



Universidade Estadual da Paraíba

# Banco de Dados

## Normalização

---

Prof. Dr. Vladimir Costa Alencar

---

valencar@gmail.com

---

<https://www.valencar.com/>



# Normalização



- São Técnicas de racionalização das estruturas de dados de um sistema, eliminando redundâncias, problemas de manipulação e armazenamento.
- A normalização é necessária (embora não suficiente) a um bom projeto relacional.
- Através desse processo, pode-se gradativamente, substituir um conjunto de entidades e relacionamentos por um outro “purificado”
- O processo é purificado em relação às anomalias de atualização (inclusão, alteração e exclusão) as quais podem causar: **grupos repetitivos** de dados, redundâncias, **perdas de informações**, etc.



# Normalização

Um bom projeto de um esquema de entidades, e sua consequente conversão para um esquema relacional, segundo as regras vistas, praticamente deixa o esquema relacional *normalizado*.

Assim, utiliza-se a normalização somente para *validar* um projeto relacional

Esquemas de relação que não sejam “satisfatórios” são decompostos em esquemas menores que satisfaçam certas propriedades desejáveis.

# Normalização - Motivação

## Habilidade Esportivas

Identidade	Nome	Endereço	Habilidade
98764532	Mariana Ohata	Rua Ceará	Natação
98764532	Mariana Ohata	Rua Ceará	Corrida
98764532	Mariana Ohata	Rua Ceará	Ciclismo
98764532	Mariana Ohata	Rua Ceará	Voleibol
98764532	Mariana Ohata	Rua Ceará	Futebol

Esta tabela não está bem estruturada...

- 1) Se Mariana mudar de endereço? (anomalia de atualização)
- 2) Um novo esporte para a triatleta Mariana? (anomalia de inclusão)
- 3) Retirar Mariana do banco de dados? (anomalia de remoção)

# Normalização - Motivação



## Habilidade Esportivas (melhoramento)

Identidade	Nome	Endereço	Habilidade
98764532	Mariana Ohata	Rua Ceará	{Natação, Corrida, Ciclismo, Voleibol, Futebol}

A atributo Habilidade é Não Atômico???

O que podemos fazer, dentro do modelo relacional?

# Normalização - Motivação



## Esportista

Identidade	Nome	Endereço
98764532	Mariana Ohata	Rua Ceará

## Habilidades

Identidade	Habilidade
98764532	Natação
98764532	Corrida
98764532	Ciclismo
98764532	Voleibol
98764532	Futebol

A repetição da coluna Identidade é uma redundância necessária

-O processo de Normalização é realizado através das formas normais

1 FN → 2 FN → 3 FN (Forma Normal)



# Primeira Forma Normal (1FN)

Um esquema de relação  $R$  está na 1FN se **todos** os seus atributos forem **atômicos** e **monovalorados**, ou seja, não possuem valores que formam atributos compostos.

- Toda tabela deve ser “minimamente” normalizada (1FN)
- Toda tabela em 1FN: O valor de uma coluna de uma tabela é indivisível
  - Elimina os atributos multivalorados e compostos, permitindo apenas atributos atômicos.

Ex1. FUNCIONÁRIOS = {Código, Nome, Endereço, Cargo}  
ENDEREÇO = {Rua, Número, Bairro, Cidade, UF}

Para colocar na 1FN:

FUNCIONÁRIOS = {Código, Nome, Rua, Número, Bairro, Cidade, UF, Cargo}



# Primeira Forma Normal (1FN)

Ex2.

Funcionários = {Código, Nome, Cargo, Projeto, DataInício, DataFim}

Para colocar na 1FN faz:

Funcionários = {CódigoFunc, Nome, Cargo}

Func\_Proj = {CódigoFunc, Projeto, , DataInício, DataFim}



# Primeira Forma Normal (1FN)

## Tabela Empregados

Matrícula	Nome	CodCargo	NomeCargo	Cod Proj	DataFim	Horas
120	João	1	Programador	01	01/07/2009	37
120	João	1	Programador	08	20/02/2011	12
123	Hélio	1	Programador	01	01/07/2009	45
200	Hélio	1	Programador	08	20/02/2011	21
200	Hélio	1	Programador	12	01/01/2007	107
250	Gabriel	2	Analista	08	20/02/2011	10
250	Gabriel	2	Analista	12	01/01/2007	38
275	Silva	3	Projetista	01	01/07/2009	22
278	Abraão	2	Analista	12	01/01/2007	31
279	Carla	1	Programador	01	01/07/2009	27
279	Carla	1	Programador	12	01/01/2007	20
301	Ana	1	Programador	12	01/01/2007	16
306	Manoel	3	Projetista	17	15/11/2010	67



# Primeira Forma Normal (1FN)

Tabela:

Empregados(Matrícula, Nome, CodCargo, NomeCargo, CodProj, DataFim, Horas)

Chave primária → {Matrícula, CodProj}

Vimos que um dos objetivos da normalização é reduzir a redundância de dados, porém com a tabela anterior aumentamos a redundância ?!?!

Precisamos realizar outros passos de normalização para termos um bom projeto.

# Primeira Forma Normal (1FN)



A 1FN possui características indesejáveis! (anomalias)

- Inserção: não podemos inserir um empregado sem que este esteja alocado num projeto, nem um projeto sem que haja um empregado trabalhando nele (integridade de entidade)
- Remoção: se precisarmos remover um projeto, as informações de empregados que estiverem lotados apenas naquele projeto serão perdidas.
- Atualização: se um empregado for promovido de cargo teremos que atualizar os atributos CodCargo e NomeCargo em todas as tuplas nas quais aquele empregado está presente.

# Primeira Forma Normal (1FN)



## Conclusão:

Uma tabela em 1FN não evita, porém, anomalias de inclusão, atualização e remoção.

É preciso uma outra normalização mais “fina”, ou outras formas normais:

- Segunda Forma Normal (2FN)
- Terceira Forma Normal (3FN)
- Esta normalização “fina” utiliza o conceito de dependência funcional.



# Dependência Funcional

Dada uma entidade qualquer, dizemos que um atributo A (ou conjunto de atributos) é dependente funcional de outro atributo B, contido na mesma entidade se:

Cada valor de B existir nas linhas da entidade em que aparece, um único valor de A.

Ex.

Pedido (NumPedido, PrazoEntrega, Cliente, Endereço)

O atributo PrazoEntrega depende funcionalmente de NumPedido

$\text{NumPedido} \rightarrow \text{PrazoEntrega}$

NumPedido funcionalmente determina PrazoEntrega

PrazoEntrega é funcionalmente dependente de NumPedido



# Segunda Forma Normal (2FN)

Uma tabela está na Segunda Forma Normal (2FN) se ela é 1FN e todo atributo do complemento de uma chave candidata é totalmente funcionalmente dependente daquela chave.

Ex. Esportista (Identidade, Nome, Endereço, Habilidade)

Identidade	Nome	Endereço	Habilidade
98764532	Mariana Ohata	Rua Ceará	Natação
98764532	Mariana Ohata	Rua Ceará	Corrida
98764532	Mariana Ohata	Rua Ceará	Ciclismo
98764532	Mariana Ohata	Rua Ceará	Voleibol
98764532	Mariana Ohata	Rua Ceará	Futebol

Chaves Candidatas:  
{Identidade},  
{Nome, Endereço}

Complem. De Chave:  
{Nome, Endereço,  
Habilidade}

Identidade → Nome  
Identidade → Endereço  
Identidade NOT → Habilidade  
(Habilidade Não depende Funcionalmente de Nome)

{Nome, Endereço} → Identidade  
{Nome, Endereço} NOT →

Habilidade

Retirar da  
Entidade  
Esportista

# Segunda Forma Normal (2FN)



Ex. Esportista (Identidade, Nome, Endereço, Habilidade}

## Esportista

<u>Identidade</u>	Nome	Endereço
98764532	Mariana Ohata	Rua Ceará

## Habilidades

<u>Identidade</u>	Habilidade
98764532	Natação
98764532	Corrida
98764532	Ciclismo
98764532	Voleibol
98764532	Futebol

- Atualizar o Endereço de Mariana: sem anomalia
- Incluir uma nova habilidade esportiva para Mariana: sem anomalia



# Segunda Forma Normal (2FN)

Tabela Empregados (não normalizada)

Matrícula	Nome	CodCargo	NomeCargo	CodProj	DataFim	Horas
120	João	1	Programador	01	01/07/2009	37
120	João	1	Programador	08	20/02/2011	12
123	Hélio	1	Programador	01	01/07/2009	45
200	Hélio	1	Programador	08	20/02/2011	21
200	Hélio	1	Programador	12	01/01/2007	107
250	Gabriel	2	Analista	08	20/02/2011	10
250	Gabriel	2	Analista	12	01/01/2007	38
275	Silva	3	Projetista	01	01/07/2009	22
278	Abraão	2	Analista	12	01/01/2007	31
279	Carla	1	Programador	01	01/07/2009	27
279	Carla	1	Programador	12	01/01/2007	20
301	Ana	1	Programador	12	01/01/2007	16
306	Manoel	3	Projetista	17	15/11/2010	67



# Segunda Forma Normal (2FN)

## Empregados (normalizada- 2FN)

Matrícula	Nome	CodCargo	NomeCargo
120	João	1	Programador
120	João	1	Programador
123	Hélio	1	Programador
200	Hélio	1	Programador
200	Hélio	1	Programador
250	Gabriel	2	Analista
250	Gabriel	2	Analista
275	Silva	3	Projetista
278	Abraão	2	Analista
279	Carla	1	Programador
279	Carla	1	Programador
301	Ana	1	Programador
306	Manoel	3	Projetista

## Projeto

CodProj	DataFim
01	01/07/2009
08	20/02/2011
01	01/07/2009
08	20/02/2011
12	01/01/2007
08	20/02/2011
12	01/01/2007
01	01/07/2009
12	01/01/2007
01	01/07/2009
12	01/01/2007
12	01/01/2007
17	15/11/2010

## Alocação

Matrícula	CodProj	Horas
120	01	37
120	08	12
123	01	45
200	08	21
200	12	107
250	08	10
250	12	38
275	01	22
278	12	31
279	01	27
279	12	20
301	12	16
306	17	67



# Segunda Forma Normal (2FN)

Empregados (normalizada- 2FN)

Matrícula	Nome	CodCargo	NomeCargo
120	João	1	Programador
120	João	1	Programador
...	...	...	...

Projeto

CodProj	DataFim
01	01/07/2009
08	20/02/2011
01	01/07/2009
...	...

Alocação

Matrícula	CodProj	Horas
120	01	37
120	08	12
123	01	45
...	...	...

## Anomalias da 2FN:

- Inserção: Só podemos criar cargos se houver empregados designados para ele
- Remoção: se removermos um empregado que ocupa unicamente um cargo na empresa, Perderemos a informação desse cargo
- Atualização: se um cargo muda de nome, precisaremos mudar todas as tabelas em que este cargo aparece

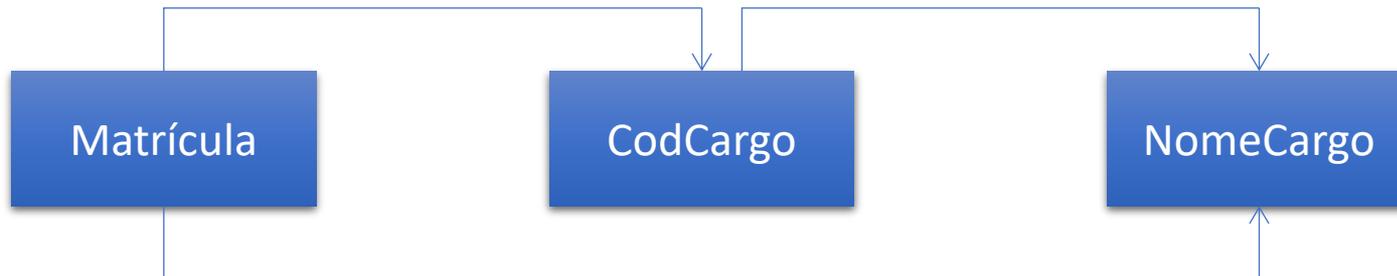


# Terceira Forma Normal (3FN)

- Envolve o conceito de **Dependência Transitiva**.

Suponha que tenhamos uma tabela com colunas A, B e C.

- Se a coluna C é funcionalmente dependente de B e B é funcionalmente dependente de A, então C é funcionalmente dependente de A.



**NomeCargo** é dependente transitivo de **Matrícula**

Uma tabela está na Terceira Forma Normal (3FN) se e somente se, estiver em 2FN e todos os atributos não-chave forem dependentes não-transitivos da chave primária.



# Terceira Forma Normal (3FN)

## Empregados

<u>Matrícula</u>	Nome	CodCargo
120	João	1
120	João	1
123	Hélio	2
200	Hélio	2
200	Hélio	2
250	Gabriel	2
250	Gabriel	2
275	Silva	3
278	Abraão	2
279	Carla	1
279	Carla	1
301	Ana	1
306	Manoel	3

## Cargo

<u>CodCargo</u>	NomeCargo
1	Programador
2	Analista
2	Analista
3	Projetista
2	Analista
1	Programador
1	Programador
1	Programador
3	Projetista

Removendo a dependência Transitiva, obteremos, As tabelas Empregado e Cargo.



# Terceira Forma Normal (3FN)

Empregados

<u>Matricula</u>	Nome	CodCargo
120	João	1
120	João	1
123	Hélio	2
200	Hélio	2
200	Hélio	2
...	...	...

Cargo

<u>CodCargo</u>	NomeCargo
1	Programador
2	Analista
3	DBA
4	Gerente
5	CIO
...	...

- “Uma relação está em 3FN se todas as colunas da tabela são dependentes da chave inteira e nada além da chave.”
- A 3FN elimina as características mais potencialmente indesejáveis dos dados que estão em 2FN ou 1FN.



# Quarta Forma Normal (4FN)

## - Eliminação da Multiplicidade de Fatos Multivalorados

Ex. Um paciente pode necessitar de vários exames e dispor de vários planos de saúde

- O paciente Pedro dispõe de dois planos de saúde e necessita fazer dois exames
- Ele quer validar os exames usando os 2 planos de saúde)

### Solicitação de exame

Paciente	PlanoDeSaúde	Exame
Pedro Cordeiro	Unimed	Testorena
Pedro Cordeiro	PBPrev	PSA (Próstata)
Pedro Cordeiro	Unimed	Hemograma (Sangue)
Pedro Cordeiro	PBPrev	Holter (Cardiologia)



# Quarta Forma Normal (4FN)

## Solicitação de exame

<u>Paciente</u>	PlanoDeSaúde	Exame
Pedro Cordeiro	Unimed	PSA (Próstata)
Pedro Cordeiro	PBPrev	PSA (Próstata)
Pedro Cordeiro	Unimed	Hemograma (Sangue)
Pedro Cordeiro	PBPrev	Hemograma (Sangue)

A tabela está normalizada na 3FN (a chave é composta pela união de 3 atributos)

→ Está difícil manipular mais de um fato multivalorado na mesma tabela

Resolvemos a situação fazendo a projeção dos subconjuntos que constituem cada fato multivalorado



# Quarta Forma Normal (4FN)

## Solicitação de exame

<u>Paciente</u>	<u>Exame</u>
Pedro Cordeiro	Testorena
Pedro Cordeiro	PSA (Próstata)
Pedro Cordeiro	Hemograma (Sangue)
Pedro Cordeiro	Holter (Cardiologia)

## Seguridade

<u>Paciente</u>	<u>PlanoDeSaúde</u>
Pedro Cordeiro	Unimed
Pedro Cordeiro	PBPrev



# Quinta Forma Normal (5FN)

A 5FN trata de casos em que a informação pode ser reconstruída de pedaços menores de informação que podem ser mantidos com menor redundância.

## Ex. Agente-Companhia-Produtos

Agente	Companhia	Produtos
Mario Alcântara	Ford	Carro
Mário Alcântara	Fiat	Trator
Mário Alcântara	Fiat	Carro
Denise Sousa	Ford	Carro

Queremos armazenar um registro de qual Agente vende qual Produto para Qual Companhia.



# Quinta Forma Normal (5FN)

Agente	Companhia	Produto
Mario Alcântara	Ford	Carro
Mário Alcântara	Fiat	Trator
Mário Alcântara	Fiat	Carro
Denise Sousa	Ford	Carro



Transformação para 5FN  
(decomposição em registros menores)

Agente	Companhia
Mario Alcântara	Ford
Mário Alcântara	Fiat
Denise Sousa	Ford

Agente	Produto
Mario Alcântara	Carro
Mário Alcântara	Trator
Denise Sousa	Carro

Companhia	Produtos
Ford	Carro
Ford	Trator
Fiat	Carro
Fiat	Trator



# Exercício - Normalização

Considere as seguintes relações e suas dependências funcionais.

Livro = {ISBN, autor, instituicaoAutor, titulo, editora, tipo, preço}

ISBN  $\rightarrow$  autor, titulo, editora, tipo

autor  $\rightarrow$  instituiçãoAutor

tipo  $\rightarrow$  preço

titulo  $\rightarrow$  ISBN, editora

Normalize.



## Exercício - Normalização

Considere as seguintes relações e suas dependências funcionais.

Peça = {codigo, fornecedor, endereçoFornecedor, qtdEstoque, peso, valor}

código, fornecedor  $\rightarrow$  valor;

fornecedor  $\rightarrow$  endereçoFornecedor;

código  $\rightarrow$  peso, qtdEstoque;

Normalize.



# Exercício - Normalização

Considere as seguintes relações e suas dependências funcionais.

Pedido (idPedido, dataPedido, codProduto, nomeProduto, qtde, valorUnitario, valorTotal)

Dependências funcionais:

IdPedido  $\rightarrow$  dataPedido

idPedido, codProduto  $\rightarrow$  qtde, valorTotal

codProduto  $\rightarrow$  nomeProduto, valorUnitário

Normalize.



# Exercício - Normalização

Normalize.

aluno ( nro\_aluno, cod\_depto, nome\_depto, sigla\_depto, cod\_orient, nome\_orient, fone\_orient, cod\_curso )

As seguintes dependências funcionais devem ser garantidas na normalização:

- $\text{cod\_depto} \rightarrow \{\text{nome\_depto}, \text{sigla\_depto}\}$
- $\text{cod\_orient} \rightarrow \{\text{nome\_orient}, \text{fone\_orient}\}$
- $\text{nro\_aluno} \rightarrow \{\text{cod\_depto}, \text{cod\_orient}, \text{cod\_curso}\}$

Observações adicionais:

- um aluno somente pode estar associado a um departamento
- um aluno cursa apenas um único curso
- um aluno somente pode ser orientado por um único orientador

# Exercício - Normalização



Normalize.

vendedor ( nro\_vend, nome\_vend, sexo\_vend, { cliente (nro\_cli, nome\_cli, end\_cli ) } )

As seguintes dependências funcionais devem ser garantidas na normalização:

- nro\_vend → nome\_vend, sexo\_vend
- nro\_cli → nome\_cli, end\_cli

Observações adicionais:

- um vendedor pode atender diversos clientes, e um cliente pode ser atendido por diversos vendedores