Pós-Graduação
Engenharia de Software

Modelagem de Dados

Aula 05

Normalização

Normalização

- **Objetivo**
  - Apresentar uma abordagem de projeto de banco de dados, denominada de Normalização, a qual permite analisar a qualidade das relações, bem como elevar a sua qualidade.
- **Principais tópicos**
  - Anomalias
  - Tuplas espúrias
  - Abordagens de Projeto de Banco de Dados
  - Dependências Funcionais
  - Regras de Inferência para DFs

Normalização

- **Principais tópicos *(Continuação)***
  - Formas Normais com base em Chaves Primárias
  - Definição Geral de Formas Normais
  - BCNF (Boyce-Codd Normal Form)
  - Dependências Multivaloradas
  - Quarta Forma Normal (4FN)

Abordagens de Projeto de BD

- **Top-down**
  - Iniciar com o agrupamento dos atributos obtidos a partir do projeto conceitual de mapeamento
  - Isso é chamado de projeto por análise
- **Bottom-up**
  - Considerar os relacionamentos entre atributos
  - Construir as relações
  - Isso é chamado projeto pela síntese
- **Nossa Abordagem**
  - Utilizar a abordagem Top-down para obter as relações
  - Utilizar a abordagem Bottom-up para melhorar a qualidade das relações obtidas anteriormente

Anomalias

- **Cuidado com redundância de informação**

EMP_DEP

NSS	NOME	DTANIV	DNUMERO	DNOME	GERENTE
21	AA	-	5	CV	91
22	BB	-	5	CV	91
23	CC	-	6	TS	93
24	DD	-	7	OS	94
25	EE	-	7	OS	94

  - Anomalias de Inserção:
    - Como inserir novo departamento sem que exista empregados?
    - Inserir empregados é difícil quando informações de departamento devem ser inseridas corretamente.
  - Anomalias de Remoção:
    - O que acontece quando removemos CC? Perdemos o departamento 6!
  - Anomalias de Alteração:
    - Se mudarmos o gerente do departamento 5, devemos mudá-lo em todas as tuplas com DNUMERO = 5.

Tuplas Espúrias

- **Não quebre uma relação em relações que possam gerar tuplas espúrias**

DNUMERO	NOME	PNOME	PLOCALIZAÇÃO
123	XX	Compras	São Paulo
123	XX	Vendas	Rio de Janeiro
124	YY	Logística	São Paulo

- **A relação pode ser quebrada em**

DNUMERO	NOME	PNOME
123	XX	Compras
123	XX	Vendas
124	YY	Logística

DNUMERO	PLOCALIZAÇÃO
123	São Paulo
123	Rio de Janeiro
124	São Paulo

- **Quando fazemos o Join, obtemos NOVAS TUPLAS!**

DNUMERO	NOME	PNOME	PLOCALIZAÇÃO
123	XX	Compras	São Paulo
123	XX	Compras	Rio de Janeiro
123	XX	Vendas	São Paulo
123	XX	Vendas	Rio de Janeiro
124	YY	Logística	São Paulo

Após o Join, o resultado não foi a relação original. Assim, houve perda de informações. Conclui-se que houve uma decomposição com perdas.

Dependências Funcionais

- **Dependências funcionais (DFs) são usadas para medir formalmente a qualidade do projeto relacional**
- **As DFs e chaves são usadas para definir formas normais de relações**
- **As DFs são restrições que são derivadas do significado dos atributos e do seus inter-relacionamentos**
- **Um conjunto de atributos X determina funcionalmente um conjunto de atributos Y se o valor de X determinar um único valor Y**
  - X → Y

Dependências Funcionais

- **X→Y**
  - Se duas tuplas tiverem o mesmo valor para X, elas devem ter o mesmo valor para Y. Ou seja:
  - Se X→Y então, para quaisquer tuplas t_1 e t_2 de r(R):
    Se $t_1[X] = t_2[X]$, então $t_1[Y] = t_2[Y]$
- **Se K é uma chave de R, então K determina funcionalmente todos os atributos de R**
  - Isso porque, nunca teremos duas tuplas distintas com $t_1[K]=t_2[K]$
- **Importante**
  - X→Y especifica uma restrição sobre todas as instâncias de R
  - As DFs são derivadas das restrições do mundo real e não de uma extensão específica da relação R

Exemplos de Restrições de DF

- **O número do seguro social determina o nome do empregado**
  - NSS → ENOME
- **O número do projeto determina o nome do projeto e a sua localização**
  - PNUMERO → { PNOME, PLOCALIZACAO }
- **O nss de empregado e o número do projeto determinam as horas semanais que o empregado trabalha no projeto**
  - { NSS, PNUMERO } → HORAS

Regras de Inferência para DFs

- **Regras de inferência de Armstrong:**
  - RI1. (Reflexiva) Se Y ⊆ X (é subconjunto de), então X→Y
    (Isso também é válido quando X=Y)
  - RI2. (Aumentativa) Se X→Y, então XZ→YZ
    (Notação: XZ significa X U Z)
  - RI3. (Transitiva) Se X→Y e Y→Z, então X→Z
- **RI1, RI2 e RI3 formam um conjunto completo de regras de inferência**

Regras de Inferência para DFs

- **Algumas regras de inferência úteis:**
  - (Decomposição) Se X→YZ, então X→Y e X→Z
  - (Aditiva) Se X→Y e X→Z, então X→YZ
  - (Pseudotransitiva) Se X→Y e WY→Z, então WX→Z

- **As três regras de inferência acima, bem como quaisquer outras regras de inferência, podem ser deduzidas a partir de RI1, RI2 e RI3 (propriedade de ser completa)**

Formas Normais com base em Chaves Primárias

- **Normalização de Relações**
- **Uso prático de Formas Normais**
- **Definições de Chaves e de Atributos que participam de Chaves**
- **Primeira Forma Normal**
- **Segunda Forma Normal**
- **Terceira Forma Normal**

Normalização de Relações

- **Normalização**
  - Processo de decompor relações “ruins” dividindo seus atributos em relações menores e “melhores”
- **Forma Normal**
  - Indica o nível de qualidade de uma relação
- **1FN**
  - Definição de relação. Atributos atômicos (indivisíveis).
- **2FN, 3FN, BCNF**
  - Baseiam-se em chaves e DFs de uma relação esquema
- **4FN e 5FN**
  - Baseiam-se em chaves e dependências multivaloradas

Uso Prático das Formas Normais

- **Na prática, a normalização é realizada para obter projetos de alta qualidade**
- **Os projetistas de bancos de dados não precisam normalizar na maior forma normal possível.**
- **Desnormalização**
  - Processo de armazenar junções de relações de forma normal superior como uma relação base que está numa forma normal inferior

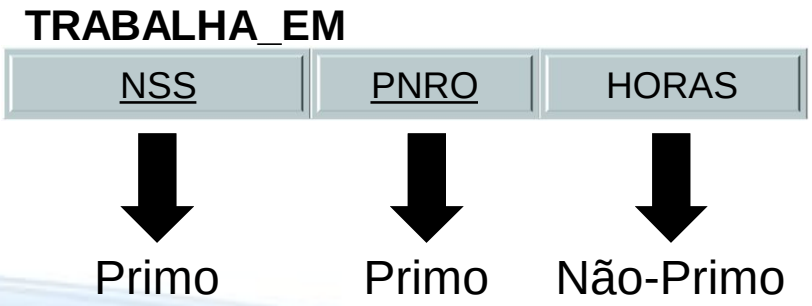
Definição de Chaves e Atributos que Participam de Chaves

- **Revisão**
  - Uma superchave, S, de uma relação esquema
    - R = {A1, A2, ..., An}
  - é um conjunto de atributos, subconjunto de R, com a propriedade de que $t_1[S] \neq t_2[S]$ para qualquer extensão r(R)
  - Uma superchave, K, é uma chave se K é uma superchave mínima
  - Se uma relação esquema tiver mais de uma chave, cada chave será chamada de chave-candidata. Uma das chaves-candidatas é arbitrariamente escolhida para ser a chave-primária e as outras são chamadas de chaves-secundárias

Definição de Chaves e Atributos que Participam de Chaves

- **Um atributo primo (ou primário) é membro de alguma chave-candidata**

- **Um atributo não-primo é um atributo que não é primo – isto é, não é membro de qualquer chave-candidata**



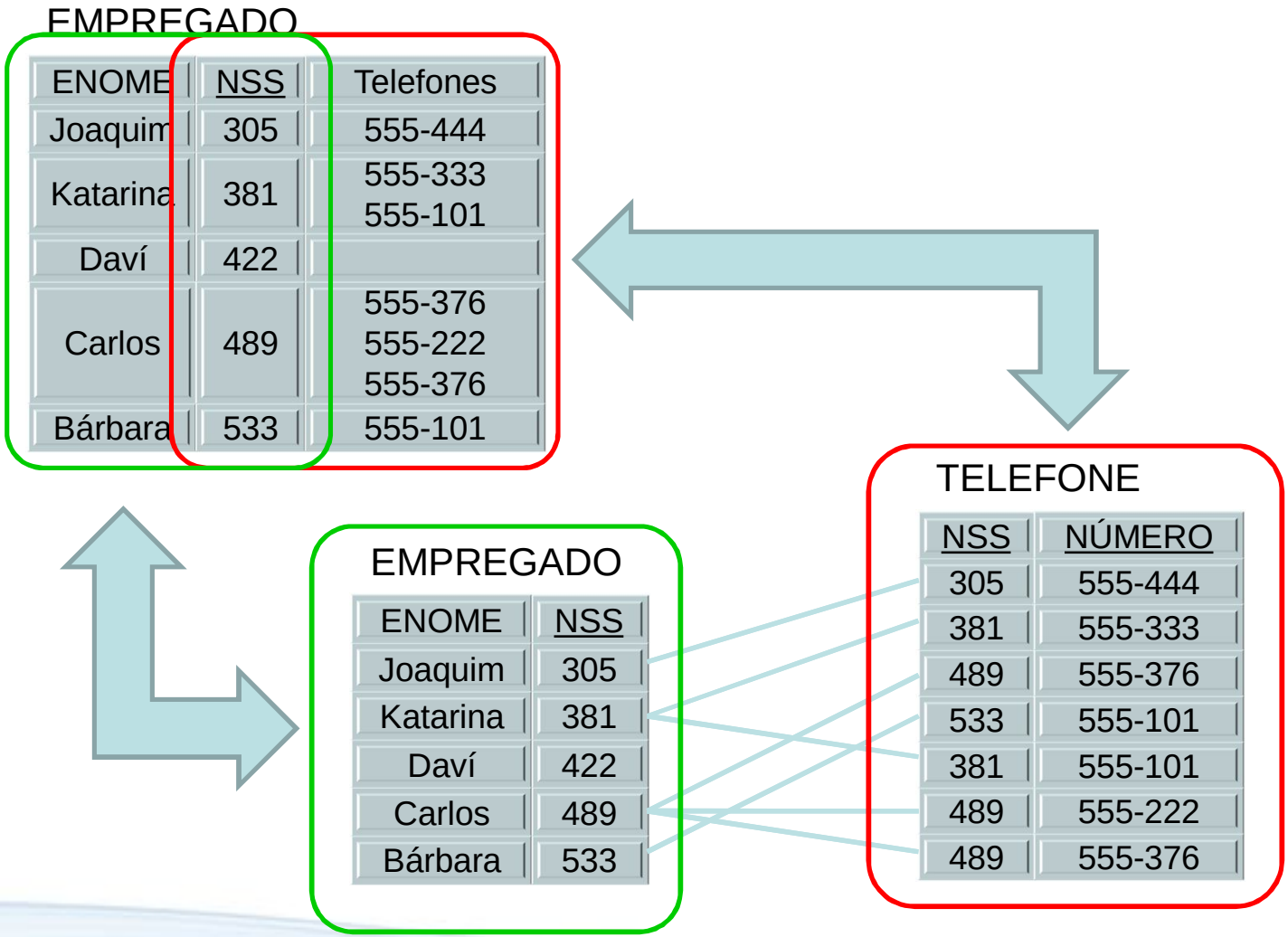
Primeira Forma Normal

- **Proíbe que relações tenham**
  - Atributos compostos
  - Atributos multivalorados
  - Relações aninhadas

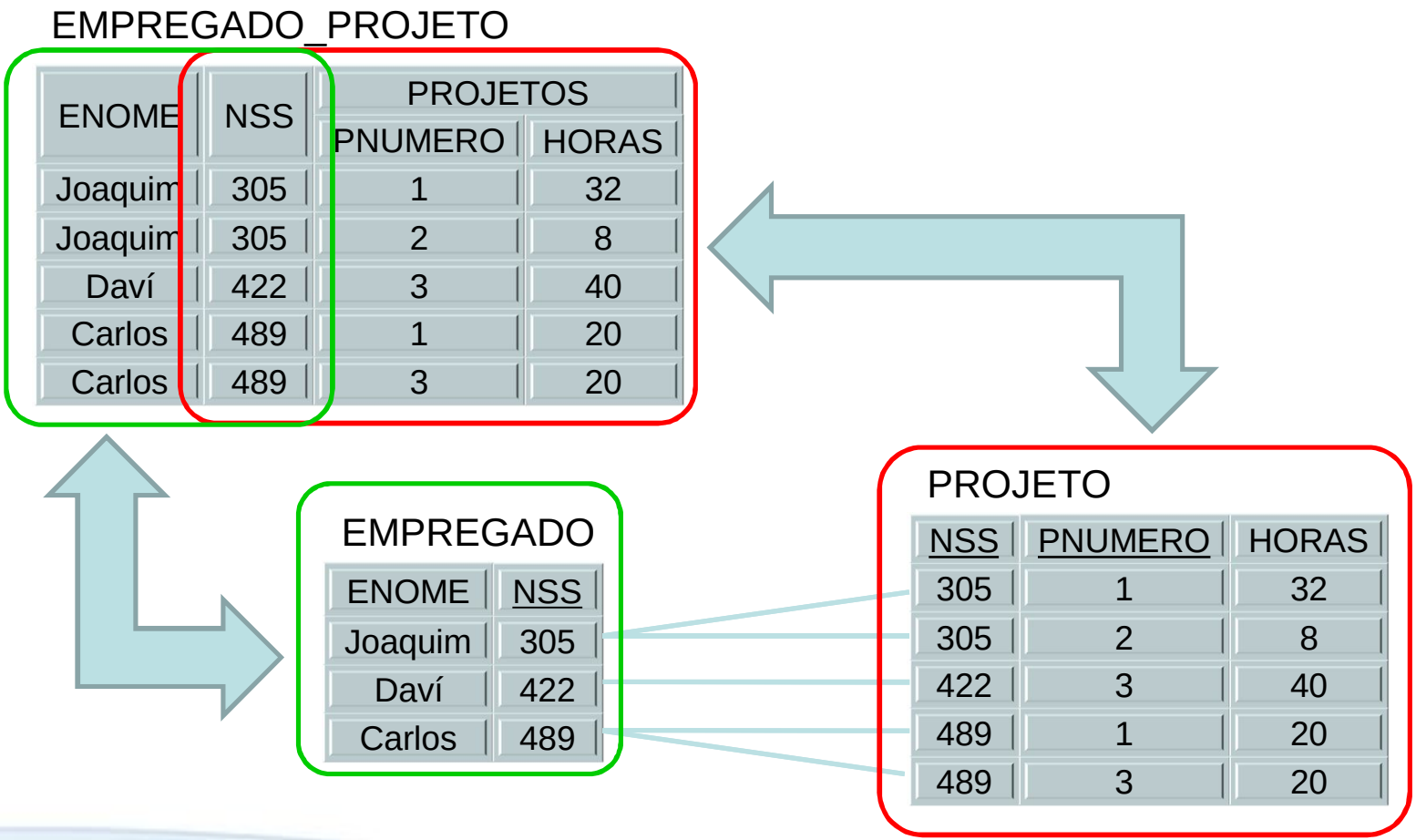
  Ou seja
  - Permite apenas atributos que sejam atômicos

- **Considerado como parte da definição de relação**

Normalização na 1 FN



Normalização de relações Aninhadas para a 1 FN



Segunda Forma Normal

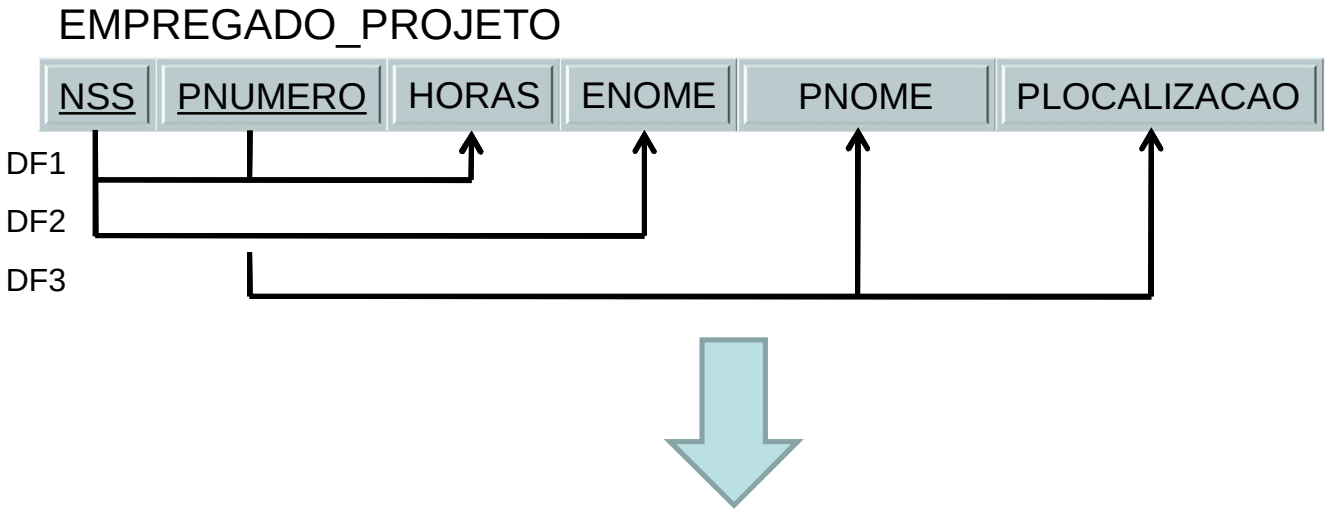
- **Para entender a 2FN precisamos entender:**
  - Dependência Funcional
  - Chave-primária
  - Atributo Não-Primo
  - **Dependência funcional total**
    - Uma DF, Y→Z, onde a remoção de qualquer atributo de Y invalida a DF. Exemplos:
      - { NSS, PNUMERO } → HORAS é dependente totalmente de { NSS, PNUMERO }, uma vez que NSS não determina HORAS e nem PNUMERO determina HORAS
      - { NSS, PNUMERO } → ENOME não é dependente totalmente de { NSS, PNUMERO }; ENOME é dependente parcialmente de { NSS, PNUMERO }, pois NSS → ENOME

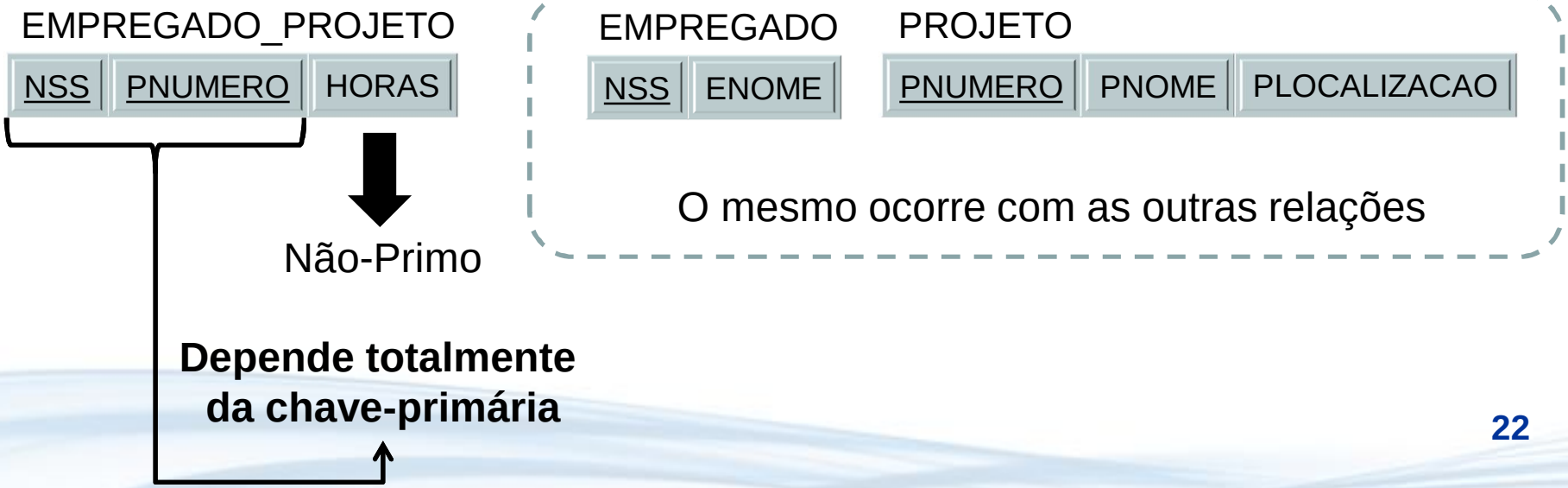
Segunda Forma Normal

- **Uma relação esquema R está na 2FN se estiver na 1FN e todos os atributos não-primos A de R forem totalmente dependentes da chave-primária**

- **R pode ser decomposto em relações que estejam na 2 FN através do processo de normalização**

Normalização para a 2FN e 3FN





Terceira Forma Normal

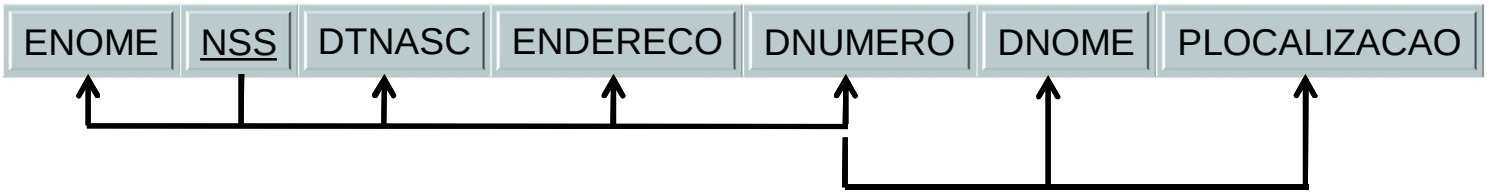
- **Para entender a 3FN precisamos entender:**
  - 2FN
  - Atributo Não-Primo
  - Dependência funcional transitiva
    - Se X→Y e Y→Z então X→Z

Terceira Forma Normal

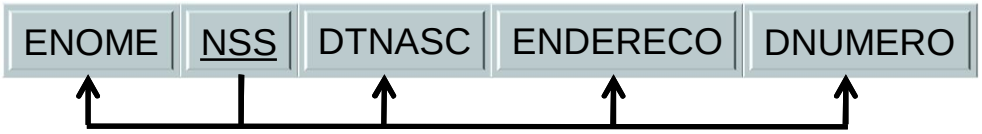
- **Uma relação esquema R está na 3FN se ela estiver na 2FN e nenhum atributo não-primo, A, for transitivamente dependente da chave-primária**
- **R pode ser decomposto em relações que estejam na 3FN via o processo de normalização**
- **NOTA:**
  - Em X→Y e Y→Z, sendo X a chave-primária, pode ser considerado um problema se, e somente se, Y não for uma chave-candidata. Quando Y é uma chave-candidata, não existe problema com a dependência transitiva
  - Por exemplo, considere EMP (NSS, Emp#, Salario).
    - Aqui, NSS → Emp# → Salario e Emp# é uma chave-candidata

Normalização para a 2FN e 3FN

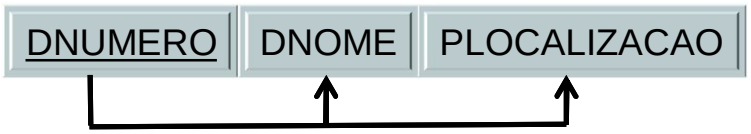
EMPREGADO_PROJETO



EMPREGADO



PROJETO

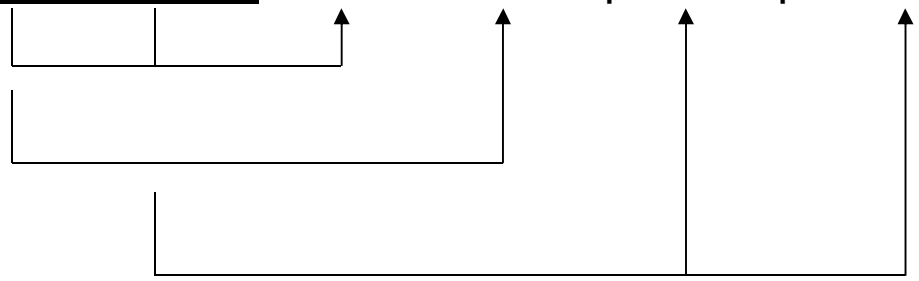


Definição Geral de Formas Normais

- **As definições anteriores consideravam somente a chave-primária**
- **As próximas definições levarão em consideração as várias chaves candidatas**

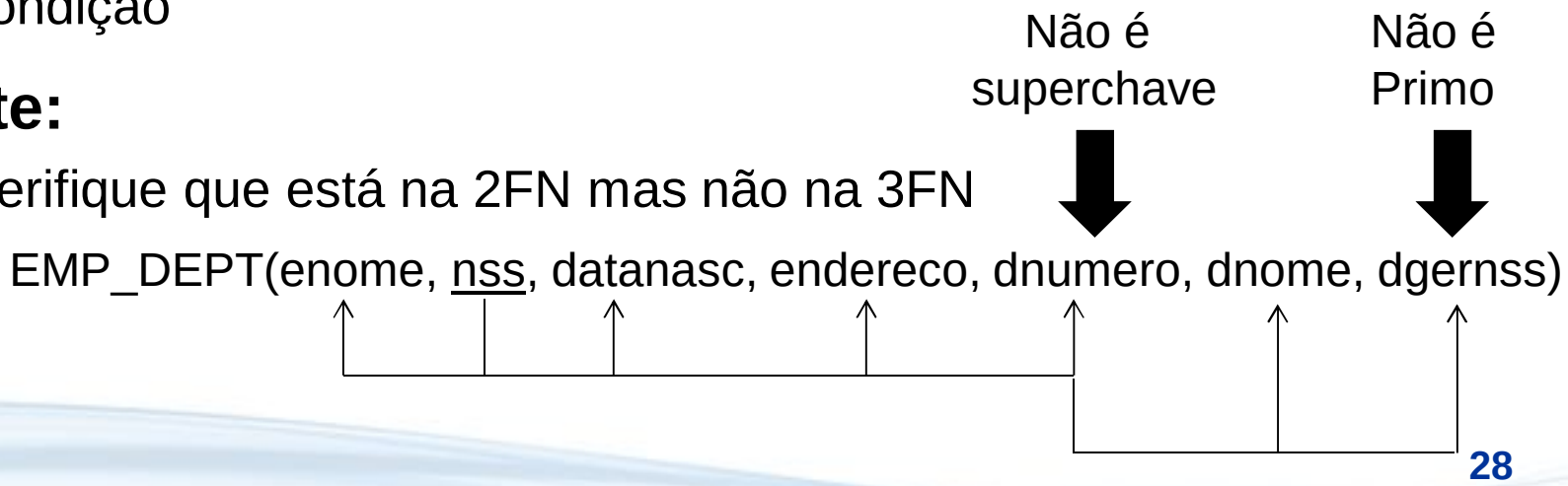
Definição Geral de Formas Normais

- **Redefinição da 2FN:**
  - Uma relação esquema R está na 2FN se todos os atributos não-primos, A, forem totalmente dependentes de todas as chaves de R
- **Teste:**
  - Verifique que EMP_PROJ não está na 2FN
    - EMP_PROJ (nss, pnúmero, horas, enome, pnome, plocalizacao)



Definição Geral de Formas Normais

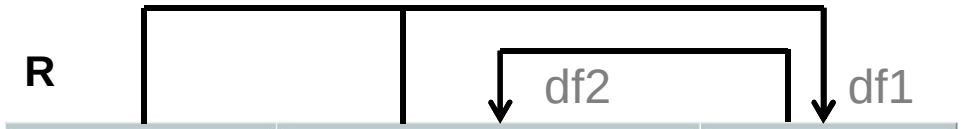
- **Redefinição de 3FN:**
  - Uma relação esquema R está na 3FN se, sempre que houver uma DF X→A, então uma das duas condições são válidas:
    - X é uma superchave de R, ou
    - A é um atributo primo de R
  - **NOTA:** A Forma normal de Boyce-Codd não admite a segunda condição
- **Teste:**
  - Verifique que está na 2FN mas não na 3FN



BCNF (Boyce-Codd Normal Form)

- **Definição de BCNF:**
  - Uma relação esquema R está na BCNF se, sempre que houver uma DF X→A em R, então X é uma superchave de R
- **Cada FN engloba a FN anterior:**
  - Toda relação em 2FN está na 1FN
  - Toda relação em 3FN está na 2FN
  - Toda relação em BCNF está na 3FN
- **Existem relações que estão na 3FN mas não em BCNF**
- **A meta é alcançar a BCNF ou 3FN em todas as relações**

Boyce-Codd normal form



ESTUDANDE	CURSO	INSTRUTOR
Nair	Banco de dados	Marcos
Silas	Banco de dados	Nico
Silas	Sistemas Operacionais	Altair
Silas	Teoria	Saulo
Wilson	Banco de Dados	Marcos
Wilson	Sistemas Operacionais	Álvaro
Wellington	Banco de Dados	Carlos
Zenaide	Banco de Dados	Nico

Relação em 3FN mas não em BCNF

Uma relação esquema R está na **3FN** se, sempre que houver uma DF X→A, então:

- X é uma superchave de R **ou**
- A é atributo primo de R.

Uma relação esquema R está na **BCNF** se, sempre que houver uma DF X→A, então:

- X é uma superchave de R

Alcançando a BCFN pela Decomposição

- **Existem duas DF em relação:**
  - df1: { estudante, curso } → instrutor
  - df2: instrutor → curso
  - Se a relação tivesse apenas df1, a relação estaria na BCNF.
  - Mas em df2, instrutor não é uma superchave, e, portanto, viola a BCNF, mas não a 3FN, pois curso é primo.
- **Uma relação que não esteja na BCNF deve ser decomposta para atender a esta propriedade, <u>mas</u> abdica da preservação das dependências funcionais nas relações decompostas**

Alcançando a BCFN pela Decomposição

- **Três possíveis decomposições para relação:**
  - R1(estudante, instrutor) e R2(estudante, curso)
  - R1(curso, instrutor) e R2(curso, estudante)
  - R1(instrutor, curso) e R2(instrutor, estudante)
- **Todas as três decomposições perdem a df1.**
  - Temos que conviver com este sacrifício, mas não podemos sacrificar a propriedade não-aditiva após a decomposição.
- **Das três, apenas a terceira decomposição não gera tuplas espúrias após a junção (join), e, assim, mantém a propriedade não-aditiva.**

Alcançando a BCFN pela Decomposição

R1

INSTRUTOR	ESTUDANDE
Marcos	Nair
Nico	Silas
Altair	Silas
Saulo	Silas
Marcos	Wilson
Álvaro	Wilson
Carlos	Wellington
Nico	Zenaide

X (JOIN)

R2

INSTRUTOR	CURSO
Marcos	Banco de dados
Nico	Banco de dados
Altair	Sistemas Operacionais
Saulo	Teoria
Álvaro	Sistemas Operacionais
Carlos	Banco de Dados

Relação original: R

ESTUDANDE	CURSO	INSTRUTOR
Nair	Banco de dados	Marcos
Silas	Banco de dados	Nico
Silas	Sistemas Operacionais	Altair
Silas	Teoria	Saulo
Wilson	Banco de Dados	Marcos
Wilson	Sistemas Operacionais	Álvaro
Wellington	Banco de Dados	Carlos
Zenaide	Banco de Dados	Nico

Note que para as outras possíveis decomposições, isso não acontece.

Decomposição sem perdas

- **A decomposição de R em X e Y é sem perdas se e somente se, pelo menos uma das duas DF for válida**
  - $X \cap Y \rightarrow X$ ou
  - $X \cap Y \rightarrow Y$
- **Caso especial**
  - Se $U \rightarrow V$, então a decomposição de R em UV e R – V é sem perdas.

Verifique que a decomposição de R satisfaz esta condição!

Algoritmo de Decomposição BCNF

- **Considere uma relação R e suas DFs associadas.**
  - Se X → Y violar a FNBC, decomponha R em XY e R − Y.
- **Aplicando esta idéia repetidamente, obteremos uma decomposição sem perdas de R em uma coleção de relações na BCNF.**
- **Em geral, mais de uma DF pode violar a BCNF. Dependendo da ordem em que as dependências são tratadas, podemos obter decomposições diferentes (e mesmo assim corretas).**

MVD e 4FN

- **As dependências multivaloradas são conseqüência da 1FN, a qual não aceita atributos multivalorados.**
  - Considere, por exemplo, a relação ACERVO abaixo:

ISBN	AUTOR	CÓPIAS
85-7323-169-6	Dantas	1, 2
0-13031-995-3	Molina, Ulman, Widom	1, 2

  - Relação Normalizada para BCNF (note que não há DFs)

ISBN	AUTOR	CÓPIAS
85-7323-169-6	Dantas	1
85-7323-169-6	Dantas	2
0-13031-995-3	Molina	1
0-13031-995-3	Molina	2
0-13031-995-3	Ulman	1
0-13031-995-3	Ulman	2
0-13031-995-3	Widom	1
0-13031-995-3	Widom	2

Mas ainda temos redundâncias por que?

Porque existem dependências multivaloradas!

ISBN →→ AUTOR
ISBN →→ CÓPIAS

MVD e 4FN

- **Sempre que X →→ Y ocorrer, dizemos que X multidetermina Y.**
- **Devido a semetria da definição, sempre que X →→ Y ocorrer em R, também ocorre X →→ Z.**
- **Por isso, X →→ Y implica X →→ Z; por isso, às vezes é escrito como X→→Y | Z.**
- **Então, na relação ACERVO do exemplo anterior:**
  - ISBN →→ AUTOR | CÓPIAS

MVD e 4FN

- **Elimina redundâncias provocadas pelas dependências multivaloradas (MVD).**
- **Uma relação está na 4FN se não contiver mais de uma MVD.**
  - Mas porque é tão ruim ter uma tabela com múltiplas dependências multivaloradas?
  - Em ACERVO
    - Para inserir mais uma cópia do ISBN 0-13031-995-3, será necessário inserir 3 tuplas, uma para cada autor.

ISBN	AUTOR	CÓPIAS
85-7323-169-6	Dantas	1
85-7323-169-6	Dantas	2
0-13031-995-3	Molina	1
0-13031-995-3	Molina	2
0-13031-995-3	Ulman	1
0-13031-995-3	Ulman	2
0-13031-995-3	Widom	1
0-13031-995-3	Widom	2

Alterações e Remoções carecem com mesmo problema

MVD e 4FN

- **A solução é decompor a relação ACERVO em duas**

De acordo com as MVD
ISBN →→ AUTOR
ISBN →→ CÓPIAS

ISBN	AUTOR
85-7323-169-6	Dantas
0-13031-995-3	Molina
0-13031-995-3	Ulman
0-13031-995-3	Widom

ISBN	CÓPIAS
85-7323-169-6	1
85-7323-169-6	2
0-13031-995-3	1
0-13031-995-3	2

- **A MVD desejável é aquele cujo determinante é superchave da relação**

Caso especial
Se R tiver as seguintes MVD
A→→ B e B →→ C
Neste caso, R estará na 4FN se, e somente se, B e C são dependentes um do outro.

5FN e Dependência de Junção

- **Algumas vezes uma relação não pode ser decomposta sem perdas em duas relações, mas pode ser decomposta em três ou mais.**
- **A 5FN capta a idéia de que uma relação esquema deve ter alguma decomposição sem perda (dependência de junção).**
- **Encontrar casos reais da 5FN é difícil.**

5FN e Dependência de Junção

- **Um pequeno exemplo**

AEP

AGENTE	EMPRESA	PRODUTO
Smith	Ford	Carro
Smith	Ford	Caminhão
Smith	GM	Carro
Smith	GM	Caminhão
Jones	Ford	Carro

Regra:
Se um AGENTE vende um certo PRODUTO e este AGENTE representa uma EMPRESA que faz este PRODUTO

então

O AGENTE deve vender o PRODUTO para a EMPRESA.

AGENTES representam EMPRESAS
EMPRESAS fazem PRODUTOS
AGENTES vendem PRODUTOS

5FN e Dependência de Junção

- **Um pequeno exemplo**

AEP

AGENTE	EMPRESA	PRODUTO
Smith	Ford	Carro
Smith	Ford	Caminhão
Smith	GM	Carro
Smith	GM	Caminhão
Jones	Ford	Carro

AE

AGENTE	EMPRESA
Smith	Ford
Smith	GM
Jones	Ford

EP

EMPRESA	PRODUTO
Ford	Carro
Ford	Caminhão
GM	Carro
GM	Caminhão

AP

AGENTE	PRODUTO
Smith	Carro
Smith	Caminhão
Jones	Carro

AEP= AE * EP * AP

5FN e Dependência de Junção

- **Dependência de Junção**
  - Uma relação R satisfaz a dependência de junção
    - JD (R_1, R_2, ..., R_n) se
      - $R = R_1 * R_2 * ... * R_n$
    - onde R_1, R_2, ..., R_n são subconjuntos dos atributos de R.
  - Note que uma dependência multivalorada é um caso especial de dependência de junção (n=2).

5FN e Dependência de Junção

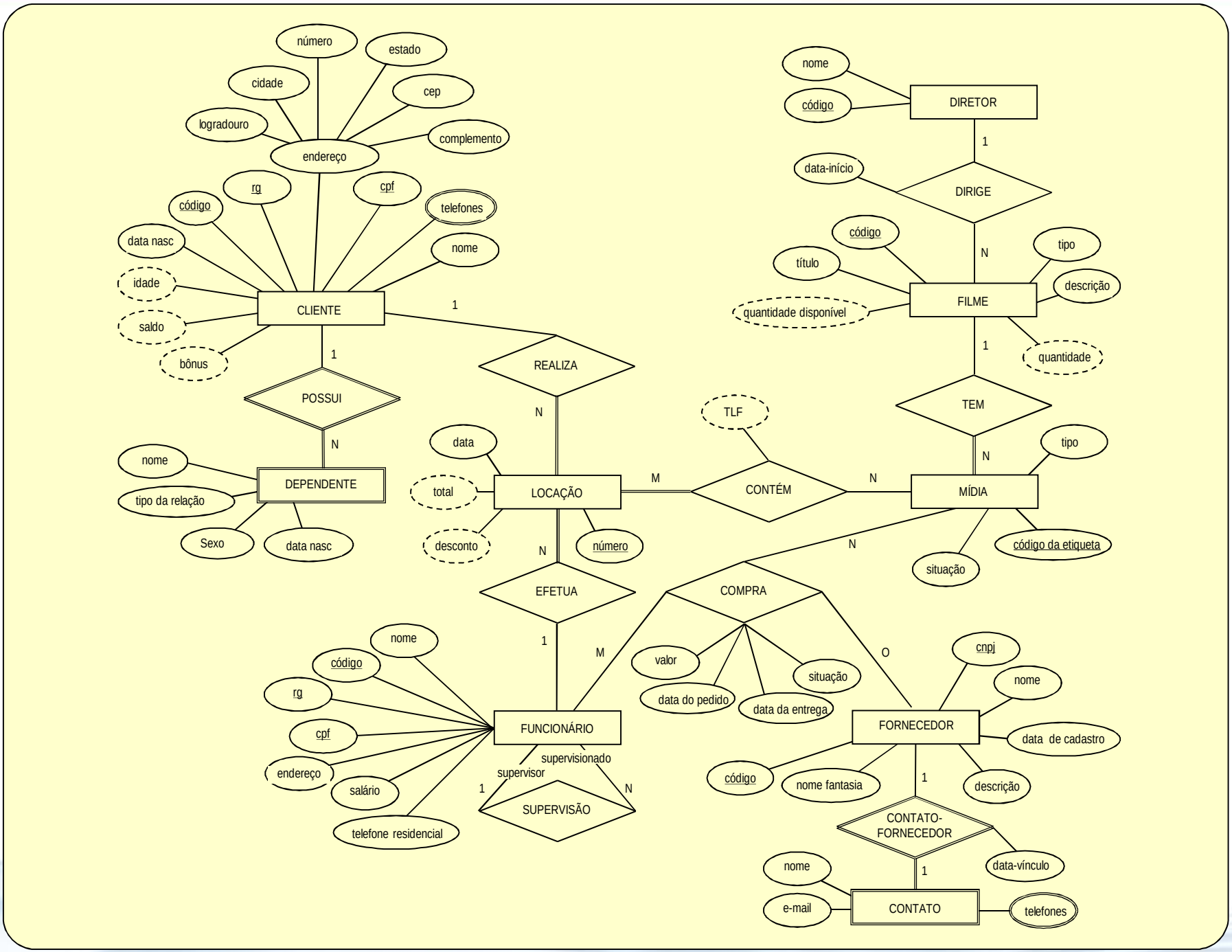
- **Uma relação R está na 5FN se, e somente se, ela estiver na 4FN e todas as suas dependências de junção forem determinadas pelas chaves candidatas.**
- **A descoberta de DJs em bancos de dados reais com centenas de atributos é praticamente impossível. Isso poderá ser feito apenas contando com um grande grau de intuição sobre os dados por parte do projetista. Por isso, a prática atual de projeto de banco de dados dá pouca atenção a elas (Elmasri & Navathe 4ª. Edição).**

Outras NFs

- **Existem outras formas normais, porém elas estão fora do escopo desta disciplina, pois são formas pouco utilizadas em projetos de banco de dados devido a sua dificuldade de aplicação prática.**

Questões de Estudo

- **Dado o DER de uma locadora de vídeo (próximo slide), e o mapeamento realizado para o esquema do BD Relacional, verifique a qualidade das relações obtidas (qual forma normal atingida) e, se necessário, normalize todos os esquemas de relações para a 3FN ou, se possível, para a BCNF.**


número
estado
cidade
cep
logradouro
complemento
endereço
rg
cpf
código
telefones
data nasc
nome
idade
CLIENTE
saldo
bônus
1
1
POSSUI
REALIZA
N
N
nome
tipo da relação
DEPENDENTE
Sexo
data nasc
data
total
LOCAÇÃO
desconto
N
número
EFETUA
1
M
TLF
M
CONTÉM
N
nome
DIRETOR
código
1
data-início
DIRIGE
N
código
tipo
título
descrição
FILME
quantidade disponível
1
quantidade
TEM
N
tipo
MÍDIA
N
código da etiqueta
situação
COMPRA
O
valor
situação
data do pedido
data da entrega
cnpj
nome
data de cadastro
FORNECEDOR
código
nome fantasia
1
descrição
CONTATO-
FORNECEDOR
1
data-vínculo
nome
e-mail
CONTATO
telefones
nome
código
rg
cpf
endereço
salário
telefone residencial
FUNCIONÁRIO
supervisionado
supervisor
1
N
SUPERVISÃO

Normalização

- **Referências Bibliográficas**

1. Elmasri, R.; Navathe, S. B. [Trad.]. Sistemas de bancos de dados. Traduzido do original: FUNDAMENTALS OF DATABASE SYSTEMS. São Paulo: Pearson(Addison Wesley), 2005. 724 p. ISBN: 85-88639-17-3.
2. Korth, H.; Silberschatz, A. Sistemas de Bancos de Dados. 3a. Edição, Makron Books, 1998.
3. Raghu Ramakrishnan e Johannes Gehrke, Database Management Systems, Second Edition, McGraw-Hill, 2000.
4. Teorey, T.; Lightstone, S.; Nadeau, T. Projeto e modelagem de bancos de dados. Editora Campus, 2007.

- **Referências Web**

1. Takai, O.K; Italiano, I.C.; Ferreira, J.E. Introdução a Banco de Dados. Apostila disponível no site: http://www.ime.usp.br/~jef/apostila.pdf. (07/07/2005).

Pós-Graduação
Engenharia de Software

Obrigado!

Prof. Gustavo Bianchi Maia
gbmaia@gmail.com