

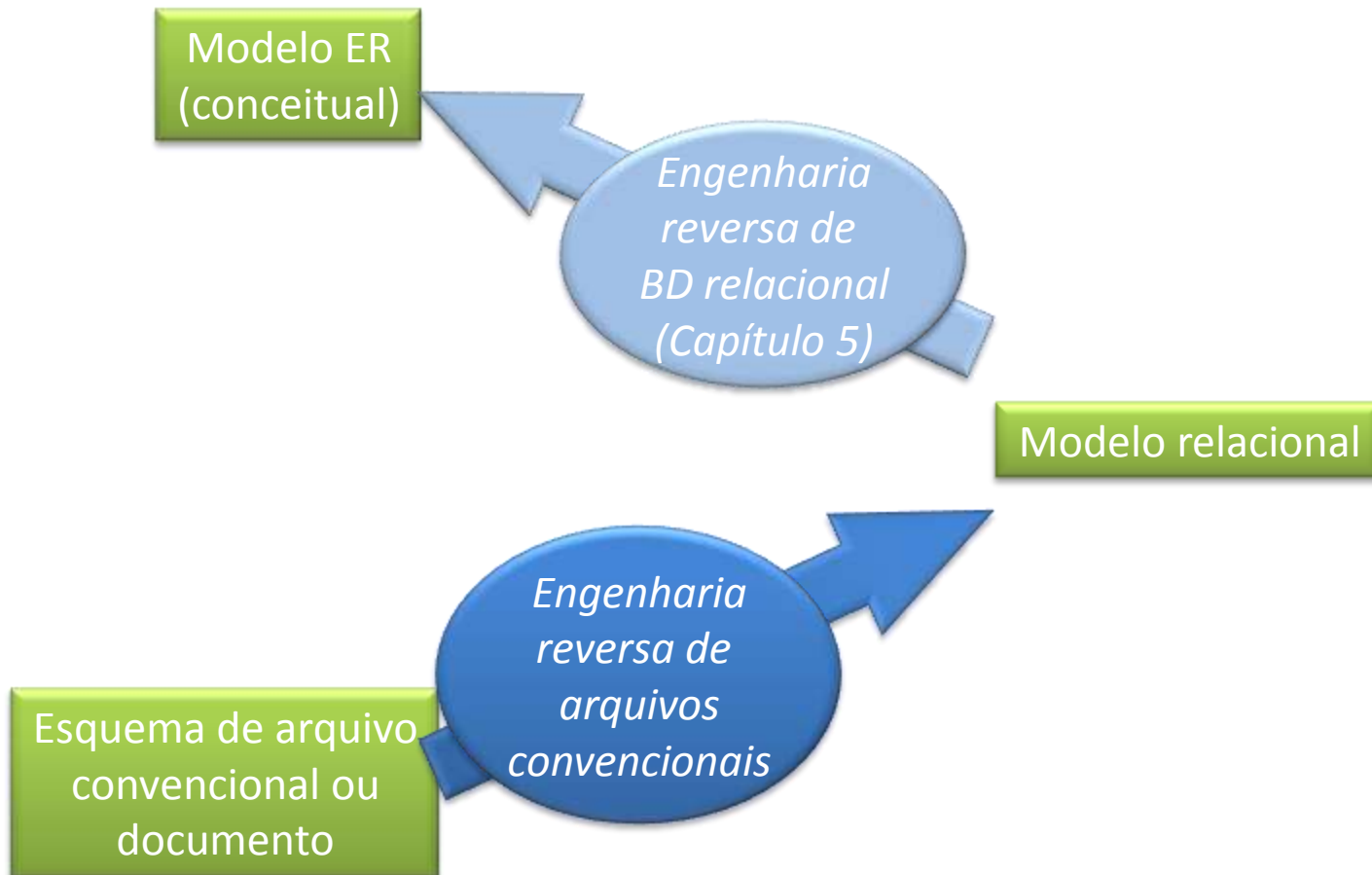
# **Engenharia reversa de arquivos e documentos**

## **Capítulo 6**

# Engenharia reversa de arquivos e documentos



# Engenharia reversa de BD relacional



# Engenharia reversa de arquivos e normalização

- Entrada do processo:
  - qualquer conjunto de dados para os quais se **disponha** de uma **descrição**:
    - documentos,
    - arquivos manuais,
    - arquivos convencionais em computador,
    - bancos de dados gerenciados por SGBD não relacional,
    - ...

# Engenharia reversa de arquivos e normalização - motivação

- Sistemas legados:
  - Raramente documentados;
  - Necessidade de modelo ER:
    - Manutenção,
    - Migração para outro tipo de BD,
    - Integração com outros BDs.

# Engenharia reversa

## passo #1

- Normalização:
  - Processo que transforma um esquema de dados qualquer em um modelo relacional.

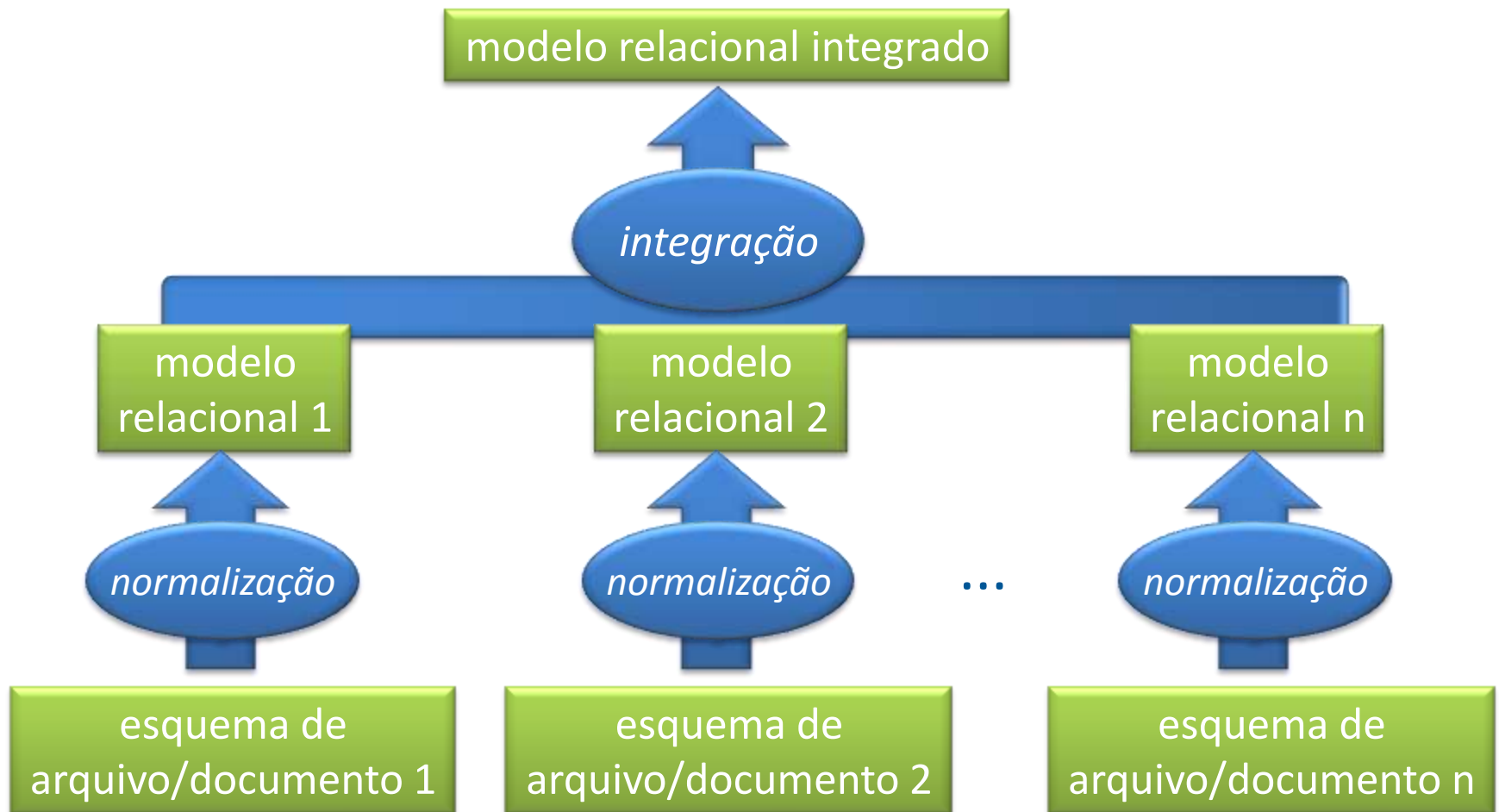


# Engenharia reversa - processo

- Normalização é executada para todos esquemas de documentos disponíveis.



# Engenharia reversa - integração

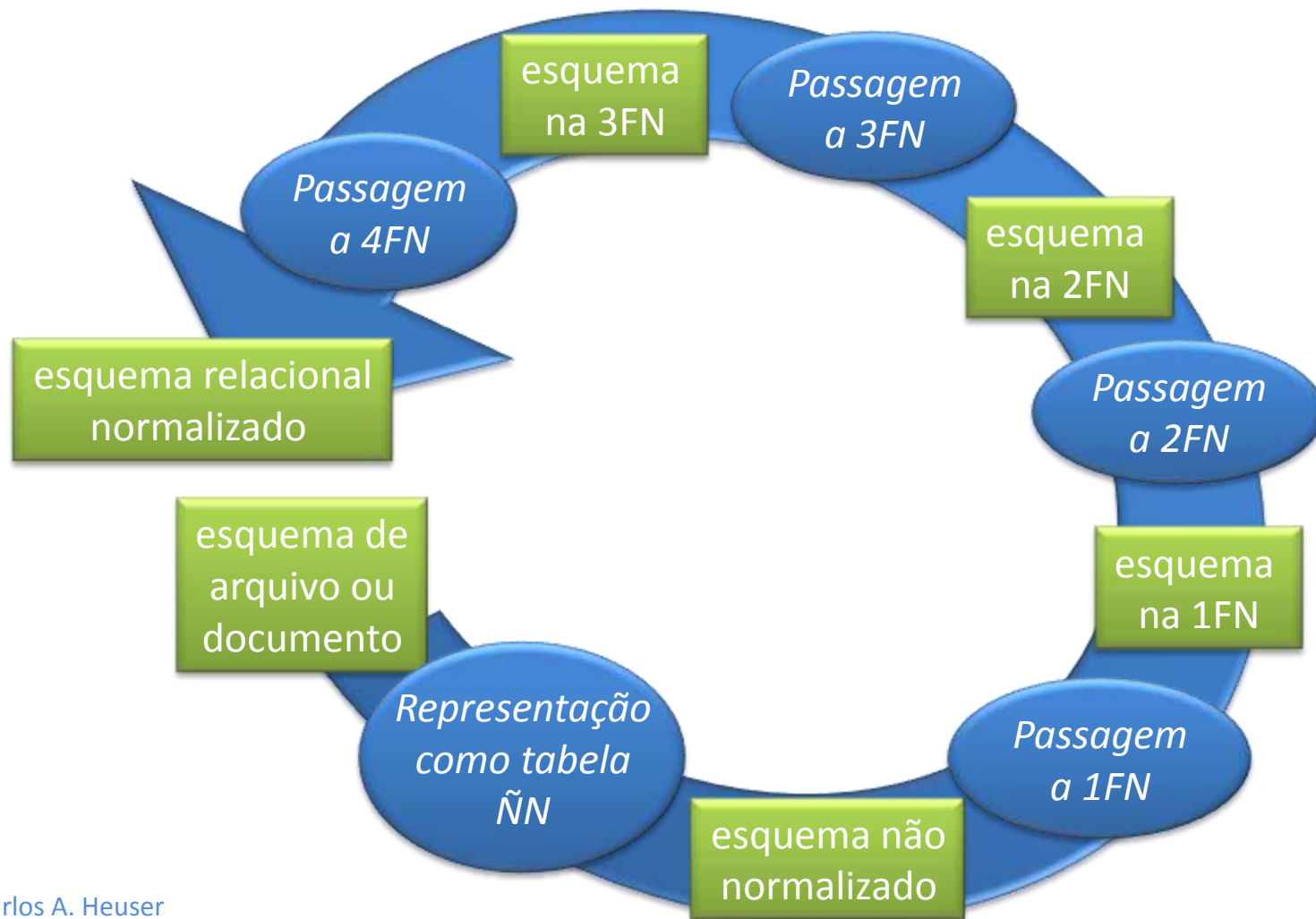




# Normalização Objetivo

- Reagrupar informações para:
  - eliminar redundâncias de dados.
- Reagrupar informações para:
  - eliminar estruturas inexistentes no modelo ER (atributos multivalorados).

# Normalização passos



# Documento exemplo para normalização

## RELATÓRIO DE ALOCAÇÃO A PROJETO

CÓDIGO DO PROJETO: LSC001 TIPO: Novo Desenv.

DESCRIÇÃO: Sistema de Estoque

CÓDIGO DO EMPREGADO	NOME	CATEGORIA FUNCIONAL	SALÁRIO	DATA DE INÍCIO NO PROJETO	TEMPO ALOCADO AO PROJETO
2146	João	A1	4	1/11/91	24
3145	Sílvio	A2	4	2/10/91	24
6126	José	B1	9	3/10/92	18
1214	Carlos	A2	4	4/10/92	18
8191	Mário	A1	4	1/11/92	12

CÓDIGO DO PROJETO: PAG02 TIPO: Manutenção

DESCRIÇÃO: Sistema de RH

CÓDIGO DO EMPREGADO	NOME	CATEGORIA FUNCIONAL	SALÁRIO	DATA DE INÍCIO NO PROJETO	TEMPO ALOCADO AO PROJETO
8191	Mário	A1	4	1/05/93	12
4112	João	A2	4	4/01/91	24
6126	José	B1	9	1/11/92	12

# Normalização – passo #1



# Tabela não normalizada

- Tabela *não-normalizada* ou tabela *não-primeira-forma-normal*:
  - possui uma ou mais *tabelas aninhadas*

# Tabela aninhada

- Tabela *não-normalizada* ou tabela *não-primeira-forma-normal*:
  - possui uma ou mais *tabelas aninhadas*

Tabela aninhada  
ou grupo repetido  
ou coluna multi-valorada  
ou coluna não atômica  
=  
coluna que ao invés de conter valores  
atômicos, contém tabelas aninhadas

# Tabela não normalizada

- Tabela *não-normalizada* ou tabela *não-primeira-forma-normal*:
  - possui uma ou mais *tabelas aninhadas*.
- Abreviatura: ÑN

# Documento exemplo na forma ÑN

CódProj	Tipo	Descr	Emp					
			CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	Novo Desenv.	Sistema de Estoque	2146	João	A1	4	1/11/91	24
			3145	Sílvio	A2	4	2/10/91	24
			6126	José	B1	9	3/10/92	18
			1214	Carlos	A2	4	4/10/92	18
			8191	Mário	A1	4	1/11/92	12
PAG02	Manute nção	Sistema de RH	8191	Mário	A1	4	1/05/93	12
			4112	João	A2	4	4/01/91	24
			6126	José	B1	9	1/11/92	12



# Tabela aninhada

CódProj	Tipo	Descr	Emp					
			CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	Novo Desenv.	Sistema de Estoque	2146	João	A1	4	1/11/91	24
			3145	Sílvio	A2	4	2/10/91	24
			6126	José	B1	9	3/10/92	18
			1214	Carlos	A2	4	4/10/92	18
			8191	Mário	A1	4	1/11/92	12
PAG02	Manute nç	Sistem	101	Mário	A1	4	1/05/93	12
				João	A2	4	4/01/91	24
				José	B1	9	1/11/92	12

tabela  
aninhada

# Tabela ÑN Esquema

```
Proj (CodProj, Tipo, Descr,  
      (CodEmp, Nome, Cat, Sal, DataIni, TempAl)  
      )
```

# Esquema de arquivo em Pascal

```
type  reg_aluno= record
      cod_al: integer;
      nome_al: char_60;
      ingressos_cursos_al: array [1..10] of record
        cod_curso: integer;
        semestre_ingresso: integer
      end;
      disciplinas_cursadas_al: array [0..200] of record
        cod_disc: integer;
        semestres_cursados: array [1..20] of record
          semestre_disc: integer;
          nota_disc: integer
        end
      end
    end;
      arq_aluno= file of reg_aluno;
```

# Esquema de arquivo COBOL - parcial

```
FD      Arq-Alunos
01      Reg-Al.
03      Cod-Al
03      Nome-Al
03      Ingr-Cursos-al OCCURS 1 TO 10
05          Cod-Curso
05          Sem-ingresso
03      Disc-Curs-Al OCCURS 0 TO 200
05          Cod-Disc
05          Sem-Cursado OCCURS 1 TO 20
07              Sem-Disc-Cursada
07              Nota-Disc
```

## Esquema ÑN para arquivos exemplo

```
Arq-Alunos (Cod-Al, Nome-Al,  
            (Cod-Curso, Sem-ingresso),  
            (Cod-Disc,  
              (Sem-Disc-Cursada,  
                Nota-Disc)))
```

# Representação em esquema não normalizada

- Nenhuma transformação é feita no modelo do documento.
- Apenas é usada outra notação.
- Notação **independe do tipo** de documento/arquivo usado como entrada do processo de normalização.

# Forma normal

- Regra que uma tabela deve obedecer para ser considerada “bem projetada”.
- Há diversas formas normais, cada vez mais rígidas, para verificar tabelas relacionais.
- Aqui tratadas:
  - primeira forma normal (1FN),
  - segunda forma normal (2FN),
  - terceira forma normal (3FN),
  - quarta forma normal (4FN).

# Passagem a 1FN





# Primeira forma normal (1FN)

*primeira forma normal (1FN)*

=

diz-se que uma tabela está na primeira forma normal, quando ela não contém tabelas aninhadas

## Passagem à 1FN - alternativas

- Para chegar a 1FN há duas alternativas:
  1. Construir uma única tabela com redundância de dados.
  2. Construir uma tabela para cada tabela aninhada.

## Passagem à 1FN – alternativa #1

- Uma tabela na qual os dados das linhas externas à tabela aninhada são repetidos para cada linha da tabela aninhada.

ÑN:

```
Proj (CodProj, Tipo, Descr,  
      (CodEmp, Nome, Cat, Sal, DataIni, TempAl)  
      )
```



1FN:

```
ProjEmp (CodProj, Tipo, Descr, CodEmp, Nome,  
          Cat, Sal, DataIni, TempAl)
```

- Dados do projeto aparecem repetidos para cada empregado do projeto.

## Passagem à 1FN -alternativa #2

- Cria-se:
  1. uma tabela referente a própria tabela que está sendo normalizada e
  2. uma tabela para cada tabela aninhada

ÑN:

```
Proj (CodProj, Tipo, Descr,  
      (CodEmp, Nome, Cat, Sal, DataIni, TempAl)  
      )
```



1FN:

```
Proj (CodProj, Tipo, Descr)
```

```
ProjEmp (CodProj,CodEmp, Nome, Cat, Sal, DataIni, TempAl)
```

## Passagem à 1FN - alternativas

- Primeira alternativa (tabela única) é a **correta**.
- Segunda alternativa - decompor uma tabela em várias tabelas:
  - podem ser **perdidas relações** entre informações.
- Ver exercício 6.17 do livro.

## Passagem à 1FN - alternativas

- Para fins práticos:
  - preferimos a segunda alternativa (*decomposição de tabelas*)
- Quando houver diversas tabelas aninhadas, eventualmente com diversos níveis de aninhamento, fica difícil visualizar a tabela na 1FN na alternativa de tabela única.

## Passagem à 1FN – passo #1

1. Criar uma tabela na 1FN referente a tabela não normalizada.
  - A chave primária da tabela na 1FN é idêntica a chave da tabela ÑN .

# Passagem à 1FN

## criar tabela referente a tabela externa

ÑN:

(CodProj, Tipo, Descr,  
(CodEmp, Nome, Cat, Sal, DataIni, TempAl))

1FN:

(CodProj, Tipo, Descr)



## Passagem à 1FN – passo #2

2. Para cada tabela aninhada:

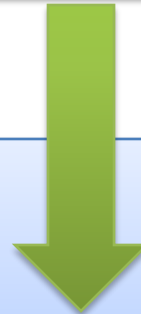
- criar uma tabela composta pelas seguintes colunas:
  - a) a chave primária de cada uma das tabelas na qual a tabela em questão está aninhada;
  - b) as colunas da própria tabela aninhada.

# Passagem à 1FN

## criar tabelas referentes a tabela aninhada

ÑN:

(CodProj, Tipo, Descr,  
(CodEmp, Nome, Cat, Sal, DataIni, TempAl))



1FN:

(CodProj, Tipo, Descr)  
(CodProj, CodEmp, Nome, Cat,  
Sal, DataIni, TempAl)

## Passagem à 1FN - passo #3

3. Definir, na 1FN, as **chaves primárias** das tabelas que correspondem a tabelas **aninhadas**.

# Passagem à 1FN – tabela externa definição de chave primária

ÑN:

(CodProj, Tipo, Descr,  
(CodEmp, Nome, Cat, Sal, DataIni, TempAl))

Tabela de nível mais  
externo:  
basta transcrever a chave  
primária

1FN

(CodProj, Tipo, Descr)  
(CodProj, CodEmp, Nome, Cat,  
Sal, DataIni, TempAl)

# Passagem à 1FN – tabelas aninhadas definição de chave primária

ÑN:

(CodProj, Tipo, Descr,  
(CodEmp, Nome, Cat, Sal, DataIni, TempAl))

qual é a chave  
primária desta  
tabela?

1FN:

(CodProj, Tipo, Descr)  
(CodProj, CodEmp, Nome, Cat,  
Sal, DataIni, TempAl)

# Passagem à 1FN – tabelas aninhadas definição de chave primária

ÑN:

(CodProj, Tipo, Descr)  
(CodEmp, Nome, Cat, Sal, DataIni, TempAl))

pergunta a ser feita:

“um valor de **CodEmp** (chave da tabela origem) aparece uma única ou várias vezes no documento?”

1FN:

(CodProj, Tipo, Descr)  
(CodProj, CodEmp, Nome, Cat,  
Sal, DataIni, TempAl)

# Documento exemplo para normalização

## RELATÓRIO DE ALOCAÇÃO A PROJETO

CÓDIGO DO PROJETO: LSC001			TIPO: Novo Desenv.		
DESCRIÇÃO: Sistema de Esto					
CÓDIGO DO EMPREGADO	NOME	CATEGORIA FUNCIONAL	TEMPO ALOCADO AO PROJETO		
2146	João	A1			24
3145	Sílvio	A2		2/10/91	24
6126	José	B1	9	3/10/92	18
1214	Carlos	A2	4	4/10/92	18
8191	Mário	A1	4	1/11/92	12
CÓDIGO DO PROJETO: PAG02			TIPO: Manutenção		
DESCRIÇÃO: Sistema de RH					
CÓDIGO DO EMPREGADO	NOME	CATEGORIA FUNCIONAL	SALÁRIO	DATA DE INÍCIO NO PROJETO	TEMPO ALOCADO AO PROJETO
8191	Mário	A1	4	1/05/93	12
4112	João	A2	4	4/01/91	24
6126	José	B1	9	1/11/92	12

um empregado pode trabalhar em vários projetos

um empregado pode trabalhar em vários projetos

# Documento exemplo para normalização

## RELATÓRIO DE ALOCAÇÃO A PROJETO

CÓDIGO DO PROJETO: LSC001

TIPO: Novo Desenv.

um valor de **CodEmp**  
(chave da tabela origem)  
aparece várias vezes no  
documento

bloque

			SALÁRIO	DATA DE INÍCIO NO PROJETO	TEMPO ALOCADO AO PROJETO
			4	1/11/91	24
3145	Sílvio	A2	4	2/10/91	24
6126	José	B1	9	3/10/92	18
1214	Carlos	A2	4	4/10/92	18
8191	Mário	A1	4	1/11/92	12

CÓDIGO DO PROJETO: PAG02

TIPO: Manutenção

DESCRIÇÃO: Sistema de RH

CÓDIGO DO EMPREGADO	NOME	CATEGORIA FUNCIONAL	SALÁRIO	DATA DE INÍCIO NO PROJETO	TEMPO ALOCADO AO PROJETO
8191	Mário	A1	4	1/05/93	12
4112	João	A2	4	4/01/91	24
6126	José	B1	9	1/11/92	12



# Passagem à 1FN – tabelas aninhadas definição de chave primária

ÑN:

(CodProj, Tipo, Descr, (CodEmp, Nome, Cat, Sal, DataIni, TempAl))

Um valor de **CodEmp**  
aparece várias vezes:

É necessário **CodProj**  
para distinguir as várias  
aparições

1FN:

(CodProj, Tipo, Descr)  
(CodProj, CodEmp, Nome, Cat,  
Sal, DataIni, TempAl)

# Passagem à 1FN – tabelas aninhadas

## definição de chave primária

ÑN:

(CodProj, Tipo, Descr,  
(CodEmp, Nome, (TempA1)))

Caso um empregado trabalhasse em único projeto (um valor de **CodEmp** aparece uma vez ao máximo)

1FN:

(CodProj, Tipo, Descr)  
(CodProj, CodEmp, Nome, Cat,  
Sal, DataIni, TempA1)

## Passagem à 1FN - exemplo

Proj:

<u>CódProj</u>	Tipo	Descr
LSC001	Novo Desenv.	Sistema de Estoque
PAG02	Manutenção	Sistema de RH

ProjEmp:

<u>CódProj</u>	<u>CodEmp</u>	Nome	Cat	Sal	DataIni	TempAl
LSC001	2146	João	A1	4	1/11/91	24
LSC001	3145	Sílvio	A2	4	2/10/91	24
LSC001	6126	José	B1	9	3/10/92	18
LSC001	1214	Carlos	A2	4	4/10/92	18
LSC001	8191	Mário	A1	4	1/11/92	12
PAG02	8191	Mário	A1	4	1/05/93	12
PAG02	4112	João	A2	4	4/01/91	24
PAG02	6126	José	B1	9	1/11/92	12

## Passagem à 1FN outro exemplo

ÑN:

Arq-Candidatos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso,  
(Cod-Cand, Nome-Cand, Escore-Cand)  
)

# Passagem à 1FN decomposição em tabelas

ÑN:

Arq-Candidatos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso,  
(Cod-Cand, Nome-Cand, Escore-Cand)  
)



1FN:

Cursos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso)

# Passagem à 1FN decomposição em tabelas

ÑN:

Arq-Candidatos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso,  
(Cod-Cand, Nome-Cand, Escore-Cand)  
)



1FN:

Cursos	(Cod-Curso, Nome-Curso, Numero-Vagas-Curso)
Candidatos	(Cod-Curso, Cod-Cand, Nome-Cand, Escore-Cand)

# Passagem à 1FN

## definição da chave primária

ÑN:

Arq-Candidatos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso,  
(Cod-Curso, Nome-Cand, Escore-Cand)  
)

Tabela de nível mais externo:  
basta transcrever a chave

1FN:

Cursos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso)  
Candidatos (Cod-Curso, Cod-Cand, Nome-Cand, Escore-Cand)

# Passagem à 1FN

## definição da chave primária

ÑN:

Arq-Candidatos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso,  
(Cod-Cand, Nome-Cand, Escore-Cand)  
)



1FN:

Cursos	( <u>Cod-Curso</u> , Nome-Curso, Numero-Vagas-Curso)
Candidatos	(Cod-Curso, <u>Cod-Cand</u> , Nome-Cand, Escore-Cand)



# Passagem à 1FN

## definição da chave primária

ÑN:

Arq-Candidatos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso,  
(Cod-Cand, Nome-Cand, Escore-Cand)  
)

Um valor de **Cod-Cand**  
aparece uma única vez.

1FN:

Cursos	( <u>Cod-Curso</u> , Nome-Curso, Numero-Vagas-Curso)
Candidatos	(Cod-Curso, <u>Cod-Cand</u> , Nome-Cand, Escore-Cand)

# Passagem a 1FN

## exemplo Pascal/COBOL

ÑN:

Arq-Alunos (Cod-Al, Nome-Al,  
                  (Cod-Curso, Sem-ingresso)  
                  (Cod-Disc,  
                          (Sem-Disc-Cursada, Nota-Disc)))

1FN:

Alunos	(Cod-Al, Nome-Al)
AlunoCurso	(Cod-Al, Cod-Curso, Sem-ingresso)
AlunoDisc	(Cod-Al, Cod-Disc)
AlunoDiscSem	(Cod-Al, Cod-Disc, Sem-Disc-Cursada, Nota-Disc)

# Passagem a 1FN

## exemplo Pascal/COBOL

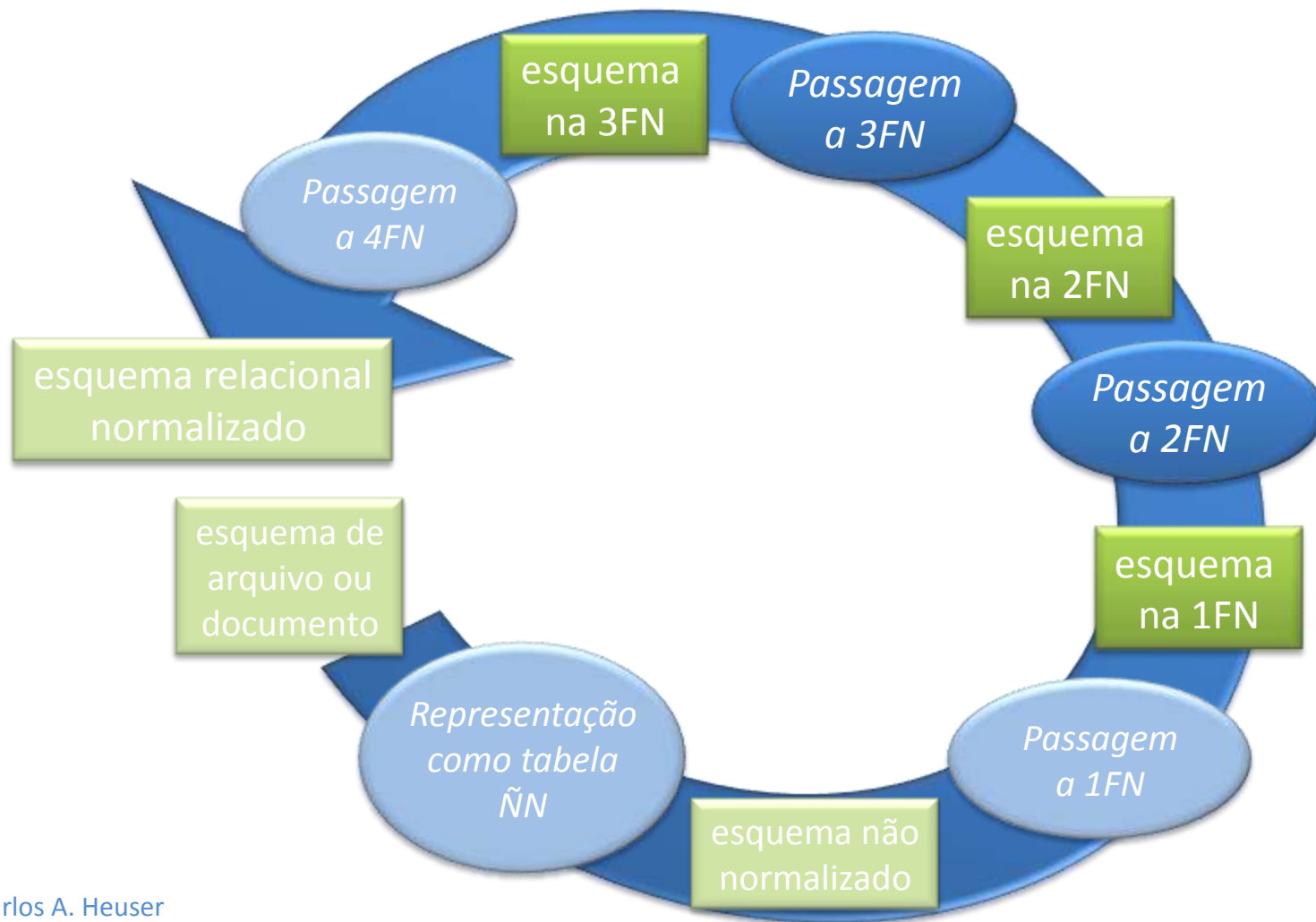
ÑN:

Arq-Alunos (Cod-Al, Nome-Al,  
                                  (Cod-Curso, Sem-ingresso)  
                                  (Cod-Disc,  
                                  (Sem-Disc-Cursada, Nota-Disc)))

1FN:

Alunos	( <u>Cod-Al</u> , Nome-Al)
AlunoCurso	( <u>Cod-Al</u> , <u>Cod-Curso</u> , Sem-ingresso)
AlunoDisc	( <u>Cod-Al</u> , <u>Cod-Disc</u> )
AlunoDiscSem	( <u>Cod-Al</u> , <u>Cod-Disc</u> , <u>Sem-Disc-Cursada</u> , Nota-Disc)

# Passagem às 2FN e 3FN



# Dependência funcional

- Para entender 2FN e 3FN:
  - é necessário compreender o conceito de *dependência funcional*.

Em uma tabela relacional, diz-se que uma coluna  $C_2$  *depende funcionalmente* de uma coluna  $C_1$  (ou que a coluna  $C_1$  *determina* a coluna  $C_2$ ) quando, em todas linhas da tabela, para cada valor de  $C_1$  que aparece na tabela, aparece o mesmo valor de  $C_2$ .

## Exemplo de dependência funcional

...	Código	...	Salário	...
...	E1	...	10	...
...	E3	...	10	...
...	E1	...	10	...
...	E2	...	5	...
...	E3	...	10	...
...	E2	...	5	...
...	E1	...	10	...

Código → Salário

# Dependências funcionais - exemplos

A	B	C	D
B	5	2	20
C	4	2	15
B	6	7	20
B	5	2	20
C	2	2	15
C	4	2	15
A	10	5	18
A	12	3	18
A	10	5	18
B	5	2	20
C	4	2	15
A	10	5	18
C	4	2	15

# Dependências funcionais - exemplos

A	B	C	D
B	5	2	20
C	4	2	15
B	6	7	20
B	5	2	20
C	2	2	15
C	4	2	15
A	10	5	18
A	12	3	18
A	10	5	18
B	5	2	20
C	4	2	15
A	10	5	18
C	4	2	15

Dependência funcional  
inexistente na tabela:

~~A  $\rightarrow$  B~~



# Dependências funcionais - exemplos

A	B	C	D
B	5	2	20
C	4	2	15
B	6	7	20
B	5	2	20
C	2	2	15
C	4	2	15
A	10	5	18
A	12	3	18
A	10	5	18
B	5	2	20
C	4	2	15
A	10	5	18
C	4	2	15

Dependência funcional  
existente na tabela

$A \rightarrow D$

# Dependências funcionais - exemplos

A	B	C	D
B	5	2	20
C	4	2	15
B	6	7	20
B	5	2	20
C	2	2	15
C	4	2	15
A	10	5	18
A	12	3	18
A	10	5	18
B	5	2	20
C	4	2	15
A	10	5	18
C	4	2	15

Uma coluna pode  
depender funcionalmente  
de uma combinação de  
mais de uma coluna

$(A,B) \rightarrow C$

# Passagem às 2FN e 3FN



## Segunda forma normal - 2FN

- Objetiva eliminar um certo tipo de redundância de dados.

- Exemplo

(CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)

- Dados referentes a empregados (Nome, Cat e Sal) são
  - redundantes, para os empregados que trabalham em mais de um projeto.

## Dados redundantes na 1FN

ProjEmp:

<u>CódProj</u>	<u>CodEmp</u>	Nome	Cat	Sal	DataIni	TempAl
LSC001	2146	João	A1	4	1/11/91	24
LSC001	3145	Sílvio	A2	4	2/10/91	24
LSC001	6126	José	B1	9	3/10/92	18
LSC001	1214	Carlos	A2	4	4/10/92	18
LSC001	8191	Mário	A1	4	1/11/92	12
PAG02	8191	Mário	A1	4	1/05/93	12
PAG02	4112	João	A2	4	4/01/91	24
PAG02	6126	José	B1	9	1/11/92	12

## Segunda forma normal - 2FN

*segunda forma normal (2FN)*

=

uma tabela encontra-se na segunda  
forma normal, quando, além de estar  
na 1FN, não contém *dependências*  
*parciais*

# Dependência funcional parcial

*dependência parcial*

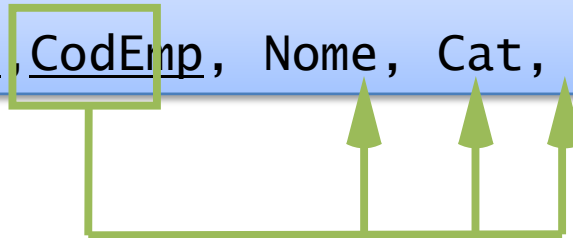
=

uma dependência (funcional) parcial  
ocorre quando uma coluna depende  
apenas *de parte de* uma chave  
primária composta

# Dependências parciais

1FN:

ProjEmp (CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)





# Dependências não parciais

1FN:

ProjEmp (CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)

The diagram illustrates a functional dependency. A horizontal line connects the primary key attributes, CodProj and CodEmp, to a vertical line that then branches into two arrows pointing to the non-key attributes, DataIni and TempAl. This indicates that the combination of CodProj and CodEmp uniquely determines the values of DataIni and TempAl.

## Passagem à 2FN

- Tabela 1FN e que possui apenas uma coluna como chave primária:
  - Não contém dependências parciais.
  - É impossível uma coluna depender de uma parte da chave primária, quando a chave primária não é composta por partes.
- Conclusão:
  - Toda tabela 1FN que possui apenas uma coluna como chave primária já está na 2FN.

# Passagem à 2FN

## Tabela com uma única coluna na chave

1FN:

(CodProj, Tipo, Descr)  
(CodProj, CodEmp, Nome, Cat,  
Sal, DataIni, TempAl)

2FN:

(CodProj, Tipo, Descr)

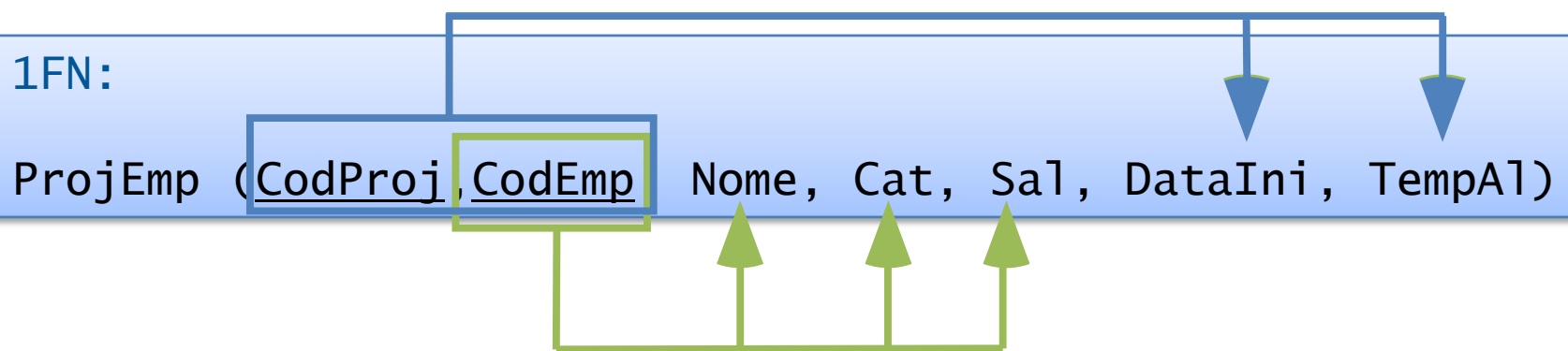
## Passagem à 2FN

- Idem para:
  - Tabela que contenha apenas colunas chave primária:
    - Impossível atributo não chave depender de parte da chave (tabela não tem colunas não chave).
  - Tabela sem colunas não chave já está na 2FN.

# Passagem à 2FN

1FN:

ProjEmp (CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)



# Passagem à 2FN

1FN:

ProjEmp (CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)

Tabela que possui *chave primária* com *várias* colunas e possui colunas *não chave* deve ser examinada

# Passagem à 2FN

1FN:

ProjEmp (CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)

Pergunta a ser feita, para cada coluna não chave:

- “a coluna depende de toda a chave ou só de parte”  
ou
- “para identificar um valor da coluna necessita de toda chave ou só de parte dela” ?

# Passagem à 2FN

1FN:

ProjEmp (CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)

Colunas que dependem de toda a chave permanecem na tabela original

2FN:

ProjEmp (CodProj, CodEmp, DataIni, TempAl)



# Passagem à 2FN

1FN:

ProjEmp (CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)



A green line connects the underlined attribute CodProj to three green arrows pointing upwards to the attributes Nome, Cat, and Sal, indicating functional dependencies.

2FN:

ProjEmp (CodProj, CodEmp, DataIni, TempAl)

Emp (CodEmp, Nome, Cat, Sal)

Colunas que dependem de  
parte da chave vão para uma  
nova tabela

## 2FN resultante

2FN:

Proj (CodProj, Tipo, Descr)

ProjEmp (CodProj, CodEmp, DataIni, TempAl)

Emp (CodEmp, Nome, Cat, Sal)

## Tabelas na 2FN - exemplo

Proj:

<u>CódProj</u>	Tipo	Descr
LSC001	Novo Desenv.	Sistema de Estoque
PAG02	Manutenção	Sistema de RH

# Tabelas na 2FN - exemplo

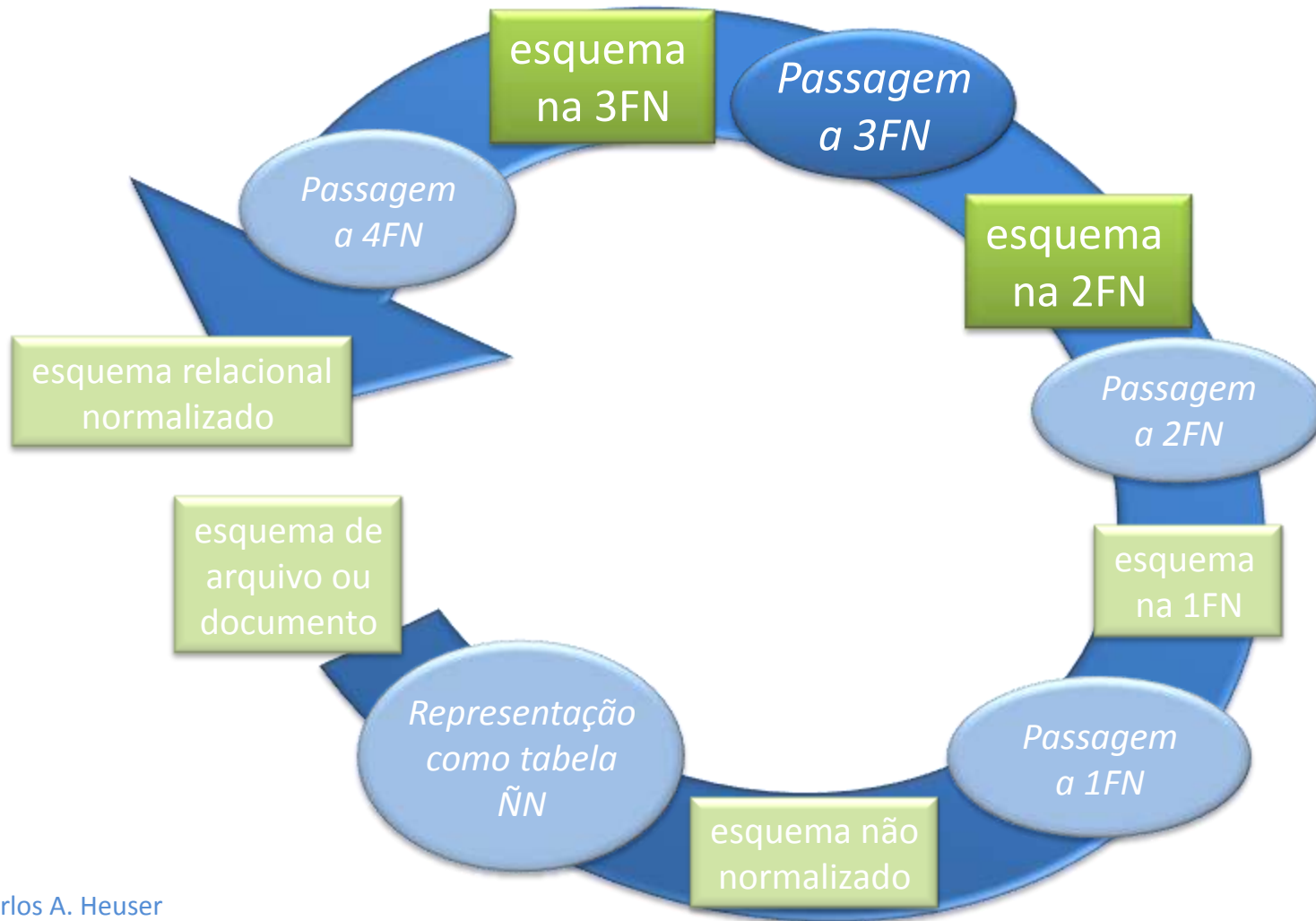
Emp:

<u>CodEmp</u>	Nome	Cat	Sal
2146	João	A1	4
3145	Sílvio	A2	4
1214	Carlos	A2	4
8191	Mário	A1	4
4112	João	A2	4
6126	José	B1	9

ProjEmp:

<u>CódProj</u>	<u>CodEmp</u>	DataIni	TempAl
LSC001	2146	1/11/91	24
LSC001	3145	2/10/91	24
LSC001	6126	3/10/92	18
LSC001	1214	4/10/92	18
LSC001	8191	1/11/92	12
PAG02	8191	1/05/93	12
PAG02	4112	4/01/91	24
PAG02	6126	1/11/92	12

# Passagem à 3FN



## Terceira forma normal (3FN)

- Trata de um outro tipo de redundância.

- Exemplo:

2FN:

Emp (CodEmp, Nome, Cat, Sal)

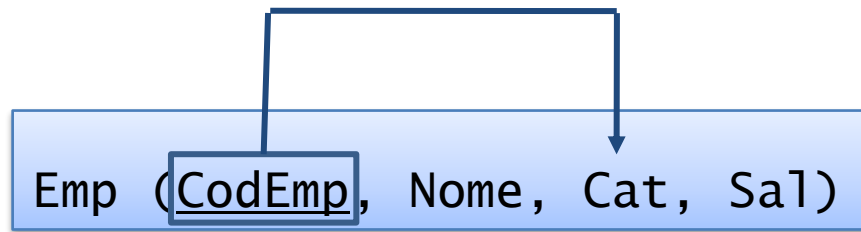
- Se
  - salário (coluna **Sal**) é determinado pela categoria funcional (coluna **Cat**)
- Salário que é pago a uma categoria funcional é armazenado tantas vezes quantos empregados possui a categoria funcional

## Terceira forma normal (3FN)

Emp:

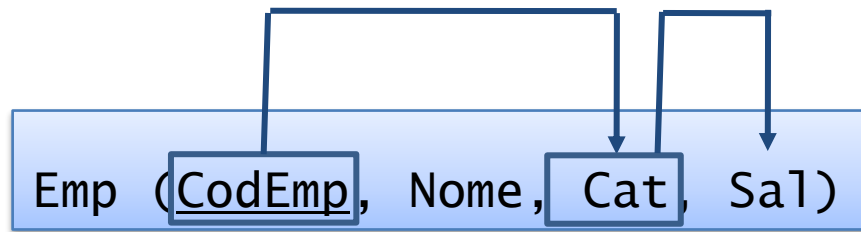
CodEmp	Nome	Cat	Sal
2146	João	A1	4
3145	Sílvio	A2	4
1214	Carlos	A2	4
8191	Mário	A1	4
4112	João	A2	4
6126	José	B1	9

# Dependências funcionais

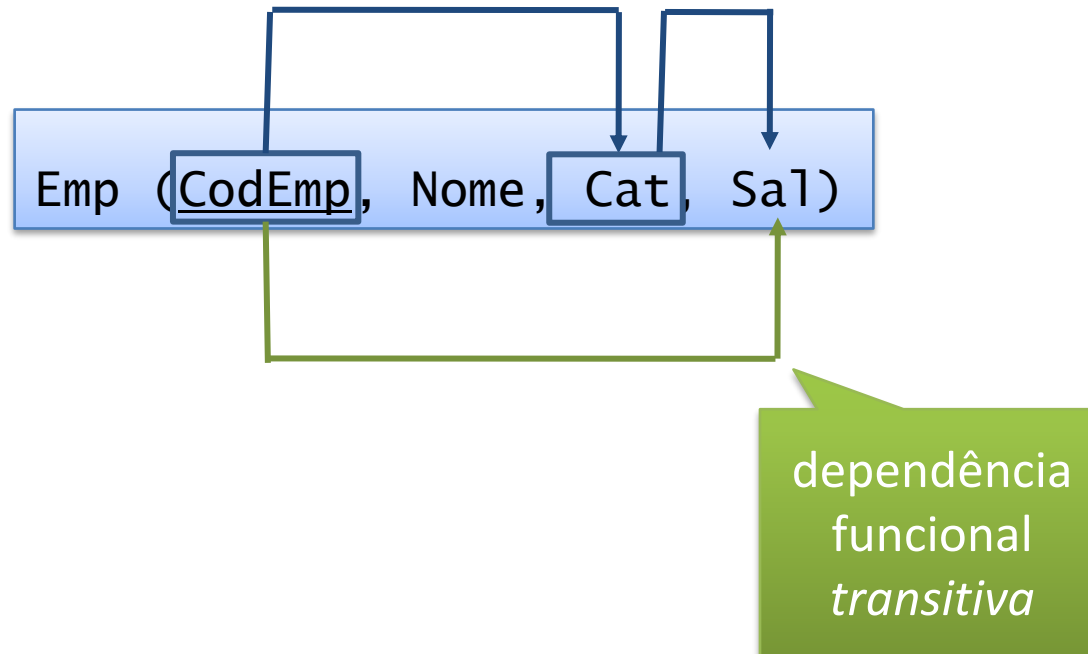




# Dependências funcionais



# Dependência transitiva



# Terceira forma normal

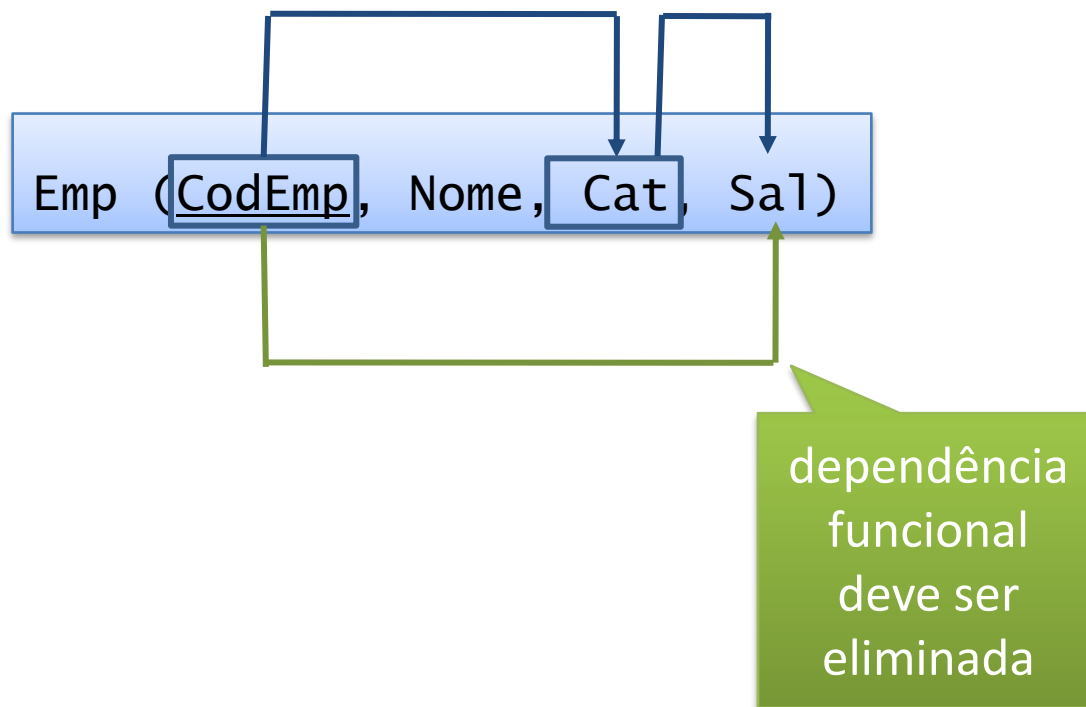
## 3FN

*terceira forma normal (3FN)*

=

uma tabela encontra-se na terceira forma normal, quando, além de estar na 2FN, não contém dependências transitivas

# Passagem à 3FN



# Passagem à 3FN

2FN:

Emp (CodEmp, Nome, Cat, Sal)

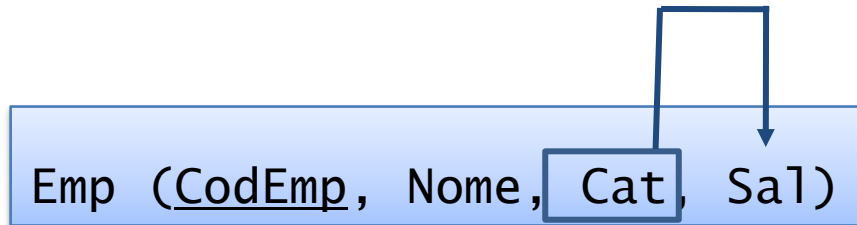


Colunas que dependem da  
chave permanecem na tabela  
original

3FN:

Emp (CodEmp, Nome, Cat)

# Passagem à 3FN



Colunas que dependem de  
coluna não chave vão para  
outra tabela

3FN:

Cat(Cat, Sal)

## 3FN do exemplo

3FN:

Proj (CodProj, Tipo, Descr)

ProjEmp (CodProj, CodEmp, DataIni, TempAl)

Emp (CodEmp, Nome, Cat)

Cat (Cat, Sal)

# Normalização do exemplo

ÑN:

```
Proj (CodProj, Tipo, Descr,  
      (CodEmp, Nome, Cat, Sal, DataIni, TempAl)  
      )
```

1FN:

```
(CodProj, Tipo, Descr)  
(CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)
```

2FN:

```
Proj (CodProj, Tipo, Descr)  
ProjEmp (CodProj, CodEmp, DataIni, TempAl)  
Emp (CodEmp, Nome, Cat, Sal)
```

3FN:

```
Proj (CodProj, Tipo, Descr)  
ProjEmp (CodProj, CodEmp, DataIni, TempAl)  
Emp (CodEmp, Nome, Cat)  
Cat (Cat, Sal)
```



## Tabelas na 3FN - exemplo

Proj:

<u>CódProj</u>	Tipo	Descr
LSC001	Novo Desenv.	Sistema de Estoque
PAG02	Manutenção	Sistema de RH

ProjEmp:

<u>CódProj</u>	<u>CodEmp</u>	DataIni	TempAl
LSC001	2146	1/11/91	24
LSC001	3145	2/10/91	24
LSC001	6126	3/10/92	18
LSC001	1214	4/10/92	18
LSC001	8191	1/11/92	12
PAG02	8191	1/05/93	12
PAG02	4112	4/01/91	24
PAG02	6126	1/11/92	12

## Tabelas na 3FN - exemplo

Emp:

<u>CodEmp</u>	Nome	Cat
2146	João	A1
3145	Sílvio	A2
1214	Carlos	A2
8191	Mário	A1
4112	João	A2
6126	José	B1

Cat:

<u>Cat</u>	Sal
A1	4
A2	4
B1	9

# Passagem à 3FN

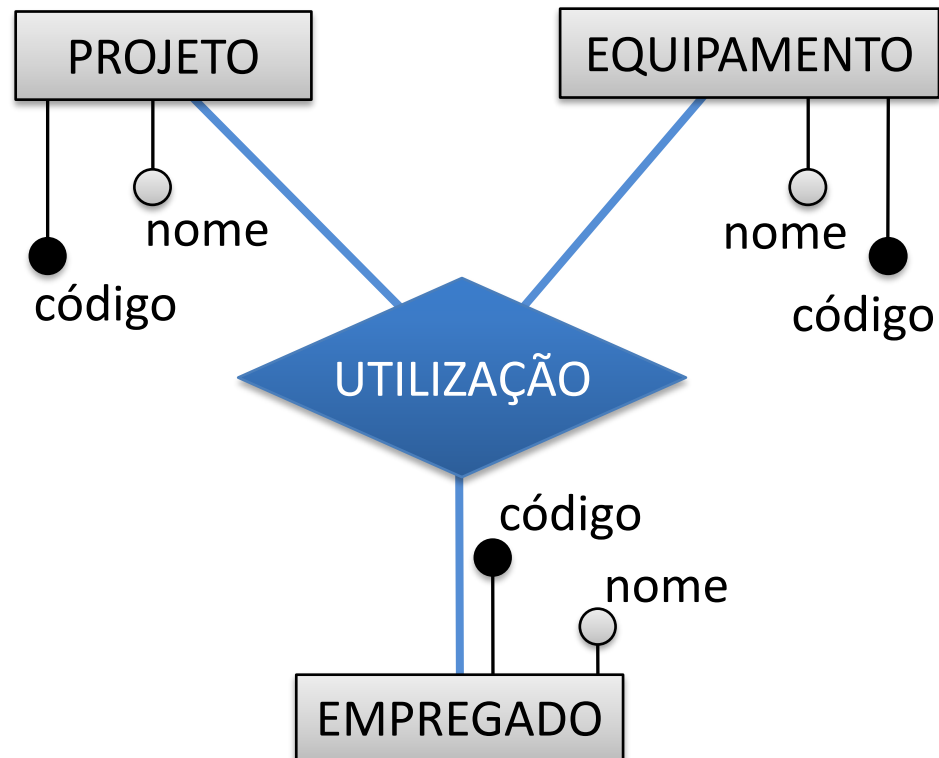


# Passagem à 4FN

- Para a maioria dos documentos e arquivos:
  - a decomposição até a 3FN é suficiente.
- Na literatura, aparecem outras formas normais:
  - forma normal de Boyce/Codd,
  - a 4FN,
  - a 5FN.

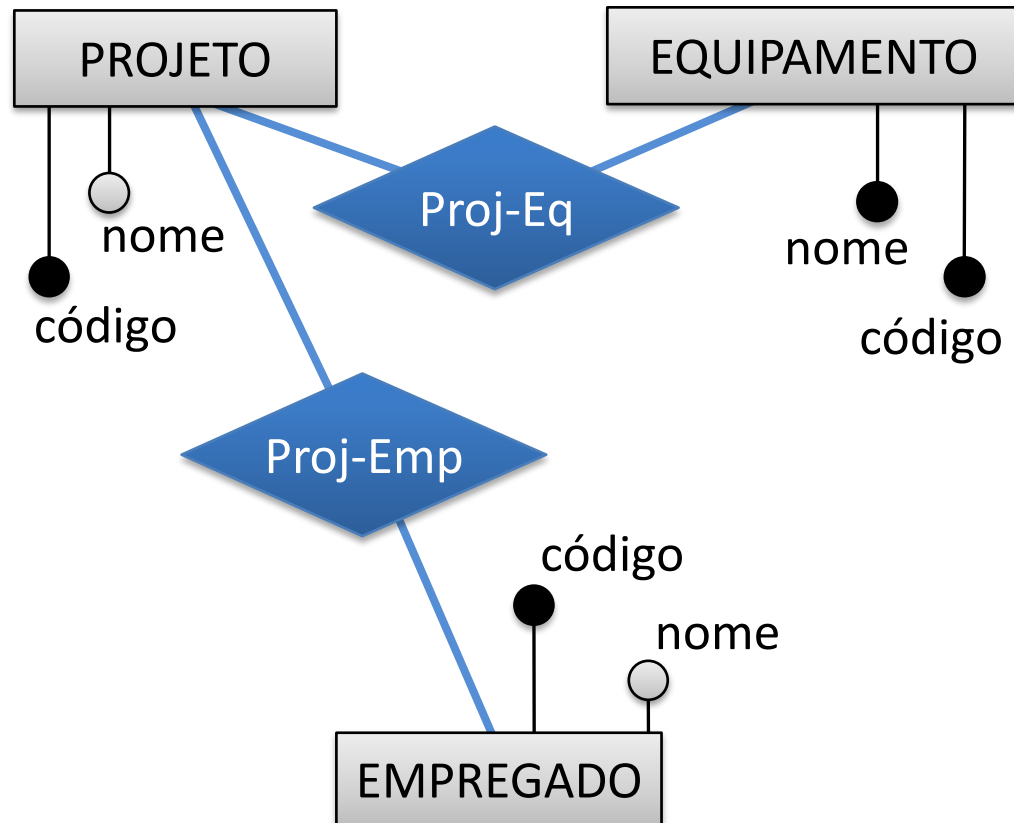
# Exemplo para 4FN

## Modelo original

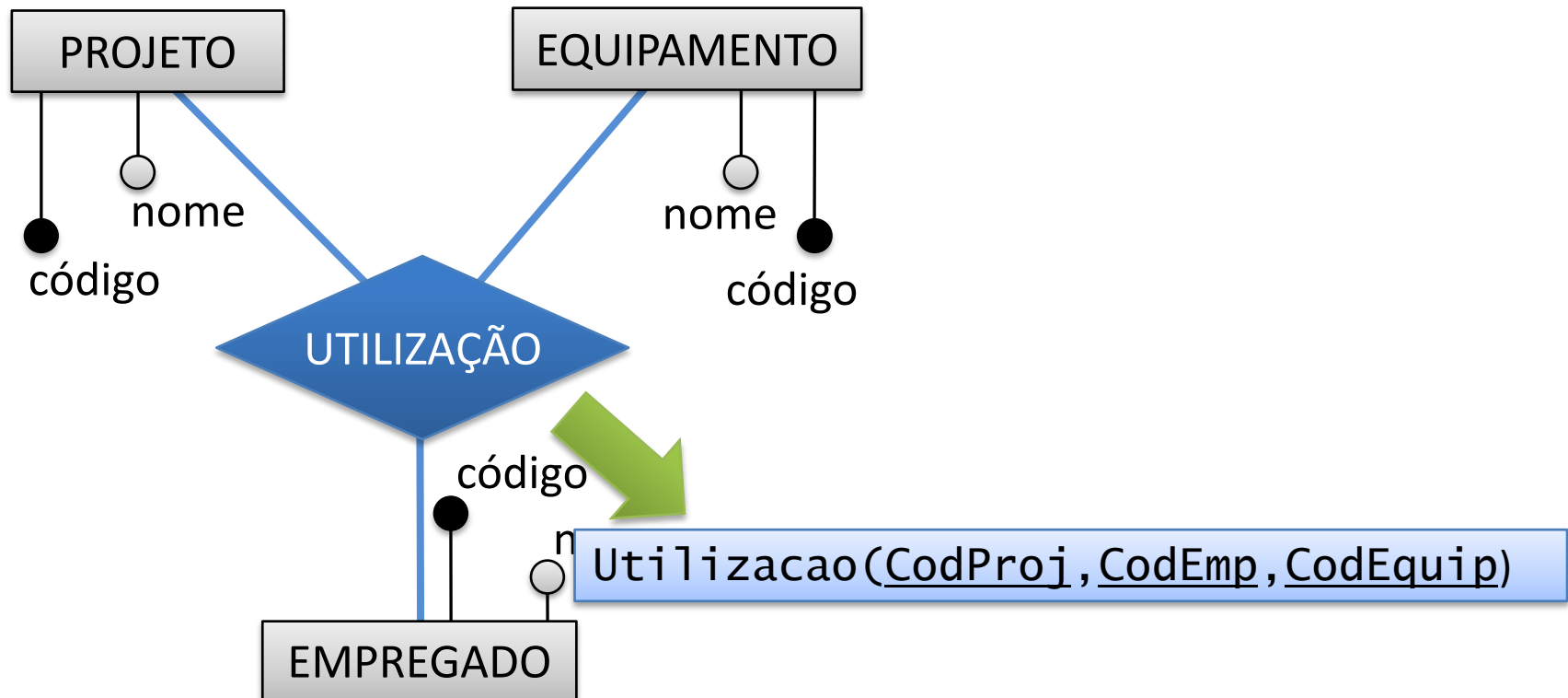


# Exemplo para 4FN

## Requisitos alterados



# Exemplo – Implementação do relacionamento



## Tabela Utilização com requisitos alterados

CodProj	CodEmp	CodEquip
1	1	1
1	2	1
1	3	1
1	1	2
1	2	2
1	3	2
2	2	2
2	2	4
3	3	1
3	4	1
3	3	3
3	4	3
3	3	5
3	4	5
4	2	5



# Tabela Utilização com requisitos alterados

CodProj	CodEmp	CodEquip
1	1	1
1	2	2
1	3	3
1	1	2
1	2	2
1	3	2
2	2	2
2	2	4
3	3	1
3	4	1
3	3	3
3	4	3
3	3	5
3	4	5
4	2	5

quais são os  
empregados que  
trabalham no  
projeto 1?

## Tabela Utilização com requisitos alterados

CodProj	CodEmp	CodEquip
1	1	1
1	2	1
1	3	1
1	1	2
1	2	
1	3	
2	2	
2	2	
3	3	1
3	4	1
3	3	3
3	4	3
3	3	5
3	4	5
4	2	5

quais são os  
empregados que  
trabalham no  
projeto 1?

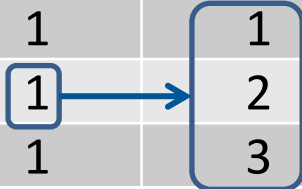
## Tabela Utilização com requisitos alterados

CodProj	CodEmp	CodEquip
1	1	1
1	2	1
1	3	1
1	1	2
1	2	2
1	3	2
2	2	2
2	2	4
3	3	1
3	4	1
3	3	3
3	4	3
3	3	5
3	4	5
4	2	5

quais são os  
equipamentos  
usados no projeto 1?

# Dependência funcional multivalorada

CodProj	CodEmp	CodEquip
1	1	1
1	2	1
1	3	1
1	1	2
1	2	2
1	3	2
2	2	2
2	3	2



# Dependência multivalorada

CodProj	CodEmp	CodEquip
1	1	1
1	2	1
1	3	1
1	1	2
1	2	2
1	3	2
2	2	2
2	2	2

CodProj  $\twoheadrightarrow$  CodEmp

# Dependência multivalorada

CodProj	CodEmp	CodEquip
1	1	1
1	2	1
1	3	1
1	1	2
1	2	2
1	3	2
2	2	2
2	3	2

CodProj  $\twoheadrightarrow$  CodEmp

CodProj  $\twoheadrightarrow$  CodEquip

# 4FN definição

*quarta forma normal (4FN)*  
=  
uma tabela encontra-se na  
quarta forma normal,  
quando,  
além de estar na 3FN,  
não contém mais de uma  
dependência multi-valorada

# 4FN

3FN:

Utilizacao(CodProj, CodEmp, CodEquip)



4FN:

ProjEmp (CodProj, CodEmp)

ProjEquip (CodProj, CodEquip)



# Problemas da normalização

1. Chaves primárias omitidas ou incorretas
2. Atributos relevantes implicitamente representados
3. Atributos irrelevantes, redundantes ou derivados

# Chaves primárias omitidas ou incorretas

- Arquivos convencionais:
  - o conceito de chave primária não é obrigatório;
  - é possível encontrar arquivos que não possuem chave primária.
- Quando um arquivo convencional não possui chave primária ou quando a chave primária nele usada difere da usual na organização:
  - deve-se proceder como se a chave primária aparecesse no arquivo;
  - deve-se inseri-la na forma ÑN.

## Chaves primárias omitidas ou incorretas exemplo

- Arquivo com dados sobre empregados de uma organização enviado para fins de fiscalização a um órgão governamental.
- Identificador de empregado usado na organização é omitido, já que este é irrelevante para o órgão fiscalizador.

## Chaves primárias omitidas ou incorretas - exemplo

- Outra situação:
  - uso de uma chave alternativa, ao invés da chave primária usual do arquivo.
- No caso mencionado acima:
  - Se o órgão governamental fosse a receita federal:
    - Arquivo poderia ter como chave primária o CIC do empregado, ao invés da chave primária normalmente usada na organização.

## Atributos relevantes implicitamente representados

- Arquivos convencionais podem conter atributos de forma implícita:
  - ordenação de registros ou de listas;
  - ponteiros físicos, etc.
- Deve-se proceder como se o atributo aparecesse explicitamente no documento.

# Atributo implícito

## Ordenação

- Exemplo:
  - arquivo contém registros referentes a **cursos** em um concurso vestibular;
  - para cada curso, há um grupo repetido aninhado, com as informações dos candidatos ao curso em questão;
  - informações dos candidatos **ordenadas** por classificação no concurso.

## Atributo implícito - Ordenação

ÑN:

Arq-Candidatos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso,  
(Cod-Cand, Nome-Cand))



4FN:

Cursos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso)  
Candidatos (Cod-Curso, Cod-Cand, Nome-Cand)

## Atributo implícito Ordenação

- Informação da classificação dos candidatos em um curso foi perdida no processo de normalização.
- Procedimento correto:
  - incluir explicitamente na tabela, já na forma ÑN, a informação que aparece implicitamente no arquivo na forma da ordenação dos registros (coluna **Ordem-Cand**).

ÑN:

Arq-Candidatos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso,  
(Cod-Cand, Nome-Cand, **Ordem-Cand**)  
)



## Atributo implícito - Ordenação

ÑN:

Arq-Candidatos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso,  
(Cod-Cand, Nome-Cand, **Ordem-Cand**)  
)



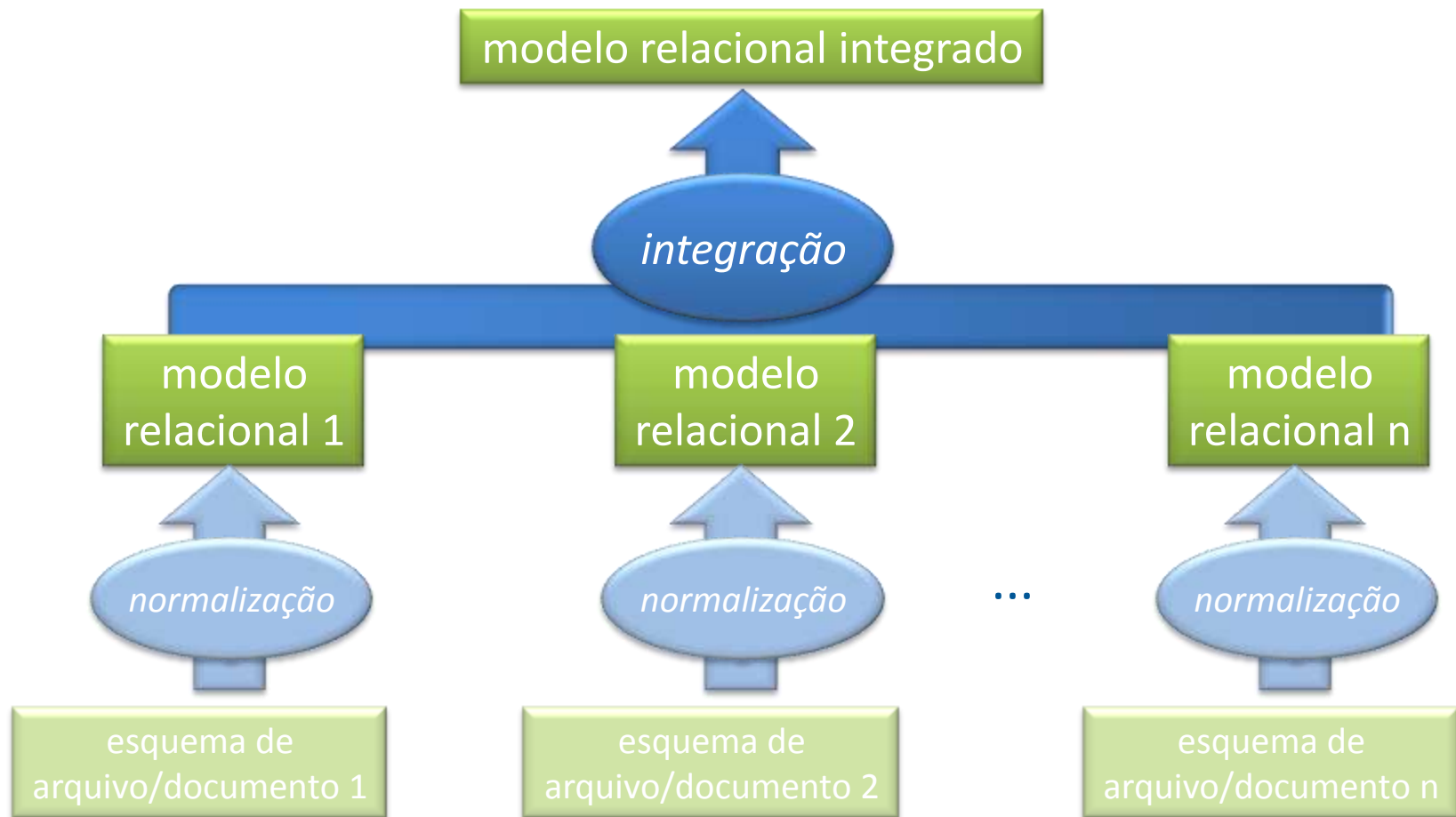
4FN:

Cursos (Cod-Curso, Nome-Curso, Numero-Vagas-Curso)  
Candidatos (Cod-Curso, Cod-Cand, Nome-Cand, **Ordem-Cand**)

# Atributos irrelevantes, redundantes ou derivados

- Atributos irrelevantes, redundantes ou derivados:
  - Devem ser eliminados já quando da passagem a forma não normalizada.

# Integração de modelos



# Integração de modelos

- Normalização de cada um dos arquivos/documentos conduz à definição de um conjunto de tabelas.
- Passo seguinte :
  - integrar os modelos obtidos para cada arquivo no modelo global do banco de dados.
- Processo é conhecido por:
  - *integração de visões*;
  - *integração de esquemas* .

# Integração de modelos objetivos

- Os atributos de uma mesma entidade (ou de um mesmo relacionamento) podem estar armazenados em diferentes arquivos:
  - juntar as tabelas em uma única tabela que representa a entidade ou relacionamento em questão.
- Tabelas dentro de um modelo livres de redundâncias.
- Tabelas entre diferentes modelos podem ter redundâncias entre si
  - integração elimina estas redundâncias.

# Integração de modelos passos

1. integração de tabelas com a mesma chave;
2. integração de tabelas com chave contida;
3. verificação de 3FN

## Integração de tabelas com mesma chave

- Junção de tabelas que possuem a *mesma* chave primária.
- “*mesma*” chave primária =
  - *domínios* e *conteúdos* das colunas que compõem a chave primária são iguais.

# Integração de tabelas com mesma chave - exemplo

## Documento 1:

Proj (CodProj, Tipo, Descr)  
ProjEmp (CodProj, CodEmp, DataIni, TempAl)  
Emp (CodEmp, Nome, Cat)  
Cat (Cat, Sal)

## Documento2:

Proj (CodProj, DataInicio, Descr, CodDepto)  
Depto (CodDepto, NomeDepto)  
ProjEquipamento (CodProj, CodEquipam, DataIni)  
ProjEmp (CodProj, CodEmp, FunçãoEmpProj)  
Equipamento (CodEquipam, Descrição)



# Integração de tabelas com mesma chave - exemplo

## Documento 1:

Proj (CodProj, Tipo, Descr)

ProjEmp (CodProj, CodEmp, DataIni, TempAl)

Emp (CodEmp, Nome, Cat)

Cat (Cat, Sal)

## Documento2:

Proj (CodProj, DataInicio, Descr, CodDepto)

Depto (CodDepto, NomeDepto)

ProjEquipamento (CodProj, CodEquipam, DataIni)

ProjEmp (CodProj, CodEmp, FunçãoEmpProj)

Equipamento (CodEquipam, Descrição)

# Integração de tabelas com mesma chave - exemplo

## Documento 1:

Proj (CodProj, Tipo, Descr)

ProjEmp (CodProj, CodEmp, DataIni, TempAl)

Emp (CodEmp, Nome, Cat)

Cat (Cat, Sal)

## Documento2:

Proj (CodProj, DataInicio, Descr, CodDepto)

Depto (CodDepto, NomeDepto)

ProjEquipamento (CodProj, CodEquipam, DataIni)

ProjEmp (CodProj, CodEmp, FunçãoEmpProj)

Equipamento (CodEquipam, Descrição)

## Integração de tabelas com mesma chave - exemplo

### Modelo integrado:

Proj (CodProj, Tipo, Descr, DataInicio, CodDepto)

ProjEmp (CodProj, CodEmp, DataIni, TempAl, FunçãoEmpProj)

Emp (CodEmp, Nome, Cat)

Cat (Cat, Sal)

Depto (CodDepto, NomeDepto)

ProjEquipamento (CodProj, CodEquipam, DataIni)

Equipamento (CodEquipam, Descrição)

# Integração de modelos problemas

- Processo baseia-se na comparação dos *nomes* de colunas e de tabelas dentro dos diferentes modelos.
- Problema :
  - *conflitos de nomes*:
    - Homônimos
    - Sinônimos

# Integração de tabelas com chaves contidas

- Tabelas são **fundidas**:
  - uma tabela contém **somente a chave primária** e
  - a chave primária é **subconjunto da chave primária de outra tabela**.
- Chave primária está contida dentro da outra:
  - chave primária deve ter o mesmo domínio e os mesmos valores.

# Integração de tabelas com chaves contidas

## Exemplo

- Exemplo:

Modelo #1:

AlunoDisc (Cod-Al, Cod-Disc)

Modelo #2:

AlunoDiscSem (Cod-Al, Cod-Disc, Sem-Disc-Cursada,  
Nota-Disc)

# Integração de tabelas com chaves contidas

- Exemplo:

Modelo #1:

AlunoDisc (Cod-Al, Cod-Disc)

Modelo #2:

AlunoDiscSem (Cod-Al, Cod-Disc, Sem-Disc-Cursada,  
Nota-Disc)

- Primeira tabela:
  - informa que um aluno cursou uma disciplina.

# Integração de tabelas com chaves contidas

- Exemplo:

Modelo #1:

AlunoDisc (Cod-Al, Cod-Disc)

Modelo #2:

AlunoDiscSem (Cod-Al, Cod-Disc, Sem-Disc-Cursada,  
Nota-Disc)

- Primeira tabela:
  - informa que um aluno cursou uma disciplina.
- Segunda tabela:
  - informa a nota obtida pelo aluno em uma disciplina em um semestre.



# Integração de tabelas com chaves contidas

Modelo #1:

AlunoDisc (Cod-Al, Cod-Disc)

Modelo #2:

AlunoDiscSem (Cod-Al, Cod-Disc, Sem-Disc-Cursada,  
Nota-Disc)

- Caso as colunas **Cod-Al** e **Cod-Disc** da tabela **AlunoDisc**
  - contenha os mesmo dados que as colunas **Cod-Al** e **Cod-Disc** da tabela **AlunoDiscSem**:
    - Informações contidas na tabela **AlunoDisc** já estão na tabela **AlunoDiscSem**;
    - Tabela **AlunoDisc** é **redundante** e pode ser eliminada sem perda de informações.

# Integração de tabelas com chaves contidas

- Não integrar quando tabela contém dados além da chave primária.

Modelo #1:

AlunoDisc (Cod-Al, Cod-Disc, BolsaSimNao)

Modelo #2:

AlunoDiscSem (Cod-Al, Cod-Disc, Sem-Disc-Cursada,  
Nota-Disc)

# Integração de tabelas com chaves contidas

- Garantir que primeira tabela efetivamente contida na segunda.
- Exemplo:

Modelo #1:

AlunoDisc (Cod-Al, SemDisc)

Modelo #2:

AlunoDiscSem (Cod-Al, Cod-Disc, SemDisc, Nota-Disc)

# Integração de tabelas com chaves contidas

- Garantir que primeira tabela efetivamente contida na segunda.
- Exemplo:

Modelo #1:

AlunoDisc (Cod-Al, SemDisc)

Modelo #2:

AlunoDiscSem (Cod-Al, Cod-Disc, SemDisc)

representa o fato de um  
aluno estar matriculado  
em um semestre

# Integração de tabelas com chaves contidas

- Garantir que primeira tabela efetivamente contida na segunda.
- Exemplo:

Modelo #1:

AlunoDisc (Cod-Al, SemDisc)

Modelo #2:

AlunoDiscSem (Cod-Al, Cod-Disc, SemDisc, Nota-Disc)

representa a nota que o  
aluno obteve em uma  
disciplina em um  
semestre

## Volta à 2FN

- A integração de dois modelos 4FN pode conduzir a um modelo que está na 2FN mas não na 3FN.
- Exemplo:

Modelo #1:

Departamento (CodDepto, NomeDepto, CodGerenteDepto)

Modelo # 2:

Departamento (CodDepto, LocalDepto, NomeGerenteDepto)

## Volta à 2FN

- Integração destes dois modelos resultaria no modelo integrado abaixo mostrado.
- Modelo integrado:

Modelo #1:

Departamento (CodDepto, NomeDepto, CodGerenteDepto,  
LocalDepto, NomeGerenteDepto)

## Volta à 2FN

- Integração destes dois modelos resultaria no modelo integrado abaixo mostrado.
- Modelo integrado:

Modelo #1:

Departamento (CodDepto, NomeDepto, CodGerenteDepto,  
LocalDepto, NomeGerenteDepto)

- Não está na 3FN



# Verificação do modelo ER

## Limitações da Normalização

- Obtido o modelo relacional normalizado pode ser construído o modelo ER correspondente (regras apresentadas no capítulo 5).
- O processo de normalização não conduz necessariamente a um modelo ER perfeito.
- Normalização apenas elimina:
  - campos multivalorados ;
  - redundâncias de dados detectadas pelas formas normais descritas.

# Verificação do modelo ER

## Limitações da Normalização

- Optamos pela alternativa de decompor tabelas na passagem à 1FN:
  - alternativa, apesar de mais simples de tratar na prática, pode levar a imperfeições no modelo.
- Há outras formas normais (Boyce/Codd e a quinta forma normal) .

# Construção do modelo ER

- Último passo da engenharia reversa:
  - construção do modelo ER através das regras para engenharia reversa de modelos relacionais;
  - verificação do modelo ER obtido, procurando corrigir imperfeições ainda existentes.