



UNIPAC - CENTRO UNIVERSITÁRIO PRESIDENTE ANTÔNIO CARLOS
CAMPUS BARBACENA

Bacharelado em Ciência da Computação



Banco de Dados

Material de Apoio

Parte VI – Normalização de Banco de Dados

Prof. José Osvano da Silva, PMP, PSM I
joseosvano@unipac.br

1º sem / 2022

Sumário

- Introdução
- Diretrizes para Esquemas de Relação
- Dependência Funcional
- Formas Normais
- Exemplos

Introdução

- **Objetivo**
- Definir o conceito de dependência funcional, que é a ferramenta básica para analisar esquemas de banco de dados relacionais;
- Descrever o processo de normalização para obter bons projetos, baseados em esquemas de relação que reduzem a redundância de dados e as chances dos dados se tornarem inconsistentes.

Diretrizes para esquemas de relação

- Quatro *diretrizes informais* podem ser usadas como *medidas para determinar a qualidade* de projeto do esquema da relação:
- **Garantir que a semântica dos atributos seja clara no esquema**
- **Reduzir a informação redundante nas tuplas**
- **Reduzir os valores NULL nas tuplas**
- **Reprovar a possibilidade de gerar tuplas falsas**

Dependência Funcional

- **Definição:** restrição entre dois conjuntos de atributos.

Definição. Uma **dependência funcional**, indicada por $X \rightarrow Y$, entre dois conjuntos de atributos X e Y , que são subconjuntos de R , *especifica* uma restrição sobre possíveis tuplas que podem formar um estado da relação r de R . A restrição é que, para quaisquer duas tuplas t_1 e t_2 em r que tenham $t_1[X] = t_2[X]$, elas também devem ter de ter $t_1[Y] = t_2[Y]$.

- Isso significa que os valores do componente Y de uma tupla em r dependem dos (ou são determinados pelos) valores do componente X .
- Assim, a funcionalidade X determina Y em um esquema de relação R se, e somente se, sempre que duas tuplas de $r(R)$ combinarem sobre seu valor X , elas devem necessariamente combinar sobre seu valor Y .

Dependência Funcional

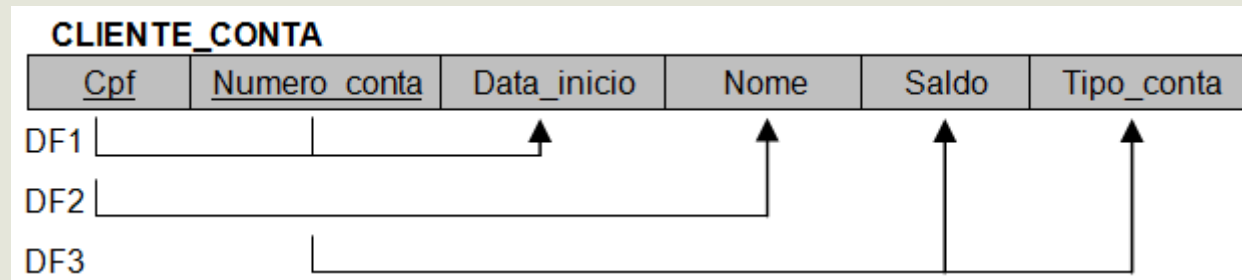


Figura 1 – Um esquema de relação sofrendo anomalias de atualização.

- Pela semântica dos atributos, sabemos que as seguintes DFs devem ser mantidas:
 - a. $Cpf \rightarrow Nome$
 - b. $Numero_conta \rightarrow \{Saldo, Tipo_conta\}$
 - c. $\{Cpf, Numero_conta\} \rightarrow Data_inicio$

Dependência Funcional

BANCO_AGENCIA		
Nome_banco	Numero_agencia	Endereco
Banco do Brasil	4336	Av. Weimar G. Torres, 2965
Banco do Brasil	3153	Av. Marcelino Pires, 1960
Itaú	464	Av. Marcelino Pires, 2830
Bradesco	3676	Rua Joaquim Teixeira Alves, 1750

Figura 2 – Um estado de relação de BANCO_AGENCIA com uma possível dependência funcional $\text{Endereco} \rightarrow \text{Numero_agencia}$. Porém, $\text{Nome_banco} \rightarrow \text{Numero_agencia}$ está excluída.

A	B	C	D
a1	b1	c1	d1
a1	b2	c2	d2
a2	b2	c2	d3
a3	b3	c4	d3

Figura 3 – Uma relação R com dependências funcionais: $B \rightarrow C$; $C \rightarrow B$; $\{A, B\} \rightarrow C$; $\{A, B\} \rightarrow D$; e $\{C, D\} \rightarrow B$. No entanto, as seguintes *não* se mantêm: $A \rightarrow B$; $B \rightarrow A$; $D \rightarrow C$.

Formas Normais

Definição. A forma normal de uma relação refere-se à condição de forma normal mais alta a que ela atende e, portanto, indica o grau ao qual ela foi normalizada.

Definição. Desnormalização é o processo de armazenar a junção de relações na forma normal mais alta como uma relação da base, que está em uma forma normal mais baixa.

Formas Normais

Definição. Uma **superchave** de um esquema de relação $R = \{A_1, A_2, \dots, A_n\}$ é um conjunto de atributos $S \subseteq R$ com a propriedade de que duas tuplas t_1 e t_2 , em qualquer estado de relação válido r de R , não terão $t_1[S] = t_2[S]$. Uma **chave** Ch é uma superchave com a propriedade adicional de que a remoção de qualquer atributo de Ch fará que Ch não seja mais uma superchave.

CLIENTE

<u>Cpf</u>	Nome	Sexo	Endereco
------------	------	------	----------

- Na Figura, $\{Cpf\}$ é uma chave para Nome, $\{Cpf, Nome, Sexo\}$ e atributos que inclua Cpf são todos

HISTORICO

Cpf_cliente	Num_conta	Data_inicio
444.555.666-77	98723-4	12-08-1979
444.555.666-77	78963-2	04-03-1980
111.222.333-44	23584-7	17-12-1997
666.777.888-99	23584-7	17-12-1997
222.333.444-55	62548-6	15-01-1994
999.666.111-88	13879-3	03-09-2013
555.444.777-33	86340-2	29-11-2010
333.888.666-22	35480-9	12-04-1985

Formas Normais

- **Primeira Forma Normal (1FN):** o domínio de um atributo deve incluir apenas valores **atômicos** (ou **indivisíveis**)
- Suponha que cada cliente possa ter *certo número de* telefones. Como podemos ver, CLIENTE não está em 1FN porque Telefone não é um atributo atômico, conforme ilustrado pela primeira tupla na Figura (b).

(b) **CLIENTE**

<u>Cpf</u>	Nome	Sexo	Endereço	Telefone
444.555.666-77	João B Silva	M	Rua Arapongas, 1234	(67)3421-1122, (67)3910-3344, (67)9999-5566
999.666.111-88	Robson Soares	M	Rua dos Ingleses, 3245	(67)3427-2255
111.222.333-44	Jennifer B Souza	F	Rua Cuiabá, 1050	(67)3422-7788

Formas Normais

- **Primeira Forma Normal (1FN)**

- Existem três técnicas principais para conseguir a primeira forma normal para tal relação:

1. Remover o atributo Telefone que viola a 1FN e colocá-lo em uma relação separada TELEFONE_CLIENTE, junto com a chave primária Cpf de CLIENTE.

A chave primária dessa relação é a combinação {Cpf, Telefone}, como mostra a Figura.

TELEFONE_CLIENTE	
<u>Cpf</u>	<u>Telefone</u>
444.555.666-77	(67)3421-1122
444.555.666-77	(67)3910-3344
444.555.666-77	(67)9999-5566
999.666.111-88	(67)3427-2255
111.222.333-44	(67)3422-7788

Formas Normais

- **Primeira Forma Normal (1FN)**

- Existem três técnicas principais para conseguir a primeira forma normal para tal relação:

2. Expandir a chave de modo que haverá uma tupla separada na relação original CLIENTE para cada telefone de um CLIENTE, como mostra a Figura (c).

(c) **CLIENTE**

<u>Cpf</u>	Nome	Sexo	Endereço	<u>Telefone</u>
444.555.666-77	João B Silva	M	Rua Arapongas, 1234	(67)3421-1122
444.555.666-77	João B Silva	M	Rua Arapongas, 1234	(67)3910-3344
444.555.666-77	João B Silva	M	Rua Arapongas, 1234	(67)9999-5566
999.666.111-88	Robson Soares	M	Rua dos Ingleses, 3245	(67)3427-2255
111.222.333-44	Jennifer B Souza	F	Rua Cuiabá, 1050	(67)3422-7788

- Essa solução tem a desvantagem de introduzir a *redundância* na relação.

Formas Normais

- **Primeira Forma Normal (1FN)**

- Existem três técnicas principais para conseguir a primeira forma normal para tal relação:
3. Se o *número máximo de valores* for conhecido para o atributo – por exemplo, se for conhecido que *no máximo três telefones* poderão existir para um cliente –, substituir o atributo Telefone pelos três atributos atômicos: Telefone1, Telefone2 e Telefone3.
- Essa solução tem a desvantagem de introduzir *valores NULL* se a maioria dos clientes tiver menos de três telefones.

Formas Normais

- **Primeira Forma Normal (1FN)**

- Portanto, devemos remover o atributo Telefone que viola a 1FN e colocá-lo em uma relação separada TELEFONE_CLIENTE, junto com a chave primária Cpf de CLIENTE.
- A chave primária dessa relação é a combinação {Cpf, Telefone}, como mostra a Figura.

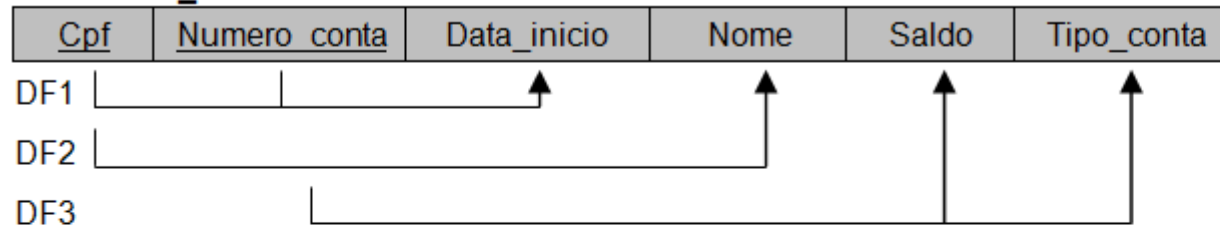
TELEFONE_CLIENTE	
<u>Cpf</u>	<u>Telefone</u>
444.555.666-77	(67)3421-1122
444.555.666-77	(67)3910-3344
444.555.666-77	(67)9999-5566
999.666.111-88	(67)3427-2255
111.222.333-44	(67)3422-7788

Formas Normais

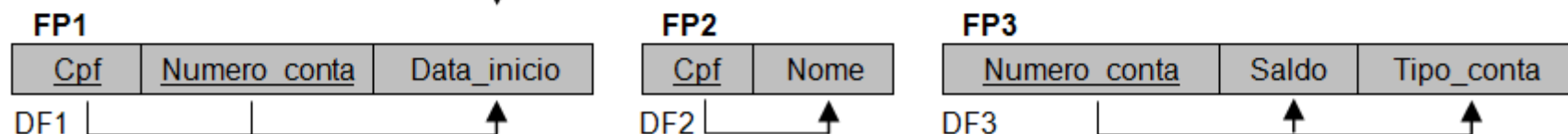
▪ Segunda Forma Normal (2FN)

Definição. Um esquema de relação R está em 2FN se cada atributo não principal A em R for *total e funcionalmente dependente* da chave primária de R .

(a) CLIENTE_CONTA



Normalização 2FN

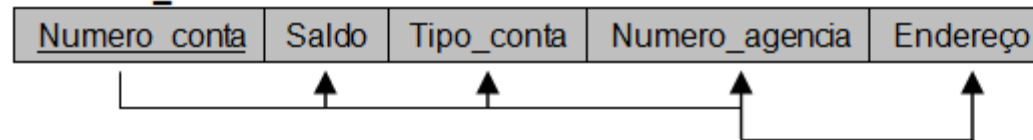


Formas Normais

▪ Terceira Forma Normal (3FN)

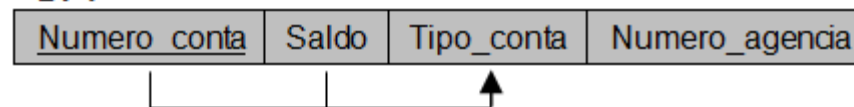
Definição. De acordo com a definição original de Codd, um esquema de relação R está na 3FN se satisfizer a 2FN e nenhum atributo não principal de R for transitivamente dependente da chave primária.

(b) CONTA_AGENCIA

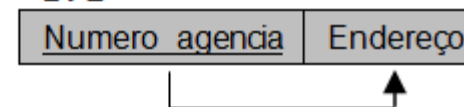


Normalização 3FN

DF1



DF2



Formas Normais

▪ Resumo das formas normais

Forma normal	Teste	Solução (normalização)
Primeira (1FN)	Relação não deve ter atributos multivalorados ou relações aninhadas.	Formar novas relações para cada atributo multivalorado ou relação aninhada.
Segunda (2FN)	Para relações em que a chave primária contém múltiplos atributos, nenhum atributo não chave deverá ser funcionalmente dependente de uma parte da chave primária.	Decompor e montar uma nova relação para cada chave parcial com seu(s) atributo(s) dependente(s). Certificar-se de manter uma relação com a chave primária original e quaisquer atributos que sejam total e funcionalmente dependentes dela.
Terceira (3FN)	A relação não deve ter um atributo não chave determinado funcionalmente por outro atributo não chave (ou por um conjunto de atributos não chave). Ou seja, não deve haver dependência transitiva de um atributo não chave sobre a chave primária.	Decompor e montar uma relação que inclua o(s) atributo(s) não chave que determina(m) funcionalmente outro(s) atributo(s) não chave.

Exemplos

- **Exemplo 1:**
- Considere a seguinte relação utilizada para armazenar itens de produtos que são pedidos ao fornecedor.

Pedido_Cliente (NumPedido, NumProduto, IDCliente, Data_Pedido, Preço_Produto, Quantidade_Produto_Pedido, Nome_Cliente, Endereco_Cliente)

Na relação acima, uma tupla descreve um pedido de um cliente que pode conter vários produtos diferentes.

Derive um esquema de banco de dados relacional para esta aplicação, de modo que cada relação esteja na 3FN.

Exemplos

▪ Solução

- Pedido (NumPedido, Data_Pedido, #IDCliente)
- Produto_Pedido (#NumPedido, #NumProduto, Quantidade_Produto_Pedido)
- Produto (NumProduto, Preço_Produto)
- Cliente (IDCliente, Nome_Cliente, Endereco_Cliente).

Exemplos

- **Exemplo 2:**

- Considere quatro relações compostas da seguinte forma:
- FUNC_PROJ1 (Cpf, Projnumero, Horas, Fnome, Projnome, Projlocal)
- FUNC_PROJ2 (Cpf, Projnumero, Horas)
- FUNC_PROJ3 (Projnumero, Projnome, Projlocal)
- FUNC_PROJ4 (Cpf, Fnome)

Exemplos

- Quanto às regras de normalização, é correto afirmar que FUNC_PROJ1, FUNC_PROJ2 e FUNC_PROJ3, estão normalizadas, respectivamente, até a:
 - a) 1FN, 2FN e 2FN.
 - b) 1FN, 3FN e 3FN.
 - c) 2FN, 2FN e 2FN.
 - d) 2FN, 3FN e 3FN.
 - e) 3FN, 3FN e 3FN.

Exemplos

- **Exemplo 3:**

- Normalize até a 3ª. Forma Normal, quando for o caso. Justificar sua normalização através das dependências funcionais.:
- OrdemCompra (codOrdem, dtEmissão, codFornecedor, nomeFornecedor, endereçoFornecedor, codMaterial (n vezes), descriçãoMaterial (n vezes), qtComprada (n vezes), valorUnitário (n vezes), valorTotalItem (n vezes), valorTotalOrdem).

Exemplos

▪ Exemplo 3:

1FN

ORDEMCOMPRA(codOrdem, dtEmissão, codFornecedor, nomeFornecedor, endereçoFornecedor, codMaterial, descriçãoMaterial, qtComprada, valorUnitário, valorTotalItem, valorTotalOrdem)

DF:

codFornecedor → nomeFornecedor, endereçoFornecedor

codOrdem, codMaterial → descriçãoMaterial, qtComprada, valorUnitario, valorTotalItem, codFornecedor

codOrdem → dtEmissao, valorTotalOrdem

Exemplos

- **Exemplo 3:**

2FN

ORDEMCOMPRA(#codOrdem, codMaterial, descricaoMaterial,
qtComprada, valorUnitario, valorTotalItem, #codFornecedor)

FORNECEDOR(codFornecedor, nomeFornecedor, enderecoFornecedor)

DADOSCOMPRA(codOrdem, dtEmissao, valorTotalOrdem)

Exemplos

▪ Exemplo 3:

3FN

ORDEMCOMPRA(#codOrdem, #codMaterial, qtComprada,
valorUnitario, valorTotalItem, #codFornecedor)

FORNECEDOR(codFornecedor, nomeFornecedor, endereçoFornecedor)

DADOSCOMPRA(codOrdem, dtEmissao, valorTotalOrdem)

MATERIAIS(codMaterial, descricaoMaterial)

Dúvidas



José Osvano da Silva
joseosvano@unipac.br