

Normalização – 2ª Forma Normal (2FN)

Introdução

Primeiramente, precisamos entender a Primeira Forma Normal (1FN) para compreendermos a Segunda Forma Normal (2FN), que é uma normalização mais “fina” que a 1ª Forma Normal.

Em sequência, precisamos, antes de esclarecermos a 2ª Forma Normal, dominar um pouco mais do que são Dependências Funcionais.

Dependências Funcionais:

Dependências Funcionais são restrições ao conjunto de relações validas, para melhor entendermos:

- $A \rightarrow B$ é um tipo de dependência funcional onde:
 - i) A funcionalmente determina B;
 - ii) B depende funcionalmente de A;
 - iii) Portanto, B não existe sem A (B é função de A).

Para cada valor de A, existe apenas um valor de B.

- $A \nrightarrow B$, (negação de $A \rightarrow B$):
 - i) A não determina B;
 - ii) B independe de A.
- A ou B podem ser conjunto de atributos:
 - i) Por exemplo: A tabela Esportista (que tem como Identidade e Habilidade, temos):
 - Identidade $\rightarrow$ Nome
 - Identidade $\rightarrow$ Endereço
 - Identidade $\nrightarrow$ Habilidade
 - Nome $\nrightarrow$ Identidade
 - Endereço $\nrightarrow$ Identidade

- Habilidade $\neg \rightarrow$ Identidade
- Identidade $\rightarrow$ Nome, Endereço

- Outra ocasião, analisando a tabela abaixo temos:

A	B	C	D
B	4	20	2
C	5	15	3
D	1	10	4
C	5	15	3
B	7	1	2
B	4	20	2
A	2	30	1
A	6	25	1
A	2	30	1
D	1	10	4

- i) $A \neg \rightarrow B$
- ii) $A \rightarrow D$
- iii) $A, B \rightarrow C$

Conclusão de Dependências Funcionais

A ideia de normalização “fina” consiste em agrupar em apenas uma tabela, dois conjuntos de atributos X e Y, onde $X \rightarrow Y$.

Então, podemos concluir que X é a chave da tabela, e Y é o complemento da chave.

2ª FORMA NORMAL (2FN)

Uma tabela está na Segunda Forma Normal (2FN) se ela é 1FN e todo atributo do complemento de uma chave candidata é **totalmente funcionalmente dependente** daquela mesma chave.

$A, B, C \Rightarrow D$ (D é totalmente funcionalmente dependente de {A, B, C}) se para todo valor de {A, B, C} só existe um valor de D, e se D não é funcionalmente dependente de A, ou B, ou C.

- Exemplo:

i) Pratica-Esporte (Identidade, Nome, Endereço, Esporte)

Chave Candidatas	Complementos da Chave
Identidade	{Nome, Endereço, Esporte}
{Nome, Endereço}	{Identidade, Esporte}

Então, temos que:

- Identidade $\rightarrow$ Nome
- Identidade $\rightarrow$ Endereço
- Identidade $\nrightarrow$ Esporte
- {Nome, Endereço} $\Rightarrow$ Identidade
- {Nome, Endereço} $\nRightarrow$ Esporte

Tomamos por conclusão então, que o atributo ESPORTE, pode ser retirado da relação Prática-Esporte.

Criamos então uma tabela Esportista com a chave principal (Identidade) e os complementos da chave (Nome, Endereço), e resumimos a tabela Prática-Esporte a apenas duas chaves (Identidade e Esporte).

Podemos então, com essa nova tabela, atualizar o Endereço sem alterar todos os outros complementos e também atualizar Esporte, sem alterar os dados do Esportista.

Notamos que assim, fica mais fácil atualizar e encontrar o Cliente e o Esporte.

- ii) Outro exemplo pode ser uma tabela de Estudante-Disciplina

E#	ENome	Sexo	Idade	D#	DNome	Opinião
E1	João	M	20	D1	Mat	BOA
E1	João	M	20	D2	Fis	MÁ
E1	João	M	20	D3	Qui	BOA
E2	Maria	F	19	D2	Fis	BOA
E2	Maria	F	19	D3	Qui	MÁ
E2	Maria	F	19	D4	Bio	MÁ
E3	Luis	M	22	D2	Fis	BOA
E3	Luis	M	22	D4	Bio	BOA

Tiramos então como Chaves Candidatas e Complementos da Chave:

Chaves Candidatas	Complementos da Chave
{E#, D#}	ENome, Sexo, Idade, DNome, Opinião
{E#, DNome}	ENome, Sexo, Idade, D#, Opinião

Tiramos como conclusão então, onde {E#, D#} são as chaves candidatas, temos:

- {E#, D#} $\rightarrow$ ENome (E# $\rightarrow$ ENome)
- {E#, D#} $\rightarrow$ Sexo (E# $\rightarrow$ Sexo)
- {E#, D#} $\rightarrow$ Idade (E# $\rightarrow$ Idade)
- {E#, D#} $\rightarrow$ DNome (D# $\rightarrow$ DNome)
- {E#, D#} $\rightarrow$ Opinião

Agora, tirando como conclusão, onde {E#, DNome} são as chaves candidatas, temos:

- {E#, DNome} $\rightarrow$ ENome (E# $\rightarrow$ ENome)
- {E#, DNome} $\rightarrow$ Sexo (E# $\rightarrow$ Sexo)
- {E#, DNome} $\rightarrow$ Idade (E# $\rightarrow$ Idade)

- $\{E\#, DNome\} \rightarrow D\#$ (DNome $\rightarrow$ D#)
- $\{E\#, DNome\} \rightarrow Opini\tilde{a}o$

Concluimos ent\~ao, que ENome, Sexo, Idade e DNome devem ser retirados de Estudante-Disciplina. Portanto, podemos criar ent\~ao duas tabelas separadas (uma de Estudantes e outra de Disciplinas) e outra que depende das duas, onde s\~ao:

Tabela de Estudantes:

E#	ENome	Sexo	Idade
E1	Jo\~ao	M	20
E2	Maria	F	19
E3	Luis	M	22

Tabela de Disciplinas:

D#	DNome
D1	Mat
D2	Fis
D3	Qui
D4	Bio

Tabela de Estudante-Disciplina:

E#	D#	Opini\~ao
E1	D1	BOA
E1	D2	M\~A
E1	D3	BOA
E2	D2	BOA
E2	D3	M\~A
E2	D4	M\~A
E3	D2	BOA
E3	D4	BOA

Analisando a tabela Estudante-Disciplina, notamos que a o modelo 2FN \u00e9 mais organizado que o modelo 1FN.

Conclusão

Assim como o modelo 1FN, o modelo 2FN tem anomalias, que são:

- Inserção: Só podemos criar cargos se houver empregados designados para ele;
- Remoção: Se removermos um empregado que ocupa unicamente um cargo na empresa, perderemos a informação deste cargo;
- Atualização: Se um cargo muda de nome precisaremos mudar todas as tabelas em que este cargo aparece.