

Bases de Dados

Parte VIII Normalização

Desenho de BDs Relacionais

- Algumas questões sobre o desenho de BDs relacionais:
 - Como é que se desenha uma boa BD relacional?
 - Qual é o critério para quantificar a qualidade e funcionalidade de um modelo relacional?
 - Porque é que um determinado agrupamento dos atributos em relações é melhor do que outro agrupamento?
- A qualidade de um modelo relacional pode ser quantificada de dois pontos de vista:
 - **Lógico ou conceptual:** como é que os utilizadores interpretam o significado das relações e dos seus atributos. Um bom modelo do ponto de vista lógico permite que os utilizadores compreendam claramente o significado dos dados e os possam manipular correctamente.
 - **Implementação:** como é que os tuplos das relações são guardados e manipulados fisicamente na BD. Um bom modelo do ponto de vista da implementação garante uma maior eficiência das operações de acesso aos dados, minimiza o espaço necessário para guardar os tuplos das relações e evita informação incorrecta ou supérflua.

Regras Para o Bom Desenho de BDs Relacionais

- **Regra 1:** Os atributos de uma relação devem representar apenas uma entidade ou um relacionamento.
 - Atributos de entidades ou relacionamentos diferentes devem estar separados o mais possível. Apenas chaves externas devem ser usadas para referenciar outras relações.
 - É mais fácil explicar o significado de uma relação se esta representar apenas uma entidade ou relacionamento. Evita ambiguidades no significado das relações.

- Exemplo de uma boa relação do ponto de vista lógico mas que viola a regra 1:
 - EMP_DEP(NomeEmp, NumBI, Endereço, DataNasc, NumDep, NomeDep, GerenteBI)
- Problemas com a relação EMP_DEP:
 - Os valores dos atributos NomeDep e GerenteBI aparecem repetidos para os empregados que trabalham num mesmo departamento.

Regras Para o Bom Desenho de BDs Relacionais

- **Regra 2:** Evitar a possibilidade de ocorrerem anomalias nas operações de inserção, remoção ou alteração.
 - Se por razões de eficiência isso não for possível, garantir que os utilizadores/programas que manipulam a BD conhecem essas anomalias e as evitam.

- Exemplo de anomalias de inserção em EMP_DEP:
 - Não é possível inserir um novo departamento a menos que seja associado a um empregado.
 - Ao inserir um empregado é necessário garantir que os valores dos atributos NomeDep e GerenteBI são consistentes com os dos restantes empregados desse departamento.
- Exemplo de anomalias de remoção em EMP_DEP:
 - Se removermos o último empregado para um determinado departamento, então a informação desse departamento também é removida.
- Exemplo de anomalias de alteração em EMP_DEP:
 - A alteração do nome de um departamento leva a que essa alteração tenha que ser feita sobre todos os tuplos dos empregados que nele trabalham.

Regras Para o Bom Desenho de BDs Relacionais

- **Regra 3:** Evitar atributos que possam ter valores NULL numa grande parte dos tuplos duma relação.
 - Colocar esse tipo de atributos em relações separadas juntamente com a chave primária.
 - Minimiza o espaço necessário para guardar os tuplos da relação e evita problemas no cálculo de funções de agregações sobre esses atributos.

- Exemplo:
 - Se apenas 5% dos empregados tiverem gabinete individual não faz sentido incluir um atributo NumGabinete na relação EMP_DEP. Uma melhor solução será criar uma relação GABINETE(EmpBI, NumGabinete) para guardar essa informação.

Regras Para o Bom Desenho de BDs Relacionais

- **Regra 4:** Evitar relações que tenham atributos relacionados que não são sejam combinações do tipo chave externa com chave primária.
 - Operações de junção sobre esses atributos poderão originar tuplos falsos. Não verificam a **propriedade de junção-não-aditiva** (ou junção-sem-perdas de informação).

- Considere as seguintes relações e assuma que a localização de cada projecto é única, ou seja, dois projectos diferentes têm sempre localizações diferentes:
 - TRAB_PROJ(EmpBI, NumProj, Horas, NomeProj, LocalizaçãoProj)
 - EMP_LOC(NomeEmp, LocalizaçãoProj)

- Problemas com as relações TRAB_PROJ e EMP_LOC:
 - A operação de junção natural TRAB_PROJ * EMP_LOC dá origem a mais tuplos do que aqueles que seriam obtidos pela junção das tabelas originais TRABALHA_EM, PROJECTO e EMPREGADO. Note que isto acontece mesmo com localizações únicas para cada projecto.

Normalização de Relações

- Processo de análise que minimiza redundância de dados e minimiza anomalias nas operações de modificação dos dados. As relações que não satisfazem certas propriedades – **formas normais** – são sucessivamente decompostas em relações mais pequenas de modo a satisfazerem as propriedades pretendidas (Codd 1972).
- As formas normais são como que orientações para o desenho de boas relações. As formas normais existentes são:
 - 1NF – Primeira forma normal
 - 2NF – Segunda forma normal
 - 3NF – Terceira forma normal
 - BCNF – Forma normal de Boyce–Codd
 - 4NF – Quarta forma normal
 - 5NF – Quinta forma normal
- Nem sempre é necessário normalizar uma BD até à última forma normal (por vezes, 3NF ou BCNF é suficiente).

1NF – Primeira Forma Normal

- Um esquema relacional está na primeira forma normal se todos os atributos forem atómicos (não divisíveis).
- Normalização 1NF
 - Decompor atributos compostos em atributos atómicos.
 - O atributo Nome pode ser decomposto em (NomeP, NomeF).
 - O atributo Endereço pode ser decomposto em (Morada, Cidade, CódigoPostal).
 - Decompor atributos multi-valor em relação com chave externa.
 - A relação DEPARTAMENTO(Nome, Num, {Localização}) pode ser decomposta em DEPARTAMENTO(Nome, Num, Localização), mas a melhor solução é decompor em DEPARTAMENTO(Nome, Num) e LOCALIZAÇÕES_DEP(NumDep, Localização) pois evita redundância.

Dependências Funcionais

- **Dependência funcional (FD)** é uma restrição entre 2 conjuntos de atributos.
- Seja $R(A_1, A_2, \dots, A_n)$ um esquema relacional e sejam X e Y dois subconjuntos de atributos de R . Diz-se que X determina funcionalmente Y (ou que Y depende funcionalmente de X), representado por $X \rightarrow Y$, se quaisquer dois tuplos de R que têm os mesmos valores em X também têm os mesmos valores em Y .

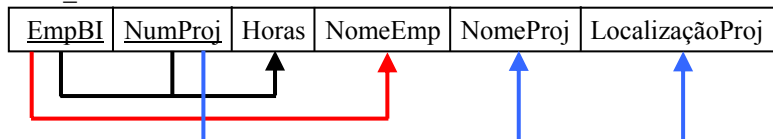
$$X \rightarrow Y \equiv \forall t_1, t_2 \in r(R): t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$

- Interpretação:
 - Os valores da componente Y de tuplos de R dependem dos valores da componente X .
 - Os valores da componente X de tuplos de R determinam os valores da componente Y .
- As chaves de uma relação são casos particulares de dependências funcionais. Se X for uma chave de R então $X \rightarrow Y$ para qualquer subconjunto Y de atributos de R .
- O facto de $X \rightarrow Y$ em R nada permite concluir acerca de $Y \rightarrow X$ em R .

Dependências Funcionais

- Considere a seguinte relação:
 - EMP_PROJ(EmpBI, NumProj, Horas, NomeEmp, NomeProj, LocalizaçãoProj)
- Dependências funcionais da relação EMP_PROJ:
 - EmpBI \rightarrow NomeEmp
 - NumProj \rightarrow {NomeProj, Localização}
 - {EmpBI, NumProj} \rightarrow Horas

EMP_PROJ

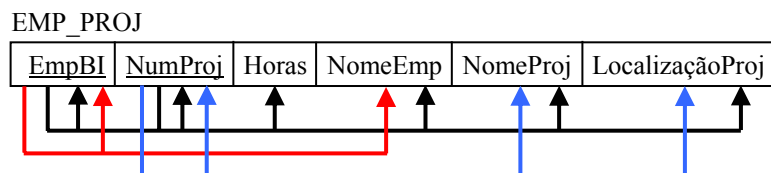


Dependências Funcionais

- Normalmente, a partir de um conjunto de dependências funcionais é possível inferir outras dependências funcionais.
- Regras de inferência de Armstrong (1974):
 - **Regra Reflexiva:** Se $X \supseteq Y$ então $X \rightarrow Y$ (ou $X \rightarrow X$).
 - **Regra Aditiva:** Se $X \rightarrow Y$ então $XZ \rightarrow YZ$ (ou se $X \rightarrow Y$ então $XZ \rightarrow Y$).
 - **Regra Transitiva:** Se $X \rightarrow Y$ e $Y \rightarrow Z$ então $X \rightarrow Z$.
- O **fecho** de um conjunto de dependências funcionais F é o conjunto F^+ de todas as dependências funcionais que podem ser inferidas a partir de F .
- O **fecho** de um conjunto de atributos X de F é o conjunto X^+ de todas os atributos que podem ser inferidos a partir de X .
- F^+ e X^+ podem ser calculados por aplicação sucessiva das regras de Armstrong.

Dependências Funcionais

- Fecho das dependências funcionais da relação EMP_PROJ:
 - $\{\text{EmpBI}\}^+ \rightarrow \{\text{EmpBI}, \text{NomeEmp}\}$
 - $\{\text{NumProj}\}^+ \rightarrow \{\text{NumProj}, \text{NomeProj}, \text{Localização}\}$
 - $\{\text{EmpBI}, \text{NumProj}\}^+ \rightarrow \{\text{EmpBI}, \text{NumProj}, \text{Horas}, \text{NomeEmp}, \text{NomeProj}, \text{LocalizaçãoProj}\}$



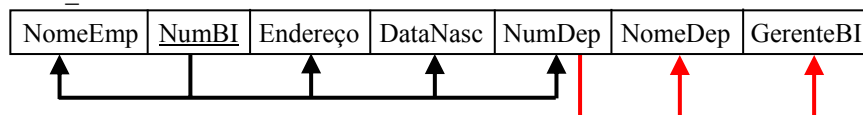
Dependências Funcionais

- Uma dependência funcional $X \rightarrow Y$ diz-se **parcial** se a remoção de algum atributo de X não deixar de determinar funcionalmente Y .
 $X \rightarrow Y$ é uma dependência funcional parcial se $\exists A \in X: (X - A) \rightarrow Y$
- Uma dependência funcional $X \rightarrow Y$ diz-se **completa** (ou **não parcial**) se a remoção de um qualquer atributo de X deixar de determinar funcionalmente Y .
 $X \rightarrow Y$ é uma dependência funcional completa se $\forall A \in X: (X - A) \not\rightarrow Y$
- Exemplos para a relação EMP_PROJ:
 - $\{\text{EmpBI}, \text{NumProj}\} \rightarrow \text{NomeEmp}$ é uma dependência funcional parcial porque $\text{EmpBI} \rightarrow \text{NomeEmp}$ também se verifica.
 - $\{\text{EmpBI}, \text{NumProj}\} \rightarrow \text{Horas}$ é uma dependência funcional completa porque nem $\text{EmpBI} \rightarrow \text{Horas}$ nem $\text{NumProj} \rightarrow \text{Horas}$ se verificam.

Dependências Funcionais

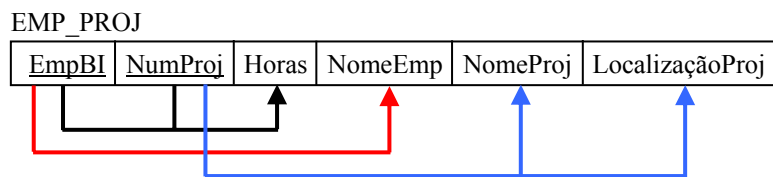
- Uma dependência funcional $X \rightarrow Y$ diz-se **transitiva** se existir um conjunto de atributos não-chave que depende funcionalmente de X e determina funcionalmente Y .
 $X \rightarrow Y$ é uma dependência funcional transitiva se $\exists Z$ não-chave: $X \rightarrow Z$ e $Z \rightarrow Y$.
- Exemplo para a relação EMP_DEP:
 - $\text{NumBI} \rightarrow \{\text{NomeDep}, \text{GerenteBI}\}$ é uma dependência funcional transitiva porque $\text{NumBI} \rightarrow \text{NumDep}$, $\text{NumDep} \rightarrow \{\text{NomeDep}, \text{GerenteBI}\}$ e NumDep não pertence a nenhuma chave candidata de EMP_DEP.

EMP_DEP



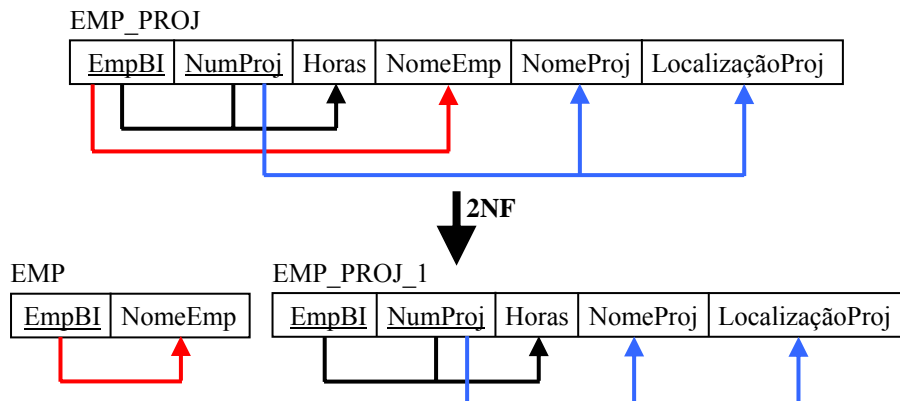
2NF – Segunda Forma Normal

- Um esquema relacional está na segunda forma normal se:
 - Está na 1NF.
 - Todos os atributos não-chave dependem por completo da chave primária.
- Na relação EMP_PROJ, as dependências funcionais $\text{EmpBI} \rightarrow \text{NomeEmp}$ e $\text{NumProj} \rightarrow \{\text{NomeProj}, \text{LocalizaçãoProj}\}$ violam a 2NF.



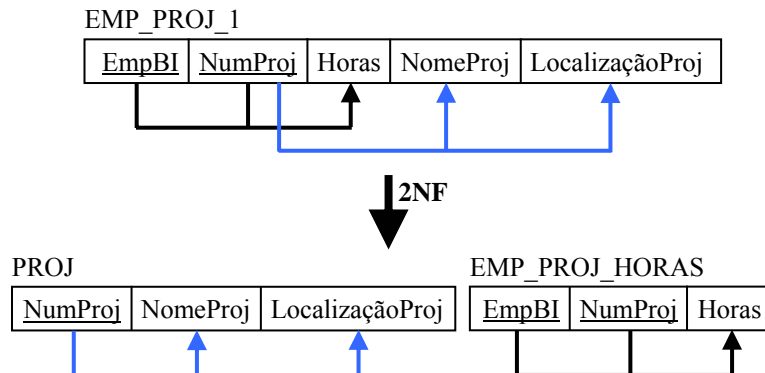
Normalização 2NF

- Decompor em relações com chave externa de forma a que, em cada relação, todos os atributos não-chave dependam por completo da chave primária correspondente.



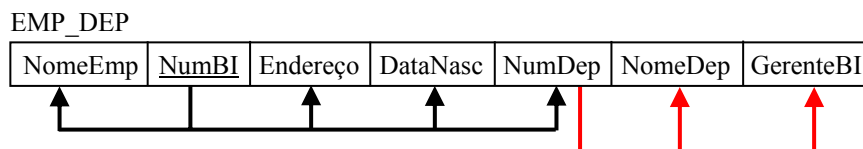
Normalização 2NF

- Decompor em relações com chave externa de forma a que, em cada relação, todos os atributos não-chave dependam por completo da chave primária correspondente.



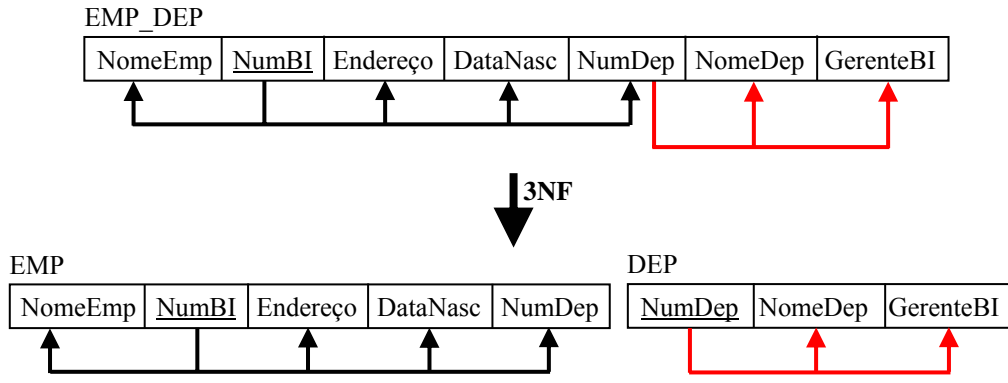
3NF – Terceira Forma Normal

- Um esquema relacional está na terceira forma normal se:
 - Está na 2NF.
 - Nenhum atributo não-chave depende por transitividade da chave primária.
- Na relação EMP_DEP, $\text{NumBI} \rightarrow \{\text{NomeDep}, \text{GerenteBI}\}$ é uma dependência transitiva que viola a 3NF.



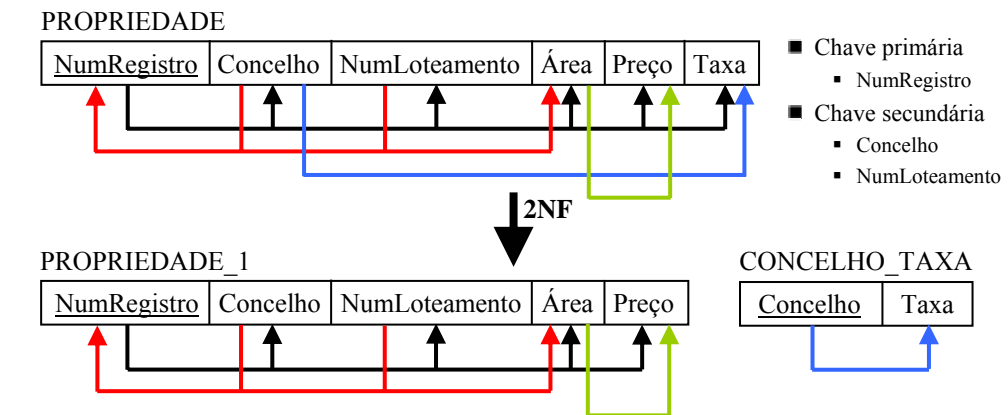
Normalização 3NF

- Decompor em relações de forma a que, em cada relação, nenhum atributo não-chave dependa por transitividade da chave primária correspondente.



2NF – Definição Mais Geral (Múltiplas Chaves)

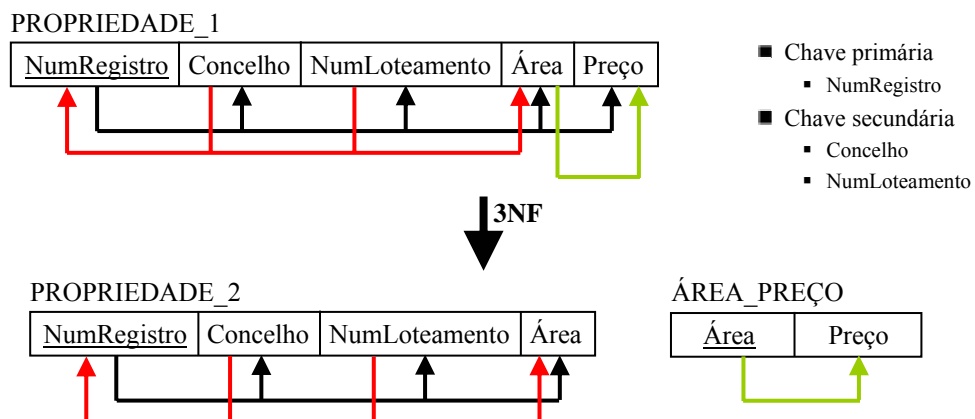
- Um esquema relacional está na segunda forma normal se todos os atributos não-chave não dependem parcialmente de qualquer chave.



3NF – Definição Mais Geral (Múltiplas Chaves)

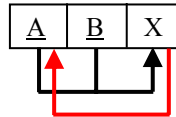
- Um esquema relacional está na terceira forma normal se para qualquer dependência funcional $X \rightarrow A$ com $A \notin X$, ou (i) X é uma superchave ou (ii) A é um atributo de uma qualquer chave.
- Dito de outro modo, um esquema relacional está na terceira forma normal se para todos os atributos não-chave A com $X \rightarrow A$ e $A \notin X$, X é uma superchave. Se X não é uma superchave, então X ou faz parte de uma chave (logo $X \rightarrow A$ é uma dependência parcial e viola 2NF) ou X não faz parte de qualquer chave ($S \rightarrow X$ para alguma chave S e logo $S \rightarrow A$ é uma dependência transitiva).
- Definição alternativa: Um esquema relacional está na terceira forma normal se todos os atributos não-chave:
 - não dependem parcialmente de qualquer chave (2NF).
 - não dependem por transitividade de qualquer chave.

3NF – Definição Mais Geral (Múltiplas Chaves)



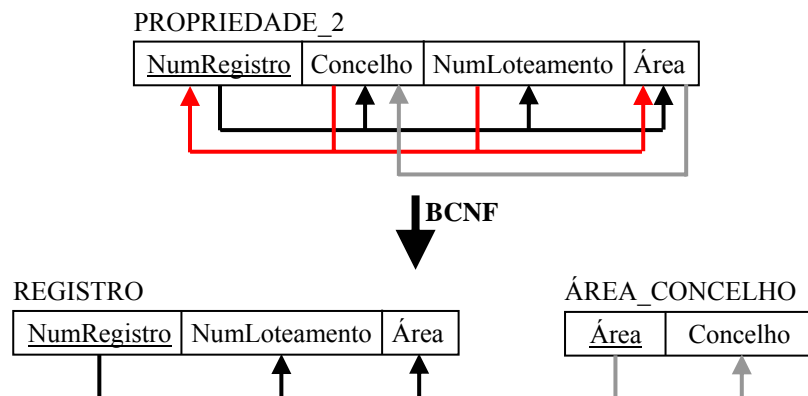
BCNF – Forma Normal de Boyce–Codd

- Um esquema relacional está na forma normal de Boyce–Codd se para qualquer dependência funcional $X \rightarrow A$ com $A \notin X$, X é uma superchave.
- A diferença para 3NF é a inexistência da possibilidade de A ser um atributo de uma qualquer chave. Note-se que se um esquema relacional está na BCNF então também está na 3NF. O inverso pode não ser verdadeiro.
- Na prática, quando um esquema relacional está na 3NF, normalmente também está na BCNF. A exceção é quando existe uma dependência $X \rightarrow A$ com X sem ser uma superchave e A sendo um atributo de uma chave.



Normalização BCNF

- A normalização BCNF não verifica a propriedade de preservação das dependências, podendo originar a perda de dependências funcionais.



Decomposição com Junção-Não-Aditiva

- Seja R um esquema relacional e seja $D=\{R_1, R_2\}$ uma decomposição de R . Diz-se que D constitui uma **decomposição com junção-não-aditiva** relativamente a um conjunto F de dependências funcionais se para qualquer estado $R = R_1 * R_2$, ou seja, as **operações de junção não originam tuplos falsos**.
- Uma decomposição $D=\{R_1, R_2\}$ de R verifica a **propriedade de junção-não-aditiva** relativamente a um conjunto F de dependências funcionais se e só se:
 - $(R_1 \cap R_2) \rightarrow (R_1 - R_2) \in F^+$, ou
 - $(R_1 \cap R_2) \rightarrow (R_2 - R_1) \in F^+$.
- As normalizações 2NF, 3NF e BCNF devem verificar sempre a propriedade de junção-não-aditiva.

Dependências Multi-Valor

- **Dependência multi-valor (MVD)** é uma restrição entre 2 atributos multi-valor e independentes da mesma relação. Para manter a relação consistente, uma MVD obriga a repetir todos os valores de um atributo para cada valor do outro atributo.
- Considere a relação EMP_PROJS_DEPS que relaciona cada empregado com os projectos em que trabalha e os seus dependentes. Um empregado pode trabalhar em vários projectos e ter vários dependentes. A dependência multi-valor resulta do facto de juntarmos dois relacionamentos 1:N na mesma relação.

EMP_PROJS_DEPS	NomeEmp	NomeProj	NomeDep
	Silva	ProjX	João
	Silva	ProjX	Maria
	Silva	ProjY	João
	Silva	ProjY	Maria

Dependências Multi-Valor

- Seja $R(A_1, A_2, \dots, A_n)$ um esquema relacional e sejam X e Y dois subconjuntos de atributos de R . Diz-se que X multi-determina Y , representado por $X \twoheadrightarrow Y$, se os valores da componente Y de tuplos de R dependem apenas do valor da componente X , mas para cada valor da componente X , os valores da componente Y aparecem repetidos para cada valor distinto da componente Z ($Z = R - (X \cup Y)$).

$$\exists t_1, t_2 \in r(R): t_1[X] = t_2[X] \Rightarrow \exists t_3, t_4 \in r(R): t_3[X] = t_4[X] = t_1[X] = t_2[X],$$

$$t_3[Y] = t_1[Y] \text{ e } t_4[Y] = t_2[Y],$$

$$t_3[Z] = t_2[Z] \text{ e } t_4[Z] = t_1[Z],$$

onde $Z = R - (X \cup Y)$.

R	X	Y	Z
t1	X1	Y1	Z1
t2	X1	Y2	Z2
t3	X1	Y1	Z2
t4	X1	Y2	Z1

Dependências Multi-Valor

- Regras de inferência para dependências funcionais (FD) e multi-valor (MVD):
 - **Regra Reflexiva para FDs:** Se $X \supseteq Y$ então $X \rightarrow Y$.
 - **Regra Aditiva para FDs:** Se $X \rightarrow Y$ então $XZ \rightarrow YZ$.
 - **Regra Transitiva para FDs:** Se $X \rightarrow Y$ e $Y \rightarrow Z$ então $X \rightarrow Z$.
 - **Regra Complementar para MVDs:** Se $X \twoheadrightarrow Y$ então $X \twoheadrightarrow (R - (X \cup Y))$.
 - **Regra Aditiva para MVDs:** Se $X \twoheadrightarrow Y$ e $W \supseteq Z$, então $WX \twoheadrightarrow YZ$.
 - **Regra Transitiva para MVDs:** Se $X \twoheadrightarrow Y$ e $Y \twoheadrightarrow Z$, então $X \twoheadrightarrow (Z - Y)$.
 - **Regra Replicação para FDs em MVDs:** Se $X \rightarrow Y$, então $X \twoheadrightarrow Y$.
 - **Regra Aglutinante para FDs e MVDs:** Se $X \rightarrow Y$ e $\exists W: W \cap Y$ é vazio, $W \rightarrow Z$ e $Y \supseteq Z$, então $X \rightarrow Z$.
- O **fecho** de um conjunto de F de FDs e MVDs é o conjunto F^+ de todas as FDs e MVDs que podem ser inferidas por aplicação sucessiva das regras de inferência.

4NF – Quarta Forma Normal

- Um esquema relacional R está na quarta forma normal relativamente a um conjunto F de dependências funcionais e multi-valor se para cada dependência multi-valor não-trivial $X \twoheadrightarrow Y$ em F^+ , X é uma superchave.
- Uma dependência multi-valor $X \twoheadrightarrow Y$ diz-se **trivial** em R quando não especifica nenhuma restrição com significado, ou seja, quando $Y \subseteq X$ ou $X \cup Y = R$.
- A 4NF evita os problemas de consistência e redundância relacionados com as dependências multi-valor.
- As dependências multi-valor da relação EMP_PROJS_DEPS violam a 4NF:
 - NomeEmp \twoheadrightarrow NomeProj
 - NomeEmp \twoheadrightarrow NomeDep

Normalização 4NF

EMP_PROJS_DEPS

<u>NomeEmp</u>	<u>NomeProj</u>	<u>NomeDep</u>
----------------	-----------------	----------------



EMP_PROJS

<u>NomeEmp</u>	<u>NomeProj</u>
----------------	-----------------

EMP_DEPS

<u>NomeEmp</u>	<u>NomeDep</u>
----------------	----------------

EMP_PROJS	NomeEmp	NomeProj
	Silva	ProjX
	Silva	ProjY

EMP_DEPS	NomeEmp	NomeDep
	Silva	João
	Silva	Maria

Normalização 4NF

- Sempre que decomposmos um esquema relacional R em dois esquemas relacionais $R_1 = (X \cup Y)$ e $R_2 = (R - Y)$ com base numa MVD $X \twoheadrightarrow Y$, então a decomposição verifica a propriedade de junção-não-aditiva.
- Dois esquemas relacionais R_1 e R_2 constituem uma decomposição com junção-não-aditiva de R relativamente a um conjunto F de dependências funcionais e multi-valor se e só se:
 - $(R_1 \cap R_2) \twoheadrightarrow (R_1 - R_2) \in F^+$, ou
 - $(R_1 \cap R_2) \twoheadrightarrow (R_2 - R_1) \in F^+$.
- Por vezes, pode acontecer que não existe uma decomposição com junção-não-aditiva em apenas duas relações, mas existe em mais do que duas relações. Isso pode verificar-se mesmo que nenhuma FD viole as formas normais até à BCNF e nenhuma MVD não-trivial viole a 4NF. Sobra então o outro tipo de dependência que nos conduz à 5NF.

Dependências de Junção

- **Dependência de junção (JD)** é uma restrição entre tuplos.
- Seja R um esquema relacional e seja R_1, R_2, \dots, R_n uma decomposição de R. Diz-se que $JD(R_1, R_2, \dots, R_n)$ é uma dependência de junção de R se qualquer estado de R tem uma decomposição com junção-não-aditiva em R_1, R_2, \dots, R_n , ou seja:

$$r(R) = * (r(R_1), r(R_2), \dots, r(R_n))$$
- Uma dependência de junção $JD(R_1, R_2, \dots, R_n)$ diz-se **trivial** em R quando não especifica nenhuma restrição com significado, ou seja, quando um dos R_i é igual a R.
- Uma MVD é um caso especial de JD para $n = 2$. Ou seja, a $JD(R_1, R_2)$ implica a MVD $(R_1 \cap R_2) \twoheadrightarrow (R_1 - R_2)$.

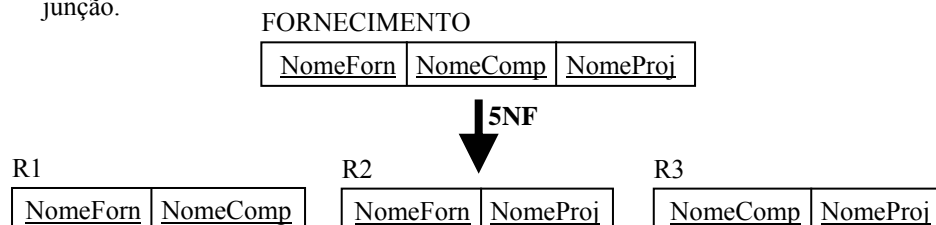
Dependências de Junção

- Considere a relação FORNECIMENTO(NomeForn, NomeComp, NomeProj) e suponha que se verifica a seguinte restrição: se um fornecedor F fornece a componente C e se um projecto P encomenda a componente C e se o fornecedor F fornece o projecto P, então o fornecedor F fornece a componente C ao projecto P.
- A restrição anterior obriga que os dois últimos tuplos da tabela ao lado existam para qualquer estado da relação FORNECIMENTO em que os primeiros cinco tuplos também existem.
- Isto significa que existe uma dependência de junção $JD(R_1, R_2, R_3)$ em que $R_1(\text{NomeForn}, \text{NomeComp})$, $R_2(\text{NomeForn}, \text{NomeProj})$ e $R_3(\text{NomeComp}, \text{NomeProj})$.

NomeForn	NomeComp	NomeProj
F1	C2	P3
F2	C1	P2
F2	C3	P1
F3	C2	P2
F3	C1	P1
F2	C1	P1
F3	C1	P2

5NF – Quinta Forma Normal

- Um esquema relacional R está na quinta forma normal relativamente a um conjunto F de dependências funcionais, multi-valor e de junção se para cada dependência de junção não-trivial $JD(R_1, R_2, \dots, R_n)$ em F^+ , cada R_i é uma superchave.
- A 5NF evita os problemas de consistência relacionados com as dependências de junção.



- Note que a junção natural de quaisquer duas das três relações pode originar tuplos falsos, mas a junção das 3 relações não!

Normalização 5NF

FORNECIMENTO	NomeForn	NomeComp	NomeProj
	F1	C2	P3
	F2	C1	P2
	F2	C3	P1
	F3	C2	P2
	F3	C1	P1
	F2	C1	P1
	F3	C1	P2

R1	NomeForn	NomeComp
	F1	C2
	F2	C1
	F2	C3
	F3	C2
	F3	C1

R2	NomeForn	NomeProj
	F1	P3
	F2	P2
	F2	P1
	F3	P2
	F3	P1

R3	NomeComp	NomeProj
	C2	P3
	C1	P2
	C3	P1
	C2	P2
	C1	P1