

BANCO DE DADOS

Normalização – Parte 2

FORMAS NORMAIS

- Segunda Forma Normal (2NF): é baseada no conceito de atributo primo e dependência funcional total.

FORMAS NORMAIS

- Segunda Forma Normal (2NF): é baseada no conceito de **atributo primo** e **dependência funcional total**.

Um atributo de um esquema de tabela T é chamado ATRIBUTO PRIMO de T se ele é membro de alguma chave candidata de T. Caso contrário ele é chamado ATRIBUTO NÃO PRIMO.

FORMAS NORMAIS

- Segunda Forma Normal (2NF): é baseada no conceito de **atributo primo** e **dependência funcional total**.

Um atributo de um esquema de tabela T é chamado ATRIBUTO PRIMO de T se ele é membro de alguma chave candidata de T. Caso contrário ele é chamado ATRIBUTO NÃO PRIMO.



FORMAS NORMAIS

- Segunda Forma Normal (2NF): é baseada no conceito de **atributo primo** e **dependência funcional total**.
- A dependência funcional $X \longrightarrow Y$ é uma **dependência funcional total** se ao remover qualquer atributo A de X a dependência funcional deixa de valer.
- Ou seja, para qualquer $A \in X$, $(X - A)$ não determina funcionalmente Y.
- Uma dependência funcional $X \longrightarrow Y$ é uma **dependência funcional parcial** se para algum atributo A $\subset X$, $(X - A) \longrightarrow Y$.



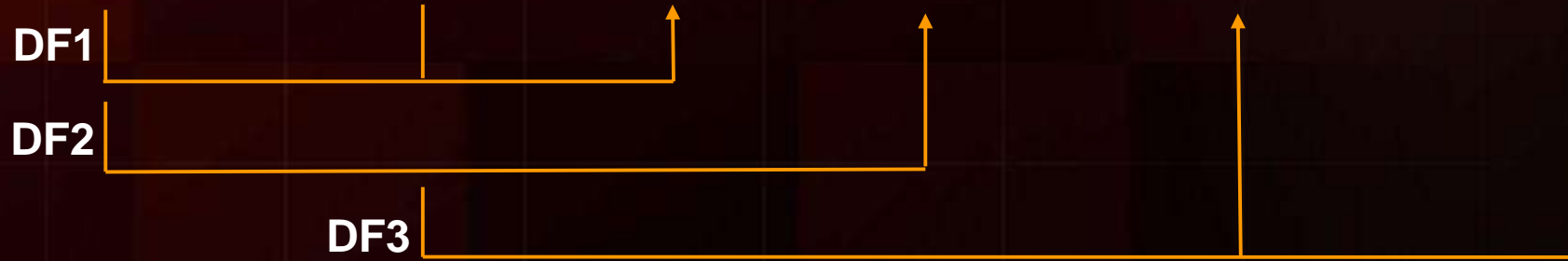
FUNC_PROJ



{func_ident, proj_numero} → horas, func_nome, proj_nome, proj_localizacao

FUNC_PROJ

<u>func_ident</u>	<u>proj_numero</u>	horas	func_nome	proj_nome	proj_localizacao
-------------------	--------------------	-------	-----------	-----------	------------------



$\{func_ident, proj_numero\} \rightarrow horas, func_nome, proj_nome, proj_localizacao$

$\{func_ident, proj_numero\} \rightarrow horas$ é dependência funcional total
 $func_ident \rightarrow horas$ e $proj_numero \rightarrow horas$ não valem no esquema

$\{func_ident, proj_numero\} \rightarrow func_nome$ é dependência funcional parcial
 $func_ident \rightarrow func_nome$ vale no esquema

FORMAS NORMAIS

- Segunda Forma Normal (2NF)

Um esquema da tabela T está na 2NF se todo atributo não primo A em T tem dependência funcional total da chave primária (ou de qualquer chave) de T .

ou, escrevendo de uma outra forma...

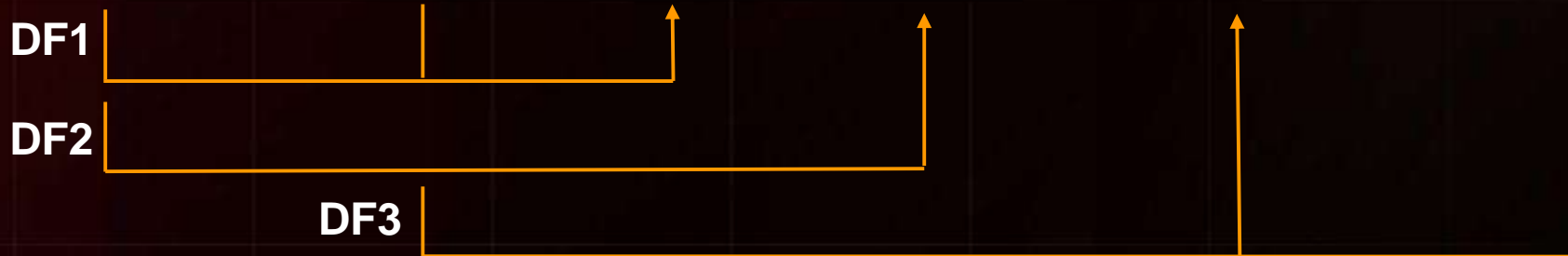
Um esquema da tabela T está na 2NF se todo atributo não primo A em T não depender parcialmente de qualquer chave de T .

FORMAS NORMAIS

- Segunda Forma Normal (2NF)

FUNC_PROJ

<u>func_ident</u>	<u>proj_numero</u>	horas	func_nome	proj_nome	proj_localizacao
-------------------	--------------------	-------	-----------	-----------	------------------

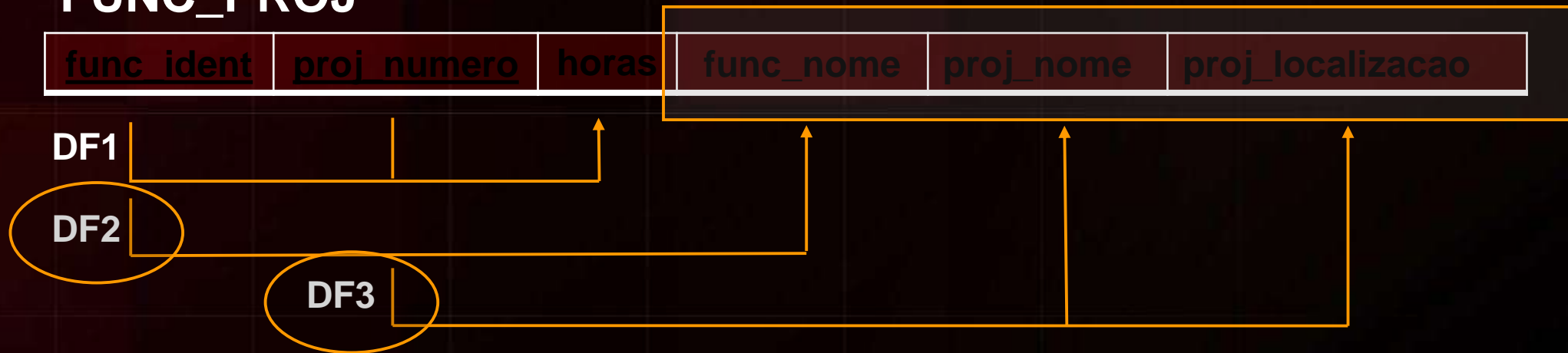


FUNC_PROJ está na 1NF mas não está na 2NF !!!!

FORMAS NORMAIS

- Segunda Forma Normal (2NF)

FUNC_PROJ



FUNC_PROJ está na 1NF mas não está na 2NF !!!!

FORMAS NORMAIS

- Decompondo o esquema em outros menores, seguindo as dependências funcionais: normalizando o esquema.

FUNC_PROJ

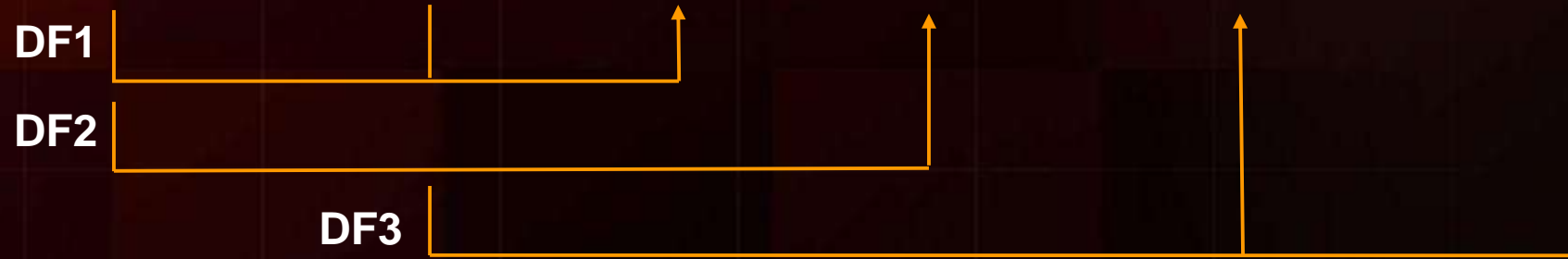
<u>func_ident</u>	<u>proj_numero</u>	horas	func_nome	proj_nome	proj_localizacao
-------------------	--------------------	-------	-----------	-----------	------------------



TRÊS NOVOS ESQUEMAS DA TABELA

FUNC_PROJ

<u>func_ident</u>	<u>proj_numero</u>	horas	func_nome	proj_nome	proj_localizacao
-------------------	--------------------	-------	-----------	-----------	------------------



FP1

<u>func_ident</u>	<u>proj_numero</u>	horas
-------------------	--------------------	-------



FP2

<u>func_ident</u>	func_nome
-------------------	-----------



FP3

<u>proj_numero</u>	proj_nome	proj_localizacao
--------------------	-----------	------------------



FORMAS NORMAIS

- Terceira Forma Normal (3NF): é baseada no conceito de **dependência transitiva**.

Uma dependência funcional $X \longrightarrow Y$ em um esquema da tabela T é uma **dependência transitiva** se:

- existe um conjunto de atributos Z em T,
- Z não é chave candidata e nem um subconjunto de qualquer chave de T, e
- tanto $X \longrightarrow Z$ quanto $Z \longrightarrow Y$ valem no esquema.



FUNC_DEPTO



$\text{func_ident} \rightarrow \text{depto_g_ident}$ é uma **dependência transitiva** por **depto_numero**.

func_ident \rightarrow **depto_numero** e **depto_numero** \rightarrow **depto_g_ident** valem no esquema

TERCEIRA FORMA NORMAL (3NF)

Um esquema de tabela T está na 3NF se ele está em 2NF e nenhum atributo não primo de T é transitivamente dependente da chave primária (ou de qualquer chave) de T.

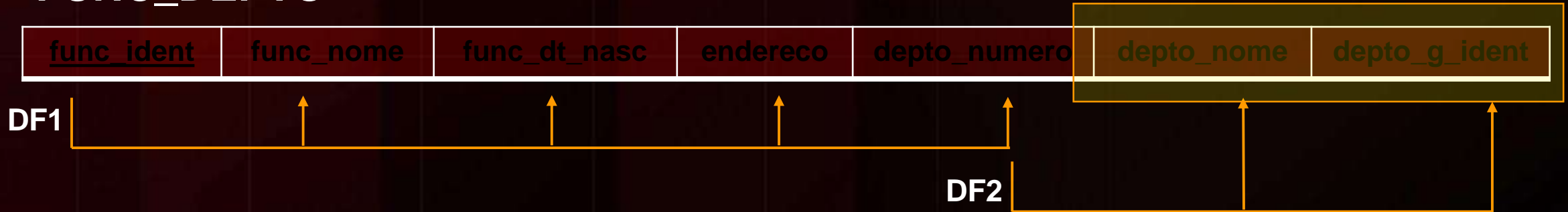
ou, escrevendo de uma outra forma ...

Um esquema de tabela T está na 3NF se todo atributo não primo em T atende ambas as condições abaixo:

- a) ele tem dependência funcional total de toda chave de T;
- b) ele é não transitivamente dependente de toda chave de T.

TERCEIRA FORMA NORMAL (3NF)

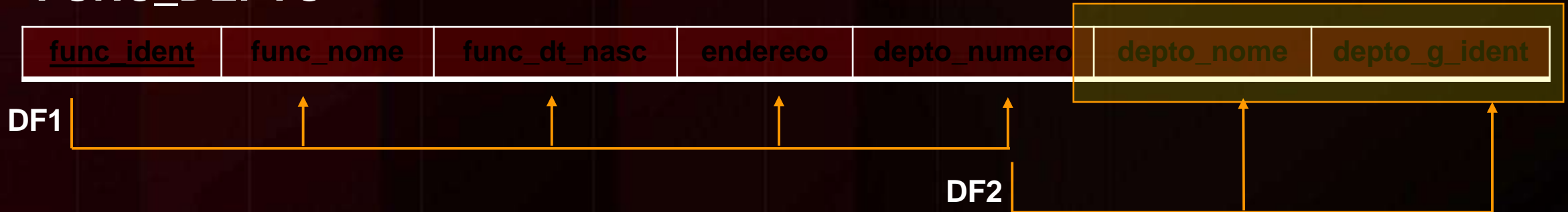
FUNC_DEPTO



FUNC_DEPTO está na 1NF e na 2NF, mas não está na 3NF !!!!

TERCEIRA FORMA NORMAL (3NF)

FUNC_DEPTO



- Decompondo o esquema em outros menores, seguindo as dependências funcionais: normalizando o esquema.

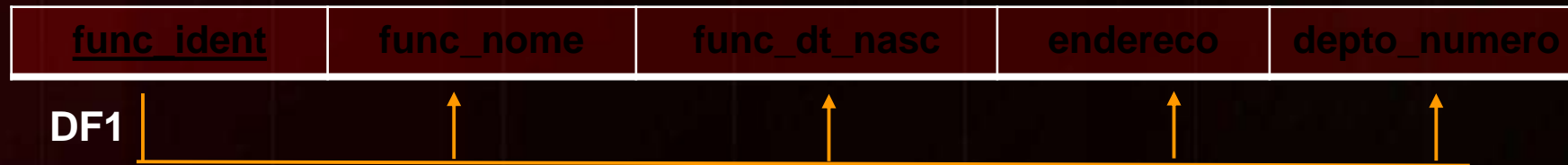


DOIS NOVOS ESQUEMAS DA TABELA

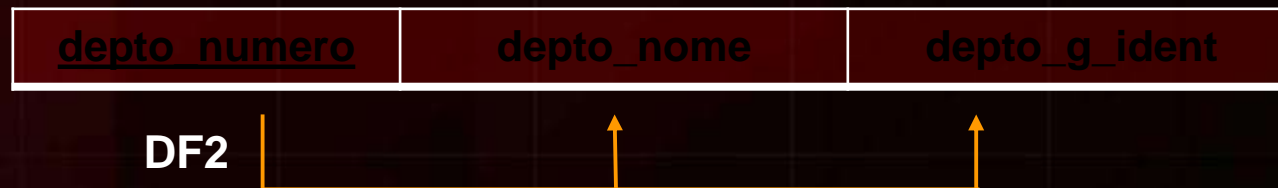
FUNC_DEPTO



FD1



FD2



BOYCE-CODD NORMAL FORM (BCNF)

- Um esquema da tabela T está na BCNF se sempre que uma dependência funcional não trivial $X \longrightarrow A$ vale em T , X é uma superchave de T .

BOYCE-CODD NORMAL FORM (BCNF)

FP1

<u>func_ident</u>	<u>proj_numero</u>	horas
-------------------	--------------------	-------



FP2

<u>func_ident</u>	<u>func_nome</u>
-------------------	------------------



FP3

<u>proj_numero</u>	proj_nome	proj_localizacao
--------------------	-----------	------------------

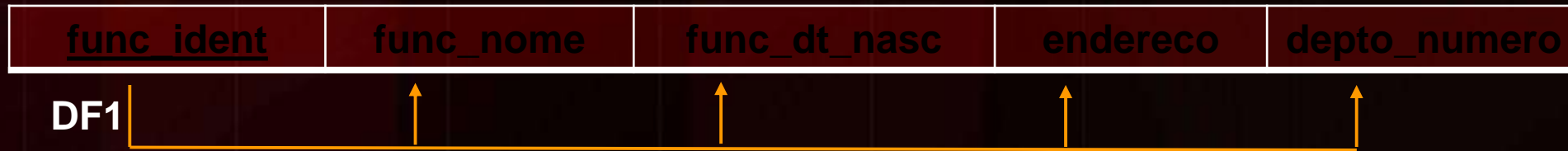


Exemplo 1

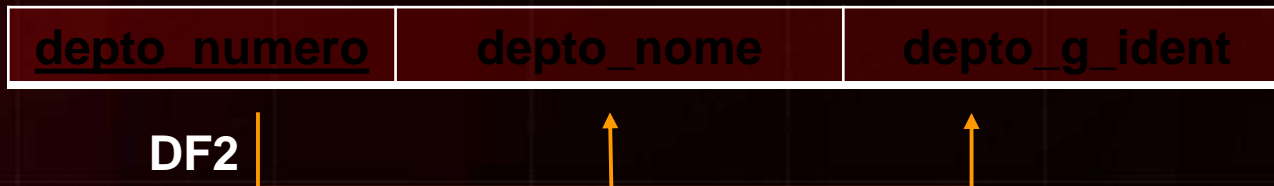
As tabelas estão na BCNF.

BOYCE-CODD NORMAL FORM (BCNF)

FD1



FD2



Exemplo 2

As tabelas estão na BCNF.

BOYCE-CODD NORMAL FORM (BCNF)

LOTES1A



????

Exemplo 3

LOTES1A



Compreendendo o esquema:

- pela DF1 sabemos que id_propriedade é uma chave candidata (ela foi a escolhida para ser chave primária)
- pela DF2 sabemos que {estado, id_lote} é uma chave candidata
- pela DF5 sabemos que todos os terrenos de mesma área estão localizados no mesmo estado

LOTES1A



Compreendendo o esquema:

- pela DF1 sabemos que id_propriedade é uma chave candidata (ela foi a escolhida para ser chave primária)
- pela DF2 sabemos que {estado, id_lote} é uma chave candidata
- pela DF5 sabemos que todos os terrenos de mesma área estão localizados no mesmo estado

<u>id_propriedade</u>	estado	id_lote	area
111	São Paulo	4	100
222	São Paulo	5	100
333	São Paulo	4	50
444	Rio de Janeiro	5	200

BOYCE-CODD NORMAL FORM (BCNF)

LOTES1A



A tabela **não está na BCNF** pois o lado esquerdo da dependência funcional não trivial **area** → **estado** não é uma superchave!!!

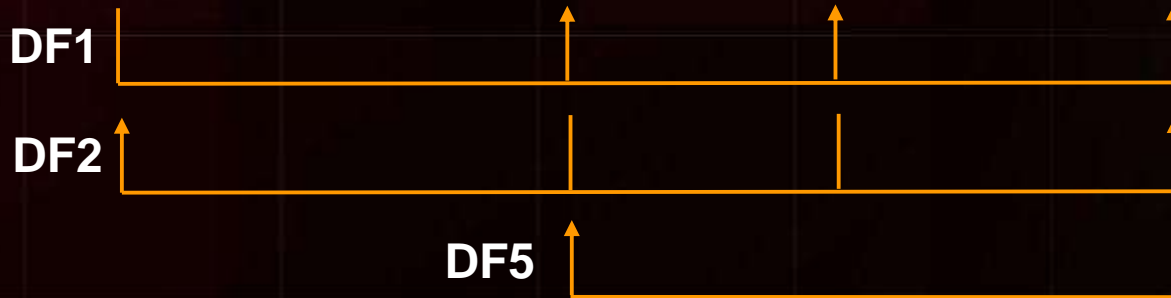
Observe que o esquema está na 3NF!!!

BOYCE-CODD NORMAL FORM (BCNF)

- Decompondo o esquema em esquemas menores, seguindo as dependências funcionais: normalizando o esquema.

LOTES1A

<u>id_propriedade</u>	estado	id_lote	area
-----------------------	--------	---------	------



DOIS NOVOS ESQUEMAS DA TABELA

LOTES1A

<u>id_propriedade</u>	estado	id_lote	area
-----------------------	--------	---------	------



LOTES1AX

<u>id_propriedade</u>	id_lote	area
-----------------------	---------	------



LOTES1AY

<u>area</u>	estado
-------------	--------



Agora as tabelas estão na BCNF, porém a DF2 foi perdida.

Formas normais

- Propriedades de decomposição relacional.
- É preciso assegurar que cada atributo de T apareça em pelo menos uma tabela dentro da decomposição.
- É uma meta ter cada uma das tabelas da composição em BCNF ou 3NF.
- É desejável preservar as dependências funcionais na decomposição.
- É preciso assegurar que a decomposição não causará perdas na junção.

Estes slides estão baseados na bibliografia:

- Elmasri, Ramez; Navathe, Shamkant B. Sistemas de Banco de Dados. Pearson, 7a edição, 1128p., 2018.

BANCO DE DADOS

Normalização – Parte 2