

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO RIO GRANDE DO NORTE
CAMPUS JOÃO CÂMARA

NORMALIZAÇÃO

Nickerson Fonseca Ferreira
nickerson.ferreira@ifrn.edu.br

Normalização

2

- Conjunto de regras aplicadas numa tabela com o objetivo de corrigir possíveis erros de projeto.
- Existem diversas regras de normalização
- Objetivos para aplicar a normalização em tabelas:
 - Eliminar redundância de dados
 - Eliminar estruturas do MER inexistentes no modelo lógico
- As regras são chamadas Formas Normais:
 - 1^º Forma Normal (1FN)
 - 2^º Forma Normal (2FN)
 - 3^º Forma Normal (3FN)

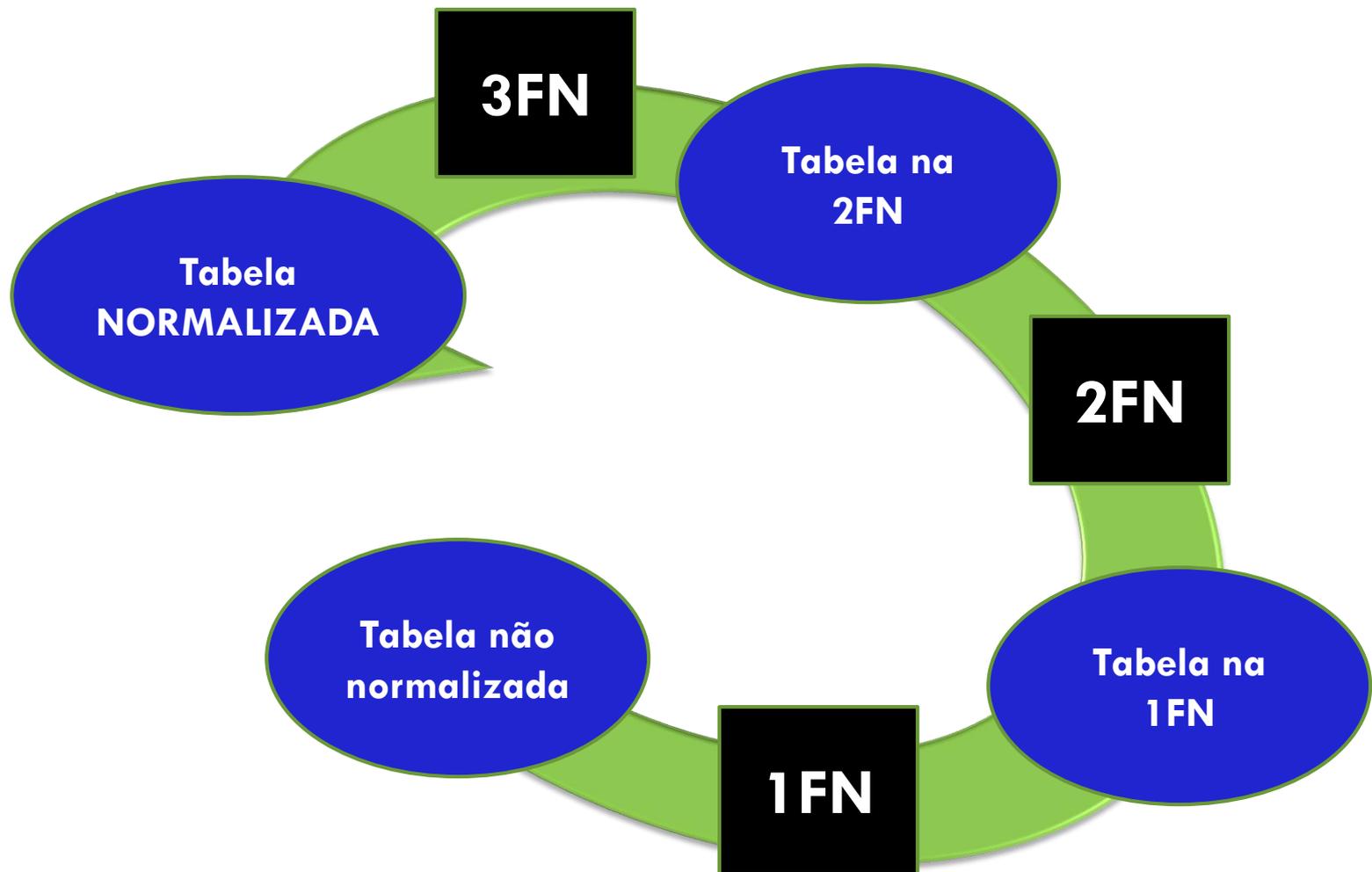
Normalização

3



Normalização

4



1ª Forma Normal (1FN)

5

- Dizemos que uma tabela está na primeira forma normal, quando ela não possui:
 - Atributos multivalorados
 - Atributos compostos
- Alguns autores se referem a estes atributos como tabelas aninhadas

QUAIS OS PROBLEMAS EM TER ESSE TIPO DE ATRIBUTO ?

1ª Forma Normal (1FN)

6

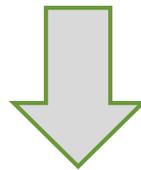
CodProj	Tipo	Descr	Emp					
			CodEmp	Nome	Cat	Sal	DataIni	Tempo
LSC001	Novo Desenv.	Sistema de Estoque	2146	João	A1	4	1/11/13	24
			3145	Sílvio	A2	4	2/10/12	24
			6126	José	B1	9	3/10/11	18
			1214	Carlos	A2	4	4/10/13	18
			8191	Mário	A1	4	1/11/12	12
PAG02	Manutenção	Sistema de RH	8192	Mariana	A1	4	1/05/12	12
			4112	João	A2	4	4/01/11	24

1ª Forma Normal (1FN)

7

- Cada atributo composto ou multivalorado deve ser transformado numa nova tabela.
- Adiciona uma chave estrangeira nessa nova tabela.

Projeto (CodProj, Tipo, Descricao, (CodEmp, Nome, Cat, Sal, DataIni, Tempo))



Projeto (CodProj, Tipo, Descricao)
EmpProj(CodEmp, CodProj, Nome, Cat, Sal, DataIni, Tempo)

Dependência Funcional

- Para avançar para a 2FN e a 3FN é necessário conhecer o conceito de dependência funcional.
- É a principal ferramenta de avaliação para identificar se o agrupamento de atributos de uma tabela é apropriado.
 - Evitando redundância de dados
 - Inconsistências
 - Perda de dados em operações de remoções ou alterações

Dependência Funcional

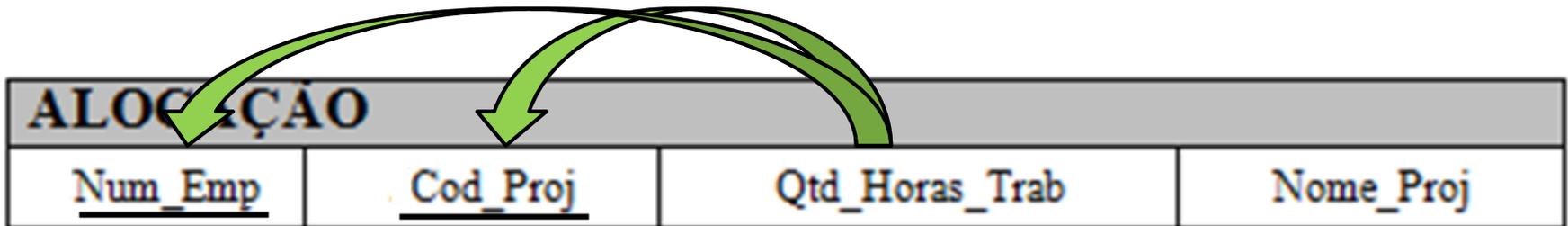
9

- Existem 3 tipos:
 - Dependência Funcional Total
 - Dependência Funcional Parcial
 - Dependência Funcional Transitiva

Dependência Funcional Total

10

- Os atributos não chave de uma tabela têm que depender totalmente da chave primária e somente dela.
- Ex: Uma determinada tabela possui sua chave primária composta pelos atributos A e B. Logo, C será dependente funcional total se e somente se C depender funcionalmente de A e B.



Dependência Funcional Parcial

11

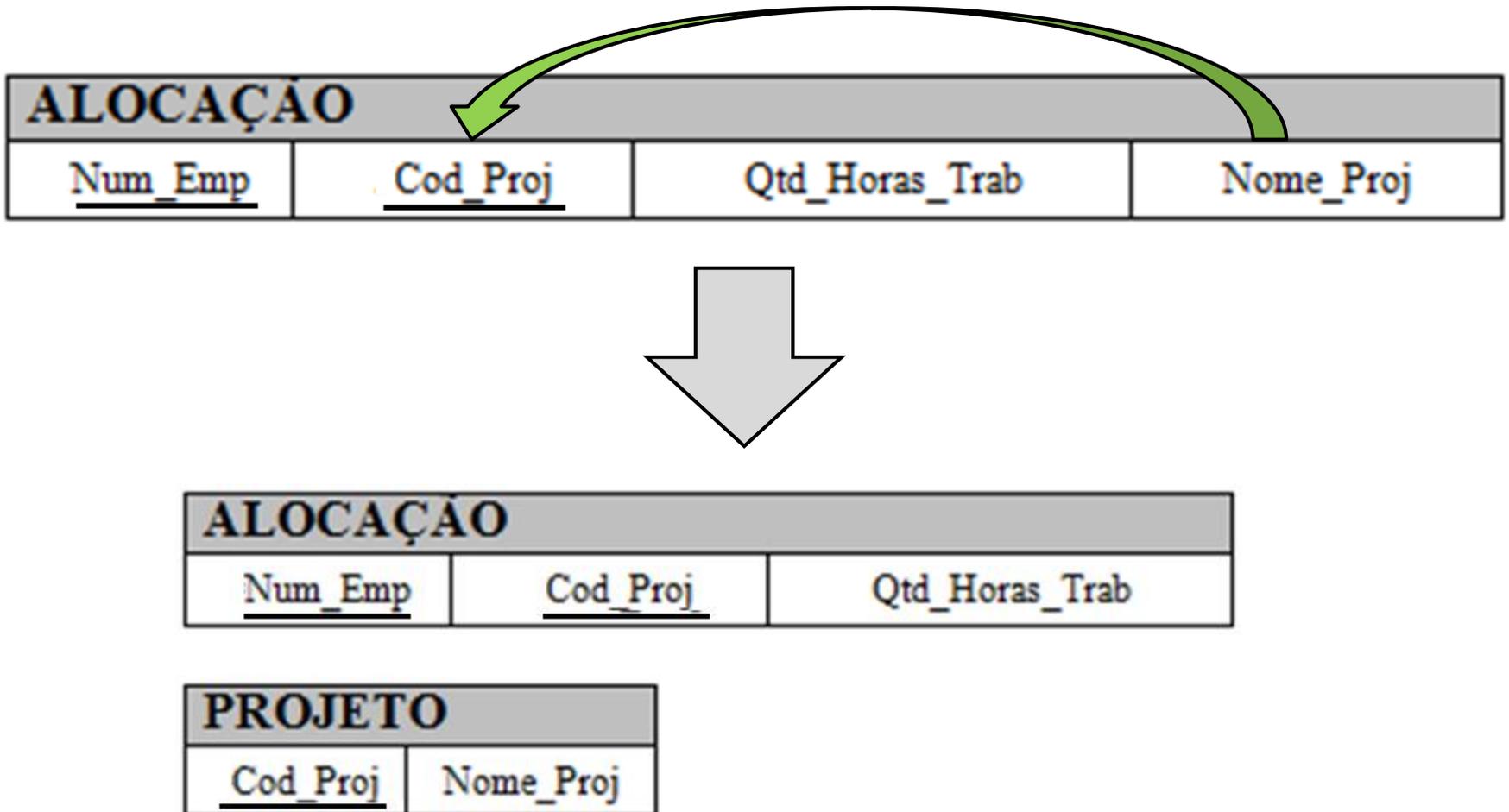
- Os atributos não chave de uma tabela dependem **de parte** da chave primária.
- Ex: Uma determinada tabela possui sua chave primária composta pelos atributos A e B. Logo, C será dependente funcional parcial se e somente se C depender funcionalmente de A ou B.

ALOCAÇÃO			
<u>Num_Emp</u>	<u>Cod_Proj</u>	Qtd_Horas_Trab	Nome_Proj

Dependência Funcional Parcial

12

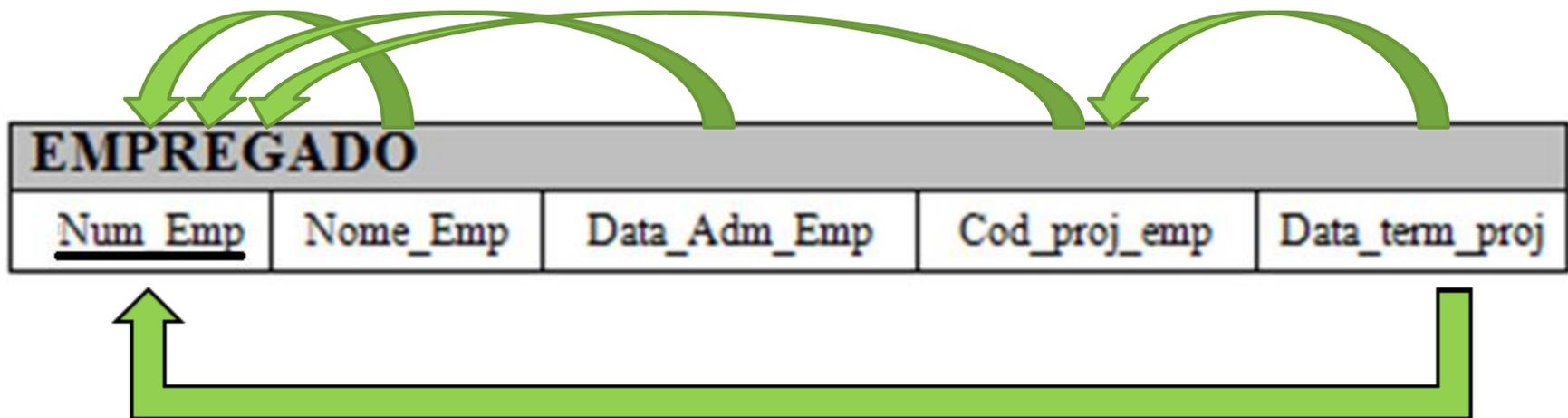
- Solução para dependência funcional parcial



Dependência Funcional Transitiva

13

- O atributo C é dependente funcional transitivo de A se C é funcionalmente dependente de B e B funcionalmente dependente de A, na mesma tabela.



Dependência Funcional Transitiva

14

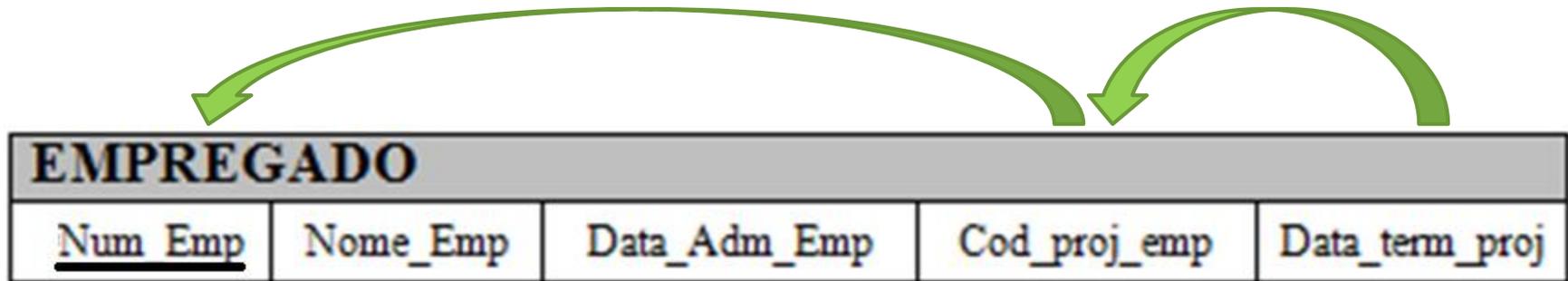
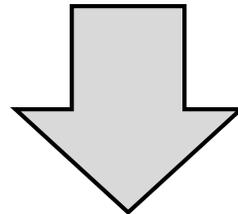


Diagram illustrating a table with five columns: Num_Emp, Nome_Emp, Data_Adm_Emp, Cod_proj_emp, and Data_term_proj. A green curved arrow points from the last column (Data_term_proj) to the first column (Num_Emp), indicating a functional dependency.

EMPREGADO				
<u>Num_Emp</u>	Nome_Emp	Data_Adm_Emp	Cod_proj_emp	Data_term_proj



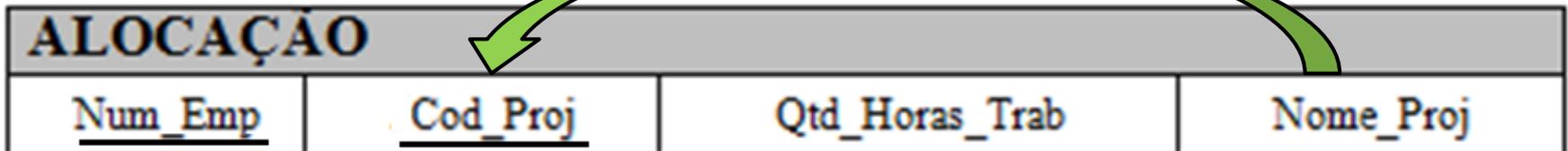
EMPREGADO			
<u>Num_Emp</u>	Nome_Emp	Data_Adm_Emp	Cod_proj_emp

PROJETO	
<u>Cod_Proj_Emp</u>	Data_term_proj

2ª Forma Normal (2FN)

15

- Uma tabela está na 2FN se ela já estiver na 1FN.
- Todo atributo que não for chave primária for dependente funcional **TOTAL**.
- Ou seja, não deve existir dependência funcional parcial.



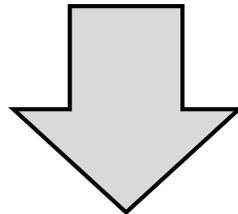
The diagram shows a table with a header row and a data row. The header row is shaded gray and contains the text 'ALOCAÇÃO'. The data row has four columns: 'Num_Emp', 'Cod_Proj', 'Qtd_Horas_Trab', and 'Nome_Proj'. The first two columns, 'Num_Emp' and 'Cod_Proj', are underlined. A green curved arrow starts from the 'Nome_Proj' column and points to the 'Cod_Proj' column, indicating a functional dependency from 'Nome_Proj' to 'Cod_Proj'.

ALOCAÇÃO			
<u>Num_Emp</u>	<u>Cod_Proj</u>	Qtd_Horas_Trab	Nome_Proj

2ª Forma Normal (2FN)

16

ALOCAÇÃO			
<u>Num_Emp</u>	<u>Cod_Proj</u>	Qtd_Horas_Trab	Nome_Proj



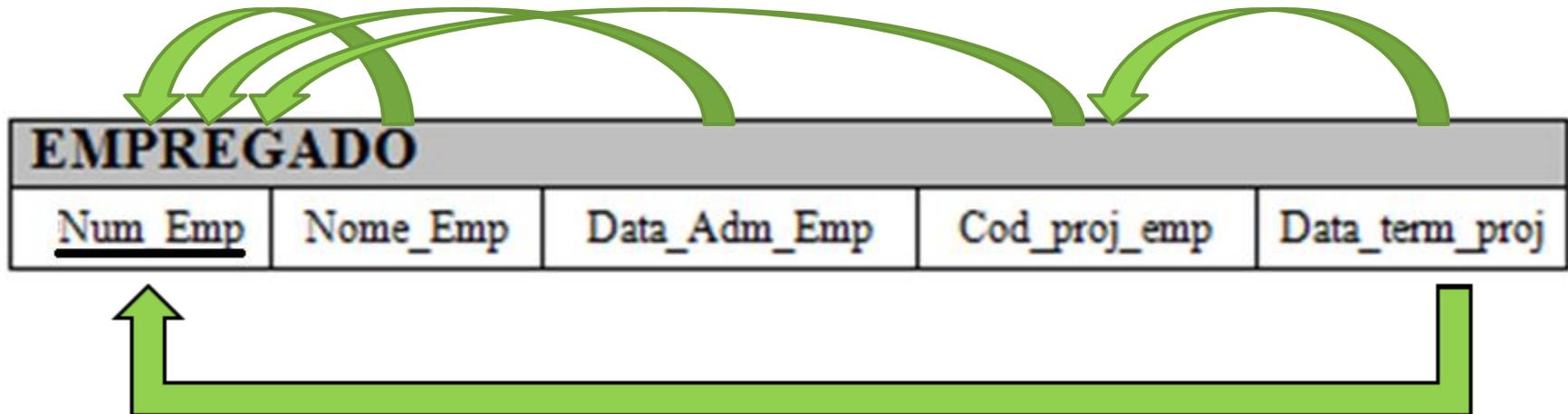
ALOCAÇÃO		
<u>Num_Emp</u>	<u>Cod_Proj</u>	Qtd_Horas_Trab

PROJETO	
<u>Cod_Proj</u>	Nome_Proj

3ª Forma Normal (3FN)

17

- Uma tabela está na 3FN se ela já estiver na 2FN.
- E **NÃO** deve existir dependência funcional transitiva entre atributos não chave.



3ª Forma Normal (3FN)

18

Diagram illustrating a table with 5 columns: **EMPREGADO**, Num_Emp, Nome_Emp, Data_Adm_Emp, Cod_proj_emp, and Data_term_proj. A green curved arrow points from the **Cod_proj_emp** column to the **Data_term_proj** column, indicating a functional dependency.

EMPREGADO				
<u>Num_Emp</u>	Nome_Emp	Data_Adm_Emp	Cod_proj_emp	Data_term_proj

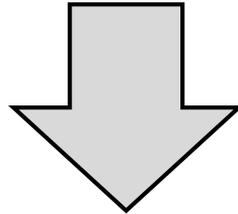


Diagram illustrating the first decomposed table with 4 columns: **EMPREGADO**, Num_Emp, Nome_Emp, Data_Adm_Emp, and Cod_proj_emp.

EMPREGADO			
<u>Num_Emp</u>	Nome_Emp	Data_Adm_Emp	Cod_proj_emp

Diagram illustrating the second decomposed table with 2 columns: **PROJETO**, Cod_Proj_Emp, and Data_term_proj.

PROJETO	
<u>Cod_Proj_Emp</u>	Data_term_proj

Exercício

19

1. vendedor (nro_vend, nome_vend, sexo_vend, { nro_cli, nome_cli, end_cli, data_atend })

As seguintes dependências funcionais devem ser garantidas na normalização:

- nro_vend → nome_vend, sexo_vend
- nro_cli → nome_cli, end_cli
- nro_vend , nro_cli → data_atend

Exercício

IFRN - Campus João Câmara

Curso: Técnico em Informática Códido do curso: 0037

Aluno: Maria Joaquina Matricula: 123456 Status: Cursando

Histórico

Disciplina (codigo)	Professor - Código	Nota	Faltas	Situação
Análise de sistemas (AN001)	Roberto Carlos - 001	7,5	7	Aprovado
Matemática (MA002)	Jandira - 002	8.0	4	Aprovado
Inglês (IN101)	Junior Villas - 003	4.5	0	Reprovado

Aluno (matricula, nome, status, codCurso, nomeCurso, {CodDisciplina, nomeDisciplina, codProf, nomeProf, nota, faltas, situacao})

Matricula -> nome, Status, CodCurso

CodCurso -> nomeCurso

CodDisciplina -> nomeDisciplina

CodProf -> nomeProf

codDisciplina, matricula -> nota, faltas, situação, codProf