

## **Aula 07**

*Banco do Brasil (Escriturário - Agente de  
Tecnologia) Passo Estratégico de  
Tecnologia de Informação - 2023  
(Pós-Edital)*

Autor:

**Thiago Rodrigues Cavalcanti**

10 de Fevereiro de 2023

# MODELAGEM CONCEITUAL DE DADOS (A ABORDAGEM ENTIDADE-RELACIONAMENTO); MODELO RELACIONAL DE DADOS (CONCEITOS BÁSICOS, NORMALIZAÇÃO)

## Sumário

Análise Estatística.....	2
Roteiro de revisão e pontos do assunto que merecem destaque .....	3
Banco de dados: conceitos básicos .....	3
Banco de dados: Modelo de dados.....	6
Banco de dados relacionais: conceitos básicos e características .....	7
Metadados .....	11
Tabelas, visões (views) e índices.....	11
Chaves e relacionamentos.....	17
Aposta estratégica.....	20
Banco de dados relacionais: conceitos básicos e características .....	21
Metadados .....	21
Tabelas (Restrições de integridade).....	22
Chaves e relacionamentos.....	24
Questões estratégicas .....	25
Questionário de revisão e aperfeiçoamento.....	26
Perguntas.....	26
Perguntas com respostas .....	27



## ANÁLISE ESTATÍSTICA

Inicialmente, convém destacar os percentuais de incidência de todos os assuntos previstos no nosso curso – quanto maior o percentual de cobrança de um dado assunto, maior sua importância:

Assunto	Quantidade	Grau de incidência em concursos similares
CESGRANRIO		
Modelagem conceitual de dados (a abordagem entidade-relacionamento); Modelo relacional de dados (conceitos básicos, normalização);	55	23,40%
Linguagem SQL2008;	49	20,85%
5. Estrutura de dados e algoritmos: Busca sequencial e busca binária sobre arrays; Ordenação (métodos da bolha, ordenação por seleção, ordenação por inserção), lista encadeada, pilha, fila e noções sobre árvore binária.	42	17,87%
6. Ferramentas e Linguagens de Programação para manipulação de dados: Ansible; Java (SE 11 e EE 8); TypeScript 4.0;	34	14,47%
Data Warehouse (modelagem conceitual para data warehouses, dados multidimensionais);	31	13,19%
Python 3.9.X aplicada para IA/ML e Analytics (bibliotecas Pandas, NumPy, SciPy, Matplotlib e Scikit-learn).	11	4,68%
2. Banco de Dados: Banco de dados NoSQL (conceitos básicos, bancos orientados a grafos, colunas, chave/valor e documentos);	4	1,70%
4. Desenvolvimento Mobile: linguagens/frameworks: Java/Kotlin e Swift. React Native 0.59; Sistemas Android api 30 e iOS xCode 10.	4	1,70%
3. Big data: Fundamentos; Técnicas de preparação e apresentação de dados.	3	1,28%
1. Aprendizagem de máquina: Fundamentos básicos; Noções de algoritmos de aprendizado supervisionados e não supervisionados	1	0,43%
Postgre-SQL;	1	0,43%
Noções de processamento de linguagem natural.	0	0,00%
Conceitos de banco de dados e sistemas gerenciadores de bancos de dados (SGBD); MongoDB;	0	0,00%



## ROTEIRO DE REVISÃO E PONTOS DO ASSUNTO QUE MERECEM DESTAQUE

*A ideia desta seção é apresentar um roteiro para que você realize uma revisão completa do assunto e, ao mesmo tempo, destacar aspectos do conteúdo que merecem atenção.*

### AVISO IMPORTANTE!

Caso você esteja estranhando o título dessa aula, preciso esclarecer que tivemos que fazer uma alteração em nosso cronograma com o intuito de redistribuir os conteúdos que faltam ser abordados. Em breve a atualização do cronograma estará disponível.

Para revisar e ficar bem preparado no assunto, você precisa, basicamente, seguir os passos a seguir:

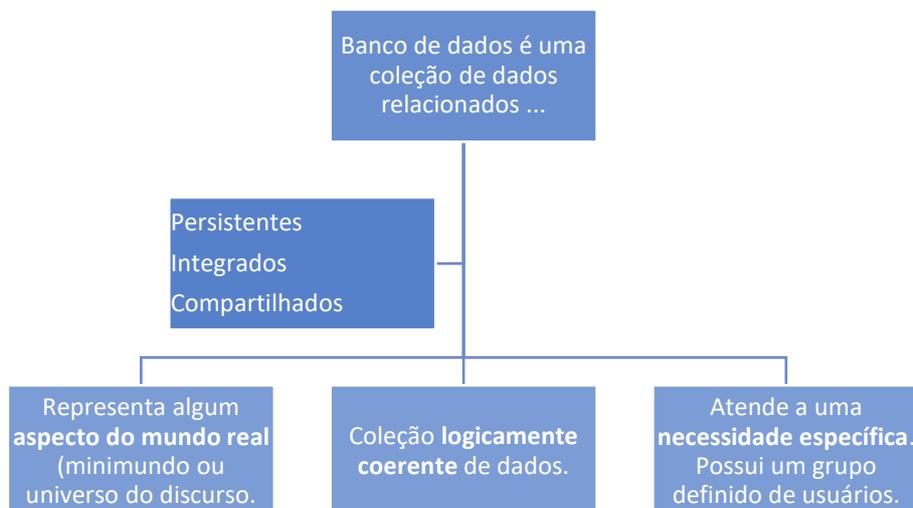
### Banco de dados: conceitos básicos

Existem alguns conceitos importantes que subsidiam o nosso estudo sobre o assunto: **banco de dados (BD)**, **sistemas de gerenciamento de banco de dados (SGBD)** e **sistemas de banco de dados (SBD)**. Vejamos uma definição rápida de cada um deles.

Um banco de dados é uma **coleção de dados persistentes, integrados e compartilhados**, usada pelos sistemas de aplicação de uma determinada empresa. Por **persistente**, queremos dizer que eles **só podem ser removidos** do banco de dados mais tarde por alguma **requisição explícita** ao SGBD. Por **integrados** entende-se que o banco de dados pode ser considerado como uma **unificação de vários arquivos**, com a **eliminação de qualquer redundância parcial ou total** entre esses arquivos.

Por **compartilhados** queremos dizer que o banco de dados pode ser **compartilhado entre diferentes usuários**, no sentido de que diferentes usuários podem ter acesso aos mesmos dados, possivelmente ao mesmo tempo. O banco de dados deve representar algum aspecto do mundo real, ser logicamente coerente e atender a uma necessidade específica. Resumindo:

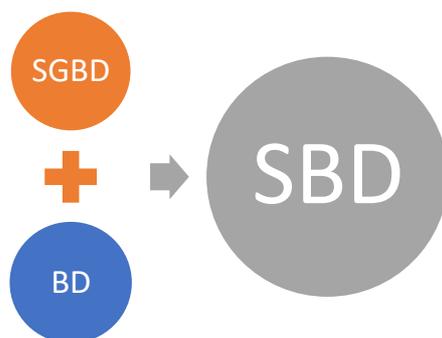




Entre o banco de dados físico – ou seja, **os dados fisicamente armazenados** – e os usuários do sistema existe uma camada de software, conhecida como **gerenciador de banco de dados** ou **servidor de banco de dados** ou, mais frequentemente, como **sistema de gerenciamento de bancos de dados (SGBD)**. Ele é responsável por **armazenar dados** de forma confiável e permitir fácil **recuperação e atualização** desses dados.

Para garantir essa confiança o SGBD apresenta funcionalidades para o controle de redundância, compartilhamento de dados (controle de concorrência), restrição de acesso não autorizado (segurança), tolerância a falhas (backup e recuperação), garantia da integridade e suporte a transações.

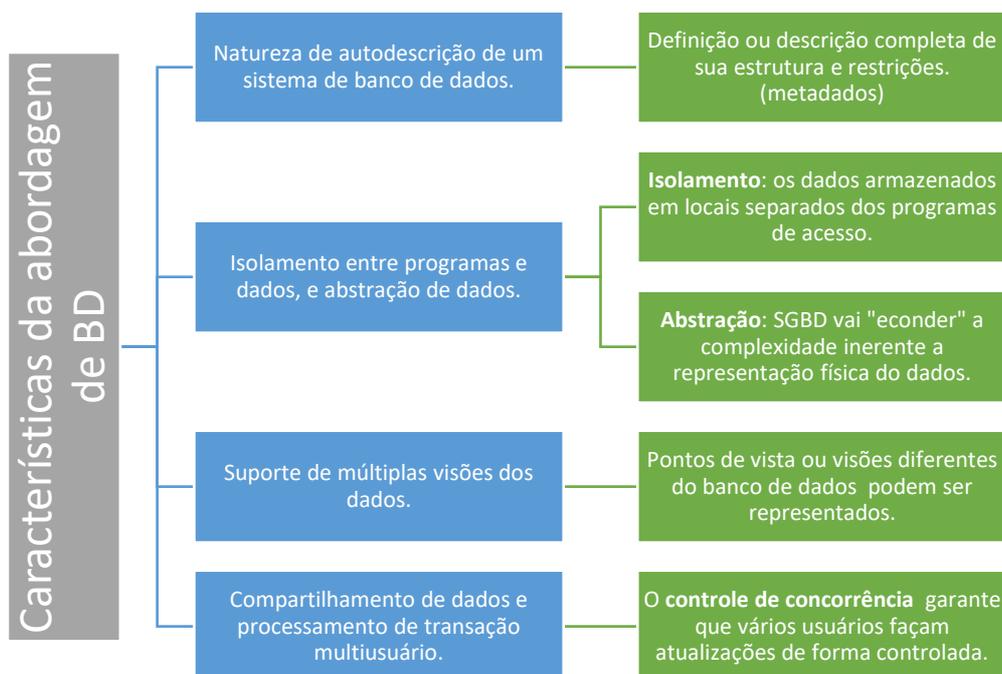
Um **sistema de banco de dados** é basicamente apenas um sistema computadorizado de manutenção de registros. São quatro os componentes fundamentais dos SBD: dados, hardware, software e usuários. O SBD é visto como a união entre o banco de dados e o SGBD. Assim, podemos chegar a seguinte fórmula:



Uma informação relevante que você precisa conhecer é que o banco de dados evoluiu dos antigos sistemas de arquivos. Na abordagem de banco de dados, **um único repositório mantém dados** que são definidos uma vez e depois acessados por vários usuários. Nos sistemas de arquivo, cada aplicação é livre para nomear os elementos de dados independentemente. Em um banco de dados, os nomes ou rótulos de dados são definidos uma vez, e usados repetidamente por consultas, transações e aplicações.

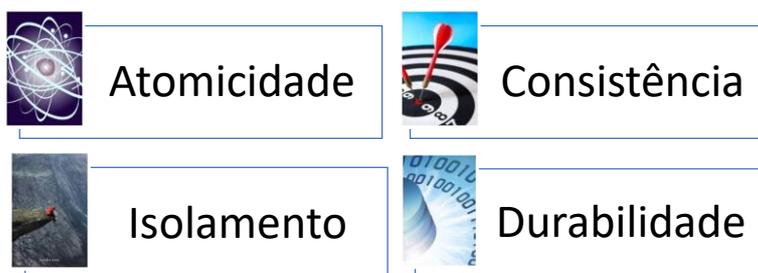
As principais características da abordagem de banco de dados versus a abordagem de processamento de arquivo são as seguintes:





Um **papel fundamental** do software SGBD multiusuário é garantir que as **transações concorrentes** operem de **maneira correta e eficiente**. O conceito de transação tem se tornado fundamental para muitas aplicações de banco de dados. Uma **transação é um programa em execução ou processo que inclui um ou mais acessos ao banco de dados**, como a leitura ou atualização de seus registros.

Uma transação executa um acesso logicamente correto a um banco de dados quando ela é executada de forma completa e sem interferência de outras transações. O SGBD precisa impor quatro propriedades da transação: **atomicidade, consistência, isolamento e durabilidade**.



A propriedade de **atomicidade** garante que **todas as operações em uma transação sejam executadas ou que nenhuma seja**. A propriedade de consistência, garante que, se uma transação for completamente executada do início ao fim sem interferência de outras transações, deve **levar o banco de dados de um estado consistente para outro**.

A propriedade de **isolamento** garante que **cada transação pareça executar isoladamente** das demais, embora centenas de transações possam estar executando concorrentemente. Já a propriedade de **durabilidade** garante que mudanças aplicadas ao banco de dados pela transação confirmada precisam persistir no banco de dados. Essas mudanças não devem ser perdidas por causa de alguma falha.



## Banco de dados: Modelo de dados

Os sistemas de banco de dados **são projetados para gerenciar grandes blocos de informação**. Esses grandes blocos de informação não existem isolados. Eles são parte da operação de alguma organização cujo produto final pode ser informações do banco de dados ou algum dispositivo ou serviço para o qual o banco de dados desempenha um papel de apoio.

Um **modelo de dados** — uma coleção de conceitos que podem ser usados para descrever **a estrutura de um banco de dados** — oferece os meios necessários para alcançar a abstração de dados. A **abstração de dados**, geralmente, se refere à supressão de detalhes da organização e armazenamento dos dados, destacando recursos essenciais para um melhor conhecimento desses dados.

Com estrutura de um banco de dados, queremos dizer os tipos, relacionamentos e restrições que se aplicam aos dados. A maioria dos modelos de dados também inclui um conjunto de operações básicas para especificar recuperações e atualizações no banco de dados.



Muitos modelos de dados foram propostos, e podemos classificá-los de acordo com **os tipos de conceitos** que eles utilizam para descrever a estrutura do banco de dados:

**Modelos de dados de alto nível ou conceituais** oferecem conceitos que são próximos ao modo como muitos usuários percebem os dados. Eles utilizam conceitos como **entidades**, **atributos** e **relacionamentos**. Uma **entidade** representa um **objeto ou conceito do mundo real**, como um aluno ou um concurso do minimundo que é descrito no banco de dados. Um **atributo** representa alguma **propriedade de interesse** que descreve melhor uma entidade, como o nome ou o cargo de interesse do aluno. Um **relacionamento** entre duas ou mais entidades representa uma associação entre elas — por exemplo, um relacionamento inscreve-se-em entre um aluno e um concurso. A figura abaixo representa um diagrama simples que se utiliza da modelagem conceitual.

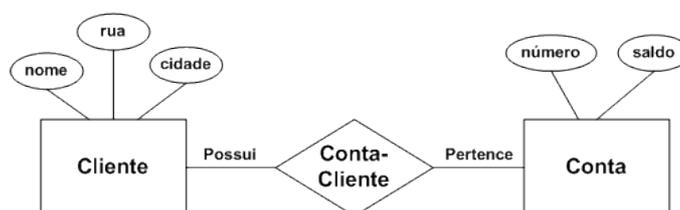


Figura 1 - Diagrama entidade relacionamento

**Modelo de dados representativos** (ou de implementação) oferece conceitos que podem ser facilmente entendidos pelos usuários finais, mas que não está muito longe do modo como os dados são organizados e armazenados no computador. Modelos de dados representativos ocultam muitos detalhes do armazenamento de dados em disco, mas podem ser implementados diretamente em um sistema de computador. São os usados



com mais frequência nos SGBDs comerciais tradicionais. Estes incluem o amplamente utilizado modelo de dados relacional, bem como os chamados modelos de dados legados — os modelos de rede e hierárquicos — que foram bastante usados no passado. Veja um exemplo de modelo lógico na figura a seguir:

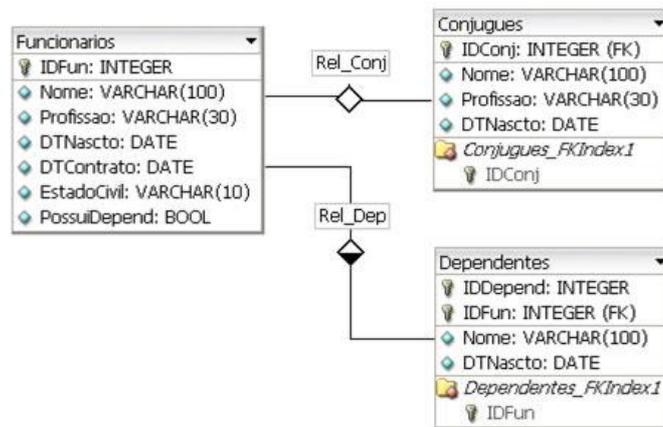


Figura 2 - Diagrama representativo do modelo relacional

**Modelos de dados de baixo nível ou físicos** oferecem conceitos que descrevem os detalhes de como os dados são armazenados no computador, em geral, em discos magnéticos. Os conceitos oferecidos pelos modelos de dados de baixo nível costumam ser voltados para especialistas de computadores, não para usuários finais.

## Banco de dados relacionais: conceitos básicos e características

O modelo relacional retrata os dados como sendo armazenados em **tabelas retangulares bidimensionais**, chamadas de **relações**, similares ao formato no qual a informação é mostrada por programas de planilha. A figura abaixo mostra um exemplo de uma relação, mais especificamente a relação aluno. Perceba que as linhas da tabela são chamadas de **tuplas** e as colunas descreve os atributos da relação.

Nome da relação		Atributos						
<b>ALUNO</b>		Nome	Cpf	Telefone_residencial	Endereco	Telefone_comercial	Idade	Media
Tuplas		Bruno Braga	305.610.243-51	(17)3783-1616	Rua das Paineiras, 2918	NULL	19	3,21
		Carlos Kim	381.620.124-45	(17)3785-4409	Rua das Goiabeiras, 125	NULL	18	2,89
		Daniel Davidson	422.111.232-70	NULL	Avenida da Paz, 3452	(17)4749-1253	25	3,53
		Roberta Passos	489.220.110-08	(17)3476-9821	Rua da Consolação, 265	(17)3749-6492	28	3,93
		Barbara Benson	533.690.123-80	(17)3239-8461	Rua Jardim, 7384	NULL	19	3,25

Na terminologia formal do modelo relacional, uma **linha** é chamada de **tupla**, um **cabeçalho da coluna** é chamado de **atributo** e a **tabela** é chamada de **relação**. O tipo de dado que descreve os tipos de valores que podem aparecer em cada coluna é representado por **um domínio de valores** possíveis.



Um sistema de banco de dados relacional é responsável por armazenar dados de forma confiável e permitir fácil recuperação e atualização desses dados e **neste caso armazena dados de forma relacional, isto é, na forma de linhas e colunas em tabelas.**

Já sabemos que um banco de dados é uma **coleção de dados persistentes** que podem ser **compartilhados e estar inter-relacionados**. Ou seja, os dados são armazenados de modo permanente, como em um disco magnético. Ele também pode ter múltiplas aplicações e usuários, fazendo acesso aos dados armazenados. Tais dados são vistos como unidades separadas e podem ser ligados para fornecer um quadro completo.

Os sistemas de banco de dados relacionais foram desenvolvidos originalmente por causa da **familiaridade e simplicidade**. Como as tabelas são utilizadas para comunicar ideias em vários campos do conhecimento, a terminologia de tabelas, linhas e colunas é facilmente compreendida por diferentes usuários. No modelo, **tanto os dados quanto os relacionamentos são representados em tabelas ou relações.**

Além da familiaridade e simplicidade dos bancos de dados relacionais, há também **uma sólida base matemática**. A matemática dos bancos de dados relacionais implica a conceitualização de tabelas como **conjuntos**. O modelo tem uma base formal sólida, que está estruturada na **teoria dos conjuntos e na lógica de primeira ordem**.

A combinação de familiaridade e simplicidade com a base matemática é tão poderosa, que os SGBDs relacionais são **líderes no mercado de armazenamento de dados operacionais**. O modelo relacional é constantemente descrito como tendo os três aspectos a seguir:



**TOME NOTA!**

**Aspecto estrutural:** os dados no banco de dados **são percebidos pelo usuário como tabelas**, e nada além de tabelas.

**Aspecto de integridade:** essas tabelas satisfazem a certas restrições de integridade ou **limitações nos valores**, que podem aparecer em cada registro.

**Aspecto manipulador:** os **operadores** disponíveis para que o usuário possa **manipular** essas **tabelas** – por exemplo, para propósitos de busca de dados – são operadores que derivam tabelas a partir de outras tabelas. Desses operadores, três particularmente importantes são os operadores de restrição, projeção e junção.

Ok! Vamos sair do texto puro e observar uma figura. Ela vai nos ajudar a entender esses conceitos de forma bem mais tranquila. Veja a figura abaixo:



Tabela Professor (**aspecto estrutural**)

Identidade	Nome	Concurso	Comida Favorita
0000001	Eduardo (Dudu)	ICMS-SP	Pudim
0000002	Roserval Junior	IBAMA	Churrasco
0000003	Ricardo Vale	MIDC	Salada
0000004	Paulo Bilynskyj	Delegado (DHPP)	McDonald's

Colunas:  
Identidade (numérica)  
Comida favorita (texto)  
**(aspecto de integridade)**

Operação:  
Projeção com as colunas  
Nome e Comida Favorita  
**(aspecto manipulador)**

Nome	Comida Favorita
Eduardo (Dudu)	Pudim
Roserval Junior	Churrasco
Ricardo Vale	Salada
Paulo Bilynskyj	McDonald's

O modelo relacional é um modelo de dado **representativo** (ou de **implementação**). Como um modelo, ele procura abstrair e organizar os dados de um banco de dados. Para isso, o modelo relacional define um conjunto de conceitos para representação dos dados. O elemento básico deste modelo está definido pelo conceito de **relação**. Um banco de dados seria, portanto, um conjunto de relações. Cada relação pode ser vista como **uma tabela**. Assim, os SGBDs relacionais devem representar os dados sob a forma de **tabelas bidimensionais** organizadas em linhas e colunas.

Observe que os sistemas relacionais só exigem que o banco de dados seja percebido pelo usuário como tabelas. As tabelas são a **estrutura lógica** em um sistema relacional, **não a estrutura física**. No nível físico, os dados podem ser organizados em outros tipos de estrutura – usando arquivos sequenciais, indexação, *hashing*, cadeias de ponteiros, etc. – desde que ele **possa mapear essa representação armazenada como tabelas no nível lógico**. Isso é um reflexo da abstração provida pelo modelo relacional.

O termo estrutura lógica pretende englobar os níveis conceitual e externo da arquitetura ANSI/SPARC. O detalhe é que os níveis conceitual e externo em um **sistema relacional** serão representados usando o modelo relacional. Os bancos de dados relacionais satisfazem a um princípio muito interessante, chamado **Princípio da Informação**:



Todo o conteúdo de informação do banco de dados é representado de um e somente um modo, ou seja, como valores explícitos em posições de colunas em linhas de tabelas.

Veja que os valores são armazenados em uma tabela. Vamos olhar, mais uma vez, para uma imagem e tentar esclarecer, de uma vez por todas, essa característica. Olhe para o modelo abaixo e perceba que temos duas



tabelas: empregado (EMP) e departamento (DEPTO). Perceba que a coluna DEPTO# da tabela EMP possui os valores dos departamentos.

DEPTO	DEPTO#	NOMEDEPTO	ORÇAMENTO
	D1	Marketing	10M
	D2	Desenvolvimento	12M
	D3	Pesquisa	5M

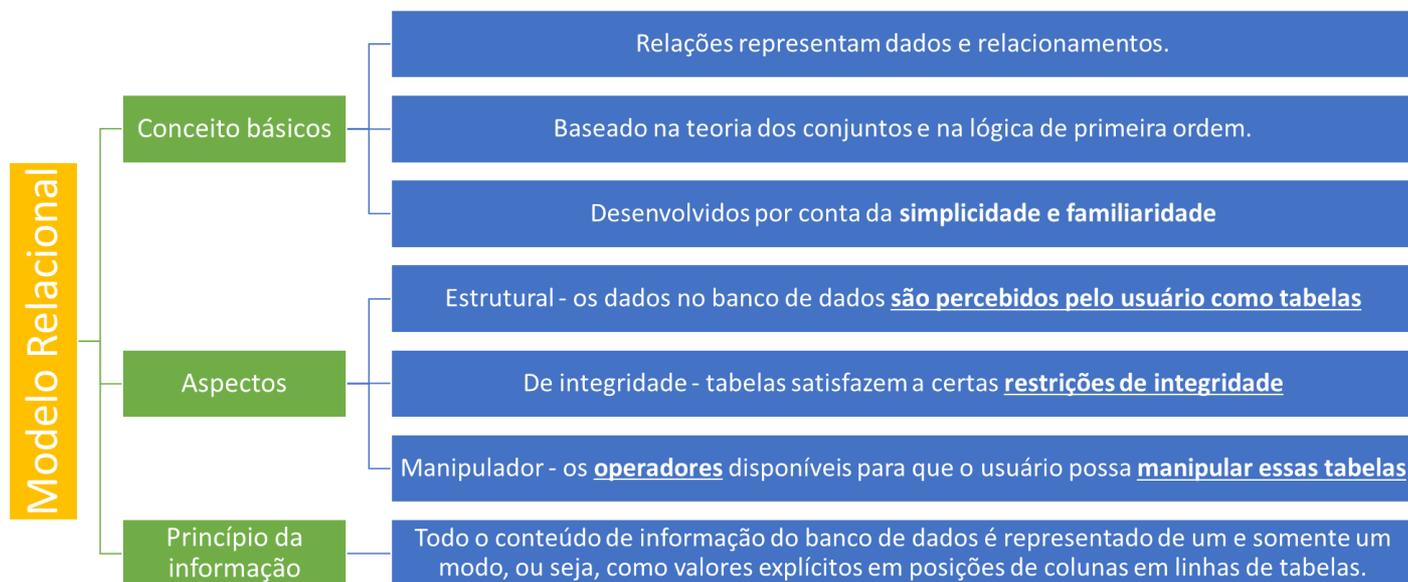
  

EMP	EMP#	NOMEEMP	DEPTO#	SALÁRIO
	E1	Lopez	D1	40K
	E2	Cheng	D1	42K
	E3	Finzi	D2	30K
	E4	Saito	D2	35K

Sendo assim, existe uma conexão entre a linha D1 da tabela DEPTO e a linha E1 da tabela EMP. Essa conexão é representada pelo **aparecimento do valor D1** na tabela EMP. Vamos agora resumir os conceitos que vimos até aqui.



## RESUMINDO



## Metadados

**Metadados:** dados que descrevem outros dados incluindo a fonte, o uso, o valor e o significado dos dados. O Sistema de Gerenciamento de Banco de Dados precisa fornecer uma **função de catálogo ou dicionário**. O catálogo é o lugar em que – dentre outras coisas – todos os diversos esquemas (externo, conceitual, interno) e todos os mapeamentos correspondentes (externo/conceitual, conceitual/interno) são mantidos.

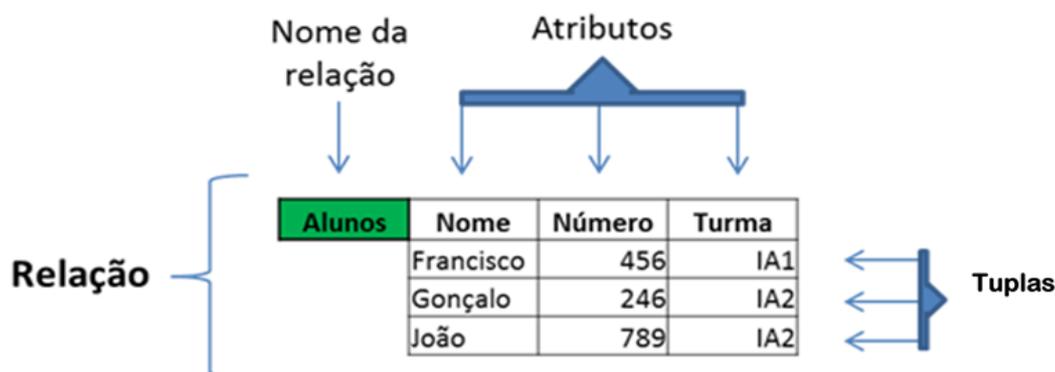
Em outras palavras, o catálogo contém informações detalhadas. Às vezes, elas são chamadas informações do descritor ou metadados, com objetos *RelVars*, índices, usuários, restrições de integridade, restrições de segurança, e assim por diante.

As informações do descritor são essenciais para que o sistema faça seu trabalho de modo apropriado. Por exemplo, o otimizador utiliza informações do catálogo a respeito de índices e outras estruturas físicas de armazenamento, bem como muitas outras informações, para ajudá-lo a decidir como implementar as requisições do usuário. Da mesma forma, o subsistema de autorização utiliza informações do catálogo sobre usuários e restrições de segurança para conceder ou negar tais requisições.

O SGBD armazena as descrições das **construções e restrições do esquema** — também denominadas **metadados** — **no catálogo do SGBD**, de modo que o software do SGBD possa recorrer ao esquema sempre que precisar. O **esquema** às vezes é chamado de **intenção**, e um **estado do banco de dados** é chamado de **extensão do esquema**.

## Tabelas, visões (views) e índices

Cada linha de uma tabela é conhecida como uma **tupla**, ou uma coleção de valores relacionados. Cada coluna é vista como **um atributo**, que possui um determinado conjunto de valores possíveis: o **domínio**. Um domínio é um conjunto de valores atômicos. A figura a seguir resume os conceitos vistos por meio da representação da **tabela Alunos**.



Percebam que a tabela é uma relação dentro do modelo de dados. Cada uma das colunas (Nome, Número e Turma) representa um atributo, que deve ser associado a cada uma das tuplas desta tabela. Cada linha descreve um aluno desta tabela. A primeira linha apresenta os dados de Francisco com seus respectivos número e turma.



Outra constatação que podemos fazer, a partir da figura anterior, é a presença de um conjunto de valores possíveis para cada uma das colunas. Não está explícito. Não sabemos a definição formal da tabela. Contudo, podemos observar que as tuplas desta relação apresentam apenas valores numéricos associados a coluna Número e caracteres associados a coluna Nome. É importante perceber que cada coluna possui um **tipo de dados**.



Um **tipo de dados** define os valores que um dado pode assumir e as operações que podem ser efetuadas sobre o mesmo. Tipos podem ser, por exemplo: inteiros, reais, caracteres, etc.

Quando especificamos **um domínio**, geralmente definimos um tipo de dado do qual são retirados os valores possíveis para o atributo. Além do tipo de dado, um domínio pode ser descrito por meio de um nome, um formato e outras informações adicionais a respeito dos dados. Apenas para exemplificar, um atributo **salário** pode ser definido como **um decimal positivo com 10 dígitos e seu valor ser definido em reais (R\$)**.

Outra característica relevante associada ao domínio é o fato dele **ser atômico**. Isso **faz parte da definição do modelo relacional**. Não se esqueça disso! Outro ponto importante está relacionado à **integridade dos dados**. O modelo apresenta um conjunto de restrições básicas para dados e relacionamentos, conhecidas como **restrições de integridade**.

Ok! Temos as nossas relações! Conseguimos definir, pelo menos de forma gráfica, um conjunto de tabelas com seus atributos. Se quisermos, podemos povoar ou inserir tuplas nessas tabelas com seus respectivos dados. Agora, se eu quiser associar ou analisar os dados de mais de uma tabela conjuntamente, ou ainda, fazer uma consulta que me retorne apenas as tuplas ou linhas com uma certa restrição, o que devo fazer?

Os **operadores relacionais** são usados para manipular as informações pertencentes a cada uma das relações do nosso modelo. Esse conjunto de operações é conhecido como **álgebra relacional**. Ela vai nos ajudar a trabalhar com os dados em nossos esquemas relacionais.

Você deve estar lembrado que o esquema é considerado o projeto do banco de dados, a forma de bolo sobre a qual nossas instâncias são armazenadas. Assim, imagine uma tabela aluno com 100 alunos. O esquema vai definir os atributos necessários para cada linha da relação. Já as linhas são consideradas instâncias da tabela. Lembrando que as linhas podem ser denominadas **tuplas da tabela**. Que tal falarmos um pouco mais sobre elas?

Uma tupla deve conter um conjunto de elementos. Cada elemento deve conter o **nome** do atributo, o nome do **tipo** e um **valor**. Por exemplo:

TUPLA {NOME:CARACTER:THIAGO, CARGO:CARACTER:PROFESSOR}

Veja que cada um dos elementos do conjunto acima, por exemplo, **NOME:CARACTER:THIAGO**, corresponde a um **componente** da tupla e é conhecido como **tripla ordenada**. A quantidade de componentes de uma tupla é conhecida como **grau** ou aridez da tupla. Neste caso, o grau da nossa tupla é igual a 2.



Os dois primeiros valores, **NOME:CARACTER**, são vistos como um par ordenado e descrevem um atributo da tupla. Neste exemplo, **THIAGO** é o valor do atributo em questão. O conjunto completo de atributos é conhecido como **cabeçalho**, este vai determinar o **tipo de uma tupla**.

As tuplas satisfazem a uma série de **propriedades importantes**, todas elas consequências imediatas das definições apresentadas até aqui:

- (a) Cada tupla contém exatamente um valor para cada um de seus atributos.
- (b) Não existe ordenação da esquerda para a direita nos componentes de uma tupla. Essa propriedade acontece porque uma tupla é definida por meio de um conjunto de componentes, e os conjuntos na matemática não possuem qualquer ordenação em seus elementos.
- (c) Cada subconjunto de uma tupla é uma tupla (e cada subconjunto de um cabeçalho é um cabeçalho). Mais que isso, essa afirmação também é válida para um subconjunto vazio em particular!

Esse último ponto merece uma explicação mais detalhada. Existe a tupla vazia (**TUPLA { }**). Às vezes, referimo-nos a uma tupla de grau zero mais explicitamente como uma **“0-tupla”**, a fim de enfatizar o fato de que ela não possui componentes. Outro nome conhecido é de tupla **nulária**.



A relação é um **objeto formal**, enquanto a tabela é um **objeto informal** que pode ser representada em uma folha de papel. Vejamos algumas das diferenças entre tabelas e relações:

Característica	Relação	Tabela
Nomes de tipo são omitidos nos cabeçalhos.	Não	Sim (normalmente)
Nomes de tipo e do atributo normalmente são omitidos no corpo.	Não	Sim
Atributos/colunas possuem uma ordem da esquerda para a direita.	Não	Sim
As tuplas/linhas possuem uma ordem de cima para baixo.	Não	Sim



Pode conter tuplas/linhas duplicatas.	Não	Sim <sup>1</sup>
---------------------------------------	-----	------------------

Falamos das propriedades importantes das tuplas. Neste momento, vamos falar das **propriedades importantes das relações**. Dentro de qualquer relação, temos as seguintes propriedades:

1. Cada tupla contém exatamente um valor (do tipo apropriado) para cada atributo. Trata-se da atomicidade dos atributos.
2. Atributos não são ordenados da esquerda para a direita. Essa propriedade acontece, pois uma tupla é definida em termos de conjuntos matemáticos, que não possuem ordenação dos seus elementos.
3. Tuplas não são ordenadas de cima para baixo.
4. Não existem tuplas duplicadas.

## Visões

---

As views ou visões são objetos SQL, ou relacionais, que podemos criar dentro dos nossos bancos de dados. Uma VIEW é um comando SQL que é armazenado no banco de dados e possui um nome associada a ela. Podemos observar algumas funções básicas. A primeira é facilitar a visualização dos dados dispersos em diversas tabelas, tornando-os mais natural ou intuitivo ao entendimento humano.

Outra função importante para a *view* está relacionada à segurança dos dados. É possível restringir o acesso aos campos e às colunas de uma tabela por meio de uma *view*. Desta forma, o usuário teria visão apenas a parte dos dados ou das informações. Esse grupo de informações deve ser compatível com as funções e as necessidades de acesso do usuário.

Uma terceira opção para o uso de *view* é sumarizar dados de diferentes tabelas, gerando relatórios. Vejamos abaixo dois exemplos do uso de **Views**. Lembrando que ela pode ser criada sobre uma ou múltiplas tabelas. Observe que o comando, basicamente, inclui a sintaxe “CREATE VIEW nome AS” antes de uma consulta ao banco de dados (comando SELECT).

---

<sup>1</sup> Neste caso, pense em uma tabela do Excel. Se estivermos falando de uma tabela no modelo relacional, ela não pode conter linhas duplicadas e todos os seus atributos devem ser atômicos.



```
V1: CREATE VIEW TRABALHA_EM1
AS SELECT Pnome, Unome, Projnome,
          Horas
FROM FUNCIONARIO, PROJETO,
      TRABALHA_EM
WHERE Cpf=Fcpf AND Pnr=Projnumero;

V2: CREATE VIEW DEP_INFO(Dep_nome, Qtd_
func, Total_sal)
AS SELECT Dnome, COUNT (*), SUM
(Salario)
FROM DEPARTAMENTO, FUNCIONARIO
WHERE Dnumero=Dnr
GROUP BY Dnome;
```

Para visualizarmos os dados de uma visão, basta escrevermos um comando SELECT sobre ela, vejamos o exemplo sobre a view **TRABALHA\_EM1** criada acima.

```
SELECT Pnome, Unome
FROM TRABALHA_EM1
WHERE Projnome='ProdutoX';
```

**A view é considerada uma tabela virtual, porque ela só existe durante o período que você está utilizando-a.** Todas as operações que são feitas sobre a tabela podem ser feitas em uma VIEW, mas a tabela é virtual e, na teoria, não deve ser armazenada no banco de dados.

Para finalizar, vamos comparar os conceitos de tabelas e visões de forma pragmática. A distinção entre uma **tabela básica** e uma visão é constantemente caracterizada desta maneira:



Tabelas básicas “existem realmente”, no sentido de que **representam dados de fato armazenados** no banco de dados.  
As visões, ao contrário, não “existem realmente”, mas apenas oferecem diferentes modos de visualização dos “dados reais”.

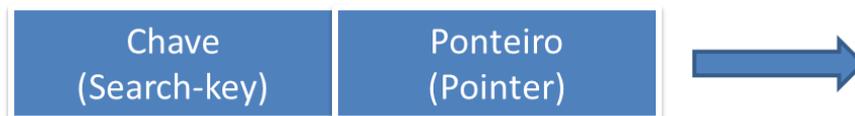
## Índices

O primeiro conceito que devemos ter em mente sobre o assunto é a definição de índices. **Um índice é um mecanismo utilizado para melhorar a velocidade de acesso aos dados.** Ele é composto por uma chave, que

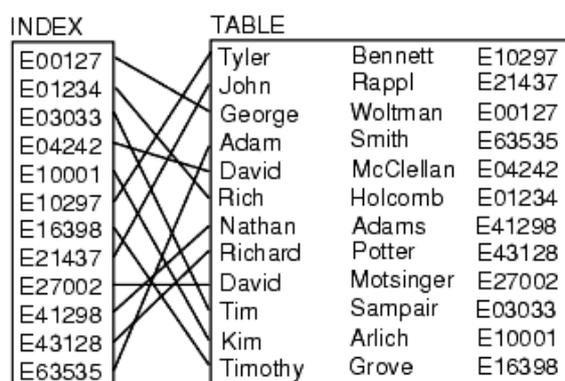


é um atributo ou um conjunto de atributos usado para procurar registros em um arquivo. Um índice também possui um ponteiro, que consiste em um identificador para um bloco de disco, além do deslocamento dentro do bloco para encontrar o registro.

Um arquivo de índice consiste em um conjunto de registros com o formato apresentado na figura abaixo. A esse registro se dá o nome de registro de índice ou entrada de índice.



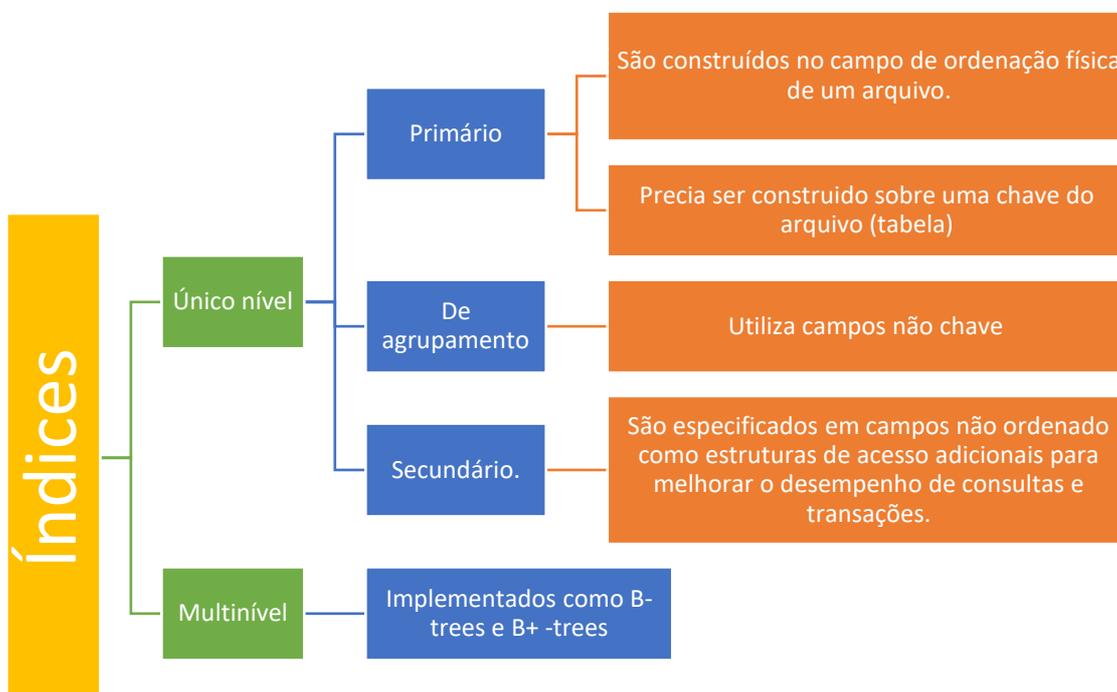
Como uma técnica para criar estruturas de dados auxiliares, os índices agilizam a busca e a recuperação de registros. Para isso, eles envolvem armazenamento de dados auxiliares. Esses dados são armazenados nos arquivos de índices. Alguns tipos de acesso podem se beneficiar dos índices, por exemplo, a localização de registro com um valor especificado e a localização de registros em um intervalo especificado de valores. Veja a figura abaixo para entender um pouco mais sobre índices:



É necessário saber se esse benefício, de fato, traz um ganho de desempenho ao sistema de banco de dados. Avaliar os índices para medir seus efeitos na performance é um passo importante. Vários SGBDs possuem utilitários que ajudam a quantificar os efeitos pretendidos com a criação de índices sobre tabelas. Esse valor é baseado em alguns fatores, entre eles, o tempo de acesso, inserção e deleção; o overhead de espaço em disco e os métodos de acesso suportados.

Os índices podem ser classificados de acordo com a figura abaixo:





## Chaves e relacionamentos

### Superchave, chave, chave candidata, chave primária e chave estrangeira

Vamos começar falando sobre **chave**. Existem alguns conceitos relacionados à chave. O primeiro deles é o de **superchave**. Trata-se de uma coluna ou uma combinação de colunas contendo valores únicos para cada linha. A combinação de todas as colunas em uma tabela sempre é uma superchave, porque as linhas de uma relação devem ser sempre únicas.

Uma definição formal afirma que uma superchave de um esquema de relação  $R = \{A_1, A_2, \dots, A_n\}$  é um conjunto de atributos  $S \subset R$  ( $S$  está contido em  $R$ ) que contenha a propriedade na qual não haverá duas tuplas  $t_1$  e  $t_2$ , em qualquer estado válido da relação, cuja  $t_1[S] = t_2[S]$ . Em outras palavras, uma superchave é um conjunto de atributos que tem a característica de restringir o conjunto de tuplas de uma relação em apenas uma linha.

Quando olhamos para um conjunto de atributos em uma tabela que não se repetem em nenhuma das linhas da tabela, podemos considerá-los uma **superchave**.

Já entendemos o conceito de superchave. Uma superchave pode ser considerada apenas uma chave. Neste caso, a chave é defendida como uma **superchave mínima ou irreduzível** (vou chama-la de  $K$ ), onde qualquer remoção de atributo de  $K$  fará com que  $K$  deixe de ser superchave da relação. Se um esquema tiver mais de uma chave, cada uma delas é chamada de **chave candidata**. Entre as chaves candidatas, uma delas é escolhida para ser a chave da relação e é denominada **chave primária**. As demais são renegadas e são denominadas **chaves secundárias ou alternativas**.



Segundo o Date, se  $K$  for um conjunto de atributos de uma variável de relação  $R$ , então  $K$  é uma **chave candidata** para  $R$ , se e somente se, ela possui ambas as propriedades.



**Unicidade:** Nenhum valor válido de  $R$  contém duas tuplas diferentes com o mesmo valor para  $K$ .

**Irreduzibilidade:** Nenhum subconjunto apropriado de  $K$  tem a propriedade de unicidade.

De posse deste conhecimento, é possível definir o conceito de **atributo primário**, que nada mais é do que um atributo membro (que faz parte) de alguma **chave candidata** da relação ( $R$ ). Por sua vez, de forma bem intuitiva, um **atributo não primário** é todo aquele que não for um atributo primário!



Nesta parte da aula, gostaria que você fixasse o conceito de chave primária e estrangeira. Esse é um conceito importante dentro do assunto de banco de dados. A chave primária vai ser usada pelo SGBD na organização dos arquivos de dados. Isso significa que o acesso aos dados pode ser feito de forma mais rápida, usando o conjunto de atributos definido como chave primária.

O conjunto de atributos deve manter a propriedade de unicidade, ou seja, cada valor associado a uma chave primária só pode aparecer em uma linha da tabela. Olhe para a figura abaixo e me responda: quais campos podem ser usados como chave primária da tabela APROVADOS?

APROVADOS





Id_Aprovado	Nome	Município	CPF	Passaporte
1	Thiago	2611606	111111111111	111111
2	Flávia	2604106	222222222222	222222
3	Vinícius	2604106	333333333333	333333
4	Lucas	Null	444444444444	444444

Chave candidata

Chave primária

Chave Alternativa

Chave Estrangeira

Toda superchave com a propriedade de irredutibilidade.

A **chave candidata designada** para identificar as linhas de uma tabela.

Atributos de uma relação cujos valores devem corresponder a valores de alguma chave candidata de alguma relação.

Id_Município	Nome	Descrição
2611606	Recife	Veneza brasileira
2604106	Caruaru	Capital do forró

Se você pensou nos atributos Id\_Aprovado, CPF e Passaporte, parabéns! Você pensou certo. Qualquer um destes campos pode ser usado como chave primária da tabela, por isso, eles são denominados **chaves candidatas**. No nosso exemplo, escolhemos o atributo Id\_aprovado para ser usado como **chave primária** da relação Aprovados e o campo Id\_Município para ser chave primária da relação MUNICIPIO. As outras chaves que foram preteridas podem ser denominadas **chave alternativa**.

Agora, vamos falar sobre **chave estrangeira**. Observe que, na relação aprovados, temos um campo denominado Município. Esse campo vai “apontar” para a chave primária da tabela Município (Id\_Município). O que acontece aqui é que os valores presentes na coluna Município da tabela aprovados devem estar presentes na chave da tabela município. Outra opção é que o valor seja definido como desconhecido (nulo). Qualquer outro valor que apareça na coluna Município fere a integridade referencial. Neste caso, é como se você estivesse se referindo a um município não cadastrado. Isso fere a integridade referencial que vimos anteriormente.

A chave estrangeira é considerada uma coluna ou uma combinação de colunas em que os valores devem corresponder aos valores de uma chave candidata. A chave estrangeira deve ter tipo de dado igual ao da chave candidata associada. As chaves estrangeiras são usadas no modelo relacional para construir relacionamentos entre as tabelas.

É possível, ainda, um relacionamento em que a chave estrangeira aponta para a mesma tabela. Neste caso, temos um auto relacionamento. Os auto-relacionamentos representam associações entre membros do mesmo conjunto. Os auto-relacionamentos não são comuns, mas são importantes quando ocorrem. No banco de dados de uma universidade, um professor pode supervisionar outros professores e ser supervisionado por um professor. Veja o exemplo de um auto relacionamento na tabela abaixo:



## Auto-relacionamento

CPFProf	NomeProf	SobrenomeProf	CidadeProf	UFProf	DeptoProf	ClassificacaoProf	SalarioProf	SupervisorProf	DataAdmProf	CEPProf
098-76-5432	LEONARD	VINCE	SEATTLE	WA	ADM	ASSISTENTE	R\$35.000	654-32-1098	01-Abril-95	98111-9921
543-21-0987	VICTORIA	EMMANUEL	BOTHELL	WA	ADM	CATEDRATICO	R\$120.000		01-Abril-96	98011-2242
654-32-1098	LEONARD	FIBON	SEATTLE	WA	ADM	ASSOCIADO	R\$70.000	543-21-0987	01-Abril-95	98121-0094
765-43-2109	NICKI	MACON	BELLEVUE	WA	FINAN	CATEDRATICO	R\$65.000		01-Abril-97	98015-9945
876-54-3210	CRISTOPHER	COLAN	SEATTLE	WA	ADM	ASSISTENTE	R\$40.000	654-32-1098	01-Abril-99	98114-1332
987-65-4321	JULIA	MILLS	SEATTLE	WA	FINAN	ASSOCIADO	R\$75.000	765-43-2109	01-Abril-00	98114-9954

## APOSTA ESTRATÉGICA

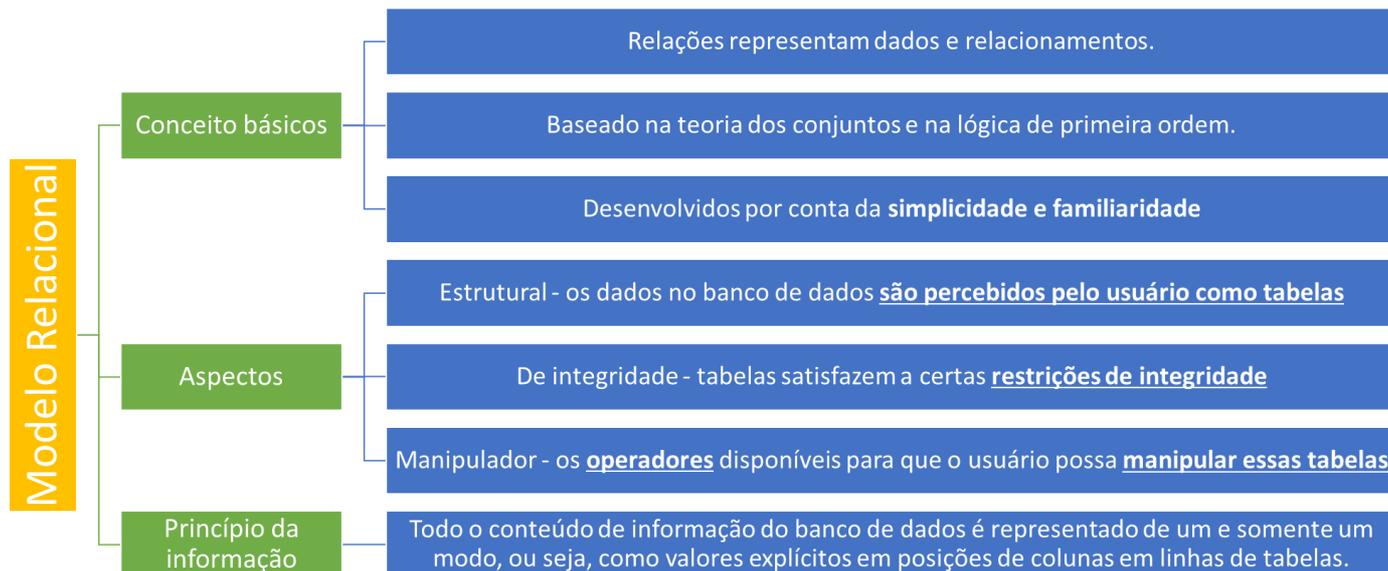
A ideia desta seção é apresentar os pontos do conteúdo que mais possuem chances de serem cobrados em prova, considerando o histórico de questões da banca em provas de nível semelhante à nossa, bem como as inovações no conteúdo, na legislação e nos entendimentos doutrinários e jurisprudenciais<sup>2</sup>.



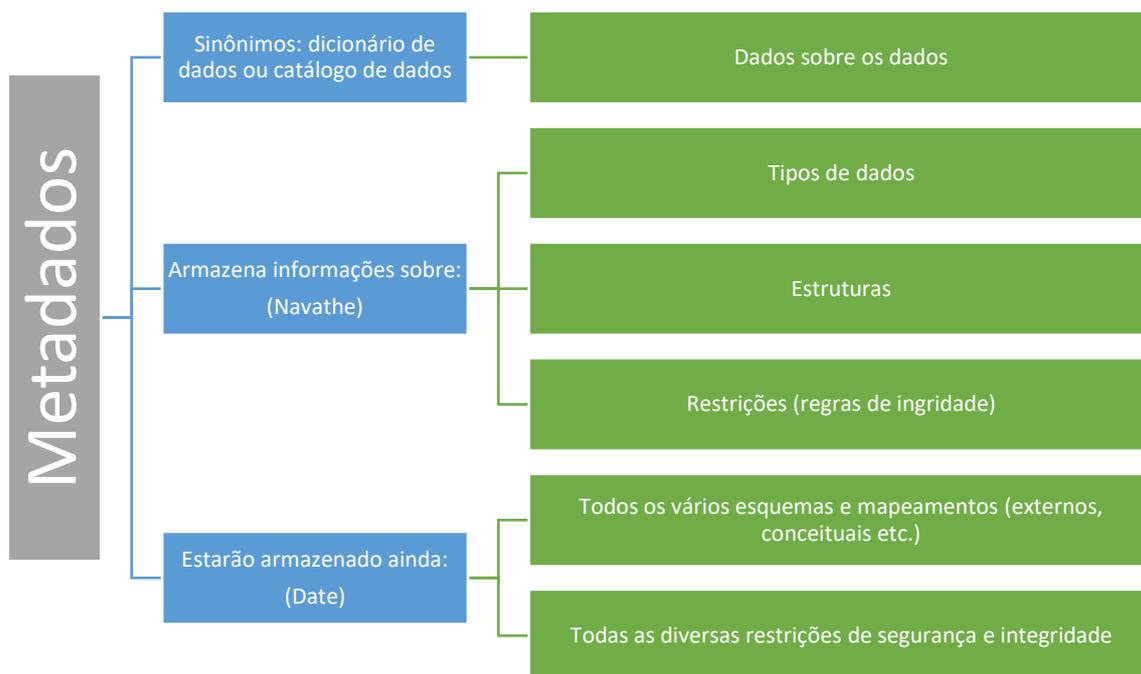
<sup>2</sup> Vale deixar claro que nem sempre será possível realizar uma aposta estratégica para um determinado assunto, considerando que às vezes não é viável identificar os pontos mais prováveis de serem cobrados a partir de critérios objetivos ou minimamente razoáveis.



## Banco de dados relacionais: conceitos básicos e características



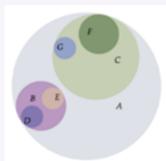
## Metadados



## Tabelas (Restrições de integridade)



## Restrições de Integridade



Integridade de Domínio

Estabelece o conjunto dos valores que podem ser atribuídos a um atributo

Comando SQL: CHECK



Integridade de Vazio

Define se um atributo aceita ou não valores nulos.

Comando SQL: NULL ou NOT NULL

Integridade de Entidade

Afirma que cada tabela deve ter uma chave primária

Estabelece que nenhum valor da chave primária pode ser null



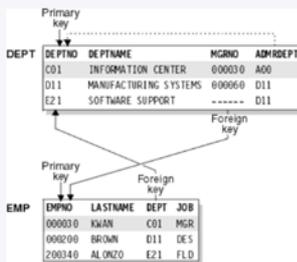
Integridade de Chave

Todos os valores da chave devem ser únicos.

Não se admite repetições de valores

Unicidade

Comando SQL: UNIQUE/PRIMARY KEY



Integridade Referencial

Faz a associação entre duas relações

Limita os valores das chaves estrangeira aos valores presentes na(s) coluna(s) referenciada(s)

Comando SQL: FOREIGN KEY



Integridade Semântica

Descreve regras de negócio da organização

Assertions ou triggers



## Chaves e relacionamentos

Id_Aprovado	Nome	Município	CPF	Passaporte
1	Thiago	2611606	11111111111	111111
2	Flávia	2604106	22222222222	222222
3	Vinícius	2604106	33333333333	333333
4	Lucas	Null	44444444444	444444



RESUMINDO

Atributos de uma relação cujos valores devem corresponder a valores de alguma chave candidata de alguma relação.

Chave candidata

Chave primária

Chave Alternativa

Chave Estrangeira

Toda superchave com a propriedade de irredutibilidade.

A **chave candidata designada** para identificar as linhas de uma tabela.

Id_Município	Nome	Descrição
2611606	Recife	Veneza brasileira
2604106	Caruaru	Capital do forró

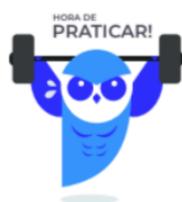
Imprima o capítulo Aposta Estratégica separadamente e dedique um tempo para absolver tudo o que está destacado nessas duas páginas. Caso tenha alguma dúvida, volte ao Roteiro de Revisão e Pontos do Assunto que Merecem Destaque. Se ainda assim restar alguma dúvida, não hesite em me perguntar no fórum.



## QUESTÕES ESTRATÉGICAS

*Nesta seção, apresentamos e comentamos uma amostra de questões objetivas selecionadas estrategicamente: são questões com nível de dificuldade semelhante ao que você deve esperar para a sua prova e que, em conjunto, abordam os principais pontos do assunto.*

*A ideia, aqui, não é que você fixe o conteúdo por meio de uma bateria extensa de questões, mas que você faça uma boa revisão global do assunto a partir de, relativamente, poucas questões.*



## QUESTIONÁRIO DE REVISÃO E APERFEIÇOAMENTO

*A ideia do questionário é elevar o nível da sua compreensão no assunto e, ao mesmo tempo, proporcionar uma outra forma de revisão de pontos importantes do conteúdo, a partir de perguntas que exigem respostas subjetivas.*

*São questões um pouco mais desafiadoras, porque a redação de seu enunciado não ajuda na sua resolução, como ocorre nas clássicas questões objetivas.*

*O objetivo é que você realize uma autoexplicação mental de alguns pontos do conteúdo, para consolidar melhor o que aprendeu ;)*

*Além disso, as questões objetivas, em regra, abordam pontos isolados de um dado assunto. Assim, ao resolver várias questões objetivas, o candidato acaba memorizando pontos isolados do conteúdo, mas muitas vezes acaba não entendendo como esses pontos se conectam.*

*Assim, no questionário, buscaremos trazer também situações que ajudem você a conectar melhor os diversos pontos do conteúdo, na medida do possível.*

*É importante frisar que não estamos adentrando em um nível de profundidade maior que o exigido na sua prova, mas apenas permitindo que você compreenda melhor o assunto de modo a facilitar a resolução de questões objetivas típicas de concursos, ok?*

*Nosso compromisso é proporcionar a você uma revisão de alto nível!*

*Vamos ao nosso questionário:*

### Perguntas

- 1. Defina os termos a seguir quando se aplicam ao modelo de dados relacional: domínio, atributo, tupla n, esquema de relação, estado de relação, grau da relação, esquema de banco de dados relacional e estado de banco de dados relacional.**
- 2. Por que as tuplas em uma relação não são ordenadas?**
- 3. Por que as tuplas duplicadas não são permitidas em uma relação?**
- 4. Qual é a diferença entre uma chave e uma superchave?**
- 5. Por que designamos uma das chaves candidatas de uma relação como sendo a chave primária?**
- 6. Descreva as características de relações que as tornam diferentes das tabelas e arquivos comuns.**
- 7. Descreva os diversos motivos que levam à ocorrência de valores NULL nas relações.**



8. **Descreva as restrições de integridade de entidade e integridade referencial. Por que são consideradas importantes?**
9. **Defina a chave estrangeira. Para que esse conceito é usado?**
10. **Quais são os principais componentes de um sistema de banco de dados?**
11. **Em um banco de dados os dados são persistentes, integrados e compartilhados, explique estes três atributos dos dados em um banco de dados.**
12. **Cite quais as etapas do processo e modelagem de dados em sua respectiva ordem evolutiva e defina cada uma delas.**

## Perguntas com respostas

**1. Defina os termos a seguir quando se aplicam ao modelo de dados relacional: domínio, atributo, tupla n, esquema de relação, estado de relação, grau da relação, esquema de banco de dados relacional e estado de banco de dados relacional.**

Um banco de dados relacional consiste em uma coleção de tabelas. Cada tabela tem um cabeçalho ou uma parte de definição, e um corpo, ou uma parte de conteúdo. A parte do cabeçalho é composta do nome da tabela e nomes de colunas. Por exemplo, uma tabela de alunos pode ter colunas de número do CPF, nome, endereço, cidade, estado, CEP, classe (primeiro, segundo, terceiro ano etc.), especialização e média geral de notas. O corpo mostra as linhas da tabela. Cada linha de uma tabela de alunos representa um aluno matriculado em uma universidade.

**2. Por que as tuplas em uma relação não são ordenadas?**

Essa propriedade também resulta do fato de que o corpo da relação é um conjunto (de tuplas); em matemática, os conjuntos não são ordenados. Quando representamos uma relação como uma tabela no papel, somos forçados a mostrar as linhas dessa tabela em alguma ordem de cima para baixo, porém, mais uma vez, você deverá ignorar essa ordem, se puder.

**3. Por que as tuplas duplicadas não são permitidas em uma relação?**

Essa propriedade resulta do fato de que o corpo da relação é um conjunto; em matemática, os conjuntos não contêm elementos duplicados (o que equivale a dizer que os elementos são todos distintos).

**4. Qual é a diferença entre uma chave e uma superchave?**

Superchave: uma coluna ou uma combinação de colunas contendo valores únicos para cada linha. A combinação de todas as colunas em uma tabela sempre é uma superchave, porque as linhas de uma tabela devem ser únicas. (SUPERCHAVE = UNICIDADE)



Chave: uma superchave mínima. A superchave é mínima se, com a remoção de qualquer coluna, ela deixa de ser única. (CHAVE = UNICIDADE + IRREDUTIBILIDADE)

### **5. Por que designamos uma das chaves candidatas de uma relação como sendo a chave primária?**

Em geral, um esquema de relação pode ter mais de uma chave. Nesse caso, cada uma das chaves é chamada de chave candidata. Chave primária é chave candidata especialmente designada. A chave primária de uma tabela não pode conter valores nulos.

### **6. Descreva as características de relações que as tornam diferentes das tabelas e arquivos comuns.**

Uma relação consiste em um cabeçalho e um corpo. Dentro de qualquer relação temos as seguintes características:

1. Cada tupla contém exatamente um valor (do tipo apropriado) para cada atributo.
2. Atributos não são ordenados da esquerda para a direita.
3. Tuplas não são ordenadas de cima para baixo.
4. Não existem tuplas em duplicata.

### **7. Descreva os diversos motivos que levam à ocorrência de valores NULL nas relações.**

Valores NULL representam atributos cujos valores são desconhecidos ou não existem para alguma tupla individual.

### **8. Descreva as restrições de integridade de entidade e integridade referencial. Por que são consideradas importantes?**

A restrição de integridade de entidade afirma que nenhum valor de chave primária pode ser NULL. Isso porque o valor da chave primária é usado para identificar tuplas individuais em uma relação. Ter valores NULL para a chave primária implica que não podemos identificar algumas tuplas.

A restrição de integridade referencial é especificada entre duas relações e usada para manter a consistência entre tuplas nas duas relações. Informalmente, a restrição de integridade referencial afirma que uma tupla em uma relação que referencia outra relação precisa se referir a uma tupla existente nessa relação.

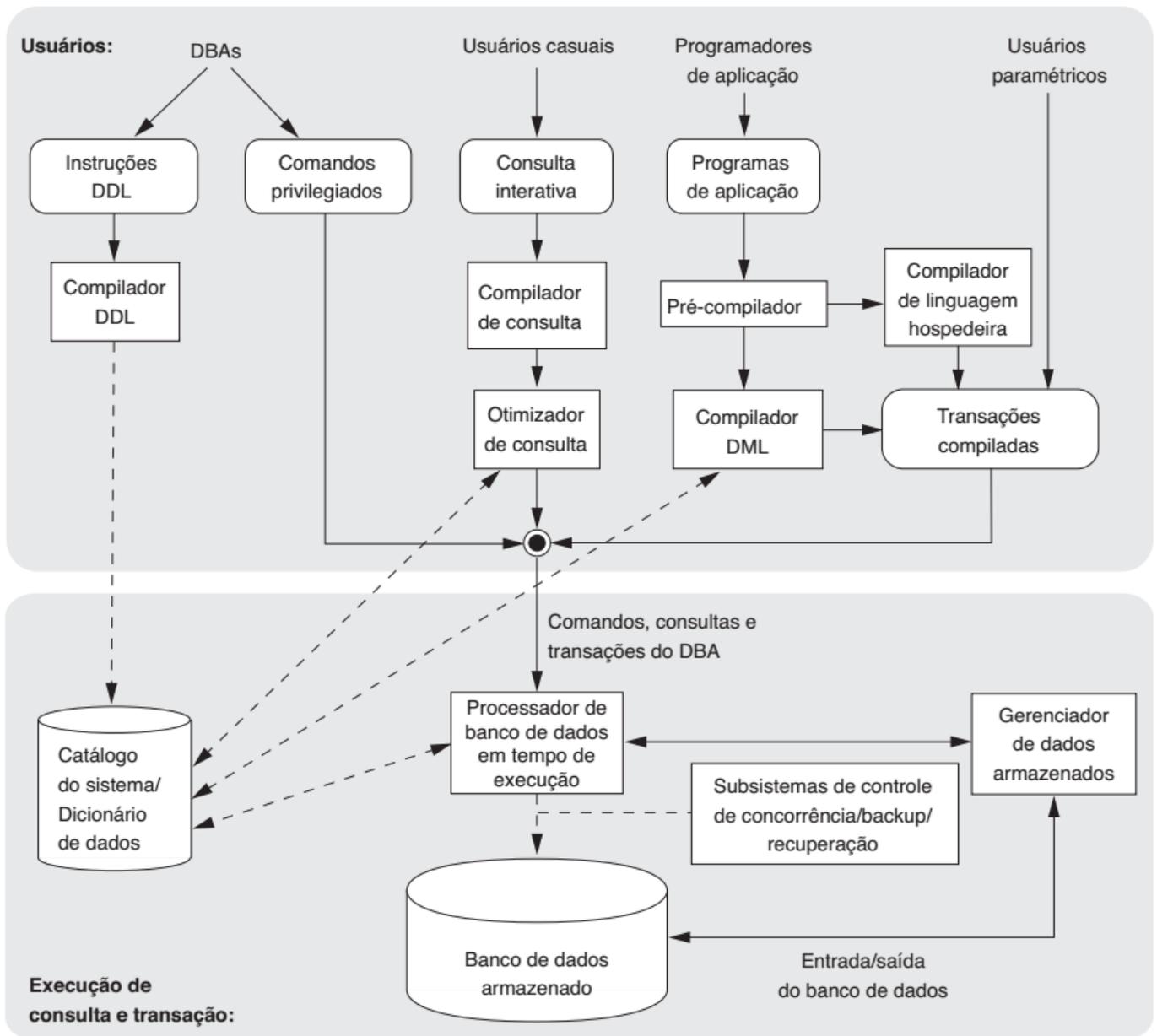
Restrições de integridade são especificadas em um esquema de banco de dados e espera-se que sejam mantidas em cada estado de banco de dados válido desse esquema.

### **9. Defina a chave estrangeira. Para que esse conceito é usado?**

Chave estrangeira pode ser definida como uma coluna ou uma combinação de colunas em que os valores devem corresponder aos valores de uma chave candidata. A chave estrangeira deve ter tipo de dado igual ao da chave candidata associada.



### 10. Quais são os principais componentes de um sistema de banco de dados?



A figura acima ilustra, de uma forma simplificada, os componentes típicos do SGBD. A figura está dividida em duas partes. A parte superior refere-se aos vários usuários do ambiente de banco de dados e suas interfaces. A parte inferior mostra os detalhes internos do SGBD, responsáveis pelo armazenamento de dados e processamento de transações.

### 11. Em um banco de dados os dados são persistentes, integrados e compartilhados, explique estes três atributos dos dados em um banco de dados.

**Persistente:** significa que os dados são armazenados de modo permanente, como em um disco magnético. Por exemplo, as organizações necessitam reter dados sobre clientes, fornecedores e estoque em armazenamento permanente porque esses dados são usados repetitivamente.



**Compartilhado:** significa que um banco de dados pode ter múltiplas aplicações e usuários. Um banco de dados fornece uma memória comum para múltiplas funções em uma organização. Por exemplo, um banco de dados pessoal pode dar suporte a cálculos de folha de pagamento, avaliações de desempenho, informações exigidas pelo governo e assim por diante.

**Integrado:** significa que dados armazenados como unidades separadas podem ser ligados para fornecer um quadro completo. Por exemplo, um banco de dados de clientes relaciona os dados do cliente (nome, endereço...) com os dados do pedido (número do pedido, data do pedido...) para facilitar o processamento dos pedidos. Os bancos de dados contêm tanto entidades como relacionamentos entre entidades.

## 12. Cite quais as etapas do processo e modelagem de dados em sua respectiva ordem evolutiva e defina cada uma delas.

Em relação aos modelos de dados temos 3 níveis: conceitual, lógico e físico. Vamos revisar cada um deles:

**Modelo Conceitual.** A modelagem conceitual baseia-se no mais alto nível e deve ser usada para envolver o cliente, pois o foco aqui é discutir os aspectos do negócio e não da tecnologia. Os exemplos de modelagem de dados vistos pelo modelo conceitual são mais fáceis de compreender, já que não há limitações ou aplicação de tecnologia específica (não temos a definição do SGBD). O diagrama de dados que deve ser construído aqui é o Diagrama de Entidade e Relacionamento, onde deverão ser identificados todas as entidades e os relacionamentos entre elas. Este diagrama é a chave para a compreensão do modelo conceitual de dados.

**Modelo Lógico.** O modelo lógico já leva em conta algumas limitações e implementa recursos como adequação de padrão e nomenclatura, define as chaves primárias e estrangeiras, normalização, integridade referencial, entre outras. Para o modelo lógico deve ser criado levando em conta os exemplos de modelagem de dados criados no modelo conceitual.

**Modelo Físico.** No modelo físico fazemos a modelagem física do modelo de banco de dados. Neste caso leva-se em conta as limitações impostas pelo SGBD escolhido e deve ser criado sempre com base nos exemplos de modelagem de dados produzidos no item anterior, modelo lógico. Aqui é trabalhado no nível mais baixo de abstração, descrevendo o modo como os dados são salvos em meios de armazenamentos, como discos e fitas, sendo exigido a definição tanto dos dispositivos de armazenamento físico como dos métodos de acesso (físico) necessários para se chegar aos dados nesses dispositivos, o que o torna dependente tanto de software como de hardware.

Forte abraço e bons estudos.

**"Hoje, o 'Eu não sei', se tornou o 'Eu ainda não sei'"**

(Bill Gates)

# Thiago Cavalcanti





**Face:** [www.facebook.com/profthiogocavalcanti](http://www.facebook.com/profthiogocavalcanti)  
**Insta:** [www.instagram.com/prof.thiago.cavalcanti](http://www.instagram.com/prof.thiago.cavalcanti)  
**YouTube:** [youtube.com/profthiogocavalcanti](http://youtube.com/profthiogocavalcanti)



# ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



**1** Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



**2** Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



**3** Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



**4** Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



**5** Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



**6** Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



**7** Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



**8** O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.