

Banco de Dados

Aula 07 - Normaliza o Avan ada

Apresentação

Na aula anterior, você aprendeu como evitar anomalias de inserção, remoção e atualização, através de duas técnicas básicas de normalização (primeira e segunda forma normal). Nesta aula, você irá aprender uma técnica avançada de normalização que permite reduzir ainda mais os efeitos das anomalias.



Vídeo 01 - Apresentação

Objetivos

- Normalizar tabelas de acordo com a Terceira Forma Normal (3FN).

Normalização

Dando continuidade ao nosso estudo sobre Normalização, nesta aula, você irá aprender como normalizar tabelas de acordo com a terceira forma normal. A 3FN é tratada separadamente da 1FN e 2FN, porque exige um maior refinamento do seu modelo relacional. Esse refinamento, por sua vez, irá exigir um maior nível de detalhe na operação de normalização e, conseqüentemente, maior complexidade das relações. Além da terceira forma normal, existem várias outras formas normais, tais como: 4FN, 5FN, Boyce-Codd Normal Form (BCNF), Domain Key normal form (DKNF), entre outras.

Nesta disciplina, iremos estudar até a 3FN, porque concordamos com a opinião de alguns autores — Powell (2006) e Connolly, Begg (2000), da literatura de banco de dados — que afirmam que, acima da 3FN, o resultado da normalização pode ficar tão complexo que compromete o desempenho do banco de dados.



Vídeo 02 - Dependência Funcional Transitiva

Terceira Forma Normal (3FN)

Uma tabela está na Terceira Forma Normal (3FN) se e somente se ela estiver na 1FN e na 2FN e todos os atributos não chave primária puderem ser obtidos somente através da chave primária.

Ao ler a definição da 3FN, você pode estar se perguntando: qual é a diferença entre a 3FN e a 2FN? Bem, para responder à sua pergunta, vamos utilizar o exemplo mostrado na **Figura 1**. Nesse exemplo, temos a estrutura da tabela Funcionário, em que a chave primária é o número do funcionário, FuncN. A tabela Funcionário serve

para armazenar todos os dados do funcionário (nome, cargo, salário), incluindo a agência onde ele trabalha (NumAg e Endereço). Note que a tabela Funcionário está na 1FN, porque todos os atributos são atômicos, e está na 2FN, porque todos os atributos não chave primária podem ser obtidos através da chave primária.

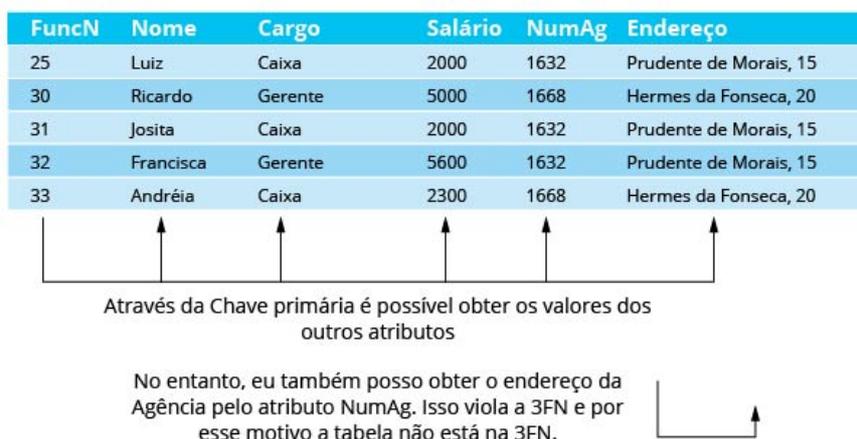
Figura 01 - Exemplo da tabela Funcionário que não está na 3FN.

FuncN	Nome	Cargo	Salário	NumAg	Endereço
25	Luiz	Caixa	2000	1632	Prudente de Moraes, 15
30	Ricardo	Gerente	5000	1668	Hermes da Fonseca, 20
31	Josita	Caixa	2000	1632	Prudente de Moraes, 15
32	Francisca	Gerente	5600	1632	Prudente de Moraes, 15
33	Andréia	Caixa	2300	1668	Hermes da Fonseca, 20

Fonte: Connolly e Begg (2000).

No entanto, a tabela Funcionário não está na 3FN, porque o atributo Endereço da agência onde o funcionário trabalha pode ser obtido por meio da chave primária FuncN **ou** por meio do atributo NumAg. Essa relação de dependência pode ser vista na **Figura 2**. Veja que na definição da 3FN cada atributo só deve ser obtido **única e exclusivamente** através da chave primária. Como isso não acontece na tabela Funcionário, em que você pode obter o endereço da agência por meio de FuncN ou NumAg, tal tabela não está na 3FN. Assim, se você souber apenas o Número da Agência (atributo NumAg), consegue obter o seu endereço.

Figura 02 - Detalhamento do motivo pelo qual a tabela Funcionário não está na 3FN.



Fonte: Adaptado de Connolly e Begg (2000).

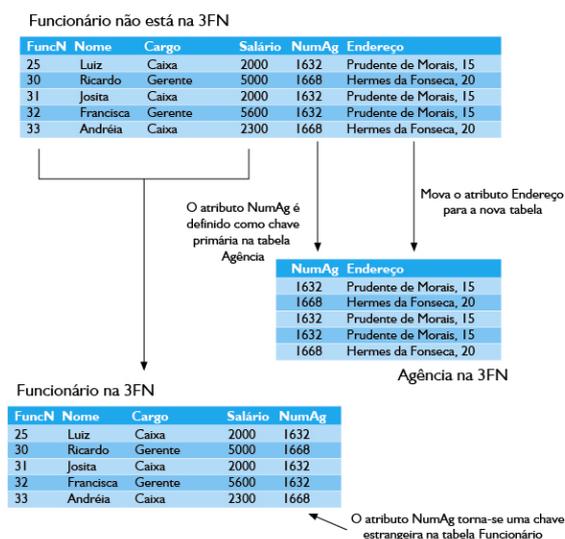
Terceira Forma Normal - Pt.2

Bem, depois de aprender como identificar se uma tabela está ou não na 3FN, agora você vai aprender como fazer a adequação da tabela. Para adequar uma tabela que não está na 3FN, é necessário cumprir os seguintes passos:

1. Criar uma tabela para conter os atributos que não podem ser obtidos exclusivamente da chave primária da tabela original;
2. Definir como chave primária da tabela criada o atributo que é capaz de obter os dados não chaves da tabela original;
3. Mover os atributos não chave que não são obtidos exclusivamente pela chave primária da tabela original para a nova tabela;
4. Definir como chave estrangeira o atributo que é capaz de obter os dados não chaves da tabela original.

A **Figura 3** ilustra os passos listados, utilizando-se do exemplo da tabela Funcionário:

Figura 03 - Normalização da Tabela Funcionário.



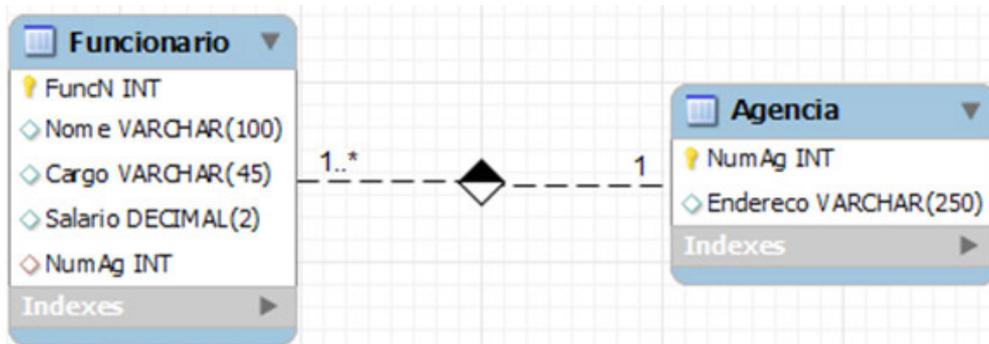
Fonte: Adaptado de Connolly e Begg (2000).

O primeiro passo é criar a tabela Agência para conter os atributos que não podem ser obtidos exclusivamente da chave primária da tabela Funcionário. Depois, você define como chave primária da tabela Agência o atributo NumAg, porque este é capaz de obter os dados não chaves da tabela Funcionário.

Lembre-se de que o atributo Endereço da tabela Funcionário pode ser obtido através da chave primária, como também por meio do atributo NumAg, que não é chave primária. Assim, como Endereço pode ser obtido através de NumAg, você deve escolher NumAg como atributo chave primária da tabela Agência.

Depois de definir NumAg como chave primária da tabela Agência, você deve mover o atributo Endereço da tabela Funcionário para Agência. Finalmente, para manter a integridade com a tabela original, você deve definir uma chave estrangeira entre o atributo NumAg da tabela Funcionário e NumAg da tabela Agência. Para perceber a diferença entre elas, veja na **Figura 4** a nova estrutura das duas tabelas. Observe que antes os dados do atributo Endereço eram duplicados e poderiam gerar anomalias. Agora, o atributo Endereço está separado em uma tabela separada e na 3FN.

Figura 04 - Estrutura das tabelas depois da Normalização.



Vídeo 03 - Terceira Forma Normal

Atividade 01

1. Dê exemplos de anomalias que podem ocorrer na Tabela mostrada na **Figura 1**.
2. A Tabela 1, mostrada logo a seguir, armazena o tempo gasto por funcionários de hotéis. O CPF é a chave primária da tabela. Com base nessas informações, responda às questões a seguir:
 - a. A Tabela 1 é suscetível a anomalias? Caso afirmativo, forneça exemplo de algumas anomalias que podem ocorrer em tal tabela.
 - b. Descreva e ilustre o processo de normalização para a 3FN da Tabela 1.

CPF	NumContrato	HorasSem	Nome	hotelNum	hotelEnd
2532514	C1024	16	Ricardo	H25	Natal
2552478	C1024	24	Luiz	H25	Natal
5874514	C1025	28	Roberta	H4	Pipa
7458472	C1025	16	Andrea	H4	Pipa

Tabela 1 – Exemplo de tabela não normalizada



Vídeo 04 - Resolução de Exercícios

Conclusão

Como foi esclarecido na apresentação da aula, existem outras formas normais que poderiam ser utilizadas para normalizar tabelas. Entretanto, na maioria dos casos, você será capaz de resolver todos os problemas de anomalias através da implementação da 3FN.

Além disso, como já comentamos, alguns autores concordam que normalizações acima da 3FN geram muitas tabelas e prejudicam o desempenho do banco de dados.

Resumo

Nesta aula, você aprendeu como normalizar uma tabela segundo a 3FN. Você viu que quanto maior a forma normal alcançada, menor será a ocorrência de anomalias. Viu ainda que muitos autores defendem que normalizações acima da 3FN podem prejudicar significativamente o desempenho do banco de dados.

Autoavaliação

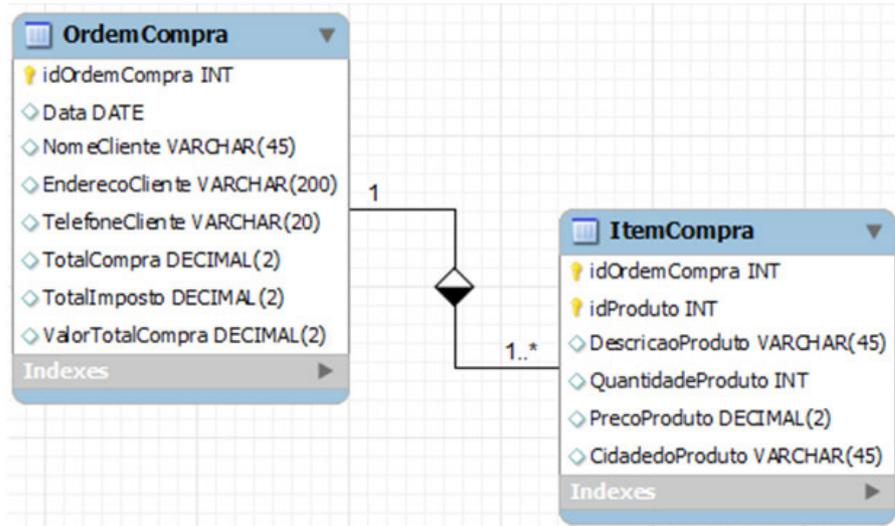
1. Dada a Tabela 2, identifique possíveis anomalias que podem ocorrer.

NomeDisc	ProfNum	ProfNome	EstNum	EstNome	Nota
Banco de Dados	234	Nelio	3410	Ricardo	6,0
fBanco de Dados	234	Nelio	3510	Luiz	7,0
Banco de Dados	234	Nelio	3610	Andrea	8,0
Programação	235	João	3410	Ricardo	6,5
Programação	235	João	3510	Luiz	9,0
Programação	235	João	3610	Andrea	9,5
WebDesing	236	Diego	3410	Ricardo	7,0
WebDesing	236	Diego	3510	Luiz	8,2
WebDesing	236	Diego	3610	Andrea	7,5

Tabela 2 – Tabela que relaciona professor e estudante

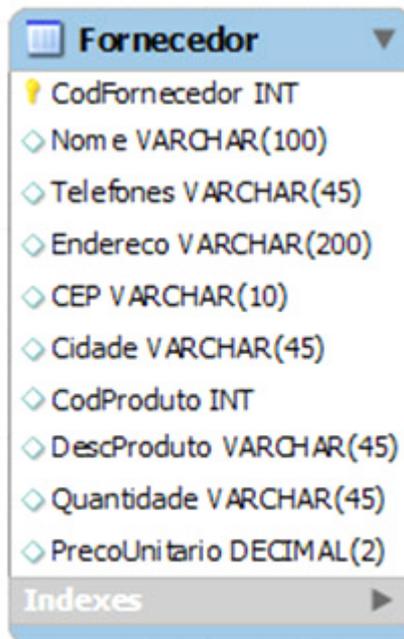
2. Dada a Tabela 2, normalize-a de modo que ela fique na 3FN.
3. Dada a estrutura das tabelas mostradas na Figura 5, normalize-as de modo que ela fique na 3FN.

Figura 05 - Exemplo de estrutura de tabela não normalizada.



4. Normalize a estrutura mostrada na Figura 6, mostrando, passo a passo, como normalizar para a 1FN, depois para a 2FN e por fim para a 3FN.

Figura 06 - Exemplo de estrutura de tabela não normalizada.



Referências

CONNOLLY, Thomas M.; BEGG, Carolyn E. **Database Solutions**: a step-by-step approach to building databases. 2nd ed. New Jersey: Pearson Education Limited, 2000.

DATE, C. J. **Introduction to Database Systems**. 7th ed. 1999.

_____. **Introdução a sistemas de banco de dados**. Rio de Janeiro: Campus, 2000.

ELMASRI, R. E.; NAVATHE, S.B. **Sistemas de banco de dados**. 4. ed. Rio de Janeiro: Addison-Wesley, 2005.

HEUSER, C. A. **Projeto de Banco de Dados**. 6. ed. Porto Alegre: Editora Bookman, 2009.

_____. **Projeto de banco de dados**. 5. ed. Porto Alegre: Sagra-Luzzatto, 2004.

POWELL, Gavin. **Beginning Database Design**. San Francisco: Wiley Publishing, 2006.