

**MODELAGEM DE DADOS**  
**Prof. Milton Palmeira Santana**



## NORMALIZAÇÃO

- » As primeiras técnicas de normalização foram criadas em 1972 por Edgar Frank Codd.
  
- » As formas normais mais populares são:
  - A primeira forma normal ou 1FN.
  
  - A segunda forma normal ou 2FN.
  
  - A terceira forma normal ou 3FN.
  
  - A quarta forma normal ou 4FN.

## NORMALIZAÇÃO

- » A forma normal é uma regra que deve ser satisfeita por uma entidade para que ela seja avaliada como uma tabela “projetada com exatidão”. São várias formas normais, com regras que vão se tornando mais rigorosas, com o objetivo de averiguar nas tabelas a existência de redundância ou dependências funcionais. No entanto, pelo menos quatro formas normais são consideradas essenciais para a construção de um bom projeto de banco de dados.

## NORMALIZAÇÃO 1FN e 2FN

- » Para atingir os objetivos da normalização, as tabelas precisam ter as seguintes propriedades:
  - Cada tabela deverá tratar de somente um único assunto, por exemplo: uma tabela com informações sobre remédio, não poderá ter informações de um médico.
  - O mesmo campo não poderá ser armazenado, desnecessariamente, em mais de uma tabela. Essa é uma garantia de que não será necessária a atualização do mesmo campo, em mais de uma tabela.

## NORMALIZAÇÃO 1FN e 2FN

- Os campos de uma tabela são dependentes da chave primária dessa tabela e de mais nenhum campo.
- A tabela deverá estar livre de anomalias de inserção, atualização e exclusão, garantindo a integridade e a consistência dos dados, por exemplo: na tabela Cliente será necessário informar a cidade de seu nascimento, não devemos deixar ele informar a cidade, mas escolher a cidade dentre as cidades previamente cadastradas ou por meio de uma busca do CEP (que trará o endereço completo).

## NORMALIZAÇÃO 1FN e 2FN

- » Para aplicar as regras da normalização, um dos alvos a ser observado são os campos (ou atributos) que fazem parte das tabelas. Podemos classificar os atributos como:
  - **Atributo simples ou atômico:** é o atributo que não é divisível, possui um sentido único, como o RG ou o CPF de uma pessoa, esses dois exemplos mostram que tanto o RG quanto o CPF não podem ser divididos em dois outros campos.
  - **Atributo composto:** é um atributo que pode ser dividido em várias partes, um bom exemplo é o endereço. Podemos dividir esse atributo em: rua, número, complemento, bairro.

## NORMALIZAÇÃO 1FN e 2FN

- **Atributo monovalorado:** é um atributo que possui apenas um valor para a tabela, como a matrícula de um aluno, esse número não poderá se repetir na tabela.
- **Atributo multivalorado:** é um atributo que pode receber mais de uma informação, o melhor exemplo é o telefone que pode receber mais de um valor.
- **Atributo derivado:** o valor desse tipo de atributo pode originar de outra tabela ou de outros campos. Digamos que para um cardiologista seja necessário saber a idade (em anos e dias). Podemos calculá-la a partir da data de nascimento e da data de atendimento no ato da consulta médica.
- **Atributo chave:** é o atributo escolhido ou criado para que possa indicar o registro (a linha) da tabela.

## NORMALIZAÇÃO 1FN e 2FN

- » A Primeira Forma Normal, ou simplesmente 1FN, possui a seguinte regra: uma tabela estará na Primeira Forma Normal se, e somente se, todos os seus atributos forem atômicos, não possuindo grupos repetitivos ou colunas que possuam mais de um valor e possua chave primária.
- » Ex: Tabela Pessoa.
- » **Pessoa** (#CPF, Nome, Sexo, Localização, Telefone)
- » **Pessoa** ('123456789', 'João', 'M', 'São Paulo, SP, Brasil', '(12) 98181-8181, (12) 89111-1111')
- » Tabela Corrigida:
- » **Pessoa** (#CPF, Nome, Sexo, Cidade, UF, País)
- » **Telefone** (#CPF, #Telefone)



## NORMALIZAÇÃO 1FN e 2FN

- » Outro exemplo: Para colocar um banco de dados no 1º nível de normalização, cada coluna da tabela deve conter um único valor e cada linha abranger as mesmas colunas.
- » A tabela deve ser assim:

CAMPO 1	CAMPO 2	CAMPO 3

- » Porém, nunca deve ser como a seguinte:

CAMPO 1	CAMPO 2		CAMPO 3	

## NORMALIZAÇÃO 1FN e 2FN

» Exemplo:

Cod_cliente	Nome	Telefones
12345	João	(12) 99192-0920 (12) 3221-1234
32145	José	(12) 3221-3221 (12) 99202-0202

» Para corrigir, crie uma tabela telefone.

## NORMALIZAÇÃO 1FN e 2FN

- » A Segunda Forma Normal, ou 2FN, deve obedecer a seguinte regra: uma tabela está na 2FN se, e somente se, estiver na 1FN e todas as suas colunas que não são chaves, dependam exclusivamente da chave primária (de toda a chave primária e não só de parte dela). Para estar na 2FN devemos aplicar as seguintes ações:
  - Identificar as colunas que não são funcionalmente dependentes da chave primária da tabela.
  - Remover o campo da tabela e criar uma nova tabela com esses dados.

## NORMALIZAÇÃO 1FN e 2FN

» **Ex:** Tabela Empregado\_Projeto.

<u>Emp_CPF</u>	<u>Proj_Cod</u>	Horas_Trabalho	Emp_Nome	Proj_Nome	Proj_Local
123456	1	100	João	MOD	SP
321123	2	120	Maria	BD	RJ
836212	3	160	José	LP	MG

» Mas qual é o problema de ter essas chaves?

456321	1	50	Julia	MOD	SP
--------	---	----	-------	-----	----

## NORMALIZAÇÃO 1FN e 2FN

» Tabela Corrigida:

<u>CPF</u>	Nome
123456	João
321123	Maria
836212	José
456321	Julia

<u>Cod</u>	Nome	Local
1	MOD	SP
2	BD	RJ
3	LP	MG

<u>Emp_CPF</u>	<u>Proj_Cod</u>	Horas_Trabalho
123456	1	100
321123	2	120
836212	3	160
456321	1	50

## EXERCÍCIO

» Aplique as formas normais na tabela abaixo. (SQL)

#idCliente	Cliente	Endereço	Nr Nota Fiscal	Valor da Nota
5412	Jonas Pedro	R. das Pedras, 15.	1456	R\$ 178,35
8532	Lena Luz	R. das Flores, 558.	1488	R\$ 587,30
4588	Caio Luiz	R. Beira Rio, 47.	1502	R\$ 358,00

## NORMALIZAÇÃO 3FN e 4FN

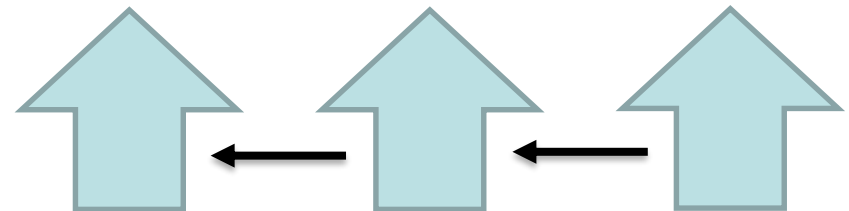
- » Uma tabela estará na Terceira Forma Normal somente se estiver na Segunda Forma Normal e todos os campos forem independentes, isso quer dizer que não poderá haver dependências funcionais entre os campos e todos os campos dependem da chave primária da tabela, ou seja, deve-se eliminar todos os campos dependentes de outras tabelas.
- » Para deixar uma tabela na Terceira Forma Normal é necessário eliminar todas as dependências transitivas, ou seja, eliminar todos os campos dependentes de outras tabelas.

## NORMALIZAÇÃO 3FN e 4FN

» Exemplo: Tabela Empregado\_Departamento

<u>Emp_CPF</u>	Emp_Nome	Emp_Sexo	Dep_Cod	Dep_Nome	Dep_Ger
123456	João	M	1	TI	Patricia
321123	Maria	F	2	ADM	Kaique
836212	José	M	2	ADM	Kaique
456321	Julia	F	1	TI	Patricia

» Dependência Transitiva





## NORMALIZAÇÃO 3FN e 4FN

- » A solução seria criar uma tabela DEPARTAMENTO e criar uma chave na tabela Empregado.
- » Exercício: Aplique a 3FN na tabela abaixo.

#cdFuncionário	Nome	idCargo	Descrição
148-9	Jane Anne	15	Professor I
721-4	Klaus Lins	16	Diretor
673-2	Sandra Costa	17	Professor II

## EXERCÍCIO

» Normalize a tabela abaixo (SQL):

idPedido	dataPeddo	codProduto	nomeProduto	qtde	valorUnitario	valorTotal
1	01/07/09	1234	HD 250GB	2	R\$ 100	R\$ 200
2	01/07/09	1235	HD 180GB	1	R\$ 80	R\$ 80
3	03/07/09	1235	HD 180GB	4	R\$ 80	R\$ 320
4	05/07/09	1234	HD 250GB	6	R\$ 100	R\$ 600

## NORMALIZAÇÃO 3FN e 4FN

- » Para realizar a 4FN uma tabela deverá estar na 3FN e ela somente estará na 4FN se não existir dependência multivalorada. Mas o que isso quer dizer? Dependência multivalorada é quando as informações inseridas nas tabelas podem ficar se repetindo e, claro, produzir redundâncias na tabela. Para evitar esse tipo de problema é melhor dividir a tabela.
- » Dependências multi-valoradas ocorrerem quando a presença de uma ou mais linhas em uma tabela implica a presença de uma ou mais outras linhas na mesma tabela.

## NORMALIZAÇÃO 3FN e 4FN

» Exemplo Tabela Disciplina

Professor	Disciplina	Titulação
Alex	Inteligência Artificial	Doutor
Alex	Lógica de Programação	Doutor
Juliana	Estrutura de Dados	Especialista
Juliana	Modelagem de Dados	Especialista

Professor	Disciplina
Alex	Inteligência Artificial
Alex	Lógica de Programação
Juliana	Estrutura de Dados
Juliana	Modelagem de Dados

Professor	Titulação
Alex	Doutor
Juliana	Especialista

BARBOZA, Fabrício Felipe Meleto; FREITAS, Pedro Henrique Chagas. **Modelagem e desenvolvimento de banco de dados**. Porto Alegre: SAGAH, 2018.

WERLICH, Claudia. **Modelagem de dados**. Londrina: Editora e Distribuidora Educacional S.A, 2018.

MANZANO, Jose Augusto Navarro Garcia. **Microsoft SQL Server 2016 Express Edition Interativo**. [S. l.]: ÉRICA, 2016.

MACHADO, Felipe Nery Rodrigues; ABREU, Mauricio Pereira de. **Projeto de Banco de Dados: Uma Visão Prática** - Edição Revisada e Ampliada. [S. l.]: ÉRICA, 2009.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Sistemas de Gerenciamento de Bancos de Dados. [S. l.]: Amgh Editora, 2011.

ALVES, WILLIAM PEREIRA. Banco de Dados. São Paulo: Saraiva, 2014

CARDOSO, VIRGÍNIA M.; CARDOSO, GISELLE CRISTINA. SISTEMA DE BANCO DE DADOS. São Paulo: Saraiva, 2013



Anhanguera