

**PROJETO LÓGICO DE BANCO DE DADOS - NORMALIZAÇÃO**



A redundância de dados é a causa de vários problemas em esquemas de Banco de Dados Relacionais. Pode trazer transtornos como: duplicidade de armazenamento de dados, anomalias de inserção, de exclusão e de atualização dos dados.

Veja o que aconteceu com um esquema que Tânia projetou para a clínica médica da sua prima:

Tânia criou uma entidade “Pacientes” com os seguintes atributos: ***Codigo***, ***Nome***, ***Endereco***, ***Telefone***, ***DataNascimento***, ***SiglaConvenio***, ***NomeConvenio***, ***EnderecoConvenio***, ***Telefone***\_***Convenio***.

Você percebeu que ela projetou os dados de pacientes e os de convênios na mesma tabela? Isso não poderia acontecer. Sabe por quê?

Porque os dados de um convênio (***nome***, ***endereço*** e ***telefone do convênio***) são repetidos para cada paciente associado a esse convênio.

Veja, os dados de um determinado convênio, o “Boa Saúde”, por exemplo, serão repetidos para cada um de seus associados.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Código Nome** | **Endereço** | **Telefone Data**  **de Nasc.** | **Sigla do Convênio** | **Nome do Convênio** | **Endereço Convênio** | **Telefone Convênio** |
| 100 Sonia de Lucca | Av. José Brejá, 700 | 3333-8899 10/08/1975 | BS | Boa Saúde | R.Arajá, 50 | 3322-5588 |
| 200 Laerte Souza | Av. Costa, 85 | 3333-5566 20/07/1983 | BS | Boa Saúde | Rua Arajá, 50 | 3322-5588 |
| 300 Maria José Pereira | R.Castro Alves, 200 | 3232-9966 30/04/1986 | BS | Boa Saúde | R. Arajá, 50 | 3322-5588 |
| 400 Dirce Maria Silva | R. Napoli, 400 | 3232-5580 17/06/1968 | APS | Ass. Paulista de Saúde | R. Amapá, 340 | 2745-3030 |
| 500 João Marques | R.15 de Novembro, 80 | 3456-9090 20/08/2004 | HSB | Hospital São Bento | R. Trelo, 40 | 3535-9060 |

Veja os **problemas que a redundância de dados** pode trazer:

# Anomalia de inserção:

- Quando se inserir um paciente é preciso inserir também os dados do convênio, mesmo que já estejam cadastrados.

- Não é possível inserir um convênio sem inserir também um paciente.

# Anomalia de exclusão:

- Ao se excluir um paciente, se este for o único associado de um convênio, então, os dados do convênio serão perdidos.

# Anomalia de modificação:

Quando for necessário modificar os dados de um convênio, é preciso atualizar os mesmos dados em todas os

registros de pacientes que estejam associados àquele convênio.



Pense, que sufoco!

Para que você não corra o risco de cometer os mesmos erros da Tânia, é importante estudar as regras de normalização. Vamos lá?

Texto adaptado de [**https://www.gsigma.ufsc.br/~popov/aulas/bd1/normalizacao\_old.html**.](https://www.gsigma.ufsc.br/~popov/aulas/bd1/normalizacao_old.html) Acessado em 21/09/2018.

**Normalização** é um processo utilizado, após o mapeamento do modelo conceitual, para acertar possíveis problemas estruturais das entidades e relacionamentos com campos criados – também chamados de anomalias – em um modelo de entidade e relacionamento.

A Normalização consiste na análise dos atributos das entidades e relacionamentos com campos, sob o ponto de vista das regras chamadas formas normais, que descrevem, com base na teoria de conjuntos, na álgebra e no cálculo relacional, o que devemos ou não fazer nas estruturas das entidades e relacionamentos de nosso modelo, baseados em conceitos matemáticos.

Essa análise pode demonstrar a necessidade de alterarmos a estrutura de nossas entidades e relacionamentos com campos, dividindo ou agrupando seus atributos para aprimorar o processo de recuperação das informações (performance) e seu armazenamento, de modo a evitar perda, redundância e distorção da informação.

Vamos estudar as formas normais que nos auxiliarão nesse trabalho:

**1ª Forma Normal**

Remove grupos de repetição. É a normalização do registro, de forma que o relacionamento entre a sua chave e seus atributos seja unívoca, isto é, para cada chave há a ocorrência de **um e somente um** dado de cada atributo. Portanto, na 1ª Forma Normal, tratamos os atributos multivalorados.

# Passos da 1ª Forma Normal

1. Identifique atributos que possuem valores para uma ocorrência da entidade.
2. Remova os atributos com uma cópia da chave primária.

# Exemplo:

Cliente (Número, Nome, {EnderecoEntrega})

**Não está na 1ª Forma Normal!**

|  |  |  |
| --- | --- | --- |
| **Número** | **Nome** | **Endereço-Entrega** |
| 124 | João dos Santos | Rua 9 de Julho, 56 |
| 256 | José Ferreira | Av. 15 de Novembro, 1980  Av. Campos Sales, 250 |
| 311 | André Alves | Av. São Carlos, 95 Rua Jorge Assef, 900 Rua Rui Barbosa, 935 |

Conversão para 1FN

Cliente (Número, Nome, End\_Entrega)

|  |  |  |
| --- | --- | --- |
| **Número** | **Nome** | **Endereço\_Entrega** |
| 124 | João dos Santos | Rua 9 de Julho, 56 |
| 124 | João dos Santos | Av. 15 de Novembro, 1980 |
| 256 | José Ferreira | Av. Campos Sales, 250 |
| 311 | André Alves | Av. São Carlos, 95 |
| 311 | André Alves | Rua Jorge Assef, 900 |
| 311 | André Alves | Rua Rui Barbosa, 935 |

**2ª Forma Normal**

É a normalização do registro de forma que, já submetido à 1FN, apresente uma **chave concatenada** que se relacione de forma integral com todos os seus atributos – Dependência Funcional Total.

A relação está na 2FN se está na 1FN e se não existir atributo não chave que é dependente de só uma parte de qualquer chave candidata.

# Passos da 2ª Forma Normal

1- Identifique atributos dependentes somente de parte da chave primária – Dependência Funcional Parcial. 2- Remova os atributos encontrados com uma cópia de parte da chave primária.

# Exemplo:

Pedido (Nr\_Ped, Data\_Pedido, Nr\_Peça, Descrição\_Peça, Quantidade\_Comprada, Preço\_Cotado)

**Não está na 2ª Forma Normal, pois:**

* A data do pedido depende apenas da chave do Número do Pedido (Nr\_Ped)
* A Descrição da peça depende apenas da chave Número da Peça (Nr\_Peça)

Portanto, esses atributos dependem de apenas uma parte da chave primária.

**Nr\_Ped => Data\_Pedido Nr\_Peça => Descrição\_Peça**

Só dependem de uma parte da chave primária.

**Anomalias de Atualização (Não está na 2ª Forma Normal)**

# Pedido

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr\_Ped** | **Data** | **Nr\_Peça** | **Descrição** | **Qtd\_Comprada** | **Preço\_Cotado** |
| 1000 | 14/11/2018 | AX12 | Bicicleta | 3 | 100,00 |
| 1020 | 15/11/2018 | BT04 | TV | 10 | 400,00 |
| 1030 | 15/11/2018 | BZ66 | Bola | 300 | 10,00 |
| 1040 | 16/11/2018 | BT04 | TV | 4 | 390,00 |
| 1050 | 17/11/2018 | CB03 | Video-Game | 5 | 380,00 |
| 1070 | 20/11/2018 | BT04 | TV | 2 | 410,00 |

Veja os problemas que essa tabela traz:

# Gasto de Espaço de Armazenamento e Anomalias

**Atualização**

* Mudar descrição de BT04 implica em várias mudanças (tempo gasto!)

# Dados Inconsistentes

* A Peça BT04 pode ter descrições diferentes.

# Adições

* Não será possível adicionar uma nova peça sem um pedido para ela porque o Número do Pedido (Nr\_Ped) é chave primária!

# Eliminações

* Se eliminar o Pedido 1000 perde-se a informação de que a peça AX12 é bicicleta.

# Para corrigir o Problema:

* Incluir nas relações os atributos correspondentes à Chave Primária apropriada (coleção mínima daqual dependem)
* Atribuir um nome para cada relação:

# 2ª Forma Normal

**Pedido (Nr\_Ped, Data) Peça (Nr\_Peça, Descrição)**

**Linha\_Pedido (Nr\_Pedido, Nr\_Peça, Quantidade\_Comprada, Preço\_Cotado)**

**3ª Forma Normal**

Uma relação está na 3FN se e somente se estiver na 2FN e todo atributo não chave não é transitivamente dependente de qualquer outro não chave: **Dependência Funcional Transitiva**.

**Passos para a 3ª Forma Normal:**

1. Identifique atributos dependentes de outros atributos não chave.
2. Remova esses atributos com uma cópia do atributo do qual depende.

# Exemplo:

Cliente (Nr\_Cli, Nome, Endereço, Nr\_Vendedor, Nome\_Vendedor)

# Não está na 3FN, pois”

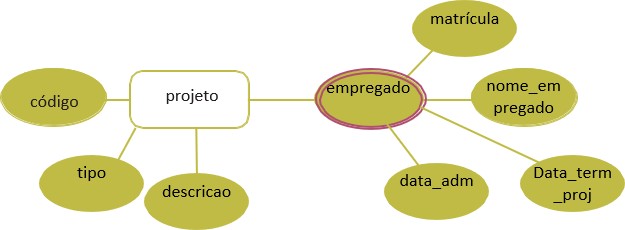
- “Nome do Vendedor” é transitivamente dependente de “Nr\_Vendedor” que por sua vez não é chave. Para corrigir o Problema:

# 3ª Forma Normal

**Cliente (Nr\_Cliente, Nome, Endereco) Vendedor (Nr\_Vendedor, Nome\_Vendedor)**



* 1. Analise o modelo abaixo e faça a sua normalização, de acordo com as regras apresentadas nessa agenda.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ProjCodigo** | **ProjTipo** | **ProjDescricao** |  | **Empregado** | |  |
| LSC001 | Novo Desenvolvimento | Sistema Contábil | **EmpMatrícula** | **EmpNome** | **EmpDataAdm** | **EmpDataTermProj** |
| 1215 | Edmundo | 30/05/2010 | 24/09/2018 |
| 3560 | Daniel | 18/09/2016 | 24/09/2018 |
| 5689 | Thiago | 02/09/2017 | 24/09/2018 |
| 2156 | Karla | 20/08/2015 | 24/09/2018 |
| PAG02 | Manutenção | Controle  de Estoque | 7856  4561 | Giovani  Marisa | 18/07/2016  08/03/2017 | 21/09/2018  21/09/2018 |

# Agora analise se essa tabela está na 2ª Forma Normal:

Alocação

|  |  |  |  |
| --- | --- | --- | --- |
| EmpMatricula | ProjCodigo | QtdeHorasTrab | ProjNome |

Para saber se você acertou, acompanhe essa explicação:

Para avançar para a 2FN e a 3FN é necessário conhecer o conceito de dependência funcional para identificar se o agrupamento de atributos de uma tabela é apropriado, evitando redundância de dados, inconsistências e perda de dados em operações de remoções ou alterações.

Existem 3 tipos de Dependência Funcional:

* Dependência Funcional Total
* Dependência Funcional Parcial
* Dependência Funcional Transitiva

# Dependência Funcional Total:

Os atributos não chave de uma tabela têm que depender da chave primária e somente dela. Por exemplo: Uma determinada tabela possui sua chave primária composta pelos atributos A e B. Logo, C será dependente funcional total se e somente se C depender funcionalmente de A e B.

Levando para o nosso exemplo:

Alocação

Nesse caso, a quantidade de horas trabalhadas refere-se às horas trabalhadas por um empregado em um determinado projeto, portanto o atributo “QtdeHorasTrab” depende tanto da chave “EmpMatricula” quanto da chave “ProjCodigo”.

|  |  |  |  |
| --- | --- | --- | --- |
| EmpMatricula | ProjCodigo | QtdeHorasTrab | ProjNome |

# Dependência Funcional Parcial:

Os atributos não chave de uma tabela dependem de parte da chave primária. Exemplo: Uma determinada tabela possui sua chave primária composta pelos atributos A e B. Logo, C será dependente funcional parcial se e somente se C depender funcionalmente de A ou B.

Levando essa definição para o nosso exemplo:

Alocação

Nesse caso, o nome do projeto está diretamente relacionado com o código do projeto e não tem nenhuma relação com o número da matrícula do funcionário, que é a outra chave da tabela. Portanto, sua dependência é parcial.

|  |  |  |  |
| --- | --- | --- | --- |
| EmpMatricula | ProjCodigo | QtdeHorasTrab | ProjNome |

A solução para a dependência funcional dessa tabela é separar em duas tabelas, conforme as dependências funcionais dos seus atributos:

Alocação Projeto

ProjNome

ProjCodigo

|  |  |  |
| --- | --- | --- |
| EmpMatricula | ProjCodigo | QtdeHorasTrab |

Agora, então, vamos analisar se a relação apresentada a seguir está na 3ª Forma Normal, verificando se há dependência funcional transitiva.

# Dependência Funcional Transitiva:

O atributo C é dependente funcional transitivo de A se C é funcionalmente dependente de B e B funcionalmente dependente de A, na mesma tabela.



Empregado

Nesse caso, o atributo “DataTerminoProj” é dependente funcional do “CodProjetoEmp” que é dependente funcional do atributo “EmpMatricula”.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EmpMatricula | EmpNome | EmpDataAdmissao | CodProjetoEmp | DataTerminoProj |

A solução para a dependência funcional dessa tabela é separar em duas tabelas, conforme as dependências funcionais dos seus atributos:

Empregado

|  |  |  |  |
| --- | --- | --- | --- |
| EmpMatricula | EmpNome | EmpDataAdmissao | CodProjetoEmp |

Projeto

DataTerminoProj

CodProjetoEmp

Enrendeu? Caso tenha dúvidas, contate o seu professor mediador!

Agora que você já aprendeu as regras da Normalização, vamos à atividade Online dessa semana?