



## Cursos


➤ Listagem de disciplinas

Selecione uma disciplina

## Aulas

- 01 Introdu  o a Banco de Dados
- 02 Modelo de Entidade e Relacionamento
- 03 Modelo Relacional
- 04 Transforma  es ER para MR
- 05 Transforma  es ER para MR e dicion rio de dados
- 06 Normaliza  o b sica
- 07 Normaliza  o avan ada
- 08 Introdu  o   Linguagem SQL e Sistemas Gerenciadores de Banco de Dados
- 09 Linguagem SQL - cria  o, inser  o e modifica  o de tabelas
- 10 Linguagem SQL - Consulta simples de tabelas
- 11 Linguagem SQL - Consulta avan ada de tabelas
- 12 Linguagem SQL - Altera  o da estrutura de tabelas e ambientes de m ltiplas tabelas
- 13 Linguagem SQL - Subconsultas
- 14 Linguagem SQL - VIS  ES
- 15 Linguagem SQL - STORED PROCEDURES
- 16 Linguagem SQL - Fun  es
- 17 Linguagem SQL - Seguran a
- 18 Engenharia Reversa
- 19 Utilizando SQL em Java
- 20 Utilizando conceitos avan ados de SQL em Java

⬅ Voltar    🖨 Imprimir    ⬆ Topo



### Sistemas de Banco de Dados

Aula 7 – Normaliza  o avan ada

**Professores autores**


**N lio Alessandro Azevedo Cacho**  
[neliocacho@ect.ufrn.br](mailto:neliocacho@ect.ufrn.br)

**Xiankleber Cavalcante Benjamim**  
[xianklebercb@gmail.com](mailto:xianklebercb@gmail.com)



### Apresenta  o

Na aula anterior voc  aprendeu como evitar anomalias de inser  o, remo  o e atualiza  o atrav s de duas t cnicas b sicas de normaliza  o. Nesta aula voc  ir  aprender uma t cnica avan ada de normaliza  o que permite reduzir ainda mais os efeitos das anomalias.



### Objetivo

**Ao final desta aula, voc  ser  capaz de:**

- Normalizar suas tabelas segundo a Terceira Forma Normal (3FN).

## Normaliza  o

Dando continuidade ao nosso estudo sobre Normalização, nesta aula você irá aprender como normalizar tabelas segundo a terceira forma normal. A 3FN é tratada separadamente da 1FN e 2FN porque esta exige um maior refinamento do seu modelo relacional. Este refinamento irá exigir um maior nível de detalhe na operação de normalização e, consequentemente, 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) etc.

Nesta disciplina iremos estudar até a 3FN porque concordamos com a opinião de alguns autores [Powell e Connolly, Begg] 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.

### 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 a sua pergunta vamos utilizar o exemplo mostrado na Figura 1. Neste exemplo, temos a estrutura da tabela Funcionário, onde 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.

| <u>FuncN</u> | Nome          | Cargo   | <u>Salario</u> | <u>NumAg</u> | <u>Endereco</u>        |
|--------------|---------------|---------|----------------|--------------|------------------------|
| 25           | Luiz          | Caixa   | 2000           | 1632         | Prudente de Moraes, 15 |
| 30           | Ricardo       | Gerente | 5000           | 1668         | Hermes da Fonseca, 20  |
| 31           | <u>Josita</u> | Caixa   | 2000           | 1632         | Prudente de Moraes, 15 |
| 32           | Francisca     | Gerente | 5600           | 1632         | Prudente de Moraes, 15 |
| 33           | Andréa        | Caixa   | 2300           | 1668         | Hermes da Fonseca, 20  |

**Figura 1** - Exemplo da tabela Funcionário que não está na 3FN  
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. Esta 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, onde você pode obter o endereço da agência por meio de FuncN ou NumAg, então tal tabela não está na 3FN. Por exemplo, se você souber apenas do Número da Agência (atributo NumAg), você consegue obter o seu endereço.

| FuncN | Nome      | Cargo   | Salario | NumAg | Endereco               |
|-------|-----------|---------|---------|-------|------------------------|
| 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    | Andréa    | Caixa   | 2300    | 1668  | Hermes da Fonseca, 20  |

Através da Chave primária é possível obter todos os valores dos outros atributos

No entanto, eu também posso obter o endereço da Agência pelo atributo NumAg. Isto viola a 3FN e por este motivo a tabela não está na 3FN.

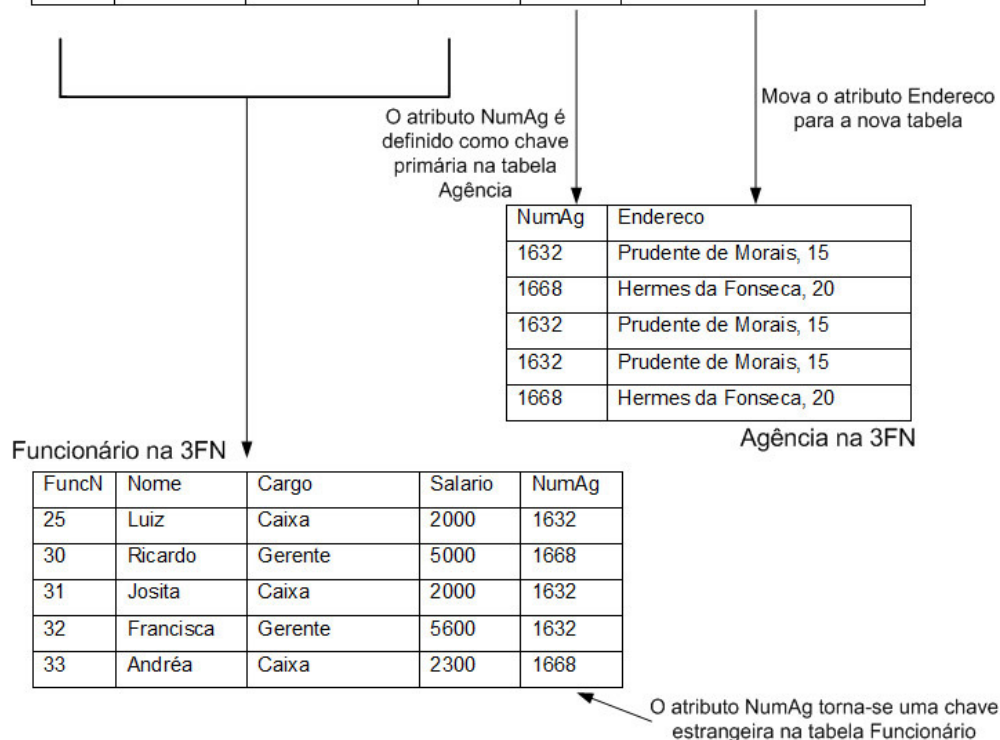
**Figura 2** – Detalhamento do motivo pelo qual a tabela Funcionário não está na 3FN. Fonte: Adaptado de Connolly e Begg (2000).

Bem, depois de aprender como identificar se uma tabela está ou não na 3FN, agora vamos aprender como fazer a adequação da tabela. Para adequar uma tabela que não está na 3FN é necessário fazer 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.

## Funcionário não esta na 3FN

| FuncN | Nome      | Cargo   | Salario | NumAg | Endereco               |
|-------|-----------|---------|---------|-------|------------------------|
| 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    | Andréa    | Caixa   | 2300    | 1668  | Hermes da Fonseca, 20  |



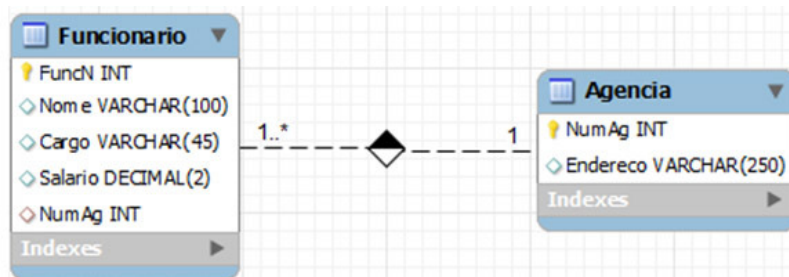
**Figura 3** – Normalização da Tabela Funcionário  
Fonte: Adaptado de Connolly e Begg (2000).

Os passos listados anteriormente são ilustrados na Figura 3 utilizando-se o exemplo da tabela Funcionário. 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 Endereco da tabela Funcionário pode ser obtido através da chave primária, como também por meio do atributo NumAg, que não é chave primária. Assim, como Endereco 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 Endereco 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, veja na Figura 4 a nova estrutura das duas tabelas. Observe que antes os dados do atributo Endereco eram duplicados e poderiam gerar anomalias. Agora, o atributo Endereco está separado em uma tabela separada e na 3FN.



**Figura 4** – Estrutura das tabelas depois da Normalização



## Pratique 01

1. Dê exemplos de anomalias que podem ocorrer na Tabela mostrada na Figura 1.
2. A tabela mostrada logo a seguir armazena o tempo gasto por funcionários de hotéis. O CPF é a chave primária da tabela.
  - 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

Como foi dito na apresentação desta 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, 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. No entanto, muitos autores defendem que normalizações acima da 3FN podem prejudicar significativamente o desempenho do banco de dados.



## Evolucao

Agora vamos relembrar e fixar nossa aula.

1. Dada a Tabela 2, identifique possíveis anomalias que podem ocorrer.
2. Dada a Tabela 2, normalize a tabela de modo que ela fique na 3FN.

| NomeDisc       | ProfNum | ProfNome | EstNum | EstNome | Nota |
|----------------|---------|----------|--------|---------|------|
| Banco de Dados | 234     | Nelio    | 3410   | Ricardo | 6,0  |
| Banco 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 |

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

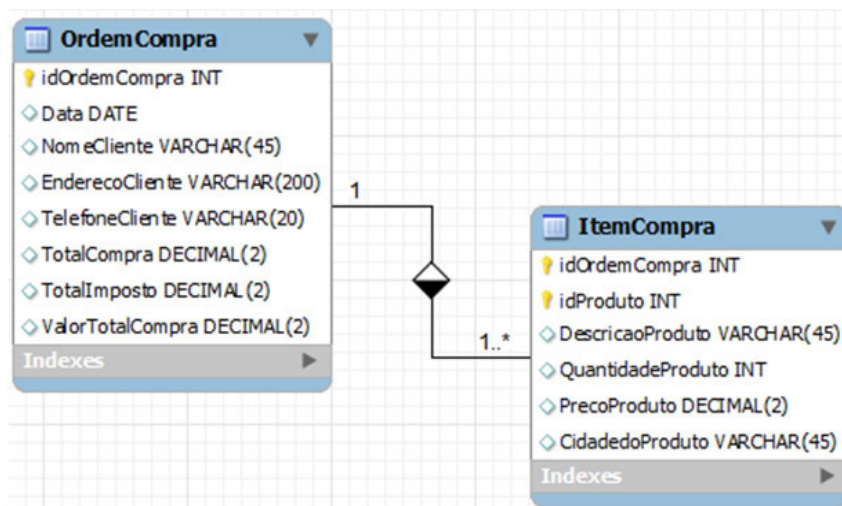


Figura 5 – Exemplo de estrutura de tabela não normalizada

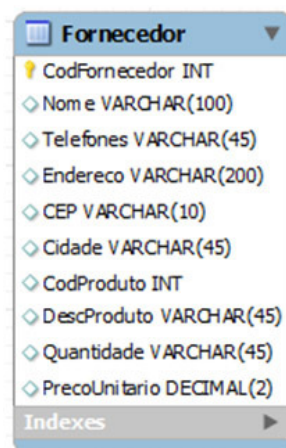


Figura 6 – Exemplo de estrutura de tabela não normalizada

## Referencias

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

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

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,

