

Arquiteturas de Computadores

Professor Msc. Aparecido Vilela Junior
aparecido.vilela@unicesumar.edu.br

Introdução

- Um dos objetivos principais do gerenciamento de Bancos de Dados é manter a integridade dos dados nele armazenados, e para esse fim, algumas regras precisam ser consideradas.
- Algumas dessas regras são garantidas pelo próprio gerenciador
 - tais como unicidade da chave, ligação entre relações através da chave estrangeira, etc.
- Outras regras são definidas nos programas de aplicação que ficam responsáveis por mantê-las.

Introdução

- É necessário que as relações sejam bem fundamentadas, no sentido de evitar redundâncias que possam gerar, entre outros problemas, inconsistência de dados.
- Para procurar garantir esse aspecto, foi desenvolvida uma técnica chamada **Normalização**.

Introdução

- Redundância é a causa de vários problemas com esquemas relacionais:
 - armazenamento redundante, anomalias de inserção, de exclusão e de atualização.
- Restrições de integridade podem ser usadas para identificar esquemas com esses problemas e para sugerir refinamentos.

Introdução

- Principal técnica de refinamento: a decomposição de um esquema em sub-esquemas.
- A decomposição deve ser usada cuidadosamente:
 - Há motivos para se decompor uma relação?
 - A decomposição pode causar problemas?

Introdução

- Considere o esquema:

Pacientes(Id, Nome, Endereço, Telefone, Sexo,
Data_nascimento, Sigla_convênio,
Nome_convênio, Endereço_convênio,
Telefone_convênio)

- Esse é um exemplo de mau projeto!
 - Os dados de pacientes e os de convênios não deveriam estar na mesma tabela.

Introdução

Pacientes(Id, Nome, Endereço, Telefone, Sexo, Data_nascimento, Sigla_convênio, Nome_convênio, Endereço_convênio, Telefone_convênio)

- Por que?
 - Os dados de um convênio (nome, endereço e telefone do convênio) são repetidos para cada paciente associado a esse convênio.
 - Por exemplo, os dados do Convênio serão repetidos para cada um de seus associados.

Introdução

- **Anomalia de Inserção:**
 - Quando se inserir um paciente é preciso inserir também os dados do convênio, mesmo que já estejam cadastrados.
 - Não é possível inserir um convênio sem inserir também um paciente.
- **Anomalia de Exclusão:**
 - Ao se excluir um paciente, se este for o único associado de um convênio então os dados do convênio serão perdidos.
- **Anomalia de Modificação:**
 - Para se modificar os dados de um convênio, é preciso atualizar os mesmos dados em todas as tuplas de pacientes que estejam associados àquele convênio.

Dependência Funcional

- Dependências funcionais (DFs) são restrições de integridade mais gerais que as restrições de chave.
- Exemplo de dependência funcional:

$$\{\text{Sigla_convênio}\} \rightarrow \{\text{Nome_convênio}, \text{Endereço_convênio}, \text{Telefone_convênio}\}$$

- Leia-se: **Sigla_convênio** determina funcionalmente Nome_convênio, Endereço_convênio e Telefone_convênio.

Dependência Funcional

- Significado: “Se duas linhas da tabela Pacientes tiverem o mesmo valor de **Sigla_convênio**, então elas tem de ter o mesmo valor de **Nome_convênio**, de **Endereço_convênio** e de **Telefone_convênio**.”

Dependência Funcional

- Identifique as dependências funcionais:

Cod_Emp	Nome	Cargo	Salario
001	João da Silva	Analista Júnior	1.000,00
003	Maria João	Programador Pleno	1.200,00
006	Joice Casa	Analista Júnior	1.000,00
009	Caio Carvalho	Programador Pleno	1.200,00
002	Carlos Villa	Analista Sênior	2.200,00

Dependência Funcional

- Uma restrição de chave é um caso especial de DF: a chave determina funcionalmente todos os outros atributos da tabela.
- Como Id é chave da tabela Pacientes, temos que:

$\{Id\} \rightarrow \{Nome, Endere\c{c}o, Telefone, Sexo, Data_nascimento, Sigla_conv\^e{nio}, Nome_conv\^e{nio}, Endere\c{c}o_conv\^e{nio}, Telefone_conv\^e{nio}\}$

Dependência Funcional

- Certas DFs causam redundância!
- Por exemplo → Para cada associado de um convênio, os dados do convênio são repetidos na tabela Pacientes.
- A causa desse problema é a DF

$\{\text{Sigla_convênio}\} \rightarrow \{\text{Nome_convênio}, \text{Endereço_convênio}, \text{Telefone_convênio}\}$

Projeto de Banco de Dados

- O objetivo do projeto de um BD relacional
 - Gerar um conjunto de esquemas de relações que permitam armazenar informações sem redundância desnecessária
 - Recuperar informações facilmente

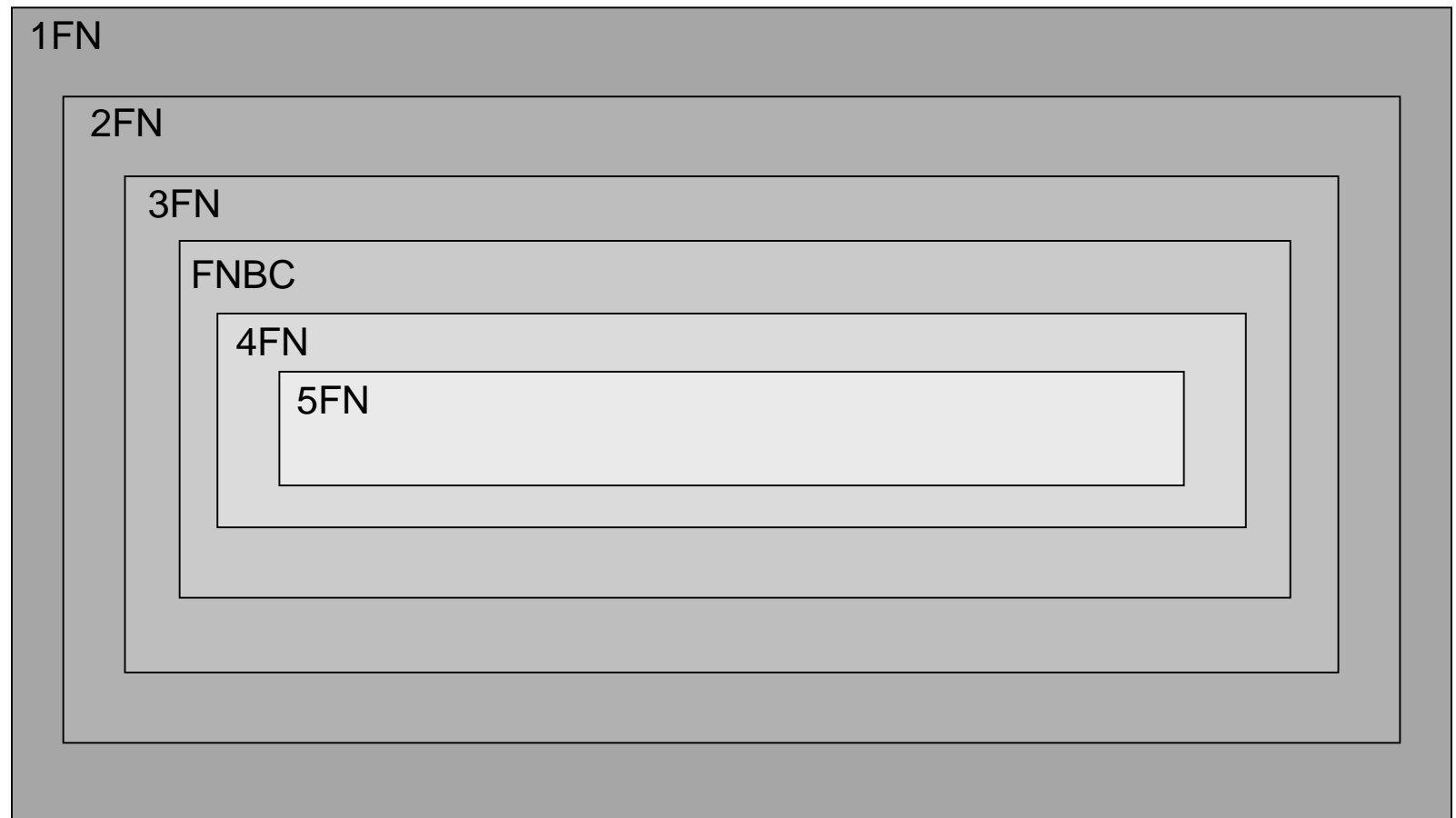
Normalização

- Conjunto de regras que ajudam na definição de bancos de dados que não contenham redundância desnecessária e que permitam o fácil acesso às informações
- Método permitindo identificar a existência de problemas (anomalias) no projeto de um BD relacional

Normalização

- Converte progressivamente uma tabela em tabelas de grau e cardinalidade menores até que pouca ou nenhuma redundância de dados exista
 - Há diferentes níveis de normalização, de acordo com as condições atendidas
 - A hierarquia entre as formas normais indica que uma tabela só pode estar numa forma mais avançada se, além de atender as condições necessárias, já estiver na forma normal imediatamente anterior

Níveis de Normalização



Normalização

- Se a normalização é bem sucedida:
 - O espaço de armazenamento dos dados diminui
 - A tabela pode ser atualizada com maior eficiência
 - A descrição do BD será imediata

Normalização

- A finalidade das regras de normalização é evitar anomalias de atualização no banco de dados
 - Anomalias de inserção
 - Evitar a repetição desnecessária de dados (redundância)
 - Anomalias de alteração
 - Evitar inconsistências e reduzir o esforço para a atualização dos dados
 - Anomalias de exclusão
 - Evitar a perda de informações associadas a um dado registro

Normalização

- Considere uma única tabela Vendas para representar as informações sobre os negócios de uma loja de CDs:

NOME_CLIENTE	COD_CD	MUSICA	CANTOR	PRECO	DATA_COMPRA
Alice Nóbrega	215621	Bem que se quis	Marisa Monte	R\$ 20,00	21/03/2003
...
Juliano Moreira	878650	Corcovado	Tom Jobim	R\$ 25,00	10/06/2003

Normalização

- Caso fosse preciso registrar a compra de 5 CDs iguais para um mesmo cliente, as seguintes anomalias seriam observadas:
 - Anomalia de inserção
 - Redundância em todas as colunas (5 linhas iguais na tabela)
 - Anomalia de alteração
 - A mudança no preço do CD deveria ser feita em todas as linhas correspondentes da tabela
 - Anomalia de exclusão
 - Só haveria registro dos CDs que fossem comprados; se a única venda de um CD fosse apagada, não haveria mais informações sobre aquele CD

Primeira Forma Normal (1FN)

- **Conceito:** Uma variável de relação (tabela) está em 1FN se, e somente se, em todo valor válido dessa variável de relação, cada tupla contém exatamente um valor para cada atributo
- Os atributos devem ser atômicos (indivisíveis)
- Atributos compostos ou multivalorados devem ser representados por novas linhas ou novas tabelas

Primeira Forma Normal (1FN)

- Exemplo: Tabela Controle de Faltas numa Escola
- A tabela abaixo não está na 1 FN

<u>COD_TURMA</u>	<u>ALUNO</u>	PROFESSOR	SALA	CAPACIDADE	QTE_FALTAS
BD1032	Alice Luna	Bruno Pereira	101	50	02
	Juliano Camargo				00
	Márcio Andrade				04
...

- Os atributos Aluno e Qte_Faltas não são atômicos (há mais de um valor para cada registro)

Primeira Forma Normal (1FN)

- Passos para obtenção da 1FN em uma tabela
 - Identificar a chave primária da tabela
 - Identificar os atributos compostos ou multivalorados
 - Incluir uma coluna/linha para cada atributo composto/multivalorado

Primeira Forma Normal (1FN)

- A tabela abaixo está na 1FN (atributos atômicos)

<u>COD_TURMA</u>	<u>ALUNO</u>	PROFESSOR	SALA	CAPACIDADE	QTE_FALTAS
BD1032	Alice Luna	Bruno Pereira	101	50	02
BD1032	Juliano Camargo	Bruno Pereira	101	50	00
BD1032	Márcio Andrade	Bruno Pereira	101	50	04
...

- O próximo passo é observar se ela está também na 2FN

Segunda Forma Normal (2FN)

- **Conceito 1:** uma variável de relação está em 2FN se, e somente se, ela está em 1FN e todo atributo não-chave é irreduzivelmente dependente da chave primária
- **Conceito 2:** uma variável de relação está em 2FN se, e somente se, ela está em 1FN e, para tabelas com chave primária composta, cada coluna não-chave depende de toda a chave, e não de apenas uma parte dela
- **Dica: tabelas em 1FN e com Chave Primária simples estão automaticamente em 2FN**

Segunda Forma Normal (2FN)

- A tabela abaixo está na 1FN mas não está na 2FN

<u>COD_TURMA</u>	<u>ALUNO</u>	PROFESSOR	SALA	CAPACIDADE	QTE_FALTAS
BD1032	Alice Luna	Bruno Pereira	101	50	02
BD1032	Juliano Camargo	Bruno Pereira	101	50	00
BD1032	Márcio Andrade	Bruno Pereira	101	50	04
...

- Os atributos Professor, Sala e Capacidade dependem apenas de Cod_Turma (repetição para todos os alunos da turma)

Segunda Forma Normal (2FN)

- Passos para obtenção da 2FN em uma tabela
 - Deixá-la em 1FN
 - Identificar os atributos que não fazem parte da chave primária da tabela
 - Para cada um desses atributos, analisar se seu valor é determinado por parte ou pela totalidade da chave
 - Criar novas tabelas para os atributos parcialmente dependentes, incluindo a parte da chave correspondente, e retirá-los da tabela original

Segunda Forma Normal (2FN)

- As tabelas abaixo estão em 2FN

<u>COD_TURMA</u>	<u>ALUNO</u>	QTE_FALTAS
BD1032	Alice Luna	02
BD1032	Juliano Camargo	00
BD1032	Márcio Andrade	04
...

<u>COD_TURMA</u>	PROFESSOR	SALA	CAPACIDADE
BD1032	Bruno Pereira	101	50
LG1512	Marina Lucena	101	50
JV8796	Ana Barbosa	101	50
...

Terceira Forma Normal (3FN)

- **Conceito 1:** uma variável de relação está em 3FN se, e somente se, ela está em 2FN e todo atributo não-chave é dependente de forma não transitiva da chave primária
- **Conceito 2:** uma variável de relação está em 3FN se, e somente se, ela está em 2FN e todo atributo não-chave depende apenas da chave, e não de outros atributos não-chave
- **Dica: tabelas em 2FN e com nenhum ou um atributo além da chave estão automaticamente em 3FN**

Terceira Forma Normal (3FN)

- A tabela abaixo está em 2FN, mas não está em 3FN

<u>COD_TURMA</u>	PROFESSOR	SALA	CAPACIDADE
BD1032	Bruno Pereira	101	50
LG1512	Marina Lucena	101	50
JV8796	Ana Barbosa	101	50
...

- O atributo Capacidade depende do atributo Sala, e não da chave Cod_Turma

Terceira Forma Normal (3FN)

- Passos para obtenção da 3FN em uma tabela
 - Deixá-la em 2FN
 - Identificar os atributos que não participam da chave primária da tabela
 - Para cada um desses atributos, analisar se seu valor é determinado por algum outro atributo não pertencente à chave primária
 - Criar novas tabelas para os atributos que não dependem exclusivamente da chave, incluindo o atributo determinante correspondente, e retirá-los da tabela original

Terceira Forma Normal (3FN)

- As tabelas abaixo estão em 3FN

<u>COD_TURMA</u>	<u>ALUNO</u>	QTE_FALTAS
BD1032	Alice Luna	02
BD1032	Juliano Camargo	00
BD1032	Márcio Andrade	04
...

<u>COD_TURMA</u>	PROFESSOR	<u>SALA</u>
BD1032	Bruno Pereira	101
LG1512	Marina Lucena	101
JV8796	Ana Barbosa	101
...

<u>SALA</u>	CAPACIDADE
101	50
201	40
301	50
...	...

- **1FN**: Eliminar atributos multivalorados ou compostos
- **2FN**: Eliminar atributos que dependem apenas de parte da chave primária composta
- **3FN**: Eliminar atributos que dependem de atributos não-chave

Normalização

- Aumentar o nível de normalização contribui para melhorar a qualidade do projeto do banco de dados
- Há ainda outras três formas normais (FNBC, 4FN e 5FN), cada uma com suas restrições
- Essas outras formas normais não serão vistas nesse curso

Exercícios...

- A tabela abaixo representa as vendas numa loja de CDs.
- Considerando as formas normais vistas (1FN, 2FN e 3FN), indicar quais são atendidas pelo projeto.
- Caso alguma delas não seja atendida, identifique o problema e proponha as mudanças necessárias.

Exercício...

TABELA VENDAS

Chave composta

<u>CLIENTE</u>	<u>COD_CD</u>	CANTOR	MUSICA	DURACAO	PRECO	DATA_COMPRA
Alice Nóbrega	215621	Marisa Monte	Beija Eu Chocolate	2:20 3:05	R\$ 20,00	21/03/2007
	878650	Tom Jobim	Corcovado Sabiá	2:50 2:10	R\$ 25,00	
...

TABELA PEDIDOS

[illegible]