

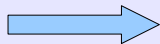
Dependências funcionais e normalização

Prof. Daniel S. Kaster

Departamento de Computação
Universidade Estadual de Londrina
dskaster@uel.br

Introdução

- Até então, foi uma modelagem conceitual utilizando o modelo Entidade-Relacionamento (Estendido)
 - Baseada em agrupamentos de atributos em relações através da intuição do projetista
 - Mapeado para o modelo relacional utilizando um algoritmo bem definido
- Entretanto, é necessário apresentar uma medida formal de qualidade de um projeto de banco de dados relacional
 - Dependências funcionais e normalização



O algoritmo de mapeamento EER->Relacional utiliza esses conceitos em sua concepção

Projeto de bancos de dados relacionais

- A teoria de dependências funcionais e de normalização é usada para projetar adequadamente um banco de dados relacional
 - Oferece mecanismos para analisar o projeto do banco de dados
 - Oferece métodos para corrigir problemas detectados
- No projeto de um banco de dados relacional, duas abordagens são possíveis
 - Bottom-up: partindo de atributos isolados e construindo gradativamente relações
 - Top-down: partindo de algumas relações e refinando-as

Dependências funcionais

- Uma Dependência Funcional (DF) é uma restrição entre dois conjuntos de atributos de um banco de dados
- Uma DF $X \rightarrow Y$, entre dois conjuntos de atributos X e Y , que são subconjuntos de um mesmo esquema relacional R especifica a seguinte restrição sobre uma relação $r(R)$
 - Se $t1[X]=t2[X]$, obrigatoriamente tem-se que $t1[Y]=t2[Y]$
 - Para quaisquer tuplas $t1, t2$ na relação r
- Diz-se que Y é funcionalmente dependente de X ou que X determina funcionalmente Y

Dependências funcionais dependem do problema modelado

- Uma dependência funcional é uma propriedade da semântica dos atributos no problema modelado
- Ex: quais dependências funcionais são válidas para o esquema a seguir?

PROF_DISC(codProfessor, nomeDisciplina, tituloLivroTexto)

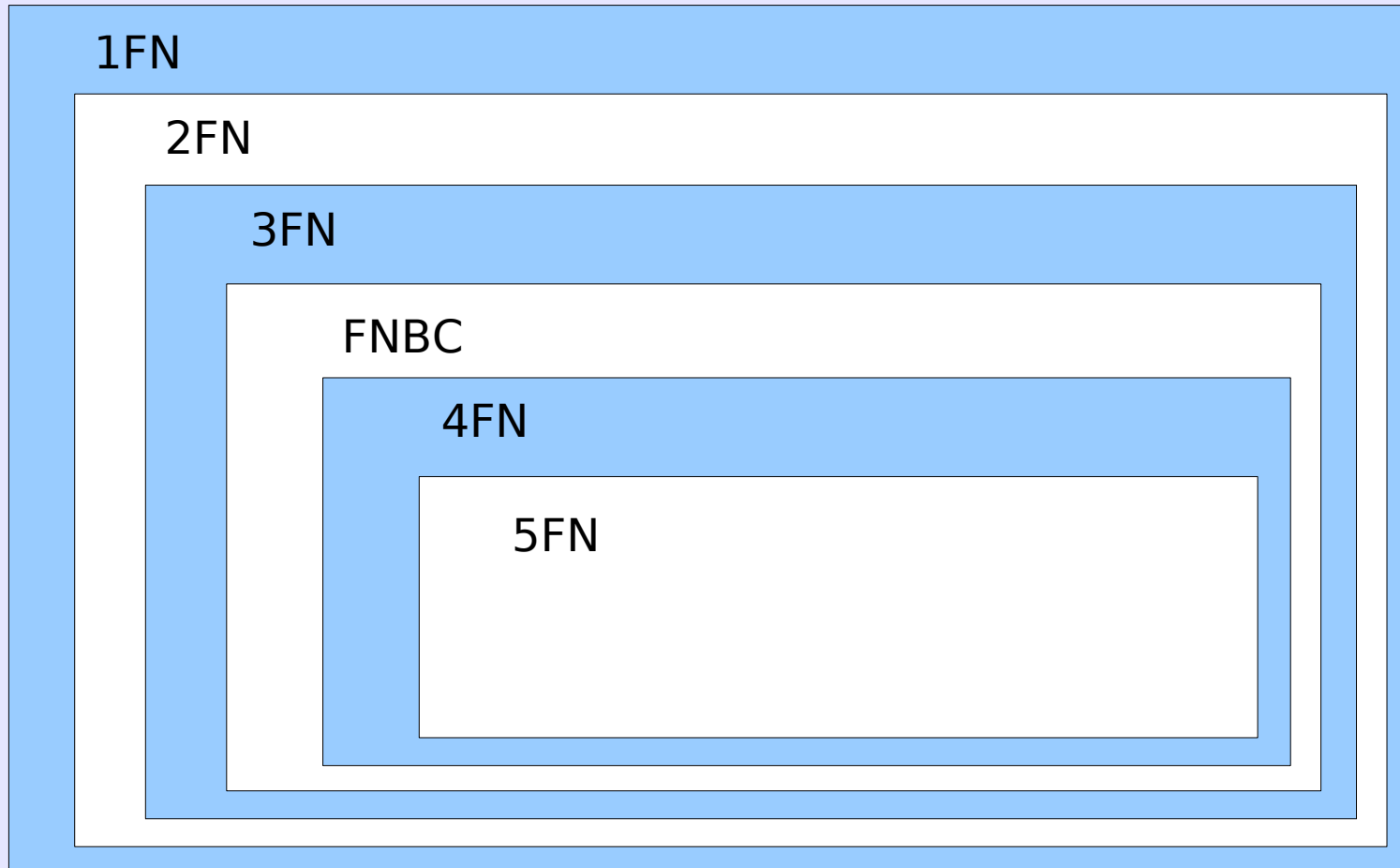
Identificação de dependências funcionais

- Identifica-se inicialmente as DFs triviais para o problema modelado
- Deriva-se as DFs utilizando regras de inferência
 - 1) Reflexiva: se $X \supseteq Y$, então $X \rightarrow Y$
 - 2) Aumentativa: $\{X \rightarrow Y\} \Rightarrow XZ \rightarrow YZ$
 - 3) Transitiva: $\{X \rightarrow Y, Y \rightarrow Z\} \Rightarrow X \rightarrow Z$
 - 4) Decomposição: $\{X \rightarrow YZ\} \Rightarrow X \rightarrow Z$
 - 5) União ou aditiva: $\{X \rightarrow Y, X \rightarrow Z\} \Rightarrow X \rightarrow YZ$
 - 6) Pseudotransitiva: $\{X \rightarrow Y, WY \rightarrow Z\} \Rightarrow WX \rightarrow Z$
- Regras 1 a 3 são legítimas e completas (regras de Armstrong)
 - Legítimas: qualquer DF derivada por essas regras satisfaz as outras DFs
 - Completas: essas regras permitem derivar qualquer DF possível

Normalização

- O processo de normalização proposto inicialmente por Codd, em 1972, realiza uma série de testes para certificar que um esquema de uma relação satisfaz uma determinada *forma normal*
 - A forma normal de um esquema relacional é a forma normal mais alta a que ele atende
- Consiste em analisar esquemas relacionais com base em suas chaves e dependências funcionais, de forma a
 - Minimizar redundâncias
 - Minimizar anomalias de atualização

Formas normais



Normalização por decomposição

- Propõe métodos para decompor esquemas que não satisfazem certas condições em esquemas relacionais menores que alcançam as propriedades desejadas (i.e. satisfazer à forma normal desejada)
 - Abordagem top-down
- Propriedades adicionais a serem satisfeitas no processo
 - Preservação de dependências
 - Garante que cada dependência funcional seja representada em alguma relação individual após a decomposição
 - Junção não aditiva (ou junção sem perdas)
 - Garante que o problema de geração de tuplas falsas não ocorra após a decomposição

Chave

- Um conjunto de atributos K é uma chave (candidata) de um esquema relacional R se
 - Todos os atributos de R são funcionalmente dependentes de K
- Os atributos que pertencem a alguma chave candidata de um esquema relacional R são chamados de **atributos principais** de R

Primeira forma normal (1FN)

- A 1FN exige que o domínio de um atributo inclua apenas valores atômicos (simples, indivisíveis) e que o valor de qualquer atributo em uma tupla tem que ser um valor simples do domínio deste atributo
- Ou seja, a 1FN não permite “relações dentro de relações” ou “relações como atributos de tuplas”
- Na verdade, a 1FN já faz parte da definição de uma relação, com atributos atômicos

Decomposições para a 1FN

- Atributos multivalorados
 - a) Separar em dois esquemas, relacionando-os via chave primária;
 - b) Fazer uma chave primária combinada – problema: redundância e anomalias;
 - c) Criar atributos para cada valor possível – problema: é preciso saber a quantidade máxima de atributos e podem haver muitos valores nulos.

<u>dnum</u>	dnome	dlocalização (multival.)
-------------	-------	-----------------------------

Decomposições para a 1FN (cont.)

- Atributos compostos

- a) Separar em outro esquema relacional a relação aninhada (atributo composto), e fazer a chave primária deste esquema a combinação da chave primária da relação inicial e a chave parcial da relação aninhada

<u>CPE</u>	enome	Projetos	
		<u>pnum</u>	horas

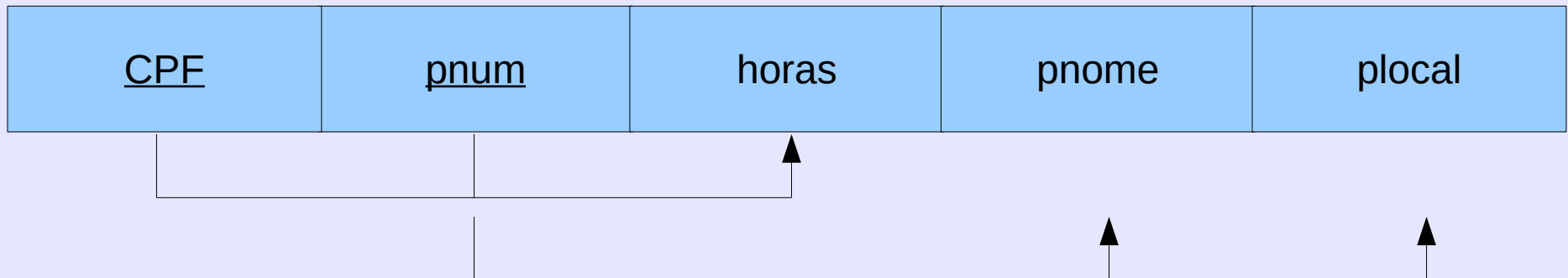
- Neste exemplo, o atributo composto projetos é multivalorado

Segunda forma normal (2FN)

- A 2FN é baseada no conceito de dependência funcional total na chave
 - Uma dependência funcional $X \rightarrow Y$ é total se, com a remoção de qualquer atributo de X esta dependência não seja mais válida, ou seja, X é o conjunto mínimo de atributos em que esta dependência funcional é válida
- Ex: quais das DF a seguir são totais?
 - $\{\text{CPF}, \text{pnum}\} \rightarrow \text{horas}$
 - $\{\text{CPF}, \text{pnum}\} \rightarrow \{\text{pnome}, \text{plocal}\}$

Decomposição para a 2FN

- Decompor a relação separando os atributos que têm dependência parcial da chave primária
- A chave primária dos novos esquemas é a parte da chave primária da relação inicial que define a dependência funcional parcial

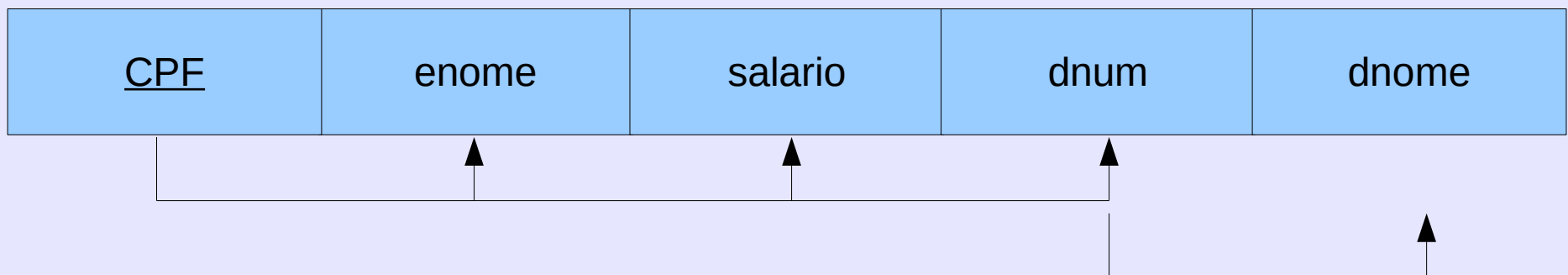


Terceira forma normal (3FN)

- A 3FN é baseada no conceito de dependência transitiva
 - Uma dependência funcional $X \rightarrow Y$ em um esquema relacional R é uma dependência transitiva se existe um conjunto de atributos Z que não são uma chave candidata de R e nem um subconjunto de uma chave candidata de R e valem as dependências $X \rightarrow Z$ e $Z \rightarrow Y$

Decomposição para a 3FN

- Decompor o esquema e criar um esquema relacional que inclua os atributos não-chave que determinam funcionalmente outros atributos não-chave



Definições gerais da 2FN e da 3FN

- Definição: Um esquema de relação R está na 2FN se cada atributo não principal A em R não for parcialmente dependente de qualquer chave (candidata) de R
- Definição: Um esquema de relação R está na 3FN se toda vez que uma dependência *não trivial* $X \rightarrow A$ se mantiver em R , então
 - a) X é superchave de R ; ou
 - b) A é um atributo principal de R

Forma normal Boyce-Codd (FNBC)

- A FNBC foi proposta inicialmente como uma forma mais simples da 3FN, mas é mais restritiva que a 3FN
- Um esquema relacional que está na FNBC está automaticamente na 3FN, mas um esquema na 3FN não necessariamente satisfaz a FNBC
- Definição: Um esquema de relação R está na FNBC se toda vez que uma dependência não trivial $X \rightarrow A$ se mantiver em R , então
 - X é superchave de R

Decomposição para a FNBC

id_propriedade	cidade	num_lote	area
----------------	--------	----------	------

- Ex:
 - Suponha um banco de dados de lotes, identificados unicamente considerando-se todas as propriedades armazenadas (id_propriedade)
 - Os lotes são numerados (num_lote) unicamente por cidade
 - Além disso, suponha que o banco de dados de lotes armazena dados apenas de duas cidades (Londrina e Tamarana) e que as áreas de lotes em Londrina têm entre 200 m² e 2500 m² e em Tamarana têm entre 3000 m² e 5000 m²
- Quais as dependências funcionais válidas nesse esquema relacional?
- Em que forma normal o esquema se encontra?