

## **CAPÍTULO V – NORMALIZAÇÃO DE DADOS**

### **INTRODUÇÃO**

Para entendermos os conceitos de normalização de dados, inicialmente, precisa-se aprender a visualizar o banco de dados em sua estrutura. Para isso, podemos dizer que o banco de dados se distribui da seguinte forma:

- Banco de Dados = Coleção de Arquivos;
- Arquivos = Coleção de Registros;
- Registro (Tuplas) = Coleção de Campos;
- Campo = Coleção de Caracteres;
- Caracter = Alfa-numéricos ou símbolos.

Atualmente, existem 9 técnicas que podem ser utilizadas no desenvolvimento de um banco de dados nas fases iniciais, as quais podemos citar:

1. Contextualização;
2. Objetivação;
3. Intitulação;
4. Especificação de Requisitos;
- 5. Normalização;**
6. Modelagem;
7. Trigramação;
8. Dicionarização;
9. Auditoria das fases de Análise e de Projeto Lógico do Sistema de BD<sup>1</sup>.

Para este curso, estaremos focados no que diz respeito à Normalização como ferramenta para o desenvolvimento de um banco de dados.

Os conceitos apresentados neste material, em seu todo ou em partes, foram retirados do livro “Projeto de Banco de Dados: uma visão prática”, de Machado, F. N. R, e Abreu, M. P., Editora Érica, 1995, Capítulo 12, Normalização.

Estes conceitos poder ser aplicados às duas formas mais comuns de Normalização: Sentido de cima para baixo (TOP-DOWN) e Sentido de baixo para cima (BOTTON-UP); os quais estaremos abordando em seguida.

---

<sup>1</sup>

Banco de Dados.

## **Normalização**

A Normalização tem como objetivo produzir um projeto de Banco de Dados capaz de ser manipulado, cada vez mais, como um simples conjunto de operações, visando minimizar a inconsistência de dados.

A teoria da Normalização é expressa tradicionalmente através de um conjunto de Formas Normais, que otimizam a estrutura e o conteúdo das relações entre as entidades.

O conceito de Normalização foi introduzido por E. F. Codd, em 1970, utilizando a primeira forma normal – 1FN. Esta técnica, pode-se dizer, é um processo matemático formal que se fundamenta na teoria dos conjuntos.

Através do processo de normalização podemos substituir, de forma gradual, um conjunto de Entidades e Relacionamentos para um modelo mais “adequado”, em relação às Anomalias de Atualização (Inclusão<sup>2</sup>, Alteração<sup>3</sup> e Exclusão<sup>4</sup>), as quais podem causar certos problemas:

- Grupos repetitivos (atributos multivalorados) de dados;
- Dependências parciais em relação a uma chave concatenada;
- Redundâncias de dados desnecessários;
- Perdas acidentais de informação;
- Dificuldade na representação de fatos da realidade observada;
- Dependências transitivas entre atributos.

Estes problemas podem ser minimizados através da aplicação da Normalização, tornando o modelo de dados bastante estável e sujeitos a poucas manutenções.

### Sentido de cima para baixo (TOP-DOWN)

Logo após a definição do modelo de dados, podemos aplicar à Normalização com o objetivo de obter uma síntese dos dados, bem como a decomposição das entidades e relacionamentos em elementos mais estáveis.

### Sentido de baixo para cima (BOTTON-UP)

Nesta técnica, aplica-se a Normalização usando relatórios, formulários e documentos utilizados pelo estudo do modelo de dados. Como exemplo, podemos citar uma nota fiscal ou cupom fiscal ao qual é aplicado em uma determinada empresa para o registro dos produtos comercializados. Os campos existentes nesta nota farão parte essencial do processo de Normalização do modelo relacional.

---

<sup>2</sup> Ao se incluir um novo cliente, por exemplo, ele tem que estar relacionado com uma venda.

<sup>3</sup> Caso o fabricante de um produto altere o preço de uma determinada classe de produto, será preciso percorrer toda a Entidade para se realizar múltiplas alterações.

<sup>4</sup> Ao se excluir um cliente, por exemplo, os dados referentes às suas compras poderão ser perdidos.

## **Dependência Funcional**

Para se descrever as próximas formas normais existentes, faz-se necessária a introdução deste conceito importante na aplicação da Normalização, a qual maior parte da teoria de Normalização foi baseada.

Dada uma entidade qualquer, dizemos que o atributo ou conjunto de atributos de A é dependente funcionalmente de outro atributo de B, neste caso há dependência funcional.

Em outras palavras, podemos dizer que em uma entidade TB\_ITENSNOTA, o atributo ITN\_CODPRODUTO depende funcionalmente do atributo PRO\_CODIGO que se encontra na entidade TB\_PRODUTO.

A análise das relações existentes entre os atributos de uma entidade deve ser feito a partir de um conhecimento prévio da sistemática aplicada nas relações entre as entidades, para que esta possa ser modelada, como no exemplo anterior.

## Exemplo 1

Dados da Empresa			Data: 01/01/0101		NF Nº: 123
CLIENTE: João da Silva				RG: 00000000000-00	
CPF:000000000000-00		CIDADE: Medianeira			UF: PR
ENDEREÇO: R. Bahia		Nº:1234	BAIRRO: Centro		
Código	Descrição	Und. de Medida	Quantidade	Vlr Unitário	Vlr Total
1	Arroz	KG	2	1,99	3,98
2	Sabonete	Unidade	1	0,79	0,79
3	Leite	Litro	3	2,12	6,36
			Valor TOTAL : 11,13		

### Definição de todos os atributos que o documento possui:

**Nota** (número da nota, data, cliente, RG, CPF, cidade, UF (Unidade da Federação), endereço, nº, bairro, código\_produto\_1, código\_produto\_2, código\_produto\_3, ..., descrição\_produto\_1, descrição\_produto\_2, descrição\_produto\_3,..., unidade\_produto\_1, unidade\_produto\_2, unidade\_produto\_3, quantidade\_produto\_1, quantidade\_produto\_2, quantidade\_produto\_3, ...,valor\_unitário\_produto\_1, valor\_unitário\_produto\_2, valor\_unitário\_produto\_3, ..., valor\_total\_produto\_1, valor\_total\_produto\_2, valor\_total\_produto\_3,..., valor\_total\_da\_nota);

- Os campos sublinhados indicam as chaves-primárias.

### ***Aplicação da 1ª Forma Normal – 1FN***

Existem situações em que nos deparamos com algumas informações que se repetem (atributo multivalorado) dentro de uma única linha e possuindo ligação a uma chave primária.

A 1FN<sup>5</sup> diz que cada ocorrência da chave primária deve corresponder a somente uma informação de cada atributo, ou seja, a entidade não deve conter atributos repetidos ou multivalorados, ou ainda, os atributos não-chave deverão ser atômicos (únicos).

Ao observarmos que certo grupo de atributos não-chave não são atômicos, ao longo do processo de entrada de dados, verificamos que este atributo deverá ser decomposto em uma nova entidade.

Nas novas entidades criadas, a chave primária será a concatenação da chave primária da entidade original, o que mantém desta forma o relacionamento entre elas, como também o conjunto de atributos que se repetem.

Por exemplo:

A 1FN diz que todos os atributos devem conter um valor atômico (único), ou seja, os dados inclusos dentro do campo não devem se repetir.

EX: o campo cliente vai ser preenchido quantas vezes na nota acima? Apenas uma vez certo? Já os campos que se referem aos itens da nota serão acrescidos uma ou muitas vezes.

**Nota** (número da nota, data, cliente, RG, CPF, endereço, nº, bairro, cidade, UF, valor\_total)

Os campos que se repetem são destinados a uma nova tabela, no nosso exemplo chamado de itens da nota. A chave principal também é inclusa na nova tabela para manter o relacionamento entre elas.

**Itens\_Nota** (número da nota, código, descrição, unidade\_de\_medida, valor\_unitário, quantidade, valor\_total)

- Note as tabelas que saem na 1FN são tabelas associativas.

<sup>5</sup>

Primeira Forma Normal.

### ***Aplicação da 2ª Forma Normal – 2FN***

A 2FN<sup>6</sup> diz que todos os atributos que não for chave têm que ser dependente direto da chave, ou seja, os atributos que contidos na nota, tem que depender diretamente da chave número da nota.

Inicialmente observa-se a entidade que possui chave primária concatenada. Para aquelas que satisfazem esta condição, analisar se existe algum atributo ou conjunto de atributos com dependência direta à chave primária.

Ex: o atributo cliente é dependente direto da nota, pois é o cliente desta nota. Já atributo RG, CPF são dependentes do cliente, não tendo nenhuma relação direta com a nota.

**Nota** (número da nota, data, código\_cliente, endereço, nº,bairro, código\_cidade, valor\_total);

- Os atributos adicionados acima “código\_cliente e código\_cidade” são para manter o relacionamento entre as entidades.
- Verifique que o campo código de endereço deve permanecer na entidade Nota, tendo em vista que se a caso deseja-se buscar uma nota emitida a 5 anos atrás cujo cliente alterou seu endereço para outra rua, o endereço que deverá aparecer na nota emitida a 5 anos atrás deve ser o correspondente ao endereço antigo do cliente e não o novo endereço.

**Cliente** (código, nome, RG, CPF);

- Os campos dependentes do cliente vêm para a tabela cliente.

**Cidade** (código, descrição, UF);

- O atributo UF que depende apenas da cidade vem para a tabela cidade.

**Itens nota** (número da nota, código produto, quantidade, valor\_unitário, valor\_total);

- O atributo “código\_produto” fica na tabela associativa para manter o relacionamento entre Nota, Itens da Nota e Produtos.

**Produto** (código, descrição, unidade\_de\_medida, valor\_unitário);

- Os atributos dependentes de produto vêm para tabela produto, repare que o atributo valor unitário foi duplicado, isso pela necessidade do valor unitário de um produto sofrer mudanças, em notas diferentes o valor unitário pode estar diferente também.

---

<sup>6</sup>

Segunda Forma Normal.

### ***Aplicação da 3ª Forma Normal – 3FN***

A 3FN<sup>7</sup> diz que o atributo precisa estar na segunda forma normal, e todos os atributos que não são chave, não podem depender de outros atributos que também não são chave.

Uma Entidade está na 3FN se nenhum de seus atributos possui dependência em relação a outros atributos da entidade. Ao verificar a existência de dependência entre as Entidades e seus atributos, devemos criar uma nova entidade que contenha os atributos dependentes.

**Nota** (número da nota, data, código\_cliente, código\_endereço, código\_bairro, nº, código\_cidade);

- Note que o atributo valor total da nota foi retirado na 3FN, todos os atributos que são gerados a partir do resultado de dois ou mais atributos são retirados, devido à consistência dos dados.

**Endereço** (código, descrição);

- Aplicando a 3FN o endereço sai da tabela nota, pois o endereço não é um atributo chave e refere-se ao endereço do cliente nesta nota, então o endereço vai também para a tabela de clientes.

**Bairro** (código, descrição);

- Aplicando a 3FN o bairro sai da tabela nota, pois o bairro não é um atributo chave e refere-se ao bairro do cliente nesta nota, então o endereço vai também para a tabela de clientes.

**Cliente** (código, nome, RG, CPF, nº, código endereço, código cidade);

- Os campos chaves código\_endereço e código\_cidade, vêm pra tabela de clientes para fazer o relacionamento entre as tabelas.

**Cidade** (código, descrição, UF);

**Itens nota** (número da nota, código produto, quantidade, valor\_unitário);

- O atributo valor total sai, pois ele é resultado de outros dois atributos, a quantidade vezes o valor unitário.

**Produto** (código, descrição, unidade\_de\_medida);

---

<sup>7</sup>

Terceira Forma Normal.

### Modelo Entidade – Relacionamento - MER

Abaixo, demonstra-se o MER do exemplo citado anteriormente.

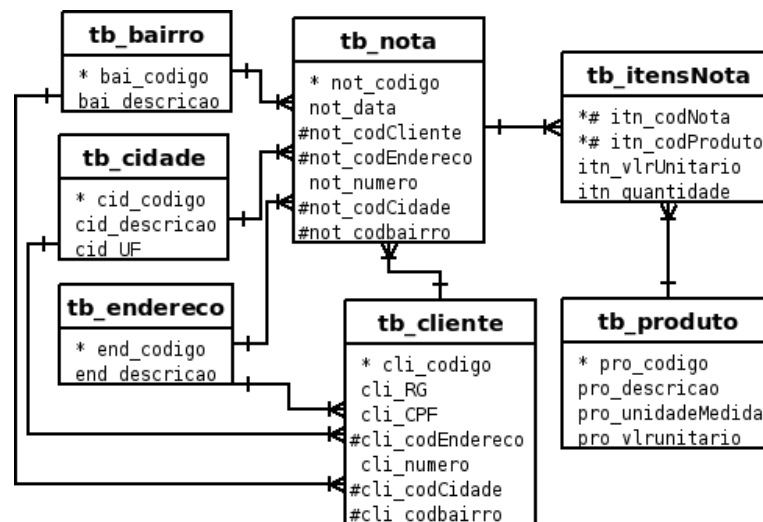


Figura 1 - Nota Fiscal.





***Definição de atributos contidos no formulário:***

**Requerimento** (código, nome, sexo, data\_nascimento, categoria\_funcional, lotação\_atual, data\_admissão, cidade, escolaridade, grau, cursos\_realizados, experiências\_profissionais, atribuições\_desenvolvidas, setor\_que\_trabalhou, data\_entrada, data\_saída, motivo\_saída, atribuições\_que\_gostaria, horário, setores/divisões/diretorias\_gostaria, motivos\_descontente, observações, data\_preenchimento, questão1, questão2, questão3, questão4, questão5, questão6, questão7, resultado, justificativa, data\_resultado)

- Os campos sublinhados indicam as chaves-primárias.

### ***Aplicação da 1ª Forma Normal - 1FN***

A 1FN diz que todos os atributos devem conter um valor atômico (único), ou seja, os dados inclusos dentro do campo não devem se repetir.

Ex: o campo que representa os cursos feitos pelo requerente será preenchido quantas vezes no formulário? Depende de quantos cursos ele fez, na possibilidade de ser preenchido duas vezes o campo é retirado para uma tabela associativa.

**Remanejamento** (código, nome, sexo, data\_nascimento, categoria\_funcional, lotação\_atual, data\_admissão, cidade, data\_preenchimento, questão1, questão2, questão3, questão4, questão5, questão6, questão7, resultado, justificativa, data\_resultado);

- Ficaram na tabela remanejamento apenas os campos com valores únicos, os campos que podem ser preenchidos uma ou mais vezes são retirados para as tabelas associativas.

**Remanejamento/escolaridade** (código remanejamento, escolaridade, grau);

**Remanejamento/Cursos** (código remanejamento, cursos);

**Remanejamento/experiências profissionais** (código remanejamento, experiências\_profissionais);

**Remanejamento/atribuições desenvolvidas** (código remanejamento, atribuições);

**Remanejamento/setores que trabalhou** (código remanejamento, setor, data\_entrada, data\_saída, motivo\_saída);

**Remanejamento/atribuições gostaria** (código remanejamento, atribuições, horário);

**Remanejamento/setores-divisões-diretorias gostaria** (código remanejamento, setor);

**Remanejamento/motivos descontente** (código remanejamento, motivos, observações).

### ***Aplicação da 2ª Forma Normal - 2FN***

A 2FN diz que todos os atributos que não for chave têm que ser dependente direto da chave, ou seja, os atributos que contidos no requerimento, tem que depender diretamente da chave código requerimento.

Inicialmente observa-se a entidade que possui chave primária concatenada. Para aquelas que satisfazem esta condição, analisar se existe algum atributo ou conjunto de atributos com dependência direta à chave primária.

**Remanejamento** (código, código\_funcionário, categoria\_funcional, lotação\_atual, data\_admissão, cidade, data\_preenchimento, questão1, questão2, questão3, questão4, questão5, questão6, questão7, resultado, justificativa, data\_resultado);

- Verifique que o campo categoria funcional, lotação atual, cidade, devem permanecer na entidade Remanejamento, tendo em vista que se a caso deseja-se buscar um formulário emitido a 5 anos atrás cujo o funcionário alterou sua categoria funcional, lotação atual ou cidade, os dados que deverão aparecer no formulário preenchido a 5 anos atrás deve ser o correspondente a categoria funcional, lotação atual ou cidade, preenchido antigamente e não os dados novos.

**Funcionário** (código, nome, sexo, data\_nascimento);

**Remanejamento/escolaridade** (código\_remanejamento, código\_escolaridade, grau);

**Escolaridade** (código, descrição);

**Remanejamento/Cursos** (código\_remanejamento, código\_cursos);

**Cursos** (código, descrição);

**Remanejamento/experiências profissionais** (código\_remanejamento, código\_experiências);

**Experiências Profissionais** (código, descrição);

**Remanejamento/atribuições desenvolvidas** (código remanejamento, código atribuições);

**Atribuições** ( código, descrição);

**Remanejamento/setores que trabalhou** (código remanejamento, código setor,  
data\_entrada, data\_saída, motivo\_saída);

**Setores** ( código, descrição);

**Remanejamento/atribuições gostaria** (código remanejamento, código atribuições,  
horário);

**Remanejamento/setores-divisões-diretorias gostaria** (código remanejamento,  
código setor);

**Remanejamento/motivos descontente** (código remanejamento, código motivos,  
observações);

**Motivo descontente** (código, descrição);

### ***Aplicação da 3ª Forma Normal – 3FN***

A 3FN diz que o atributo precisa estar na segunda forma normal, e todos os atributos que não são chave, não podem depender de outros atributos que também não são chave.

Uma Entidade está na 3FN se nenhum de seus atributos possui dependência em relação a outros atributos da entidade. Ao verificar a existência de dependência entre as Entidades e seus atributos, devemos criar uma nova entidade que contenha os atributos dependentes.

**Remanejamento** (código, código\_funcionário, código\_categoria\_funcional, código\_lotação, data\_admissão, código\_cidade, data\_preenchimento, questão1, questão2, questão3, questão4, questão5, questão6, questão7, resultado, justificativa, data\_resultado);

**Categoria Funcional** (código, descrição);

**Lotação** (código, descrição);

**Cidade** (código, descrição);

**Funcionário** (código, nome, sexo, data\_nascimento);

**Remanejamento/escolaridade** (código\_remanejamento, código\_escolaridade, grau);

**Escolaridade** (código, descrição);

**Remanejamento/Cursos** (código\_remanejamento, código\_cursos);

**Cursos** (código, descrição);

**Remanejamento/experiências profissionais** (código\_remanejamento, código\_experiências);

**Experiências Profissionais** (código, descrição);

**Remanejamento/atribuições desenvolvidas** (código remanejamento, código atribuições);

**Atribuições** (código, descrição);

**Remanejamento/setores que trabalhou** (código remanejamento, código setor,  
data\_entrada, data\_saída, motivo\_saída);

**Setores** (código, descrição);

**Remanejamento/atribuições gostaria** (código remanejamento, código atribuições,  
horário);

**Remanejamento/setores-divisões-diretorias-gostaria** (código remanejamento,  
código setor);

**Remanejamento/motivos descontente** (código remanejamento, código motivos,  
observações);

**Motivo descontente** (código, descrição).



## Modelo Entidade – Relacionamento - MER

Abaixo, demonstra-se o MER do exemplo anterior.

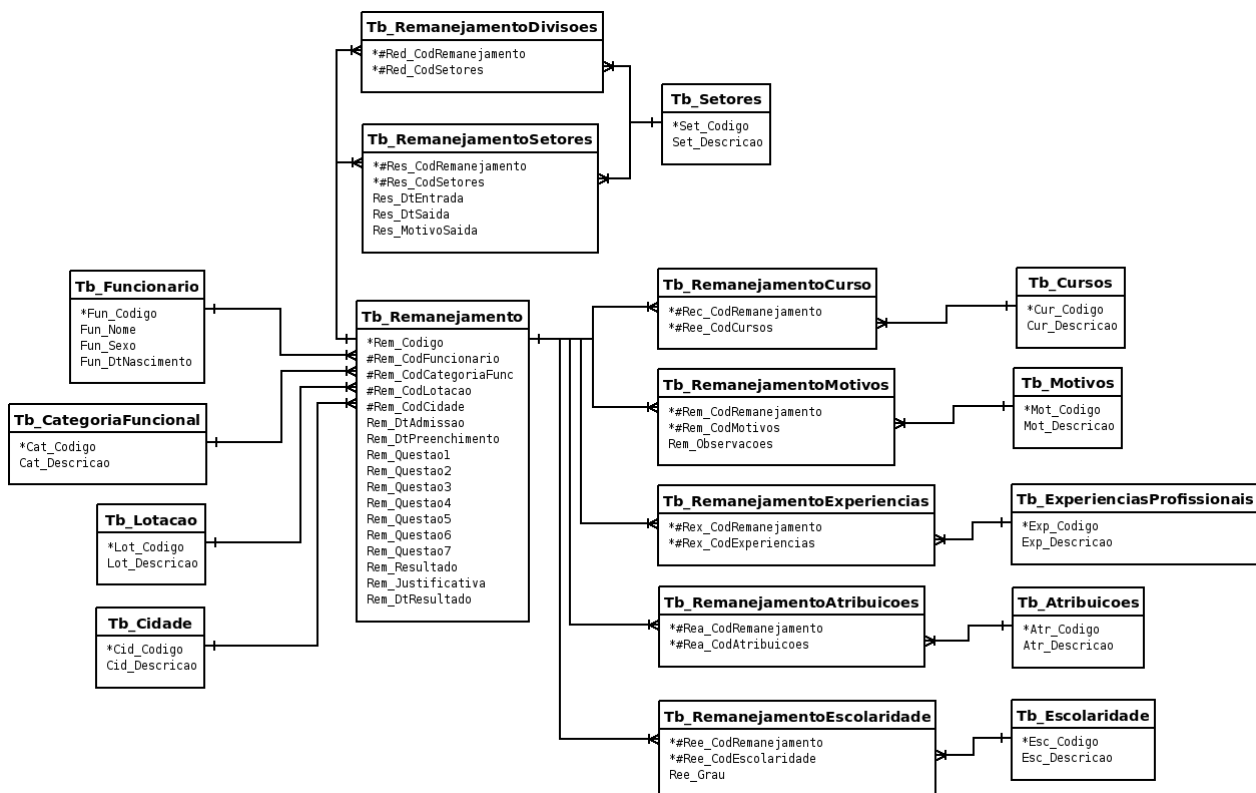


Figura 2 - Requerimento de Remanejamento