

Normalização de dados e as formas normais

Docente : Pedro F. Carvalho

2009

Email : contato@pedrofcarvalho.com.br

Docente : Pedro F. Carvalho
Analista de Sistemas

São José do Rio Preto – SP
Email : pedrofcarvalho@yahoo.com.br

Normalização de dados é o processo formal passo a passo que examina os atributos de uma entidade, com o objetivo de evitar anomalias observadas na inclusão, exclusão e alteração de registros.

Uma regra de ouro que devemos observar quando do projeto de um Banco de Dados baseado no Modelo Relacional de Dados é a de "não misturar assuntos em uma mesma Tabela". Por exemplo: na Tabela [Clientes](#) devemos colocar somente campos relacionados com o assunto Clientes. Não devemos misturar campos relacionados com outros assuntos, tais como Pedidos, Produtos, etc. Essa "Mistura de Assuntos" em uma mesma tabela acaba por gerar repetição desnecessária dos dados bem como inconsistência dos dados.

Normalmente após a aplicação das regras de **normalização de dados**, algumas tabelas acabam sendo divididas em duas ou mais tabelas, o que no final gera um número maior de tabelas do que o originalmente existente. Este processo causa a simplificação dos atributos de uma tabela, colaborando significativamente para a estabilidade do modelo de dados, reduzindo-se consideravelmente as necessidades de manutenção.

Objetivos

- Minimização de redundâncias e inconsistências;
- Facilidade de manipulações do banco de dados;
- Facilidade de manutenção do sistema de [Informação](#).

As formas normais

O Processo de normalização aplica uma série de regras sobre as tabelas de um banco de dados, para verificar se estas estão corretamente projetadas. Embora existam cinco **formas normais** (ou regras de normalização), na prática usamos um conjunto de três **Formas Normais**.

Vejamos as três primeiras formas normais do processo de **normalização de dados**.

- [Primeira Forma Normal \(1FN\)](#)
- [Segunda Forma Normal \(2FN\)](#)
- [Terceira Forma Normal \(3FN\)](#)

Apesar de existir outras **formas normais** como a quarta forma normal e quinta forma normal, apenas as três primeiras tem sido considerada atualmente.

As **formas normais** são [importantes](#) instrumentos para resolver antecipadamente problemas na estrutura do banco de dados.

Para aplicar a normalização de dados é necessário considerar a sequência das formas normais, isto é, para aplicar a segunda forma normal por exemplo, é necessário que seja aplicado a primeira

forma normal. Da mesma forma, para aplicar a terceira forma normal é necessário que já tenha sido feita a **normalização** na segunda forma normal.

Docente : Pedro F. Carvalho
Analista de Sistemas

São José do Rio Preto – SP
Email : pedrofcarvalho@yahoo.com.br

Primeira Forma Normal 1FN - Normalização de dados

Uma relação estará na **Primeira forma normal** 1FN, se e somente se todos os domínios básicos contiverem somente valores atômicos (não contiver grupos repetitivos).

Em outras palavras podemos definir que a **primeira forma normal** não admite repetições ou campos que tenha mais que um valor.

Procedimentos:

- a) Identificar a chave primária da entidade;
- b) Identificar o grupo repetitivo e removê-lo da entidade;
- c) Criar uma nova entidade com a chave primária da entidade anterior e o grupo repetitivo.

A chave primária da nova entidade será obtida pela concatenação da chave primária da entidade inicial e a do grupo repetitivo.

Exemplo de normalização de dados. Primeira forma normal

Considere a tabela cliente abaixo:

Cliente

Código_cliente

Nome

*Telefone

Endereço

Agora a tabela com os dados:

Código_cliente	Nome	Telefone	Endereço
C001	José	9563-6352 9847-2501	Rua Seis, 85 Morumbi 12536-965
C002	Maria	3265-8596	Rua Onze, 64 Moema 65985-963
C003	Janio	8545-8956 9598-6301	Praça ramos Liberdade 68858-633

Tabela desnormalizada, ou seja, não está na 1ª forma normal

Analisando teremos:

Docente : Pedro F. Carvalho
Analista de Sistemas

São José do Rio Preto – SP
Email : pedrofcarvalho@yahoo.com.br

Todos os clientes possuem Rua, CEP e Bairro, e essas informações estão na mesma célula da tabela, logo ela não está na **primeira forma normal**. Para normalizar, deveremos colocar cada informação em uma coluna diferente, como no exemplo a seguir:

Código_cliente	Nome	Telefone	Rua	Bairro	Cep
C001	José	9563-6352 9847-2501	Rua Seis, 85	Morumbi	12536-965
C002	Maria	3265-8596	Rua Onze, 64	Moema	65985-963
C003	Janio	8545-8956 9598-6301	Praça ramos	Liberdade	68858-633

Tabela ainda não está na primeira forma normal

Mesmo com o ajuste acima, a tabela ainda não está na **primeira forma normal**, pois há clientes com mais de um telefone e os valores estão em uma mesma célula. Para normalizar será necessário criar uma nova tabela para armazenar os números dos telefones e o campo-chave da tabela cliente. [Veja](#) o resultado a seguir:

Código_cliente	Nome	Rua	Bairro	Cep
C001	José	Rua Seis, 85	Morumbi	12536-965
C002	Maria	Rua Onze, 64	Moema	65985-963
C003	<u>Janio</u>	Praça ramos	Liberdade	68858-633

Tabela na primeira forma normal

Código_cliente	Telefone
C001	9563-6352
C001	9847-2501
C002	3265-8596
C003	8545-8956
C003	9598-6301

Tabela na 1ª forma normal

No exemplo acima foi gerado uma segunda entidade para que a **primeira forma normal** fosse satisfeita, contudo é possível manter a tabela [original](#), admitindo-se valores duplos em uma mesma coluna, como exemplo o campo telefone ficaria assim: 11-3400-3563 e 19-3500-9650.

Segunda Forma Normal 2FN - Normalização de dados

Uma tabela está na **Segunda Forma Normal 2FN** se ela estiver na 1FN e todos os atributos não chave forem totalmente dependentes da chave primária (dependente de toda a chave e não apenas de parte dela).

Se o nome do produto já existe na tabela produtos, então não é necessário que ele exista na tabela de vendas. A **segunda forma normal** trata destas anomalias e evita que valores fiquem em redundância no banco de dados.

Procedimentos:

- a) Identificar os atributos que não são funcionalmente dependentes de toda a chave primária;
- b) Remover da entidade todos esses atributos identificados e criar uma nova entidade com eles.

A chave primária da nova entidade será o atributo do qual os atributos do qual os atributos removidos são funcionalmente dependentes.

Exemplo de segunda forma normal

Considere a tabela vendas abaixo:

Vendas
N_pedido
Código_produto
Produto
Quant
Valor_unit
Subtotal

Agora a tabela com os dados:

N_pedido	Codigo_produto	Produto	Quant	Valor_unit	Subtotal
1005	1-934	Impressora laser	5	1.500,00	7.500,00
1006	1-956	Impressora desjet	3	350,00	1.050,00
1007	1-923	Impressora matricial	1	190,00	190,00
1008	1-908	Impressora mobile	6	980,00	5.880,00

Tabela não está na segunda forma normal

Analisando teremos:

Docente : Pedro F. Carvalho
Analista de Sistemas

São José do Rio Preto – SP
Email : pedrofcarvalho@yahoo.com.br

O nome do produto depende do código do produto, porém não depende de N_pedido que é a chave primária da tabela, portanto não está na **segunda forma normal**. Isto gera problemas com a manutenção dos dados, pois se houver alteração no nome do produto teremos que alterar em todos os registros da tabela venda.

Para normalizar esta tabela teremos de criar a tabela Produto que ficará com os atributos Código_produto e produto e na tabela Venda manteremos somente os atributos N_pedido, código_produto, quant, valor_unit e subtotal. [Veja](#) o resultado abaixo:

Codigo_produto	Produto
1-934	Impressora laser
1-956	Impressora desjet
1-923	Impressora matricial
1-908	Impressora mobile

Tabela na segunda forma normal

N_pedido	Codigo_produto	Quant	Valor_unit	Subtotal
1005	1-934	5	1.500,00	7.500,00
1006	1-956	3	350,00	1.050,00
1007	1-923	1	190,00	190,00
1008	1-908	6	980,00	5.880,00

Tabela na 2ª forma normal

Conforme visto na [Primeira forma normal](#), quando aplicamos normalização é comum gerar novas tabelas a fim de satisfazer as formas normais que estão sendo aplicadas.

Terceira Forma Normal 3FN - Normalização de dados

Uma tabela está na **Terceira Forma Normal** 3FN se ela estiver na 2FN e se nenhuma coluna não-chave depender de outra coluna não-chave.

Na **terceira forma normal** temos de eliminar aqueles campos que podem ser obtidos pela equação de outros campos da mesma tabela.

Procedimentos:

- a) Identificar todos os atributos que são funcionalmente dependentes de outros atributos não chave;
- b) Removê-los.

A chave primária da nova entidade será o atributo do qual os atributos removidos são funcionalmente dependentes.

Exemplo de normalização na terceira forma normal

Considere a tabela abaixo:

N_pedido	Codigo_produto	Quant	Valor_unit	Subtotal
1005	1-934	5	1.500,00	7.500,00
1006	1-956	3	350,00	1.050,00
1007	1-923	1	190,00	190,00
1008	1-908	6	980,00	5.880,00

Tabela não está na terceira forma normal

Considerando ainda a nossa tabela Venda, veremos que a mesma não está na **terceira forma normal**, pois o subtotal é o resultado da multiplicação Quant X Valor_unit, desta forma a coluna subtotal depende de outras colunas não-chave.

Para normalizar esta tabela na **terceira forma normal** teremos de eliminar a coluna subtotal, como no exemplo a seguir:

N_pedido	Codigo_produto	Quant	Valor_unit
1005	1-934	5	1.500,00
1006	1-956	3	350,00
1007	1-923	1	190,00
1008	1-908	6	980,00

Tabela na terceira forma normal