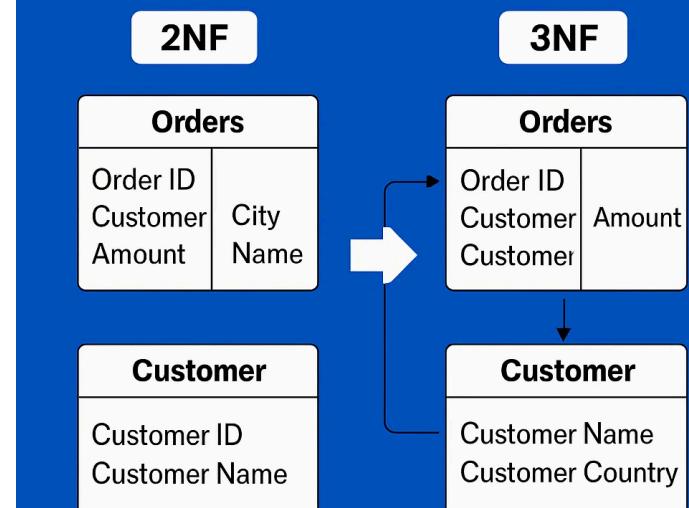
Passo 3: Criar tabelas separadas para dependências parciais

Passo 4: Estabelecer relacionamentos via chaves estrangeiras

★ Resultado: Cada atributo depende da chave primária **COMPLETA**, eliminando redundâncias causadas por dependências parciais.

2NF to 3NF Transformation

Schema Diagrams for Normalization



In 3NF, non-key columns must depend on the primary key, eliminating transitive dependency

Terceira Forma Normal (3FN)

Regras da 3FN

Uma tabela está na Terceira Forma Normal quando:

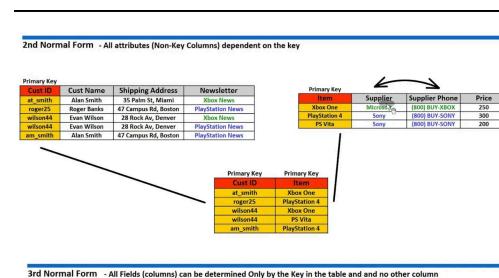
Está na **2FN** (sem dependências parciais)

NÃO possui dependências transitivas

Todos os atributos não-chave dependem **DIRETAMENTE** da chave primária

Dependência Transitiva

Ocorre quando um atributo não-chave depende de **outro atributo não-chave**, e não diretamente da chave primária. Exemplo: A → B e B → C, então A → C (transitiva).



✗ ANTES da 3FN (com dependência transitiva)

ID_Pedido	Cliente_CPF	Cliente_Nome	Cliente_Endereco
100	12345678901	Ana Silva	Rua X, 123
101	98765432109	Bruno Costa	Av. Y, 456
102	12345678901	Ana Silva	Rua X, 123

Identificando Dependência Transitiva

Exemplo: Na tabela Pedidos com chave primária ID_Pedido:

- ID_Pedido → Cliente_CPF (OK, dependência direta)
- Cliente_CPF → Cliente_Nome (problema!)
- Cliente_CPF → Cliente_Endereco (problema!)

Resultado: ID_Pedido → Cliente_CPF → Cliente_Nome é uma dependência **TRANSITIVA**. O nome do cliente depende do CPF, não diretamente do pedido.

✓ DEPOIS da 3FN (sem dependências transitivas)

Tabela: Clientes		
Cliente_CPF	Cliente_Nome	Cliente_Endereco
12345678901	Ana Silva	Rua X, 123
98765432109	Bruno Costa	Av. Y, 456

Tabela: Pedidos	
ID_Pedido	Cliente_CPF (FK)
100	12345678901
101	98765432109
102	12345678901

✓ Benefícios: Cada cliente cadastrado apenas 1 vez | Sem redundância | Atualização em único local | Todas as dependências são diretas

Atividade Prática - Normalização Passo a Passo

Cenário: Sistema de Biblioteca

Você foi contratado para normalizar o banco de dados de uma biblioteca que está com problemas de redundância e inconsistência. Analise a tabela abaixo e aplique as três formas normais.

Tabela Desnormalizada: EmprestimosGeral

ID_Emprestimo	Livro_Titulo	Livro_ISBN	Livro_Autor	Livro_Eitora	Usuario_Nome	Usuario_CPF	Usuario_Telefones	Data_Emprestimo	Data_Devolucao
1001	Clean Code	978-0132350	Robert Martin	Prentice Hall	João Silva	11111111111	(11)1111-1111, (11)2222-2222	2024-01-10	2024-01-24
1002	Design Patterns	978-0201633	Gang of Four	Addison	Maria Santos	22222222222	(11)3333-3333	2024-01-12	2024-01-26
1003	Clean Code	978-0132350	Robert Martin	Prentice Hall	João Silva	11111111111	(11)1111-1111, (11)2222-2222	2024-02-05	2024-02-19
1004	Refactoring	978-0134757	Martin Fowler	Addison	Pedro Costa	33333333333	(11)4444-4444	2024-02-10	2024-02-24

⚠ Observe: Campo "Usuario_Telefones" contém múltiplos valores! Dados de livros e usuários se repetem!

Organização

Trabalhem em **grupos de 3 a 4 pessoas**. Cada grupo deve analisar a tabela e aplicar sistematicamente as três formas normais (1FN, 2FN e 3FN).

Sua Tarefa

Desenhe o esquema normalizado final com **todas as tabelas resultantes**, indicando chaves primárias e estrangeiras.

Instruções Detalhadas da Atividade

PARTE 1: 1FN

Eliminar Grupos Repetitivos

1 Identificar Campos Multivalorados

Analise cada coluna e identifique campos que contêm múltiplos valores separados por vírgula ou ponto-e-vírgula.

2 Separar Valores

Para cada campo multivalorado, crie linhas adicionais. Cada valor deve estar em uma linha separada.

3 Definir Chave Primária

Identifique qual deve ser a chave primária composta da tabela resultante.

4 Verificar

Confirme que todas as células contêm apenas um valor e não há campos multivalorados.

PARTE 2: 2FN

Eliminar Dependências Parciais

1 Identificar Chave Composta

Qual é a chave primária da tabela após 1FN? Quais campos a compõem?

2 Analisar Dependências

Para cada atributo não-chave, pergunte: depende de TODA a chave ou apenas de PARTE?

3 Criar Tabelas Separadas

Crie uma tabela para cada grupo de dependência parcial identificado (Emprestimos, Livros, Usuarios, Telefones).

4 Definir Relacionamentos

Estabeleça chaves primárias e estrangeiras para relacionar as tabelas.

PARTE 3: 3FN

Eliminar Dependências Transitivas

1 Analisar Atributos Não-Chave

Há algum atributo não-chave que depende de outro atributo não-chave?

2 Identificar Dependências Transitivas

Exemplo: Livro_ISBN determina Titulo, Autor e Editora? Se sim, há dependência transitiva!

3 Criar Tabela para Atributo Determinante

Crie nova tabela com o atributo determinante como chave primária e mova os dependentes.

4 Verificar Esquema Final

Confirme que todas as dependências são diretas e não há mais dependências transitivas.

✓ Checklist de Validação Final

- Todos os valores são atômicos? (1FN)
- Não há dependências transitivas? (3FN)
- Atualizações exigem modificar apenas 1 lugar?
- Não há dependências parciais? (2FN)
- Posso cadastrar entidades independentemente?
- Exclusões não causam perda de dados?

Síntese e Consolidação

1FN

Primeira Forma Normal

Eliminar grupos repetitivos e garantir que todos os atributos contenham apenas valores atômicos

2FN

Segunda Forma Normal

Estar na 1FN e eliminar dependências parciais da chave primária composta

3FN

Terceira Forma Normal

Estar na 2FN e eliminar dependências transitivas entre atributos não-chave

Benefícios da Normalização



Eliminação de Redundância

Cada dado armazenado uma única vez, reduzindo desperdício de espaço e garantindo economia de recursos



Garantia de Integridade

Consistência automática dos dados, sem informações conflitantes sobre a mesma entidade



Facilidade de Manutenção

Atualizações realizadas em um único local, simplificando operações e reduzindo erros



Prevenção de Anomalias

Eliminação das anomalias de inserção, atualização e exclusão, garantindo operações seguras



Estrutura Organizada

Tabelas menores e interligadas, facilitando compreensão e manutenção do sistema



Melhor Desempenho

Consultas mais eficientes e menor uso de espaço em disco, otimizando o banco de dados

A normalização transforma um banco de dados caótico em um sistema **organizado, eficiente e confiável**.

Aplicando 1FN, 2FN e 3FN, você garante qualidade, integridade e facilidade de manutenção dos seus dados!