Motivação

Parece ser melhor decompor em:

nConta	nomeCliente
1	Carlos
1	José
2	Carlos
2	Maria
3	Maria
3	José

nomeCliente	moradaCliente
Carlos	morada1
Carlos	morada2
José	morada3
Maria	morada1
Maria	morada4

- Mas porquê? Que propriedades têm estes dados que permitem dizer isto? Como as exprimir?
- É certo que um dado cliente não têm sempre a mesma morada (independentemente da conta).
- Mas tem sempre o mesmo conjunto de moradas, independentemente da conta!

Dependências Multi-valor

Definição (Dependências Multi-valor)

Seja R um esquema e $\alpha \subseteq R$ e $\beta \subseteq R$. A **dependência multi-valor**

$$\alpha \twoheadrightarrow \beta$$

é verdadeira em R se em toda a relação possível $r(R)$, para todo o par de tuplo t_1, t_2 em r, se $t_1[\alpha] = t_2[\alpha]$, então existem necessariamente tuplos t_3 e t_4 em r tal que:

- $t_1[\alpha] = t_2[\alpha] = t_3[\alpha] = t_4[\alpha]$
- $t_3[\beta] = t_1[\beta]$
- $t_3[R - \beta] = t_2[R - \beta]$
- $t_4[\beta] = t_2[\beta]$
- $t_4[R - \beta] = t_1[R - \beta]$

Dependências Multi-valor (Cont.)

- Representação de $\alpha \twoheadrightarrow \beta$ em tabela:

	α	β	$R - \alpha - \beta$
t_1	$a_1 \ldots a_i$	$a_{i+1} \ldots a_j$	$a_{j+1} \ldots a_n$
t_2	$a_1 \ldots a_i$	$b_{i+1} \ldots b_j$	$b_{j+1} \ldots b_n$
t_3	$a_1 \ldots a_i$	$a_{i+1} \ldots a_j$	$b_{j+1} \ldots b_n$
t_4	$a_1 \ldots a_i$	$b_{i+1} \ldots b_j$	$a_{j+1} \ldots a_n$

- Para $\texttt{nomeCliente} \twoheadrightarrow \texttt{moradaCliente}$:
  - Se dois tuplos têm o mesmo `nomeCliente`, tendo um `moradaCliente`$= m_1$ e `nConta` $= n_1$ e outro `moradaCliente` $= m_2$ e `nConta` $= n_2$.
  - Então têm que haver mais dois tuplos com esse `nomeCliente`:
    - um com `moradaCliente` $= m_1$ e `nConta` $= n_2$;
    - outro com `moradaCliente` $= m_2$ e `nConta` $= n_1$.

Exemplo

- Seja R um esquema com um conjunto de atributos particionados em 3 subconjuntos não vazios.

$$Y, Z, W$$

- Diz-se que $Y \twoheadrightarrow Z$ (Y multi-determina Z) sse para todas as possíveis $r(R)$
  Se $\quad <y_1, z_1, w_1> \in r \quad$ e $\quad <y_1, z_2, w_2> \in r$
  então $\quad <y_1, z_1, w_2> \in r \quad$ e $\quad <y_1, z_2, w_1> \in r$.
- Note-se que, como esta definição é simétrica em Z e W, segue-se que $Y \twoheadrightarrow Z$ sse $Y \twoheadrightarrow W$ (isto é $Y \twoheadrightarrow R - Y - Z$).
- Note-se ainda que:
  - Se $Y \to Z$ então $Y \twoheadrightarrow Z$.
  - De facto, se $Y \to Z$ então $z_1 = z_2$, e logo $Y \twoheadrightarrow Z$.

Exemplo (Cont.)

- No nosso exemplo:

$$\begin{array}{lcl} \texttt{nomeCliente} & \twoheadrightarrow & \texttt{moradaCliente} \\ \texttt{nomeCliente} & \twoheadrightarrow & \texttt{nConta} \end{array}$$

- Esta definição formaliza a ideia de que cada valor particular de *Y* (`nomeCliente`) tem associado um conjunto de valores *Z* (`moradaCliente`) e um conjunto de valores de *W* (`nConta`), e que estes dois conjuntos são independentes.

- Se são independentes, porque não metê-los em relações separadas?

Uso de Dependências Multi-valor

- Usam-se dependências multi-valor para:
  1. Testar relações, para verificar se são ou não relações válidas, dado um conjunto de dependência multi-valor.
  2. Especificar restrições no conjunto de (instâncias) de relações válidas. Assim, só devemos ter relações que satisfaçam o conjunto (pré-definido) de dependências funcionais e multi-valor.

- Se uma relação *r* não satisfizer uma dada dependência multi-valor, então é sempre possível construir uma relação *r′*, por adição de tuplos em *r*, que satisfaz a dependência.

Teoria de Dependências Multi-valor

- Da definição de dependência multi-valor, podemos demonstrar:
  - Se $\alpha \rightarrow \beta$, então $\alpha \twoheadrightarrow \beta$, isto é, toda a dependência funcional é também dependência multi-valor.
  - $\alpha \twoheadrightarrow \beta$ é trivial sse $\beta \subseteq \alpha$ ou $\alpha \bigcup \beta = R$.

- Em geral temos um conjunto *D* de dependências funcionais e dependências multi-valor.

- O fecho D^+ de *D* é o conjunto de todas as dependências funcionais e multi-valor que são implicadas por *D*.
  - Pode calcular-se D^+ a partir de *D*, usando as definições de dependência funcional e multi-valor.
  - Tal como para dependências funcionais, há sistemas de inferência para calcular este fecho.

Inferência com Dependências Multi-valor

- Podem encontrar-se todas as dependências em D^+ por aplicação dos seguintes Axiomas (onde os primeiros 3 são os Axiomas de Armstrong) :

Definição (Axiomas Multi-valor)

- *Se $\beta \subseteq \alpha$ então $\alpha \rightarrow \beta$* *(reflexividade)*
- *Se $\alpha \rightarrow \beta$ então $\gamma\alpha \rightarrow \gamma\beta$* *(aumento)*
- *Se $\alpha \rightarrow \beta$ e $\beta \rightarrow \gamma$ então $\alpha \rightarrow \gamma$* *(transitividade)*
- *Se $\alpha \rightarrow \beta$ então $\alpha \twoheadrightarrow \beta$* *(replicação)*
- *Se $\alpha \twoheadrightarrow \beta$ então $\alpha \twoheadrightarrow R - \beta - \alpha$* *(complemento)*
- *Se $\alpha \twoheadrightarrow \beta$, $\gamma \subseteq R$ e $\delta \subseteq \gamma$ então $\gamma\alpha \twoheadrightarrow \delta\beta$* *(multi-aumento)*
- *Se $\alpha \twoheadrightarrow \beta$ e $\beta \twoheadrightarrow \gamma$ então $\alpha \twoheadrightarrow \gamma - \beta$* *(multi-transitividade)*
- *Se $\alpha \twoheadrightarrow \beta$, $\gamma \subseteq \beta$ e existe $\delta \subseteq R$ tal que $\delta \bigcap \beta = \emptyset$ e $\delta \rightarrow \gamma$ então $\alpha \rightarrow \gamma$* *(coalescência)*

- Este conjunto de axiomas é coerente e completo.

4ª Forma Normal

Definição (4ª Forma Normal)

Um esquema R, com conjunto de dependência funcionais e multi-valor D, está na 4ª Forma Normal, 4NF, se para toda a dependência multi-valor $\alpha \twoheadrightarrow \beta$ pertencente a D^+, onde os atributos de α e β estão em R, pelo menos uma das seguintes condições é verdadeira:

- *$\alpha \twoheadrightarrow \beta$ é trivial, isto é $\beta \subseteq \alpha$ ou $\alpha \cup \beta = R$.*
- *α é super chave de R, isto é $\alpha \rightarrow R$.*

Se um esquema está na 4NF também está na BCNF

Restrição de Dependências Multi-valor

- A restrição do conjunto D a R_i é o conjunto de D_i com:
  - Todas as dependências em D^+ que só contêm atributos de R_i;
  - Todas as dependências multi-valor: $\alpha \twoheadrightarrow (\beta \bigcap R_i)$ onde $\alpha \subseteq R_i$ e $\alpha \twoheadrightarrow \beta \in D^+$.
- Com dependências multi-valor, a decomposição de R em R_1 e R_2 é sem perdas sse pelo menos uma das dependências abaixo pertence a D^+:
  - $R_1 \bigcap R_2 \twoheadrightarrow R_1$;
  - $R_1 \bigcap R_2 \twoheadrightarrow R_2$.

Algoritmo de Decomposição para 4NF

```
resultado := {R};
acabou := falso;
calcular D+;
Seja Di a restrição de D+ a Ri
enquanto (não acabou)
  se existe esquema Ri ∈ resultado que não está na 4NF
então
    Seja α ↠ β não trivial e verdadeira em Ri tal que
      α → Ri ∉ Di, e α ⋂ β = ∅;
    resultado := (resultado − Ri) ⋃ (Ri − β) ⋃ (α, β);
  senão acabou := verdade;
fimenquanto
```

Nota: A decomposição é sem perdas

Exemplo

- $$\begin{aligned} R &= (A, B, C, G, H, I) \\ D &= \{A \twoheadrightarrow B \\ &\quad\; B \twoheadrightarrow HI \\ &\quad\; CG \rightarrow H\} \end{aligned}$$
- R não está na 4NF pois $A \twoheadrightarrow B \in F$ e A não é super-chave de R e $\{A, B\} \neq R$.
- Decomposição:
  1. $R_1 = (A, B)$ (R_1 está na 4NF. A única dep. em R_1 é trivial);
  2. $R_2 = (A, C, G, H, I)$ (R_2 não está na 4NF: $CG \rightarrow H$);
  3. $R_3 = (C, G, H)$ (R_3 está na 4NF);
  4. $R_4 = (A, C, G, I)$ (R_4 não está na 4NF: $A \twoheadrightarrow I$).
     Como $A \twoheadrightarrow B$ e $B \twoheadrightarrow HI$ então $A \twoheadrightarrow HI \in D^+$, e $A \twoheadrightarrow I$ está na restrição de D^+ a R_4.
  5. $R_5 = (A, I)$ (R_5 está na 4NF);
  6. $R_6 = (A, C, G)$ (R_6 está na 4NF).

Ter-se-ia então a decomposição de R em $\{R_1, R_3, R_5, R_6\}$.

4NF e Preservação de Dependências

- Tal como a BCNF, a 4NF pode não preservar as dependências:
  - $R = (A, B, C, G, H, I)$ com $D = \{A \twoheadrightarrow B, B \twoheadrightarrow HI, CG \rightarrow H\}$, foi decomposto em $\{(A, B), (C, G, H), (A, I), (A, C, G)\}$.
  - A dependência $B \twoheadrightarrow HI$ não pode ser testada apenas numa destas relações.
- Aplicam-se aqui as mesmas soluções de compromisso que entre a BCNF e a 3NF:
  - Objectivos numa primeira fase:
    - 4NF;
    - decomposição sem perdas;
    - preservação de dependências.
- Se tal não for possível, então há que optar por uma de duas possíveis soluções:
  - Não preservação de dependências.
  - Alguma redundância:
    - tentar BCNF;
    - se tal ainda não preserva dependências, normalizar para a 3NF.

Mais Formas Normais

As **dependências de junção** generalizam as multi-valor.
Dão origem à **forma normal projecção-junção (PJNF)** (também chamada de 5ª forma normal).

Uma classe ainda mais geral de restrições leva à **forma normal de domínio-chave**.

Problemas com estas restrições muito gerais:

- é difícil raciocinar sobre elas;
- não têm conjuntos coerentes e completos de regras de inferência.

Logo, raramente são usadas.

Visão Global Sobre a Concepção de Bases de Dados

Temos assumido que o esquema R é dado:

- R pode ter sido obtido ao passar um diagrama E-R para tabelas;
- R pode ser uma única relação contendo todos os atributos de interesse para os dados (relação universal). A normalização há-de decompor R em relações mais pequenas;
- R pode ser o resultado de algum design “ad hoc”.

Quando o diagrama E-R está foi concebido de forma cuidadosa, o esquema gerado pode já estar numa dada forma normal. Bastará nesse caso uma simples verificação para constatar se isso é verdade ou não.

Na prática, haverá muitos diagramas E-R imperfeitos que levam a que dependências que queremos impor não tenham o lado esquerdo como chave.

Por exemplo entidade `Empregado` com atributos `codDepartamento` e `moradaDep`, e a dependência `codDepartamento` $\rightarrow$ `moradaDep`.
Num bom esquema `Departamentos` seria um outro conjunto de entidades.

Em algumas situações deste tipo a normalização permite corrigir situações incorrectas.
Noutras ter-se-á de re-desenhar o esquema.