

# Banco de Dados

Professor Msc. Aparecido Vilela Junior  
[aparecido.vilela@unicesumar.edu.br](mailto: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.

# Engenharia Reversa

## Entrada:

- ▢ qualquer conjunto de dados para os quais se disponha de uma descrição
  - ▢ documentos
  - ▢ arquivos manuais
  - ▢ arquivos convencionais em computador
  - ▢ bancos de dados gerenciados por SGBD não relacional

# Engenharia Reversa

Modelo ER  
(conceitual)

*Engenharia  
reversa de  
BD relacional*

Modelo relacional

Esquema de arquivo  
convencional ou  
documento

*Engenharia  
reversa de  
arquivos  
convencionais*



# 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:

$$\{\text{Sigla\_convênio}\} \rightarrow \{\text{Nome\_convênio}, \text{Endereço\_convênio}, \text{Telefone\_convênio}\}$$

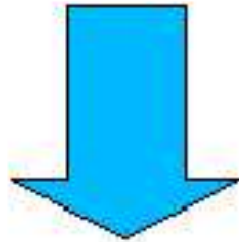
Leia-se: **Sigla\_convênio** determina funcionalmente Nome\_convênio, Endereço\_convênio e Telefone\_convênio.

# 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**.”

# Exemplo

Proj (CodProj, Tipo, Descr,  
(CodEmp, Nome, Cat, Sal, DataIni, TempAl))



ProjEmp (CodProj, Tipo, Descr, CodEmp, Nome, Cat,  
Sal, DataIni, TempAl)



Dados do projeto aparecem repetidos para cada empregado do projeto

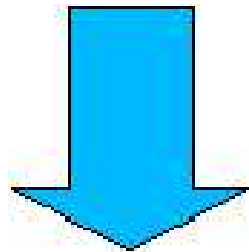
# Passagem à 1aFN

uma tabela para cada tabela aninhada

- ➡ Cria-se uma tabela referente a própria tabela que está sendo normalizada e uma tabela para cada tabela aninhada

# Exemplo

Proj (CodProj, Tipo, Descr,  
(CodEmp, Nome, Cat, Sal, DataIni, TempAl))



Proj (CodProj, Tipo, Descr)

ProjEmp (CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)

# Alternativas

- ➡ Primeira alternativa (tabela única) é mais correta
- ➡ Decompor uma tabela em várias tabelas (segunda alternativa)
  - ➡ podem ser perdidas relações entre informações



# Alternativas

➡ Para fins práticos

➡ preferimos a segunda alternativa  
*(decomposição de tabelas)*

➡ Quando houver diversas tabelas aninhadas, eventualmente com diversos níveis de aninhamento, fica difícil visualizar a tabela na 1FN na alternativa de tabela única

# Alternativas

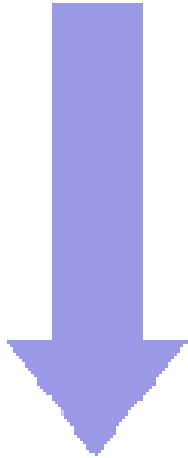
- ➡ Criar uma tabela na 1FN referente à tabela não normalizada
- ➡ A chave primária da tabela na 1FN é idêntica a chave da tabela ÑN

# Passagem à 1FN

**ÑN**

(CodProj, Tipo, Descr,

(CodEmp, Nome, Cat, Sal, DataIni, TempAl))



**1FN**

(CodProj, Tipo, Descr)

(CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)

# Passagem à 1FN

➡ Para cada tabela aninhada

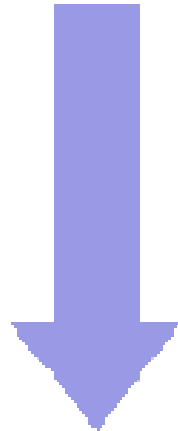
- ➡ criar uma tabela na 1FN composta pelas seguintes colunas:
  - ➡ a chave primária de cada uma das tabelas na qual a tabela em questão está aninhada
  - ➡ as colunas da própria tabela aninhada

# Passagem à 1FN

**ÑN**

(CodProj, Tipo, Descr,

(CodEmp, Nome, Cat, Sal, DataIni, TempAl))



**1FN**

(CodProj, Tipo, Descr)

(CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)

# Passagem à 1FN

**ÑN**

(CodProj, Tipo, Descr,

(CodEmp, Nome, Cat, Sal, DataIni, TempAl))



Tabela de nível mais externo:  
basta transcrever a chave

**1FN**

(CodProj, Tipo, Descr)

(CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)

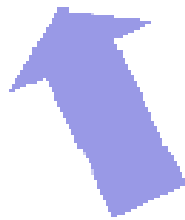
# Definição de chave primária

## 1FN

(CodProj, Tipo, Descr)

(CodProj, CodEmp, Nome, Cat,

Sal, DataIni, TempAl)



Qual é a chave primária desta tabela?

Pergunta a fazer:

"um valor de **CodEmp** (chave da tabela origem) aparece uma vez só no documento, ou várias?"

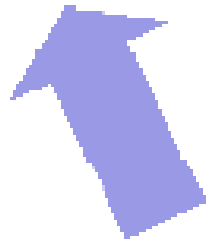
# Definição de chave primária

## 1FN

(CodProj, Tipo, Descr)

(CodProj, CodEmp, Nome, Cat,

Sal, DataIni, TempAI)



Como um valor de **CodEmp** aparece várias vezes, é necessário **CodProj** para distinguir as várias aparições



# Definição de chave primária

Proj:

CodProj	Tipo	Descr
LSC001	Novo Desenv.	Sistema de Estoque
PAG02	Manutenção	Sistema de RH

ProjEmp:

CodProj	CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	2146	João	A1	4	1/11/91	24
LSC001	3145	Sílvio	A2	4	2/10/91	24
LSC001	6126	José	B1	9	3/10/92	18
LSC001	1214	Carlos	A2	4	4/10/92	18
LSC001	8191	Mário	A1	4	1/11/92	12
PAG02	8191	Mário	A1	4	1/05/93	12
PAG02	4112	João	A2	4	4/01/91	24
PAG02	6126	José	B1	9	1/11/92	12

# Dependência Funcional

Identifique as dependências funcionais, crie as tabelas e inclua no Mysql:

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

Identifique as dependências funcionais, crie as tabelas e inclua no Mysql:

<u>Num_NF</u>	<u>Cod_Produto</u>	Descricao	Preco	Data	Quantidade
001	PC	Computador	1.500,00	15/04/1998	02
001	IMP	Impressora	500,00	15/04/1998	01
002	PC	Computador	1.500,00	16/04/1998	01
003	PC	Computador	1.500,00	14/04/1998	03

## Dependências funcionais

Num\_NF → Data

Cod\_Produto → Descricao,Preco

Num\_NF,Cod\_Produto → Quantidade

# 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



# Tabela Não Normalizada

- ➡ Possui uma ou mais **tabelas aninhadas**
- ➡ **Tabela Aninhada** (ou grupo repetido ou coluna multi-valorada ou coluna não atômica)
  - ➡ coluna que ao invés de conter valores atômicos, contém tabelas aninhadas
- ➡ Abreviatura: **ÑN**



# Tabela Aninhada - Exemplo

**Proj**

CódProj	Tipo	Descr	Emp					
			CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	Novo Desenv.	Sistema de Estoque	2146	João	A1	4	1/11/91	24
			3145	Sílvio	A2	4	2/10/91	24
			6126	José	B1	9	3/10/92	18
			1214	Carlos	A2	4	4/10/92	18
			8191	Mário	A1	4	1/11/92	12
PAG02	Manutenção	Sistema de RH	8191	Mário	A1	4	1/05/93	12
			4112	João	A2	4	4/01/91	24
			6126	José	B1	9	1/11/92	12

# Tabela Aninhada

**Proj**

CódProj	Tipo	Descr	Emp					
			CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	Novo Desenv.	Sistema de Estoque	2146	João	A1	4	1/11/91	24
			3145	Sílvio	A2	4	2/10/91	24
			6126	José	B1	9	3/10/92	18
			1214	Carlos	A2	4	4/10/92	18
			8191	Mário	A1	4	1/11/92	12
PAG02	Manutenção	Sistema de RH	8191	Mário	A1	4	1/05/93	12
			4112	João	A2	4	4/01/91	24
			6126	José	B1	9	1/11/92	12

**Proj** (CodProj, Tipo, Descr,

(CodEmp, Nome, Cat, Sal, DataIni, TempAl))

# 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

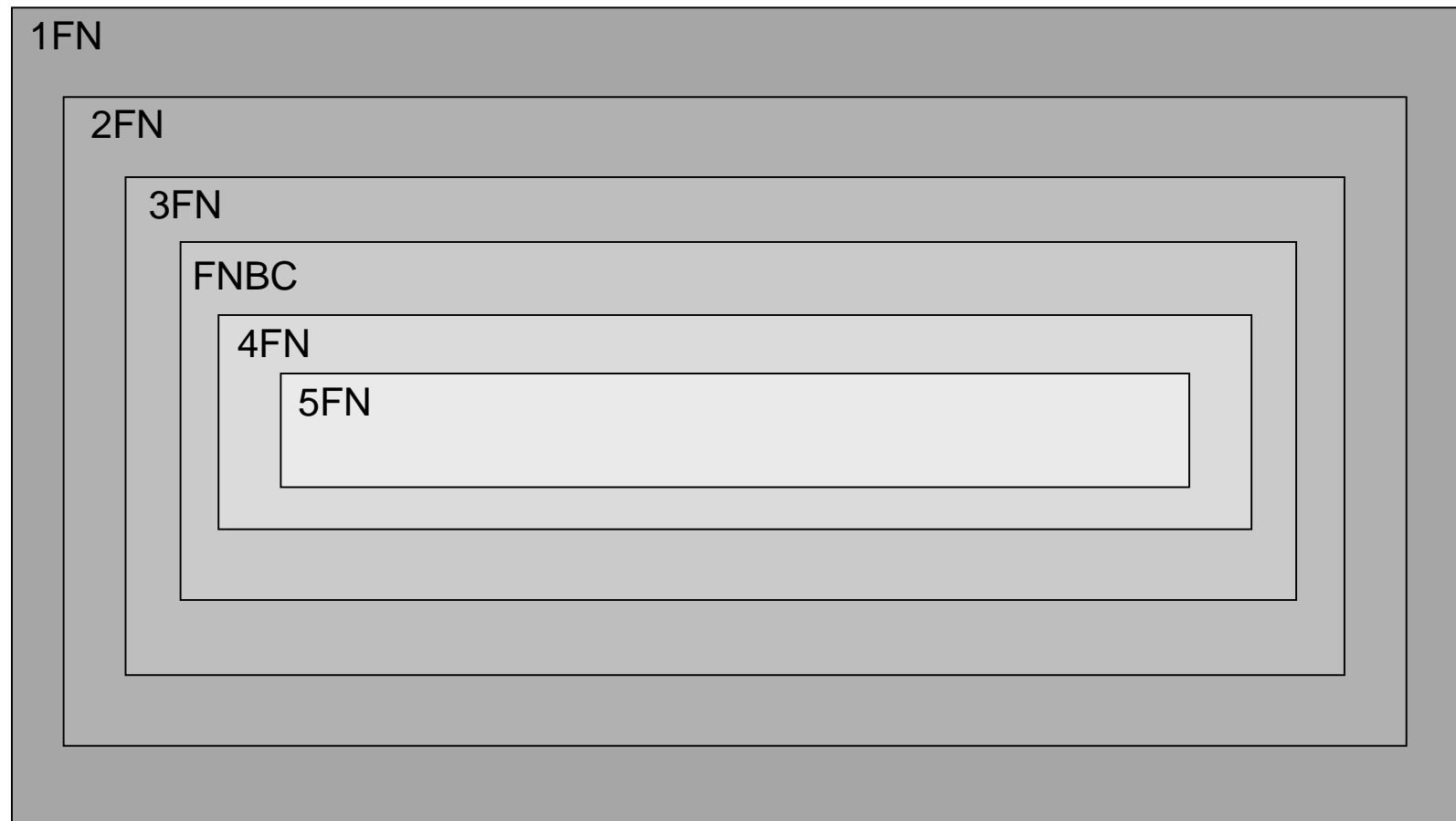
# Normalização

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 Juliano Camargo Márcio Andrade	Bruno Pereira	101	50	02 00 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)

<b><u>COD_TURMA</u></b>	<b><u>ALUNO</u></b>	<b>PROFESSOR</b>	<b>SALA</b>	<b>CAPACIDADE</b>	<b>QTE_FALTAS</b>
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 irredutivelmente 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

<b><u>COD_TURMA</u></b>	<b>PROFESSOR</b>	<b>SALA</b>	<b>CAPACIDADE</b>
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
...	...

# Regras Gerais – Normalização

- 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	2:20	R\$ 20,00	21/03/2007
			Chocolate	3:05		
	878650	Tom Jobim	Corcovado	2:50	R\$ 25,00	
			Sabiá	2:10		
...	...	...	...	...	...	...

# Solução

A tabela Vendas não está na 1FN, pois há vários atributos não atômicos

Para deixá-la em 1FN, é preciso dividir esses atributos em linhas

TABELA VENDAS

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

# Solução

A tabela Vendas não está na 2FN, pois há atributos que dependem apenas de parte da chave primária composta

Para deixá-la em 2FN, é preciso criar uma nova tabela

# Solução

TABELA VENDAS

<u>CLIENTE</u>	<u>COD_CD</u>	DATA_COMPRA
Alice Nóbrega	215621	21/03/2007
Alice Nóbrega	878650	21/03/2007
...	...	...

TABELA CDs

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

# Solução

A tabela CDs não está na 3FN, pois há atributos que dependem de atributos não-chave

Para deixá-la em 3FN, é preciso criar uma nova tabela

# Solução

TABELA CDs

<b>COD_CD</b>	<b>CANTOR</b>	<b><u>MUSICA</u></b>	<b>PRECO</b>
215621	Marisa Monte	Beija Eu	R\$ 20,00
215621	Marisa Monte	Chocolate	R\$ 20,00
878650	Tom Jobim	Corcovado	R\$ 25,00
878650	Tom Jobim	Sabiá	R\$ 25,00

TABELA MUSICAS

<b>MUSICA</b>	<b>DURACAO</b>
Beija Eu	2:20
Chocolate	3:05
Corcovado	2:50
Sabiá	2:10

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

TABELA VENDAS

CLIENTE	<u>COD_CD</u>	DATA_COMPRA
Alice Nóbrega	215621	21/03/2007
Alice Nóbrega	878650	21/03/2007
...	...	...

TABELA MUSICAS

MUSICA	DURACAO
Beija Eu	2:20
Chocolate	3:05
Corcovado	2:50
Sabiá	2:10

TABELA CDs

<u>COD_CD</u>	CANTOR	<u>MUSICA</u>	PRECO
215621	Marisa Monte	Beija Eu	R\$ 20,00
215621	Marisa Monte	Chocolate	R\$ 20,00
878650	Tom Jobim	Corcovado	R\$ 25,00
878650	Tom Jobim	Sabiá	R\$ 25,00

# Observação...

Veja que as tabelas Vendas, CDs e Musicas já estão em 3FN, mas ainda apresentam algumas redundâncias desnecessárias

    Cliente e Data\_Compra na tabela Vendas

    Cod\_CD, Cantor e Preco na tabela CDs

Para resolver esses problemas, seria necessário aplicar outras formas normais



# Exercício...

A tabela abaixo representa os pedidos de produtos de software para uma loja e não obedece nenhuma das formas normais vistas (1FN, 2FN e 3FN).

Indique os passos para deixá-la em cada uma dessas formas normais.

## TABELA PEDIDOS

[illegible]

# Solução...

Para deixar a tabela em 1FN, é preciso dividir os atributos não-atômicos em linhas

A chave da tabela é composta por Num\_Pedido e Cod\_Produto

## TABELA PEDIDOS

[illegible]

# Solução

Para deixar a tabela em 2FN, é preciso criar novas tabelas para os atributos que dependem apenas de parte da chave primária composta

Data, Fornecedor, CNPJ e Endereço dependem apenas de Num\_Pedido

Nome depende apenas de Cod\_Produto

Quantidade e Preço dependem da chave composta

# Solução

TABELA PEDIDOS

NUM_PEDIDO	COD_PRODUTO	QUANT	PRECO
003	033A	04	R\$ 130
003	002M	01	R\$ 499
003	145J	13	R\$ 256
004	002M	02	R\$ 450
004	083P	10	R\$ 85
004	145J	50	R\$ 110

TABELA PRODUTOS

COD_PRODUTO	NOME
033A	DOS
002M	Corel
145J	ABC
083P	ZAPT

TABELA DADOS\_PEDIDOS

NUM_PEDIDO	DATA	FORNECEDOR	CNPJ	ENDERECO
003	20/01/03	CasaSoftware	8888	R. Lapa, 77
004	27/01/03	BrasilSoftware	5555	Al. Itú, 49

# Solução

Para deixar as tabelas em 3FN, é preciso criar novas tabelas para os atributos dependentes de atributos não-chave

CNPJ e Endereco dependem de Fornecedor

# Solução

TABELA PEDIDOS

<u>NUM_PEDIDO</u>	<u>COD_PRODUTO</u>	QUANT	PRECO
003	033A	04	R\$ 130
003	002M	01	R\$ 499
003	145J	13	R\$ 256
004	002M	02	R\$ 450
004	083P	10	R\$ 85
004	145J	50	R\$ 110

TABELA PRODUTOS

<u>COD_PRODUTO</u>	NOME
033A	DOS
002M	Corel
145J	ABC
083P	ZAPT

TABELA DADOS\_PEDIDOS

<u>NUM_PEDIDO</u>	DATA	<u>CNPJ</u>
003	20/01/03	8888
004	27/01/03	5555

TABELA FORNECEDORES

<u>CNPJ</u>	FORNECEDOR	ENDERECO
8888	CasaSoftware	R. Lapa, 77
5555	BrasilSoftware	Al. Itú, 49



# Exercício 04

Considere a tabela abaixo, não necessariamente normalizada, referente à base de dados da questão anterior.

**Indicação (CPF, NomeCand, (NoCartao, OrdemPreferencia, NomeOrientador) )**

Esta tabela foi obtida a partir de uma página WEB que lista os candidatos a um programa de pós-graduação seguidos das suas indicações de orientadores preferenciais.

# Exercício 04

As dependências funcionais (podendo incluir dependências transitivas) que existem nesta tabela são as seguintes:

$(CPF) \rightarrow NomeCand$

$(NoCartao) \rightarrow NomeOrientador$

$(CPF, NoCartao) \rightarrow OrdemPreferencia$

1. Diga em que forma normal encontra-se a tabela.
2. Caso a tabela não se encontre na terceira forma normal, mostre a transformação da tabela para a terceira forma normal. Mostre cada forma normal intermediária, entre aquela em que a tabela se encontra e a terceira forma normal.

# Solução

1. A tabela se encontra na forma ÑN pois contém tabelas aninhadas.
2. 1FN: (eliminação de tabelas aninhadas)

Tabela1 (CPF, NomeCand,

Tabela2 (CPF, NoCartao, OrdemPreferencia, NomeOrientador) )

# Solução

2FN: (eliminação das dependências funcionais parciais)

Tabela1 (CPF, NomeCand,

Tabela2 (CPF, NoCartao, OrdemPreferencia, NomeOrientador)

Tabela3 (NoCartao, NomeOrientador)

3FN=2FN (não há dependências funcionais transitivas)

# Exercício 05

No contexto de um sistema de controle acadêmico, considera a tabela abaixo:

## **Matricula**

(CodAluno,CodTurma,CodDisciplina,NomeDisciplina,NomeAluno,  
CodLocalNascAluno,NomeLocalNascAluno)

As colunas possuem o seguinte significado:

**CodAluno** - código do aluno matriculado

**CodTurma** - código da turma na qual o aluno está matriculado (código é o identificador de turma)

**CodDisciplina** - código que identifica a disciplina da turma

**NomeDisciplina** - nome de uma disciplina da turma

**NomeAluno** - nome do aluno matriculado

**CodLocalNascAluno** - código da localidade em que nasceu o aluno

**NomeLocalNascAluno** - nome da localidade em que nasceu o aluno

Verifique se a tabela obedece a segunda e a terceira forma normais. Caso não obedeça, faça as transformações necessárias

# Exercício 05 - Solução

A tabela não se encontra na 2FN pois contém dependências funcionais parciais. A passagem a 2FN resulta nas seguintes tabelas:

(CodAluno, CodTurma)

(CodAluno, NomeAluno, CodLocalNascAluno, NomeLocalNascAluno)

(CodTurma, CodDisciplina, NomeDisciplina)

As duas últimas tabelas não estão na 3FN, já que contém dependências transitivas. Sua eliminação resulta no seguinte modelo relacional:

(CodAluno, CodTurma)

(CodAluno, NomeAluno, CodLocalNascAluno)

(CodLocalNascAluno, NomeLocalNascAluno)

(CodTurma, CodDisciplina)

(CodDisciplina, NomeDisciplina)

# Exercício 06

Situação de atendimento de OCs pendentes em dd/mm/aa

OC número: (---noOC---) Data da OC: (---data---)

Código do Fornecedor: (---codFornec---)

Nome do Fornecedor: (---NomeFornec)

Código Peça	Descrição Peça	Quantidade Pedida	Data Entrega	Quantidade Entregue
(---)	(---)	(---)	(---data---)	(---)
			(---data---)	(---)
		<b>Total entregue:</b>		(---)
(---)	(---)	(---)	(---data---)	(---)
		<b>Total entregue:</b>		(---)
(---)	(---)	(---)	(---data---)	(---)
			(---data---)	(---)
			(---data---)	(---)
		<b>Total entregue:</b>		(---)
(---)	(---)	(---)		
		<b>Total entregue:</b>		0

# Exercício 06

O exercício 06 apresenta a estrutura de um documento que lista a situação de atendimento de ordens de compra (OC). Uma OC pode estar parcialmente atendida, isto é, para cada OC podem ocorrer entregas parciais (apenas algumas peças, parte da quantidade).

O documento lista em seu cabeçalho o código da OC, sua data e seu fornecedor (código e nome). Para cada peça encomendada na OC, o documento lista o código, a descrição e a quantidade pedida. A seguir, para cada entrega é listada a data da entrega e a quantidade entregue. A soma da quantidade já entregue é listada a seguir. Observe que certas peças podem não ter tido entregas.