

BCNF versus 3NF (exemplo)

- ▶ Considere o esquema Gestores = (balcao, cliente, gestor), com as dependências:

gestor → balcao
cliente, balcao → gestor

- ▶ Está na 3NF:
 - ▶ {cliente, balcao} é chave de Gestores;
 - ▶ {balcao} - {gestor} = {balcao} ⊆ na chave candidata de Gestores (que é {cliente, balcao}).

Mas:

balcao	cliente	gestor
Lx	Maria	Ana
Cbr	Maria	João
Lx	Pedro	Ana
Lx	José	Ana
Cbr	null	Mário

- ▶ Necessidade de ter um campo com null para associar gestores (ainda) sem clientes, a balcões;
- ▶ Dados redundantes (balcao).

2014/02/15 (v91)
293 / 311

BCNF ou 3NF?

- ▶ Objectivos do design, numa primeira fase:
 - ▶ BCNF;
 - ▶ Decomposição sem perdas;
 - ▶ Preservação de dependências.
- ▶ Se tal não for possível, então há que optar por:
 - ▶ Não preservação de dependências;
 - ▶ Alguma redundância (devido ao uso da 3NF).

O SQL não fornece nenhuma forma directa de impor dependências que não sejam super-Chaves. Pode fazer-se usando `assertion` mas isso é muito ineficiente.

Mesmo quando a decomposição preserva as dependências, com o SQL não é possível testar de forma eficiente dependências cujo lado esquerdo não seja uma chave.

2014/02/15 (v91)
294 / 311

Dependências Multi-valor - Motivação

Há esquemas que estão na BCNF, que preservam as dependências, mas que, mesmo assim, não parecem estar suficientemente normalizadas.

Considere o seguinte esquema para o exemplo do banco:

depositante(nConta, nomeCliente, moradaCliente)

Se tivermos a dependência funcional

nomeCliente → moradaCliente

então este esquema não está na BCNF.

Mas o banco quer deixar que um cliente possa ter mais do que uma morada, isto é, não quer impor esta dependência. Nesse caso, `depositante` já está na BCNF.

2014/02/15 (v91)
295 / 311

Motivação

Depositante

nConta	nomeCliente	moradaCliente
1	Carlos	morada1
1	Carlos	morada2
1	José	morada3
2	Carlos	morada1
2	Carlos	morada2
2	Maria	morada1
2	Maria	morada4
3	José	morada3
3	Maria	morada1
3	Maria	morada4

Está na BCNF (verificar).

Mas:

- ▶ Redundância!!!
- ▶ Problemas na inserção — Se quisermos adicionar uma nova morada (morada5) para o José, é necessário introduzir 2 tuplos:

(1, José, morada5) (3, José, morada5)

2014/02/15 (v91)
296 / 311

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!

2014/02/15 (v91)
297 / 311

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:

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

2014/02/15 (v91)
298 / 311

Dependências Multi-valor (Cont.)

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

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

- ▶ Seja $R = \{\alpha, \beta, \gamma\}$.
- ▶ Diz-se que $\alpha \twoheadrightarrow \beta$ (α multi-determina β) sse para todas as possíveis $r(R)$
Se $\langle \alpha_1, \beta_1, \gamma_1 \rangle \in r$ e $\langle \alpha_1, \beta_2, \gamma_2 \rangle \in r$
então $\langle \alpha_1, \beta_1, \gamma_2 \rangle \in r$ e $\langle \alpha_1, \beta_2, \gamma_1 \rangle \in r$.
- ▶ Note-se que, como esta definição é simétrica em β e γ , segue-se que $\alpha \twoheadrightarrow \beta$ sse $\alpha \twoheadrightarrow \gamma$ (isto é $\alpha \twoheadrightarrow R - \alpha - \beta$).
- ▶ Note-se ainda que:
 - ▶ Se $\alpha \twoheadrightarrow \beta$ então $\alpha \twoheadrightarrow \beta$.
 - ▶ De facto, se $\alpha \twoheadrightarrow \beta$ então $\beta_1 = \beta_2$, e logo $\alpha \twoheadrightarrow \beta$.

2014/02/15 (v91)
299 / 311

Exemplo

- ▶ Para Depositante tem-se que: nomeCliente \twoheadrightarrow 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 nConta = n_2 e moradaCliente = m_1 ;
 - ▶ outro com nConta = n_1 e moradaCliente = m_2 .

Depositante

nConta	nomeCliente	moradaCliente
1	Carlos	morada1
1	Carlos	morada2
1	José	morada3
2	Carlos	morada1
2	Carlos	morada2
2	Maria	morada1
2	Maria	morada4
3	José	morada3
3	Maria	morada1
3	Maria	morada4

2014/02/15 (v91)
300 / 311

Exemplo (Cont.)

- ▶ No nosso exemplo:

nomeCliente \rightarrow moradaCliente
nomeCliente \rightarrow nConta

- ▶ Esta definição formaliza a ideia de que cada valor particular de α (nomeCliente) tem associado um conjunto de valores β (moradaCliente) e um conjunto de valores de γ (nConta), e que estes dois conjuntos são independentes.
- ▶ Se são independentes, porque não metê-los em relações separadas?

2014/02/15 (v91)
301/311

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.

2014/02/15 (v91)
302/311

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 $\alpha \cup \beta = R$.
- ▶ 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.

2014/02/15 (v91)
303/311

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 $\delta \cap \beta = \emptyset$ e $\delta \rightarrow \gamma$ então $\alpha \rightarrow \gamma$ (coalescência)

- ▶ Este conjunto de axiomas é coerente e completo.

2014/02/15 (v91)
304/311