

# Normalização

**Prof. Dr. Luiz Cláudio Demes da Mata Sousa**

**E-mail: [claudio.demes@ufpi.edu.br](mailto:claudio.demes@ufpi.edu.br)**



## Objetivo

“Gerar um conjunto de esquemas para armazenar informações sem redundância”. (Silberschatz)

Verifique a relação abaixo e identifique as redundâncias.

Identidade	Nome	Endereço	Habilidade
8795835	Édson Arantes	Ponta da Praia	Futebol
8795835	Édson Arantes	Ponta da Praia	Voleibol
8795835	Édson Arantes	Ponta da Praia	Basquete
8795835	Édson Arantes	Ponta da Praia	Atletismo
8795835	Édson Arantes	Ponta da Praia	Tênis

Considerações importantes:

1. Se Édson muda de nome?
2. Exclusão do registro de Édson?
3. Édson passou a praticar um novo esporte!



## Normalização

Idealmente, o armazenamento deveria ser como mostrado abaixo:

Identidade	Nome	Endereço	Habilidade
8795835	Édson Arantes	Ponta da Praia	Futebol
			Voleibol
			Basquete
			Atletismo
			Tênis

Problema:

O Modelo Relacional não aceita esse tipo de armazenamento.



# Normalização

Solução:

Identidade	Nome	Endereço
8795835	Édson Arantes	Ponta da Praia
...	...	...

1

n

Identidade	Esporte
8795835	Futebol
8795835	Voleibol
8795835	Basquetebol
8795835	Atletismo
8795835	Tênis

- Foi eliminada a redundância?
- A normalização trouxe vantagens?
- A normalização trouxe desvantagens?
- Existem regras de normalização?



# 1ª FN – 1ª Forma Normal

## 1ª Forma Normal (1ª FN)

- Uma Entidade está na 1ª FN se, e somente se, todos os seus valores contém **domínios atômicos**. Colunas com mais de um valor também devem ser eliminadas.

Exemplo:

Não pode ocorrer um domínio **endereço** com subdomínios: **cidade**, **número** e **logradouro**. Uma relação normalizada (1ªFN) teria *cidade*, *logradouro* e *número*, como domínios separados.



# 1FN – 1ª Forma Normal

**Itens repetitivos** (listas) devem ser eliminados.

Relação não normalizada:

empregado	<b>dependentes</b>	salário
Pedro	José, Paulo	3500,00
Carlos	Afonso	1300,00

Solução:

empregado	<b>dependentes</b>	salário
Pedro	José	3500,00
Pedro	Paulo	3500,00
Carlos	Afonso	1300,00



## Relação na 1ª Forma Normal

Matrícula	Nome	Cod Cargo	NomeCargo	CodProj	DataFim	Horas
120	João	1	Programador	01	17/07/95	37
120	João	1	Programador	08	12/01/96	12
121	Hélio	1	Programador	01	17/07/95	45
121	Hélio	1	Programador	08	12/01/96	21
121	Hélio	1	Programador	12	21/03/96	107
270	Gabriel	2	Analista	08	12/01/96	10
270	Gabriel	2	Analista	12	21/03/96	38
273	Silva	3	Projetista	01	17/07/95	22
274	Abraão	2	Analista	12	21/03/96	31
279	Carla	1	Programador	01	17/07/96	27
279	Carla	1	Programador	08	12/01/96	20
279	Carla	1	Programador	12	21/03/96	51
301	Ana	1	Programador	12	21/03/96	16
306	Manoel	3	Projetista	17	21/03/96	67

**Ao estar na 1ª FN, ainda existe redundância. São necessários outros passos de normalização para obtermos um bom projeto.**



## 1ªFN – Anomalias

Matrícula	Nome	Cod Cargo	NomeCargo	CodProj	DataFim	Horas
-----------	------	-----------	-----------	---------	---------	-------

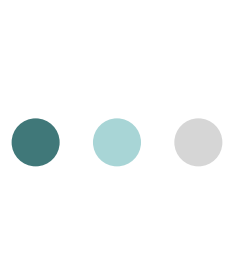
**Inserção:** não podemos inserir um **empregado** sem que este esteja alocado a um projeto, nem inserir um **projeto** sem que haja um empregado trabalhando nele (integridade da entidade).

**Remoção:** se precisarmos remover um projeto, as informações dos empregados que estiverem trabalhando 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.

Sendo assim, faz-se necessário refinar a normalização utilizando o conceito de **dependência funcional**.





# Dependência Funcional

Dada uma entidade  $E$ , um atributo  $B$  de  $E$  é funcionalmente dependente do atributo  $A$  de  $E$  se, e somente se, cada valor de  $A$  em  $E$  está associado a apenas um valor de  $B$  em  $E$ .

Em outras palavras -  $A$  **determina univocamente**  $B$ .

$A \rightarrow B$
-------------------

lê - se:

$A$  **funcionalmente determina**  $B$

$B$  é **funcionalmente dependente** de  $A$

$B$  é **função** de  $A$

Para cada valor de  $A$  só existe um valor de  $B$ . Exemplo: CPF  $\rightarrow$  Nome do Cliente.

## 2FN – 2ª Forma Normal

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

- Ela está na 1ª FN e;
- Todo atributo não chave for totalmente funcionalmente dependente da chave primária. Isto é, se não houver dependências parciais na chave.

<u>Matricula</u>	Nome	CodCargo	NomeCargo	<u>CodProj</u>	DataFim	Horas
120	João	1	Programador	01	17/07/95	37
120	João	1	Programador	08	12/01/96	12
121	Hélio	1	Programador	01	17/07/95	45
121	Hélio	1	Programador	08	12/01/96	21
121	Hélio	1	Programador	12	21/03/96	107
270	Gabriel	2	Analista	08	12/01/96	10

Chave = {Matricula, CodProj}



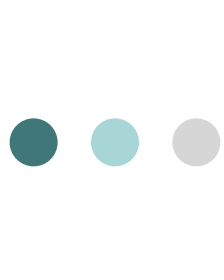
## Relação na 2ª FN

Matrícula	Nome	Cod Cargo	NomeCargo	CodProj	DataFim	Horas
-----------	------	-----------	-----------	---------	---------	-------

Matrícula	Nome	Cod Cargo	NomeCargo
120	João	1	Programador
121	Hélio	1	Programador
270	Gabriel	2	Analista

CodProj	DataFim
01	17/07/95
08	12/01/96
01	17/07/95
08	12/01/96
12	21/03/96
08	12/01/96

Matrícula	CodProj	Horas
120	01	37
120	08	12
121	01	45
121	08	21
121	12	107
270	08	10



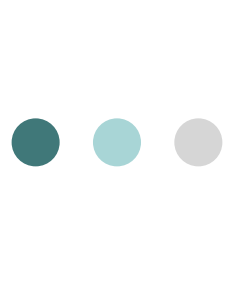
## 2ªFN – Anomalias

Matrícula	Nome	Cod Cargo	NomeCargo
-----------	------	-----------	-----------

**Inserção:** Só podemos criar **cargos** se houver **empregados** designados para ele.

**Remoção:** Se removermos o único empregado que ocupa um determinado cargo na empresa (ex. Analista), perderemos a informação sobre este cargo.

**Atualização:** Se um cargo muda de nome precisaremos alterar todas as linhas nas quais este cargo aparece.



## 3ª FN – 3ª Forma Normal

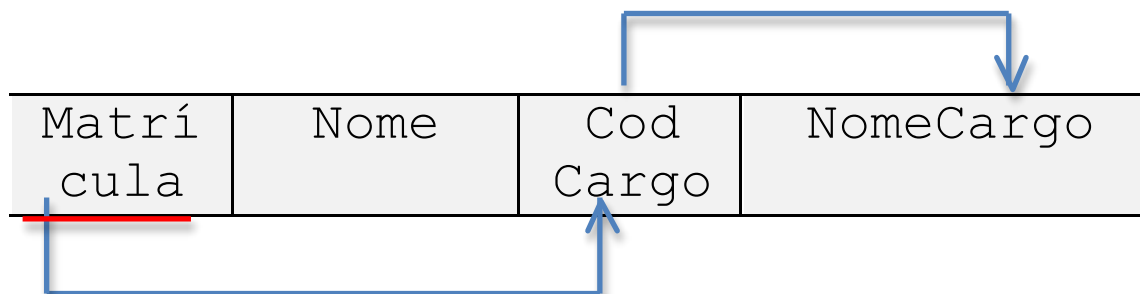
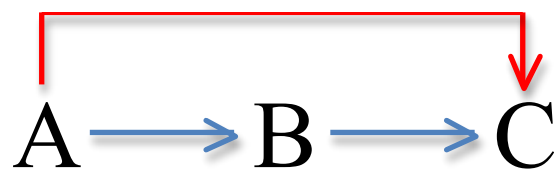
Uma entidade E está na 3ª FN se, e somente se:

- estiver na 2ª FN e
- todos os atributos não-chave de E forem dependentes não-transitivos da chave primária.

Em resumo:

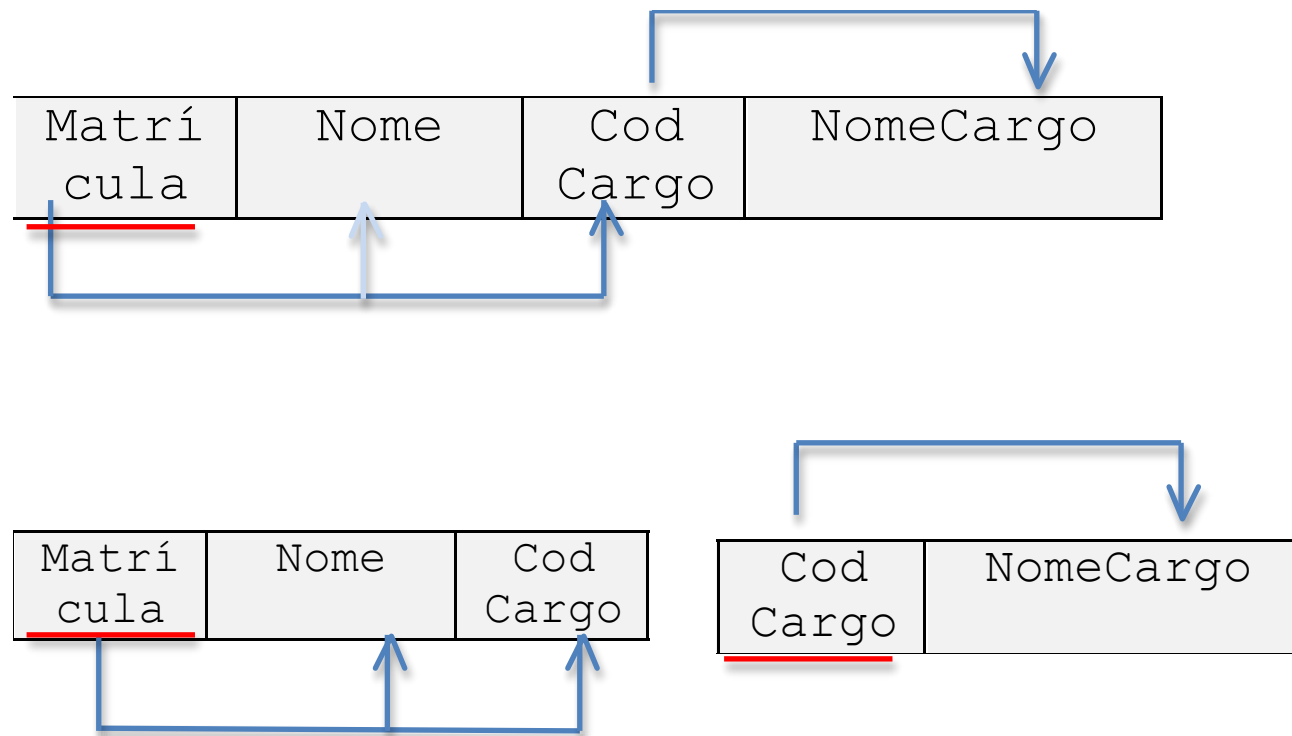
Todas as colunas da tabela são funcionalmente dependentes da chave inteira e de nenhum outro atributo além da chave.

# Dependência Transitiva



Eliminação de dependências transitivas, através da decomposição por projeção da relação original.

# Dependência Transitiva





# Dependência Transitiva

Removendo a dependência transitiva, obteremos as seguintes tabelas:

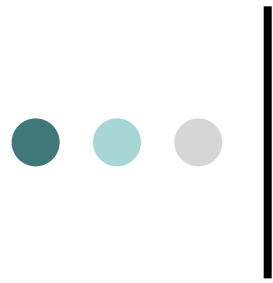
## **Empregado**

Matrícula	Nome	CodCargo
120	João	1
121	Hélio	1
270	Gabriel	2
273	Silva	3
274	Abraão	2
279	Carla	1
301	Ana	1
306	Manuel	3

## **Cargo**

CodCargo	Nome
1	Programador
2	Analista
3	Projetista





# Metodologia de Normalização

Passo 1: Tome projeções de tabelas na 1ªFN para eliminar todas as dependências funcionais não-totais. O resultado é uma coleção de tabelas 2ªFN.

Passo 2: Tome projeções das tabelas obtidas no passo 1 para eliminar todas as dependências transitivas. O resultado é uma coleção de relações na 3ªFN.