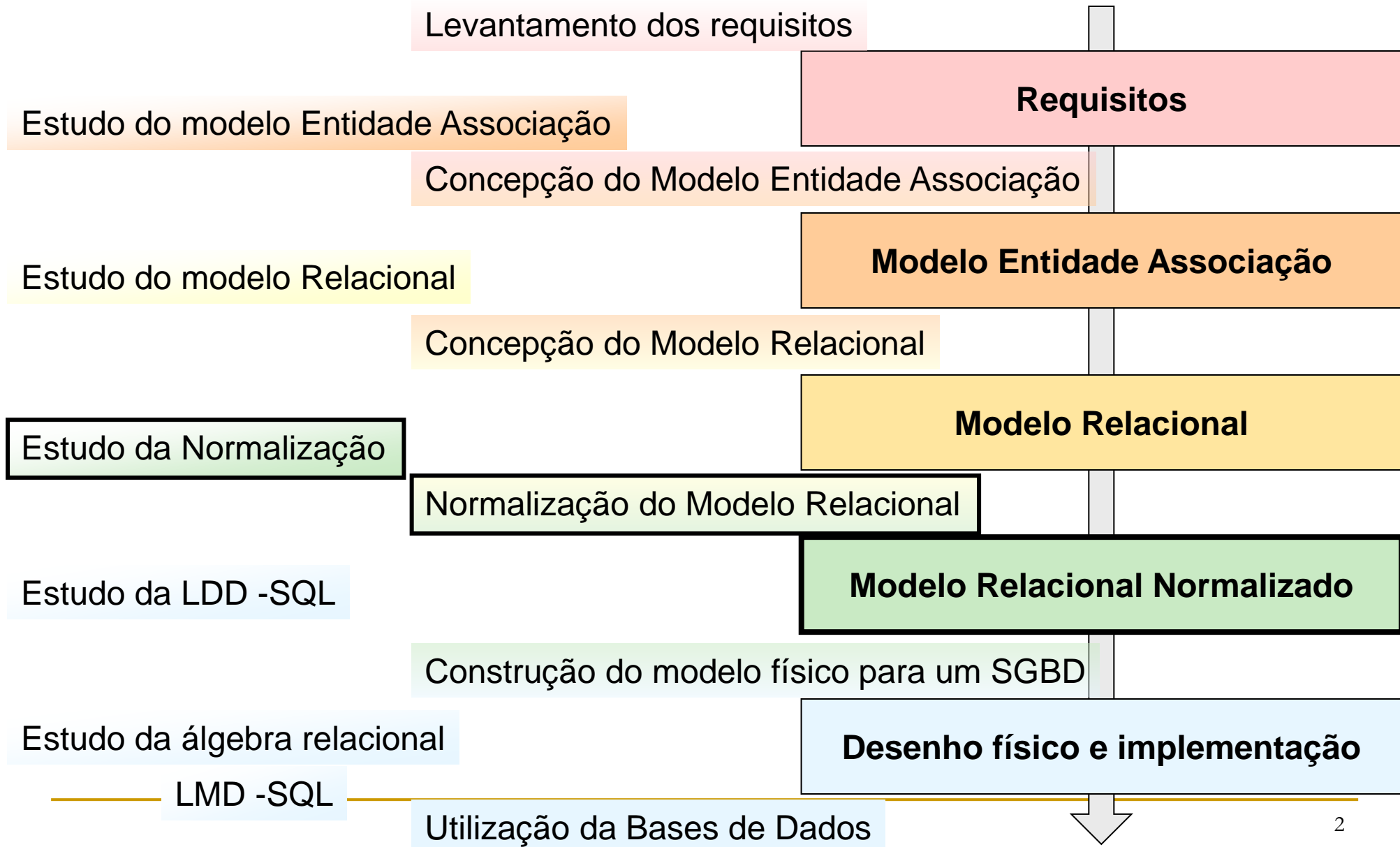


08 – Normalização

Baseado nos slides do professor Paulo Trigo
Todas as alterações são da responsabilidade do professor António Teófilo

Etapas do processo



Estratégias de construção do Modelo

Conceptual e modelo lógico (relacional)

- Do Geral para o Particular (*top-down*) - geralmente usado em grandes projectos
 - Construção do Modelo Modelo Entidade - Associação
 - Construção do Modelo Lógico – relacional
- Do Particular para o Geral (*bottom-up*) - geralmente usado em pequenos projectos (até 6 ou 8 tabelas)
 - Identificação de uma Relação Universal (contendo todos os atributos)
 - Análise de Dependências Funcionais
 - Construção do Modelo Lógico – relacional

Redundância e Anomalias

- Informação sobre fornecedores e respectivos produtos

R1

<u>nomeFornecedor</u>	<u>endereço</u>	<u>produto</u>	<u>preço</u>
Zorra, Lda	Rua do Arrozal, Nº 33	Arroz Agulha	220
Zorra, Lda	Rua do Arrozal, Nº 33	Arroz Normal	180
Dad&Co	Av. das Descobertas	Farinha Milho	140
Folha&Grão	Travessa Encoberta, 12	Arroz Agulha	190
Folha&Grão	Travessa Encoberta, 12	Café	270

- A “Redundância” é um problema sério desta Relação
 - ❑ o endereço do fornecedor está repetido por cada produto que fornece.
- Este problema está na origem de três tipos de Anomalias:
 - ❑ Alteração
 - ❑ Inserção
 - ❑ Remoção

Anomalias

<u>nomeFornecedor</u>	<u>endereço</u>	<u>produto</u>	<u>preço</u>
Zorra, Lda	Rua do Arrozal, Nº 33	Arroz Agulha	220
Zorra, Lda	Rua do Arrozal, Nº 33	Arroz Normal	180
Dad&Co	Av. das Descobertas	Farinha Milho	140
Folha&Grão	Travessa Encoberta, 12	Arroz Agulha	190
Folha&Grão	Travessa Encoberta, 12	Café	270

■ Anomalia de Alteração

- ❑ Como fazer para alterar o endereço do fornecedor 'Zorra, Lta' ?
- ❑ E se essa alteração “ficar esquecida” para algum tuplo ?

■ Anomalia de Inserção

- ❑ Como fazer para registar um novo fornecedor ?
- ❑ E se ele “ainda” não fornece nenhum produto ?
- ❑ E se colocarmos NULL nos atributos 'produto' e 'preço' ?
- ❑ E se ao inserir o primeiro tuplo correspondente ao primeiro produto fornecido, “nos esquecermos” de apagar “o tal” tuplo com NULL ?
- ❑ Mas os atributos 'produto' e 'nomeFornecedor' são chave da Relação, pelo que nenhum deles pode ser NULL !

■ Anomalia de Remoção

- ❑ Como fazer para remover todos os produtos de um fornecedor ?
- ❑ E que acontece à informação relativa a esse fornecedor ?

Decomposição

- Como ultrapassar as anomalias detectadas ?
 - Decompondo a Relação em “outras” Relações
- Como decompor a Relação ?
 - Analisando primeiro aquilo que não pode ser decomposto
- O que não pode ser decomposto ?
 - ‘nomeFornecedor’ e ‘endereço’ não podem ser separados
 - para determinado ‘nomeFornecedor’ o seu endereço não varia
 - ‘nomeFornecedor’, ‘produto’ e ‘preço’ não podem ser separados
 - para determinado ‘nomeFornecedor’ cada ‘produto’ por ele fornecido tem o seu ‘preço’

Decomposição (cont.)

- Partindo de,
 - FORNECEDOR (nomeFornecedor, endereco, produto, preco)
- Mantendo “junto” aquilo que não pode ser decomposto, ficamos com,
 - FORNECEDOR (nomeFornecedor, endereco)
 - PRODUTO_FORNECIDO (nomeFornecedor, produto, preco)
- Agora pretende-se saber quais os nomes e os endereços dos fornecedores de ‘Café’
 - $\pi_{\text{nome, endereco}} (\text{FORNECEDOR} \bowtie (\sigma_{\text{produto} = \text{'Café'}} (\text{PRODUTO_FORNECIDO})))$
- Para responder à questão foi necessário efectuar uma operação de Junção (*Join*): o que é uma operação complexa
 - existe um compromisso entre a eliminação de Anomalias e a eficiência

Dependências entre Dados

- O conhecimento necessário ao processo de decomposição está traduzido nas dependências existentes entre os dados.
- Existem três tipos de dependências entre os dados:
 - Funcional
 - Multivalor
 - Junção
- Com base no conceito associado a cada um dos três tipos de dependência, vão-se estabelecer as “Formas Normais” (que posteriormente serão apresentadas)
 - Conceito de “Dependência Funcional” para estabelecer a “2ª Forma Normal”, a “3ª Forma Normal” e a “Forma Normal de *Boyce-Codd*”
 - Conceito de “Dependência Multivalor” para estabelecer a “4ª Forma Normal”
 - Conceito de “Dependência de Junção” para estabelecer a “5ª Forma Normal”

Dependência Funcional

* Note-se que o inverso poderá não ser verdade !

- Existe uma dependência funcional do atributo A para o atributo B, se a um valor de A corresponder sempre um e só um valor de B.
 - Diz-se que A identifica B
 - Representa-se por: $A \rightarrow B$
- Exemplo:
 - em qualquer instante e em qualquer ponto da Base de Dados onde figurem os atributos **numeroContribuinte** e **nomeCliente**, sabe-se que a determinado **numeroContribuinte** corresponderá sempre necessariamente o mesmo **nomeCliente** *. Então existe uma dependência funcional de **numeroContribuinte** para **nomeCliente**: **numeroContribuinte** \rightarrow **nomeCliente**

- Dependências funcionais em R1 :

- **nomeFornecedor** \rightarrow **endereço**
- {**nomeFornecedor**, **produto**} \rightarrow **preço**

R1

<u>nomeFornecedor</u>	<u>endereço</u>	<u>produto</u>	<u>preço</u>
Zorra, Lda	Rua do Arrozal, Nº 33	Arroz Agulha	220
Zorra, Lda	Rua do Arrozal, Nº 33	Arroz Normal	180
Dad&Co	Av. das Descobertas	Farinha Milho	140
Folha&Grão	Travessa Encoberta, 12	Arroz Agulha	190
Folha&Grão	Travessa Encoberta, 12	Café	270

Dependência Funcional - Definição

- Sejam A_1, A_2, \dots, A_n todos os atributos existentes na base de dados
- Seja $R(A_1, A_2, \dots, A_n)$ um Esquema de Relação Universal *
 - Esquema de Relação com todos os Atributos
- Sejam X e Y subconjuntos, não vazios, de $\{A_1, A_2, \dots, A_n\}$
- Y é Funcionalmente Dependente de X e escreve-se $X \rightarrow Y$ se,
 - para quaisquer dois tuplos t_1 e t_2 em R tal que $t_1[X] = t_2[X]$, então
 - tem-se também $t_1[Y] = t_2[Y]$
- Formalmente $X \rightarrow Y$ se,
 - $\forall (t_1, t_2), t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$

* É o esquema que contém todos os atributos da base de dados

Dependência Funcional - Convenções

- Se $X \rightarrow Y$ diz-se que,
 - Existe uma Dependência Funcional de X para Y
 - X identifica Y
 - Os valores de X determinam univocamente os valores de Y
 - A cada valor de X está associado um e um só valor de Y
 - Y é funcionalmente depende de X
 - Os valores de Y “dependem de / são determinados por” os valores de X

- Se $X \rightarrow Y$ diz-se que,
 - X é o “lado esquerdo” da Dependência Funcional
 - Y é o “lado direito” da Dependência Funcional

- A sigla DF é a abreviatura de “Dependência Funcional”

Identificação de Dependências Funcionais

- Normalmente a identificação de Dependências Funcionais não é obtida apenas a partir da análise das linhas de uma tabela mas:
 - Através das próprias propriedades das Atributos.

EMPREGADO

numero	nome	apelido	salario
1001	Sofia	Martins	900
5509	João	Martins	300
1234	Ana	Silva	100
2003	Sofia	Antunes	454

- “apelido” é Funcionalmente Dependente de “numero” ?
 - $\text{numero} \rightarrow \text{apelido}$?
 - Sim. Conhecendo o número do empregado (“numero” é unívoco), fica determinado o seu “apelido” (um empregado só pode ter um apelido).

- $\text{nome} \rightarrow \text{apelido}$?

$$X \rightarrow Y \text{ se, } \forall (t1, t2), t1[X] = t2[X] \Rightarrow t1[Y] = t2[Y]$$

- $t1[\text{nome}] = t2[\text{nome}] = \text{'Sofia'}$
- $(t1[\text{apelido}] = \text{'Martins'}) \neq (t2[\text{apelido}] = \text{'Antunes'})$
- $\therefore \text{nome} \not\rightarrow \text{apelido}$

$\text{numero} \rightarrow \text{nome}, \text{numero} \rightarrow \text{apelido}, \text{numero} \rightarrow \text{salario}$

Identificação de Dependências Funcionais (cont.)

- Qual o(s) atributo(s) que determinam o atributo “preço” ?

VENDE

supermercado	artigo	preço
Continente	Arroz	150
Feira Nova	Detergente Loixa	300
Pingo Doce	Borba Tinto	700
Pão de Açúcar	Arroz	157
Continente	Batatas	300

- artigo → preço ?
 - ❑ Não. Cada artigo pode ter preços diferentes consoante o supermercado
- supermercado → preço ?
 - ❑ Não. Cada supermercado pode ter um preço para cada artigo que vende
- “preço” depende funcionalmente de ambos
 - ❑ {supermercado, artigo} → preço

Álgebra Relacional e Dependência Funcional

- Como analisar uma Relação, para provar que não existe uma determinada Dependência Funcional ?
- Se existe $X \rightarrow Y$, então:
 - $\forall (t1, t2), t1[X] = t2[X] \Rightarrow t1[Y] = t2[Y]$
- Se não existe $X \rightarrow Y$, então:
 - $\neg \forall (t1, t2), t1[X] = t2[X] \Rightarrow t1[Y] = t2[Y]$
- Como $\neg \forall (z), p(z)$ é equivalente a $\exists (z), \neg p(z)$, como $P(z)$ é a expressão $A \Rightarrow B$ e que é equivalente a $\neg A \vee B$, fica $\exists (z), \neg (\neg A \vee B)$, que resulta em $\exists (z), A \wedge \neg B$
- Então, uma DF ($X \rightarrow Y$) não existe se: $\exists (t1, t2), t1[X] = t2[X] \wedge t1[Y] \neq t2[Y]$
- Que se lê: se existirem dois tuplos, tal que os atributos do conjunto X sejam iguais, mas que os respectivos atributos do conjunto Y sejam diferentes, então não existe dependência funcional entre X e Y

Operadores relacionais e Dependência Funcional

- Considere a seguinte Relação:

R1

atrA	atrB	atrC
a	b	b
a	b	c
c	c	c
e	d	b

- Será que existe: $\{atrA, atrB\} \rightarrow atrC$?
- Não existirá a DF se,
 - $\exists (t1, t2), t1[AtrA, AtrB] = t2[AtrA, AtrB] \wedge t1[AtrC] \neq t2[AtrC]$
- E existem t1 e t2 obtendo a Relação resultante de $R1 \bowtie_{1=1 \wedge 2=2 \wedge 3 \neq 3} R1$
- De modo geral uma DF $\{a_1, a_2, \dots, a_n\} \rightarrow b$ existe numa relação R se:
 - $R \bowtie_{a1=a1 \wedge a2=a2 \wedge \dots \wedge an=an \wedge b \neq b} R$

Manipulação das DFs

Transformação de um conjunto de DFs
noutro conjunto de DFs

Manipulação de Dependências Funcionais

- A manipulação de DF permite reduzir ou alterar um dado conjunto de DF transformando-o num outro conjunto equivalente
- Para isso recorre-se a seis regras (de R1 até R6) de inferência:

- (R1) Reflexividade

- $Y \subseteq X \Rightarrow X \rightarrow Y$

$$X \supseteq Y \Rightarrow X \rightarrow Y$$

- (R2) Aumento

- $X \rightarrow Y \wedge W \subseteq Z \Rightarrow XZ \rightarrow YW$

$$X \rightarrow Y \wedge Z \supseteq W \Rightarrow XZ \rightarrow YW$$

- (R3) Transitividade

- $X \rightarrow Y \wedge Y \rightarrow Z \Rightarrow X \rightarrow Z$

$$X \rightarrow Y, \forall (t1, t2), t1[X] = t2[X] \Rightarrow t1[Y] = t2[Y]$$

Manipulação de Dependências Funcionais (cont.)

- (R4) Decomposição

- $X \rightarrow YZ \Rightarrow X \rightarrow Y \wedge X \rightarrow Z$

- (R5) União

- $X \rightarrow Y \wedge X \rightarrow Z \Rightarrow X \rightarrow YZ$

- (R6) Pseudo-transitividade

- $X \rightarrow Y \wedge YW \rightarrow Z \Rightarrow XW \rightarrow Z$

- As regras R1, R2 e R3 são conhecidas por:

- “Regras de Inferência de *Armstrong*”, ou por

- “Axiomas de *Armstrong*”, embora não sejam axiomas no sentido matemático do termo.

- As regras R4, R5 e R6 são conhecidas por:

- “Propriedades Derivadas” (dos “Axiomas de *Armstrong*”)

Prova das Regras de Inferência

- Prova de R1 (Reflexividade: se $X \supseteq Y$ então $X \rightarrow Y$)
 - Admita-se que $X \supseteq Y$ e que existem dois tuplos $t1$ e $t2$ de R tais que $t1[X]=t2[X]$. Então, como $X \supseteq Y$, ainda se tem $t1[Y]=t2[Y]$, logo $X \rightarrow Y$
 - As dependências $X \rightarrow Y$, com $X \supseteq Y$ denominam-se de dependências triviais
- Prova de R2 (Aumento: se $X \rightarrow Y$ e $Z \supseteq W$ então $XZ \rightarrow YW$)
 - Esta prova será feita por contradição.
 - Admitindo que se tem $X \rightarrow Y$, mas que não se tem $XZ \rightarrow YW$,
 - teriam que existir dois tuplos $t1$ e $t2$ tais que:
 - (a) $t1[X] = t2[X]$,
 - (b) $t1[Y] = t2[Y]$,
 - (c) $t1[XZ] = t2[XZ]$,
 - (d) $t1[YW] \neq t2[YW]$
 - isso não é possível pois,
 - de (a) e de (c) conclui-se (e) $t1[Z] = t2[Z]$,
 - de (b) e de (e) conclui-se que $t1[YZ] = t2[YZ]$, o que contradiz (d) pois tendo-se $Z \supseteq W$ certamente que ainda se terá $t1[YW] = t2[YW]$

Prova das Regras de Inferência (cont.)

- Prova de R3 (Transitividade: se $X \rightarrow Y$ e $Y \rightarrow Z$ então $X \rightarrow Z$)
 - Admitindo que se tem $X \rightarrow Y$ e $Y \rightarrow Z$,
 - para quaisquer tuplos $t1$ e $t2$ tem-se que:
 - se $t1[X]=t2[X]$ então $t1[Y]=t2[Y]$
 - e como $t1[Y]=t2[Y]$ tem-se também $t1[Z]=t2[Z]$
 - pelo que $X \rightarrow Z$

- As provas de R4, R5 e R6 serão feitas à custa de R1, R2 e R3.

- Prova de R4 (Decomposição: se $X \rightarrow YZ$ então $X \rightarrow Y$ e $X \rightarrow Z$)
 - 1) $X \rightarrow YZ$
 - 2) $YZ \rightarrow Y$ (usando R1 e sabendo que $Y \subseteq YZ$)
 - 3) $X \rightarrow Y$ (usando R3, em 1 e 2)
 - de forma análoga se prova que $X \rightarrow Z$

R1 – reflexividade
R3 – transitividade

Prova das Regras de Inferência (cont. 1)

- Prova de R5 (União: se $X \rightarrow Y$ e $X \rightarrow Z$ então $X \rightarrow YZ$)
 - 1) $X \rightarrow Y$
 - 2) $X \rightarrow Z$
 - 3) $X \rightarrow XY$ (usando R2 em 1 aumentado com X - note-se que $XX = X$)
 - 4) $XY \rightarrow YZ$ (usando R2 em 2 aumentado com Y)
 - 5) $X \rightarrow YZ$ (usando R3 em 3 e 4)

 - Prova de R6 (Pseudo-transitividade: se $X \rightarrow Y$ e $YW \rightarrow Z$ então $XW \rightarrow Z$)
 - 1) $X \rightarrow Y$
 - 2) $YW \rightarrow Z$
 - 3) $XW \rightarrow YW$ (usando R2 em 1 aumentado com W)
 - 4) $XW \rightarrow Z$ (usando R3 em 3 e 2)
- R2 – aumento
R3 – transitividade

Representação de DF como um Grafo

Representação de DF como um Grafo

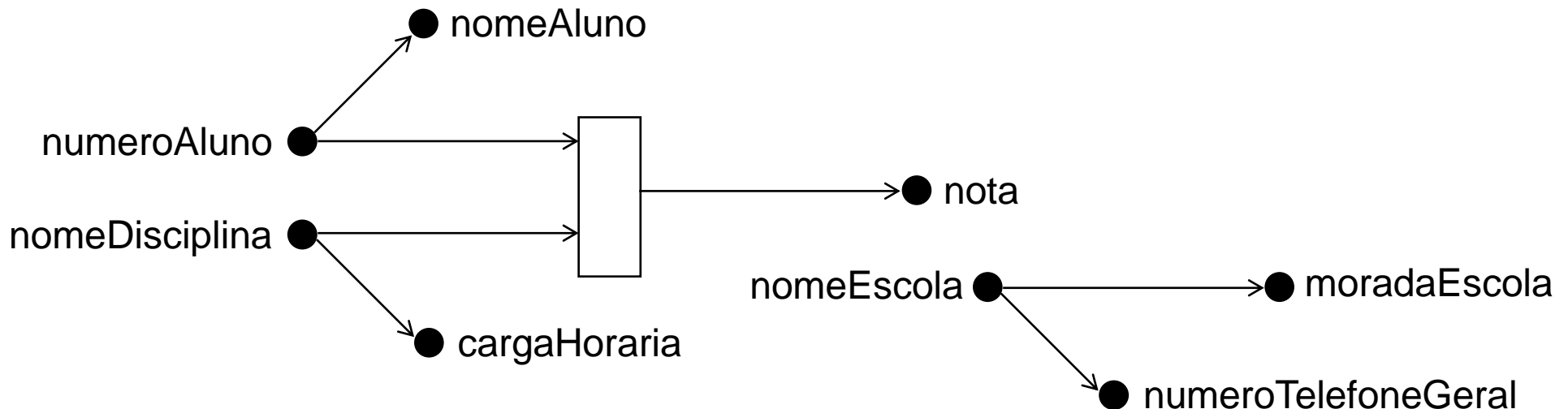
- Seja,
 - $\{A_1, A_2, \dots, A_n\}$ um conjunto de Atributos e
 - $F = \{X_i \rightarrow A_j\}$ um conjunto de Dependências Funcionais

- F pode ser representado como um grafo, tal que,
 - a todo o Atributo A_i corresponde,
 - um nó com a marca A_i
 - a toda a DF, $X_i \rightarrow A_j$ onde $X_i = \{A_{i1}, \dots, A_{ip}\}$ com $p > 1$, corresponde,
 - um nó auxiliar composto com a marca X_i e
 - um conjunto de arcos que ligam os vários A_{i1}, \dots, A_{ip} a X_i e
 - um arco que liga X_i ao nó A_j
 - a toda a DF, $X_i \rightarrow A_j$ corresponde,
 - um arco de X_i (um vértice simples ou composto) para A_j

Representação de DF como um Grafo (cont.)

■ Seja,

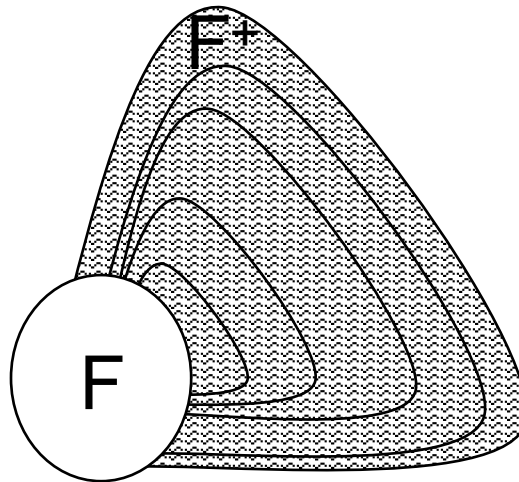
- {numeroAluno, nomeAluno, nomeDisciplina, nota, cargaHoraria, nomeEscola, moradaEscola, numeroTelefoneGeral}
- $F = \{ \text{numeroAluno} \rightarrow \text{nomeAluno},$
 $\{\text{numeroAluno}, \text{nomeDisciplina}\} \rightarrow \text{nota},$
 $\text{nomeDisciplina} \rightarrow \text{cargaHoraria},$
 $\text{nomeEscola} \rightarrow \{\text{moradaEscola}, \text{numeroTelefoneGeral}\} \}$



Fecho de um conjunto de DFs

Fecho de um conjunto de Dependências Funcionais

- Seja F um conjunto de Dependências Funcionais
- Designa-se por Fecho de F (representa-se F^+) ao conjunto de todas as Dependências Funcionais logicamente implicadas por F
 - $F^+ = \{ X \rightarrow Y \mid F \models X \rightarrow Y \}$



- F^+ constrói-se partindo de F e, por aplicação das “Regras de Inferência de *Armstrong*”, expandindo cada novo conjunto obtido

$F^+ = \{ X \rightarrow Y \mid F \models X \rightarrow Y \}$, F^+ é o conjunto de todas as dependências $X \rightarrow Y$, tal que $X \rightarrow Y$ é uma implicação lógica de F , ou seja, de F pode-se inferir (por lógica) $X \rightarrow Y$

Fecho de um conjunto de DF - Exemplo

- Considere-se,
 - $R(A, B, C)$ com as Dependências Funcionais F
 - $F = \{ A \rightarrow B, B \rightarrow C \}$
- Qual será o F^+ , ou seja o Fecho de F :
 - $F^+ = \{ A \rightarrow A, A \rightarrow B, A \rightarrow C, AB \rightarrow A, AB \rightarrow B, AB \rightarrow C, AB \rightarrow AB, AB \rightarrow AC, AB \rightarrow ABC, AC \rightarrow A, AC \rightarrow B, AC \rightarrow C, AC \rightarrow AC, AC \rightarrow AB, AC \rightarrow ABC, ABC \rightarrow A, ABC \rightarrow B, ABC \rightarrow C, ABC \rightarrow AB, ABC \rightarrow BC, ABC \rightarrow AC, ABC \rightarrow ABC, B \rightarrow B, B \rightarrow C, B \rightarrow BC, BC \rightarrow B, BC \rightarrow C, BC \rightarrow BC, C \rightarrow C, A \rightarrow \emptyset, B \rightarrow \emptyset, C \rightarrow \emptyset, AB \rightarrow \emptyset, AC \rightarrow \emptyset, BC \rightarrow \emptyset, ABC \rightarrow \emptyset, \dots \}$
- É muito difícil ao construir F^+ não nos “escapar” alguma DF !
- É muito difícil “lidar” com tantas Dependências Funcionais !

Verificar se determinada DF pertence a F^+

- Será que $X \rightarrow Y$ pertence a F^+ ?
- Para responder a esta questão podemos,
 - partir de F , calcular todo o F^+ e depois verificar se $X \rightarrow Y$ lá está
 - mas já vimos que o cálculo de F^+ não é simples !
- Um modo sistemático de verificar se determinada Dependência Funcional $X \rightarrow Y$ pertence a F^+ consiste em,
 - considerar o lado esquerdo X da Dependência Funcional
 - usando F , aplicar as “Regras de Inferência de *Armstrong*” a X , para encontrar o conjunto W de todos os Atributos funcionalmente dependentes de X
 - se W contiver Y então podemos concluir que $X \rightarrow Y$ pertence a F^+
- Exemplo:
 - Sendo $F = \{ A \rightarrow BD, AB \rightarrow C \}$, será que $A \rightarrow C$ pertence a F^+ ?
 - $A^+ = \{ A, B, D, C \}$ contém $\{C\}$, portanto $A \rightarrow C$ pertence a F^+

Fecho de um conjunto de Atributos

- Seja X um conjunto de Atributos
- Designa-se por Fecho X (representa-se por X^+) ao conjunto de todos os Atributos que em F são funcionalmente dependentes de X
 - $X^+ = \{ A \mid F \models X \rightarrow A \}$
- Algoritmo para determinar o fecho (X^+) de X em F ,
 $X^+ := X$
 repetir
 Alterações = FALSE
 Para todas as DFs $Y \rightarrow Z$ em F
 Com cada uma fazer
 se $Y \subseteq X^+$ e $Z \notin X^+$ então $\{ X^+ := X^+ \cup Z; \text{Alterações} = \text{TRUE} \}$
 fimfazer
 enquanto (Alterações == TRUE)

Fecho de um conjunto de atributos – Exemplo

- Seja,
 - $F = \{ \text{numeroEmpregado} \rightarrow \text{nomeEmpregado},$
 $\text{numeroProjecto} \rightarrow \{ \text{nomeProjecto}, \text{localizacao} \},$
 $\{ \text{numeroEmpregado}, \text{numeroProjecto} \} \rightarrow \text{horasEstimadas} \}$
 - $\{ \text{numeroEmpregado} \}^+ = \{ \text{numeroEmpregado}, \text{nomeEmpregado} \}$
 - $\{ \text{numeroProjecto} \}^+ = \{ \text{numeroProjecto}, \text{nomeProjecto}, \text{localizacao} \}$
 - $\{ \text{numeroEmpregado}, \text{numeroProjecto} \}^+ =$
 $\{ \text{numeroEmpregado}, \text{numeroProjecto}, \text{nomeEmpregado},$
 $\text{nomeProjecto}, \text{localizacao}, \text{horasEstimadas} \}$

Fecho de um conjunto de atributos – Exemplo 2

- Considerando o esquema de relação R e as dependências funcionais em F, determine o fecho de B e o fecho de AF

R (A, B, C, D, E, F)

F = { $AF \rightarrow C$, $BE \rightarrow D$, $B \rightarrow C$, $C \rightarrow BE$ }

B⁺ iterando:

B⁺ = {B}

B⁺ = {BC}

B⁺ = {BCE}

B⁺ = {BCED}, valor final

AF⁺ iterando:

AF⁺ = {AF}

AF⁺ = {AFC}

AF⁺ = {AFCBE}

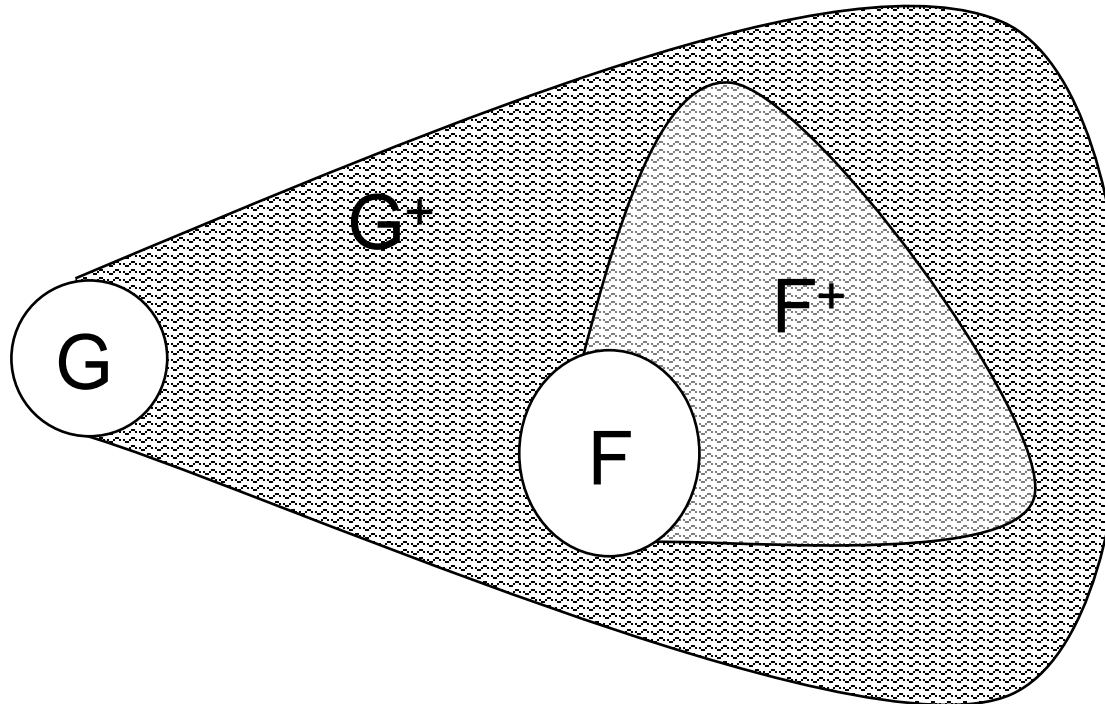
AF⁺ = {AFCBED}, valor final

Cobertura de um conjunto de Dependências Funcionais

Cobertura mínima

Cobertura de um conjunto de Dependências Funcionais

- Sejam G e F dois conjuntos de Dependências Funcionais
- Diz-se que G é uma Cobertura de F se,
 - o Fecho de G (G^+) contiver todas as Dependências Funcionais de F
 - notar que G^+ será maior ou igual que F^+ ($G^+ \supseteq F^+$)



Verificar se G é Cobertura de F

- Será que G é Cobertura de F ?
- Para responder a esta questão podemos,
 - calcular G^+ e F^+ e depois verificar se G^+ contém todo o F^+
 - mas já vimos que o cálculo de G^+ e F^+ não é simples !
- Um modo sistemático de verificar se G é Cobertura de F consiste em
 - para cada Dependência Funcional $X \rightarrow Y$ em F ,
 - usando as Dependências Funcionais de G , calcular X^+
 - se $X^+ \supseteq Y$
 - então, continuar o ciclo (ou seja, verificar a próxima DF de F)
 - caso contrário, concluir que G não é Cobertura de F
- G é uma Cobertura de F se durante a aplicação deste modo sistemático apresentado, nunca se concluir o contrário
 - G é uma Cobertura de F se. $\forall X \rightarrow Y \in F, X^+(G) \supseteq Y$

$X^+(G)$ representa X^+ em G

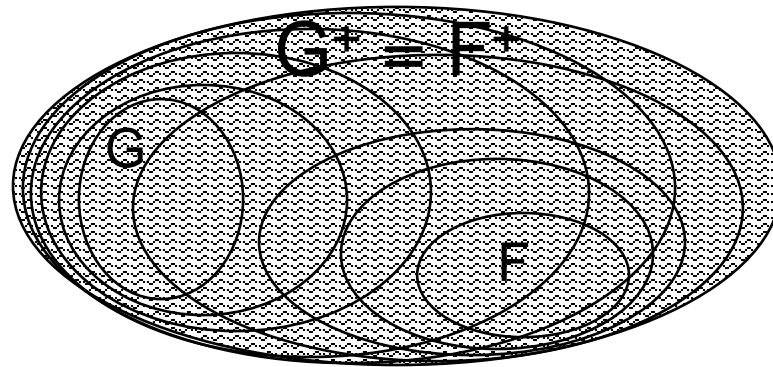
Verificar se G é Cobertura de F - Exemplo

- Seja,
 - $G = \{ A \rightarrow BC, C \rightarrow D \}$
 - $F = \{ AB \rightarrow C, CE \rightarrow D, CA \rightarrow ACD \}$
- G é uma Cobertura de F ?
 - lados esquerdos em F
 - AB, CE, CA
 - usando as Dependências Funcionais de G, temos
 - $AB \rightarrow C$, $AB^+(G) = ABCD$, como $C \in AB^+$ então $AB \rightarrow C \in G^+$
 - $CE \rightarrow D$, $CE^+(G) = CED$ que inclui D (lado direito de CE em F)
 - $CA \rightarrow ACD$, $CA^+(G) = CABD$ que inclui ACD (lado direito de CA em F)
 - $\therefore G$ é uma Cobertura de F
- F é uma Cobertura de G ?
 - $A \rightarrow BC$, $A^+(F) = A$ que não inclui BC (lado direito de A em G)
 - $\therefore G$ não é uma Cobertura de F

$X^+(G)$ representa X^+ em G

Conjuntos de Dependências Funcionais Equivalentes

- Diz-se que G é Equivalente a F (e escreve-se $F \equiv G$) se,
 - o Fecho de G é igual ao Fecho de F ($G^+ = F^+$)



- Um modo sistemático para verificar se G e F são conjuntos de Dependências Funcionais Equivalentes consiste em,
 - ver se G é uma Cobertura de F ($G^+ \supseteq F^+$)
 - ver se F é uma Cobertura de G ($F^+ \supseteq G^+$)
 - se isso acontecer então G e F são Equivalentes ($G^+ = F^+$)

Cobertura Mínima (CM) - Motivação

- F^+ é geralmente demasiado grande para ser tratável
 - para ser tratável quanto menos Dependências Funcionais melhor !
- F^+ tem muitas Dependências Funcionais redundantes
 - para ser tratável quanto menos redundância melhor !
- Partindo de F é sempre possível encontrar um conjunto CM, que é,
 - o menor subconjunto de F ,
 - onde não existe redundância nas dependências Funcionais
 - e a partir do qual se pode obter F^+
- F e CM são Equivalentes
- CM de certeza que não tem Dependências Funcionais redundantes
 - F poderia ter
- CM é o conjunto com que convém lidar !

Cobertura Mínima

- Diz-se que G é Cobertura Mínima de F se,
 - G é uma Cobertura de F e
 - em G , o lado direito de todas as Dependências Funcionais apenas tem um único Atributo (Dependências Funcionais Singulares) e
 - em G o lado esquerdo de todas as Dependências Funcionais não tem nenhum Atributo que seja redundante *
 - retirando qualquer Dependência Funcional de G , este deixa de ser Equivalente a F e
 - Formalmente, G é Cobertura Mínima de F se e só se,
 - $G^+ \supseteq F$
 - $\forall X \rightarrow A \in G$, então A é Singular (ou seja, A é um único Atributo)
 - $\neg \exists X \rightarrow A \in G$, para $Z \subset X$, $\{ \{G - \{X \rightarrow A\}\} \cup \{Z \rightarrow A\} \}^+ = F^+$
 - $\neg \exists X \rightarrow A \in G$, $\{ G - \{X \rightarrow A\} \}^+ = F^+$
- * Em $\{A \rightarrow B, B \rightarrow C, AD \rightarrow C\}$ o atributo D em $AD \rightarrow C$ é redundante

Algoritmo para determinar uma Cobertura Mínima

- Entrada: Conjunto F de Dependências Funcionais
- Saída: Conjunto G Cobertura Mínima de F

- (1) considerar F uma Cobertura dele mesmo e chamar-lhe G
 - $G := F$

- (2) tornar singular cada Dependência Funcional de G
 - substituir cada $X \rightarrow Y_1, Y_2, \dots Y_n$ em G por
 - $X \rightarrow Y_1, X \rightarrow Y_2, \dots X \rightarrow Y_n$

Algoritmo para det. uma Cobertura Mínima (cont.)

■ (3) eliminar Atributos redundantes nos lados esquerdos das DF de G

□ para cada $X \rightarrow A$ em G, X com mais que um atributo

■ considerar $Z := X$

■ para cada elemento $B \in Z$,

□ se, com base em G, tivermos $A \in (Z - B)^+$

□ então fazer $Z := (Z - B)$

□ substituir $X \rightarrow A$ por $Z \rightarrow A$

Exemplo:

$G = \{ D \rightarrow A, A \rightarrow B, AB \rightarrow C, AD \rightarrow C \}$

Consideremos $AB \rightarrow C$:

Consideremos o elemento A: $B^+ = B$, logo $C \notin B^+$, pelo que A não é redundante

Consideremos o elemento B: $A^+ = ABC$, como $C \in A^+$, logo $AB \rightarrow C$ é substituído por $A \rightarrow C$

$G = \{ D \rightarrow A, A \rightarrow B, A \rightarrow C, AD \rightarrow C \}$

Com $AD \rightarrow C$: pode ser substituído por $A \rightarrow C$ ou $D \rightarrow C$

■ (4) encontrar um menor subconjunto G de DF que é equivalente a F

Exemplo:

□ para cada $X \rightarrow A$ em G

■ calcular X^+ a partir de $\{ G - \{X \rightarrow A\} \}$,

■ se $A \in X^+$ então remover $X \rightarrow A$ de G

$G = \{ D \rightarrow A, A \rightarrow B, A \rightarrow C, D \rightarrow C \}$

$z = D \rightarrow A$, $D^+(G - z) = DC$, $D \rightarrow A$ não é redundante

$z = A \rightarrow B$, $A^+(G - z) = AC$, $A \rightarrow B$ não é redundante

$z = A \rightarrow C$, $A^+(G - z) = AB$, $A \rightarrow C$ não é redundante

$z = D \rightarrow C$, $D^+(G - z) = DABC$, remover $D \rightarrow C$

G final = $\{ D \rightarrow A, A \rightarrow B, A \rightarrow C \}$

Determinar uma Cobertura Mínima - Exemplo

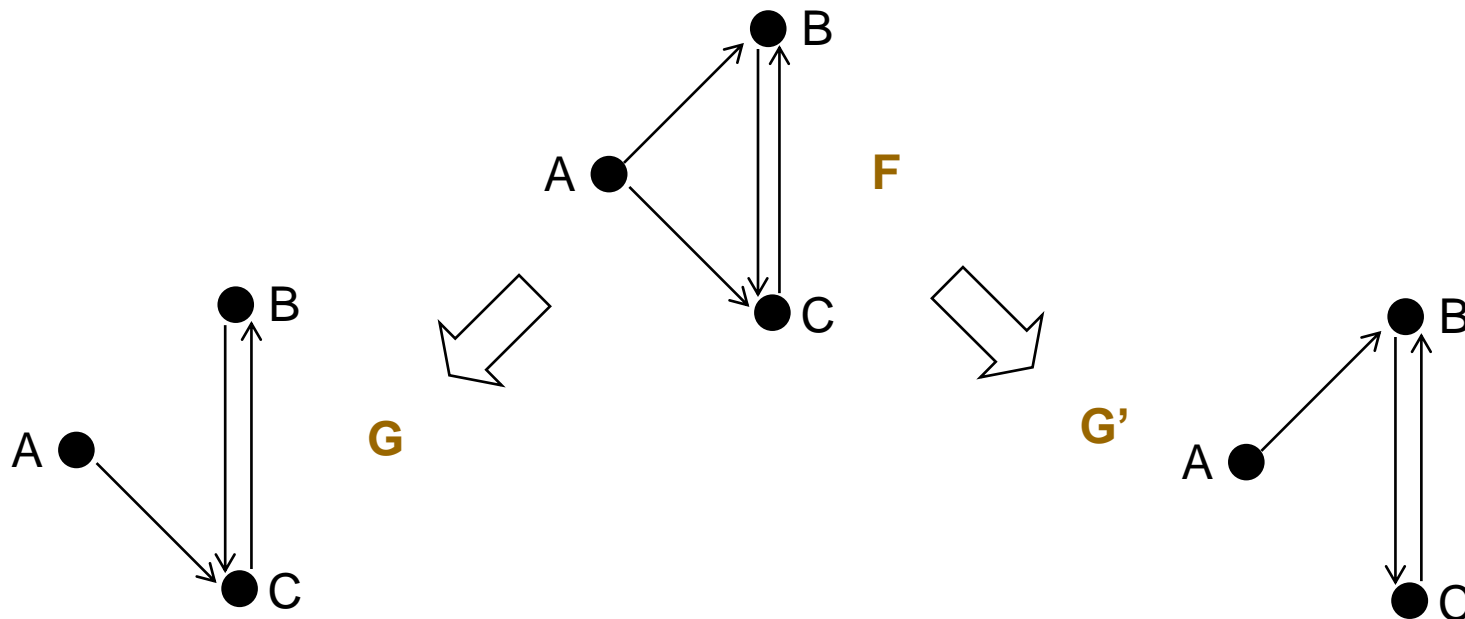
- Cobertura Mínima de $F = \{ D \rightarrow AC, A \rightarrow B, AB \rightarrow C, AD \rightarrow CE \}$?
 - (2), de (1) temos $G = F$, e vamos construir G apenas com DF simples
 - $G = \{ D \rightarrow A, D \rightarrow C, A \rightarrow B, AB \rightarrow C, AD \rightarrow C, AD \rightarrow E \}$
 - (3) eliminar Atributos redundantes nos lados esquerdos: AB, AD
 - $A^+ = ABC$, pelo que $AB \rightarrow C$ e $AD \rightarrow C$ são substituídos por $A \rightarrow C$
 - $D^+ = DACEB$, pelo que $AD \rightarrow E$ é substituído por $D \rightarrow E$
 - (4) temos $G = \{ D \rightarrow A, D \rightarrow C, A \rightarrow B, A \rightarrow C, D \rightarrow E \}$
 - encontrar o menor subconjunto de G
 - $D \rightarrow A, \{ D \rightarrow C, A \rightarrow B, A \rightarrow C, D \rightarrow E \}, D^+ = DCE, A \notin D^+$
 - $D \rightarrow C, \{ D \rightarrow A, A \rightarrow B, A \rightarrow C, D \rightarrow E \}, D^+ = DABCE, C \in D^+$
 - pelo que $D \rightarrow C$ é redundante, sendo portanto eliminado de G
 - $A \rightarrow B, \{ D \rightarrow A, A \rightarrow C, D \rightarrow E \}, A^+ = AC, B \notin A^+$
 - $A \rightarrow C, \{ D \rightarrow A, A \rightarrow B, D \rightarrow E \}, A^+ = AB, C \notin A^+$
 - $D \rightarrow E, \{ D \rightarrow A, A \rightarrow B, A \rightarrow C \}, D^+ = DABC, E \notin D^+$
- $G = \{ D \rightarrow A, A \rightarrow B, A \rightarrow C, D \rightarrow E \}$ é uma Cobertura Mínima de F

Pode existir mais do que uma Cobertura Mínima

- Cobertura Mínima de $F = \{ A \rightarrow BC, B \rightarrow C, C \rightarrow B \}$?
 - (2) temos $G = F$, e vamos construir G apenas com DF simples
 - $G = \{ A \rightarrow B, A \rightarrow C, B \rightarrow C, C \rightarrow B \}$
 - (3) em G os lados esquerdos apenas têm um Atributo
 - (4) temos $G = \{ A \rightarrow B, A \rightarrow C, B \rightarrow C, C \rightarrow B \}$
 - $A \rightarrow B, \{ A \rightarrow C, B \rightarrow C, C \rightarrow B \}, A^+ = ACB, B \in A^+$
 - portanto $A \rightarrow B$ é redundante, sendo eliminado de G
 - $G = \{ A \rightarrow C, B \rightarrow C, C \rightarrow B \}$
 - $A \rightarrow C, \{ B \rightarrow C, C \rightarrow B \}, A^+ = A, C \notin A^+$
 - $B \rightarrow C, \{ A \rightarrow C, C \rightarrow B \}, B^+ = B, C \notin B^+$
 - $C \rightarrow B, \{ A \rightarrow C, B \rightarrow C \}, C^+ = C, B \notin C^+$
 - $\therefore G = \{ A \rightarrow C, B \rightarrow C, C \rightarrow B \}$ é uma Cobertura Mínima de F
- Mas, se em (4), se começar por verificar se $A \rightarrow C$ é redundante, temos:
 - $A \rightarrow C, \{ A \rightarrow B, B \rightarrow C, C \rightarrow B \}, A^+ = ABC, C \in A^+$, portanto $A \rightarrow C$ é redundante
 - $\therefore G' = \{ A \rightarrow B, B \rightarrow C, C \rightarrow B \}$ é outra Cobertura Mínima de F

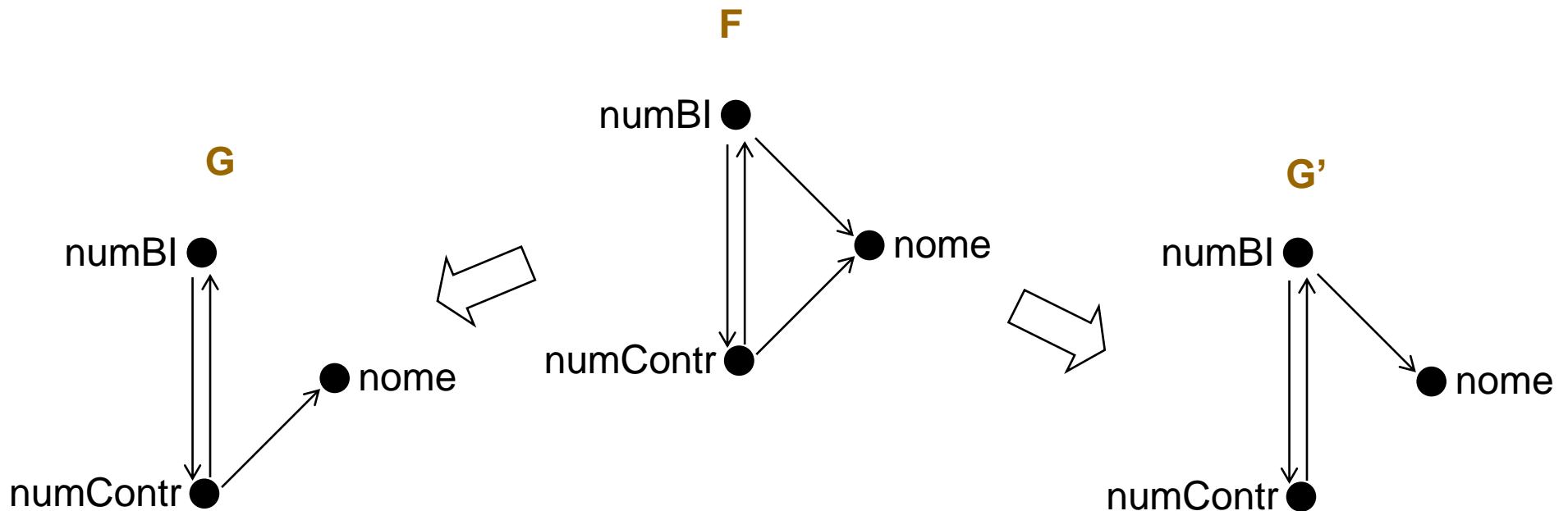
Cobertura Mínima, representação gráfica

- Cobertura Mínima de $F = \{ A \rightarrow BC, B \rightarrow C, C \rightarrow B \}$?
- $G = \{ A \rightarrow C, B \rightarrow C, C \rightarrow B \}$ é uma Cobertura Mínima de F
- $G' = \{ A \rightarrow B, B \rightarrow C, C \rightarrow B \}$ é outra Cobertura Mínima de F
- Representação de F , G e G' através de um grafo:



Cobertura Mínima, outro exemplo

- Outro exemplo de existência de mais do que uma Cobertura Mínima
 - $F = \{ \text{numBl} \rightarrow \{ \text{nome}, \text{numContr} \}, \text{numContr} \rightarrow \{ \text{nome}, \text{numBl} \} \}$?
- Representação de F e das suas Coberturas Mínimas G e G' , através de um grafo:



Consideração

- A “CM - cobertura mínima” de F , é portanto o conjunto mínimo das DFs mais simplificadas, que é equivalente a F
- A utilização desse conjunto facilita as verificações, tais como
 - Se uma DF se verifica em F
 - Se F “é equivalente / contém / é contido” em relação a outro conjunto de DFs

Superchaves, chaves candidatas e chaves primárias

Atributo primo

Superchave, Chave Candidata e Atributo Primo

- Seja $R(A_1, A_2, \dots, A_n)$ e $X \subseteq \{A_1, A_2, \dots, A_n\}$
 - com um conjunto de Dependências Funcionais F
- X é Superchave de R se e só se
 - $X \rightarrow A_1, A_2, \dots, A_n \in F^+$
- Diz-se que X é Chave Candidata de R quando,
 - X é Superchave e
 - X é o conjunto mínimo de Atributos que pode ser Superchave de R
- Formalmente, X é Chave Candidata de R se e só se
 - $X \rightarrow A_1, A_2, \dots, A_n \in F^+$, e
 - $\neg \exists Y \subset X, Y \rightarrow A_1, A_2, \dots, A_n \in F^+$
- Diz-se que um Atributo A é Primo se
 - pertencer a alguma das Chaves Candidatas, ou seja se,
 - $\exists X$ Chave Candidata, $A \in X$

Chave Candidata e Chave Primária

- Qualquer Esquema de Relação $R(A_1, A_2, \dots, A_n)$ com um conjunto de Dependências Funcionais F tem pelo menos uma Chave Candidata
- A Chave Candidata será no limite composta por todos os Atributos,
 - $A_1, A_2, \dots, A_n \rightarrow A_1, A_2, \dots, A_n$
- **Chave Primária é**
 - a Chave Candidata escolhida
- Um modo sistemático para encontrar todas as Chaves Candidatas consiste em considerar todos os subconjuntos $Z \subseteq \{A_1, A_2, \dots, A_n\}$ e
 - escolher os menores subconjuntos que têm
 - $Z^+ = \{A_1, A_2, \dots, A_n\}$

Encontrar as Chaves Candidatas

- Considerando
 - R (numSeq, numFact, ano, numLinha, codCli, codProd, quantidade)
 - F = { {numSeq} → numFact,
 {numSeq} → ano,
 {ano, numFact} → numSeq,
 {ano, numFact, numLinha} → codProd
 {ano, numFact, numLinha} → quantidade
 {ano, numFact} → codCli }
 - encontrar as Chaves Candidatas de R
- Determinar todos os subconjuntos do conjunto de Atributos,
 - é um problema com muitas soluções, e algumas delas podem, à partida, não ser necessárias para encontrar as Chaves Candidatas

Encontrar as Chaves Candidatas (cont.)

- Heurísticas que reduzem a quantidade de subconjuntos a tratar:
 - os Atributos que só aparecem do “lado esquerdo” das DF **pertencem** de certeza à(s) Chave(s) Candidata(s) (nenhum Atributo os determina)
 - os Atributos que não aparecem nem do “lado esquerdo” nem do “lado direito” das DF **pertencem** de certeza à(s) Chave(s) Candidata(s) (nenhum Atributo os determina)
 - os Atributos que só aparecem do “lado direito” das DF **não pertencem** de certeza a nenhuma Chave Candidata (não colaboram para determinar nenhum Atributo)
 - os restantes Atributos **podem pertencer ou não** a alguma(s) Chave(s) Candidata(s) (têm que ser analisados caso a caso) (lado drt. e esq.)
- Notar que, se apenas com os Atributos que têm que pertencer a todas as Chaves Candidatas (os que só estão do “lado esquerdo” mais aqueles que não aparecem nas DF) for possível determinar todos os restantes Atributos, então essa Chave Candidata é única

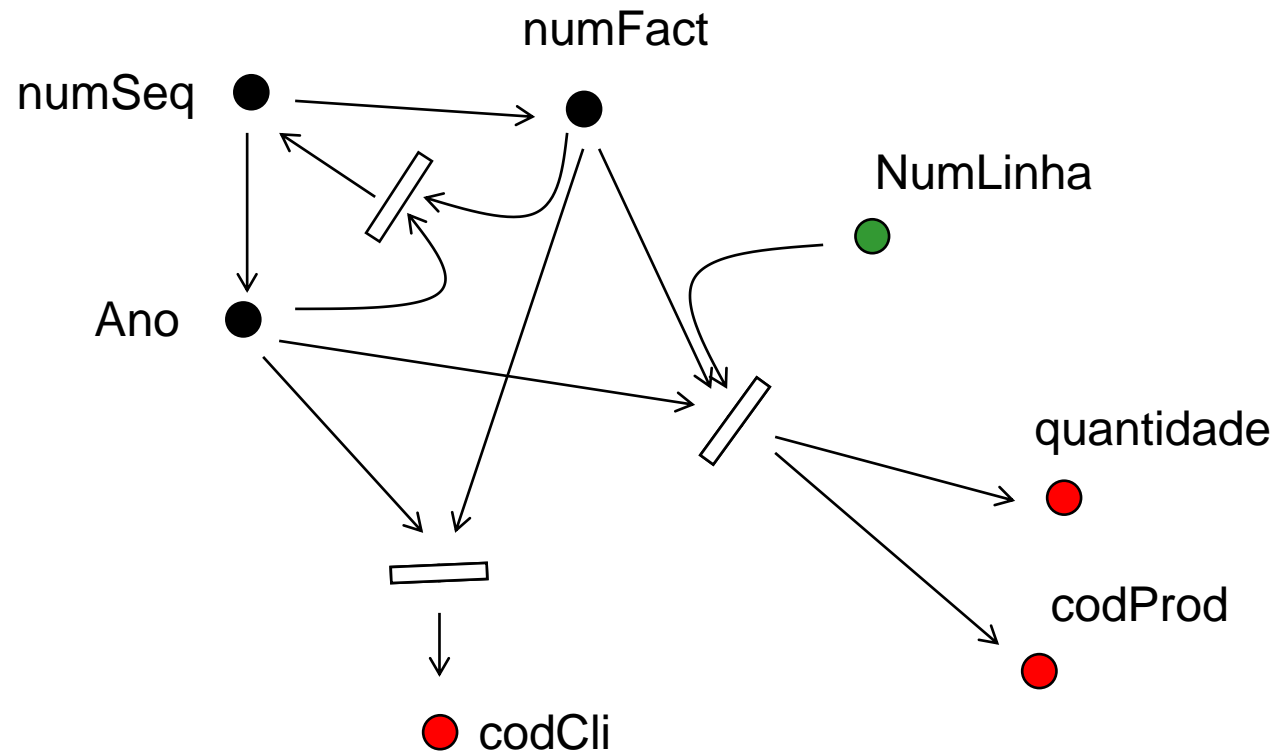
Encontrar as Chaves Candidatas (cont. 1)

- Considerando
 - R (numSeq, numFact, ano, numLinha, codCli, codProd, quantidade)
 - F = { {numSeq} → numFact,
 {numSeq} → ano,
 {ano, numFact} → numSeq,
 {ano, numFact, numLinha} → codProd
 {ano, numFact, numLinha} → quantidade
 {ano, numFact} → codCli }
 - encontrar as Chaves Candidatas de R
- Só aparecem do “lado esquerdo” das DF, ou não aparecem em F
 - E que existirão sempre nas chaves candidatas:
 - {numLinha}
- Aparecem do “lado esquerdo” e do “lado direito” das DF
 - E que poderão ou não fazer parte das chaves candidatas:
 - {numSeq, numFact, ano}

Encontrar as Chaves Candidatas (cont. 2)

- $\{\text{numLinha}\}$ tem que pertencer à(s) Chave(s) Candidata(s)
 - Será que ele sozinho é Chave Candidata ? Se sim será a única Chave
 - mas $\{\text{numLinha}\}^+ = \{\text{numLinha}\} \neq R \therefore$ não é Chave Candidata
- É portanto necessário verificar quais os menores subconjuntos de
 - $\{\text{numLinha}\} \cup \{\text{numSeq}, \text{numFact}, \text{ano}\}$
 - que incluem $\{\text{numLinha}\}$ e cujo Fecho tem todos os Atributos de R
- Como se pretende encontrar os menores subconjuntos pode-se começar por calcular o Fecho dos subconjuntos “mais pequenos”
 - $\{\text{numLinha}, \text{numSeq}\}^+ = \{\text{todos os Atributos}\} \therefore$ é Chave Candidata
 - $\{\text{numLinha}, \text{numFact}\}^+ \neq \{\text{todos os Atributos}\} \therefore$ não é Chave Candidata
 - $\{\text{numLinha}, \text{ano}\}^+ \neq \{\text{todos os Atributos}\} \therefore$ não é Chave Candidata
 - $\{\text{numLinha}, \text{numFact}, \text{ano}\}^+ = \{\text{todos os Atributos}\} \therefore$ é Chave Candidata
- Chaves Candidatas: $\{\text{numLinha}, \text{numSeq}\}$ e $\{\text{numLinha}, \text{numFact}, \text{ano}\}$

Análise gráfica das Chaves candidatas



- **Fazem parte**, só tem arcos de saída, ou sem arcos
- **Não fazem parte**, só têm arcos de entrada
- **Talvez façam parte**, têm arcos de entrada e saída

Encontrar as chaves candidatas por análise do grafo das dependências

- Pelo gráfico pode-se visualizar o fecho de um conjunto de atributos fazendo:
 - ❑ cada atributo do conjunto inicial é marcado como identificado;
 - ❑ cada nó atributo identificado propaga a identificação a cada arco de saída;
 - ❑ cada nó de composição necessita de ter todos os arcos de entrada com identificação activada e nesse caso propaga a identificação pelos arcos de saída
 - ❑ No final teremos todos os atributos do fecho de conjunto inicial marcados como identificados
- Para encontrar as chaves candidatas graficamente, temos de encontrar quais os menores conjuntos que têm um fecho igual ao conjunto de atributos da relação

Encontrar as chaves candidatas por análise do grafo das dependências - procedimento

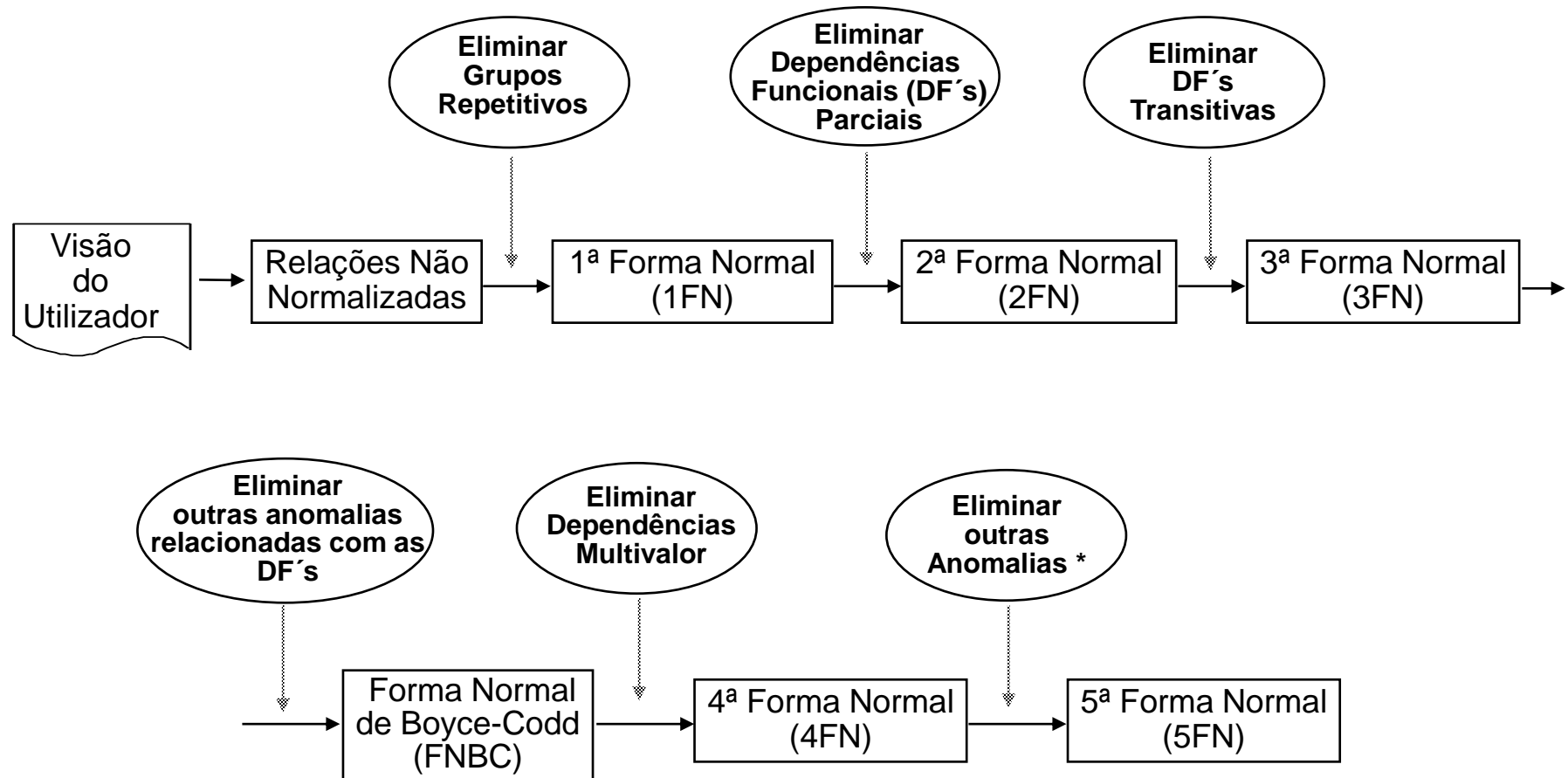
- Considerando apenas os atributos que são obrigatórios nas chaves, verificar se são uma chave. Se sim, são chave única e terminar.
- Se não, verificar quais os conjuntos, formados pelo conjunto de atributos obrigatórios em união com um dos conjuntos de cardinalidade 1 dos atributos possíveis, que são chaves candidatas.
 - Dos que forem chaves, retirar os atributos possíveis considerados do conjunto dos atributos possíveis.
- Repetir aumentando a cardinalidade
 - até esgotar as possibilidades de sucesso

Normalização

Objectivo da Normalização

- O projecto de uma Base de Dados passa tipicamente por,
 - construção do Modelo Entidade-Associação (Modelo Conceptual dos Dados), e
 - sua transformação para o Esquema Relacional (Modelo Lógico) equivalente.
- O Esquema Relacional obtido representa a estrutura da informação de um modo natural e completo.
- A questão que se coloca é a de saber se esse,
 - Esquema Relacional tem o mínimo de redundância possível ?
- O objectivo da Normalização é o de,
 - avaliar a qualidade do Esquema Relacional e
 - transformar (em caso de necessidade) num Esquema Relacional equivalente, menos redundante e mais estável

Etapas na Normalização



* Dependências de Junção

Exemplo

- Numa escola, pretende-se manter informação sobre:
 - os estudantes da escola
 - nº do estudante, nome do estudante e curso a que pertence
 - as disciplinas que são ministradas na escola
 - nº da disciplina e nome da disciplina, e quem é o seu único professor
 - os professores contratados pela escola
 - código do professor, nome do professor e grau académico
 - as notas obtidas pelos alunos nas disciplinas que frequentam
- Para este exemplo, vamos começar por construir um Esquema de Relação **com toda a informação** que se pretende registar

Esquema de Relação com todos os Atributos

■ Esquema de Relação ESCOLA

<u>Nº</u> Estudante	Nome Estudante	Curso	Nº Disciplina	Nome Disciplina	Cód. Professor	Nome Professor	Grau Professor	Nota
36578	João	Informática	003, 104, 550	Estatística, Compiladores, Inglês	256, 777, 891	Gil Pascoal Steve	Prof. Agreg. Assistente Prof. Conv.	15 12 16
17738	Miguel	Gestão	100, 111, 003, 550	Economia, Cálc. Financeiro, Estatística, Inglês	256 444, 256, 891	Gil, Margarida Gil Steve	Prof. Agreg. Prof. Conv. Prof. Agreg. Prof. Conv.	13 17 11 17
24451	Pedro	Biologia	300, 003,	Química, Estatística,	333, 256,	Joana Gil	Prof. Agreg. Prof. Agreg.	14 12

1ª Forma Normal

- Um Esquema de Relação está na 1ª Forma Normal (1FN) quando:
 - os Domínios dos seus atributos contêm apenas valores atômicos.
- No modelo Relacional todos os Esquemas de Relação têm que verificar a 1FN, ou seja,
 - é “proibido” ter Esquemas de Relação com “relações dentro de relações”
- No exemplo anterior, a tabela ESCOLA não está na 1FN porque,
 - cada Estudante tem várias Disciplinas, com toda a informação correspondente
 - geram-se assim Atributos com valores não atômicos - todos os valores de cada um desses Atributos descrevem a mesma característica
 - os Atributos com valores não atômicos são:
 - N°Disciplina, NomeDisciplina, Cod.Professor, NomeProfessor, GrauProfessor, Nota.

1ª Forma Normal (cont. 1)

- Os próximos Esquemas de Relação também não estão na 1FN:

RESTRIÇÃO_ACESSO

<u>Nome</u> <u>Utilizador</u>	<u>Cód</u> <u>Aplicação</u>	Restrição de Acesso às Opções de Menu
pmsilva	145	FileNew, InsertItem
mrcosta	145	FileNew, FileOpen, FileClose, InsertItem
jpmartins	775	FileNew

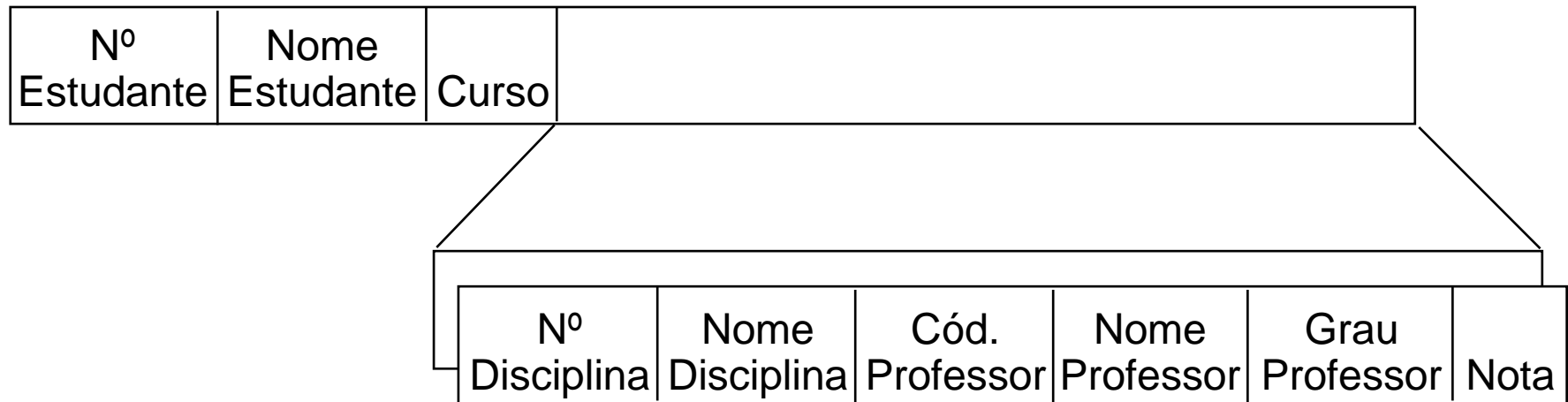
- O domínio do Atributo “Restrição de Acesso às Opções de Menu” contém valores não atômicos.

LIVRO

Autor	Título	Nº Páginas
Gabriel Garcia Marquez	Cem Anos de Solidão, O Outono do	328, 212
Milorad Pavic	Dicionário Khazar	258

- O domínio dos Atributos “Título” e “Nº Páginas” contém valores não atômicos.

1ª Forma Normal (cont. 2)

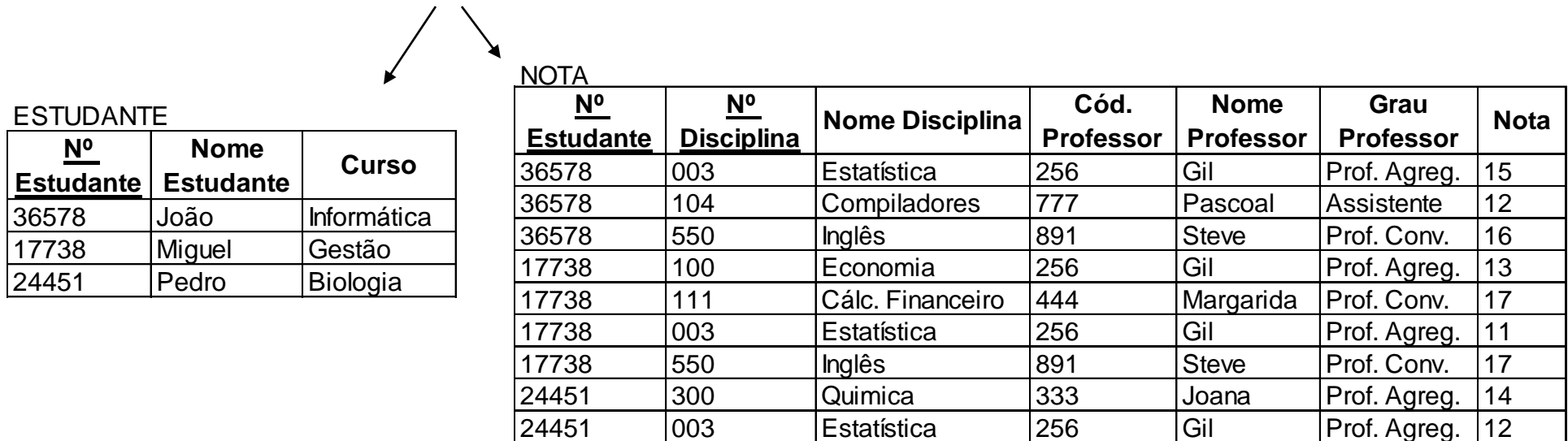


- Para que o Esquema de Relação ESCOLA passe à 1FN tem de ser dividido nos seguintes Esquemas de Relação:
 - **ESTUDANTE** (NºEstudante, NomeEstudante, Curso)
 - Chave Candidata: NºEstudante
 - **NOTA** (NºEstudante, NºDisciplina, NomeDisciplina, CodProfessor, NomeProfessor, GrauProfessor, Nota)
 - Chave Candidata: NºEstudante, NºDisciplina

Outra solução seria replicar a informação de aluno. Mas nesse caso a chave teria de ser Nº Estudante e Nº Disciplina

1ª Forma Normal (cont. 3)

<u>Nº</u> <u>Estudante</u>	Nome Estudante	Curso	Nº Disciplina	Nome Disciplina	Cód. Professor	Nome Professor	Grau Professor	Nota
36578	João	Informática	003, 104, 550	Estatística, Compiladores, Inglês	256, 777, 891	Gil, Pascoal, Steve	Prof. Agreg. Assistente Prof. Conv.	15 12 16
17738	Miguel	Gestão	100, 111, 003, 550	Economia, Cálc. Financeiro, Estatística, Inglês	256 444, 256, 891	Gil, Margarida, Gil, Steve	Prof. Agreg. Prof. Conv. Prof. Agreg. Prof. Conv.	13 17 11 17
24451	Pedro	Biologia	300, 003,	Química, Estatística,	333, 256,	Joana, Gil	Prof. Agreg. Prof. Agreg.	14 12



Inconvenientes da 1FN

- Anomalia de Inserção de um novo tuplo (linha)
 - Se for pretendido inserir informação sobre uma nova disciplina, por exemplo:
 - 035 - Planeamento Estratégico, dada pelo professor 087 - Esteves
 - Não é possível inserir os dados relativos à nova disciplina enquanto não existirem alunos inscritos para essa nova disciplina (o atributo N^oEstudante faz parte da Chave do Esquema de Relação).
- Anomalia de Remoção de um tuplo (linha) já existente
 - Se for pretendido apagar a informação sobre todos os alunos que têm uma determinada disciplina, então perde-se toda a informação dessa disciplina e do respectivo professor.
 - Eliminação de informação que não se pretendia eliminar.

Inconvenientes da 1FN (cont.)

- Anomalia de Actualização de um tuplo (linha) já existente
 - Se for pretendido modificar o nome de uma disciplina, por exemplo,
 - “Inglês” passa a “Inglês Técnico”
 - É necessário percorrer todas as linhas da tabela e fazer essa modificação para cada um dos alunos que tenha essa disciplina.
- No caso da aplicação que realiza a actualização pretendida (de “Inglês” para “Inglês Técnico”) falhar em alguma das linhas da tabela esta poderá ficar com dados inconsistentes.
- Estes inconvenientes que a 1FN exhibe, deverão ser resolvidos através da passagem dos dados para a 2FN e 3FN.
- A 2FN utiliza o conceito de “Dependência Funcional Total”

Dependência Funcional Total e Parcial

- Uma Dependência Funcional $X \rightarrow Y$ diz-se **total** de um conjunto de atributos Z , se $X = Z$
- Uma Dependência Funcional $X \rightarrow Y$ diz-se **parcial** de um conjunto de atributos Z , se $X \not\subset Z$ e $\exists A_i \in X, A_i \in Z$
 - Ou seja, X contém parte de Z

- Considere-se os seguinte dados:

- $R = \{\text{nomeForn}, \text{prod}, \text{preco}, \text{endereço}\}$ e
- $F = \{ \{\text{nomeForn}, \text{prod}\} \rightarrow \{\text{preco}\}, \{\text{nomeForn}\} \rightarrow \{\text{endereço}\} \}$

Vamos modelar esta informação num só esquema de relação, que terá como chave primária a chave $\{\text{nomeForn}, \text{prod}\}$, que é a única chave candidata:

- $\text{FORNEC_PROD}(\underline{\text{nomeForn}}, \underline{\text{prod}}, \text{preco}, \text{endereço}),$

Neste esqRel existem portanto as seguintes DFs: $\{ \{\mathbf{\text{nomeForn}}, \mathbf{\text{prod}}\} \rightarrow \{\mathbf{\text{preco}}\}, \{\mathbf{\text{nomeForn}}, \mathbf{\text{prod}}\} \rightarrow \{\mathbf{\text{endereço}}\}, \{\text{nomeForn}\} \rightarrow \{\text{endereço}\} \}$

Onde as DFs a negrito resultam da existência da chave primária

Verifica-se que $\{\text{nomeForn}\} \rightarrow \{\text{endereço}\}$ é uma Dependência Funcional Parcial da chave primária

2ª Forma Normal

- Um Esquema de Relação está na 2ª Forma Normal (2FN) quando
 - está na 1FN e
 - os atributos não primos devem depender da totalidade de uma ou mais chaves candidatas (cada DF deve conter, no seu lado esquerdo, apenas uma chave candidata)
- Formalmente, seja $R(A_1, A_2, \dots, A_n)$, R está na 2FN se,
 - R está na 1FN e
 - $\forall X$ Chave Candidata e $\forall A_i \notin \text{Chave Candidata}, X \rightarrow A_i$ é Total

Atributo não primo – atributo que não pertence a qualquer chave candidata

Exemplo da existência de Dependências Parciais numa relação mal construída

NOTA

<u>Nº Estudante</u>	<u>Nº Disciplina</u>	Nome Disciplina	Cód. Professor	Nome Professor	Grau Professor	Nota
36578	003	Estatística	256	Gil	Prof. Agreg.	15
36578	104	Compiladores	777	Pascoal	Assistente	12
36578	550	Inglês	891	Steve	Prof. Conv.	16
17738	100	Economia	256	Gil	Prof. Agreg.	13
17738	111	Cálc. Financeiro	444	Margarida	Prof. Conv.	17
17738	003	Estatística	256	Gil	Prof. Agreg.	11
17738	550	Inglês	891	Steve	Prof. Conv.	17
24451	300	Química	333	Joana	Prof. Agreg.	14
24451	003	Estatística	256	Gil	Prof. Agreg.	12

DFs derivadas da chave primária:

- {NºEstudante, NºDisciplina} → NomeDisciplina
- {NºEstudante, NºDisciplina} → codProfessor
- {NºEstudante, NºDisciplina} → NomeProfessor
- {NºEstudante, NºDisciplina} → GrauProfessor
- {NºEstudante, NºDisciplina} → Nota

Contudo existem as seguintes DFs:

- {NºDisciplina} → NomeDisciplina
- {NºDisciplina} → codProfessor
- {NºDisciplina} → NomeProfessor
- {NºDisciplina} → GrauProfessor

Além de:

- {codProfessor} → NomeProfessor
- {codProfessor} → GrauProfessor

As DFs a negrito são DFs Parciais de chaves candidatas

Resultando que o esquema de relação não está na 2FN.

Desdobrar para passar à 2FN

NOTA

<u>Nº</u> <u>Estudante</u>	<u>Nº</u> <u>Disciplina</u>	Nome Disciplina	Cód. Professor	Nome Professor	Grau Professor	Nota
36578	003	Estatística	256	Gil	Prof. Agreg.	15
36578	104	Compiladores	777	Pascoal	Assistente	12
36578	550	Inglês	891	Steve	Prof. Conv.	16
17738	100	Economia	256	Gil	Prof. Agreg.	13
17738	111	Cálc. Financeiro	444	Margarida	Prof. Conv.	17
17738	003	Estatística	256	Gil	Prof. Agreg.	11
17738	550	Inglês	891	Steve	Prof. Conv.	17
24451	300	Química	333	Joana	Prof. Agreg.	14
24451	003	Estatística	256	Gil	Prof. Agreg.	12

Chave e atributos que dependem da totalidade da chave.

Atributos que dependem de parte da chave, mais a respectiva parte da chave.

NOTA

<u>Nº</u> <u>Estudante</u>	<u>Nº</u> <u>Disciplina</u>	Nota
36578	003	15
36578	104	12
36578	550	16
17738	100	13
17738	111	17
17738	003	11
17738	550	17
24451	300	14
24451	003	12

DISCIPLINA

<u>Nº</u> <u>Disciplina</u>	Nome Disciplina	Cód. Professor	Nome Professor	Grau Professor
003	Estatística	256	Gil	Prof. Agreg.
104	Compiladores	777	Pascoal	Assistente
550	Inglês	891	Steve	Prof. Conv.
100	Economia	256	Gil	Prof. Agreg.
111	Cálc. Financeiro	444	Margarida	Prof. Conv.
300	Química	333	Joana	Prof. Agreg.

Inconvenientes da 2FN

- Anomalia de Inserção de um novo tuplo (linha)
 - se for pretendido inserir informação sobre um novo professor,
 - 087 - Esteves, com grau de Assistente Estagiário
 - não é possível inserir os dados relativos ao novo professor enquanto não lhe for atribuída pelo menos uma disciplina (a chave é N^o Disciplina)
- Anomalia de Remoção de um tuplo (linha) já existente
 - se for pretendido apagar a informação sobre determinada disciplina que é dada por um professor que apenas dá essa disciplina, então perde-se toda a informação relativa a esse professor
 - eliminação de informação que não se pretendia eliminar.

Inconvenientes da 2FN (cont.)

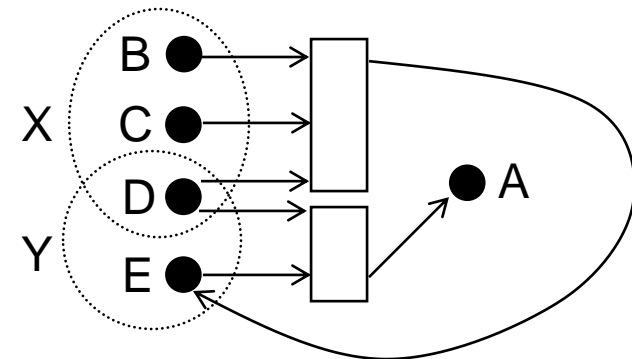
- Anomalia de Actualização de um tuplo (linha) já existente
 - se for pretendido modificar o grau de um professor,
 - “Assistente” para “Professor Convidado”
 - é necessário percorrer todas as linhas da tabela e fazer essa modificação para cada uma das disciplina que esse professor ensina
- No caso da aplicação que realiza a actualização pretendida (de “Assistente” para “Professor Convidado”) falhar em alguma das linhas da tabela esta poderá ficar com dados inconsistentes.
- Estes inconvenientes que a 2FN exhibe, deverão ser resolvidos através da passagem dos dados para a 3FN
- A 3FN utiliza o conceito de “Dependência Funcional Transitiva”

Dependência Funcional Transitiva

- Seja $R(A_1, A_2, \dots, A_n)$, $X \subseteq \{A_1, A_2, \dots, A_n\}$ e A um Atributo de R
- Diz-se que $X \rightarrow A$ é uma Dependência Funcional Transitiva sse,
 - $\exists Y \subseteq \{A_1, A_2, \dots, A_n\}$ tal que $X \rightarrow Y$, $Y \rightarrow A$, $Y \not\rightarrow X$, $A \notin X$, $Y \not\subseteq X$

- Exemplo

- $F = \{BCD \rightarrow E, DE \rightarrow A\}$
 - $BCD \rightarrow A$ é uma DF Transitiva ?



- $BCD \rightarrow D$ (Reflexividade), pelo que $BCD \rightarrow DE$ (União)
- temos assim: $BCD \rightarrow DE$ e $DE \rightarrow A$
- concluindo-se que $BCD \rightarrow A$ é uma DF Transitiva

3ª Forma Normal

- Um Esquema de Relação está na 3ª Forma Normal (3FN) quando:
 - ❑ não existirem Dependências Funcionais Transitivas de Atributos Primos para Atributos não-Primos*
 - ❑ Cada atributo não-primário for directamente dependente de uma das chaves candidatas *
 - ❑ Cada atributo não-primário “providenciar um facto acerca de uma chave candidata, de toda a chave e nada mais do que a chave” *
- A Dependência Funcional Parcial é um caso particular da Dependência Funcional Transitiva
 - ❑ Se X for uma chave, Y um subconjunto de X ($Y \subset X$), A um atributo não primário,
 - ❑ então $X \rightarrow A$, que provém do facto de X ser chave
 - ❑ Se existir a DF parcial $Y \rightarrow A$, então temos uma DF transitiva, porque $X \rightarrow Y$ e $Y \rightarrow A$
- Assim, obrigando a que não existam Dependências Funcionais Transitivas de Atributos Primos para Atributos não-Primos, também não existirão Dependências Funcionais Parciais de Atributos Primos para Atributos não-Primos
- Um Esquema de Relação na 3FN está também na 2FN

* definições alternativas

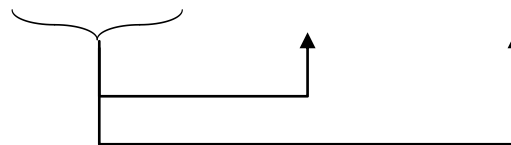
3ª Forma Normal (cont.)

- Outra forma de apresentar a definição de 3FN - Um Esquema de Relação está na 3ª Forma Normal (3FN) se e só se:
 - sempre que $X \rightarrow A$ se verifica e $A \notin X$, tem-se que,
 - X é uma chave, ou
 - A é um Atributo Primo
- Exemplo de um Esquema de Relação que não está na 3FN:
 - PROD_LOCAL (codProd, regioao, pais) com {regiao \rightarrow pais}
 - {regiao} não é Superchave e {pais} não é Atributo Primo
 - {codProd \rightarrow pais} é uma Dependência Funcional Transitiva
- PROD_LOCAL pode ser decomposto em dois na 3FN
 - LOCAL (codProd, regioao)
 - PAIS (regiao , pais)

Dependências Funcionais Transitivas

DISCIPLINA

<u>Nº Disciplina</u>	Nome Disciplina	Cód. Professor	Nome Professor	Grau Professor
003	Estatística	256	Gil	Prof. Agreg.
104	Compiladores	777	Pascoal	Assistente
550	Inglês	891	Steve	Prof. Conv.
100	Economia	555	Silva	Assistente
111	Cálc. Financeiro	444	Margarida	Prof. Conv.



- Os atributos **Nome Professor** e **Grau Professor**, dependem funcionalmente do atributo **Cód. Professor**, que não é chave da relação e portanto:
 - $N^o \text{ Disciplina} \rightarrow \text{Nome Professor}$,
 - $N^o \text{ Disciplina} \rightarrow \text{Grau Professor}$,
 - Podem ser obtidas por aplicação da Regra da Transitividade (são Dependências Funcionais Transitivas de atributos primos para atributos não-primos)

Desdobrar para passar à 3FN

DISCIPLINA

<u>Nº Disciplina</u>	Nome Disciplina	Cód. Professor	Nome Professor	Grau Professor
003	Estatística	256	Gil	Prof. Agreg.
104	Compiladores	777	Pascoal	Assistente
550	Inglês	891	Steve	Prof. Conv.
100	Economia	256	Gil	Prof. Agreg.
111	Cálc. Financeiro	444	Margarida	Prof. Conv.
300	Química	333	Joana	Prof. Agreg.

Atributos que apenas
dependem da chave



DISCIPLINA

<u>Nº Disciplina</u>	Nome Disciplina	Cód. Professor
003	Estatística	256
104	Compiladores	777
550	Inglês	891
100	Economia	256
111	Cálc. Financeiro	444
300	Química	333

Atributos que dependem
do atributo não chave,
mais esse atributo

PROFESSOR

<u>Cód. Professor</u>	Nome Professor	Grau Professor
256	Gil	Prof. Agreg.
777	Pascoal	Assistente
891	Steve	Prof. Conv.
444	Margarida	Prof. Conv.
333	Joana	Prof. Agreg.

Método intuitivo de Decomposição

- Considerando $R(A, B, C, D, E)$ e $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E\}$,
 - construir um Esquema Relacional na 3FN
- A Chave Candidata é $\{A, B\}$
 - $(AB)^+ = ABCDE$
- Não existem Dependências Funcionais Parciais de qualquer Atributo em relação à Chave Candidata, logo R já está na 2FN
- Mas existem dependências transitivas de atributos primos para atributos não-primos: $AB \rightarrow D$ e $AB \rightarrow E$, logo não está na 3FN
- Uma possível abordagem consiste em colocar a parte mais direita das DFs transitivas num esquema de relação próprio

Decomposição de $R(A, B, C, D, E)$ em:

- $R_1(\underline{A}, \underline{B}, C)$ com $F = \{AB \rightarrow C\}$
- $R_2(\underline{C}, D, E)$ com $F = \{C \rightarrow D, D \rightarrow E\}$ mas este esquema não está na 3FN

Nova decomposição, agora com todos os esquemas na 3FN:

- $R_1(\underline{A}, \underline{B}, C)$ com $F = \{AB \rightarrow C\}$
- $R_2(\underline{C}, D)$ com $F = \{C \rightarrow D\}$
- $R_3(\underline{D}, E)$ com $F = \{D \rightarrow E\}$

Método intuitivo de Decomposição

- Considerando novamente:
 - $R(A, B, C, D, E)$ e $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E\}$,
 - Com $(AB)^+ = ABCDE$
- Uma possível Decomposição de $R(A, B, C, D, E)$ na 3FN será,
 - $R1(\underline{A}, B, C)$
 - $R2(\underline{D}, E)$
- Mas será que com $R1$ e $R2$ ainda se têm todo o F ?
 - Não.
- Esta Decomposição está na 3FN mas perdeu-se a $C \rightarrow D$!

Preservação de Dependências - Motivação

- É importante que cada $X \rightarrow Y$ existente em F esteja
 - num dos Esquemas de Relação R_i resultante da Decomposição, ou
 - possa ser inferido a partir das DF que aparecem em algum dos R_i
- Se isso acontecer existe “Preservação de Dependências”
- É importante existir “Preservação de Dependências”,
 - pois cada DF representa uma restrição que os dados devem respeitar

Projecção de DF num Esquema de Relação

- Sejam F as DF de R e uma Decomposição $D = \{R_1, R_2, \dots, R_n\}$ de R
- A Projecção de F em R_i escreve-se $\pi_{R_i}(F)$ e consiste em,
 - $X \rightarrow Y \in F^+$ e $X \cup Y \subseteq R_i$
- Seja $R(A, B, C, D, E)$, com $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E\}$ e a Decomposição $D = \{R1(\underline{A}, \underline{B}, D), R2(\underline{C}, E)\}$
 - $AB \rightarrow D$ pertence a $\pi_{R1}(F)$?
 - $\{A, B\} \cup \{D\} \subseteq \{A, B, D\}$ e $AB \rightarrow D \in F^+ \therefore AB \rightarrow D$ pertence à $\pi_{R1}(F)$
 - $C \rightarrow D$ pertence a $\pi_{R1}(F)$ ou $\pi_{R2}(F)$?
 - $\{C\} \cup \{D\} \not\subseteq \{A, B, D\} \therefore C \rightarrow D$ não pertence à $\pi_{R1}(F)$
 - $\{C\} \cup \{D\} \not\subseteq \{C, E\} \therefore C \rightarrow D$ não pertence à $\pi_{R2}(F)$
 - $D \rightarrow E$ pertence a $\pi_{R1}(F)$ ou $\pi_{R2}(F)$?
 - $\{D\} \cup \{E\} \not\subseteq \{A, B, D\} \therefore D \rightarrow E$ não pertence à $\pi_{R1}(F)$
 - $\{D\} \cup \{E\} \not\subseteq \{C, E\} \therefore D \rightarrow E$ não pertence à $\pi_{R2}(F)$
 - $C \rightarrow E$ pertence a $\pi_{R2}(F)$?
 - $\{C\} \cup \{E\} \subseteq \{C, E\}$ e $C \rightarrow E \in F^+ \therefore C \rightarrow E$ pertence à $\pi_{R2}(F)$

Preservação de Dependências

- Uma Decomposição $D = \{R_1, R_2, \dots, R_n\}$ de R diz-se que “Preserva Dependências” relativamente a um conjunto de Dependências Funcionais F em R , se
 - a união das Projecções de F em cada R_i for equivalente a F , ou seja,
 - $(\pi_{R_1}(F) \cup \dots \cup \pi_{R_n}(F))^+ = F^+$
- Seja $R(A, B, C, D, E)$ e $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E\}$,
 - a Decomposição $D = \{R1(\underline{A}, \underline{B}, D), R2(\underline{C}, E)\}$ preserva Dependências ?
- Analisando D temos,
 - $\pi_{R_1}(F) = \{AB \rightarrow D\}^+$
 - $\pi_{R_2}(F) = \{C \rightarrow E\}^+$
 - $C \rightarrow D \notin (\pi_{R_1}(F) \cup \pi_{R_2}(F))^+, D \rightarrow E \notin (\pi_{R_1}(F) \cup \pi_{R_2}(F))^+$
 - Como $C \rightarrow D \in F$ e $D \rightarrow E \in F$
 - \therefore a Decomposição $R1, R2$ não Preserva Dependências

Sem Perda por Junção (*Lossless Join*)

- É importante assegurar que a informação obtida através da Junção Natural dos Esquemas de Relação da Decomposição não é diferente da informação no Esquema de Relação inicial
 - se isso acontecer diz-se que a Decomposição “Preserva Informação”
- Uma Decomposição “Preserva Informação” se tiver a propriedade “Sem Perda por Junção” (do inglês *Lossless Join*)
- Uma Decomposição $D = \{R_1, R_2, \dots, R_n\}$ de R diz-se *Lossless Join* se
 - para toda a instância r de R tivermos,
 - $r = (\pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_n}(r))$

Sem Perda por Junção (*Lossless Join*) - Exemplo

- Sendo $R(\text{numBI}, \text{nome}, \text{numMatriculaAutomovel})$ e
 - $F = \{\text{numBI} \rightarrow \text{nome}, \text{numMatriculaAutomovel} \rightarrow \text{nome}\}$
 - a Decomposição $R1(\text{numBI}, \text{nome}), R2(\text{numMatriculaAutomovel}, \text{nome})$ Preserva Informação ?

- Considerando

R

numBI	numMatriculaAutomovel	nome
1111111	33 - 24 - RH	Miguel Sousa
2222222	33 - 24 - RH	Miguel Sousa
1111111	77 - 77 - XX	Miguel Sousa

- Tem-se

R1

numBI	nome
1111111	Miguel Sousa
2222222	Miguel Sousa

R2

numMatriculaAutomovel	nome
33 - 24 - RH	Miguel Sousa
77 - 77 - XX	Miguel Sousa

- e $R1 \bowtie R2 \neq R$

numBI	numMatriculaAutomovel	nome
1111111	33 - 24 - RH	Miguel Sousa
2222222	33 - 24 - RH	Miguel Sousa
1111111	77 - 77 - XX	Miguel Sousa
2222222	77 - 77 - XX	Miguel Sousa

Algoritmo para verificar a propriedade *Lossless Join*

- Entrada: F , $R(A_1, \dots, A_n)$ e uma Decomposição $D = \{R_1, \dots, R_k\}$
- Saída: Verificação, ou não, da propriedade *Lossless Join*
- (1) Construir matriz $M(i, j)$ com
 - linhas i para cada R_i e colunas j para cada A_j , tal que
 - se $A_j \in R_i$, $M(i, j) = a_j$
 - se $A_j \notin R_i$, $M(i, j) = b_{ij}$
- (2) Para cada $X \rightarrow Y$ em F ,
 - procurar linhas com o mesmo valor em todas as colunas de X e,
 - para essas linhas igualar os valores de Y do seguinte modo,
 - se um dos valores for a_j , igualar os restantes a a_j
 - se os valores forem apenas b_{ij} , igualar a um qualquer dos b_{ij}
- (3) Se existir uma linha i apenas com valores a_j , conclui-se que a Decomposição é *Lossless Join*

Verificar a propriedade *Lossless Join* - Exemplo

■ Exemplo,

- considerando $R(A, B, C, D, E, F)$ e $G = \{A \rightarrow B, C \rightarrow DF, AC \rightarrow E\}$ e
- a Decomposição $R1(A,B), R2(C,D,F), R3(A,C,E)$ preserva Informação ?

■ (1) Construir matriz $M(i, j)$

	A	B	C	D	E	F
R1(A,B)	a1	a2	b13	b14	b15	b16
R2(C,D,F)	b21	b22	a3	a4	b25	a6
R3(A,C,E)	a1	b32	a3	b34	a5	b36

■ (2)

- de $A \rightarrow B$ resulta

	A	B	C	D	E	F
R1(A,B)	a1	a2	b13	b14	b15	b16
R2(C,D,F)	b21	b22	a3	a4	b25	a6
R3(A,C,E)	a1	a2	a3	b34	a5	b36

- de $C \rightarrow DF$ resulta

	A	B	C	D	E	F
R1(A,B)	a1	a2	b13	b14	b15	b16
R2(C,D,F)	b21	b22	a3	a4	b25	a6
R3(A,C,E)	a1	a2	a3	a4	a5	a6

- Decomposição é *Lossless Join* – a última linha da matriz é toda a_i

Decomposição na 3FN - Algoritmo

- Entrada: $R(A_1, \dots, A_n)$ e F um conjunto de Dependências Funcionais
- Saída: Decomposição $D = \{R_1, R_2, \dots, R_n\}$ de R na 3FN que
 - Preserva Dependências Funcionais e
 - Preserva Informação (é *Lossless Join*)
- Este Algoritmo tem três passos:
 - (1) Partir do menor conjunto de Dependências Funcionais G , que é equivalente a F , ou seja, a sua cobertura mínima
 - (2) A partir de G , construir uma Decomposição D que Preserva as Dependências Funcionais existentes em F
 - (3) Garantir que a Decomposição D é *Lossless Join*

Decomposição na 3FN - Algoritmo (cont.)

- (1) Encontrar uma Cobertura Mínima G de F , $D = \{ \}$

Construir uma Decomposição D que preserve as Dependências Funcionais existentes em F

- (2) Para cada Dependência Funcional $X \rightarrow A$ de G
 - se não existe em D um Esquema de Relação que contenha X e A ,
 - então $i = \text{cardinal}(D)+1$ e $D = D \cup R_i (X, A)$

Garantir que a Decomposição D é *Lossless Join*

- (3) Para todas as chaves candidatas de R
 - Para cada chave candidata C , se não existir em D pelo menos um Esquema de Relação que a inclua:
 - então $i = \text{cardinal}(D)+1$ e $D = D \cup R_i (C)$

Algoritmo para 3FN - Exemplo 1

Algoritmo de cobertura mínima G

- 1) $G = F$
- 2) Eliminar partes direita não singulares
- 3) Eliminar redundância da parte esquerda
- 4) Eliminar dependências redundantes

- Sendo $R(A, B, C, D, E)$ e
 - $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E, CD \rightarrow E\}$,
 - aplicar o algoritmo para encontrar uma **Decomposição na 3FN**

- (1) Cobertura Mínima G de F
 - $G = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E\}$

- (2) Decomposição que Preserva as Dependências Funcionais de F
 - $\text{Decomp} = \{ R1(A, B, C), R2(C,D), R3(D,E) \}$

- (3) Garantir que a Decomposição Decomp é *Lossless Join*
 - Determinar chaves candidatas de R e G
 - $AB^+ = ABCDE$, portanto AB é a única Chave Candidata de R
 - R1 inclui AB, pelo que este passo não altera Decomp

- \therefore Decomp é uma Decomposição na 3FN que
 - Preserva Dependências Funcionais e que é *Lossless Join*

Algoritmo para 3FN - Exemplo 2

- Sendo $R(\text{numBI}, \text{nome}, \text{numMatriculaAutomovel})$ e
 - $F = \{\text{numBI} \rightarrow \text{nome}, \text{numMatriculaAutomovel} \rightarrow \text{nome}\}$
 - aplicar o algoritmo para encontrar uma Decomposição na 3FN
- (1) Cobertura Mínima G de F
 - $G = \{\text{numBI} \rightarrow \text{nome}, \text{numMatriculaAutomovel} \rightarrow \text{nome}\}$
- (2) Decomposição que Preserva as Dependências Funcionais de F
 - $D = \{R1(\text{numBI}, \text{nome}), R2(\text{numMatriculaAutomovel}, \text{nome})\}$
- (3) Garantir que a Decomposição D é *Lossless Join*
 - Determinar chaves candidatas de R e G
 - $\{\text{numBI}, \text{numMatriculaAutomovel}\}^+ = \{\text{numBI}, \text{numMatriculaAutomovel}, \text{nome}\}$
 - Logo $\{\text{numBI}, \text{numMatriculaAutomovel}\}$ é a única chave candidata de R
 - pelo que $D = D \cup \{R3(\text{numBI}, \text{numMatriculaAutomovel})\}$
- $\therefore D = \{ R1(\text{numBI}, \text{nome}), R2(\text{numMatriculaAutomovel}, \text{nome}), R3(\text{numBI}, \text{numMatriculaAutomovel}) \}$

Algoritmo para 3FN - Exemplo 3

- Sendo $R(A, B, C, D, E)$ e $F = \{A \rightarrow BCDE, CD \rightarrow E\}$
 - aplicar o “processo intuitivo” para encontrar uma Decomposição na 3FN
 - aplicar o algoritmo para encontrar uma Decomposição na 3FN
- Aplicando o “processo intuitivo”
 - $D = \{R1(A, B, C, D), R2(C, D, E)\}$
- Aplicando o Algoritmo
 - $G = \{A \rightarrow B, A \rightarrow C, A \rightarrow D, CD \rightarrow E\}$
 - Chaves candidatas de R: A
 - $D = \{R1(A, B), R2(A, C), R3(A, D), R4(C, D, E)\}$
- Neste exemplo em ambas as Decomposições temos que:
 - não existe perda de Dependências Funcionais
 - são *Lossless Join*
- Neste exemplo com o “processo intuitivo” chega-se a uma Decomposição com menos elementos que a gerada pelo Algoritmo

Motivação para a Forma Normal de *Boyce-Codd* (FNBC)

- Considere-se um cenário sujeito às seguintes regras:
 - ❑ Um estudante pode frequentar vários anos simultaneamente
 - ❑ Cada estudante, e para cada ano que frequenta, está afecto a um professor responsável
 - ❑ A cada ano estão afectos vários professores como responsáveis
 - ❑ Cada professor é responsável apenas por um ano

- Dependências Funcionais
 - ❑ {numEstudante, anoLicenciatura} → professorResponsavel
 - ❑ professorResponsavel → anoLicenciatura

Motivação para a FNBC (cont.)

■ Chaves Candidatas

- $\{\text{numEstudante}, \text{anoLicenciatura}\}^+ =$
 - $\{\text{numEstudante}, \text{anoLicenciatura}, \text{professorResponsavel}\}$
- $\{\text{numEstudante}, \text{ProfessorResponsavel}\}^+ =$
 - $\{\text{numEstudante}, \text{professorResponsavel}, \text{anoLicenciatura}\}$

■ Esquema de Relação na 3FN

- **RESPONSABEL**(numEstudante, anoLicenciatura, professorResponsavel)

■ Relação

RESPONSABEL

<u>numEstudante</u>	<u>anoLicenciatura</u>	professorResponsavel
36578	2º	Gil
36578	3º	Pascoal
17738	2º	Gil
17738	3º	Silva
17738	4º	Margarida
24451	1º	Joana

Inconvenientes do Esquema de Relação RESPONSÁVEL

- Embora o Esquema de Relação RESPONSÁVEL esteja na 3FN, tem ainda também problemas.
- Anomalia de Inserção de um novo tuplo (linha)
 - Se for pretendido inserir informação indicando que professor é responsável por determinado ano lectivo:
 - José Santos, responsável pelo 1º ano
 - Não é possível inserir essa informação enquanto esse professor não for afecto a pelo menos um estudante
- Anomalia de Remoção de um tuplo (linha) já existente
 - Se para determinado ano lectivo, um professor for responsável por um único estudante e se esse estudante for removido, perde-se a informação de que o professor é co-responsável por esse ano
 - se o estudante 36478 anular a sua inscrição no 3º ano, é perdida a informação do Professor Pascoal ser co-responsável pelo 3º ano

Inconvenientes (cont.)

- Anomalia de Actualização de um tuplo (linha) já existente
 - Se determinado aluno mudar a sua inscrição de um ano lectivo para outro, é necessário garantir que o Professor Responsável é um dos que é responsável por esse novo ano lectivo
 - Se o estudante 36578, mudar a sua inscrição do 2º para o 3º ano, é necessário garantir que o Professor responsável seja um dos do 3º ano. Qual deles ? O Pascoal , o Silva ou, no caso do Gil apenas ser responsável por este estudante passar o Gil para co-responsável pelo 3º ano ?
- Estes inconvenientes que a 3FN exhibe, deverão ser resolvidos através da passagem dos dados para a Forma Normal *Boyce-Codd*

Forma Normal *Boyce-Codd* (FNBC)

- Um Esquema de Relação está na Forma Normal *Boyce-Codd* (FNBC) se e só se:
 - sempre que $X \rightarrow A$ se verifica e $A \notin X$, tem-se que,
 - X é uma chave candidata
- A única diferença entre a 3FN e a FNBC é que:
 - A 3FN permitia que Atributos Primos sejam Funcionalmente Dependentes de Atributo não-Primos, e isso não é possível na FNBC
- Considerando o exemplo anterior temos
 - `RESPONSAVEL(numEstudante, anoLicenciatura, professorResponsavel)`
 - e $\{\text{professorResponsavel} \rightarrow \text{anoLicenciatura}\}$
 - e `professorResponsavel` não é uma Superchave
 - \therefore o Esquema de Relação `RESPONSAVEL` não está na FNBC

Desdobrar para passar à FNBC

RESPONSAVEL

<u>numEstudante</u>	<u>anoLicenciatura</u>	professorResponsavel
---------------------	------------------------	----------------------

Atributos da DF

ProfessorResponsável → AnoLicenciatura

Restantes atributos
menos
a parte direita da DF

PROFESSOR

<u>professorResponsável</u>	<u>anoLicenciatura</u>
Gil	2º
Pascoal	3º
Silva	3º
Margarida	4º
Joana	1º

Aluno_Responsavel

<u>numEstudante</u>	<u>professorResponsavel</u>
36578	Gil
36578	Pascoal
17738	Gil
17738	Silva
17738	Margarida
24451	Joana

- Decomposição na FNBC mas perdeu a Dependência Funcional
 - {numEstudante, anoLicenciatura} → professorResponsavel

Decomposição na FNBC - Algoritmo

- Entrada: $R(A_1, \dots, A_n)$ e F um conjunto de Dependências Funcionais
- Saída: Decomposição $D = \{R_1, R_2, \dots, R_n\}$ de R na FNBC que
 - Preserva Informação (é *Lossless Join*)
 - mas que pode não preservar as Dependências Funcionais
- Este Algoritmo tem dois passos:
 - (1) Considerar D o próprio R
 - (2.a) Para cada $X \rightarrow Y$ em que $Y \not\subseteq X$ e X não é Superchave, decompor D em Esquemas de Relação $R_1(\underline{X}Y)$ e R_2 com os Atributos de $R - Y$.
 - (2.b) As Chaves Candidatas de R_2 são todas as Chaves Candidatas de R menos as que incluíam o Atributo Y
 - (2.c) Considerar $D := R_1 \cup$ aplicação recursiva deste passo (2) a R_2

Algoritmo para FNBC - Exemplo

- Sendo AULA(aluno, disciplina, professor) e
 - $F = \{ \{aluno, disciplina\} \rightarrow professor, \{professor\} \rightarrow disciplina \}$,
 - aplicar o algoritmo para encontrar uma Decomposição na FNBC
- (1) $D = \{ AULA(aluno, disciplina, professor) \}$
- (2) Chaves Candidatas: $\{aluno, disciplina\}$ e $\{aluno, professor\}$
 - Considerando $professor \rightarrow disciplina$, $X = professor$ e $A = disciplina$
 - $R1(\underline{professor}, disciplina)$
 - $R2(\underline{aluno}, \underline{professor})$
 - tem os Atributos de R - $A = \{aluno, professor\}$
 - $D = \{ R1(\underline{professor}, disciplina), R2(\underline{aluno}, \underline{professor}) \}$
- D está na FNBC mas não Preserva a Dependência Funcional
 - $\{aluno, disciplina\} \rightarrow professor$

Exemplo

- Considere-se a descrição do seguinte cenário:
 - “*Um hospital está dividido em serviços (Cirurgia, Cardiologia, etc). Cada médico está inserido num serviço. Os pacientes num serviço são sempre atendidos por um só médico.*”
- Desta descrição pode-se inferir:
 - $F = \{ \text{medico} \rightarrow \text{servico}, \{ \text{paciente}, \text{servico} \} \rightarrow \text{medico} \}$
 - $R (\text{medico}, \text{servico}, \text{paciente})$
- Sendo que as chaves candidatas de R e F são:
 - $CC = \{ \{ \text{paciente}, \text{Medico} \} \}$
- Resultante no seguinte esquema de Relação, que está na 3FN:
 - $R (\underline{\text{paciente}}, \underline{\text{servico}}, \text{medico})$

Exemplo (cont. 1)

R (paciente, servico, medico)

- O Esquema de Relação R está na 3FN, no entanto apresenta algumas anomalias.
- Por exemplo, só é possível registrar que um médico pertence a um dado serviço quando lhe for atribuído um paciente.
- R não está na FNBC, ou seja, não está a ser atendida a DF:
 - ❑ medico → servico
- Uma Decomposição na FNBC será:
 - ❑ R1 (medico, servico)
 - ❑ R2 (paciente, medico)

Exemplo (cont. 2)

- R1 e R2 estão na FNBC. Esta solução, apesar de resolver alguns problemas, introduz outro:
 - É agora possível registar o mesmo paciente como podendo ser atendido por dois médicos do mesmo serviço.

R1

<u>medico</u>	servico
Miguel	Pediatria
Maria	Pediatria

R2

<u>paciente</u>	<u>medico</u>
Jorge	Miguel
Jorge	Maria

- Ou seja, R1 e R2 não podem ser tratadas de forma independente
- Este problema acontece porque na passagem de 3FN para a FNBC se perdeu a DF
 - {paciente, servico} → medico
- Dado que esta DF não se encontra incorporada no Modelo Relacional encontrado, deverá ser tratada no nível aplicativo

Algo análogo aconteceria se $D = \{ R1(\text{medico}, \text{servico}), R2(\text{paciente}, \text{servico}, \text{medico}) \}$

Dependências Multivalor (4FN)

- Seja $R(A_1, A_2, \dots, A_n)$ um Esquema de Relação
- Diz-se que existe uma Dependência Multivalor entre os conjuntos de Atributos $X \subseteq \{A_1, A_2, \dots, A_n\}$ e $Y \subseteq \{A_1, A_2, \dots, A_n\}$ e representa-se por $X \twoheadrightarrow Y$, quando cada valor de X determina um conjunto de valores de Y independente dos valores dos restantes Atributos de R

livro	autor	conteudo
Fundamentals of Database Systems	Elmasri	Modelo Relacional
Fundamentals of Database Systems	Elmasri	Desenho de Bases de Dados
Fundamentals of Database Systems	Navathe	Modelo Relacional
Fundamentals of Database Systems	Navathe	Desenho de Bases de Dados
The ODBC Solution	Robert Signore	Estrutura do ODBC
The ODBC Solution	Robert Signore	Directivas SQL

- Cada livro tem os seus autores e não existe uma identificação de cada autor com cada um dos conteúdos de cada livro
 - Existem duas Dependências Multivalor (DMV)
 - $\text{livro} \twoheadrightarrow \text{autor}$
 - $\text{livro} \twoheadrightarrow \text{conteudo}$
- Um livro pode ter vários autores
Um livro pode ter vários conteúdos
Um autor pode ter vários conteúdos, mas depende de livro

Dependências Multivalor (cont.)

- Para definir formalmente $X \twoheadrightarrow Y$, considere-se
 - $R(A_1, A_2, \dots, A_n)$ um Esquema de Relação
 - $Z = R - X - Y$
 - $Y_{xz} = \{ y : \langle x, y, z \rangle \in R \}$ para $z \in Z$
- Diz-se que a Dependência Multivalor $X \twoheadrightarrow Y$ existe se e só se
 - $Y_{xz} = Y_{xz'} \neq \emptyset$ para $\forall x, z, z'$
- Regras de Inferência
 - Tal como existem Regras de Inferência para as Dependências Funcionais também estão definidas regras para as Dependências Multivalor
 - Existem também regras que envolvem simultaneamente Dependências Funcionais e Multivalor

Regras de Inferência

■ Básicas

- $Y \subseteq X \Rightarrow X \twoheadrightarrow Y$ (Reflexibilidade)
- $X \twoheadrightarrow Y \Rightarrow X \twoheadrightarrow R-XY$ (Complementaridade)
- $X \twoheadrightarrow Y \wedge W \subseteq Z \Rightarrow XZ \twoheadrightarrow YW$ (Aumento)
- $X \twoheadrightarrow Y \wedge Y \twoheadrightarrow Z \Rightarrow X \twoheadrightarrow (Z-Y)$ (Transitividade)

■ Deriváveis

- $X \twoheadrightarrow Y \wedge YW \twoheadrightarrow Z \Rightarrow XW \twoheadrightarrow (Z-YW)$ (Pseudo-Transitividade)
- $X \twoheadrightarrow Y \wedge X \twoheadrightarrow Z \Rightarrow X \twoheadrightarrow YZ$ (União)
- $X \twoheadrightarrow YZ \Rightarrow X \twoheadrightarrow (Y-Z) \wedge X \twoheadrightarrow (Z-Y) \wedge X \twoheadrightarrow (Y \cap Z)$ (Decomposição)

■ Para Dependências Multivalor e Funcionais

- $X \rightarrow Y \Rightarrow X \twoheadrightarrow Y$ (Replicação)
- $Z \subseteq W \wedge Y \cap W = \emptyset \wedge X \twoheadrightarrow W \wedge Y \rightarrow Z \Rightarrow X \rightarrow Z$ (Coalescência)

Quarta Forma Normal 4FN

- Dependências Multivalor (DMV) Elementares
 - Dado um Esquema de Relação R , $X \twoheadrightarrow Y$ é Elementar se existirem em R outros Atributos, se X e Y são disjuntos e se não existirem em R outras DMV $X' \twoheadrightarrow Y'$ tais que $X' \subset X$ e $Y' \subset Y$
- Um Esquema de Relação está na 4ª Forma Normal (4FN) em relação a um conjunto de Dependências F , quando:
 - para qualquer Dependência Multivalor elementar $X \twoheadrightarrow Y$ em F
 - então X é uma Superchave de R
- No exemplo anterior $R(\text{livro}, \text{autor}, \text{conteudo})$ não estava na 4FN
- $R(X, Y, Z)$ com $X \twoheadrightarrow Y$ for elementar e X não-Primo, não está na 4FN. Pode ser decomposto em
 - $R_1(X, Y)$ e $R_2(X, Z)$
- No exemplo fica $R_1(\text{livro}, \text{autor})$ e $R_2(\text{livro}, \text{conteudo})$

Quinta Forma Normal 5FN

- Também denominada de:
 - **Project-Join Normal Form (PJ/NF)**
- Uma relação está na 5FN se e só se está na 4FN e cada dependência de junção é implicada pelas chaves candidatas
- Dependência de Junção
 - Seja $R = \{R_1 \cup R_2 \cup \dots \cup R_N\}$, a relação $r(R)$ satisfaz a dependência de junção $*(R_1, R_2, \dots, R_N)$ se:
 - $r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_N}(r)$

Outras formas normais

■ Forma Normal Domínio/Chave (Domain/key normal form) DKNF

- requer que o esquema de relação não esteja sujeito a nenhuma restrição, para além das restrições de domínio e de chave

■ Sexta Forma Normal 6FN

- Um esquema de relação está na 6FN se e só se não tem dependências de junção não triviais *
- Uma relação na 6FN também está na 5FN

* Uma dependência de junção é não trivial se uma das R_i é o próprio R

Resumo das Formas Normais

- **1FN:** Os valores de todos os Domínios (de Atributos) são atômicos
- **2FN:** 1FN + todos os Atributos não-Primos são totalmente dependentes de cada uma das Chaves Candidatas
- **3FN:** 2FN + nenhum dos Atributos não-Primos é transitivamente dependente de alguma das Chaves Candidatas
- **FNBC:** 3FN + nenhum dos Atributos Primos é dependente de algum Atributo não-Primo
- **4FN:** FNBC + não existem Dependências Multivalor elementares de Atributos não-Primos
- **5FN** (PJNF - “Project Join Normal Form”)
- **DKNF** (“Domain Key Normal Form”)
Para mais informações acerca da 4FN, 5FN, DKNF consultar Korth e Silberchatz, “Database System Concepts”, McGraw-Hill

Estratégias de Normalização

- A normalização raramente percorre todas as Formas Normais
- Frequentemente, o analista reconhece, por experiência própria, que um determinado Esquema de Relação não está normalizado e coloca-o directamente na 3FN ou FNBC
- Uma estratégia usada consiste em
 - começar a normalização a partir de um único Esquema de Relação - Relação Universal, e
 - através de iterações sucessivas, utilizando a análise de Dependências Funcionais, construir os Esquemas de Relação normalizados
- Outra estratégia consiste em,
 - normalizar, refinando os Esquemas de Relação resultantes da aplicação da modelação com diagramas Entidade-Associação

Consequências de Normalização

- Na prática, a normalização não deve ser levada às últimas consequências, pois a proliferação de Esquemas de Relação pode ter consequências ao nível do desempenho global do sistema.
- Quanto “mais espalhados” estiverem os dados por várias tabelas, mais junções entre tabelas vão ser necessárias para obter a mesma informação.
- Deparam-se assim dois objectivos, frequentemente conflituosos:
 - Pretendem-se sistemas flexíveis, sem problemas de redundância.
 - Pretendem-se sistemas com alto desempenho.
- Pretende-se que o esquema de uma Base de Dados seja equilibrado que nunca ponha em risco a integridade dos dados, mas que simultaneamente tenha um desempenho aceitável pois só assim será utilizado.
 - Assim, na maioria dos casos, a normalização pára na 3FN ou na FNBC.

Desnormalização

- Por vezes, a necessidade de obter determinados níveis de desempenho pode obrigar a proceder a uma desnormalização.
- Desnormalização - De uma forma consciente e com o único objectivo de proporcionar melhor nível de desempenho, é introduzida redundância em partes muito concretas do modelo.
- A desnormalização tem como objectivo adequar o modelo ao padrão de acessos mais frequentes por parte do nível aplicacional.
- O processo de desnormalização apenas deve ocorrer depois de se ter obtido um conhecimento concreto dos padrões de acesso que se pretendem otimizar e de qual o efectivo desempenho exibido pelo modelo normalizado.
- O esquema desnormalizado assim como a sua justificação devem constar de forma clara na documentação do sistema.