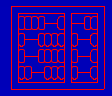
Normalização de BDs relacionais

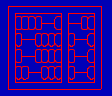


Motivação

Esses conceitos teóricos têm objetivos práticos:

- projetar BDs possivelmente sem inconsistências,
- com menos redundância de informação,
- determinar certos tipos de restrições sobre atributos de uma relação,
- determinar as possíveis chaves de uma relação.

Normalização de BDs relacionais



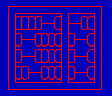
Exemplo de BD relacional *não normalizada*:

Fornecedores(numf, nome, end, tel, contato, CNPJ*, num-mat, prazo, preço)*

tem uma linha para cada possível material que o fornecedor fornece, o que leva a problemas:

- redundância: nome, endereço e CNPJ repetidos inúmeras vezes;
- atualização: endereço do fornecedor muda e alguma linha onde ele aparece não é atualizada;

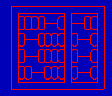
Normalização de BDs relacionais



- inserção de fornecedor sem materiais homologados: valores *nulos* teriam que ser inseridos; quando materiais forem homologados, a linha com esses nulos poderia ser esquecida na tabela;

- remoção sucessiva de materiais fornecidos: pode levar à perda completa dos dados do fornecedor.

Normalização de BDs relacionais



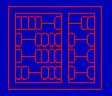
Problema básico: atributos de CEs distintos colocados na mesma tabela: Fornecimentos de materiais com atributos de Fornecedores.

Solução: eliminar informação redundante.

Custo: proliferação de tabelas (com menos colunas) e processamento para gerar relatórios com dados de diversas tabelas.

UNICAMP

Normalização de BDs relacionais



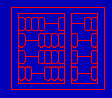
Para que serve a teoria:

em projetos *top down*: projetar desde o início tabelas sem os problemas citados;

em projetos *bottom up*: corrigir maus projetos, *decompondo* de forma apropriada relações *mal projetadas*, isto é, *não normalizadas*.

UNICAMP

O conceito de dependência funcional



Notação:

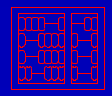
Dado o esquema $R(A_1, A_2, \ldots, A_n)$,

letras do final do alfabeto, X, Y, Z... denotarão subconjuntos de atributos de R;

atributos compostos como {A,B} e {B,A} são idênticos e denotados por AB ou BA.

Definição: X determina funcionalmente Y ($X \rightarrow Y$) se dadas duas tuplas de R em que os valores de X são iguais, então necessariamente os valores de Y também serão iguais.

O conceito de dependência funcional



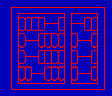
Dependência funcional é uma propriedade do projeto da base de dados, isto é, da semântica dos dados armazenados na relação.

Definição: $A_i, A_j, \ldots, A_k$ é uma *chave* de uma relação R se e somente se:

(i) $A_i, A_j, \ldots, A_k$ determina funcionalmente todos os outros atributos de R;

(ii) nenhum subconjunto de $A_i, A_j, \ldots, A_k$ determina funcionalmente os atributos restantes de R (minimalidade).

O conceito de dependência funcional



Definição: Uma *superchave* de R é um subconjunto dos atributos de R que contém uma chave.

Toda chave é uma superchave, mas nem toda superchave é uma chave!

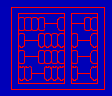
(superchaves podem violar a parte (ii) da definição de chave vista acima).

Definição: *Atributo não primo (ANP)* - não está contido em nenhuma chave de R.

Atributo primo é um atributo contido em alguma chave de R.

UNICAMP

Regras de inferência de Armstrong



a1 Se $X \supseteq Y$ então $X \rightarrow Y$ (*regra da reflexividade*)

a2 Se $X \rightarrow Y$ então para qualquer Z, $XZ \rightarrow YZ$ (*regra aumentativa*)

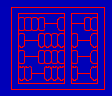
a3 Se $X \rightarrow Y$ e $Y \rightarrow Z$ então $X \rightarrow Z$ (*regra da transitividade*)

As regras acima são *sólidas*: só levam a deduções de dependências funcionais verdadeiras;

são *completas*: dado um conjunto de dfs F válidas para R, permitem obter *todas* as dfs que podem ser inferidas a partir de F e também válidas para R.

UNICAMP

Regras de inferência de Armstrong



As seguintes regras são simples de provar:

Regra da união: se $X \rightarrow Y$ e $X \rightarrow Z$ então $X \rightarrow YZ$

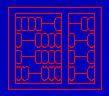
Prova: $X \rightarrow YX$ por a2,
e, como $X \rightarrow Z$, $XY \rightarrow ZY$ por a2,
logo $X \rightarrow ZY$ por a3 e como $ZY = YZ$,
segue o resultado

Regra da decomposição:

Se $X \rightarrow WZ$ então $X \rightarrow W$ e $X \rightarrow Z$

A prova segue diretamente de a1 ($WZ \rightarrow W$) seguida de a3: $X \rightarrow W$. Idem para provar $X \rightarrow Z$.

Regras de inferência de Armstrong



As regras de união e decomposição nos permitem provar:

Lema 1: Regra geral da união/decomposição:

$X \rightarrow A_1, A_2, \ldots, A_n$ se e somente se
$X \rightarrow A_1, X \rightarrow A_2, \ldots, X \rightarrow A_n$,

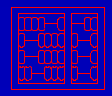
o que nos permite simplificar certas provas, supondo que todas as dfs são do tipo

$X \rightarrow A_i$, onde A_i é um atributo simples de R;

nesse caso dizemos que F está na *forma canônica*.

UNICAMP

Regras de inferência de Armstrong



Uma df do tipo $X \rightarrow A_i$, onde X contém mais de um atributo simples, em geral não pode ser *decomposta* em dfs válidas com menos atributos à esquerda!

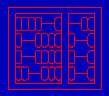
As regras de inferência de Armstrong são úteis em dois aspectos práticos:

1. dadas algumas dfs válidas para uma relação R, elas nos permitem descobrir outras dfs também válidas para R;

2. elas podem nos ajudar a descobrir uma ou mais chaves de uma relação.

UNICAMP

Regras de inferência de Armstrong

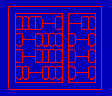


Definição: A df obtida pelo axioma a1 é também chamada de *dependência funcional trivial*

Chamaremos de *não trivial* toda df que não é trivial.

UNICAMP

Formas normais (FN) de relações



São regras crescentemente restritivas para esquemas de relações *normalizadas*:

relações obedecendo à FN n devem obedecer à FN n - 1, mas não vice-versa.

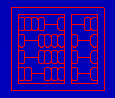
Cada forma normal *mais alta* elimina anomalias não cobertas pela forma normal *mais baixa*.

A 1ª FN é considerada como a própria definição do MR: nehuma relação possui atributos multivalorados nem relações aninhadas.

Suporemos no que se segue, que todas as dfs dadas de uma relação são não triviais.

UNICAMP

Segunda Forma Normal



Dependência funcional parcial:

um atributo A é *dependente funcional parcial* de um atributo X, se $X \rightarrow A$ e existe um subconjunto próprio Y de X (isto é, com menos atributos que X) tal que $Y \rightarrow A$

Def: uma relação R está na 2ª FN se nenhum ANP de R é dependente funcional parcial de qualquer chave de R.

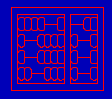
Parece complicado, mas não é! Exemplo:

$Forntos(\underline{numf, num\_mat}$, preço, endf), com

$F = \{numf\ num\_mat \rightarrow preço,\ numf \rightarrow endf\}$ não está na 2ª FN.

UNICAMP

Terceira Forma Normal



Dadas: $X \rightarrow Y$, $Y \not\rightarrow X$, e $Y \rightarrow Z$
dizemos que “Z depende transitivamente” de X.

As aspas são intencionais: esse tipo de dependência transitiva é indesejável, porém o requisito:

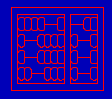
$Y \not\rightarrow X$ (X não é dependente funcional de Y)

é importante: se X e Y forem chaves de R e Z for um outro atributo qualquer de R, então

$X \rightarrow Y, Y \rightarrow X$ e $Y \rightarrow Z$

pois, afinal, Y também é chave de R

Terceira Forma Normal



Def: R está na 3ª FN se nenhum ANP de R "depende transitivamente" de qualquer chave de R.

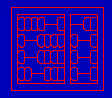
Exemplo:
Funcionários(numf, RG, nome, depto, nome_depto)

numf é chave primária, *RG* é chave e $depto \rightarrow nome\_depto$ é uma df válida,

R não está na 3ª FN, pois nome_depto é um ANP que "depende transitivamente" da chave numf.

UNICAMP

Forma Normal de Boyce Codd



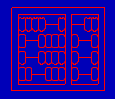
Def: uma relação R está na FNBC se para toda df não trivial, $X \rightarrow Y$, válida para R, então X é uma superchave R.

numa relação na FNBC, as únicas dependências funcionais não triviais são derivadas de chaves da relação.

Projeto *top down* de relações normalizadas segundo a FNBC: todas as dfs da relação devem ter uma chave do lado esquerdo!

UNICAMP

3ª FN ou FNBC?



Excepcionalmente não se pode ter todas as relações na FNBC. Exemplo clássico:

Endereços(Cidade, Rua, CEP),

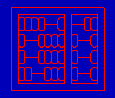
com dfs: $CidadeRua \rightarrow CEP$ e
$CEP \rightarrow Cidade$

É comum num sistema postal:

- o par *Cidade Rua* "determina" o *CEP*, mas nenhum deles determina o CEP,
- dado um CEP, a Cidade fica determinada, (mas CEP não determina Rua).

UNICAMP

Quarta forma normal



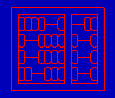
Origem: o mapeamento na mesma relação de dois ou mais atributos multivalorados independentes entre si gera redundâncias indesejáveis. Exemplo:

Livros(num_tombo, autores, assuntos),

onde um dado livro pode ter vários autores e tratar de vários assuntos.

Esta propriedade é refletida na chave primária da relação.

Quarta forma normal

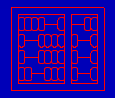


Por exemplo, poderíamos ter na tabela Livros seis linhas para o livro 31416:

num_tombo	autores	assuntos
	⋮	
31416	Aho	algoritmos
31416	Aho	complexidade
31416	Aho	busca
31416	Ullman	algoritmos
31416	Ullman	complexidade
31416	Ullman	busca

Livros está em FNBC, mas, existe grande redundância: um livro com k autores e m assuntos, poderia ter k*m linhas nessa relação e inconsistências podem ocorrer.

Quarta forma normal



Solução óbvia: decompor a tabela Livros em duas tabelas *Livros1* e *Livros2:*

Livros1(num_tombo, autores) e,

Livros2(num_tombo, assuntos).

Apenas cinco linhas vão aparecer nas duas tabelas para o livro 31416.

Caso geral de um livro com k autores e m assuntos:

ele apareceria k + m vezes nas duas tabelas em vez de k*m vezes

Eliminamos as redundâncias: um autor aparece uma só vez (para um dado livro), assim como um assunto.

Continência de formas normais

Relações na 4ª FN, também estão na FNBC; por sua vez, relações na FNBC estão em 3ª FN e estas, por sua vez, estão na 2ª FN; estas restrições sucessivas podem ser visualizadas assim: