

# Banco de Dados

**Centro Universitário Senac**

Prof. Msc. Fabio Versolatto  
[fabio.rversolatto@sp.senac.br](mailto:fabio.rversolatto@sp.senac.br)

# Onde estávamos

- Conceitos Fundamentais de Banco de Dados
- SGBD
- Projeto Lógico – MER
- Introdução ao SQL (implementação do Modelo Físico)
  - ESTRUTURA: CREATE, ALTER
  - CRUD (INSERT, SELECT, UPDATE, DELETE)
  - CLAUSULA WHERE 😊

# Pra onde vamos...

- Normalização

# Aviso Rápido...

- Base de Dados de Estudo disponibilizado no Midiateca
- Passo a passo
- Script SQL
- MySQL
- Objetivo: fazermos alguns hands-on semana que vem...

# Normalização

- Cascata dos problemas “comuns” em modelos de dados:
- Repetição de Informações
  - Dificuldade na leitura da informação
    - Performance
  - Dificuldade na atualização da informação
    - Informação não confiável
    - Necessidade de garantir (de alguma forma) que está atualizando a informação correta
    - Espaço em disco

# Normalização

- Pedra fundamental
  - Regras da boa modelagem
  - Validade, Completeza, Consistência;
- O controle de consistência pode ser exercido:
  - Pelo gerenciador;
  - Pelos aplicativos;
  - Pela própria construção do BD.

# Normalização

- Pela própria construção do sistema:
  - Controlar a construção do sistema através da criação de tabelas segundo regras que garantam a manutenção de certas propriedades.
  - As tabelas que atendem a um determinado conjunto de regras, diz-se estarem em uma determinada forma normal.

# Normalização

- Normalização é tratada com pouca atenção no momento do projeto.... É depois com o passar do tempo, verificamos que o modelo não era “bem aquele”...
- Um exemplo (pouco encontrado ?!):
  - encontrar muitas tabelas sem chave primária
  - com redundância de informações;
- Um outro exemplo (encontrado em bem menor escala):
  - Projetos cheios de anomalias de atualização
  - “Atualizar o preço de um produto em 28 tabelas”
  - Isso é comum ???
  - Isso é certo ???





# Normalização

- Objetivos da Normalização:
- Organizar e dividir as tabelas da forma mais eficiente possível, diminuindo a redundância e permitindo a evolução do BD com o mínimo de efeito colateral
- Garantir a integridade dos dados, evitando que informações sem sentido sejam inseridas
- Como funciona??
  - Implementação de regras chamadas formas normais
  - Processo passo a passo (formas)

# Normalização

- Conceitos envolvidos:
- Relações: podem ser interpretados como o equivalente matemático para “tabela”;
- Chaves: chaves candidatas, chaves primárias e chaves estrangeiras
- Dependência Funcional (DF)

# Normalização

- Dependência Funcional (DF)
- Sempre que o atributo X identifica um atributo Y (X é o determinante e Y é o dependente);
- Ex: Dada uma cidade (não considerando cidades homônimas) sabemos o seu estado e com o estado temos o país;
- Cidade  $\rightarrow$  estado (estado é funcionalmente dependente de cidade);
- Estado  $\rightarrow$  país (país é funcionalmente dependente de estado);  
ou cidade determina estado e estado determina país

# Normalização

- Dependência Funcional (DF)
- Se o valor de um conjunto de atributos A permite descobrir o valor de um outro conjunto B:
  - Dizemos que A determina funcionalmente B, ou que B depende de A, e representamos:  $A \rightarrow B$
- Se A for chave da relação e o valor de um subconjunto de atributos de A permite descobrir o valor de um outro conjunto B:
  - Dizemos que B possui dependência funcional em relação a A.

# Normalização

- Dependência Funcional (DF)
- Uma dependência funcional  $X \rightarrow Y$  é total:
  - Se removemos um atributo A qualquer do componente X e desta forma, a dependência funcional deixa de existir.
- A dependência funcional  $X \rightarrow Y$  é uma dependência funcional parcial:
  - Se existir um atributo A qualquer do componente X que pode ser removido e a dependência funcional  $X \rightarrow Y$  não deixa de existir

- Dependência Funcional (DF)

Exemplo:

RGEmpregado, NúmeroProjeto → Horas

- É uma dependência funcional total
- Se removermos o atributo RGEmpregado ou o atributo NúmeroProjeto, a dependência funcional deixa de existir

# Normalização

- Trivialidade: DF trivial indica que um determinante com mais de um atributo pode determinar seus próprios membros quando isolados

Ex: {banco, agência} → banco (DF trivial, pois banco é parte do determinante);

Ex: {banco, agência} → agência (DF trivial, pois agência é parte do determinante);

- Quando um determinante identifica outro atributo qualquer, temos uma DF não trivial (essa DF é a que nos interessa no processo de normalização):

Ex: {banco, agência} → cidade (DF não trivial, pois cidade não faz parte do determinante);

# Normalização

- DF Irredutível à esquerda
  - Dizemos que o lado esquerdo de uma DF é irredutível quando o determinante está em sua forma mínima;
  - Temos a forma mínima quando não é possível reduzir a quantidade de atributos determinantes sem perder a DF;

Ex: {cidade, estado}  $\rightarrow$  país  
(não está irredutível, pois podemos ter somente estado);

Ex: estado  $\rightarrow$  país

- Obs.: nem sempre estar na forma irredutível à esquerda significa possuir um determinante com apenas uma coluna.



# Normalização

- O processo de normalização envolve a quebra ou decomposição de uma tabela em partes menores;
- Essa decomposição tem que ser reversível, de modo que nenhuma informação seja perdida no processo;
- O fato de uma decomposição ser ou não sem perda está intimamente ligado ao conceito de DF;
- Uma boa forma de começar é identificar anomalias nos comandos DELETE, INSERT e UPDATE...

# Normalização

- ***Tabela Original***

Funcionário	Sexo	Salario
Carlos	M	5.000,00
Marcos	M	4.500,00

- ***Decomposição A***

Funcionário	Sexo
Carlos	M
Marcos	M

Funcionário	Salario
Carlos	5.000,00
Marcos	4.500,00

- ***Decomposição B***

Funcionário	Sexo
Carlos	M
Marcos	M

Sexo	Salario
M	5.000,00
M	4.500,00

# Normalização

- Na decomposição a letra a é possível restaurar a tabela original, pois com o nome do funcionário obtemos o sexo e o salário correspondentes (decomposição sem perdas);
- Na letra b não é possível obter a tabela original, pois para o funcionário “Carlos” o sexo “M”, encontraremos dois salários na segunda tabela.

# Normalização

- Importância da Normalização
- Tabelas flexíveis e de fácil manutenção: uma base sem as anomalias de atualização e com uma estrutura de armazenamento eficaz é mais simples de atualizar e evoluir (embora o processo de consulta se torne mais complexo, pois a normalização geralmente aumenta o número de tabelas);
- Eliminação de redundâncias: Sem redundâncias, as tabelas ficam menores, mas consistentes e menos sujeitas a discrepâncias. Evite o que o mercado chama de “redundância controlada”

# Normalização

- O processo de Normalização
- Vamos trabalhar com o seguinte exemplo: em uma determinada empresa, os produtos recebidos de um fornecedor são registrados em um formulário próprio.
- Vamos informatizar esse processo criando uma base de dados para armazenar as informações deste formulário.

# Normalização

TB_Pedido
«column»
NroNota
Data
CodForn
Nome
Telefone
Endereço
CodItem
CodProd
Preço
Quantidade
TotalItem

- Primeira Forma Normal (1FN)
- Uma entidade está na 1FN quando cada atributo contém somente um valor, em somente um lugar;
- Essa exigência também é conhecida como atomicidade de dados;
- As regras gerais para obtenção da 1FN são:
  - Não podemos ter atributos multivalorados. Todos os registros têm que ser diferentes;
  - Nesse caso, colocamos cada valor do atributo em uma linha diferente e repetimos os dados de todas as outras colunas;

- Primeira Forma Normal (1FN)
- Não podemos ter atributos repetidos, como Tel1, Tel2, etc. A solução é semelhante ao item anterior;
- A entidade não pode ter mais de duas dimensões;
- Cada atributo deve ter somente um tipo de dado.
- Uma violação comum dessa regra, por exemplo, é a criação de um campo para armazenar o CPF e o CNPJ, alternadamente.
- Esse cenário deve ser evitado pois cria complicações para a evolução da regra de negócio.



# Normalização

- Primeira Forma Normal (1FN)
- Para trabalhar com a 1FN precisamos definir uma chave primária;
- A chave escolhida será: NroNota e CodItem, pois através deles determinamos todos os outros campos;

NroNota  $\rightarrow$  {Data, CodForn, Nome, Telefone, Endereço}

{NroNota, CodItem}  $\rightarrow$  {CodProd, Produto, Quantidade, Preço, TotalItem}

# Normalização



- Segunda Forma Normal (2FN)
- Essa forma normal visa a diminuição da redundância e o desagrupamento de informações;
- Com a 2FN, uma tabela passa a representar uma quantidade menor de entidades (o ideal é que cada entidade seja armazenada em apenas uma tabela);
- Obs.: no exemplo a tabela está agrupando entidades como Nota Fiscal, Item da Nota, Fornecedor e Produto.

- Segunda Forma Normal (2FN)
- Uma tabela está na 2FN se estiver em 1FN e todo atributo não-chave for determinado por todos os campos da chave primária;
- Ou seja, é necessário eliminar as DF parciais;
  - Obs.: a tabela exemplo viola a 2FN pois os campos Data, CodForm, Nome, Telefone e Endereço não são determinados pela chave primária completa (o campo
  - CodItem não é necessário para identificar essas informações):
  - NroNota  $\rightarrow$  {Data, CodForn, Nome, Telefone, Endereço}

# Normalização

## Tb\_Pedido



«column»

**\*PK** NroNota  
Data  
CodForn  
Nome  
Endereço  
Telefone

## Tb\_Item



«column»

**\*PK** CodItem  
CodProd  
Preço  
Quantidade  
TotalItem

# Normalização

- Segunda Forma Normal (2FN)
- A 2FN deve ser aplicada através dos passos:
  - Eleger a chave primária da tabela;
  - Verificar as DF parciais;
  - Mover os campos não enquadrados na 2FN para uma nova tabela, fazendo a decomposição sem perdas;
  - Na tabela criada, repetir os passos anteriores até eliminar a DF parcial.

# Normalização

Tb_Pedido
«column»
*PK NroNota
Data
CodForn
Nome
Endereço
Telefone

Tb_Item
«column»
*PK CodItem
CodProd
Preço
Quantidade
TotatlItem
* NroNota

- Terceira Forma Normal (3FN)
- A 3FN dá continuidade ao objetivo da 2FN: reduzir as redundâncias, desagrupando as tabelas de forma que cada uma represente apenas uma entidade;
- A técnica utilizada pela 3FN é a identificação e eliminação da transitividade;
- Uma tabela está na 3FN se estiver na 2FN e todo atributo não chave for determinado de forma não transitiva pela chave primária.
- Ou seja, todo atributo não chave deve ser determinado somente pela chave primária.



- Terceira Forma Normal (3FN)
- Analisando a tabela criada pela 2FN:
  - Os campos Nome, Telefone e Endereço podem ser determinados tanto pela chave primária quanto pelo campo CodForm:

$\text{NroNota} \rightarrow \{\text{Data}, \text{CodForn}, \text{Nome}, \text{Telefone}, \text{Endereço}\}$

$\text{CodForn} \rightarrow \{\text{Nome}, \text{Telefone}, \text{Endereço}\}$

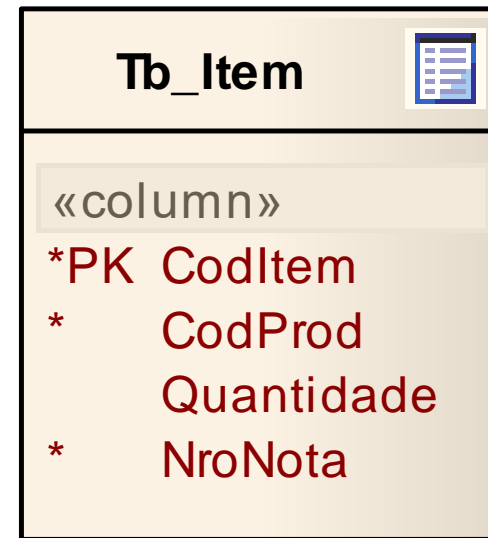
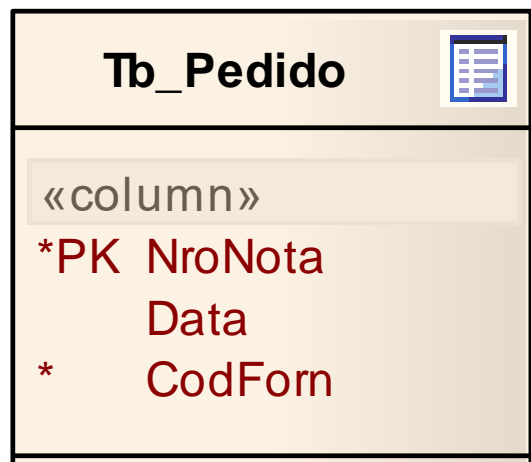
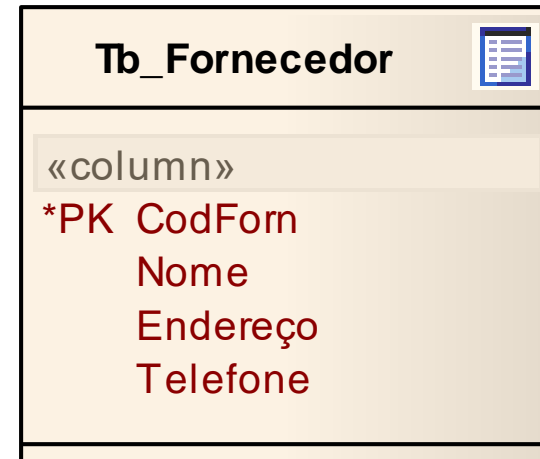
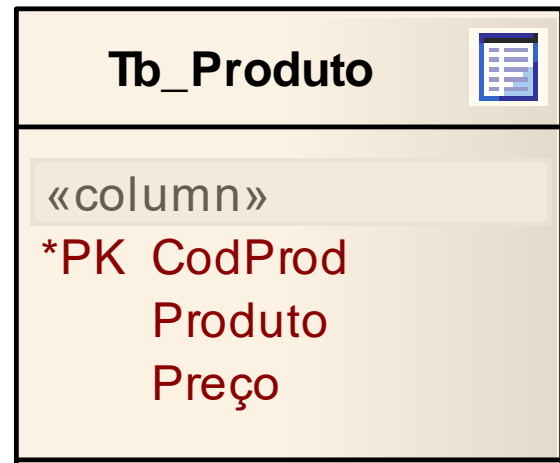
- Esses campos possuem DF transitiva com a PK:

$\{\text{NroNota}\} \rightarrow \text{CodForn} \rightarrow \{\text{Nome}, \text{Telefone}, \text{Endereço}\}$

- Também temos transitividade na segunda tabela:

$\{\text{NroNota}, \text{CodItem}\} \rightarrow \text{CodProd} \rightarrow \{\text{Produto}, \text{Preço}\}$

# Normalização



# Normalização

- Terceira Forma Normal (3FN)
- Outro tipo de violação são os campos calculados, que também possuem transitividade;
- Na 3FN, todos os campos calculados são removidos da base de dados;
- Veja representação:

$\{\text{NroNota}, \text{CodItem}\} \rightarrow \text{Preço}, \text{Quantidade} \rightarrow \{\text{TotalItem}\}$

# Normalização

- Terceira Forma Normal (3FN)
- Para adequar as tabelas à 3FN devemos:
  - Mover os campos com transitividade para uma nova tabela;
  - Criar uma PK na tabela nova com o(s) campo(s) da tabela original que determinava(m) diretamente os campos movidos;
  - Na nova tabela, repetir os passos anteriores até eliminar totalmente a transitividade;

- Terceira Forma Normal (3FN)
- Neste ponto temos a organização ideal para a base de dados, pelos motivos a seguir:
- A decomposição foi feita sem perdas;
- As 4 entidades possuem tabelas exclusivas, eliminando o agrupamento de informações e a redundância.
- As tabelas foram separadas de tal forma que as anomalias de atualização não poderão ocorrer;
- As tabelas são fáceis de evoluir e manter.
- Do ponto de vista relacional, os dados serão armazenados e distribuídos de forma eficiente.

# Normalização



Calma !!!  
Temos até a  
5ª Forma 😊

Mas por enquanto  
é isso 😊