

Normalização

BCD29008 – Engenharia de Telecomunicações

Prof. Emerson Ribeiro de Mello

mello@ifsc.edu.br

Licenciamento



Slides licenciados sob [Creative Commons "Atribuição 4.0 Internacional"](https://creativecommons.org/licenses/by/4.0/)

Metodologia de projeto de banco de dados

O projeto de um banco de dados tem por objetivo representar os dados de forma a evitar **anomalias** e minimizar a **redundância de dados**

Uma relação com informações redundantes

Id	Aluno	Curso	Disciplina	Professor	Sala
123	Juca	Telecomunicações	Sinais	João	S01
456	Amélia	Elétrica	Sinais	João	S01
789	Breno	Computação	Programação I	Martin	S02
900	Jucé	Automação	Programação I	Martin	S02
334	Maira	Telecomunicações	Sistemas Distribuídos	Paulo	S03
453	Célio	Telecomunicações	Cálculo II	Luíza	S04
112	Cícero	Computação	Cálculo II	Luíza	S04
322	Marco	Automação	Cálculo II	Luíza	S04
567	Alonso	Computação	Sistemas Distribuídos	Paulo	S03
257	Luiz	Telecomunicações	Sinais	João	S01

Um banco de dados mal projetado gera anomalias

Aluno	Disciplina	Sala
João	bcd	LS01
Pedro	bcd	LS01
Ana	bcd	LS01

redundância da informação

Se toda disciplina acontece somente em uma única sala, então existe aqui uma **redundância da informação**

Um banco de dados mal projetado gera anomalias

Aluno	Disciplina	Sala
João	bcd	LS01
Pedro	bcd	LabApoio
Ana	bcd	LS01

anomalia de atualização

Se atualizarmos o número da sala somente em uma tupla, então teríamos aqui uma inconsistência dos dados

Um banco de dados mal projetado gera anomalias

Aluno

Disciplina

Sala

anomalia de exclusão

Se todos alunos desistem da disciplina, então perderíamos a informação entre disciplina e qual sala era ofertada

Um banco de dados mal projetado gera anomalias

Aluno	Disciplina	Sala
João	bcd	LS01
Pedro	bcd	LS01
Ana	bcd	LS01
	std	LS02

anomalia de inserção

Não é possível reservar uma sala para uma disciplina sem associar a um aluno

Um banco de dados mal projetado gera anomalias

Esse projeto é melhor que o anterior?

Aluno	Disciplina
João	bcd
João	std
Pedro	bcd
Pedro	csf

Disciplina	Sala
bcd	LS01
std	LS02
csf	LabApoio

- Seria possível reservar uma sala para uma disciplina, mesmo que essa ainda não tenha alunos matriculados?
- Seria possível alterar a sala de uma disciplina sem gerar inconsistência?

Cardinalidade no SQL

- Refere-se quantos valores únicos uma determinada coluna contém para um número específico de tuplas
- Quanto mais baixa a cardinalidade, maior o número de tuplas com o mesmo valor na coluna
- SGBD relacionais usam a cardinalidade para determinar o **plano de consulta** otimizado para uma determinada consulta

Informações redundantes

A cardinalidade pode ser usada para determinar se a tabela precisaria ser **normalizada**

Cardinalidade no SQL

<u>matricula</u>	idPessoa	nome	endereco	sexo	curso
1	234	João	Rua Abc	M	EngTelecom
2	456	José	Rua Xyz	M	EngTelecom
3	789	Maria	Rua Hee	F	EngTelecom
4	890	Ana	Rua Abc	F	EngTelecom
5	234	João	Rua Jeer	M	EngComputação

■ Cardinalidade alta

- Colunas cujos valores são únicos ou é bem incomum haver a repetição desse por diversas tuplas (Ex: **idPessoa**)

■ Cardinalidade normal

- Colunas cujos valores podem repetir, mas não é tão comum haver um grande número de repetições (Ex: coluna **endereco**)

■ Cardinalidade baixa

- Colunas cujos valores se repetem muito constantemente (Ex: **curso**)

Diretrizes para projetos de esquemas relacionais

■ Diretriz 1 – Clareza

- Modelar uma relação de forma que seja fácil explicar seu significado.
- Não combinar atributos de diferentes tipos de entidades e relacionamento

■ Diretriz 2 – Minimizar anomalias

- Modelar esquemas de relações de forma que não aconteça anomalias de inserção, exclusão ou atualização

■ Diretriz 3 – Evitar valores nulos

- Evite colocar atributos cujos valores frequentemente possam ser nulos.
- O valor nulo deveria ser considerado como exceção e não padrão para a maioria das tuplas

■ Diretriz 4 – Permitir junções

- Projete relações que possam ser unidas (*join*) com igualdade entre chaves primárias e chaves estrangeiras, evitando assim a perda de informações

Decomposição que gera perda de informação

Id	Aluno	Curso	Disciplina	Professor	Sala
123	Juca	Telecomunicações	Sinais	João	S01
789	Juca	Computação	Programação I	Martin	S02

decomposição

Id	Aluno
123	Juca
789	Juca

Aluno	Curso	Disciplina	Professor	Sala
Juca	Telecomunicações	Sinais	João	S01
Juca	Computação	Programação I	Martin	S02

JOIN

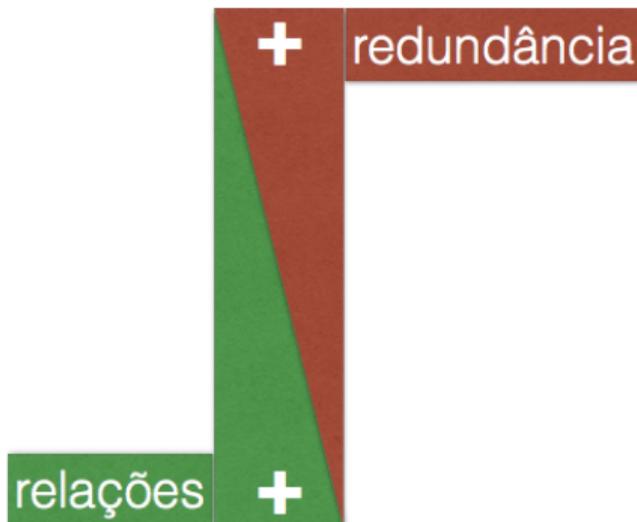
Id	Aluno	Curso	Disciplina	Professor	Sala
123	Juca	Telecomunicações	Sinais	João	S01
123	Juca	Computação	Programação I	Martin	S02
789	Juca	Telecomunicações	Sinais	João	S01
789	Juca	Computação	Programação I	Martin	S02

Normalização

Formas normais - (Normal Form)

Processo de normalização sujeita uma relação a uma série de testes para certificar-se de que essa satisfaça uma certa **forma normal**

- 1ª Forma Normal (1FN)
- 2ª Forma Normal (2FN)
- **3ª Forma Normal (3FN)**
- **Forma Normal de Boyce/Cood (BCNF)**
- 4ª Forma Normal (4FN)
- 5ª Forma Normal (5FN)



Normalização

Dependência funcional (DF)

Normalização de dados

Consiste na análise das tabelas com base em suas **dependências funcionais** e **chaves primárias**

Dependência funcional (DF)

Dependência funcional

Relação entre dois conjuntos de colunas de uma tabela

- Conjunto A **determina funcionalmente** o conjunto B se para quaisquer duas tuplas t_1 e t_2 :

$$t_1[A] = t_2[A], \text{ implicar em } t_1[B] = t_2[B]$$

- $(A \rightarrow B)$ é a notação usada para representar a dependência funcional entre os conjuntos A e B e pode ser interpretado como:
 - Os valores no atributo B de uma tupla dependem dos valores em A ; ou
 - Os valores em B de uma tupla são determinados por valores em A ; ou
 - B é funcionalmente dependente de A

Dependência funcional (DF)

Os valores em B de uma tupla são determinados por valores em A ?

iD	Aluno	Disciplina	Sala
123	João	bcd	LS01
456	Pedro	bcd	LS01
789	Ana	bcd	LS01
123	João	std	LS02

■ $Disciplina \rightarrow Sala$?

■ $Disciplina \rightarrow Aluno$?

Dependência funcional (DF)

Os valores em B de uma tupla são determinados por valores em A ?

	ID	Aluno	Disciplina	Sala
t_1	123	João	bcd	LS01
t_2	456	Pedro	bcd	LS01
t_3	789	Ana	bcd	LS01
t_4	123	João	std	LS02

■ $Disciplina \rightarrow Sala$?

■ No atual estado da relação, o valor da Disciplina determina, exclusivamente, a Sala

■ $Disciplina \rightarrow Aluno$?

Dependência funcional (DF)

Os valores em B de uma tupla são determinados por valores em A ?

iD	Aluno	Disciplina	Sala
123	João ←	bcd	LS01
456	Pedro ←	bcd	LS01
789	Ana ←	bcd	LS01
123	João	std	LS02

■ $Disciplina \rightarrow Sala$?

- No atual estado da relação, o valor da Disciplina determina, exclusivamente, a Sala

■ $Disciplina \rightarrow Aluno$?

- Não é uma dependência funcional

Dependência funcional (DF)

Os valores em B de uma tupla são determinados por valores em A ?

	ID	Aluno	Disciplina	Sala
t_1	123	João	bcd ←	LS01
t_2	456	Pedro	bcd ←	LS01
t_3	789	Ana	bcd ←	LS01
t_4	123	João	std	LS02

■ $Sala \rightarrow Disciplina$?

■ No atual estado da relação, a dependência funcional está assegurada

Dependência funcional (DF)

Os valores em B de uma tupla são determinados por valores em A ?

	ID	Aluno	Disciplina	Sala
t₁	123	João	bcd	← LS01
t₂	456	Pedro	bcd	← LS01
t₃	789	Ana	bcd	← LS01
t₄	123	João	std	LS02
t₅	123	João	poo	← LS01

■ $Sala \rightarrow Disciplina$?

■ Porém, não está assegurada para esse estado da relação

Dependência funcional (DF)

Os valores em B de uma tupla são determinados por valores em A ?

	iD	Aluno	Disciplina	Sala
t_1	123	João	bcd ←	LS01
t_2	456	Pedro	bcd ←	LS01
t_3	789	Ana	bcd ←	LS01
t_4	123	João	std	LS02
t_5	123	João	poo ←	LS01



Analisando um estado de uma relação é **possível verificar se uma DF é violada**, porém **não é possível provar que uma dada DF é propriedade do esquema** de uma relação

- Somente o conhecimento do domínio do problema permite isso

Dependência funcional (DF)

Os valores em B de uma tupla são determinados por valores em A ?

	iD	Aluno	Disciplina	Sala
t₁	123	João	bcd ←	LS01
t₂	456	Pedro	bcd ←	LS01
t₃	789	Ana	bcd ←	LS01
t₄	123	João	std	LS02
t₅	123	João	poo ←	LS01

- Exemplos de regras de negócio
 - Uma sala pode ser usada por mais de uma disciplina
 - Uma disciplina só pode usar uma única sala

Exercício

Encontre dependências funcionais

ID	Nome	Ramal	Cargo
123	João	101	Contador
456	Maria	201	Vendedor
789	Pedro	201	Vendedor
365	Cláudia	101	Gerente

Exercício

Encontre dependências funcionais

ID	Nome	Ramal	Cargo
123	João	101	Contador
456	Maria	201 ←	Vendedor
789	Pedro	201 ←	Vendedor
365	Cláudia	101	Gerente

- Cargo → Ramal – **dependência assegurada**

Exercício

Encontre dependências funcionais

ID	Nome	Ramal	Cargo
123	João	101 →	Contador
456	Maria	201	Vendedor
789	Pedro	201	Vendedor
365	Cláudia	101 →	Gerente

- Cargo → Ramal – **dependência assegurada**
- Ramal → Cargo – **dependência não assegurada**

Formas normais 1FN, 2FN e 3FN

1ª Forma normal (1FN)

Regra

Domínio de um campo só deve incluir valores atômicos, ou seja, não são permitidos campos multivalorados

Aluno	Disciplina
João	{bcd, std}
Pedro	{bcd, csf}
....	...

Viola a 1FN

Aluno	Disciplina
João	bcd
João	std
Pedro	bcd
Pedro	csf

Está na 1FN

1ª Forma normal (1FN)

Técnicas para adequação da tabela

- 1 Decomposição
- 2 Ampliação da chave
- 3 Uso de atributos atômicos

<u>Numero</u>	Curso	Vagas	Campus
123	Tele	32	{SJE, FLN, PHB}
456	Comp	36	{CCO}
789	Adm	50	{FLN}

1ª Forma normal (1FN)

Técnicas para adequação da tabela

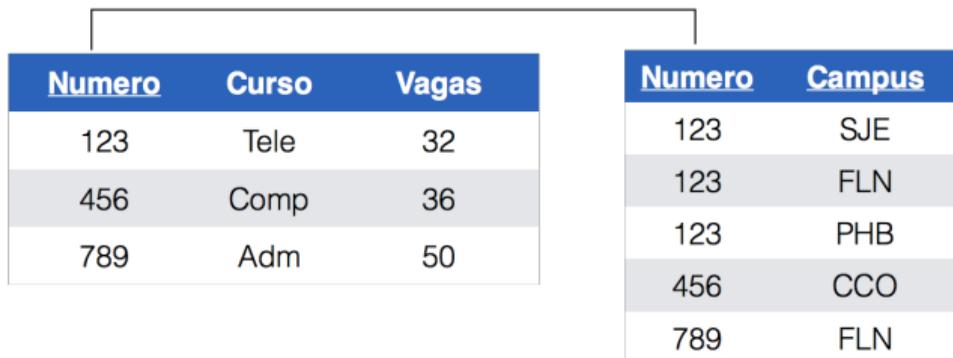
1 Decomposição

2 Ampliação da chave

3 Uso de atributos atômicos

Numero	Curso	Vagas	Campus
123	Tele	32	{SJE, FLN, PHB}
456	Comp	36	{CCO}
789	Adm	50	{FLN}

- Remover o atributo que viola a 1FN e colocá-lo em uma outra relação



1ª Forma normal (1FN)

Técnicas para adequação da tabela

1 Decomposição

2 Ampliação da chave

3 Uso de atributos atômicos

<u>Numero</u>	Curso	Vagas	Campus
123	Tele	32	{SJE, FLN, PHB}
456	Comp	36	{CCO}
789	Adm	50	{FLN}

<u>Numero</u>	Curso	Vagas	<u>Campus</u>
123	Tele	32	SJE
123	Tele	32	FLN
123	Tele	32	PHB
456	Comp	36	CCO
789	Adm	50	FLN

1ª Forma normal (1FN)

Técnicas para adequação da tabela

- 1 Decomposição
- 2 Ampliação da chave
- 3 **Uso de atributos atômicos**

Numero	Curso	Vagas	Campus
123	Tele	32	{SJE, FLN, PHB}
456	Comp	36	{CCO}
789	Adm	50	{FLN}

Numero	Curso	Vagas	Campus1	Campus2	Campus3
123	Tele	32	SJE	FLN	PHB
456	Comp	36	CCO		
789	Adm	50	FLN		

1ª Forma normal (1FN)

- A primeira solução (decomposição) seria a mais adequada, pois não causa redundância e é genérica
 - Não limita o número de campus onde o curso poderá ser ofertado
- A segunda solução seria decomposta nos próximos passos da normalização, o que levaria a primeira solução
- A terceira solução somente em casos muito específicos e sua vantagem é que não precisaria fazer junções (*JOIN*)

2ª Forma normal (2FN)

Baseada no conceito de dependência funcional total

Regra

- R está na 2FN se satisfizer a 1FN e todo atributo não primário x em R tem **dependência funcional total** de toda a chave primária de R
 - atributo não-chave não poderá ser dependente de apenas parte da chave
- $A \rightarrow B$ será uma **dependência funcional total** se a remoção de qualquer atributo x de A implicar que a dependência não é mais assegurada
 - $x \in A, (A - \{x\})$ não determina funcionalmente B
- $A \rightarrow B$ é uma **dependência parcial** se um atributo $x \in A$ puder ser removido de A e ainda assim a dependência continuar existindo
 - $x \in A, (A - \{x\}) \rightarrow B$

2ª Forma normal (2FN)

Todo atributo não primário \times tem dependência funcional total da chave primária

Funcionario_Projeto

fID	pID	Horas	fNome	pNome	pCampus
123	1	20	Juca	IoTW	SJE
456	2	6	Cláudia	CCwifi	SJE
789	1	4	Carlos	IoTW	SJE
123	4	10	Juca	aFean	PHB

■ Dependências funcionais

DF1: $\{fID, pID\} \rightarrow Horas$

DF2: $fID \rightarrow fNome$

DF3: $pID \rightarrow \{pNome, pCampus\}$

- $\{fID, pID\} \rightarrow Horas$ é uma **dependência funcional total**
 - $fID \rightarrow Horas$ e $pID \rightarrow Horas$ não são asseguradas
- $\{fID, pID\} \rightarrow fNome$ é uma **dependência funcional parcial**
 - $fID \rightarrow fNome$ é assegurada

2ª Forma normal (2FN)

Todo atributo não primário \times tem dependência funcional total da chave primária

Funcionario_Projeto

<u>fID</u>	<u>pID</u>	Horas	fNome	pNome	pCampus
123	1	20	Juca	IoTW	SJE
456	2	6	Cláudia	CCwifi	SJE
789	1	4	Carlos	IoTW	SJE
123	4	10	Juca	aFean	PHB

■ Dependências funcionais

DF1: $\{fID, pID\} \rightarrow Horas$

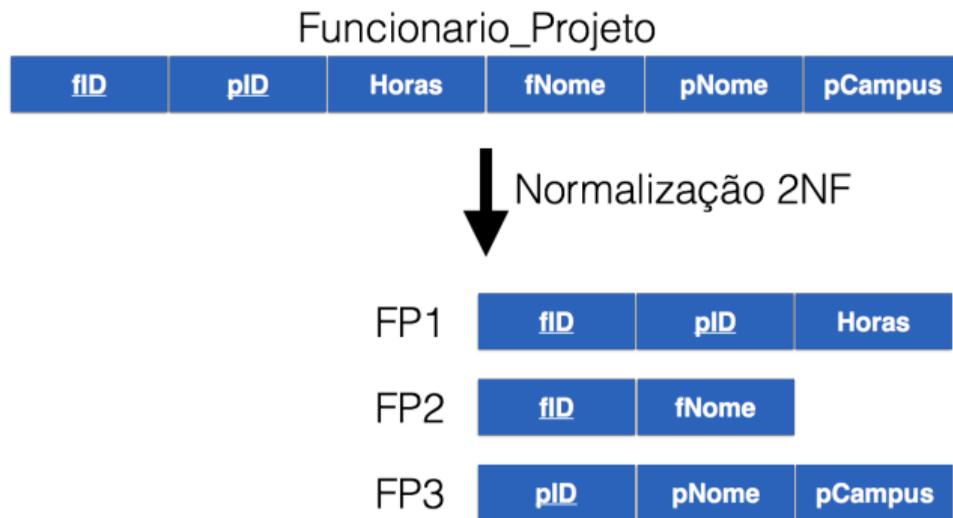
DF2: $fID \rightarrow fNome$

DF3: $pID \rightarrow \{pNome, pCampus\}$

- DF2 e DF3 fazem **fNome**, **pNome** e **pCampus** parcialmente dependente da chave primária $\{fID, pID\}$, violando assim a 2FN

2ª Forma normal (2FN)

Normalização por meio da decomposição



■ Dependências funcionais

DF1: $\{fID, pID\} \rightarrow Horas$

DF2: $fID \rightarrow fNome$

DF3: $pID \rightarrow \{pNome, pCampus\}$

Exercício

2ª Forma normal (2FN)

A relação abaixo está na 2FN?

Funcionario_Campus

<u>fCpf</u>	fNome	fDNasc	fEnd	cId	cNome	cDiretorCpf
123	Juca	1980-01-01	Rua X	1	SJE	555
456	Cláudia	1960-02-07	Rua Y	2	FLN	789
789	Carlos	1956-11-23	Rua Z	2	FLN	789
444	Ana	1990-10-10	Rua P	3	PHB	432

Exercício

2ª Forma normal (2FN)

A relação abaixo está na 2FN? **SIM!** chave só possui 1 atributo

Funcionario_Campus

<u>fCpf</u>	fNome	fDNasc	fEnd	cId	cNome	cDiretorCpf
123	Juca	1980-01-01	Rua X	1	SJE	555
456	Cláudia	1960-02-07	Rua Y	2	FLN	789
789	Carlos	1956-11-23	Rua Z	2	FLN	789
444	Ana	1990-10-10	Rua P	3	PHB	432

3ª Forma normal (3FN)

Regra

Satisfazer a 2FN e todo atributo que não faça parte da chave não poderá ser **transitivamente dependente** da chave primária

- Não possuir atributos que não pertençam a uma chave, funcionalmente determinados por outro conjunto de atributos que também não pertença a uma chave
- $A \rightarrow B$ será uma **dependência transitiva** se existir um conjunto de atributos C que não é uma chave candidata e nem um conjunto qualquer de chaves de R , e ambas $A \rightarrow C$ e $C \rightarrow B$ forem asseguradas

3ª Forma normal (3FN)

Se todo atributo não primário de R não for transitivamente dependente da chave primária.

Dependência transitiva: $A \rightarrow B$, $A \rightarrow C$ e $C \rightarrow B$

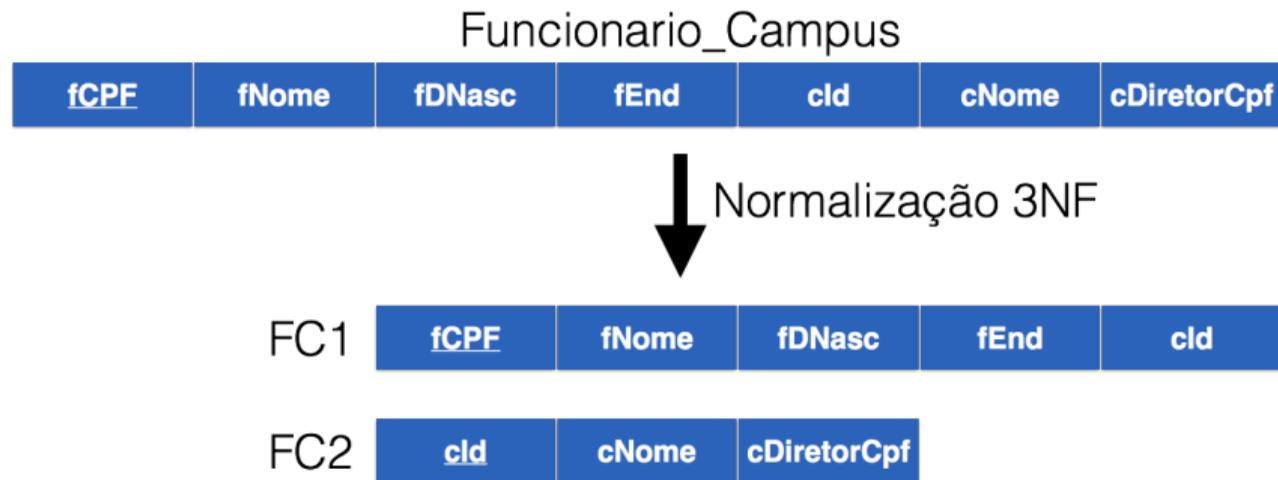
Funcionario_Campus

<u>fCpf</u>	fNome	fDNasc	fEnd	cld	cNome	cDiretorCpf
123	Juca	1980-01-01	Rua X	1	SJE	555
456	Cláudia	1960-02-07	Rua Y	2	FLN	789
789	Carlos	1956-11-23	Rua Z	2	FLN	789
444	Ana	1990-10-10	Rua P	3	PHB	432

- $fCpf \rightarrow cDiretorCpf$ é transitiva para cld
 - $fCpf \rightarrow cld$
 - $cld \rightarrow cDiretorCpf$
- cld não é chave primária e nem um subconjunto da chave
 - $cDiretorCpf \rightarrow cld$ é indesejável, uma vez que cld não é chave da relação

3ª Forma normal (3FN)

Normalizando por meio da decomposição



- $fCpf \rightarrow \{fNome, fDNasc, fEnd, cld\}$
- $cld \rightarrow \{cNome, cDiretorCpf\}$

Resumo das formas normais 1FN, 2FN e 3FN

- **1FN**

- **2FN**

- **3FN**

- Deve conter somente atributos atômicos

- Decompor e criar uma nova relação para cada atributo não atômico

Resumo das formas normais 1FN, 2FN e 3FN

- 1FN
 - 2FN
 - 3FN
- Relações que possuam chaves primárias com vários atributos, nenhum atributo externo à chave deve ser funcionalmente dependente de parte da chave primária
 - Decompor e montar uma nova relação para cada chave parcial com seus atributos dependentes

Resumo das formas normais 1FN, 2FN e 3FN

- 1FN
- 2FN
- 3FN
 - Não possuir atributos que não pertençam a uma chave, funcionalmente determinados por outro conjunto de atributos que também não pertença a uma chave
 - Decompor e montar uma relação que contenha os atributos não-chave que determinam funcionalmente outros atributos

Definições gerais para 2FN e 3FN

Definições gerais para 2FN e 3FN

Regra

Em definições mais gerais para 2FN e 3FN se **considera todas as chaves candidatas** da relação e **não somente a chave primária**

- Passos para normalização de relações para 3FN desaprovam as dependências parciais e transitivas na **chave primária**
- Nessa definição não são consideradas outras chaves candidatas, caso existam

Exemplo: Lotes à venda em várias cidades de um estado

- Cada lote tem uma área, uma taxa de imposto e um preço
- Cada lote tem um número único dentro da cidade, mas é possível que o mesmo número de lote apareça em diferentes cidades
- A taxa de imposto é fixa para cada cidade, ou seja, a taxa será a mesma para todos os lotes de uma cidade
- O preço de um lote é determinado pela sua área, independente da cidade a que ele pertença

<u>idPropriedade</u>	Cidade	NumeroLote	Area	Preco	Imposto
----------------------	--------	------------	------	-------	---------

Exemplo: Lotes à venda em várias cidades de um estado

<u>idPropriedade</u>	Cidade	NumeroLote	Area	Preco	Imposto
----------------------	--------	------------	------	-------	---------

■ Chaves candidatas

- **idPropriedade** – diferente em qualquer cidade
- **{Cidade, NumeroLote}** – número dos lotes são diferentes apenas dentro de cada cidade

■ Dependências funcionais

DF1: *idPropriedade* → {*Cidade, NumeroLote, Area, Preco, Imposto*}

DF2: {*Cidade, NumeroLote*} → {*idPropriedade, Area, Preco, Imposto*}

DF3: *Cidade* → *Imposto*

DF4: *Area* → *Preco*

Exemplo: Lotes à venda em várias cidades de um estado

Normalização para 2FN

Todo atributo não primário de R deve possuir **dependência funcional total** de cada chave de R

<u>idPropriedade</u>	Cidade	NumeroLote	Area	Preco	Imposto
----------------------	--------	------------	------	-------	---------

- **Imposto** é parcialmente dependente da chave candidata {*Cidade, NumeroLote*}

Exemplo: Lotes à venda em várias cidades de um estado

Normalização para 2FN

Todo atributo não primário de R deve possuir **dependência funcional total** de cada chave de R

<u>idPropriedade</u>	Cidade	NumeroLote	Area	Preco	Imposto
----------------------	--------	------------	------	-------	---------

- **Imposto** é parcialmente dependente da chave candidata $\{Cidade, NumeroLote\}$

L1

<u>idPropriedade</u>	Cidade	NumeroLote	Area	Preco
----------------------	--------	------------	------	-------

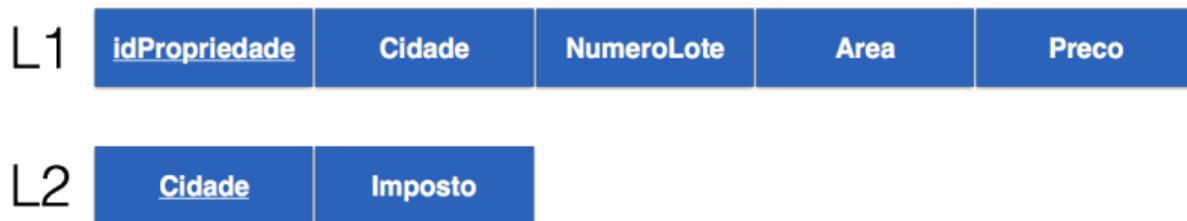
L2

<u>Cidade</u>	Imposto
---------------	---------

Exemplo: Lotes à venda em várias cidades de um estado

Normalização para 3FN

Todo atributo, que não é uma chave, não ser transitivamente dependente de nenhuma chave de R



- **Area** não é uma chave e **Preco** não é uma chave de L1

Exemplo: Lotes à venda em várias cidades de um estado

Normalização para 3FN

Todo atributo, que não é uma chave, não ser transitivamente dependente de nenhuma chave de R

L1	<u>idPropriedade</u>	Cidade	NumeroLote	Area	Preco
----	----------------------	--------	------------	------	-------

L2	<u>Cidade</u>	Imposto
----	---------------	---------

- **Area** não é uma chave e **Preco** não é uma chave de L1

L1a	<u>idPropriedade</u>	Cidade	NumeroLote	Area
-----	----------------------	--------	------------	------

L1b	<u>Area</u>	Preco
-----	-------------	-------

L2	<u>Cidade</u>	Imposto
----	---------------	---------

Forma normal de Boyce/Cood (BCNF)

Mais rígida que a 3FN

Regra

Todos os atributos são funcionalmente dependentes somente da chave e de nada mais

- $A \rightarrow B$ é uma **boa DF** se A for uma superchave, caso contrário é uma **DF ruim**
 - B é um conjunto com todos os demais atributos da relação
- Normalizar para BCNF consiste basicamente em remover as DF ruins
- Na prática a maioria dos esquemas de relação que está na 3FN também estará na BCNF
- Contudo, quando um esquema de relação assegurar $A \rightarrow B$ e A não for uma superchave e B for um atributo primário, então a relação estará na 3FN, mas não na BCNF

Forma normal de Boyce/Cood (BCNF)

Forma normal de Boyce/Cood (BCNF)

Exemplo

L1a

<u>idPropriedade</u>	Cidade	NumeroLote	Area
----------------------	--------	------------	------

- Suponha que existam milhares de lotes dentro de uma relação, porém esses são somente de duas cidades: São José e Florianópolis
- Também suponha que na cidade de São José as áreas dos lotes só podem ser: 0.5, 0.6 e 0.7 acres. Na cidade de Florianópolis as áreas dos lotes só podem ser: 1.1, 1.2 e 1.3
 - Cria-se então a **DF5**: *Area* → *Cidade*

Forma normal de Boyce/Cood (BCNF)

Exemplo

L1a

<u>idPropriedade</u>	Cidade	NumeroLote	Area
----------------------	--------	------------	------

- Suponha que existam milhares de lotes dentro de uma relação, porém esses são somente de duas cidades: São José e Florianópolis
- Também suponha que na cidade de São José as áreas dos lotes só podem ser: 0.5, 0.6 e 0.7 acres. Na cidade de Florianópolis as áreas dos lotes só podem ser: 1.1, 1.2 e 1.3
 - Cria-se então a **DF5**: *Area* → *Cidade*
 - L1a ainda estaria na 3FN, pois Cidade é um atributo primário (cidade,numeroLote)

Forma normal de Boyce/Cood (BCNF)

Exemplo

L1a

<u>idPropriedade</u>	Cidade	NumeroLote	Area
----------------------	--------	------------	------

- Suponha que existam milhares de lotes dentro de uma relação, porém esses são somente de duas cidades: São José e Florianópolis
- Também suponha que na cidade de São José as áreas dos lotes só podem ser: 0.5, 0.6 e 0.7 acres. Na cidade de Florianópolis as áreas dos lotes só podem ser: 1.1, 1.2 e 1.3
 - Cria-se então a **DF5**: *Area* → *Cidade*
 - L1a ainda estaria na 3FN, pois Cidade é um atributo primário (cidade,numeroLote)
- **DF5** viola a **BCNF**, pois o atributo Area não é uma superchave de L1a.
 - Isso gera redundâncias, podendo resultar em anomalias

Forma normal de Boyce/Cood (BCNF)

L1ax	<u>idPropriedade</u>	NumeroLote	Area
L1ay	<u>Area</u>	Cidade	

Exercícios

Exercício 1

- 1 Explique por que não está na 1FN
- 2 Ilustre os passos para transformar para a 3FN. É necessário identificar a chave primárias, chaves candidatas e chaves estrangeiras

<u>idFilial</u>	cidadeFilial	telefones
1	FLN	5300-0001, 5300-0002, 5300-0003
2	SJE	3381-2800, 3381-2880
3	JOI	5500-2222

Exercício 2

- 1 Explique por que não está na 2FN
- 2 Ilustre os passos para transformar para a 3FN. É necessário identificar a chave primárias, chaves candidatas e chaves estrangeiras

idFunc	idFilial	cidadeFilial	nome	cargo	cargaHorária
12345	1	FLN	Juca	Técnico Ref	20
12345	2	SJE	Juca	Técnico Ref	20
82992	2	SJE	Pedro	Analista TI	30
82992	1	FLN	Pedro	Analista TI	10

Exercício 3

- 1 Explique por que não está na 3FN
- 2 Ilustre os passos para transformar para a 3FN. É necessário identificar a chave primárias, chaves candidatas e chaves estrangeiras

idFunc	nome	cargo	salário	idFilial	cidadeFilial	telefone
12345	Juca	Técnico Ref	5000	1	FLN	5300-0001
76889	Ana	Gerente	4800	1	FLN	5300-0001
82992	Pedro	Analista TI	4500	2	SJE	3381-2800
12022	João	Engenheiro	6000	3	JOI	5500-2222
91011	Silvia	Analista TI	4600	2	SJE	3381-2800

Exercício 4

Deixe a tabela abaixo na 3FN.

<u>estudante</u>	<u>professor</u>	<u>Disciplina</u>	<u>fase</u>	<u>Sala</u>	<u>Nota</u>	<u>Livro</u>	<u>profEmail</u>
Clerina	Rosa	CAL	4f	Sala 16	6	Cálculo integral	rosa@ifsc
João	Carlos	BCD	1f	LabSW II	7	Sistemas BD	c@ifsc
João	Juca	POO	2f	LabProg	8	Como programar OO	ju@ifsc
Ana	Carlos	BCD	1f	LabSW II	6	Sistemas BD	c@ifsc
Jucelina	Rosa	CAL	4f	Sala 16	8	Cálculo integral	rosa@ifsc
Sílvia	Juca	STD	5f	LabSW I	4	SD: teoria	ju@ifsc
Célia	Rosso	CAL	4f	Sala 15	3	Cálculo integral	rosso@ifsc

Exercício 5

Crie um conjunto de relações normalizadas das quais seja possível extrair a seguinte nota fiscal

Código do Hóspede:		12345		Data:		15/10/2018
Nome do hóspede:		Juca Trip		Data check-in		13/10/2018
Endereço do hóspede:		Rua ABC, 123		Data check-out		15/10/2018
Cidade, estado e CEP:		São José, SC, 88103-310				
Centro de custo	Nome do centro	Data	Código do item	Descrição	Valor	Balanco
123	Quarto	13/10/2018	1000	Quarto luxo	230,00	
		13/10/2018	1010	Água	3,00	
		14/10/2018	1000	Quarto luxo	230,00	
				Subtotal		463,00
245	Restaurante	13/10/2018	2000	Almoço	30,00	
		14/10/2018	2100	Jantar	50,00	
				Subtotal		80,00
367	Lavanderia	14/10/2018	3200	Passar camisa	20,00	
				Subtotal		20,00
				Total		563,00

Aulas baseadas em

-  Henry F.; Sudarshan Silberschatz, Abraham; Korth.
Sistemas de banco de dados.
6a. Edição - Editora Campus, 2012
-  Navathe, S.
Sistemas de banco de dados
4a. Edição - Editora Person Addison, 2005
-  Ré, C.
CS145 Introduction to databases – Stanford University