

Base de Dados

BD 06 - Normalização

Vitor Vaz da Silva

Base de Dados - Índice

- Motivação
- Dependências Funcionais
- Fecho de um conjunto de Dependências Funcionais
- Fecho de um conjunto de Atributos
- Chaves Candidatas
- Normalização

TABLE 5.1

A Sample Report Layout

PROJECT NUMBER	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS	CHARGE/ HOUR	HOURS BILLED	TOTAL CHARGE
15	Evergreen	103	June E. Arbough	Elec. Engineer	\$ 85.50	23.8	\$ 2,034.90
		101	John G. News	Database Designer	\$105.00	19.4	\$ 2,037.00
		105	Alice K. Johnson *	Database Designer	\$105.00	35.7	\$ 3,748.50
		106	William Smithfield	Programmer	\$ 35.75	12.6	\$ 450.45
		102	David H. Senior	Systems Analyst	\$ 96.75	23.8	\$ 2,302.65
				Subtotal			\$10,573.50
18	Amber Wave	114	Annelise Jones	Applications Designer	\$ 48.10	25.6	\$ 1,183.26
		118	James J. Frommer	General Support	\$ 18.36	45.3	\$ 831.71
		104	Anne K. Ramoras *	Systems Analyst	\$ 96.75	32.4	\$ 3,134.70
		112	Darlene M. Smithson	DSS Analyst	\$ 45.95	45.0	\$ 2,067.75
				Subtotal			\$ 7,265.52
22	Rolling Tide	105	Alice K. Johnson	Database Designer	\$105.00	65.7	\$ 6,998.50
		104	Anne K. Ramoras	Systems Analyst	\$ 96.75	48.4	\$ 4,682.70
		113	Delbert K. Joenbrood	Applications Designer	\$ 48.10	23.6	\$ 1,135.16
		111	Geoff B. Wabash	Clerical Support	\$ 26.87	22.0	\$ 591.14
		106	William Smithfield	Programmer	\$ 35.75	12.8	\$ 457.60
				Subtotal			\$13,765.10
25	Starflight	107	Maria D. Alonzo	Programmer	\$ 35.75	25.6	\$ 915.20
		115	Travis B. Bawangi	Systems Analyst	\$ 96.75	45.8	\$ 4,431.15
		101	John G. News *	Database Designer	\$105.00	56.3	\$ 5,911.50
		114	Annelise Jones	Applications Designer	\$ 48.10	33.1	\$ 1,592.11
		108	Ralph B. Washington	Systems Analyst	\$ 96.75	23.6	\$ 2,283.30
		118	James J. Frommer	General Support	\$ 18.36	30.5	\$ 559.98
		112	Darlene M. Smithson	DSS Analyst	\$ 45.95	41.4	\$ 1,902.33
				Subtotal			\$17,595.57
				Total			\$49,199.69

Note: * indicates project leader

Table name: RPT_FORMAT

Database name: Ch05_ConstructCo

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
		101	John G. News	Database Designer	105.00	19.4
		105	Alice K. Johnson *	Database Designer	105.00	35.7
		106	William Smithfield	Programmer	35.75	12.6
		102	David H. Senior	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	48.10	24.6
		118	James J. Frommer	General Support	18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
		111	Geoff B. Wabash	Clerical Support	26.87	22.0
		106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	96.75	45.8
		101	John G. News *	Database Designer	105.00	56.3
		114	Annelise Jones	Applications Designer	48.10	33.1
		108	Ralph B. Washington	Systems Analyst	96.75	23.6
		118	James J. Frommer	General Support	18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	45.95	41.4

- Chaves com valores a NULL
- Um Empregado pode trabalhar em diversos projectos
- Há redundâncias, ex: nos nomes dos Empregados
- Ao preencher a tabela pode haver uma inconsistência caso alguém escreva Elect. Eng. Em vez de Elect Engineer

Anomalias

- Alterar – por exemplo se o JOB_CLASS de 105 mudar
- Adicionar – adicionar um Empregado que ainda não tem projecto associado. (*Criar projecto “fantasma”?*)
- Apagar – apagar um Projecto apagaria todos os dados de Empregados que teriam apenas esse projecto
apagar um Empregado de um projecto onde ele fosse o único iria apagar o projecto. (*Criar Empregado “fantasma”?*)

Normalização

Normalização é um processo formal para decidir quais os atributos que devem ser agrupados numa relação evitando anomalias que provoquem inconsistência:

Alterar, Adicionar, Apagar

consiste em reduzir sucessivamente as relações com anomalias, produzindo relações mais pequenas sem anomalias.

Objectivos da Normalização

- ❖ Minimizar redundância de dados, evitando anomalias e desperdício de espaço de armazenamento;
- ❖ Simplificar o cumprimento das restrições de integridade do modelo relacional;
- ❖ Tornar fácil as operações de inserção, atualização e remoção de dados;
- ❖ Para além de representar uma realidade o modelo encontrado deve facilitar futuros desenvolvimentos.

Normalização

A normalização é realizada por passos, em que cada passo corresponde a uma **Forma Normal**;

- ❖ Uma **Forma Normal** corresponde ao estado de uma relação que garante que certas regras são satisfeitas;
- ❖ Estas regras estão relacionadas com dependências entre atributos, designadas por **DEPENDÊNCIAS FUNCIONAIS**.

Dependência Funcional

- ✓ Um atributo pode ser funcional dependente de dois ou mais atributos, por exemplo, dada a relação:

EMPREGADO_CURSO(EmpID, NomeCurso,
DataObtenção)

podemos estabelecer que a seguinte Dependência Funcional:

$\text{EmpID, NomeCurso} \rightarrow \text{DataObtenção}$

DataObtenção é funcionalmente dependente
de EmpID e NomeCurso

Determinantes

NIF → Nome, Endereço, DataNasc

Matricula → Marca, Modelo, Cor

ISBN → Titulo, Autor, Editor

EmpID, NomeCurso → DataObtenção

EmpID → Nome, Depart, Salario

Chaves

- ✓ **Chave Candidata:** Atributo ou combinação de atributos que identifica unicamente uma linha de uma relação e tem as seguintes propriedades:
 - ❖ **Identificação única:** Cada atributo não chave da relação tem de ser funcionalmente dependente da chave;
 - ❖ **Não redundante:** Nenhum atributo na chave pode ser removido sem destruir a propriedade anterior de identificação única.

- ✓ **Chave Primária:** Chave escolhida entre todas as chaves Candidatas como a que melhor determina qualquer linha da relação

Table name: DATA_ORG_1NF

Database name: Ch05_ConstructCo

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
15	Evergreen	101	John G. News	Database Designer	105.00	19.4
15	Evergreen	105	Alice K. Johnson *	Database Designer	105.00	35.7
15	Evergreen	106	William Smithfield	Programmer	35.75	12.6
15	Evergreen	102	David H. Senior	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	48.10	24.6
18	Amber Wave	118	James J. Frommer	General Support	18.36	45.3
18	Amber Wave	104	Anne K. Ramoras *	Systems Analyst	96.75	32.4
18	Amber Wave	112	Darlene M. Smithson	DSS Analyst	45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	105.00	64.7
22	Rolling Tide	104	Anne K. Ramoras	Systems Analyst	96.75	48.4
22	Rolling Tide	113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
22	Rolling Tide	111	Geoff B. Wabash	Clerical Support	26.87	22.0
22	Rolling Tide	106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	35.75	24.6
25	Starflight	115	Travis B. Bawangi	Systems Analyst	96.75	45.8
25	Starflight	101	John G. News *	Database Designer	105.00	56.3
25	Starflight	114	Annelise Jones	Applications Designer	48.10	33.1
25	Starflight	108	Ralph B. Washington	Systems Analyst	96.75	23.6
25	Starflight	118	James J. Frommer	General Support	18.36	30.5
25	Starflight	112	Darlene M. Smithson	DSS Analyst	45.95	41.4

Importante

Desenvolver esquema de relação cujo significado seja fácil de explicar. (ex: evitar misturar informação proveniente de várias entidades no mesmo esquema de relação)

Desenhar os esquemas de relação de forma a que não ocorram anomalias de inserção, remoção e atualização (evitar redundância).

Evitar a utilização de atributos que possam ter uma frequência grande de ocorrências de valores NULL.

Desenvolver os esquemas das relações de forma a que a informação relacionada possa ser recuperada através de junções naturais (igualdade nos pares de atributos chave primária e chave estrangeira relacionados)

Dependência funcional

Uma relação R
satisfaz a dependência funcional $X \rightarrow Y$
se quaisquer dois tuplos $t1$ e $t2$
que apresentem o mesmo valor nos atributos X
também apresentam o mesmo valor nos atributos Y .

Exemplo – Dependência Funcional

Aluno

NumAl	NomeAl
41932	Ana
42083	Bruno

{NumAl} → {NomeAl}

Disciplina

Codigo	NomeAl
BD	Base de Dados
SI-1	Sistemas de Informação 1

{Código} → {NomeDisc}

Frequencia

NumAl	Codigo	Nota
42083	SI-1	9
41932	BD	14

{NumAl, Código} → {Nota}

Dependência Funcional - Regras de inferência

Uma **dependência funcional** ($X \rightarrow Y$)
decorre logicamente, (ou **deriva** logicamente, ou é inferida)
de um **conjunto de dependências funcionais** F ($F \models X \rightarrow Y$)
se $X \rightarrow Y$ **pertencer a** F
ou se for derivada das dependências funcionais existentes em F
pelas seguintes regras de inferência:

- | | |
|--------------------------------|--|
| 1 - Reflexividade: | se $X \supseteq Y$, então $X \rightarrow Y$ |
| 2 - Aumento | $\{X \rightarrow Y\} \models XZ \rightarrow YZ$ |
| 3 - Transitividade: | $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$ |
| 4 - Decomposição ou projecção: | $\{X \rightarrow YZ\} \models X \rightarrow Y$ |
| 5 - União, ou adição | $\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$ |
| 6 - Pseudo-transitividade: | $\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$ |

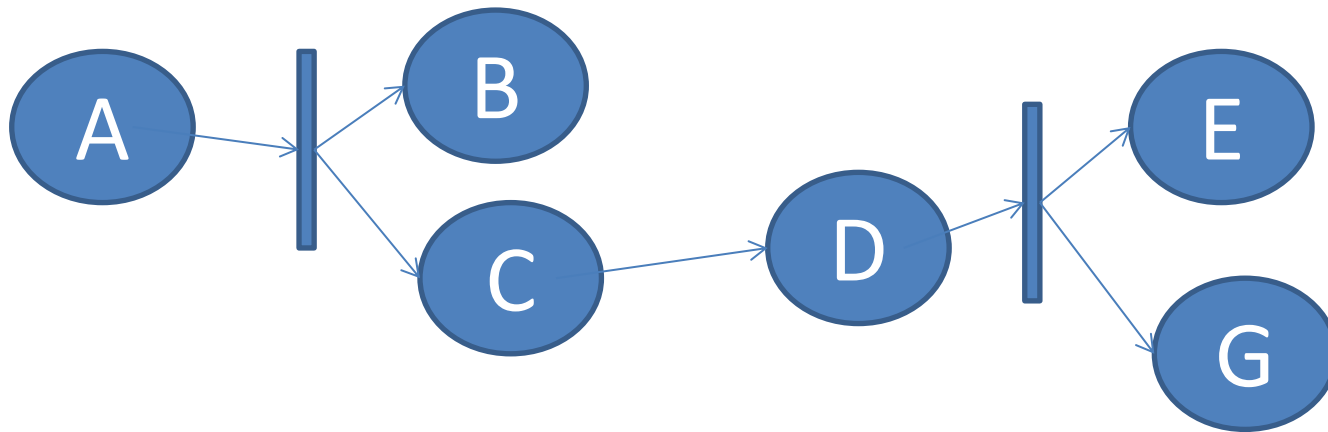
Axiomas
(ou regras de
inferência)
de Armstrong

Exemplo: $F = \{\{A\} \rightarrow \{B,C\}, \{C\} \rightarrow \{D\}, \{D\} \rightarrow \{E,G\}\} \models \{A\} \rightarrow \{G\}$

Exemplo:

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

$F \models \{A\} \rightarrow \{G\}$



Dependência Funcional - Grafos

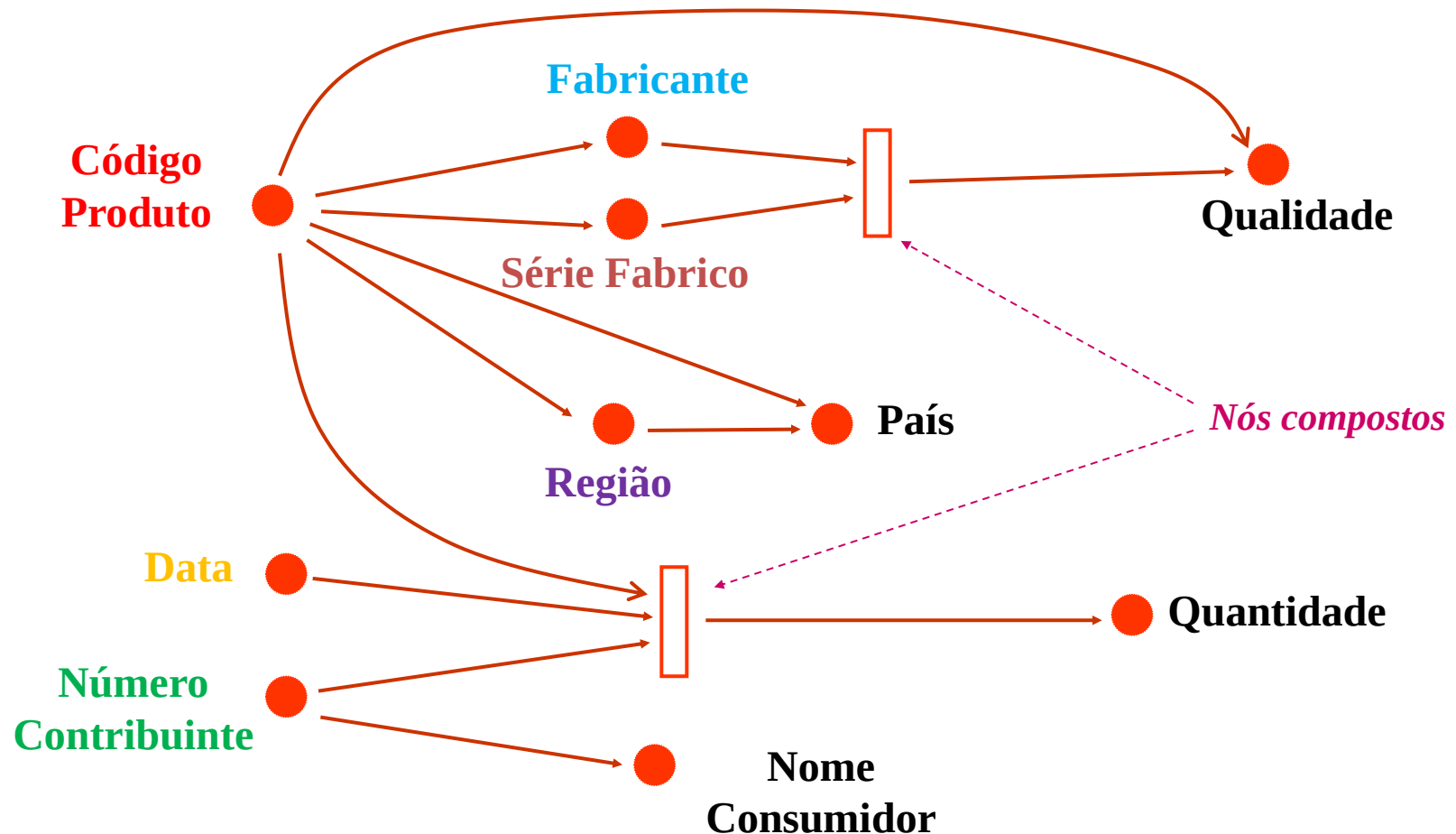
Seja a relação:

CONSUMOS(Código Produto, Fabricante, Série Fabrico, Qualidade, Região, País, Número Contribuinte, Nome Consumidor, Data, Quantidade)

e as seguintes dependências funcionais sobre os seus atributos:

**F = {{Código Produto} → {Fabricante},
{Código Produto} → {Série Fabrico},
{Código Produto} → {Qualidade},
{Código Produto} → {Região},
{Código Produto} → {País},
{Região} → {País},
{Fabricante, Série Fabrico} → {Qualidade},
{Número Contribuinte} → {Nome consumidor},
{Código Produto, Número contribuinte, Data} → {Quantidade}}
}**

$F = \{$
 $\{\text{Código Produto}\} \rightarrow \{\text{Fabricante}\}, \quad \{\text{Código Produto}\} \rightarrow \{\text{Série Fabrico}\},$
 $\{\text{Código Produto}\} \rightarrow \{\text{Qualidade}\}, \quad \{\text{Código Produto}\} \rightarrow \{\text{Região}\},$
 $\{\text{Código Produto}\} \rightarrow \{\text{País}\}, \quad \{\text{Número Contribuinte}\} \rightarrow \{\text{Nome consumidor}\},$
 $\{\text{Região}\} \rightarrow \{\text{País}\}, \quad \{\text{Fabricante}, \text{Série Fabrico}\} \rightarrow \{\text{Qualidade}\},$
 $\{\text{Código Produto}, \text{Número contribuinte}, \text{Data}\} \rightarrow \{\text{Quantidade}\}$
 $\}$



Fecho e Cobertura de um Conjunto de Dependências Funcionais

Dado um conjunto de dependências funcionais F ,
o fecho de F (F^+) é o conjunto de todas as dependências funcionais derivadas de F por utilização das regras de inferência

Dois conjuntos de dependências funcionais F_1 e F_2 são equivalentes
se $F_1^+ = F_2^+$

Uma DF $X \rightarrow Y$ deriva de um conjunto de DF F
se se verificar que $X \rightarrow Y \in F^+$

Dados dois conjuntos de DF F_1 e F_2 ,
**se toda a dependência funcional de F_2 derivar de F_1 ,
diz-se que F_1 é uma cobertura de F_2 ($F_1 \models F_2$).**
Também se pode dizer que F_2 deriva logicamente de F_1
 $F_2 \in F_1$

Fecho de um Conjunto de Dependências Funcionais

Seja M :

$R(\text{NumAl}, \text{BiPai}, \text{NomePai})$

e

$F = \{\{\text{NumAl}\} \rightarrow \{\text{BiPai}\}, \{\text{BiPai}\} \rightarrow \{\text{NomePai}\}\}$

$F^+ = \{\{\text{NumAl}\} \rightarrow \{\text{BiPai}\}, \{\text{BiPai}\} \rightarrow \{\text{NomePai}\},$

$\{\} \rightarrow \{\}, \{\text{NumAl}\} \rightarrow \{\}, \{\text{NumAl}\} \rightarrow \{\text{NumAl}\}, \{\text{BiPai}\} \rightarrow \{\}, \{\text{BiPai}\} \rightarrow \{\text{BiPai}\},$

$\{\text{NomePai}\} \rightarrow \{\}, \{\text{NomePai}\} \rightarrow \{\text{NomePai}\},$

$\{\text{NumAl}, \text{BiPai}\} \rightarrow \{\}, \{\text{NumAl}, \text{BiPai}\} \rightarrow \{\text{NumAl}\}, \{\text{NumAl}, \text{BiPai}\} \rightarrow \{\text{BiPai}\},$

$\{\text{NumAl}, \text{BiPai}\} \rightarrow \{\text{NumAl}, \text{BiPai}\}, \{\text{NumAl}, \text{NomePai}\} \rightarrow \{\},$

$\{\text{NumAl}, \text{NomePai}\} \rightarrow \{\text{NumAl}\}, \{\text{NumAl}, \text{NomePai}\} \rightarrow \{\text{NomePai}\},$

$\{\text{BiPai}, \text{NomePai}\} \rightarrow \{\}, \{\text{BiPai}, \text{NomePai}\} \rightarrow \{\text{BiPai}\},$

$\{\text{BiPai}, \text{NomePai}\} \rightarrow \{\text{NomePai}\}, \{\text{BiPai}, \text{NomePai}\} \rightarrow \{\text{BiPai}, \text{NomePai}\},$

$\{\text{NumAl}, \text{BiPai}, \text{NomePai}\} \rightarrow \{\}, \{\text{NumAl}, \text{BiPai}, \text{NomePai}\} \rightarrow \{\text{NumAl}\},$

$\{\text{NumAl}, \text{BiPai}, \text{NomePai}\} \rightarrow \{\text{BiPai}\}, \{\text{NumAl}, \text{BiPai}, \text{NomePai}\} \rightarrow \{\text{NomePai}\},$

$\{\text{NumAl}, \text{BiPai}, \text{NomePai}\} \rightarrow \{\text{NumAl}, \text{BiPai}\},$

$\{\text{NumAl}, \text{BiPai}, \text{NomePai}\} \rightarrow \{\text{NumAl}, \text{NomePai}\},$

$\{\text{NumAl}, \text{BiPai}, \text{NomePai}\} \rightarrow \{\text{BiPai}, \text{NomePai}\},$

$\{\text{NumAl}, \text{BiPai}, \text{NomePai}\} \rightarrow \{\text{NumAl}, \text{BiPai}, \text{NomePai}\}, \{\text{NumAl}\} \rightarrow \{\text{NomePai}\}, \{\text{NumAl}, \text{NomePai}\} \rightarrow \{\text{BiPai}\}$

~~$\{\text{NumAl}, \text{BiPai}\} \rightarrow \{\text{BiPai}\}, \{\text{NumAl}, \text{NomePai}\} \rightarrow \{\text{BiPai}, \text{NomePai}\},$~~

~~$\{\text{NumAl}, \text{BiPai}, \text{NomePai}\} \rightarrow \{\text{BiPai}, \text{NomePai}\},$~~

$\{\text{BiPai}, \text{NumAl}\} \rightarrow \{\text{NomePai}, \text{NumAl}\},$

~~$\{\text{BiPai}, \text{NomePai}, \text{NumAl}\} \rightarrow \{\text{BiPai}, \text{NomePai}, \text{NumAl}\}, \dots$~~

}

Por reflexividade

Por transitividade

Por pseudo-transitividade

Por aumento

Explosão combinatória! (veremos alternativas, mais tarde)

✕ repetido

Fecho de um Conjunto de Atributos

Seja **F** um conjunto de **dependências funcionais** relativas a um universo de **atributos U**, e **X** um **subconjunto de U**.

Define-se **fecho do conjunto X relativamente a F** como o conjunto **X^+ dos atributos A** tais que, ou **$A \in X$** , ou **$X \rightarrow A$** pode ser derivado de **$F^{(*)}$** .

Algoritmo:Determinar X^+ , dados X e F

$X^+ = X$

Repetir

Aux = X^+

Para cada dependência funcional $Y \rightarrow Z$ em F fazer

Se $Aux \supseteq Y$, então $X^+ = X^+ \cup Z$

até que $X^+ = Aux$

*aplicar-se as regras de
inferência de forma
sistemática*

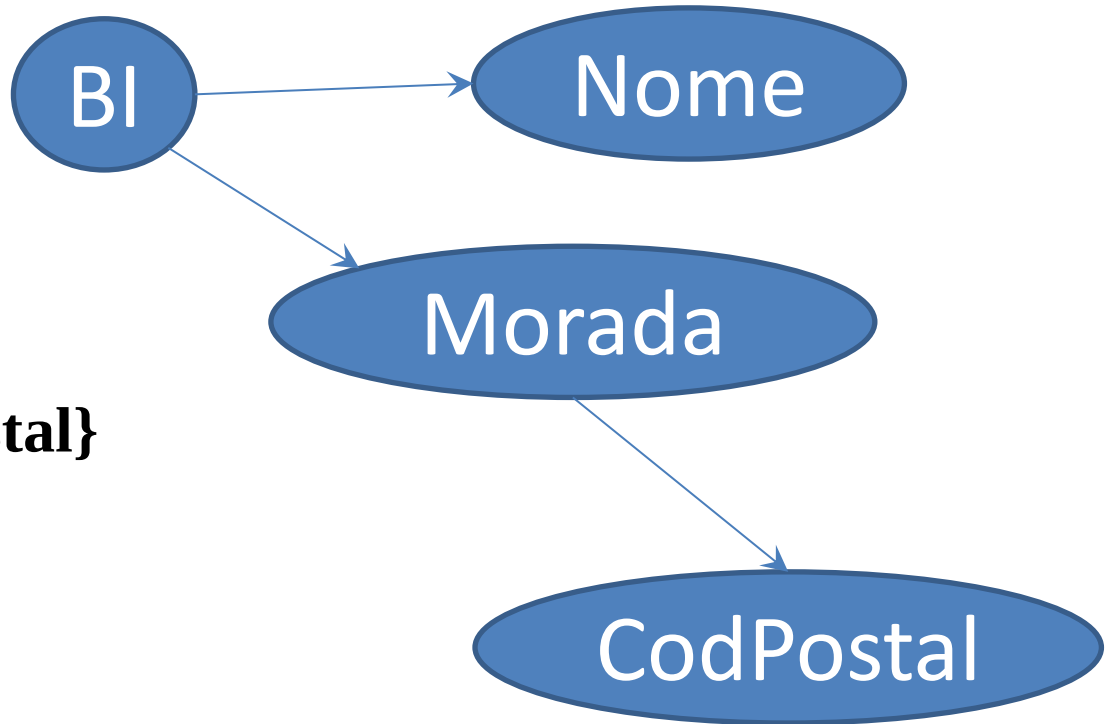
(*) Sempre que for necessário indicar o conjunto de dependências funcionais F sobre o qual se calcula X^+ , escreveremos X^+_F

Fecho de um Conjunto de Atributos

$F = \{$
 $\{BI\} \rightarrow \{Nome\},$
 $\{BI\} \rightarrow \{Morada\},$
 $\{Morada\} \rightarrow \{CodPostal\}$
 $\}$

$X = \{BI\}$

$X^+ = \{BI, Nome, Morada, CodPostal\}$



Dependências Funcionais - chaves

Seja $R(A_1, \dots, A_n)$ um esquema de relação, F o seu conjunto de dependências funcionais e X um subconjunto de $A_1, \dots, A_n$.

Diz-se que **X é uma super-chave de R** se:

Para todo o A_i , tem-se $F \models X \rightarrow A_i$.

Ou seja, a dependência de todos os atributos relativamente a X é dada ou segue logicamente das dependências dadas

(também se pode dizer que $X \rightarrow A_i \in F^+$,

ou que $A_i \in X^+$ calculado relativamente a F).

Diz-se que **X é uma chave candidata de R** se,

para além da condição anterior, (i.e. ser superchave) se verificar:

Qualquer que seja $Y \subset X$, pelo menos uma das dependências $Y \rightarrow A_1, \dots, Y \rightarrow A_n$ não pertence a F^+ (ou, pelo menos um dos atributos $A_1, \dots, A_n$ não pertence a Y^+).

Ou seja, **as chaves são conjuntos mínimos de atributos**.

Exemplo

R(NIF, NumCC, NumAluno, Nome, Nota)

SuperChaves = {

- {NIF}, {NumCC}, {NumAluno},
- {NIF, NumCC},
- {NIF, NumCC, NumAluno},
- {NumCC, NumAluno},
- {NIF, Nome},
- {NIF, NumCC, Nome},
- {NIF, NumCC, NumAluno, Nome},
- {NumCC, Nome},
- {NumCC, NumAluno, Nome},
- {NumAluno, Nome}

.....

}

Chave Candidatas = {
 {NIF},
 {NumCC},
 {NumAluno}
}

Determinar as Chaves Candidatas

Determinar quais as chaves candidatas para um esquema de relação R , dado o conjunto de dependências funcionais F associado:

Testam-se **todos os subconjuntos (X) de R** com um único atributo.

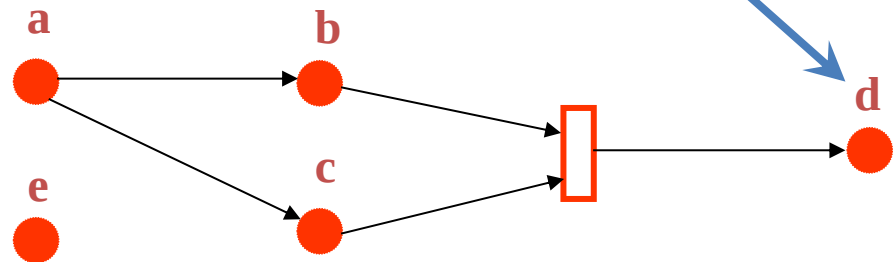
Todos os X **que verifiquem $X^+ = R$ são chaves candidatas.**

Repete-se, sucessivamente, o processo com os subconjuntos (X) de R com 2, 3, ... n elementos que não incluam as chaves já obtidas
São chaves candidatas todos os subconjuntos tais que $X^+ = R$.

Chaves Candidatas – através da análise dos grafos de dependências funcionais

$R(a, b, c, d, e)$
 $F = \{$
 $\{a\} \rightarrow \{b\},$
 $\{a\} \rightarrow \{c\},$
 $\{b, c\} \rightarrow \{d\}$
 $\}$

*É destino,
mas não é origem de qualquer DF,
logo não pertence a uma chave*



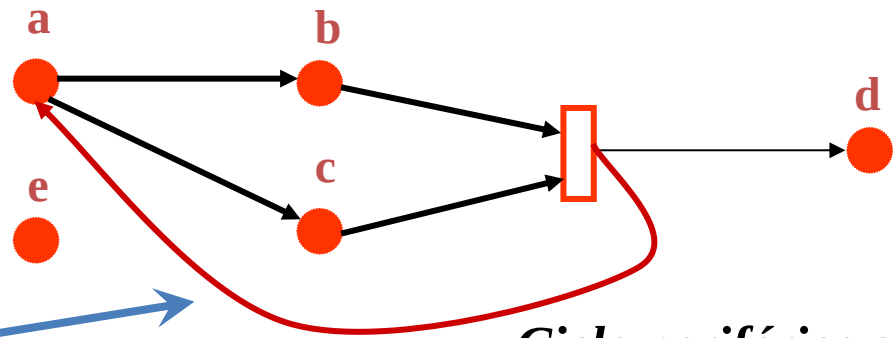
*Não são destinos de qualquer DF,
logo tem de pertencer às chaves*

Chaves Candidatas $\{\{a, e\}\}$

Chaves Candidatas – análise dos grafos

R(a, b, c, d, e)
F = {
 {a} → {b},
 {a} → {c},
 {b, c} → {a, d}
}

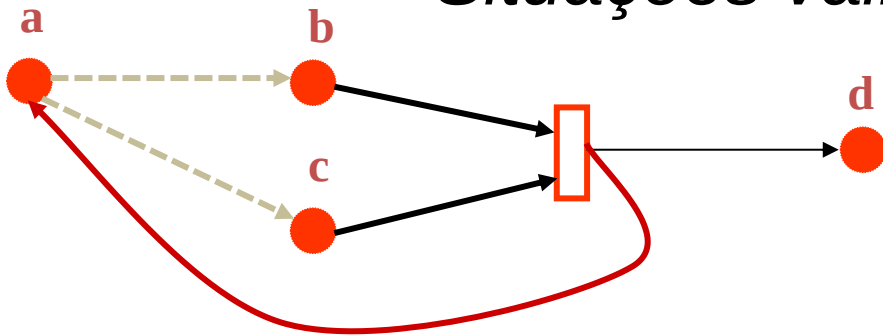
Chaves Candidatas { {a, e}, {b, c, e} }



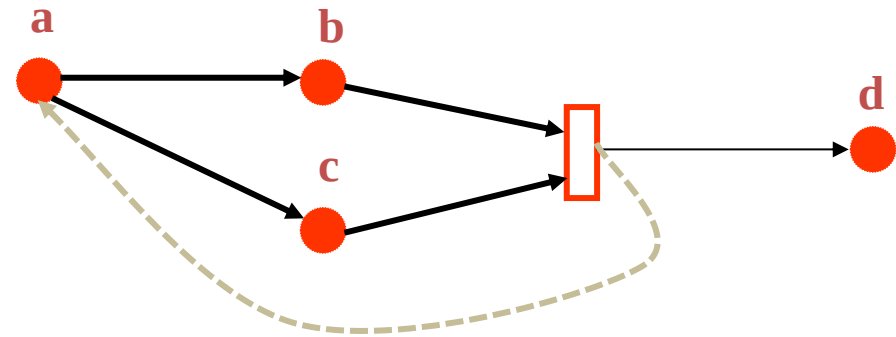
Ciclo periférico sugere mais do que uma chave candidata

Para cada nó do ciclo, se eliminarmos todos os arcos à sua saída, pelo menos um atributo deixa de ser destino de qualquer arco. Cada um dos nós (simples ou composto) envolvido no ciclo pertence a uma chave candidata

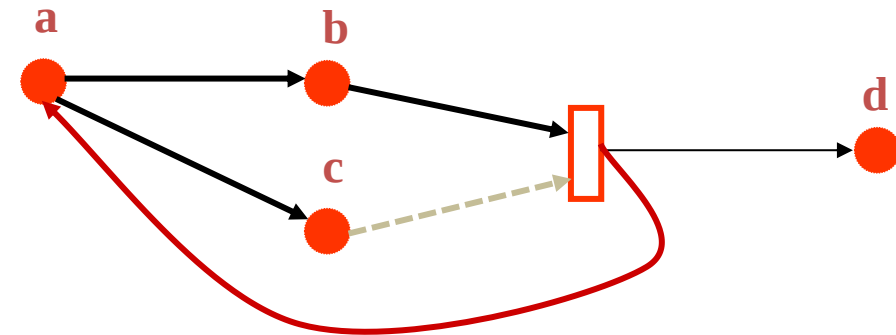
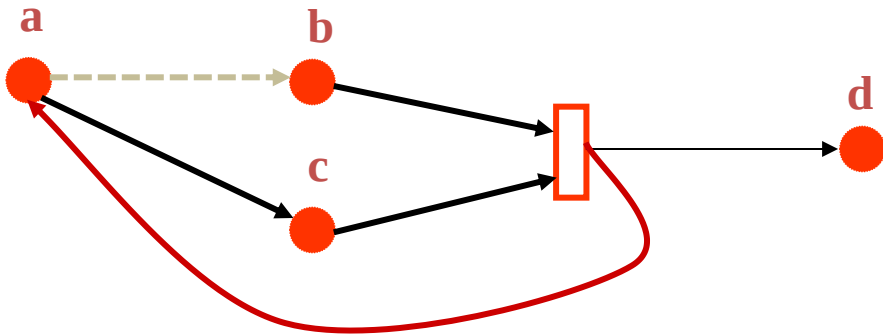
Situações válidas de corte



Chave Candidata $\{b, c\}$



Chave Candidata $\{a\}$



Situações inválidas de corte

1 - Calcular as Chaves Candidatas

$R(a, b, c, d)$

$F = \{$

$\{a\} \rightarrow \{b\},$

$\{a\} \rightarrow \{c\},$

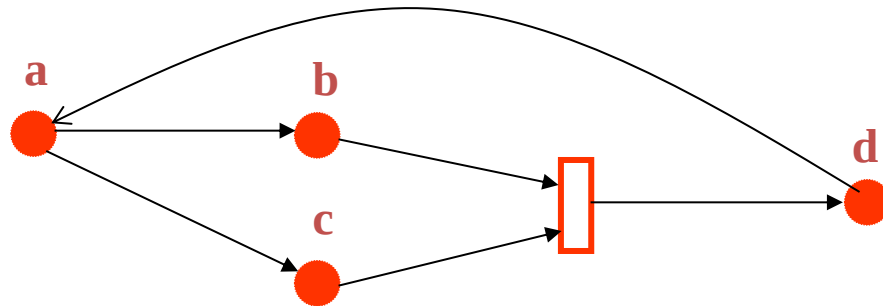
$\{d\} \rightarrow \{a\},$

$\{b, c\} \rightarrow \{d\}$

$\}$

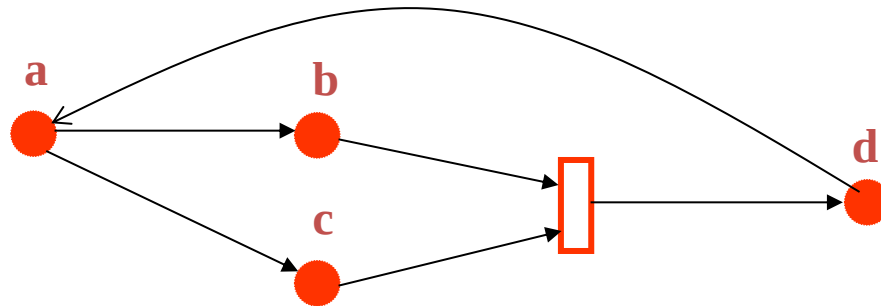
1 - Calcular as Chaves Candidatas

$R(a, b, c, d)$
 $F = \{$
 $\{a\} \rightarrow \{b\},$
 $\{a\} \rightarrow \{c\},$
 $\{d\} \rightarrow \{a\},$
 $\{b, c\} \rightarrow \{d\}$
 $\}$



1 - Calcular as Chaves Candidatas

$R(a, b, c, d)$
 $F = \{$
 $\{a\} \rightarrow \{b\},$
 $\{a\} \rightarrow \{c\},$
 $\{d\} \rightarrow \{a\},$
 $\{b, c\} \rightarrow \{d\}$
 $\}$



Chaves Candidatas $\{\{a\}, \{d\}, \{b, c\}\}$

2 - Calcule as Chaves Candidatas

$R(a, b, c)$

**$F = \{$
 $\{c\} \rightarrow \{a\},$
 $\{b, a\} \rightarrow \{c\}$
 $\}$**

2 - Calcular as Chaves Candidatas

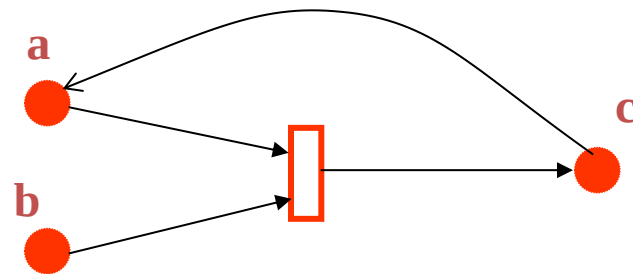
$R(a, b, c)$

$F = \{$

$\{c\} \rightarrow \{a\},$

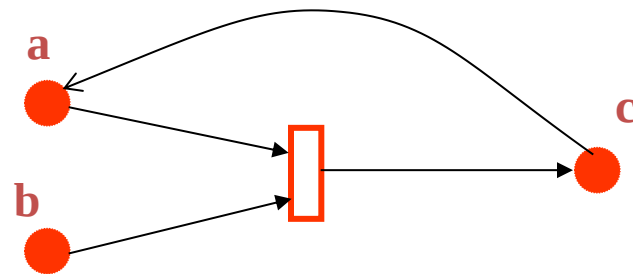
$\{b, a\} \rightarrow \{c\}$

$\}$



2 - Calcular as Chaves Candidatas

$R(a, b, c)$
 $F = \{$
 $\{c\} \rightarrow \{a\},$
 $\{b, a\} \rightarrow \{c\}$
 $\}$



Chaves Candidatas $\{\{a, b\}, \{c, b\}\}$

3 - Calcular as Chaves Candidatas

$R(a, b, c, d, e, f)$

$F = \{$

$\{c\} \rightarrow \{a\},$

$\{c\} \rightarrow \{d\},$

$\{e\} \rightarrow \{f\},$

$\{f\} \rightarrow \{e\},$

$\{a, b\} \rightarrow \{c\}$

$\}$

3 - Calcular as Chaves Candidatas

$R(a, b, c, d, e, f)$

$F = \{$

$\{c\} \rightarrow \{a\},$

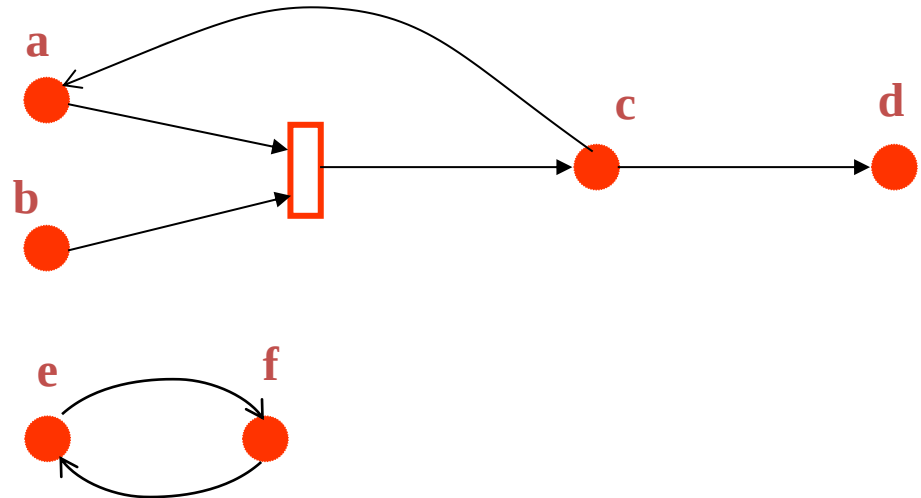
$\{c\} \rightarrow \{d\},$

$\{e\} \rightarrow \{f\},$

$\{f\} \rightarrow \{e\},$

$\{a, b\} \rightarrow \{c\}$

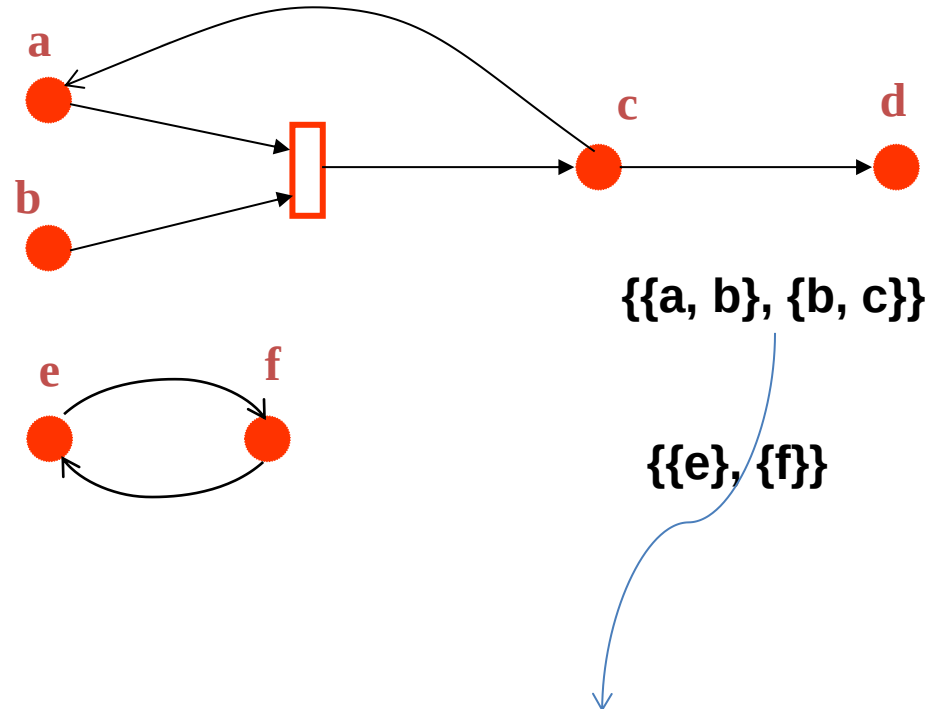
$\}$



3 - Calcular as Chaves Candidatas

$R(a, b, c, d, e, f)$

$F = \{$
 $\{c\} \rightarrow \{a\},$
 $\{c\} \rightarrow \{d\},$
 $\{e\} \rightarrow \{f\},$
 $\{f\} \rightarrow \{e\},$
 $\{a, b\} \rightarrow \{c\}$
 $\}$



Chaves Candidatas $\{\{a, b, e\}, \{a, b, f\}, \{b, c, e\}, \{b, c, f\}\}$

4 - Calcular as Chaves Candidatas

$R(a, b, c, d, e)$

$F = \{$

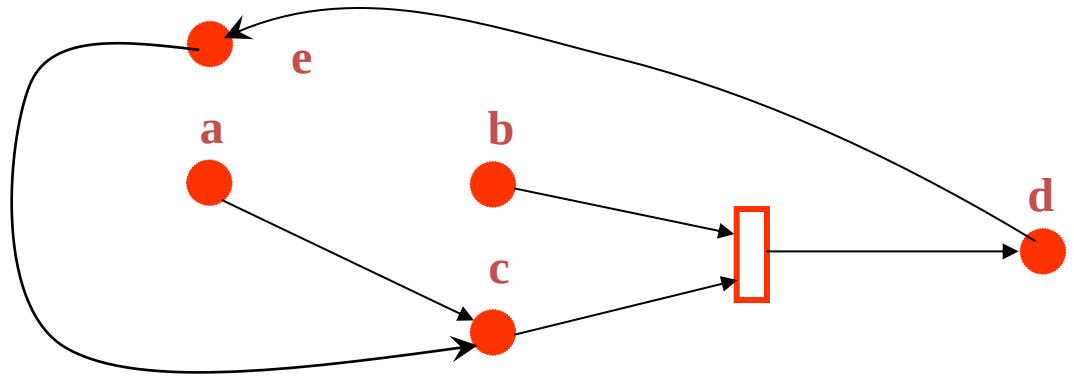
$\{a\} \rightarrow \{c\},$

$\{b, c\} \rightarrow \{d\},$

$\{d\} \rightarrow \{e\},$

$\{e\} \rightarrow \{c\}$

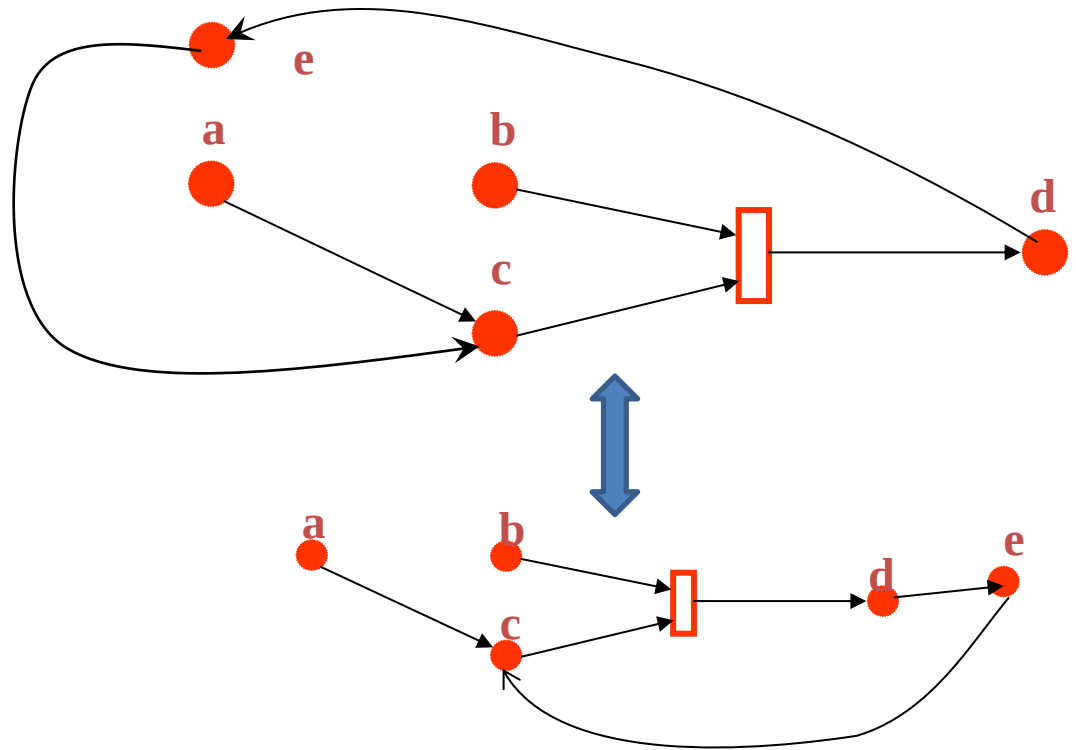
$\}$



4 - Calcule as Chaves Candidatas

$R(a, b, c, d, e)$

$F = \{$
 $\{a\} \rightarrow \{c\},$
 $\{b, c\} \rightarrow \{d\},$
 $\{d\} \rightarrow \{e\},$
 $\{e\} \rightarrow \{c\}$
 $\}$



Colocar numa forma fácil de entender
Dependencia Funcional com o sentido $\rightarrow$

4 - Calcule as Chaves Candidatas

R(a, b, c, d, e)

F = {

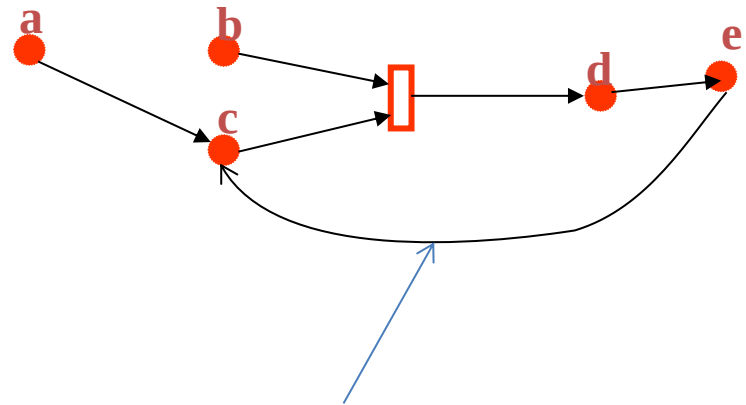
{a}→{c},

{b, c} →{d},

{d}→{e},

{e}→{c}

}



Não é arco periférico!

Chaves Candidatas {{a, b}}

5 - Calcular as Chaves Candidatas

R(a, b, c, d, e)

F = {

{a} → {c},

{d} → {e},

{e} → {a},

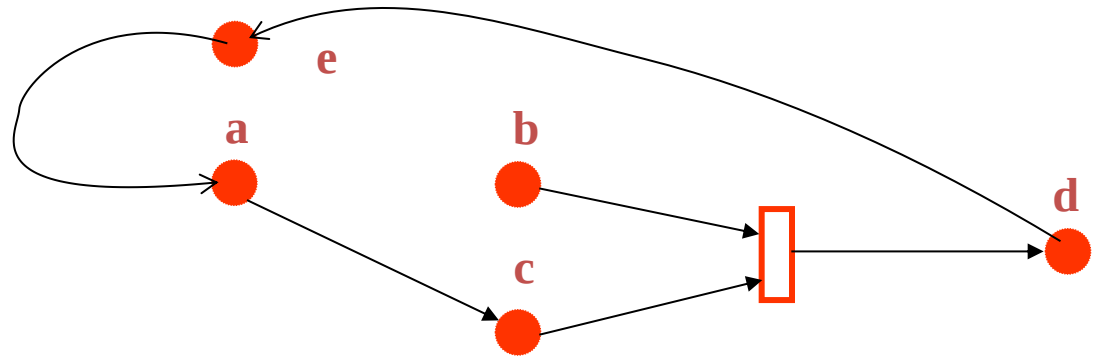
{c, b} → {d}

}

5 - Calcular as Chaves Candidatas

$R(a, b, c, d, e)$

$F = \{$
 $\{a\} \rightarrow \{c\},$
 $\{d\} \rightarrow \{e\},$
 $\{e\} \rightarrow \{a\},$
 $\{c, b\} \rightarrow \{d\}$
 $\}$



5 - Calcular as Chaves Candidatas

$R(a, b, c, d, e)$

$F = \{$

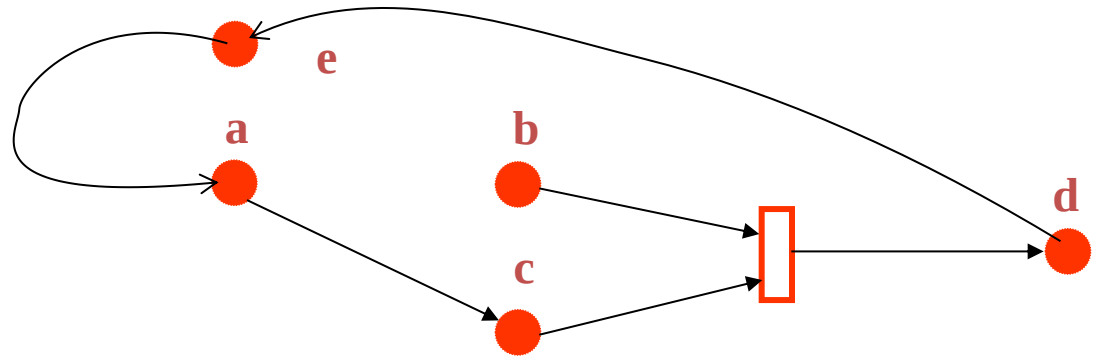
$\{a\} \rightarrow \{c\},$

$\{d\} \rightarrow \{e\},$

$\{e\} \rightarrow \{a\},$

$\{c, b\} \rightarrow \{d\}$

$\}$



Chaves Candidatas $\{\{a,b\}, \{d,b\}, \{c,b\}, \{e,b\}\}$

6 - Calcule as Chaves Candidatas

$R(a, b, c, d)$ $F = \{\}$

6 - Calcular as Chaves Candidatas

$R(a, b, c, d)$ $F = \{\}$

a ●
b ●
c ●
d ●

6 - Calcule as Chaves Candidatas

R(a, b, c, d)

F = {}

a ●

b ●

c ●

d ●

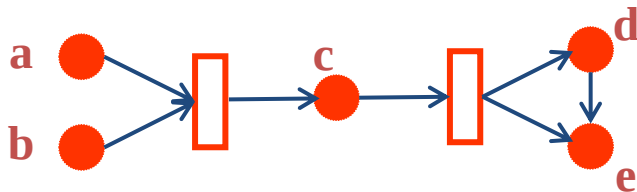
Chaves Candidatas { {a, b, c, d} }

7 - Mostre que os conjuntos de dependências funcionais seguintes são equivalentes:

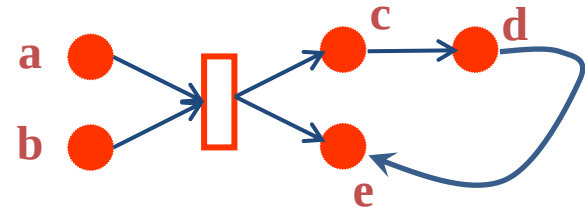
$F1 = \{\{a,b\} \rightarrow \{c\}, \{d\} \rightarrow \{e\}, \{c\} \rightarrow \{d,e\}\}$

$F2 = \{\{a,b\} \rightarrow \{c,e\}, \{d\} \rightarrow \{e\}, \{c\} \rightarrow \{d\}\}$

F1



F2



7 – Método 1 (através do Fecho)

Gerar explicitamente $F1^+$ e $F2^+$ e mostrar que $F1^+ = F2^+$

$$F1 = \{\{a,b\} \rightarrow \{c\}, \{d\} \rightarrow \{e\}, \{c\} \rightarrow \{d,e\}\}$$

$$F2 = \{\{a,b\} \rightarrow \{c,e\}, \{d\} \rightarrow \{e\}, \{c\} \rightarrow \{d\}\}$$

$$F1^+ = \{\{a,b\} \rightarrow \{c\}, \{d\} \rightarrow \{e\}, \{c\} \rightarrow \{d,e\}, \{a,b\} \rightarrow \{d,e\}, \{a,b\} \rightarrow \{e\}, \\ \{a,b\} \rightarrow \{d\}, \{a,b\} \rightarrow \{c,d,e\}, \{a,b\} \rightarrow \{c,e\}, \{a,b\} \rightarrow \{c,d\}, \{c\} \rightarrow \{d\}, \\ \{c\} \rightarrow \{e\}, \dots\}$$

$$F2^+ = \{\{a,b\} \rightarrow \{c,e\}, \{d\} \rightarrow \{e\}, \{c\} \rightarrow \{d\}, \{c\} \rightarrow \{e\}, \{c\} \rightarrow \{d,e\}, \{a,b\} \rightarrow \{e\}, \\ \{a,b\} \rightarrow \{c\}, \{a,b\} \rightarrow \{d\}, \{a,b\} \rightarrow \{c,d,e\}, \{a,b\} \rightarrow \{c,d\}, \{a,b\} \rightarrow \{d,e\}, \dots\}$$

Os “...” correspondem às DFs restantes.

MUITO COMPLEXO!!!

7 – Método 2 (através da cobertura)

Mostrar que $F1 \models F2$ e que $F2 \models F1$

$$F1 = \{ \{a,b\} \rightarrow \{c\}, \{d\} \rightarrow \{e\}, \{c\} \rightarrow \{d,e\} \}$$

$$F2 = \{ \{a,b\} \rightarrow \{c,e\}, \{d\} \rightarrow \{e\}, \{c\} \rightarrow \{d\} \}$$

$F2 \models F1$: Para cada DF $X \rightarrow Y$ em $F1$ mostrar que $F2 \models X \rightarrow Y$.

Podemos fazê-lo mostrando que $Y \subseteq X^+$, calculado relativamente a $F2$:

$$F2 \models \{a,b\} \rightarrow \{c\} \text{ porque: } \{a,b\}_{F2}^+ = \{a,b,c,e,d\} \supseteq \{c\}$$

$$F2 \models \{d\} \rightarrow \{e\} \text{ porque: } \{d\}_{F2}^+ = \{d,e\} \supseteq \{e\}$$

$$F2 \models \{c\} \rightarrow \{d,e\} \text{ porque: } \{c\}_{F2}^+ = \{c,d,e\} \supseteq \{d,e\}$$

$F1 \models F2$: Para cada DF $X \rightarrow Y$ em $F2$ mostrar que $F1 \models X \rightarrow Y$.

Podemos fazê-lo mostrando que $Y \subseteq X^+$, calculado relativamente a $F1$:

$$F1 \models \{a,b\} \rightarrow \{c,e\} \text{ porque: } \{a,b\}_{F1}^+ = \{a,b,c,d,e\} \supseteq \{c,e\}$$

$$F1 \models \{d\} \rightarrow \{e\} \text{ porque: } \{d\}_{F1}^+ = \{d,e\} \supseteq \{e\}$$

$$F1 \models \{c\} \rightarrow \{d\} \text{ porque: } \{c\}_{F1}^+ = \{c,d,e\} \supseteq \{d\}$$

8 - Calcular as Chaves Candidatas

$R(a,b,c,d)$

$F = \{$

$\{c\} \rightarrow \{a,b\},$

$\{a,b\} \rightarrow \{c\},$

$\{c\} \rightarrow \{d\}$

$\}$

8 - Calcular as Chaves Candidatas

$R(a,b,c,d)$

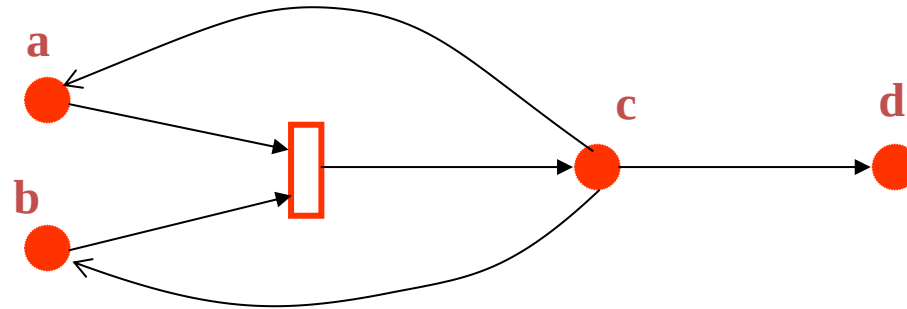
$F = \{$

$\{c\} \rightarrow \{a,b\},$

$\{a,b\} \rightarrow \{c\},$

$\{c\} \rightarrow \{d\}$

$\}$



$R(a,b,c,d)$

$F = \{$

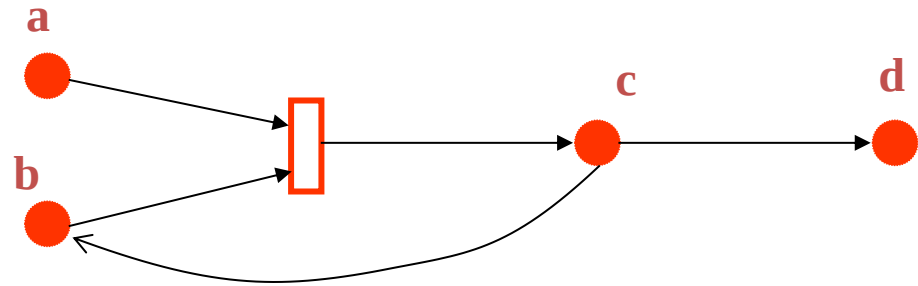
$\{c\} \rightarrow \{a,b\},$

$\{a,b\} \rightarrow \{c\},$

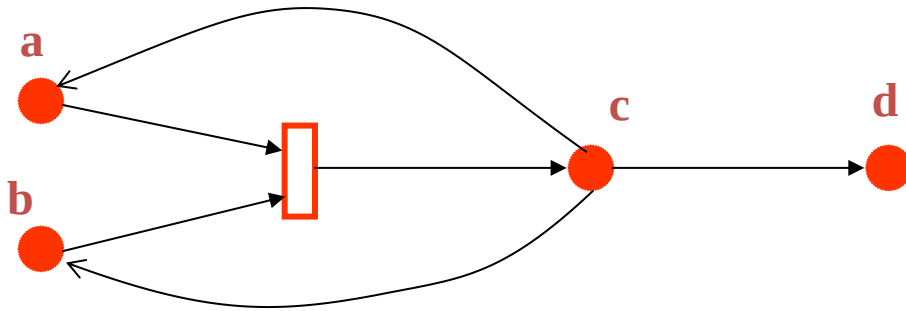
$\{c\} \rightarrow \{d\}$

$\}$

8 - Calcular as Chaves

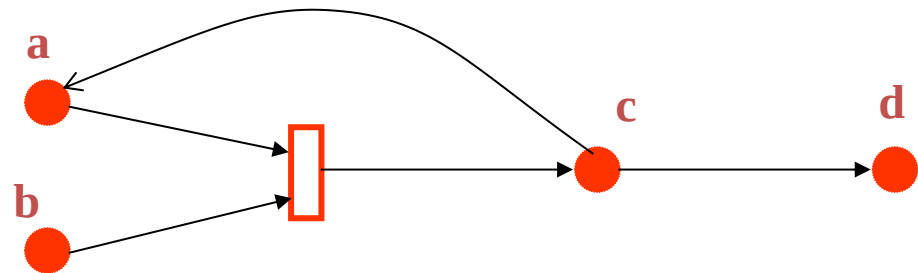


Chaves $\{ \{a, b\}, \{a, c\} \}$



Chaves $\{ \{c\} \}$

União de dois ciclos separados



Chaves $\{ \{a, b\}, \{b, c\} \}$

(União) Chaves $\{ \{a, b\}, \{a, c\}, \{b, c\}, \{c\} \}$

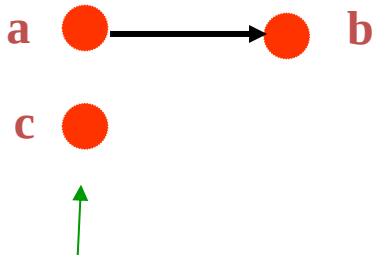
Chaves Candidatas $\{ \{a, b\}, \{c\} \}$

SuperChaves

9 – Exercício

Dado os esquema de relação $R(a,b,c)$

e o conjunto de dependências funcionais $F = \{\{a\} \rightarrow \{b\}\}$,
determine quais são as chaves candidatas de R



*Não são destino de qualquer DF,
logo têm de pertencer às chaves*

$$\{a,c\}^+ = \{a,c,b\} = R$$

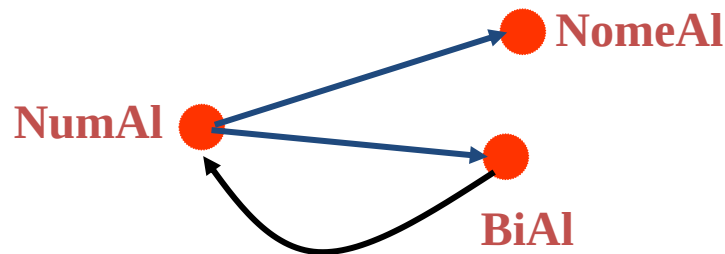
Chaves Candidatas $\{\{a,c\}\}$

10 - Exercício

Dado os esquema de relação Aluno(NnumAl, NomeAl, BiAl) e o conjunto de dependências funcionais

$F = \{\{NumAl\} \rightarrow \{NomeAl, BiAl\}, \{BiAl\} \rightarrow \{NumAl\}\}$,

determine quais são as chaves candidatas de Aluno



O ciclo do grafo de DFs sugere a existência de duas chaves candidatas.

$\{NumAl\}^+ = \{NumAl, NomeAl, BiAl\} = Aluno$

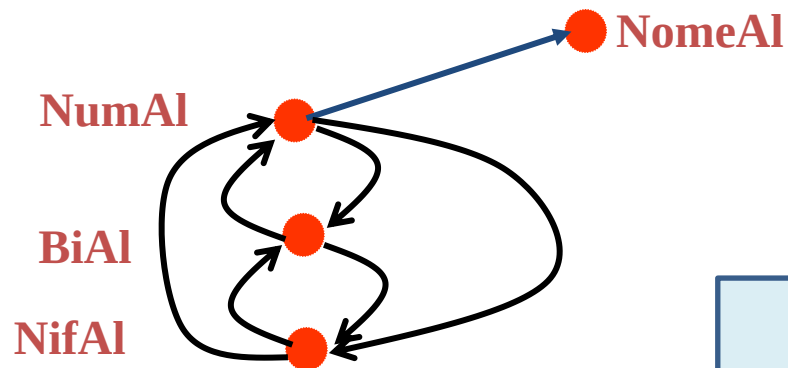
$\{BiAl\}^+ = \{BiAl, NumAl, NomeAl\} = Aluno$

Chaves Candidatas $\{\{NumAl\}, \{BiAl\}\}$

11 - Exercício

Dado os esquema de relação Alunos(NnumAl, NomeAl ,BiAl) e o conjunto de dependências funcionais

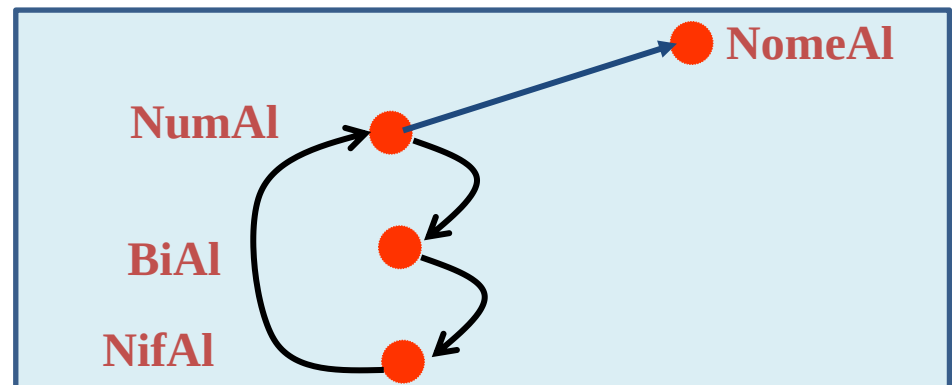
$F = \{ \{ \text{NumAl} \} \rightarrow \{ \text{NomeAl}, \text{BiAl}, \text{NifAl} \}, \{ \text{BiAl} \} \rightarrow \{ \text{NumAl}, \text{NifAl} \}, \{ \text{NifAl} \} \rightarrow \{ \text{NumAl}, \text{BiAl} \} \}$,
determine quais são as chaves candidatas de Alunos.



Chaves Candidatas
 $\{ \{ \text{NumAl} \}, \{ \text{BiAl} \}, \{ \text{NifAl} \} \}$

se os conjuntos de dependências funcionais tiver redundância, pode ficar complicado.

Uma cobertura mínima de F



Cobertura Mínima

Um conjunto F de dependências funcionais é mínimo se:

Todas as dependências de F têm apenas um atributo à sua direita

Nenhuma dependência $X \rightarrow A$ em F pode ser substituído por $Y \rightarrow A$ onde Y é um subconjunto próprio de X , e manter a equivalência com F .

Nenhuma das dependências de F pode ser removida, mantendo a equivalência com F

A **cobertura mínima** de um conjunto de dependências funcionais F é um conjunto mínimo de dependências funcionais de F equivalente a F

Cobertura Mínima - Algoritmo

Dado um conjunto de dependências funcionais F , determinar uma das suas coberturas mínimas (G)

Procedimento:

$G = F$

(tornar singulares as DFs)

Substituir em G toda a dependência funcional $X \rightarrow \{A_1, \dots, A_k\}$
pelas k dependências funcionais $X \rightarrow A_1, \dots, X \rightarrow A_k$.

(eliminar atributos redundantes)

Para cada dependência funcional $X \rightarrow A$ em G (1)

Para cada atributo $B \in X$

Se $((G - \{X \rightarrow A\}) \cup \{(X - \{B\}) \rightarrow A\})$ for equivalente a G ,
então substituir em G $X \rightarrow A$ por $(X - \{B\}) \rightarrow A$.

(Eliminar DFs redundantes)

Para cada uma das dependências funcionais $X \rightarrow A$ restantes em G

Se $(G - \{X \rightarrow A\})$ for equivalente a G (ou seja, $(G - \{X \rightarrow A\}) \models X \rightarrow A$),
então remover $X \rightarrow A$ de G

(1) Se $G' = ((G - \{X \rightarrow A\}) \cup \{(X - \{B\}) \rightarrow A\})$, temos de mostrar que $G' \models G$ (trivial) e que $G \models G'$, mas para isto basta mostrar que $G \models (X - \{B\}) \rightarrow A$

Decomposição

Dado um esquema de relação R ,
diz-se que o conjunto de esquemas de relação

$\rho = \{R_1, \dots, R_n\}$ é uma decomposição de R
sse $R = R_1 \cup R_2 \dots \cup R_n$.

Cada R_i é uma projeção de R sobre um subconjunto dos seus atributos.

Decomposição - Exemplo

Sejam:

$R(A,B,C,D)$,

$R1(A,B) = \pi_{\{A,B\}}(R)$,

$R2(C,D) = \pi_{\{C,D\}}(R)$

Então $\rho = \{R1, R2\}$ é uma decomposição de R

Decomposições sem perda por junção

Uma decomposição de um esquema de relação R
diz-se sem perda por junção
(“*lossless join decomposition*”), ou não aditiva,
se for a decomposição de R
no conjunto $\{R_1, \dots, R_n\}$ de projecções,
tais que, para qualquer instância r de R, se tem:

$$r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \dots \bowtie \pi_{R_n}(r)$$

Lossless Join - Exemplo

Sejam:

Inscrições(**Número**,**CodDisc**,DescrDisc,Nome)

$F = \{$
 {Número} $\rightarrow$ {Nome},
 {CodDisc} $\rightarrow$ {DescrDisc}
 $\}$

A decomposição de Inscrições em:

R1(**Número**, **CodDisc**, DescrDisc)

R2(DescrDisc, Nome)

Não é Lossless

A junção natural destas duas relações produz tuplos não existentes originalmente em Inscrições

Número	CodDisc	Nome	DescrDisc
1670	SI	Rui	Sist. Inf.
2577	SI	Ana	Sist. Inf.

Inscrições

Número	CodDisc	DescrDisc
1670	SI	Sist. Inf.
2577	SI	Sist. Inf.

R1



R2

DescrDisc	Nome
Sist. Inf.	Rui
Sist. Inf.	Ana

Número	CodDisc	Nome	DescrDisc
1670	SI	Rui	Sist. Inf.
1670	SI	Ana	Sist. Inf.
2577	SI	Rui	Sist. Inf.
2577	SI	Ana	Sist. Inf.

< > Inscrições

Lossless Join - Exemplo

Sejam:

Inscrições(**Número**,**CodDisc**,DescrDisc,Nome)

$F = \{$
 {Número} $\rightarrow$ {Nome},
 {CodDisc} $\rightarrow$ {DescrDisc}
 $\}$



Chave estrangeira?

A decomposição de Inscrições em: 

R1(**Número**, **CodDisc**, Nome)

R2(**CodDisc**, DescrDisc)

É Lossless

A junção natural destas duas relações é lossless

Número	CodDisc	Nome	DescrDisc
1670	SI	Rui	Sist. Inf.
2577	SI	Ana	Sist. Inf.

Inscrições

Número	CodDisc	Nome
1670	SI	Rui
2577	SI	Ana

R1



R2

CodDisc	DescrDisc
SI	Sist. Inf.

Número	CodDisc	Nome	DescrDisc
1670	SI	Rui	Sist. Inf.
2577	SI	Ana	Sist. Inf.

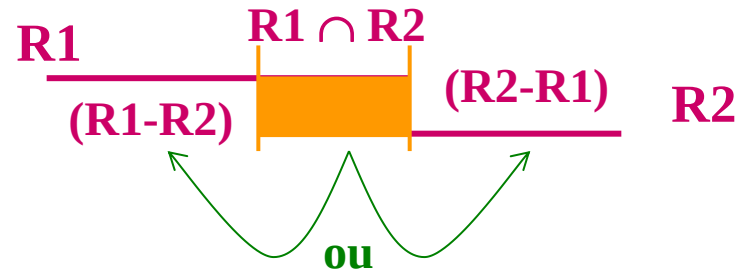
= Inscrições

Lossless Join

Decomposições sem perda:

Se uma decomposição de R apenas contém dois esquemas de relação R1 e R2, então a decomposição é *lossless join* relativamente a um conjunto de DFs F, se se verificar, pelo menos, uma das seguintes condições:

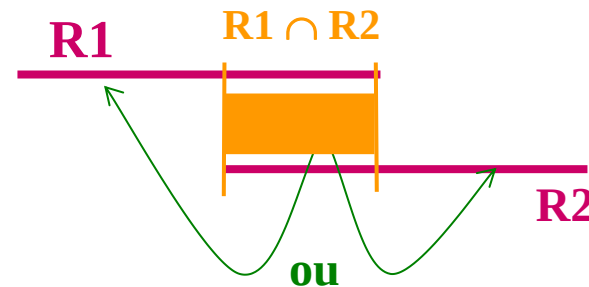
1. $(R1 \cap R2) \rightarrow (R1 - R2) \in F^+$
2. $(R1 \cap R2) \rightarrow (R2 - R1) \in F^+$



Lossless Join

se a decomposição é *lossless join*, também se verifica uma das condições:

1. $(R1 \cap R2) \rightarrow R1 \in F^+$
2. $(R1 \cap R2) \rightarrow R2 \in F^+$



Projecção de dependências funcionais

Dado um conjunto de dependências funcionais F ,
a sua projecção sobre um conjunto de atributos Z
é o conjunto das dependências funcionais de F^+
que apenas contém atributos existentes em Z :

$$\pi_Z(F) = \{X \rightarrow A \mid X \rightarrow A \in F^+ \text{ e } (X \cup A) \subseteq Z\}$$

Projectão de Dependências Funcionais - Exemplo

Sejam $R(A,B,C,D)$

$F = \{$

$\{B\} \rightarrow \{A\},$

$\{C\} \rightarrow \{B, D\}$

$\}$

e $R_1(A,B,C)$.

A projectão de F sobre os atributos de R_1 é

$$\pi_{R_1}(F) = \{ \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{B\} \}$$

Projectão de Dependências Funcionais - Exemplo

Sejam $R(A,B,C,D)$

$F = \{$

$\{A\} \rightarrow \{B\},$

$\{B\} \rightarrow \{C\}$

$\}$

e $R_1(A, C)$.

A projectão de F sobre os atributos de R_1 é

$$\pi_{R_1}(F) = \{ \{A\} \rightarrow \{C\} \}$$

Preservação das Dependências Funcionais

Diz-se que uma decomposição de um esquema de relação R preserva as dependências funcionais

se a decomposição de R no conjunto $\{R_1, \dots, R_n\}$ de projecções, é tal que o conjunto das dependências funcionais de R seja equivalente à reunião das projecções das dependências funcionais dos vários R_i , ou seja:

$$(\pi_{R_1}(F) \cup \dots \cup \pi_{R_n}(F))^+ = F^+.$$

Preservação das Dependências Funcionais - Exemplo

Sejam $R(A,B,C,D)$ e $F = \{ \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{B,D\} \}$
e a decomposição $\rho = \{ R1(A,B,C), R2(C,D) \}$.

Temos:

$$\pi_{R1}(F) = \{ \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{B\} \},$$

$$\pi_{R2}(F) = \{ \{C\} \rightarrow \{D\} \}$$

É fácil verificar $(\pi_{R1}(F) \cup \pi_{R2}(F))^+ = \{ \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{B,D\}, \dots \} = F^+$

ρ = preserva as DFs

Preservação das Dependências Funcionais - Exemplo

Sejam $R(A,B,C,D)$ e $F = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}\}$
e a decomposição $\rho = \{R_1(A,C), R_2(B,C,D)\}$.

Temos:

$$\pi_{R_1}(F) = \{\{A\} \rightarrow \{C\}\} \text{ e } \pi_{R_2}(F) = \{\{B\} \rightarrow \{C\}\} \quad \text{Perdeu-se } \{A\} \rightarrow \{B\}$$

É fácil verificar $(\pi_{R_1}(F) \cup \pi_{R_2}(F))^+ = \{\{A\} \rightarrow \{C\}, \{B\} \rightarrow \{C\}, \dots\} \neq F^+$

$\rho = \text{n\~ao preserva as DFs}$

Mas se $\rho_2 = \{R_1(A,B), R_2(B,C,D)\}$, temos:

$$\pi_{R_1}(F) = \{\{A\} \rightarrow \{B\}\} \text{ e } \pi_{R_2}(F) = \{\{B\} \rightarrow \{C\}\}$$

É fácil verificar $(\pi_{R_1}(F) \cup \pi_{R_2}(F))^+ = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \dots\} = F^+$

$\rho_2 = \text{preserva as DFs}$

Preservação das Dependências Funcionais - Exemplo



A	B	C
1	3	a
2	4	b

$F = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}\}$



A	C
1	a
2	b

$\pi_{R1}(F) = \{\{A\} \rightarrow \{C\}\}$



B	C
3	a
4	b

$\pi_{R2}(F) = \{\{B\} \rightarrow \{C\}\}$

Acrescentar {5,a} a R2
Irá provocar uma anomalia não detectada



A	B	C
1	3	a
2	4	b
1	5	a



A	C
1	a
2	b

B	C
3	a
4	b
5	a



O modelo não permite
impedir a associação
simultânea destes tuplos

Perdeu-se $\{A\} \rightarrow \{B\}$

Preservação das Dependências Funcionais - Exemplo



A	B	C
1	3	a
2	4	b

$F = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}\}$



A	B
1	3
2	4

$\pi_{R1}(F) = \{\{A\} \rightarrow \{C\}\}$



B	C
3	a
4	b

$\pi_{R2}(F) = \{\{B\} \rightarrow \{C\}\}$

Não é possível acrescentar {5,a} a R2
Porque a anomalia é detectada



A	B	C
1	3	a
2	4	b



A	B
1	3
2	4
1	5

B	C
3	a
4	b
5	a

O modelo consegue
impedir a associação
simultânea destes tuplos

Mantiveram-se $\{A\} \rightarrow \{B\}$ e $\{B\} \rightarrow \{C\}$

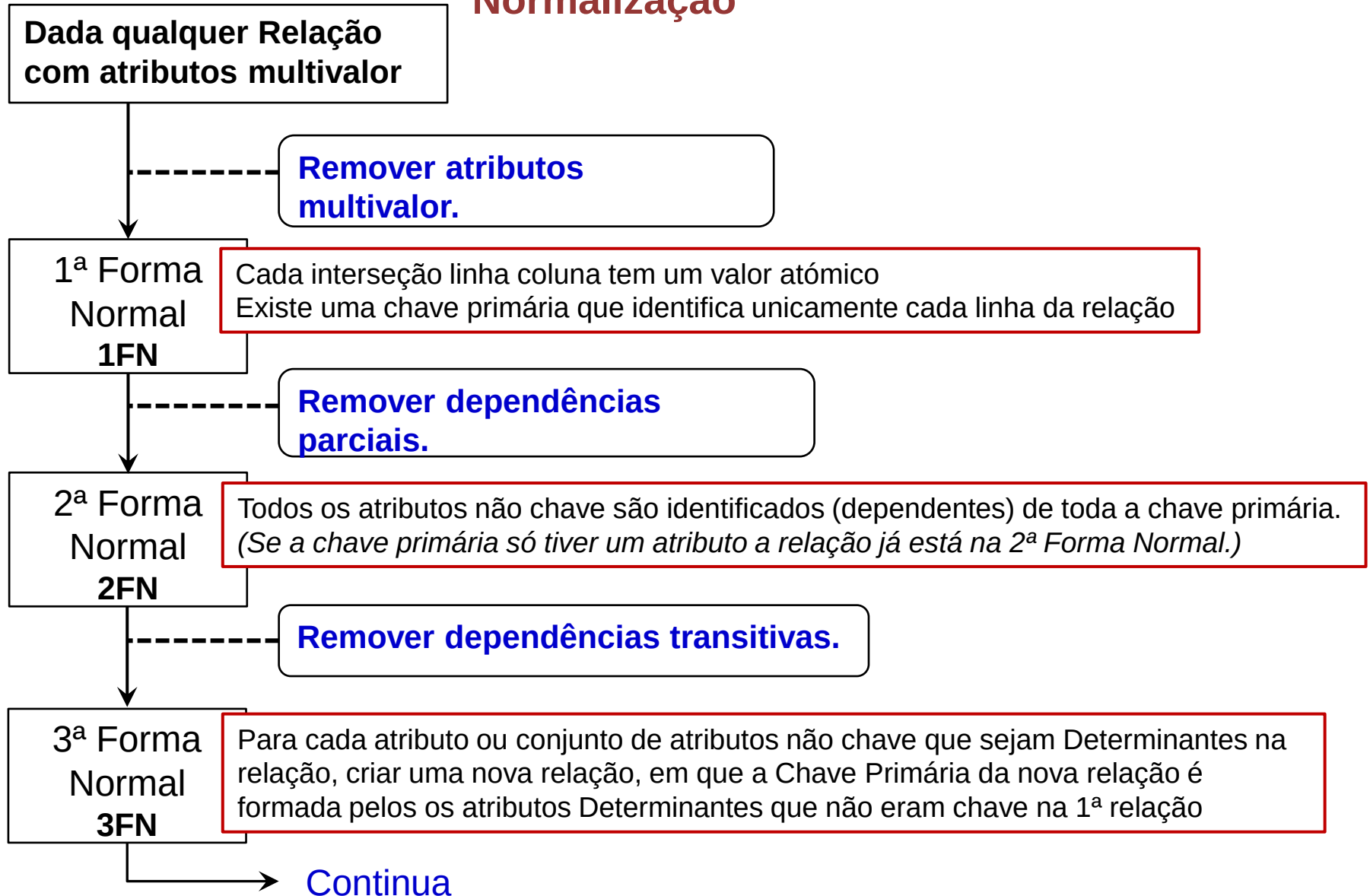
Normalização

TABLE
5.2

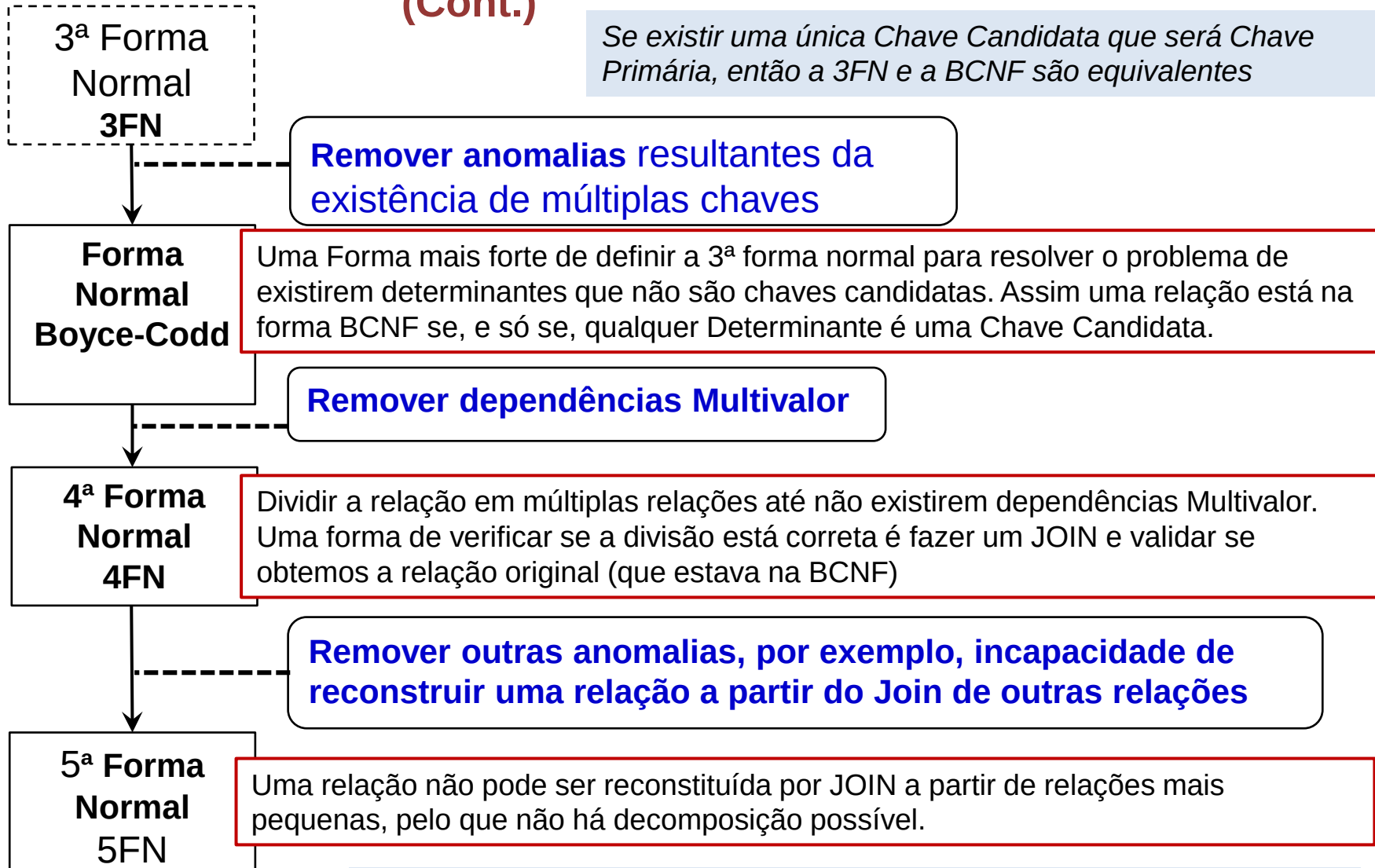
Normal Forms

NORMAL FORM	CHARACTERISTIC
First normal form (1NF)	Table format, no repeating groups, and PK identified
Second normal form (2NF)	1NF and no partial dependencies
Third normal form (3NF)	2NF and no transitive dependencies
Boyce-Codd normal form (BCNF)	Every determinant is a candidate key (special case of 3NF)
Fourth normal form (4NF)	3NF and no independent multivalued dependencies

Passos da Normalização



Passos da Normalização (Cont.)



Uma relação que esteja na 3FN e só tiver uma chave simples está na 5FN.

Noção de Normalização

Para o armazenamento de informação acerca de fornecedores e produtos, definiu-se a seguinte Relação R:

CodigoFornecedor	Nome	Morada	Produto	Preco
12345	Pregos, Lda	Azinhaga Trinta e Um, 32, R/C Drt.	Cavilhas	0,05 €
12345	Pregos, Lda	Azinhaga Trinta e Um, 32, R/C Drt.	Parafusos 6mm	0,20 €
99912	Sizal & Filhos	Av. Das Descobertas 78	Cordel	0,50 €
14566	Borracheiro	Rua Bica funda, 12, Lote 3	Mangueira 30mm	0,25 €
12345	Pregos, Lda	Azinhaga Trinta e Um, 32, R/C Drt.	Anilhas	0,01 €
14566	Borracheiro	Rua Bica funda, 12, Lote 3	Botas Borracha	5,00 €

A Relação apresentada apresenta **informação redundante**. Quais os problemas que se colocam?

Noção de Normalização

Problemas de manutenção

A redundância significa armazenamento **repetido** da mesma informação.

Por essa razão, a alteração ou remoção de informação pode implicar o acesso a diversas partes da base de dados, tornando-se **difícil manter a coerência** dos dados.

Noção de Normalização

Custos de armazenamento

Uma vez que existe informação repetida, existe espaço que está a ser desperdiçado.

Esse desperdício pode ser maior ou menor consoante exista muita ou pouca redundância.

Mesmo com o preço do Tera Byte a descer, quando o volume de informação é grande, este é um problema a ter em conta.

Noção de Normalização

Problemas de desempenho

Embora menos evidente, se existe desperdício de espaço, é necessário um maior acesso ao disco para aceder à mesma informação.

Como o suporte magnético é tendencialmente lento, esses acessos traduzem-se em tempo desperdiçado.

Noção de Normalização

Anomalia Alteração

Como alterar a morada do fornecedor 'Pregos Lda.' ?

Que aconteceria se essa alteração não fosse feita para um dos tuplos?

CodigoFornecedor	Nome	Morada	Produto	Preco
12345	Pregos, Lda	Azinhaga Trinta e Um, 32, R/C Drt.	Cavilhas	0,05 €
12345	Pregos, Lda	Azinhaga Trinta e Um, 32, R/C Drt.	Parafusos 6mm	0,20 €
99912	Sizal & Filhos	Av. Das Descobertas 78	Cordel	0,50 €
14566	Borracheiro	Rua Bica funda, 12, Lote 3	Mangueira 30mm	0,25 €
12345	Pregos, Lda	Azinhaga Trinta e Um, 32, R/C Drt.	Anilhas	0,01 €
14566	Borracheiro	Rua Bica funda, 12, Lote 3	Botas Borracha	5,00 €

Noção de Normalização

Anomalia Remoção

Como proceder para remover os tuplos relacionados com um determinado Fornecedor?

Os produtos por eles fornecidos serão igualmente apagados?

Como remover um produto?

Tanto CodigoFornecedor e Produto são chaves!

Noção de Normalização

Anomalia Inserção

Como se adiciona um novo fornecedor?
Será possível essa adição sem ele fornecer algum produto?

- Colocando NULL nos campos respeitantes aos Produtos
- Mas, e se no fornecimento do primeiro produto esses valores a NULL ficassem esquecidos?

Produto identifica univocamente um produto –
Não é permitido ter valor NULL!

*Como resolver as anomalias
apresentadas ?*

Noção de Normalização

Decompondo

Acabando com a informação repetida, ou seja, redundante.

Essa redundância pode ser ultrapassada
decompondo as Relações noutras Relações mais pequenas

A decomposição passa primeiro pela análise do que não pode ser decomposto

Noção de Normalização

Decompondo

CodigoFornecedor	Nome	Morada	Produto	Preco
12345	Pregos, Lda	Azinhaga Trinta e Um, 32, R/C Drt.	Cavilhas	0,05 €
12345	Pregos, Lda	Azinhaga Trinta e Um, 32, R/C Drt.	Parafusos 6mm	0,20 €
99912	Sizal & Filhos	Av. Das Descobertas 78	Cordel	0,50 €
14566	Borracheiro	Rua Bica funda, 12, Lote 3	Mangueira 30mm	0,25 €
12345	Pregos, Lda	Azinhaga Trinta e Um, 32, R/C Drt.	Anilhas	0,01 €
14566	Borracheiro	Rua Bica funda, 12, Lote 3	Botas Borracha	5,00 €

No exemplo:

CodigoFornecedor não pode ser separado de **nome** e **morada** – Existe uma dependência entre eles

CodigoFornecedor não pode ser separado de **produto** e de **preço** – Cada fornecedor fornece um ou vários produtos

Noção de Normalização

Sabendo que **CodigoFornecedor** determina o *nome* e a *morada*

e que **CodigoFornecedor** determina o *produto* e *preço*, a Relação R será decomposta em:

FORNECEDOR(CodigoFornecedor, Nome,
Morada)

PRODUTO_FORNECIDO(CodigoFornecedor,
Produto, Preço)

Noção de Normalização

Mas será que se consegue obter a mesma informação que estava na Relação R?

Pretende-se determinar o nome do fornecedor de 'cordel':

$$\pi_{\text{nome}}(\sigma_{\text{FORNECEDOR.CodigoFornecedor} = \text{PRODUTO_FORNECIDO.CodigoFornecedor} \ \&\& \ \text{PRODUTO_FORNECIDO.Produto} = \text{'cordel'}})$$

Não se perdeu informação com a decomposição

Noção de Normalização

Foi necessário efectuar uma junção para se responder à questão

Existe um equilíbrio que é necessário manter, entre performance e decomposição

Noção de Normalização

Nos dados estão patentes algumas dependências que são necessárias conhecer para se proceder à decomposição de Relações

Essas dependências podem ser:

- Dependências Funcionais
- Dependências Multi-valor
- Dependências de Junção

Noção de Normalização

Com base nestas dependências serão definidas um conjunto de regras para resolver os problemas de redundância designadas de **Formas Normais**

Para as primeiras dependências são definidas quatro *formas normais*:

- 1ª, 2ª, 3ª e Boyce-Cood

Para as dependências multi-valor é definida a 4ª Forma Normal

Para as dependências de junção é definida a 5ª Forma Normal

Noção de Normalização

Dependência Funcional

Seja $R(A_1, \dots, A_n)$ um esquema de uma relação e X e Y subconjuntos de $\{A_1, \dots, A_n\}$.

Diz-se que X determina funcionalmente Y ou que Y é funcionalmente dependente de X ($X \rightarrow Y$)

se para quaisquer que forem os tuplos da relação,

não for possível existirem dois tuplos que concordem em todos atributos em X e discordem em pelo menos um atributo em Y .

Noção de Normalização

Dependência Funcional

Na definição dos problemas, existem subjacentes aos dados algumas relações perfeitamente definidas

Na Relação **Fornecedor**, **codigoFornecedor** identifica sempre o nome e a morada de um fornecedor.

Ou seja, existe uma Dependência Funcional entre **codigoFornecedor** e os outros dois atributos: *nome* e *morada*



No entanto, o inverso pode não ser verdade !

Noção de Normalização

FORNECEDOR

CodigoFornecedor	Nome	Morada
12345	Pregos, Lda	Azinhaga Trinta e Um, 32, R/C Drt.
14566	Borracheiro	Rua Bica funda, 12, Lote 3
99912	Sizal & Filhos	Av. Das Descobertas 78

PRODUTO_FORNECIDO

CodigoFornecedor	Produto	Preco
12345	Cavilhas	0,05 €
12345	Parafusos 6mm	0,20 €
99912	Cordel	0,50 €
14566	Mangueira 30mm	0,25 €
12345	Anilhas	0,01 €
14566	Botas Borracha	5,00 €

Noção de Normalização

Identificação de Dependências Funcionais

As Dependências Funcionais estão intrinsecamente ligadas ao problema em questão e não podem ser inferidas directamente a partir de algumas linhas das tabelas

No entanto, através da semântica dos atributos é possível extrair algumas dependências funcionais

numero	nome	dataNasc
12345	Antonio	10-10-1970
43321	Nuno	21-09-1970
12238	Antonio	22-10-1968
12231	Maria	12-05-1976

Noção de Normalização

Identificação de Dependências Funcionais

- **nome** depende de **numero**?

numero	nome	dataNasc
12345	Antonio	10-10-1970
43321	Nuno	21-09-1970
12238	Antonio	22-10-1968
12231	Maria	12-05-1976

Sabendo que cada aluno só pode ter um número, conhecendo esse número sabe-se o nome do aluno.

*Assim **nome** é funcionalmente dependente de **número**.*

No entanto foi necessário ter conhecimento do problema!

Noção de Normalização

Identificação de Dependências Funcionais

- **nome** determina **dataNasc**?

nome → **dataNasc**

numero	nome	dataNasc
12345	Antonio	10-10-1970
43321	Nuno	21-09-1970
12238	Antonio	22-10-1968
12231	Maria	12-05-1976

$t1[\text{nome}] = t2[\text{nome}] \rightarrow t1[\text{dataNasc}] = t2[\text{dataNasc}]$

$t1[\text{nome}] = \text{'Antonio'} = t2[\text{nome}] = \text{Antonio} \rightarrow$
 $t1[\text{dataNasc}] = \text{'10-10-1970'} = t2[\text{dataNasc}] = \text{'22-10-1968'}$

Sendo falso, **nome não determina dataNasc**

Noção de Normalização

Identificação de Dependências Funcionais (cont.)

Fornecedor	Nome	Produto	Preco
12345	Pregos, Lda	Cavilhas	0,05 €
99912	Sizal & Filhos	Cordel	0,50 €
1122	Latão Douro	Cavilhas	0,20 €
14566	Borracheiro	Mangueira 30mm	0,25 €
14566	Borracheiro	Botas Borracha	5,00 €

Quais os atributos que determinam o preço dos produtos

Fornecedor → Preço ?????

Produto → Preço ?????

Noção de Normalização

Identificação de Dependências Funcionais (cont.)

Quais os atributos que determinam o preço dos produtos

Fornecedor $\rightarrow$ Preço ????

Não. Cada fornecedor tem um preço para os produtos que vende

Produto $\rightarrow$ Preço ????

Não. Dependendo do fornecedor, o preço dos produtos variam

Ou seja, **preço** é funcionalmente dependente de dois atributos

(fornecedor, produto) $\rightarrow$ preço

Exemplo Noção de Normalização

Pretende-se armazenar informação sobre clientes e as suas encomendas

- Para os clientes
 - Numero de cliente (único), nome, morada
- Para os produtos
 - Código do produto (único), designação
- Para as encomendas
 - Numero de encomenda (único), data de encomenda, dados do Cliente, dados dos produtos encomendados, quantidade de cada produto

Noção de Normalização

Exemplo

Encomenda(numEnc, numcli, cliente, morada,
dataEnc, codProd, produto, quantEnc)

Noção de Normalização

Exemplo

numEnc	numCli	cliente	morada	dataEnc	codProd	produto	quantEnc
1	1	Antonio	Rua 1	10-03-2002	1	Batatas	2
1	1	Antonio	Rua 1	10-03-2002	2	Feijão	1
1	1	Antonio	Rua 1	10-03-2002	3	Couves	1
2	2	Maria	Rua 2	12-04-2002	1	Batatas	5
2	2	Maria	Rua 2	12-04-2002	2	Feijão	1
3	3	João	Rua 3	12-05-2001	4	Laranjas	4
3	3	João	Rua 3	12-05-2001	1	Batatas	2

A tabela acima ilustra a informação que pode ser armazenada

- Da forma como é apresentada, colocam-se alguns problemas relacionados com a redundância
- Com a Normalização, esses problemas serão resolvidos

Noção de Normalização

1ª Forma Normal - 1NF

Um Esquema de Relação está na 1NF se os valores para o seu domínio forem atômicos

- Ou, por outras palavras, um Esquema de Relação está na 1NF *se não existirem “Relações dentro de Relações”*
- A Relação anterior pode ser visto como:

numEnc	numCli	cliente	morada	dataEnc	codProd	produto	quantEnc
1	1	Antonio	Rua 1	10-03-2002	1	Batatas	2
					2	Feijão	1
					3	Couves	1
2	2	Maria	Rua 2	12-04-2002	1	Batatas	5
					2	Feijão	1
3	3	João	Rua 3	12-05-2001	4	Laranjas	4
					1	Batatas	2

- Os atributos codProd, produto e quantEnc não são atômicos ! (cada encomenda tem vários produtos)

Noção de Normalização

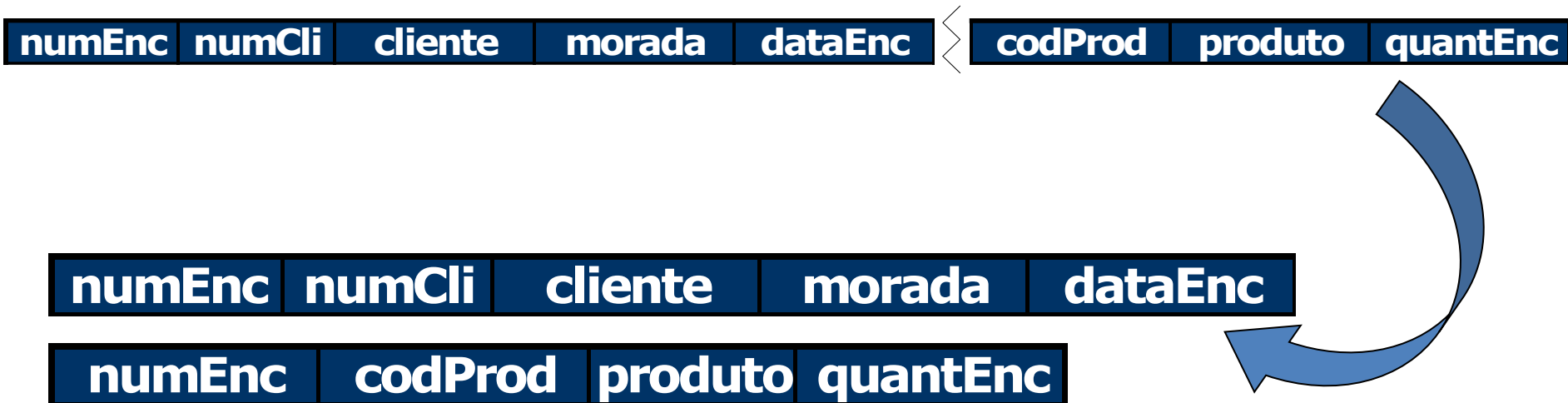
1ª Forma Normal - 1NF

- A forma de colocar o Esquema de Relação na 1NF é decompondo, até que todos os Esquemas de Relação recém criados estejam na 1NF

No exemplo, o Esquema de Relação tem de ser decomposto em dois

Noção de Normalização

1ª Forma Normal - 1NF



Como ambos os novos Esquemas de Relação estão na 1NF, não é necessário continuar a decompor

Isso não quer dizer,
que não exista ainda redundância!

Noção de Normalização

1ª Forma Normal - 1NF

Esquema de Relação Encomenda:

- Encomenda(numEnc, numcli, cliente, morada, dataEnc)
 - Onde a chave candidata/chave primária é **numEnc**
- Esquema de Relação Linhas_Encomenda:
- Linhas_Encomenda (numEnc, codProd, produto, quantEnc)
 - Onde a chave candidata/chave primária é composta por **numEnc**, **codProd**

Noção de Normalização

1ª Forma Normal - 1NF

numEnc	numCli	cliente	morada	dataEnc	Encomenda
1	1	Antonio	Rua 1	10-03-2002	
2	2	Maria	Rua 2	12-04-2002	
3	3	João	Rua 3	12-05-2001	

Linhas_Encomenda

numEnc	codProd	produto	quantEnc
1	1	Batatas	2
1	2	Feijão	1
1	3	Couves	1
2	1	Batatas	5
2	2	Feijão	1
3	4	Laranjas	4
3	1	Batatas	2

Noção de Normalização

1ª Forma Normal - 1NF

Será que os problemas ficaram todos resolvidos?

Anomalia de inserção

Anomalia de remoção

Anomalia de alteração

*(Estas anomalias caso existam
serão resolvidas recorrendo às 2NF e 3NF)*

Noção de Normalização

- Anomalia de inserção

– Pretende-se adicionar um novo produto para que possa estar disponível para novas encomendas

numEnc	numCli	cliente	morada	dataEnc
1	1	Antonio	Rua 1	10-03-2002
2	2	Maria	Rua 2	12-04-2002
3	3	João	Rua 3	12-05-2001

– Mas como **numEnc** faz parte da chave da tabela, não é possível adicionar um novo produto sem existir uma encomenda para ele !

numEnc	codProd	produto	quantEnc
1	1	Batatas	2
1	2	Feijão	1
1	3	Couves	1
2	1	Batatas	5
2	2	Feijão	1
3	4	Laranjas	4
3	1	Batatas	2

Noção de Normalização

- Anomalia de remoção

– Se se pretender remover informação sobre um determinado produto, perde-se informação desse produto nas encomendas

numEnc	numCli	cliente	morada	dataEnc
1	1	Antonio	Rua 1	10-03-2002
2	2	Maria	Rua 2	12-04-2002
3	3	João	Rua 3	12-05-2001

numEnc	codProd	produto	quantEnc
1	1	Batatas	2
1	2	Feijão	1
1	3	Couves	1
2	1	Batatas	5
2	2	Feijão	1
3	4	Laranjas	4
3	1	Batatas	2

Noção de Normalização

- Anomalia de alteração
 - Se se pretender alterar a nome do produto ‘Feijão’ para ‘Feijão Frade’, é necessário percorrer um conjunto de tuplos e proceder a essa alteração em cada um deles

numEnc	numCli	cliente	morada	dataEnc
1	1	Antonio	Rua 1	10-03-2002
2	2	Maria	Rua 2	12-04-2002
3	3	João	Rua 3	12-05-2001

numEnc	codProd	produto	quantEnc
1	1	Batatas	2
1	2	Feijão	1
1	3	Couves	1
2	1	Batatas	5
2	2	Feijão	1
3	4	Laranjas	4
3	1	Batatas	2

Noção de Normalização

2ª Forma Normal - 2NF

Quando um ou mais atributos não primos
(i.e. não pertencente a uma chave candidata)
dependem parcialmente da chave, o Esquema
de Relação não está na 2NF

- Quando um Esquema de Relação não está na 2NF é porque existem DF (Dependência Funcionais) que não são totais

Noção de Normalização

2ª Forma Normal - 2NF

- Uma DF $X \rightarrow Y$ diz-se total quando

$$\neg \exists Z : X (X - \{ Z \}) \rightarrow Y$$

– Ou seja, Y não é Funcionalmente dependente de nenhum subconjunto de X

- Pelo contrário
- uma DF $X \rightarrow Y$ diz-se parcial quando

$$\exists Z : X (X - \{ Z \}) \rightarrow Y$$

Noção de Normalização

2ª Forma Normal - 2NF

Formalmente:

Seja $R(A_1, A_2, \dots, A_n)$

R está na 2NF se:

- ✓ R está na 1NF
- ✓ $X \in \{\text{Chaves Candidatas}\}$ e $A_i \notin \{\text{Chaves Candidatas}\}$,
a DF $X \rightarrow A_i$ é total

Noção de Normalização

2ª Forma Normal - 2NF

Sempre que um Esquema de Relação está na 1NF e tem uma chave constituída por um único atributo esse Esquema de Relação está na 2NF

No exemplo existem dependências parciais:

O atributo **produto**, no Esquema **linhas_encomenda** depende parcialmente da chave, uma vez que apenas o **codProd** o determina

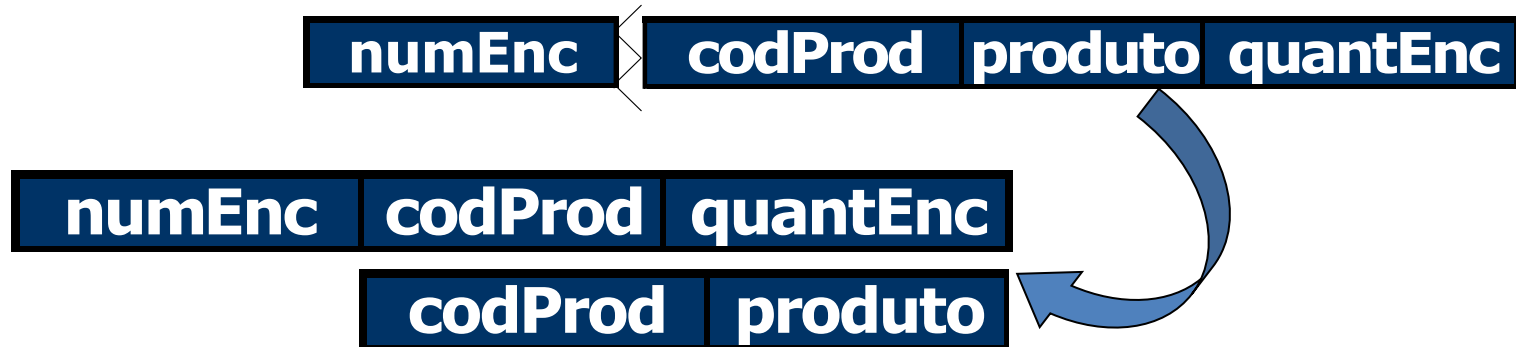
numEnc	codProd	produto	quantEnc
1	1	Batatas	2
1	2	Feijão	1
1	3	Couves	1
2	1	Batatas	5
2	2	Feijão	1
3	4	Laranjas	4
3	1	Batatas	2

Noção de Normalização

2ª Forma Normal - 2NF

Mais uma vez, é necessário decompor os Esquemas de Relação para resolver as anomalias subjacentes às DF parciais

O Esquema linhas_encomenda será decomposto



Noção de Normalização

2ª Forma Normal - 2NF



Para o primeiro esquema entram a chave e os atributos que dela dependem totalmente



No outro esquema ficam os atributos que dependem parcialmente da chave, mais essa parte da chave



Noção de Normalização

2ª Forma Normal - 2NF

Esquema de Relação Linhas_Encomenda:

Linhas_Encomenda(numEnc, codProd, quantEnc)

Onde a chave candidata/chave primária é composta por
numEnc, codProd

Linhas_Encomenda

numEnc	codProd	quantEnc
1	1	2
1	2	1
1	3	1
2	1	5
2	2	2
3	4	4
3	1	2

Esquema de Relação Produto
Produto(codProd, produto)

Onde a chave candidata/chave primária é **codProd**

Produto

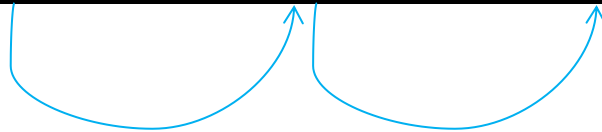
codProd	produto
1	Batatas
2	Feijão
3	Couves
4	Laranjas

Noção de Normalização

3ª Forma Normal - 3NF

Diz-se que um esquema de uma relação está na 3NF se nenhum dos seus atributos não primos for transitivamente dependente de alguma das suas chaves.

numEnc	numCli	cliente	morada	dataEnc
1	1	Antonio	Rua 1	10-03-2002
2	2	Maria	Rua 2	12-04-2002
3	3	João	Rua 3	12-05-2001



Na relação encomenda cliente, morada são transitivamente dependentes da chave **numEnc** através de **numCli**

A -> B e B -> C então A->C

Noção de Normalização

3ª Forma Normal - 3NF

Mais uma vez, é necessário decompor os Esquemas de Relação para resolver as anomalias subjacentes às DF transitivas
O Esquema *encomenda* será decomposto



- Para o primeiro esquema entram a chave e os atributos que dela dependem totalmente
- No outro esquema ficam os atributos que dependem transitivamente da chave mais o atributo que os determina

Noção de Normalização

3ª Forma Normal - 3NF



- Para o primeiro esquema entram a chave e os atributos que dela dependem totalmente



- No outro esquema ficam os atributos que dependem transitivamente da chave mais o atributo que os determina



Noção de Normalização

3ª Forma Normal - 3NF

- Esquema de Relação Encomenda:
- Encomenda(numEnc, numCli, dataEnc)

Encomenda

Onde a chave candidata/chave primária é **numEnc**

numEnc	numCli	dataEnc
1	1	10-03-2002
2	2	12-04-2002
3	3	12-05-2001

Esquema de Relação Cliente
Cliente(numCli, cliente, morada)

Cliente

Onde a chave candidata/chave primária é **numCli**

numCli	cliente	morada
1	Antonio	Rua 1
2	Maria	Rua 2
3	João	Rua 3

Noção de Normalização

3ª Forma Normal - 3NF

Existem alguns problemas neste “método intuitivo” de Decomposição

- Considerando

$R(A,B,C,D,E)$ $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E\}$

Pretende-se colocar o Esquema R na 3NF

– A chave de R será dada por $\{AB\}$

Noção de Normalização

3ª Forma Normal - 3NF

$R(A, B, C, D, E)$ e $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E\}$

Pretende-se colocar o Esquema R na 3NF

A chave de R será dada por $\{AB\}$

Uma possível Decomposição de R na 3FN será:

$R_1(A, B, C)$

$R_2(D, E)$

Será que com R_1 e R_2 ainda se têm todo o F ?

Perdeu-se a DF C $\rightarrow$ D!

Noção de Normalização

Preservação da Dependências Funcionais

- Quando se normaliza um Esquema de Relação é importante que se mantenham todas as DF (Dependências Funcionais) existentes em F
- Essa DF podem ser inferidas
 - Num dos Esquemas de Relação resultantes da decomposição
 - Ou a partir de alguma DF que apareça nesses Esquemas

Noção de Normalização

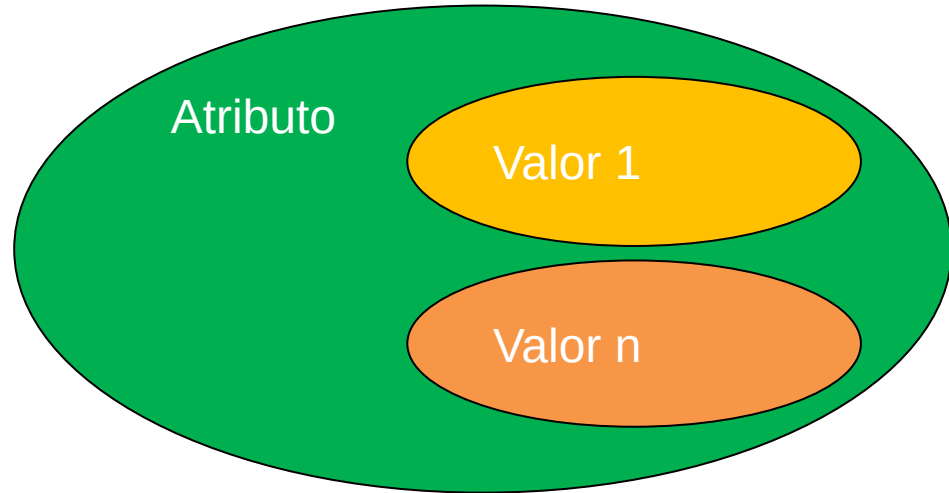
Preservação da Dependências Funcionais

- Quando isto acontece, diz-se que houve preservação das Dependências Funcionais
- É importante que se preservem as DF, pois elas representam restrições que os dados devem respeitar

Base de Dados

- Resumindo
- A Normalização é importante para evitar anomalias ao:
 - Acrescentar (Inserir)
 - Actualizar
 - Apagar

Base de Dados



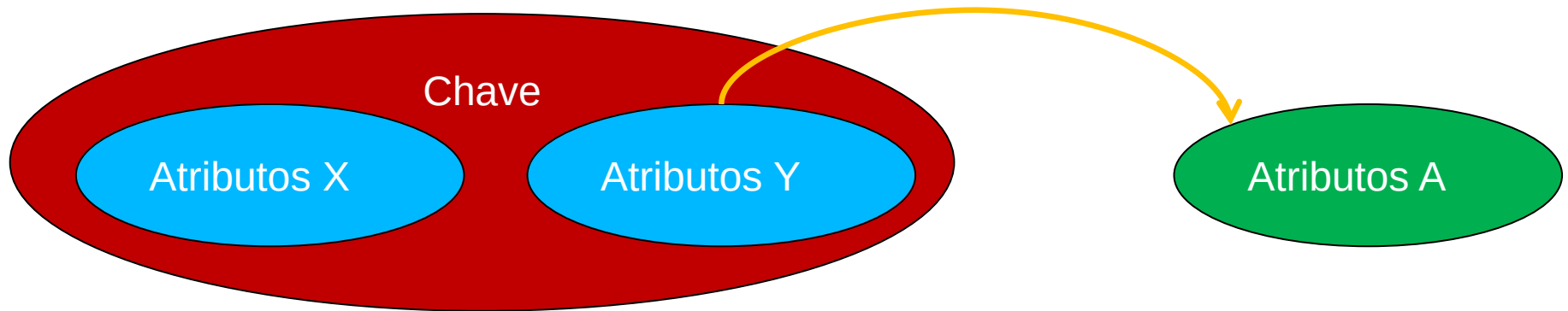
Atributo como vários valores

Se isto acontecer não está na 1ªFN

1FN →

**Todos os atributos estão definidos em domínios
que contêm apenas valores atômicos**

Base de Dados



Dependência Parcial

Se isto acontecer não está na 2ªFN

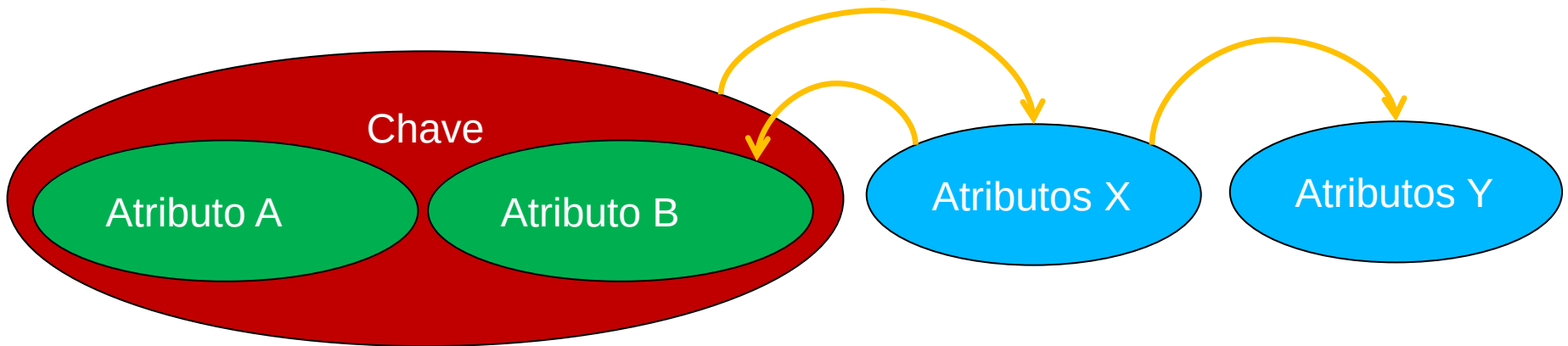
2FN →

Tem de estar na 1FN

**Todos os atributos que não pertencem à chave,
dependem da chave através de uma dependência funcional elementar
(têm de depender da chave completa e não apenas de parte dela)**

Base de Dados

Dependência Transitiva



Se isto acontecer não está na 3^aFN

3FN →

Tem de estar na 2FN

**Todos os atributos que não pertencem à chave,
devem ser mutuamente independentes**

(depender só da chave e não de outro atributo ou conjunto de atributos)

Base de Dados

Não Normalizado

Resolver grupos repetidos

1ªFN

Resolver dependências parciais

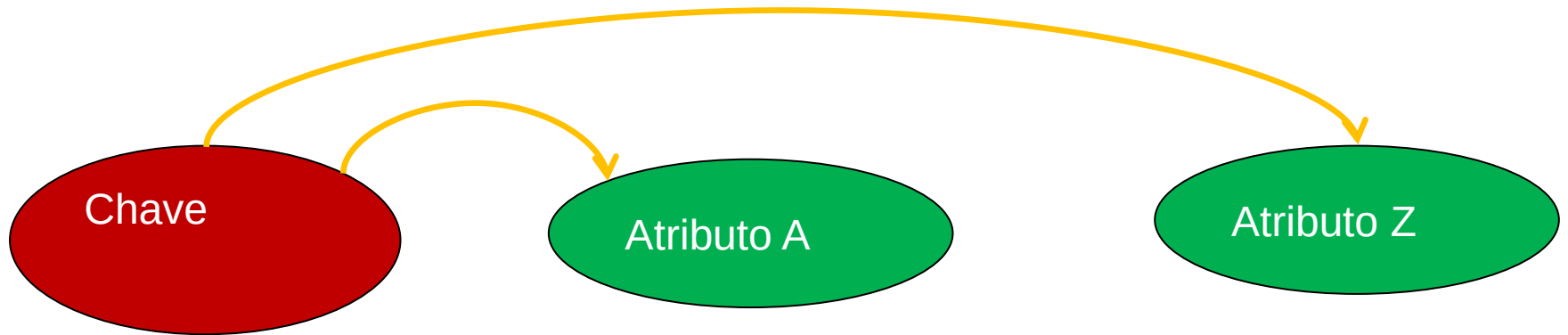
2ªFN

Resolver dependências transitivas

3ªFN

Resolver anomalias de dependências funcionais

BCNF



BCNF - Boyce/Codd Normal Form
(é um caso particular da 3FN)

Perdas por junção

(pela definição)

Se a relação R for decomposta nas relações R1 e R2

Há perdas se $R1 \bowtie R2 \supset R$

Sem perdas se $R1 \bowtie R2 = R$

Perdas por junção

(Verificar através das dependências funcionais)

Se a relação R for decomposta nas relações R1 e R2

A União dos atributos de R1 e R2 têm de ser iguais aos de R.

(Cada atributo de R tem estar em R1 e ou R2)

$$\mathbf{Att(R1) \cup Att(R2) = Att(R)}$$

A intersecção dos atributos de R1 e R2 não pode ser nula

$$\mathbf{Att(R1) \cap Att(R2) \neq \Phi}$$

Os atributos comuns têm de ser chave pelo menos numa das relações (R1 ou R2)

$$\mathbf{Att(R1) \cap Att(R2) \rightarrow Att(R1) \vee Att(R1) \cap Att(R2) \rightarrow Att(R2)}$$

Preservar Dependências Funcionais

Se a relação R for decomposta nas relações $R1$ e $R2$

Há preservação das dependências funcionais se:

Todas as dependências funcionais de R
têm de fazer parte de $R1$ e ou $R2$
e ou terem de ser deriváveis
através da combinação das DF de $R1$ e $R2$

Perdas por Junção e preservação de DF

É possível haver as 4 combinações

Sem perdas por junção e preservar DF

Com perdas por junção e preservar DF

Sem perdas por junção e não preservar DF

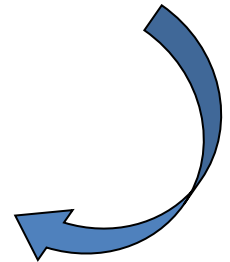
Com perdas por junção e não preservar DF

numEnc	numCli	cliente	morada	dataEnc	codProd	produto	quantEnc
1	1	Antonio	Rua 1	10-03-2002	1	Batatas	2
					2	Feijão	1
					3	Couves	1
2	2	Maria	Rua 2	12-04-2002	1	Batatas	5
3	3	João	Rua 3	12-05-2001	2	Feijão	1
					4	Laranjas	4
					1	Batatas	2

numEnc	numCli	cliente	morada	dataEnc	codProd	produto	quantEnc
--------	--------	---------	--------	---------	---------	---------	----------

numEnc	numCli	cliente	morada	dataEnc
--------	--------	---------	--------	---------

numEnc	codProd	produto	quantEnc
--------	---------	---------	----------



numEnc	numCli	cliente	morada	dataEnc
1	1	Antonio	Rua 1	10-03-2002
2	2	Maria	Rua 2	12-04-2002
3	3	João	Rua 3	12-05-2001

Encomenda

Agora está na 1ªFN

Linhas_Encomenda

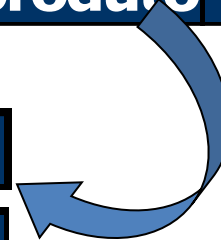
numEnc	codProd	produto	quantEnc
1	1	Batatas	2
1	2	Feijão	1
1	3	Couves	1
2	1	Batatas	5
2	2	Feijão	1
3	4	Laranjas	4
3	1	Batatas	2

numEnc	codProd	produto	quantEnc
1	1	Batatas	2
1	2	Feijão	1
1	3	Couves	1
2	1	Batatas	5
2	2	Feijão	1
3	4	Laranjas	4
3	1	Batatas	2

numEnc	codProd	produto	quantEnc
--------	---------	---------	----------

numEnc	codProd	quantEnc
--------	---------	----------

codProd	produto
---------	---------



Linhas_Encomenda

numEnc	codProd	quantEnc
1	1	2
1	2	1
1	3	1
2	1	5
2	2	2
3	4	4
3	1	2

Produto

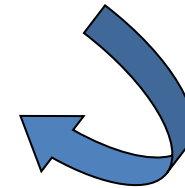
codProd	produto
1	Batatas
2	Feijão
3	Couves
4	Laranjas

numEnc	numCli	cliente	morada	dataEnc
1	1	Antonio	Rua 1	10-03-2002
2	2	Maria	Rua 2	12-04-2002
3	3	João	Rua 3	12-05-2001

numEnc	numCli	cliente	morada	dataEnc
--------	--------	---------	--------	---------

numEnc	numCli	dataEnc
--------	--------	---------

numCli	cliente	morada
--------	---------	--------



Agora está na 3ªFN

numEnc	numCli	dataEnc
1	1	10-03-2002
2	2	12-04-2002
3	3	12-05-2001

Encomenda

numCli	cliente	morada
1	Antonio	Rua 1
2	Maria	Rua 2
3	João	Rua 3

Cliente

Linhas_Encomenda

numEnc	codProd	quantEnc
1	1	2
1	2	1
1	3	1
2	1	5
2	2	2
3	4	4
3	1	2

Produto

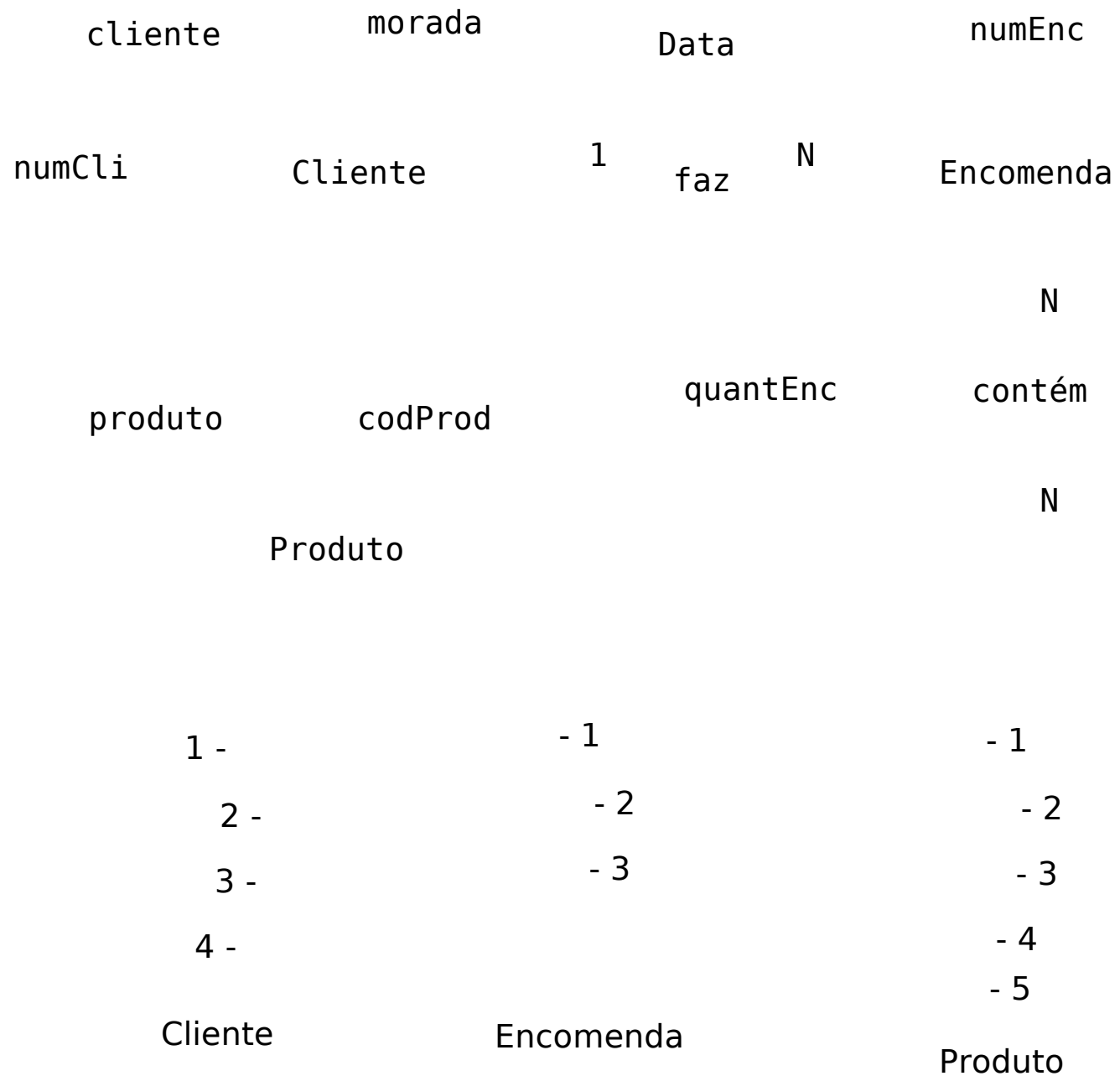
codProd	produto
1	Batatas
2	Feijão
3	Couves
4	Laranjas

numEnc	numCli	dataEnc
1	1	10-03-2002
2	2	12-04-2002
3	3	12-05-2001

Encomenda

Cliente

numCli	cliente	morada
1	Antonio	Rua 1
2	Maria	Rua 2
3	João	Rua 3



Cliente(cliente, morada, numCli) cc={{numCli}}

Produto(produto, codProd) cc={{codProd}}

Encomenda(numEnc) cc={{numEnc}}



Encomenda(numEnc, numCli, data)

cc={{numEnc}}

ce={{numCli}} numCli: Cliente(numCli)

Contem(numEnc, numCli, quantEnc)

← Relação Contém
(*Linhas_Encomenda*)

cc={{numEnc , numCli}}

ce={{numCli, numEnc}}

numCli: Cliente(numCli)

numEnc: Encomenda(numEnc)

VulcaoErupcao(**numeroVulcao**, nomeVulcao, numeroErupcao, eruptionCategory, activityArea, vei, veiModifier, startYearModifier, startYear, startYearUncertainty, startMonth, startDayModifier, startDay, startDayUncertainty, evidenceMethod, endYearModifier, endYear, endYearUncertainty, endMonth, endMonthModifier, endDay, endDayUncertainty);

VulcaoEvento(**numeroVulcao**, nomeVulcao, **numeroErupcao**, eruptionStartYear, eventNumber, eventType, event, eventDateYearModifier, eventDateYear, eventDateYearUncertainty, eventDateMonth, eventDateMonthModifier, eventDateDay, eventDateDayUncertainty);

VulcaoReferencia(**numeroVulcao**, nomeVulcao, autor, anoPublicacao, titulo, fonte);

VulcaoVulcao(numeroVulcao, nomeVulcao, tipoPrimario, ultimaErupcao, pais, regioao, subRegiao, latitude, longitude, elevacao, tectonicSetting, evidenceCategory, majorRock1, majorRock2, majorRock3, majorRock4, majorRock5, minorRock1, minorRock2, minorRock3, minorRock4, minorRock5, population5Km, population10Km, population30Km, population100Km);

VulcaoVulcao(numeroVulcao,

nomeVulcao, tipoPrimario, pais, regioao, subRegiao, latitude,
longitude, elevacao, tectonicSetting, evidenceCategory,

majorRock1, majorRock2, majorRock3, majorRock4,
majorRock5,
minorRock1, minorRock2, minorRock3, minorRock4,
minorRock5,

population5Km, population10Km, population30Km,
population100Km

ultimaErupcao

);

```
VulcaoErupcao(numeroErupcao,  
eruptionCategory, activityArea, vei, veiModifier,  
startYearModifier, startYear, startYearUncertainty, startMonth,  
startDayModifier, startDay, startDayUncertainty,  
evidenceMethod, endYearModifier, endYear,  
endYearUncertainty, endMonth, endMonthModifier, endDay,  
endDayUncertainty  
  
numeroVulcao, nomeVulcao,  
  
);
```

```
VulcaoEvento(eventNumber,  
  
eventType, event, eventDateYearModifier, eventDateYear,  
eventDateYearUncertainty, eventDateMonth,  
eventDateMonthModifier, eventDateDay,  
eventDateDayUncertainty  
  
numeroVulcao, nomeVulcao,  
  
numeroErupcao, eruptionStartYear,  
  
);  
  
// Eventos causados pela erupção
```

```
VulcaoReferencia(  
    numeroVulcao,  
    nomeVulcao,  
    autor, anoPublicacao, titulo, fonte  
);
```

Base de Dados

- 1ª Função de Normalização

Alunos

Nome	Num	Morada	Disciplinas
Ana	2341234	Pc. A Lisboa	Mat, Fis, Bio
Bruno	3453453	Av. B Oeiras	Mat, Des
Carla	8796834	R. C Loures	Soc, Port

Base de Dados

- 1ª Função de Normalização
- Todos os atributos (colunas) definidos em domínios que contêm apenas valores atómicos

- 1ª Função de Normalização

Base de Dados

Alunos

Nome	Num	Morada	Disciplina
Ana	2341234	Pc. A Lisboa	Mat
Ana	2341234	Pc. A Lisboa	Fis
Ana	2341234	Pc. A Lisboa	Bio
Bruno	3453453	Av. B Oeiras	Mat
Bruno	3453453	Av. B Oeiras	Des
Carla	8796834	R. C Loures	Soc
Carla	8796834	R. C Loures	Port

Base de Dados

Problemas

- **Actualização** – alterar a morada
- **Inserção** – aluno que não tem nenhuma matrícula (não aparece na tabela)
- **Eliminação** – ao eliminar a inscrição do aluno na disciplina (desaparece da tabela)

Base de Dados

- 1ª Função de Normalização

Nome	Num	Morada
Ana	2341234	Pc. A Lisboa
Bruno	3453453	Av. B Oeiras
Carla	8796834	R. C Loures

Alunos

Disciplina	Id
Mat	21
Fis	31
Bio	8
Des	45
Port	6
Soc	79

Disciplinas

Aluno	Disciplina
2341234	21
2341234	31
2341234	8
3453453	21
3453453	45
8796834	79
8796834	6

Matriculas

Base de Dados

- 2ª Função de Normalização
- Todos os atributos (colunas) dependem apenas da chave.
- (não dependem de subconjuntos da chave)

Numero	Nome	Ano
110100043	A	2000
181903004	B	2003
182405004	C	2005
180704124	D	2004

Base de Dados

Chave

- 3ª Função de Normalização
- Todos os atributos (colunas) que não pertencem à chave são mutuamente independentes.

Numero	Nome	Senha
110100043	A	2
181903004	B	1
182405004	C	4
180704124	D	5

São mutuamente independentes se puderem ser actualizados independentemente dos outros

Base de Dados

Referências

<http://rdbms.opengrass.net/index.html>

<http://www.ict.griffith.edu.au/~jw/normalization/>

