

**MODELAGEM DE DADOS**  
**Prof. Milton Palmeira Santana**



## EXERCÍCIO

- » Elaborar um DER para uma seguradora de automóveis  
Entidades: **Cliente**, **Apólice**, **Carro** e **Acidentes**.

- » **REQUISITOS:**

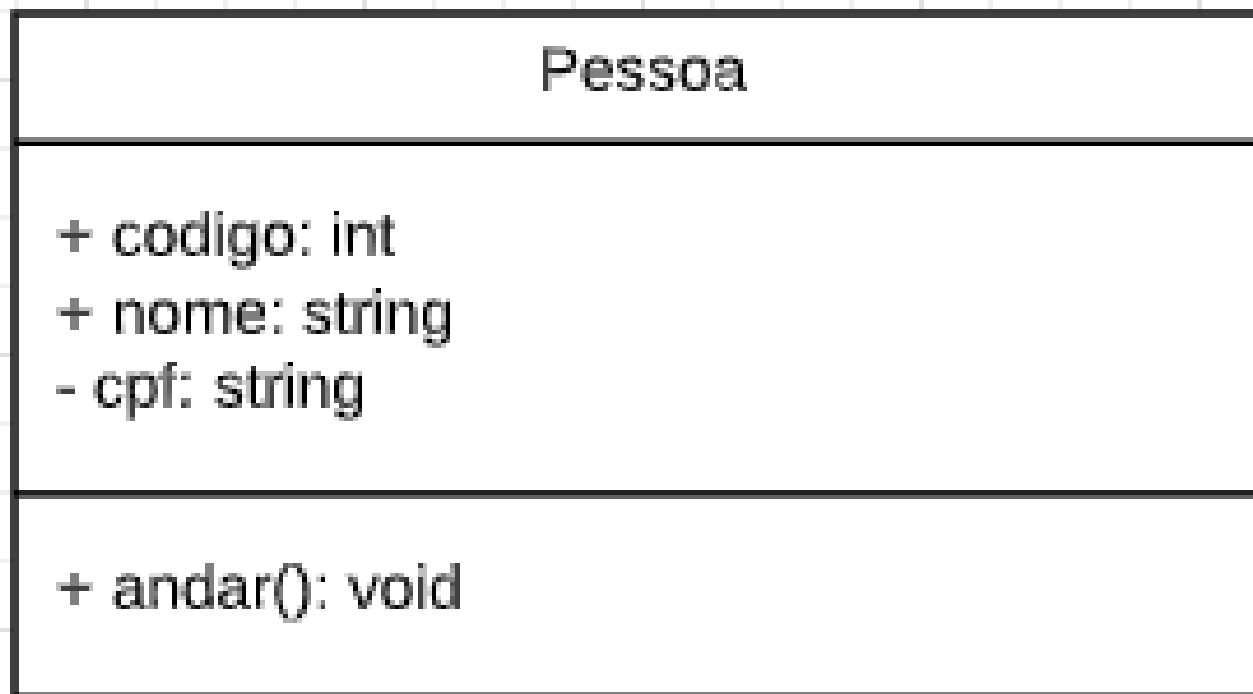
- Um cliente pode ter várias apólices (no mínimo uma);
- Cada apólice somente dá cobertura a um carro;
- Um carro pode ter zero ou n registros de acidentes a ele.

- » **ATRIBUTOS:**

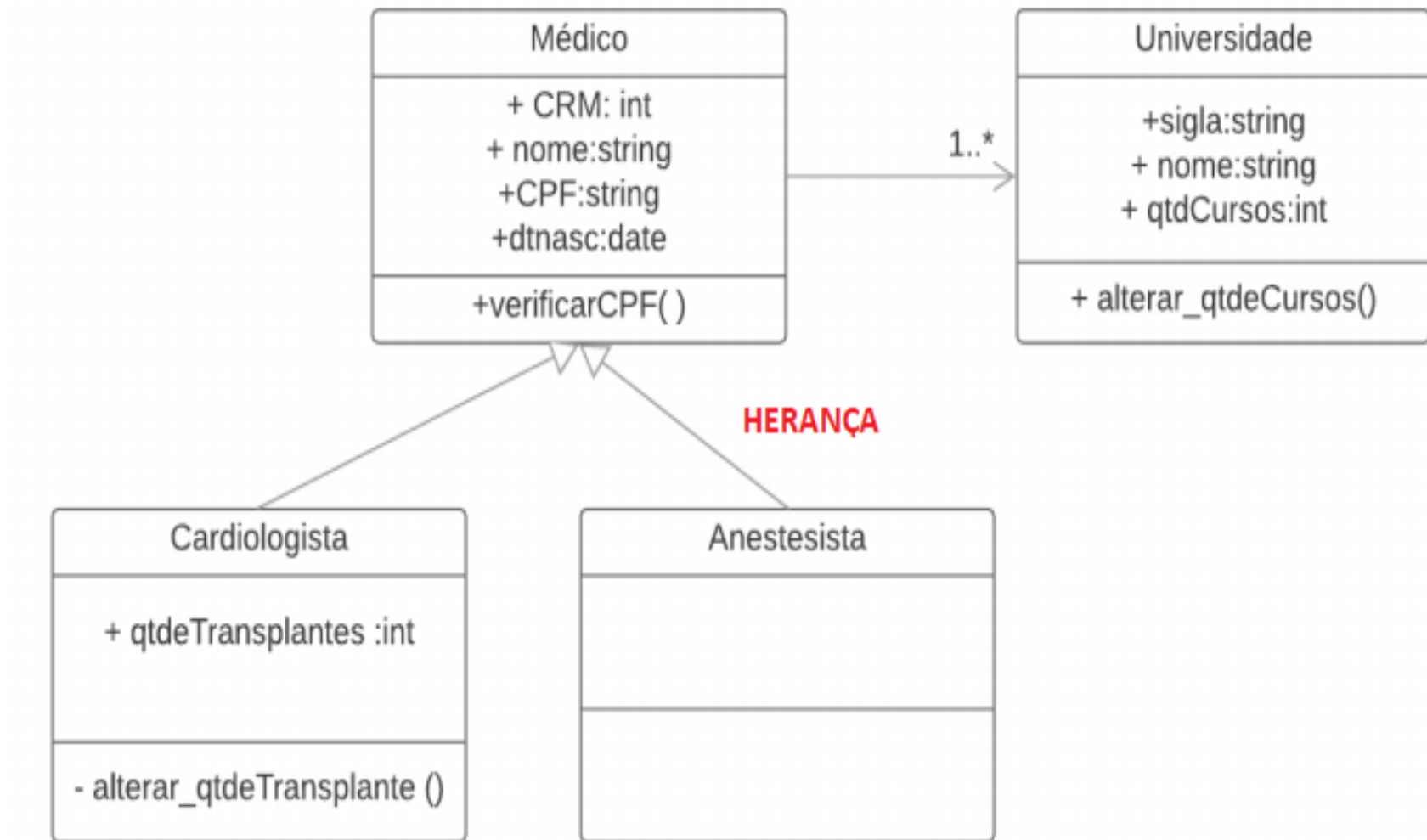
- **Cliente:** Número, Nome e Endereço;
- **Apólice:** Número e Valor;
- **Carro:** Registro e Marca;
- **Acidente:** Data, Hora e Local;

## DIAGRAMA DE CLASSES

» O Diagrama de classes é o mais utilizado.



## DIAGRAMA DE CLASSES



## EXERCÍCIOS

1. A Linguagem de Modelagem Unificada ou UML (**Unified Modeling Language**) é uma ferramenta que auxilia na modelagem de sistemas orientados a objetos.

Marque a afirmativa correta sobre a UML:

- a) Serve para a programação somente de banco de dados.
- b) É utilizada como um SGBD pelo fato de permitir a orientação a objetos.
- c) Permite padronizar projetos de sistemas orientados a objetos e de banco de dados.
- d) É uma linguagem de programação orientada a objetos reutilizável.
- e) É utilizada somente para a criação do modelo lógico de banco de dados de um SGBD.

## EXERCÍCIOS

**2.** Os atributos ou campos são características de determinadas classes ou entidades. Eles devem ter nomes apropriados ao sentido do que vão armazenar e um tipo de dados que deverá ser declarado, podendo ser: inteiro, caractere, decimal, entre outros.

Assinale a alternativa correta referente à diferença entre classes e entidades.

**a)** Uma classe é somente um conjunto de campos, exatamente igual as entidades dos modelos conceituais de banco de dados.

**b)** Uma das vantagens da classe é possuir métodos que podem manipular os seus atributos, já as entidades possuem o objetivo de armazenar as informações em seus atributos.

## EXERCÍCIOS

- c)** As classes e as entidades possuem métodos que permitem, de forma moderada, a alteração de seus atributos.
- d)** Uma entidade possui a vantagem de ter métodos que podem ser utilizados para modificar os atributos da própria tabela, algo que em classe ainda não é permitido.
- e)** Uma vez criados os atributos em classes ou em entidades, não é possível alterar o conteúdo armazenado, isto é, assim que o atributo for criado, recebe um valor e não pode mais ser alterado

## EXERCÍCIOS

**1. C**

**2. B**



## NORMALIZAÇÃO

- » A modelagem de dados tem como sua essência mais pura o refinamento de processos. Isso significa que sempre podemos melhorar o que já foi realizado. Uma vez criado o Diagrama Entidade-Relacionamento (DER) já podemos implementá-lo em um Sistema Gerenciador de Banco de Dados? Não. O correto é revisar, procurar imperfeições e melhorar, prevendo possíveis problemas.
- » Em um banco de dados o maior problema é a redundância, pois ela pode causar danos enormes e pode acontecer disso ser notado somente quando o banco de dados já estiver sendo usado pela empresa. Os danos que a redundância pode causar e que geram mais problemas é a repetição da mesma informação em várias tabelas, ocasionando, além de duplicidade, possíveis erros em relatórios.

## NORMALIZAÇÃO

» Qual o problema com o exemplo abaixo?

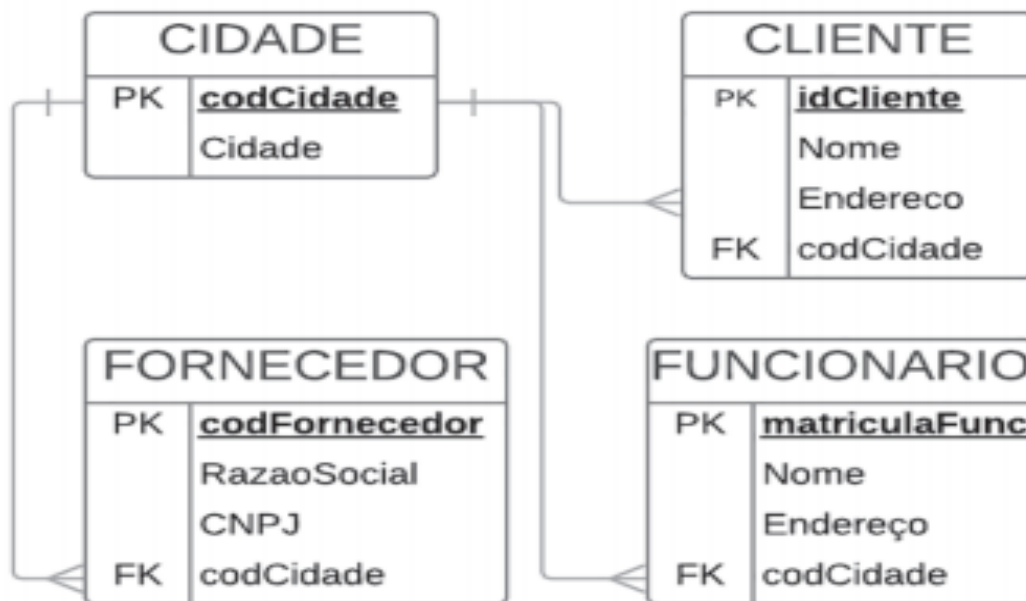
CLIENTE	
PK	<u>idCliente</u>
	Nome Endereco CidadeResid

FORNECEDOR	
PK	<u>codFornecedor</u>
	RazaoSocial CNPJ Cidade

FUNCIONÁRIO	
PK	<u>matriculaFunc</u>
	Nome CPF CidadeNasc

## NORMALIZAÇÃO

- » O problema não é só a redundância, mas e se o usuário cadastrar a informação de maneira diferente em cada tabela?  
Ex: Taubaté, Taubate, tau.
- » Nesse caso, como poderíamos evitar que o usuário cadastrasse a informação várias vezes em várias tabelas diferentes?



## NORMALIZAÇÃO

- » Às vezes é interessante manter uma informação redundante no banco de dados. Por questões de desempenho de alguma pesquisa ou software a redundância pode ocorrer, fato esse que é chamado de redundância controlada e é recomendado quando o campo recebe uma grande quantidade de consultas e poucas alterações. Um exemplo é uma tabela de nota fiscal, o campo “valor\_total\_da\_nota”, que poderia ser obtido automaticamente, como o resultado da soma de todos os itens vendidos e multiplicados pelo seu preço.
- » Entretanto, é comum criarmos manualmente o campo “valor\_total\_da\_nota” para evitar que, caso haja alteração do preço do produto vendido, o valor da nota não seja alterado (causando problemas contábeis). É uma redundância que sabemos que existe, mas é necessária.

## NORMALIZAÇÃO

» Outro exemplo de redundância:

Tabela: funcionário				
Matricula	Nome	Valor_Hora	CodDepartamento	NomeDepartamento
13467-4	Marco Antonio Liz	R\$ 18,22	DP - 450	Expedição
34562-5	Anna Pietro	R\$ 18,22	DP - 450	Expedição
76321-0	Carlos Werner	R\$ 13,50	DP - 450	Expedição
58309-3	Sandro Lopez	R\$ 28,70	DA - 780	Contabilidade



REDUNDÂNCIA

## NORMALIZAÇÃO

- » A normalização é o ato de aplicar regras para todas as tabelas com o objetivo de reduzir a redundância e eliminar campos que não dizem respeito a determinada tabela.
- » Um bom exemplo disso é uma tabela que armazena informações sobre a cidade. De forma alguma poderemos armazenar informações sobre um funcionário dentro da tabela.

## NORMALIZAÇÃO

- » Podemos listar alguns objetivos e vantagens da normalização de um esquema de banco de dados:
  - Diminuição de dados repetitivos deixando o banco de dados mais compacto.
  - Aumento da performance no Sistema Gerenciador de Banco de Dados.
  - Armazenamento dos dados de forma lógica.
  - Facilidade na criação de consultas.
  - Permite a concatenação de índices (chaves) de acordo com a quantidade de tabelas envolvidas.
  - Facilidade na manutenção do banco de dados.

## NORMALIZAÇÃO

» Tabela não normalizada.

Tabela: produto					
idProd	Produto	Preço	TipoProduto	CodForn	Fornecedor
1415	Sabão	R\$ 4,71	Limpeza	708	Tem Tudo
7841	Álcool	R\$ 5,80	Limpezas	708	Tem de Tudo
8543	Arroz	R\$ 7,84	Grão	516	Compra Boa
9124	Trigo	R\$ 5,45	Grãos	516	Compra B.



## NORMALIZAÇÃO

- » Uma das regras básicas da normalização é verificar se determinado campo realmente pertence à tabela.
  
- » No exemplo anterior, temos:
  - **TipoProduto**: podemos criar uma tabela para armazenar os tipos de produtos.
  
  - **CodForn** e **Fornecedor**: devemos criar uma tabela chamada fornecedor.

## NORMALIZAÇÃO

» Tabela normalizada.

Tabela: produto				
idProd	Produto	Preço	idTipoProduto	CodFornecedor
1415	Sabão	R\$ 4,71	23	708
7841	Álcool	R\$ 5,80	23	708
8543	Arroz	R\$ 7,84	18	516
9124	Trigo	R\$ 5,45	18	516

Tabela: TipoProduto	
idTipoProduto	TipoProduto
23	Limpeza
18	Grãos

Tabela: fornecedor	
codFornecedor	Fornecedor
708	Tem Tudo
516	Compra Boa

## NORMALIZAÇÃO

- » As primeiras técnicas de normalização foram criadas em 1972 por Edgar Frank Codd.
  
- » As formas normais mais populares são:
  - A primeira forma normal ou 1FN.
  
  - A segunda forma normal ou 2FN.
  
  - A terceira forma normal ou 3FN.
  
  - A quarta forma normal ou 4FN.

## NORMALIZAÇÃO

- » A forma normal é uma regra que deve ser satisfeita por uma entidade para que ela seja avaliada como uma tabela “projetada com exatidão”. São várias formas normais, com regras que vão se tornando mais rigorosas, com o objetivo de averiguar nas tabelas a existência de redundância ou dependências funcionais. No entanto, pelo menos quatro formas normais são consideradas essenciais para a construção de um bom projeto de banco de dados.

## NORMALIZAÇÃO 1FN e 2FN

- » Para atingir os objetivos da normalização, as tabelas precisam ter as seguintes propriedades:
  - Cada tabela deverá tratar de somente um único assunto, por exemplo: uma tabela com informações sobre remédio, não poderá ter informações de um médico.
  - O mesmo campo não poderá ser armazenado, desnecessariamente, em mais de uma tabela. Essa é uma garantia de que não será necessária a atualização do mesmo campo, em mais de uma tabela.

## NORMALIZAÇÃO 1FN e 2FN

- Os campos de uma tabela são dependentes da chave primária dessa tabela e de mais nenhum campo.
- A tabela deverá estar livre de anomalias de inserção, atualização e exclusão, garantindo a integridade e a consistência dos dados, por exemplo: na tabela Cliente será necessário informar a cidade de seu nascimento, não devemos deixar ele informar a cidade, mas escolher a cidade dentre as cidades previamente cadastradas ou por meio de uma busca do CEP (que trará o endereço completo).

## NORMALIZAÇÃO 1FN e 2FN

- » Para aplicar as regras da normalização, um dos alvos a ser observado são os campos (ou atributos) que fazem parte das tabelas. Podemos classificar os atributos como:
  - **Atributo simples ou atômico:** é o atributo que não é divisível, possui um sentido único, como o RG ou o CPF de uma pessoa, esses dois exemplos mostram que tanto o RG quanto o CPF não podem ser divididos em dois outros campos.
  - **Atributo composto:** é um atributo que pode ser dividido em várias partes, um bom exemplo é o endereço. Podemos dividir esse atributo em: rua, número, complemento, bairro.

## NORMALIZAÇÃO 1FN e 2FN

- **Atributo monovalorado:** é um atributo que possui apenas um valor para a tabela, como a matrícula de um aluno, esse número não poderá se repetir na tabela.
- **Atributo multivalorado:** é um atributo que pode receber mais de uma informação, o melhor exemplo é o telefone que pode receber mais de um valor.
- **Atributo derivado:** o valor desse tipo de atributo pode originar de outra tabela ou de outros campos. Digamos que para um cardiologista seja necessário saber a idade (em anos e dias). Podemos calculá-la a partir da data de nascimento e da data de atendimento no ato da consulta médica.
- **Atributo chave:** é o atributo escolhido ou criado para que possa indicar o registro (a linha) da tabela.



BARBOZA, Fabrício Felipe Meleto; FREITAS, Pedro Henrique Chagas. **Modelagem e desenvolvimento de banco de dados**. Porto Alegre: SAGAH, 2018.

WERLICH, Claudia. **Modelagem de dados**. Londrina: Editora e Distribuidora Educacional S.A, 2018.

MANZANO, Jose Augusto Navarro Garcia. **Microsoft SQL Server 2016 Express Edition Interativo**. [S. l.]: ÉRICA, 2016.

MACHADO, Felipe Nery Rodrigues; ABREU, Mauricio Pereira de. **Projeto de Banco de Dados: Uma Visão Prática** - Edição Revisada e Ampliada. [S. l.]: ÉRICA, 2009.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Sistemas de Gerenciamento de Bancos de Dados. [S. l.]: Amgh Editora, 2011.

ALVES, WILLIAM PEREIRA. Banco de Dados. São Paulo: Saraiva, 2014

CARDOSO, VIRGÍNIA M.; CARDOSO, GISELLE CRISTINA. SISTEMA DE BANCO DE DADOS. São Paulo: Saraiva, 2013



Anhanguera