

TI TOTAL



TI PARA CONCURSOS



Professor
Ramon Souza

Tecnologia da Informação

TEORIA

Bancos de Dados

SUMÁRIO

| | |
|--|----|
| DIRETIVAS DA AULA | 3 |
| GLOSSÁRIO DE TERMOS | 5 |
| 1. BANCOS DE DADOS | 6 |
| 1.1 Conceitos preliminares | 6 |
| 1.2 Características dos bancos de dados | 11 |
| 1.3 Processamento de transações em bancos de dados | 18 |
| 1.4 Metadados e catálogo de dados | 25 |
| 1.5 Projeto de um banco de dados | 28 |
| 1.5.1 Modelos de bancos de dados | 28 |
| 1.5.2 Esquema x Instância | 33 |
| 1.5.3 Arquitetura de três esquemas | 34 |
| 1.5.4 Independência lógica e física de dados | 39 |
| 1.5.5 Modelo de dados x arquitetura de três esquemas | 42 |
| 1.6 Principais modelos de dados (modelos lógicos) | 44 |
| 2. ESQUEMAS DE AULA | 51 |
| 3. MAPA MENTAL | 57 |
| 4. REFERÊNCIAS | 58 |

A nossa aula é bem esquematizada, então para facilitar o seu acesso aos **esquemas**, você pode usar o seguinte índice:

| | |
|--|----|
| Esquema 1 – Banco de Dados (BD) | 6 |
| Esquema 2 – Sistema Gerenciador de Bancos de Dados (SGBD) | 7 |
| Esquema 3 – Sistema de Bancos de Dados (SBD) | 8 |
| Esquema 4 – Características dos bancos de dados | 15 |
| Esquema 5 – Desvantagens da abordagem de SGBD | 16 |
| Esquema 6 – Propriedades ACID | 21 |
| Esquema 7 – Catálogo ou dicionário de dados e metadados | 26 |
| Esquema 8 – Modelos de bancos de dados | 30 |
| Esquema 9 – Esquema x Instância | 33 |
| Esquema 10 – Arquitetura de 3 esquemas | 36 |
| Esquema 11 – Independência lógica e física de dados | 40 |
| Esquema 12 – Modelos de Dados X Arquitetura de três esquemas | 43 |
| Esquema 13 – Modelos de dados (modelos lógicos) | 48 |

DIRETIVAS DA AULA

Esta aula cobre um dos temas mais tradicionais nos concursos públicos: os bancos de dados.

Nesta aula, você entenderá o que é um banco de dados (BD) e um sistema gerenciador de banco de dados (SGBD). Esses são dois conceitos importantes, pois além de serem base para entendimento de outros conteúdos, são cobrados de maneira direta em algumas questões.

Em seguida, você irá conhecer as principais características da abordagem de banco de dados que a diferem de outras abordagens (como o sistema de arquivos). São muitas características, mas aqui sugiro que você se concentre nas principais que são detalhadas na aula e apenas “passe o olho” nas demais, para saber que elas existem.

As principais características são então detalhadas em tópicos específicos. Um dos mais importantes e mais cobrados é o que trata das transações e das propriedades ACID. Muita atenção a esse tópico, pois ele despenca em questões, sendo o favorito das mais diversas bancas. Saiba quais são as quatro propriedades e saiba diferenciar cada uma delas.

O conceito de metadados e catálogo também é cobrado diretamente em algumas questões, mas esse é bem simples e direto. Então, fixe-o e garanta esses pontinhos.

Nesta aula, tratamos do projeto de banco de dados, que inclui principalmente os modelos de dados e arquitetura de três esquemas. Você deve saber claramente diferenciar cada um dos modelos (conceitual, lógico e físico) e também cada um dos níveis da arquitetura (externo, conceitual e interno). Ah, e cuidado para não confundir uma coisa com outra.

Por fim, tratamos dos principais modelos lógicos. Nenhum modelo é detalhado a fundo nessa aula, mas sim mostrado de forma geral, destacando a estrutura de representação utilizada por cada um. Seu objetivo é saber diferenciar os modelos para as questões conceituais. Quando e se necessário aprofundar, faremos em aula específica.

Essa é uma aula conceitual/introdutória, mas de muita importância como base para outros conteúdos e também com conteúdo diretamente cobrado em prova. Então a faça com muito carinho. Não pule nada, apenas ligue o modo atenção total nos seguintes pontos:

- 1.3 Processamento de transações.
- 1.4 Metadados e catálogos de dados.
- 1.5.1 Modelos de bancos de dados.
- 1.5.3 Arquitetura de três esquemas.
- 1.5.4 Independência lógica e física

Um destaque importante é que esse tema é raramente cobrado sozinho, quero dizer, mesmo que o edital cobre apenas “bancos de dados” ou “conceitos de bancos de dados”, ele inclui implicitamente também os bancos relacionais, mas não se preocupe, pois há aula específica sobre o modelo relacional que complementa essa aula com as especificidades do modelo. Essas duas aulas foram divididas apenas para fins didáticos, mas constituem um bloco único de conhecimento para os editais.

ORIENTAÇÕES ESPECIAIS PARA AS BANCAS

CEBRASPE

A banca CEBRASPE costuma “amar” os conteúdos dessa aula. Sendo uma banca mais conceitual, então aqui ela encontra muito subsídio para as questões. Logo, se está estudando para um concurso dessa banca, a sua atenção aqui deve ser redobrada.

Além das recomendações gerais, a banca costuma cobrar muitas questões também sobre os principais modelos, incluindo o de objetos, de rede e hierárquico. Então, além dos temas de foco geral, atenção também ao tópico a seguir:

- 1.6 Principais modelos de dados

FCC

A FCC costuma não cobrar tantas questões conceituais, então vamos encontrar menos questões sobre essa aula. Contudo, sugiro que ainda assim não pule essa aula ou deixe de dar atenção a ela, pois como disse antes, ela é base para entendimento de outros conteúdos.

FGV

A banca FGV costuma ir um pouco além na cobrança de alguns conteúdos, quero dizer, costuma aprofundar na abordagem e nos temas. Sendo assim, para essa banca vale atentar também para os quadros de aprofundamento da aula, pois trazem temas que são cobrados em algumas questões mais avançadas.

CESGRANRIO

A banca CESGRANRIO pode cobrar alguns temas de maneira mais aprofundada, então além das orientações gerais, vale prestar atenção nos quadros de aprofundamento da aula, pois serão necessários para resolução de algumas questões da banca.

VUNESP

A banca VUNESP costuma fazer uma abordagem bem teórico/conceitual dos temas dessa aula. Então vale ter uma atenção redobrada com as definições. Também apresenta algumas questões sobre os modelos lógicos, incluindo objetos, rede e hierárquico, então, além dos temas de foco geral, atenção também ao tópico a seguir:

- 1.6 Principais modelos de dados

GLOSSÁRIO DE TERMOS

Abstração: capacidade de esconder os detalhes de implementação.

Aplicação, aplicativo ou software: programa de computador com objetivo de desempenhar tarefas práticas, em geral ligadas ao processamento de dados.

Atomicidade: propriedade das transações que garante que uma transação seja executada ou por completo ou de forma alguma.

Banco de Dados (BD): coleção coerente de dados que representa um aspecto do mundo real com uma finalidade específica.

Becape ou backup: cópia de segurança dos dados de um dispositivo de armazenamento.

Catálogo ou dicionário de dados: local do SGBD onde ficam os metadados. Possui informações sobre a estrutura de cada arquivo, o tipo e o formato de armazenamento de cada item de dados e diversas restrições sobre os dados.

Concorrência: disputa pelo uso de um mesmo recurso.

Consistência: propriedade das transações que visa a manutenção do cumprimento das regras e restrições em um banco de dados.

Durabilidade: propriedade das transações que garante o armazenamento permanente dos dados.

Esquema de banco de dados: estrutura do banco, sem os dados.

Índice: estrutura de acesso auxiliar que permite agilizar a recuperação de registros. Oferece caminho de acesso secundário.

Instância de banco de dados: dados armazenados em um instante ("fotografia" do banco).

Integridade: capacidade de garantir que a informação não foi alterada sem autorização.

Isolamento: propriedade das transações que garante a não interferência entre transações.

Metadados: dados que descrevem, identificam, explicam e localizam outros dados. Descrevem a estrutura do banco de dados.

Modelo de dados: descrição dos tipos de informações que estão armazenadas em um banco de dados.

Persistência de dados: armazenamento permanente.

Redundância: duplicação dos dados.

Relação: sinônimo de tabela. Bloco de montagem básico do modelo relacional.

Restrição de integridade: regras usadas para garantir a exatidão e a consistência dos dados.

Sistema Gerenciador de Banco de Dados (SGBD): software para criar e manter um banco de dados.

Sistema de Banco de Dados: união do BD com o SGBD.

Transação: programa em execução ou processo que inclui um ou mais acessos ou operações de banco de dados.

Visão ou view: tabela virtual derivada de outras tabelas. Maneira alternativa de visualização dos dados. Consulta pré-definida ou armazenada, executada sempre que referenciada.

1. BANCOS DE DADOS

1.1 Conceitos preliminares

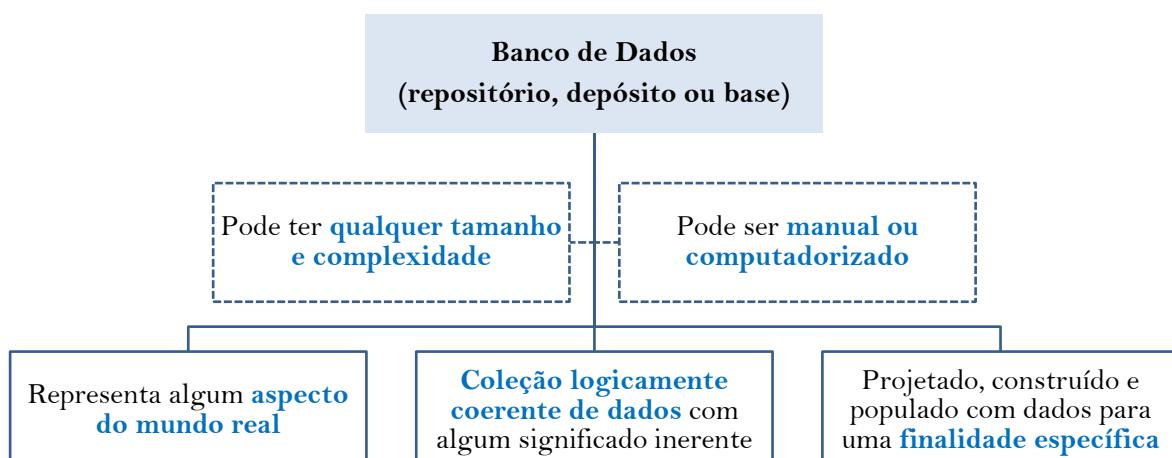
O primeiro conceito fundamental para nossa aula é o de banco de dados.

Um **banco de dados** é uma **coleção de dados relacionados**, em que os dados podem ser entendidos como fatos conhecidos que podem ser registrados e que possuem significado implícito. Com base nessa definição bastante genérica, qualquer conjunto de dados pode ser considerado um banco de dados (até mesmo esta página que você está lendo!!!). Contudo, o uso comum do termo banco de dados não segue essa definição genérica. Um banco de dados pode também ser referido como um **repositório, depósito ou base de dados**.

De modo mais específico, o uso comum do termo banco de dados relaciona-se as seguintes propriedades implícitas:

1. Um **banco de dados** representa algum **aspecto do mundo real**, às vezes chamado de minimundo ou de universo de discurso. As mudanças no minimundo são refletidas no banco de dados.
2. Um **banco de dados** é uma **coleção logicamente coerente de dados** com algum significado inerente. Assim, um banco de dados **não é variedade aleatória de dados**.
3. Um **banco de dados** é projetado, construído e populado com dados para uma **finalidade específica**. Ele possui um grupo definido de usuários e algumas aplicações previamente concebidas nas quais esses usuários estão interessados.

É importante destacar, ainda, que um banco de dados pode ter **qualquer tamanho e complexidade**, não se restringindo a grandes quantidades de dados relacionados. Além disso, pode ser gerado e mantido **manualmente ou de forma computadorizada**, sendo esta última mais comum.



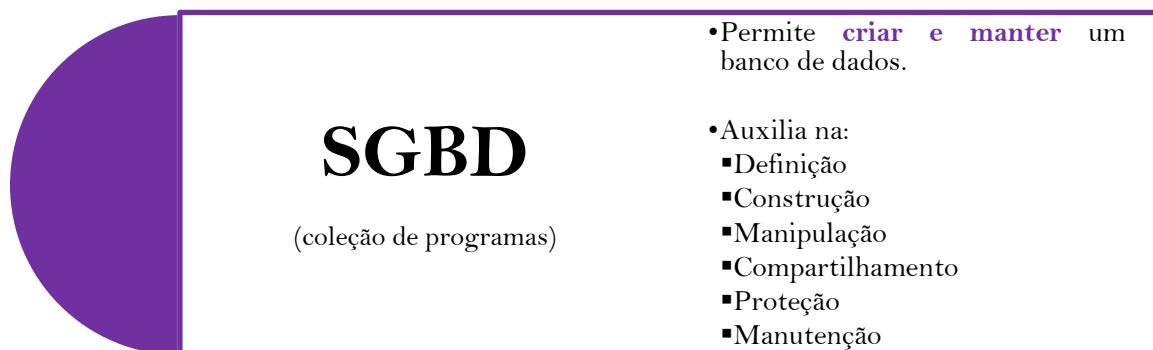
Esquema 1 – Banco de Dados (BD).

Professor, entendi que o banco de dados é o local onde os dados são armazenados, mas como manipular esse banco de dados? Para isso, vamos precisar usar um sistema para interceder por nós nessa comunicação, isto é, um software que consegue realizar as atividades no banco de dados. Esse sistema é o **Sistema Gerenciador de Banco de Dados (SGBD)**.

Um **Sistema Gerenciador de Banco de Dados (SGBD)** é uma **coleção de programas** que permite aos usuários **criar e manter um banco de dados**. O SGBD é um sistema de software de uso geral que facilita os seguintes processos:

- **Definição:** especificação de tipos, estruturas e restrições de dados.
- **Construção:** armazenamento dos dados em algum meio controlado.
- **Manipulação:** consulta, atualização e geração de relatórios.
- **Compartilhamento:** acesso simultâneo por usuários e sistemas.
- **Proteção:** proteção do sistema contra defeitos e proteção de segurança contra acesso não autorizado ou malicioso.
- **Manutenção:** evolução do sistema ao longo do tempo.

Assim,

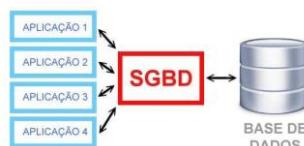


Esquema 2 – Sistema Gerenciador de Bancos de Dados (SGBD).

Existem SGBDS de diversos fornecedores como o Oracle, MySQL, SQL Server, PostgreSQL, entre outros.

ESCLARECENDO!!!

O processo de comunicação com o banco de dados pode ser ilustrado da seguinte forma:



Note que existe um banco de dados e que algumas aplicações desejam o acessar para realizar funções (consultar, alterar ou inserir dados ou até criar e modificar as estruturas do banco). Para isso, essas aplicações deverão acionar o SGBD, que é quem sabe “conversar” com o banco de dados. Assim, através do SGBD, é possível criar e definir as estruturas do banco de dados e manipular os dados. Além disso, o SGBD vai garantir uma série de propriedades como a segurança e o compartilhamento dos dados.

Um outro conceito que precisamos ter em mente é o de **Sistema de Bancos de Dados (SBD)**, que é a **união do banco de dados com o software SGBD**.

Em esquema temos:



Esquema 3 – Sistema de Bancos de Dados (SBD).

EXEMPLIFICANDO!!!

Para que você consiga visualizar melhor esses conceitos, tomemos como exemplo uma lojinha fictícia: lojinha do aprovado.



Essa lojinha precisa armazenar os dados referentes a seus clientes, seus produtos, suas vendas e muito mais. Para isso, ela usará um banco de dados (BD). Vejamos se esse BD possui as três características necessárias para ser assim classificado:

1. Representa um minimundo ou Universo do Discurso? **sim!** Esse BD representa aspectos do mundo real, que são as próprias entidades representadas como os clientes, produtos, vendas, etc.

2. É uma coleção logicamente coerente de dados? **sim!** Esse BD possui dados que estão relacionados ao negócio da lojinha. Não é uma variedade qualquer de dados colocadas em um repositório.

3. Possui uma finalidade específica? **sim!** Esse BD existe para auxiliar o dono da lojinha a gerenciar seu negócio. Com o auxílio desse banco, ele poderá consultar informações sobre seus produtos, saber sobre suas vendas e realizar outras muitas análises.

Então, nesse exemplo, fechamos o conceito de **Banco de Dados**.

Para gerenciar esse banco de dados, isto é, para criar o banco e suas estruturas, bem como para manipular os dados, deve ser usado um **Sistema Gerenciador de Banco de Dados (SGBD)**. Existem SGBDS de diversos fornecedores como o Oracle, MySQL, SQL Server, PostgreSQL, entre outros. Suponha que foi escolhido o Oracle para gerenciar o banco de dados da lojinha, logo o Oracle será o SGBD do nosso exemplo.

Agora ficou clara a distinção entre BD e SGBD? Espero que sim.

E o **Sistema de Banco de Dados (SBD)**? É só o termo conceitual para se referir a união do BD da lojinha com o SGBD Oracle.

Vejamos como os conceitos preliminares já foram abordados em prova:

1- (CESPE / CEBRASPE - 2024 – MPE-GO – Analista do Ministério Público)
Acerca dos conceitos de banco de dados e de modelagem, julgue o item a seguir.

Um conjunto de dados inter-relacionados, representando informações sobre um domínio específico, que pode ser armazenado em meio eletrônico, em uma máquina local ou remota, gerenciado por algum sistema específico de software denomina-se banco de dados.

Resolução:

A assertiva define corretamente um banco de dados.

Um **banco de dados** é uma **coleção logicamente coerente de dados** com algum significado inerente e é projetado, construído e populado com dados para uma **finalidade específica (domínio específico)**.

O sistema de software que gerencia esse banco de dados referido pela questão é o **Sistema Gerenciador de Banco de Dados (SGDB)**, que é uma **coleção de programas para criar e manter o banco de dados**.

Gabarito: Certo.

2- (CESGRANRIO - 2023 - BB – Escriturário) Um Sistema Gerenciador de Banco de Dados (SGBD) é um software bastante utilizado em empresas que precisam armazenar, tratar e utilizar dados em geral. O SGBD é especializado em realizar atividades relacionadas aos dados.

Uma das várias funcionalidades que um SGBD pode executar é

- a) alertar os administradores da infraestrutura de TI de uma empresa quando há vírus circulando na rede.
- b) controlar que usuários podem ter acesso a que dados.
- c) estimular os gestores de uma empresa a compartilhar dados em benefício de todos.
- d) garantir a sequência de execução de programas, em especial quando há dependências de dados entre eles.
- e) identificar que dados importantes ao processo decisório de uma empresa estão ausentes e deveriam ser coletados.

Resolução:

O SGBD é um sistema de software de uso geral que facilita os seguintes processos:

- **Definição:** especificação de tipos, estruturas e restrições de dados.
- **Construção:** armazenamento dos dados em algum meio controlado.
- **Manipulação:** consulta, atualização e geração de relatórios.
- **Compartilhamento:** acesso simultâneo por usuários e sistemas.

- **Proteção:** proteção do sistema contra defeitos e proteção de segurança contra acesso não autorizado ou malicioso.
- **Manutenção:** evolução do sistema ao longo do tempo.

Agora vamos aos itens:

- Incórrito:** detectar vírus não é função de um SGBD, mas sim de um antivírus.
- Correto:** o controle de usuários está entre as funções do SGBD, que deve permitir o compartilhamento dos dados e o acesso simultâneo por usuários e sistemas.
- Incórrito:** um SGBD não possui função de incentivar os gestores a compartilhar dados.
- Incórrito:** o SGBD não controla os programas. Na verdade, o SGBD é uma interface entre o programa/aplicação do usuário e o banco de dados.
- Incórrito:** não é função do SGBD identificar dados importantes para o processo decisório. Isso será realizado com base em análises dos dados.

Gabarito: Letra B.

3- (CESPE - 2019 - SEFAZ-RS - Auditor Fiscal da Receita Estadual) As funções de um sistema de gerenciamento de banco de dados (SGBD) incluem

- gerenciar o backup e a recuperação de dados, bem como o escalonamento de processos no processador por meio do banco de dados.
- gerenciar o sistema de arquivos e a segurança do banco de dados.
- gerenciar a entrada e saída de dispositivos, linguagens de acesso ao banco de dados e interfaces de programação de aplicações.
- gerenciar a integridade de dados, o dicionário e o armazenamento de dados, bem como a memória do computador enquanto o SGBD estiver em execução.
- transformar e apresentar dados, controlar o acesso de multiusuário e prover interfaces de comunicação do banco de dados.

Resolução:

Vamos analisar cada um dos itens:

- Incórrito:** escalonamento de processos não é função do SGBD.
- Incórrito:** gerenciar o sistema de arquivos não é função do SGBD.
- Incórrito:** gerenciar dispositivos de entrada e saída não é função do SGBD.
- Incórrito:** gerenciar a memória do computador não é função do SGBD.
- Correto:** todas são funções do SGBD.

As funções destacadas nos itens de a) a d) são do sistema operacional.

Gabarito: Letra E.

1.2 Características dos bancos de dados

Vale ressaltar que a abordagem de bancos de dados não é a única possível para armazenar dados. Outra abordagem comum é a de processamento de arquivos. O uso de uma ou de outra irá depender bastante da necessidade e de quais características são desejadas no armazenamento e acesso aos dados.

ESCLARECENDO!!!

Vamos tratar rapidamente da abordagem de processamento de arquivos.

Tradicionalmente, o armazenamento de dados era realizado com o uso de arquivos (planilhas Excel, arquivos CSV e outros). Essa abordagem é relativamente simples, pois os arquivos podem ser facilmente abertos em softwares editores e manipulados. Veja a planilha a seguir que ilustra um controle de fluxo de caixa.



The screenshot shows an Excel spreadsheet titled "FLUXO DE CAIXA". The top section displays a summary of cash flows:

| ENTRADAS | SAÍDAS | SALDO |
|--------------|--------------|--------------|
| R\$ 2.180,00 | R\$ 1.105,00 | R\$ 1.075,00 |

Below this is a table of transactions:

| DATA | TIPO | VALOR | DESCRIÇÃO | OBSERVAÇÃO | SUBTOTAL |
|------------|---------|------------|---------------------|---------------------|--------------|
| 01/01/2023 | ENTRADA | R\$ 100,00 | CLIENTE TIJOLOS 123 | Boleto | R\$ 100,00 |
| 01/01/2023 | ENTRADA | R\$ 200,00 | CLIENTE BLOCO ABC | | R\$ 300,00 |
| 01/01/2023 | SAÍDA | R\$ 150,00 | CONTA DE LUZ | Crédito em 12x 1/12 | R\$ 150,00 |
| 01/01/2023 | ENTRADA | R\$ 380,00 | RENDIMENTOS | | R\$ 530,00 |
| 01/01/2023 | SAÍDA | R\$ 255,00 | FORNECEDOR 123ABC | | R\$ 275,00 |
| 02/01/2023 | ENTRADA | R\$ 100,00 | FORNECEDOR 123ABC | | R\$ 375,00 |
| 03/01/2023 | ENTRADA | R\$ 100,00 | CLIENTE TIJOLOS 123 | | R\$ 475,00 |
| 04/01/2023 | ENTRADA | R\$ 200,00 | CLIENTE BLOCO ABC | | R\$ 675,00 |
| 05/01/2023 | SAÍDA | R\$ 150,00 | CONTA DE LUZ | | R\$ 525,00 |
| 06/01/2023 | ENTRADA | R\$ 300,00 | RENDIMENTOS | | R\$ 825,00 |
| 07/01/2023 | SAÍDA | R\$ 200,00 | FORNECEDOR 123ABC | Cartão de Crédito | R\$ 625,00 |
| 08/01/2023 | ENTRADA | R\$ 100,00 | FORNECEDOR 123ABC | | R\$ 725,00 |
| 09/01/2023 | ENTRADA | R\$ 100,00 | CLIENTE TIJOLOS 123 | | R\$ 825,00 |
| 10/01/2023 | ENTRADA | R\$ 200,00 | CLIENTE BLOCO ABC | | R\$ 1.025,00 |
| 11/01/2023 | SAÍDA | R\$ 150,00 | CONTA DE LUZ | | R\$ 875,00 |

The bottom of the spreadsheet has a footer: "PLANILHA DE FLUXO DE CAIXA" and a "Novo" button.

No entanto, a abordagem de sistemas de arquivos apresenta uma série de limitações e problemas para alguns usos. Vejamos algumas delas:

- Dados redundantes:** as informações presentes em um arquivo podem precisar ser duplicadas em diversos outros arquivos.
- Difícil padronização:** embora possa existir algum padrão no arquivo, é difícil garantir que esse padrão não será alterado. Por exemplo, o usuário pode criar uma coluna nova no Excel sem nenhuma restrição.
- Dificuldade de controlar múltiplos acessos:** é muito difícil controlar os múltiplos acessos e as atualizações por parte de diferentes pessoas.

Existem outros problemas, mas esses três já dão uma boa noção sobre a dificuldade de se trabalhar com sistema de arquivos. Não estou dizendo que eles não são úteis, pois essa abordagem ainda é bem utilizada para algumas finalidades. No entanto, para usos mais complexos e que requerem maiores controles, é mais recomendado usar a abordagem de bancos de dados.

Vejamos as principais características da abordagem de bancos de dados. Para Elsmari e Navathe, os bancos de dados divergem dos antigos sistemas de arquivos por apresentarem **quatro características principais**:

- **Natureza de autodescrição de um sistema de banco de dados:** um banco de dados contém uma **definição ou descrição completa de sua estrutura e restrições**. Essa definição é armazenada no **catálogo do SGBD**, que possui informações como a estrutura de cada arquivo, o tipo e o formato de armazenamento de cada item de dados e diversas restrições sobre os dados. A informação armazenada no catálogo é chamada de **metadados**, e descreve a estrutura do banco de dados principal. Perceba então que um **banco de dados não deve conter apenas os dados**, mas também as descrições das estruturas de armazenamento desses dados. Assim, por exemplo, para uma tabela clientes, haverá o armazenamento não só dos dados dos clientes, mas também da própria estrutura da tabela (nome, colunas, tipos de dados e restrições aplicáveis).

EXEMPLIFICANDO!!!

As tabelas a seguir representam um trecho da estrutura de um banco de dados.

| RELACIONES | | COLUNAS | | |
|--------------|-------------------|-------------|---------------|--------------------|
| Nome_relacao | Numero_de_colunas | Nome_coluna | Tipo_de_dado | Pertence_a_relacao |
| ALUNO | 4 | Nome | Caractere(30) | ALUNO |
| DISCIPLINA | 4 | Tipo_aluno | Inteiro(1) | ALUNO |
| TURMA | 5 | Curso | Tipo_curso | ALUNO |

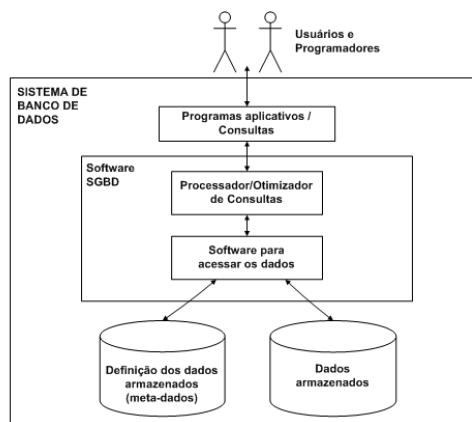
Podemos verificar quais são as tabelas existentes no banco de dados, quais são as colunas das tabelas e quais são os tipos de dados dessas colunas. Logo, perceba que a própria estrutura do banco de dados também é armazenada.

Obviamente que o banco de dados irá armazenar os dados em si. Logo, o banco de dados irá armazenar os dados dos alunos (“João”, “Maria”), das disciplinas (“Matemática”, “TI”) e das turmas (“A”, “B”).

- **Isolamento entre programas e dados; e abstração de dados:** a **abstração de dados** é a capacidade de um SGBD oferecer aos usuários **uma representação conceitual de dados que não inclui muitos detalhes** de como os dados são armazenados ou como as operações são implementadas. Para o usuário, não importa se uma informação é armazenada de forma A ou B, ele só deseja usar o banco e realizar suas operações. Decorrem deste conceito:
 - **Independência dos dados dos programas:** a **estrutura dos arquivos de dados** é armazenada no catálogo do SGBD **separadamente dos programas** de acesso.
 - **Independência da operação do programa:** alguns bancos de dados trabalham com o conceito de operações, que são funções ou métodos que podem ser invocados pelo programa. Os programas podem atuar invocando essas operações, **independentemente de como estas estão implementadas**.

EXEMPLIFICANDO!!!

A figura a seguir ilustra um sistema de banco de dados.



Perceba que os bancos de dados são representados em separado dos programas que os acessam (independência dos dados do programa). Os bancos de dados podem estar até mesmo em locais totalmente diferentes dos programas, isto é, em servidores diferentes.

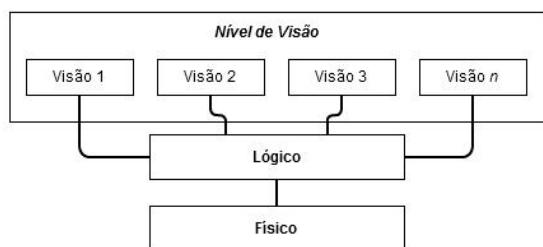
Da mesma forma, os programas aplicativos não estão interessados em como as funções de acesso ou manipulação estão implementadas (independência das operações do programa). Assim, por exemplo, se um sistema de gestão de recursos humanos deseja consultar a base de funcionários para verificar quais funcionários possuem dependentes, ele irá chamar a função, sem necessariamente saber como a consulta é realizada.

Isso é abstração. A complexidade ou os detalhes de como as coisas funcionam por “debaixo dos panos” não precisam ser conhecidos pelos usuários ou softwares aplicativos. Não importa nem mesmo onde ou como os dados são armazenados.

- **Suporte de múltiplas visões de dados:** cada usuário pode exigir um **ponto de vista ou visão diferente do banco de dados**. Um SGBD multiusuário, cujos usuários têm uma série de aplicações distintas, precisa oferecer facilidades para definir múltiplas visões.

EXEMPLIFICANDO!!!

A figura a seguir ilustra o modelo de suporte a múltiplas visões:

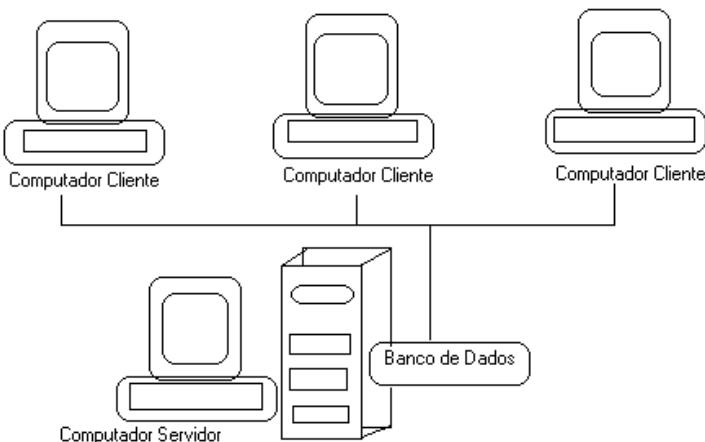


Perceba que um mesmo modelo pode gerar visões diferentes. Vamos supor uma tabela de funcionários. Nesse caso, podemos ter uma visão para o setor de RH com a apresentação de todos os dados dos funcionários. Já para o setor de projetos, também pode ser gerada uma visão, contudo somente com informações específicas dos funcionários, não apresentando, por exemplo, seu endereço, CPF e outros dados que não são de interesse desse setor.

- **Compartilhamento de dados e processamento de transação multiusuário:** o SGBD precisa incluir um software de **controle de concorrência** para **garantir que vários usuários tentando atualizar o mesmo dado façam isso de maneira controlada**, de modo que o resultado dessas atualizações seja correto.

EXEMPLIFICANDO!!!

A figura a seguir apresenta um acesso compartilhado a um banco de dados.



Perceba que várias máquinas clientes acessam o mesmo banco de dados. Sendo assim, é importante controlar esse acesso para que não haja interferência indevida nos usos simultâneos.

Além dessas características principais, existem diversas outras. Vamos apresentar a seguir uma lista com as características trazidas pelos principais autores.

DICA DO PROFESSOR!!!

Não se preocupe em memorizar todas estas listas, mas tenha noção de que elas se propõem a distinguir a abordagem de bancos de dados da antiga abordagem por sistema de arquivos. Caso seja necessário se aprofundar de alguma destas características, faremos isto em tópico específico.

Vejamos as principais características da abordagem de banco de dados:

Elsmari e Navathe
(quatro principais **características** dos bancos de dados)

- Natureza de autodescrição dos dados.
- Isolamento entre programas e dados; abstração de dados.
- Suporte a múltiplas visões.
- Compartilhamento de dados e processamento de transação multiusuário.

Elsmari e Navathe
(**vantagens** de usar a abordagem **SGBD**)

- Controle de redundância.
- Restrição de acesso não autorizado.
- Armazenamento persistente para objetos do programa.
- Estruturas de armazenamento e técnicas de pesquisa para o processamento eficiente de consulta.
- Backup e recuperação.
- Múltiplas interfaces do usuário.
- Representação de relacionamentos complexos entre dados.
- Restrições de integridade.
- Dedução e ação usando regras.
- Potencial para garantir padrões.
- Tempo reduzido para o desenvolvimento de aplicações.
- Flexibilidade.
- Disponibilidade de informações atualizadas.
- Economias de escala.

Date
(**benefícios** da abordagem de **BD**)

- O dado pode ser compartilhado.
- A redundância pode ser reduzida.
- Inconsistências podem ser evitadas.
- Pode-se utilizar o suporte a transações.
- A integridade pode ser mantida.
- A segurança pode ser aperfeiçoada.
- Requisitos conflitantes podem ser balanceados.
- Padrões podem ser utilizados.

Sylberchatz, Korth e Sudarshan
(**desvantagens** de usar **sistema de arquivo**)

- Redundância e inconsistência de dados.
- Dificuldade de acesso a dados.
- Isolamento dos dados.
- Problemas de integridade.
- Problemas de atomicidade.
- Anomalias de acesso concorrente.
- Problemas de segurança.

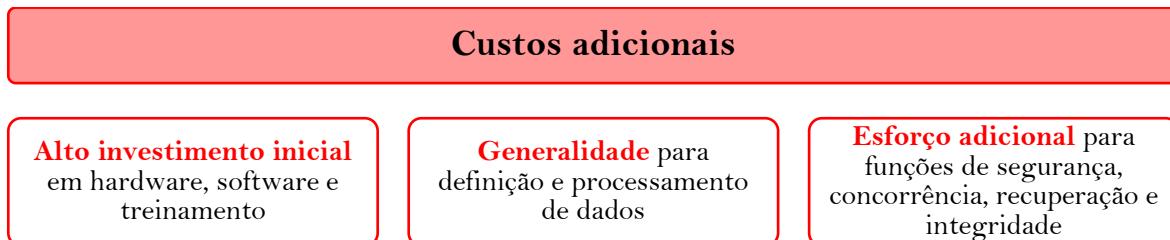
Esquema 4 – Características dos bancos de dados.

Então professor, os bancos de dados resolvem todos os meus problemas?

Não é bem assim, essa abordagem possui também algumas desvantagens, principalmente no que se refere à **custos adicionais do uso de um SGBD** que devem ser considerados. Estes custos adicionais podem ser associados aos seguintes fatores:

- **Alto investimento inicial** em hardware, software e treinamento.
- A **generalidade** que um SGBD oferece **para a definição e o processamento de dados**.
- **Esforço adicional para oferecer funções** de segurança, controle de concorrência, recuperação e integridade.

De forma esquemática:



Esquema 5 – Desvantagens da abordagem de SGBD.

4- (CESPE / CEBRASPE - 2024 – MPE-TO – Analista Ministerial) A respeito de arquitetura de bancos de dados, julgue o item a seguir.

Uma das características de um sistema gerenciador de banco de dados (SGBD) é a independência de dados, ou seja, o entendimento dos dados pelo SGBD ocorre mesmo sem uma aplicação.

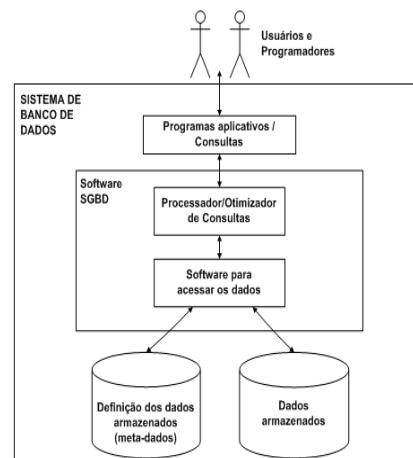
Resolução:

Umas das principais características da abordagem de banco de dados (e consequentemente dos SGBDs) é o **isolamento entre os programas e os dados**. A estrutura dos dados é separada das aplicações que os acessam, portanto, não é necessário haver uma aplicação para se entender os dados.

A figura mostra esse isolamento. Perceba que os **bancos de dados são representados em separado dos programas que os acessam (independência dos dados do programa)**. Os bancos de dados podem estar até mesmo em locais totalmente diferentes dos programas, isto é, em servidores diferentes.

Vale citar que também há a **independência da operação do programa**, em que programas podem atuar invocando operações, **independentemente de como estas estão implementadas**.

Gabarito: Certo.



5- (CESPE - 2018 - IPHAN – Analista I – Área 7) Acerca da abordagem relacional, da normalização e do SGBD, entre outros conceitos relativos a banco de dados, julgue o item a seguir.

Padrões a serem impostos e requisitos contraditórios a serem equilibrados são considerados como desvantagens da abordagem de banco de dados.

Resolução:

Padrões a serem impostos e requisitos contraditórios a serem equilibrados são considerados como **vantagens** da abordagem de banco de dados.

Conhecendo os requisitos globais da empresa, em oposição aos requisitos de usuários individuais, o DBA (Administrador de Banco de Dados), mais uma vez sob a orientação do administrador de dados, pode estruturar o sistema de modo a oferecer um serviço global que seja o melhor para empresa.

Gabarito: Errado.

6- (CESPE - 2013 - MS - Analista Administrativo) Uma das vantagens do uso do SGBD, em relação ao uso do sistema de arquivos tradicional, é a diminuição da ocorrência de redundância de dados, fenômeno que se refere às inconsistências entre as diversas representações do mesmo fragmento de dado em diferentes sistemas e arquivos.

Resolução:

Questão canto da sereia. Cuidado para não ser induzido ao erro pelo examinador maldoso. Em uma análise rápida, a questão parece está correta, mas vamos ver com mais calma.

De fato, uma das vantagens do uso do SGBD, em relação ao uso do sistema de arquivos tradicional, é a diminuição da ocorrência de redundância de dados. Porém, a redundância de dados não se está necessariamente relacionada a inconsistências.

Redundância é o armazenamento dos mesmos dados várias vezes e que pode gerar vários problemas. Porém, na prática, às vezes é necessário usar a redundância controlada para melhorar o desempenho das consultas.

Assim, a assertiva está errada, pois define redundância como inconsistências. O raciocínio é o seguinte: redundância não é inconsistência, mas pode gerar inconsistências.

Gabarito: Errado.

1.3 Processamento de transações em bancos de dados

Vamos detalhar a característica dos bancos de dados do **compartilhamento de dados e processamento de transação multiusuário**, que é uma das mais cobradas em provas. Um programa de aplicação de banco de dados normalmente executa uma ou mais transações. Dessa forma, devemos então entender o que é uma transação e quais as suas propriedades.

Uma **transação** é um **programa em execução ou processo que inclui um ou mais acessos ou operações de banco de dados**, como fazer a leitura do banco de dados ou inserir, excluir e atualizar dados. Dito de outro modo, uma transação é uma **unidade lógica de trabalho** que executa um conjunto de operações em um banco de dados.

Uma única transação pode envolver qualquer número de operações, mas **ao final de uma transação**, ela precisar **deixar o banco de dados em um estado válido ou coerente**, que satisfaça todas as restrições especificadas para o banco, isto é, deve **manter a integridade do banco de dados**.

ESCLARECENDO!!!

Ao falar em unidade lógica estamos nos referindo ao conjunto de operações (consulta, inserções, alterações e exclusões) que podem ser realizadas em um banco de dados, isto é, uma transação pode conter muitas operações.

As operações básicas em um banco de dados são referidas pelo acrônimo **CRUD (Create, Read, Update e Delete)**, respectivamente para **inserção, leitura, atualização e exclusão** de dados.



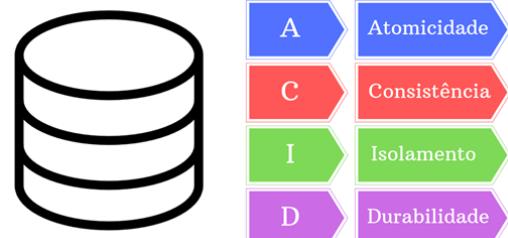
EXEMPLIFICANDO!!!

Por exemplo, se estamos diante de uma transação de transferência financeira, podemos considerar que a transação conterá, simplificadamente, as seguintes operações:

1. Leitura do saldo na conta de origem. (**Read**)
2. Alteração do saldo na conta de origem com a redução do valor. (**Update**)
3. Leitura do saldo da conta de destino. (**Read**)
4. Alteração do saldo da conta de destino com o aumento do valor. (**Update**)

Essas 4 operações constituem **uma única transação**: a de transferência de um valor.

As transações devem possuir quatro propriedades ou princípios básicos, chamadas **propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade)**, que devem ser impostas pelos métodos de controle de concorrência e recuperação do SGBD.



Vamos estudar cada uma das propriedades:

- A **atomicidade** é a propriedade que garante que uma transação é uma **unidade de processamento atômica**; ela deve ser **realizada em sua totalidade ou não deve ser realizada** de forma alguma. Assim, a propriedade de atomicidade exige que uma transação seja executada até o fim. Se uma transação não for completada por algum motivo, como uma falha no sistema no meio da execução da transação, a técnica de recuperação precisa desfazer quaisquer efeitos da transação no banco de dados (**rollback**). Por sua vez, as operações de gravação de uma transação confirmada devem ser, por fim, gravadas no disco (**commit**). A responsabilidade por garantir essa propriedade é do **subsistema de recuperação de transação**.
- Segundo a **consistência**, uma transação deve, se for completamente executada do início ao fim sem interferência de outras transações, **levar o banco de dados de um estado consistente (válido) para outro**. Um **estado consistente** do banco de dados é aquele que **satisfaz todas as suas regras e restrições**. A responsabilidade pela consistência é atribuída aos **programadores** que escrevem os programas de bancos de dados ou ao **módulo do SGBD que impõe restrições de integridade**.
- De acordo com o **isolamento**, uma transação deve parecer executar isoladamente das demais, embora centenas de transações possam ser executadas concorrentemente. Este princípio funciona como um mecanismo de controle que visa assegurar que **nenhuma outra transação**, operando no mesmo sistema, **possa interferir no funcionamento da transação corrente**. Outras transações não podem visualizar os resultados parciais das operações de uma transação em andamento. A responsabilidade por esta propriedade é do **subsistema de controle de concorrência** do SGBD.
- A **durabilidade** trata de **persistir no banco de dados** as mudanças aplicadas por uma transação confirmada. Esta propriedade garante que os resultados de uma transação são permanentes e somente podem ser desfeitos por uma transação subsequente. As **mudanças não devem ser perdidas por causa de alguma falha após a realização da transação**. Esta propriedade é de responsabilidade do **subsistema de recuperação** do SGBD.

ATENÇÃO!!!

Tanto a atomicidade quanto a durabilidade fazem referência a falhas, porém em momentos distintos. Então muito cuidado, pois as bancas vão tentar te confundir sobre isso. Fixe:

A **Atomicidade** trata das **fallas Antes** de finalizada uma transação.

A **Durabilidade** trata das **fallas Depois** de finalizada uma transação.

EXEMPLIFICANDO!!!

Um exemplo clássico de transação é uma transferência de fundos entre duas contas correntes. Suponha uma transação T1 que consiste na transferência de R\$ 100,00 de uma conta corrente X para uma conta corrente Z. Ao final dessa transação, todas as propriedades ACID devem ter sido respeitadas:

Atomicidade: o saldo de X deve ser reduzido em R\$ 100,00 e o de Z deve ser aumentado em R\$ 100,00. Caso ocorra qualquer falha, então os saldos iniciais devem ser retornados.

Consistência: as restrições devem ser respeitadas. Por exemplo, a transferência não deve ser permitida se o saldo de X for menor que R\$ 100,00.

Isolamento: não devemos ter mais de uma transação alterando o saldo das contas simultaneamente.

Suponha uma segunda transação T2 que consiste na transferência de R\$ 200,00 de uma conta Y para a mesma conta Z. Suponha que as transações T1 e T2 executem em paralelo e que os saldos iniciais das contas sejam:

$$X = R\$ 1000,00;$$

$$Y = R\$ 2000,00;$$

$$Z = R\$ 0,00.$$

Vamos supor agora que ocorra o seguinte:

1. A transação T1 inicia, reduz o saldo de X em R\$ 100,00 e aumenta o saldo de Z em R\$ 100,00. Logo,

$$X = R\$ 900,00;$$

$$Y = R\$ 2000,00;$$

$$Z = R\$ 100,00.$$

2. A transação T2 também inicia e transfere R\$ 200,00 para Z. Contudo, nesse momento, ela irá ler o valor R\$ 100,00 na conta Z e, portanto, irá alterar para R\$ 300,00.

$$X = R\$ 900,00;$$

$$Y = R\$ 1800,00$$

$$Z = R\$ 300,00.$$

E qual o problema? Nenhum.

3. O problema vem agora. Suponha que a transação T1 não foi concretizada, isto é, não foi finalizada. Dessa forma, ela deve retroceder ao estado inicial, como se nunca tivesse ocorrido. Sendo assim, o saldo de Z não deveria ter os R\$ 100,00 adicionados.
4. Perceba que a transação T2 realizou a leitura de um saldo em Z que não era um saldo efetivado (leu R\$ 100,00 ao invés de ler R\$ 0,00). Logo, realizou uma leitura suja e acrescentou os R\$ 200,00 nesse saldo.
5. No fim das contas, se isso fosse permitido, os saldos ficariam inconsistentes. Pois, no caso concreto (ROLLBACK de T1 e COMMIT de T2), os saldos deveriam ser:

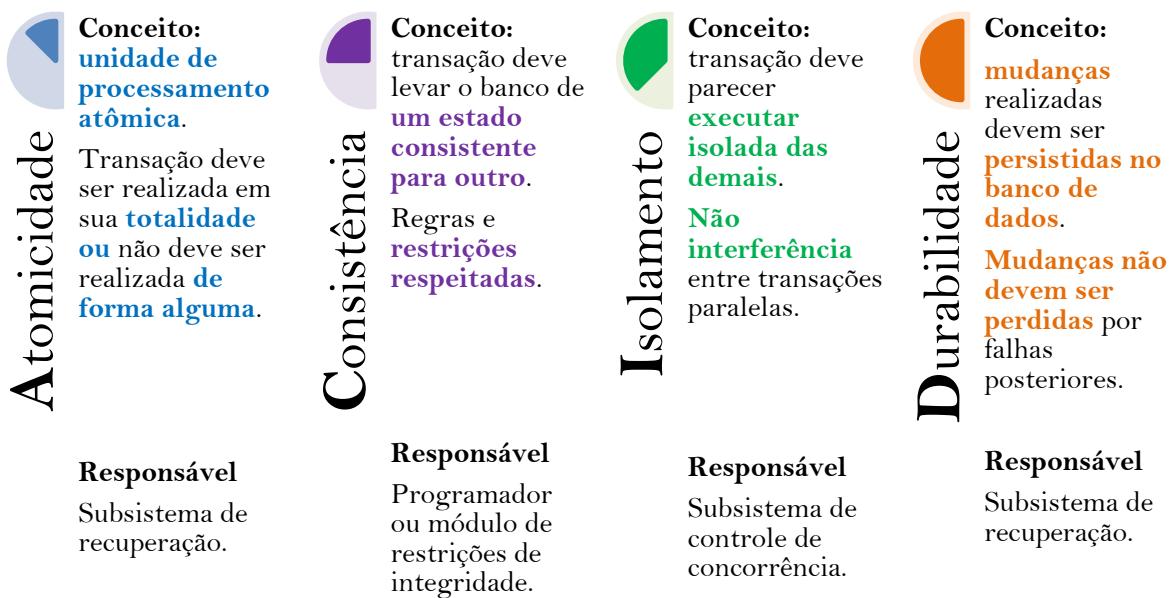
$X = R\$ 1000,00$ (sem alteração no saldo, pois a transação T1 foi revertida)

$Y = R\$ 1800,00$ (a transação T2 foi executada, logo o saldo foi reduzido em R\$ 200,00).

$Z = R\$ 200,00$ (a transação T1 foi revertida, portanto o saldo somente deve sofrer o aumento de R\$ 200,00 da transação T2 e não dos R\$ 100,00 da transação T1). Logo, o valor correto seria R\$ 200,00 e não R\$ 300,00.

Durabilidade: após a transferência ser finalizada, os saldos devem ser persistidos até que novas transações ocorram.

Vamos fixar as propriedades das transações com um **esquema**:



Esquema 6 – Propriedades ACID.

APROFUNDANDO!!!

Vamos abordar os níveis de isolamento de uma transação, pois eles são objeto de algumas questões quando a banca quer “puxar” mais um pouco no tema.

Os **níveis de isolamento da transação** são uma **medida da extensão em que o isolamento de transação é bem-sucedido**. Os níveis de isolamento da transação são definidos pela presença ou pela ausência dos seguintes fenômenos:

- **Leitura suja (dirty read):** ocorre quando uma **transação lê dados que ainda não foram confirmados**. Por exemplo, suponha que a Transação 1 (T1) atualize o saldo de uma conta bancária de R\$100 para R\$200. A Transação 2 (T2) lê o saldo da conta e vê R\$200 antes que T1 confirme a atualização. Se T1 reverter a alteração, o saldo volta para R\$100. Nesse caso, T2 leu um saldo que efetivamente nunca existiu.
- **Leitura não repetível (non-repeatable read):** ocorre quando uma **transação lê a mesma linha duas vezes, mas obtém dados diferentes a cada vez**. Por exemplo, a Transação 1 (T1) lê o saldo de uma conta bancária que é R\$100. A Transação 2 (T2) então atualiza o saldo dessa conta para R\$200 e confirma a atualização. Se T1 reler o saldo da conta, verá \$200, obtendo assim dois valores diferentes (R\$100 e R\$200) para a mesma leitura da linha.
- **Leitura fantasma (phantom read):** um fantasma é uma **linha que corresponde aos critérios de pesquisa, mas não é vista inicialmente**. Por exemplo, suponha que a Transação 1 (T1) leia todas as contas com saldo superior a R\$150 e encontra duas contas (Conta A: R\$200, Conta B: R\$300). A Transação 2 (T2) insere uma nova conta (Conta C) com saldo de R\$250 e confirma a inserção. Quando T1 executa novamente a instrução de leitura, ela encontrará três contas (Conta A: R\$200, Conta B: R\$300, Conta C: R\$250), incluindo a nova conta C que não estava presente na primeira leitura.

CONTINUA...

CONTINUAÇÃO...

Em termos desses fenômenos, há quatro níveis de isolamento, conforme quadro a seguir:

| Nível de isolamento | Fenômenos que podem ocorrer | | |
|--|-----------------------------|-------------------------|--------------------|
| | Leituras sujas | Leituras não repetíveis | Leituras fantasmas |
| Leitura não confirmada (Read uncommitted) | ✓ | ✓ | ✓ |
| Leitura confirmada (Read committed) | ✗ | ✓ | ✓ |
| Leitura repetível (Repeatable read) | ✗ | ✗ | ✓ |
| Serializável (Serializable) | ✗ | ✗ | ✗ |

Cada nível de isolamento oferece um balanço entre a consistência dos dados e a performance do sistema de banco de dados. A escolha do nível apropriado depende dos requisitos específicos de integridade dos dados e desempenho de cada aplicação.

Existem diversos mecanismos que podem ser implementados para garantir o isolamento, mas os principais se baseiam no **uso de bloqueios (locks)**, principalmente o **bloqueio de duas fases (2PL – Two Phase Locking)**.

O protocolo de **bloqueio de duas fases (2PL – Two Phase Locking)** opera em duas fases:

- **Fase de crescimento (Growing Phase)**: a transação pode adquirir bloqueios em qualquer recurso, mas não pode liberar nenhum bloqueio. A transação solicita e obtém bloqueios de leitura ou escrita conforme necessário. Essa fase continua até que a transação tenha adquirido todos os bloqueios necessários.
- **Fase de Encolhimento (Shrinkink Phase)**: inicia quando a transação libera seu primeiro bloqueio. Nesta fase, a transação pode liberar bloqueios que não são mais necessários, mas não pode adquirir novos bloqueios. A transação continua a liberar os bloqueios à medida que vai completando suas operações.

Essa abordagem assegura que as transações sejam executadas de forma serializável, mantendo a integridade dos dados no sistema de gerenciamento de banco de dados. Contudo, uma desvantagem do **2PL** é que ele **não previne deadlocks**, podendo ocorrer que duas ou mais transações fiquem esperando indefinidamente por recursos bloqueados. Imagine duas transações, T1 e T2, que precisam dos recursos A e B. Se T1 bloqueia A e T2 bloqueia B, então, quando T1 tentar bloquear B, terá que esperar, pois T2 já o bloqueou. Da mesma forma, T2 tentará bloquear A, mas precisará esperar porque T1 já o bloqueou. Assim, nenhuma das transações prossegue.

7- (CESPE / CEBRASPE - 2024 – CAU-BR - Assistente) Julgue o próximo item, relativo a banco de dados.

A atomicidade do dado garante que os efeitos de uma transação bem-sucedida sejam permanentemente armazenados no banco de dados, mesmo em caso de falhas no sistema.

Resolução:

Cuidado, pois a propriedade que garante o armazenamento persistente é a **durabilidade**. A **atomicidade** está relacionada a realização de todas as operações ou nenhuma delas. Vale ressaltar que tanto a atomicidade quanto a durabilidade fazem referência a falhas, porém em momentos distintos:

- A **Atomicidade** trata das **fallas Antes** de finalizada uma transação.
- A **Durabilidade** trata das **fallas Depois** de finalizada de uma transação.

A questão cita os **efeitos** de transação bem-sucedida, logo a transação foi finalizada. Portanto, as falhas referidas são após a transação, sendo, portanto, responsabilidade da durabilidade e não da atomicidade.

Gabarito: Errado.

8- (CESPE / CEBRASPE - 2024 – MPE-TO – Analista Ministerial) A respeito de arquitetura de bancos de dados, julgue o item a seguir.

As propriedades autenticidade, consistência, isolamento e durabilidade garantem confiabilidade às transações executadas em um banco de dados.

Resolução:

A **Autenticidade não é uma das propriedades das transações**, mas sim a **Atomicidade**.

As quatro propriedades das transações são Atomicidade, Consistência, Isolamento e Durabilidade.

Gabarito: Errado.

9- (FGV - 2024 – TJ-MS – Técnico de Nível Superior) No processamento de transações em bancos de dados, é fundamental garantir que uma transação não sofra interferências de outras transações executadas simultaneamente. Para isso, o resultado parcial de uma transação não deve ser acessado por outra transação, até a sua efetivação.

Nesse sentido, deve ser assegurada a propriedade de transação conhecida como:

- atomicidade;
- consistência;
- isolamento;
- durabilidade;
- normalização.

Resolução:

A propriedade que garante que uma transação não sofra interferência indevida de outras é o **isolamento**. As quatro propriedades das transações são:

| Atomicidade | Consistência | Isolamento | Durabilidade |
|--|---|--|--|
| Conceito: unidade de processamento atômica. Transação deve ser realizada em sua totalidade ou não deve ser realizada de forma alguma . | Conceito: transação deve levar o banco de um estado consistente para outro. Regras e restrições respeitadas. | Conceito: transação deve parecer executar isolada das demais. Não interferência entre transações paralelas. | Conceito: mudanças realizadas devem ser persistidas no banco de dados. Mudanças não devem ser perdidas por falhas posteriores. |
| Responsável Subsistema de recuperação. | Responsável Programador ou módulo de restrições de integridade. | Responsável Subsistema de controle de concorrência. | Responsável Subsistema de recuperação. |

Gabarito: Letra C.

10- (CESPE / CEBRASPE - 2022 – PC-PB – Perito Oficial) O nível de isolamento especificado pelo padrão SQL em que não é permitida leitura, atualização, inserção ou remoção de qualquer registro que está sendo lido, até que se encontre o commit ou rollback, é chamado de

- a) serializável.
- b) escritas sujas
- c) leitura confirmada.
- d) leitura não confirmada.
- e) leitura repetitiva.

Resolução:

O nível de isolamento mais rigoroso é o serializável. No nível serializável, para evitar os problemas de leitura suja, não repetível e fantasma, uma transação só pode ler registros que foram confirmados por outras transações. Os níveis de isolamento são:

| Nível de isolamento | Fenômenos que podem ocorrer | | |
|--|-----------------------------|-------------------------|--------------------|
| | Leituras sujas | Leituras não repetíveis | Leituras fantasmas |
| Leitura não confirmada (Read uncommitted) | ✓ | ✓ | ✓ |
| Leitura confirmada (Read committed) | ✗ | ✓ | ✓ |
| Leitura repetível (Repeatable read) | ✗ | ✗ | ✓ |
| Serializável (Serializable) | ✗ | ✗ | ✗ |

Gabarito: Letra A.

1.4 Metadados e catálogo de dados

Uma das características dos bancos de dados é a **natureza da autodescrição do sistema de banco de dados** por meio da definição ou descrição completa de sua estrutura e restrições, através dos metadados ou informações do descritor.

Os **metadados (informações do descritor)** são dados estruturados que **descrevem, identificam, explicam, localizam** e, portanto, facilitam a recuperação, uso e gestão de recursos de informação. São os chamados **dados sobre outros dados**.

A **definição ou descrição completa dos metadados** fica armazenada no **catálogo ou dicionário de dados** do SGBD. Esse catálogo contém informações como **a estrutura de cada arquivo, o tipo e o formato de armazenamento** de cada item de dados e diversas **restrições** sobre os dados.

O catálogo armazena informações sobre todos os esquemas e todos os mapeamentos correspondentes entre esses esquemas. Dito de outro modo, o catálogo armazena informações sobre os objetos do banco de dados, incluindo os códigos de definições das estruturas desses objetos.

O catálogo é usado pelo software de SGBD e também pelos usuários do banco de dados que precisam de informações sobre a estrutura do banco de dados.

EXEMPLIFICANDO!!!

Para ajudar no entendimento do que é o catálogo ou dicionário de dados, vamos exemplificar por meio de um trecho de um catálogo de uma universidade fictícia.

| RELAÇÕES | | COLUNAS | | |
|--------------|-------------------|-------------|---------------|--------------------|
| Nome_relacao | Numero_de_colunas | Nome_coluna | Tipo_de_dado | Pertence_a_relacao |
| ALUNO | 4 | Nome | Caractere(30) | ALUNO |
| DISCIPLINA | 4 | Tipo_aluno | Inteiro(1) | ALUNO |
| TURMA | 5 | Curso | Tipo_curso | ALUNO |

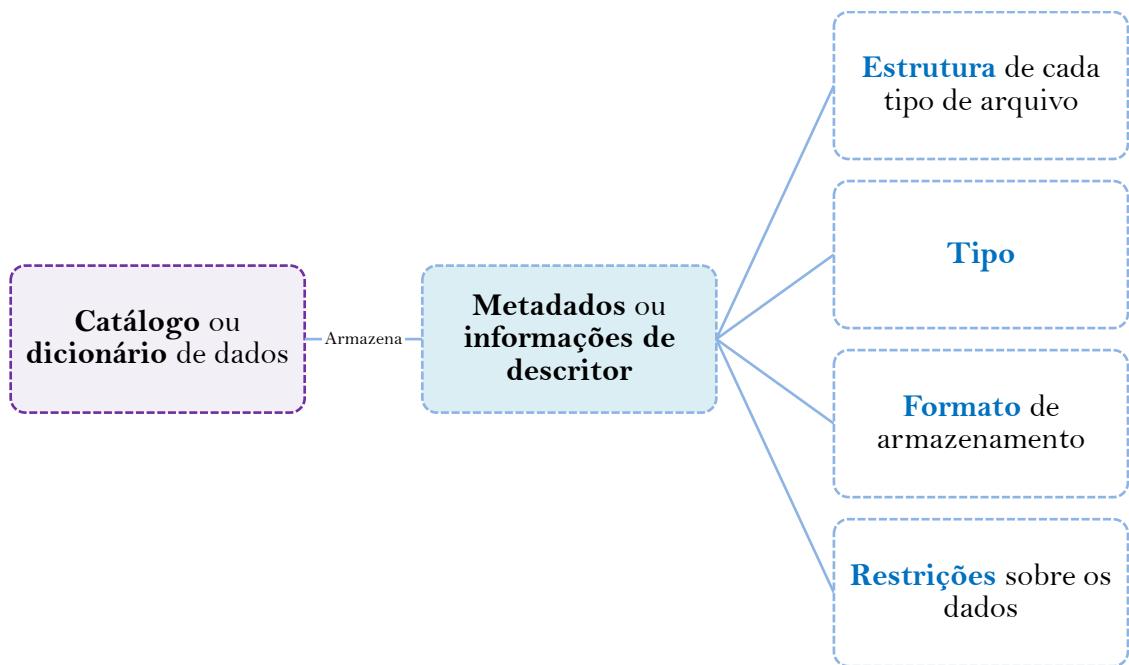
O catálogo do exemplo apresenta as definições das relações que compõem o banco de dados, informando os nomes das relações e o número de colunas de cada uma delas.

Além disso, são apresentadas algumas descrições das colunas, através das definições de seus nomes, dos tipos de dados que podem ser inseridos nestas colunas e a qual relação elas pertencem.

Podemos perceber através do catálogo que a relação aluno é composta de 4 colunas e que, por exemplo, a coluna nome permite a inserção de dados do tipo caractere limitado a um tamanho 30.

Todas essas informações que nos permitem entender os dados são os **metadados**.

E para consolidar o entendimento deste tópico, trazemos **um esquema!!!**



Esquema 7 – Catálogo ou dicionário de dados e metadados.

11- (CESPE / CEBRASPE - 2023 – Prefeitura de Fortaleza – Analista Fazendário Municipal) Julgue o item a seguir, a respeito de arquitetura de dados, metadados e linguagens de bancos de dados.

Os metadados possibilitam uma visão integrada do ambiente de dados, pois explicitam os inter-relacionamentos existentes entre os dados.

Resolução:

Os **metadados** apresentam informações sobre os dados, incluindo **a estrutura de cada arquivo, o tipo e o formato de armazenamento** de cada item de dados e diversas **restrições** sobre os dados.

Essas informações possibilitam a visão integrada dos dados, pois através dos metadados é possível saber como os dados estão organizados e relacionados.

Gabarito: Certo.

12- (CESGRANRIO - 2023 - TRANSPETRO – Profissional Transpetro de Nível Superior) Em um Sistema de Gerenciamento de Banco de Dados (SGBD) relacional, um banco de metadados é um repositório especializado que armazena informações sobre os dados que estão armazenados no banco de dados propriamente dito. Em geral, os metadados ajudam os desenvolvedores, os administradores de banco de dados e outras partes interessadas a entender, a gerenciar e a trabalhar eficientemente com o banco de dados. Dentre as informações típicas que são encontradas nos metadados de um banco de dados relacional, está(ão)

- a) a quantidade total de vendas realizadas, registrada pelas transações.
- b) as senhas de usuários e suas permissões de sistema operacional.
- c) o código-fonte dos aplicativos que acessam o banco de dados.
- d) os backups dos dados armazenados nas tabelas.
- e) os esquemas de tabelas, incluindo os tipos de dados e as restrições de colunas.

Resolução:

Vamos analisar cada um dos itens:

- a) **Incorreto:** a quantidade total de vendas realizadas, registrada pelas transações representam dados em si e não metadados.
- b) **Incorreto:** as senhas de usuários e suas permissões de sistema operacional não são metadados, mas sim dados que devem ser mantidos em um banco de dados de forma segura.
- c) **Incorreto:** o código-fonte dos aplicativos que acessam o banco de dados fazem parte das aplicações. Estas aplicações são isoladas dos bancos de dados, portanto, não tendo relação com os metadados.
- d) **Incorreto:** os backups dos dados armazenados nas tabelas são cópias dos dados, ou seja, no fim das contas são também dados em si e não metadados.
- e) **Correto:** os esquemas de tabelas, incluindo os tipos de dados e as restrições de colunas representam metadados, isto é, dados sobre os dados. Os **metadados (informações do descriptor)** são dados estruturados que **descrevem, identificam, explicam, localizam** e, portanto, facilitam a recuperação, uso e gestão de recursos de informação.

Gabarito: Letra E.

13- (CESPE / CEBRASPE - 2021 – Polícia Federal – Agente de Polícia Federal)
Julgue o próximo item, relativo a conceitos de metadados de arquivos.

A função do metadado de arquivo é descrever o destino final do arquivo definido pelo emissor da mensagem e proprietário do arquivo.

Resolução:

Os **metadados não possuem função de descrever destino** final de arquivos, mas sim de descrever os dados, incluindo **a estrutura de cada arquivo, o tipo e o formato de armazenamento** de cada item de dados e diversas **restrições** sobre os dados.

Gabarito: Errado.

1.5 Projeto de um banco de dados

1.5.1 Modelos de bancos de dados

É importante destacar que a estrutura de um banco de dados pode ser descrita com base em um **modelo de dados**. Um **modelo de (banco de) dados** é uma **descrição dos tipos de informações que estão armazenadas em um banco de dados**.

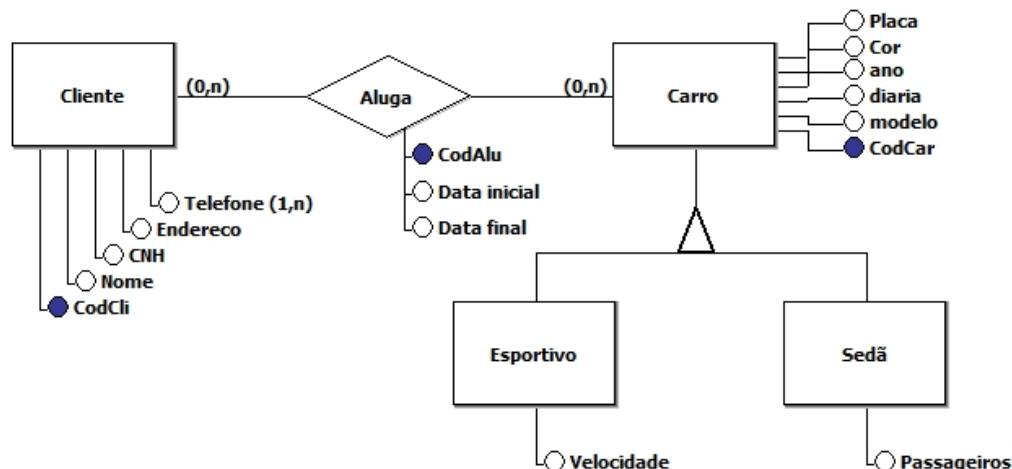
Dependendo do grau de abstração utilizado para representar esta estrutura, falamos em diferentes modelos:

- **O modelo conceitual ou de alto nível** representa os **conceitos que são facilmente compreendidos por usuários**, como entidades, atributos e relacionamentos. Através deste modelo, teremos uma visão macro do ambiente de dados, facilitando o entendimento geral. Este modelo **independente de hardware ou software**, não dependendo de nenhum SGBD específico para implantá-lo. Portanto, as alterações no software ou hardware não terão efeito no modelo conceitual. O modelo conceitual **registra que dados podem aparecer no banco de dados**, mas **não registra como estes dados estão armazenados a nível de SGBD**. Esse modelo é elaborado na chamada **modelagem conceitual**.

EXEMPLIFICANDO!!!

O principal exemplo de modelo conceitual é o **Modelo Entidade Relacionamento (MER)**, representado por Diagramas Entidade-Relacionamento (DER).

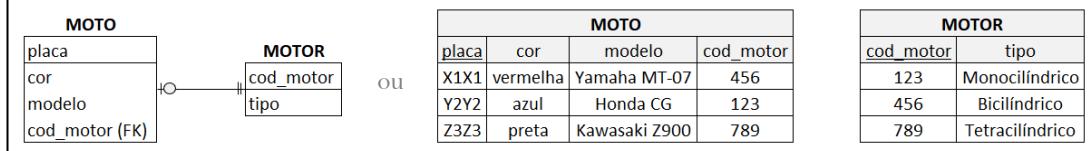
Não se preocupe em entender todos os elementos do diagrama a seguir, mas perceba que ele representa de maneira simplificada que dados serão armazenados no banco de dados. Podemos, por exemplo, perceber que teremos dados de clientes como CodCli, Nome, CNH, Endereço e Telefone.



- No **modelo lógico, representativo ou de implementação**, os conceitos podem ser compreendidos pelos usuários, mas se aproximam da organização e armazenamento dos dados. O **modelo lógico** constitui uma **representação específica de um modelo interno, utilizando as estruturas de banco de dados suportadas**. Um modelo lógico é uma **descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD**. Assim, o modelo lógico é **dependente do tipo particular de SGBD** que está sendo usado (ex.: relacional, objeto, etc.). Esse modelo é elaborado no **Projeto Lógico**.

EXEMPLIFICANDO!!!

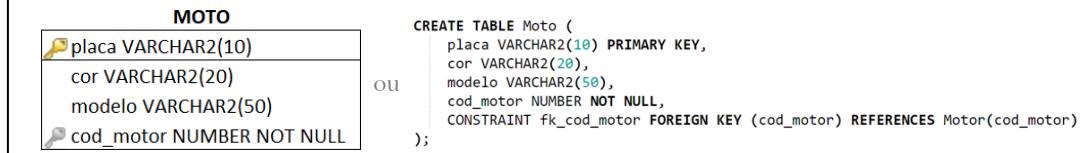
O modelo relacional é um dos principais exemplos de modelos lógicos. Esse modelo utiliza as estruturas de relações (ou tabelas) para representar os dados. O modelo lógico relacional pode apresentar alguns detalhes, a exemplo de chaves primárias que distinguem os registros e chaves estrangeiras para relacionar os registros.



- **O modelo físico descreve o armazenamento dos dados**, sendo **dependente do SGBD específico** (ex.: Oracle, MySQL, etc.). Neste modelo, utiliza-se a linguagem de definição dos dados para criar as estruturas físicas como tabelas, campos, tipos de dados, restrições, etc. Ele é fundamental para o dimensionamento de índices e dados. Esse modelo é elaborado no **Projeto Físico**.

EXEMPLIFICANDO!!!

O modelo físico vai descrever o armazenamento físico, incluindo os tipos e tamanho de campos, indexação, restrições. O modelo a seguir representa a definição de uma tabela no SGBD Oracle. Por exemplo, há informação de que o campo placa é chave primária e é do tipo VARCHAR2 com até 10 caracteres.



ATENÇÃO!!!

Vale destacar que, em relação a dependência de SGBD, algumas questões apenas citam que o modelo lógico depende do SGBD, sem deixar claro se está se referindo ao tipo ou a ferramenta específica. Essas afirmações genéricas, embora confusas, têm sido consideradas corretas. Mas, em regra, temos que:

Modelo conceitual independe de SGBD.

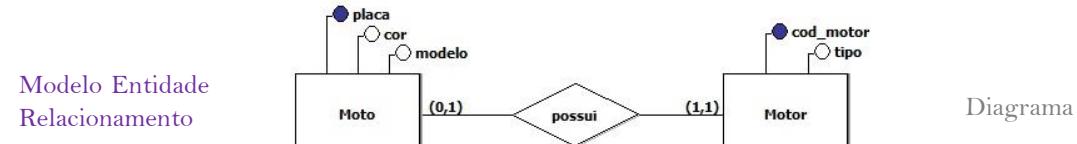
Modelo lógico depende do **tipo ou classe de SGBD**, mas **não do SGBD específico**.

Modelo físico depende **totalmente do SGBD**, isto é, da **ferramenta específica**.

EXEMPLIFICANDO!!!

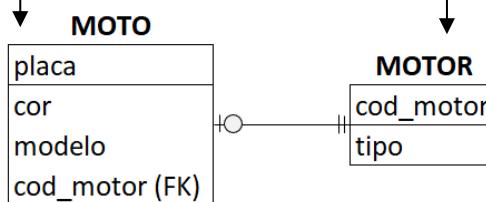
Para consolidar a diferença entre os modelos, veja o esquema a seguir:

Modelo conceitual



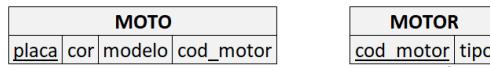
Modelo lógico

Modelo Relacional



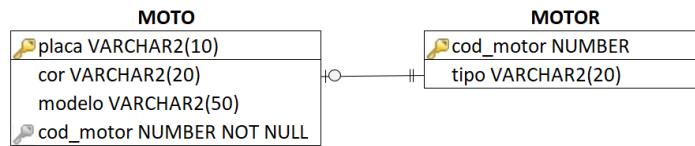
ou

Tabela



Modelo físico

Esquema para implementação no SGDB Oracle



Diagrama

ou

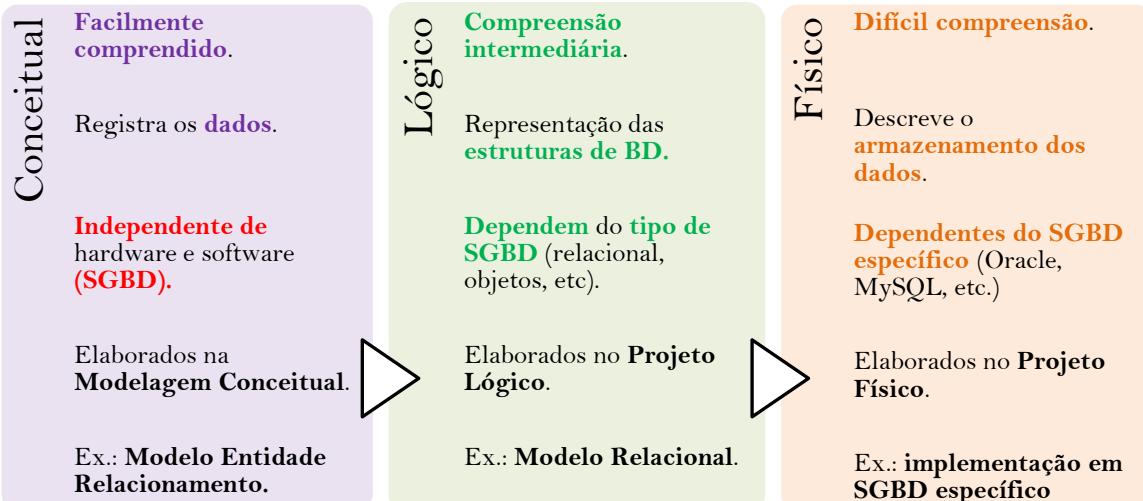
Código

```

CREATE TABLE Motor (
    cod_motor NUMBER PRIMARY KEY,
    tipo VARCHAR2(20)
);

CREATE TABLE Moto (
    placa VARCHAR2(10) PRIMARY KEY,
    cor VARCHAR2(20),
    modelo VARCHAR2(50),
    cod_motor NUMBER NOT NULL,
    CONSTRAINT fk_cod_motor FOREIGN KEY (cod_motor) REFERENCES Motor(cod_motor)
);
  
```

Por fim, um esquema matador para fixar as diferenças entre os modelos:



Esquema 8 – Modelos de bancos de dados.

14- (FGV – 2024 – TJ-RJ – Residente) Com relação às três fases de um projeto de um novo banco de dados, avalie se as afirmativas a seguir são verdadeiras (V) ou falsas (F).

() O modelo conceitual pode ter a forma de um diagrama entidade-relacionamentos e captura as necessidades de uma organização em termos de armazenamento de dados independentemente da sua implementação.

() O projeto lógico tem como objetivo transformar o modelo conceitual obtido na primeira fase em um modelo lógico que definirá como o banco de dados será implementado em um SGBD.

() Na etapa do projeto físico, o modelo de banco de dados é enriquecido com detalhes que influenciam no desempenho do banco mas interferem em suas funcionalidades.

As afirmativas são, respectivamente,

- a) V – F – F.
- b) V – V – F.
- c) F – F – F.
- d) F – V – V.
- e) F – F – V.

Resolução:

Vamos analisar cada um dos itens:

(V) O principal exemplo de **modelo conceitual** é realmente o diagrama entidade-relacionamento. Além disso, o **modelo conceitual** não se preocupa com os detalhes de implementação, por isso, dizemos que ele é **independente de hardware ou software**.

(V) O projeto lógico visa construir o **modelo lógico**, que representa as estruturas que serão suportadas pelo banco de dados e pelo SGBD. Aqui cabe a ressalva que não é necessário saber o SGBD específico, isto é, qual o fabricante (se Oracle, se MySQL, etc.), mas é necessário definir o tipo do SGBD (se relacional, se objeto, etc.).

(F) O modelo físico não altera as funcionalidades definidas, mas sim define os detalhes de armazenamento, como estruturas e restrições. Logo, o projeto físico pode influenciar no desempenho ao escolher as estruturas e forma de armazenamento mais eficientes possíveis. Logo, o item ficaria correto sem o final: “Na etapa do projeto físico, o modelo de banco de dados é enriquecido com detalhes que influenciam no desempenho do banco **mas interferem em suas funcionalidades**.”

Gabarito: Letra B.

15- (CESPE / CEBRASPE - 2024 - CNPq – Analista em Ciência e Tecnologia)
 Julgue o item a seguir, a respeito de mineração de dados, de arquitetura dos dados e de modelagem de dados.

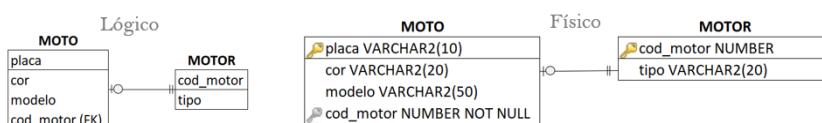
Modelos de dados lógicos fornecem maiores detalhes acerca dos conceitos e relacionamentos no domínio em consideração, indicando atributos de dados, como tipos de dados e seus respectivos comprimentos, e relacionamentos entre entidades, além de informar como será organizado e quais regras de negócios estão envolvidas.

Resolução:

O **modelo lógico não trata dos detalhes de armazenamento** como os tipos de dados e seus comprimentos. Isso fica a cargo do modelo físico.

O **modelo lógico** até define a estrutura de representação dos dados, a exemplos de tabelas em um modelo relacional, mas não entra nos detalhes particulares de implementação e armazenamento. É no **modelo físico** que se definem as estruturas físicas como tabelas, campos, tipos de dados e restrições utilizando as estruturas disponíveis de um SGBD específico (como Oracle ou PostgreSQL).

Perceba a seguir a diferença entre o nível de detalhes do modelo lógico e físico:



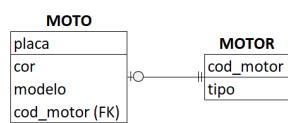
Gabarito: Errado.

16- (CESPE / CEBRASPE - 2024 - INPI – Analista de Planejamento, Gestão e Infraestrutura de Propriedade e Intelectual) Julgue o seguinte item, relacionados a modelagem de dados.

O modelo lógico pode conter chaves primárias e estrangeiras e pode ser usado em vários bancos de dados, tais como SQL Server, MySql, Oracle e PostgreSQL.

Resolução:

O **modelo lógico** define as **estruturas de banco de dados suportadas**. O ponto é que o **modelo lógico não depende do SGBD específico**, mas apenas do tipo. Assim, um mesmo modelo lógico pode ser aproveitado por vários SGBDs diferentes. Nesse caso, para cada SGBD específico será derivado um modelo físico. Um dos principais modelos lógicos é o relacional que define as tabelas como forma de representação dos dados. A representação pode incluir as chaves, como no exemplo:



Nesse esquema, as chaves primárias são representadas na parte superior de cada caixa, assim, placa e cod_motor são chaves, respectivamente, das tabelas MOTO e MOTOR. A chave estrangeira também é representada com (FK), logo o atributo cod_motor é chave estrangeira na tabela MOTO.

Esse modelo lógico pode ser derivado em modelos físicos específicos para qualquer SGBD relacional, a exemplo de Oracle ou MySQL, através do detalhamento das estruturas utilizadas pelo SGBD desejado.

Gabarito: Certo.

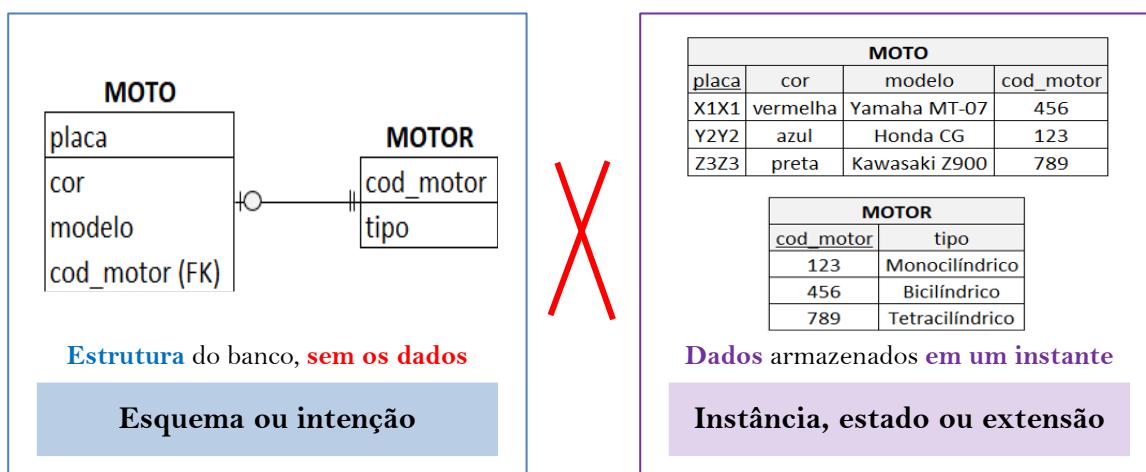
1.5.2 Esquema x Instância

Dois conceitos relacionados a bancos de dados que são, por vezes, cobrados em provas, são os conceitos de esquema e instância de banco de dados. Vejamos então esses termos:

Um **esquema ou intenção** de banco de dados é um **esboço** de um banco de dados **planejado**, isto é, representa a **estrutura do banco**, mas **sem os dados**. Em outra definição, é o **conjunto de regras que governa um banco** de dados. O esquema pode existir tanto como uma representação visual quanto como um conjunto de fórmulas conhecidas como restrições de integridade que regem um banco de dados.

Uma **instância, estado ou extensão** de um banco de dados é uma “fotografia” do seu esquema em um dado momento no tempo, isto é, representa os **dados** que estão **armazenados em um dado instante**.

Sendo assim, instâncias de banco de dados podem mudar ao longo do tempo, enquanto um esquema de banco de dados é geralmente estático, já que mudar a estrutura de um banco de dados a partir do momento que estiver operacional é menos comum.



Esquema 9 – Esquema x Instância.

17- (CESPE / CEBRASPE - 2022 – TCE-SC – Auditor Fiscal de Controle Externo)

Julgue o próximo item, com relação à segurança e arquitetura de banco de dados e administração de dados e de banco de dados.

Um esquema de banco de dados pode ser definido como a representação gráfica e simbólica dos componentes de um banco de dados, seja comercial, industrial ou residencial.

Resolução:

Questão bem capciosa. Embora um esquema possa ser representado graficamente, não é correto defini-lo apenas como tal. O **esquema ou intenção** representa a **estrutura do banco de dados, sem os dados**. Essa estrutura pode ser descrita tanto graficamente quanto textualmente, através das definições de tabelas, tipos de dados e restrições de integridade. Portanto, a definição dada na questão não reflete o que de fato significa um esquema.

Gabarito: Errado.

1.5.3 Arquitetura de três esquemas

De modo a auxiliar no isolamento de programas dos dados e no suporte a múltiplas visões de usuários, além de tornar um banco de dados autodescritivo, foi proposta uma arquitetura de três esquemas. Assim, a **arquitetura de três esquemas ou arquitetura ANSI/SPARC** visa **separar as aplicações do usuário do banco de dados físico**.

Vejamos esses três níveis:

- **No nível externo ou de visão**, cada esquema externo **descreve a parte do banco de dados em que um usuário em particular está interessado** e **oculta o restante do banco de dados** do grupo de usuários. Cada esquema externo é comumente implementado usando um **modelo de dados representativo (lógico)**, possivelmente baseado em um projeto de esquema externo em um modelo de dados de alto nível. Esse nível é também conhecido como **nível lógico de usuário**, pois representa a interação de um usuário ou grupo de usuários com uma parte do banco de dados.
- **O nível conceitual** possui um esquema conceitual que **descreve a estrutura do banco de dados inteiro para uma comunidade de usuários**. O esquema conceitual **oculta os detalhes das estruturas de armazenamento físico** e se concentra na descrição das entidades, tipos de dados, relacionamentos, operações do usuário e restrições. Normalmente, um **modelo de dados representativo (lógico)** é usado para descrever o esquema conceitual quando um sistema de banco de dados é implementado. Esse esquema conceitual de implementação costuma estar baseado em um projeto de esquema conceitual em um modelo de dados de alto nível. Esse nível é também conhecido como **nível lógico de comunidade**, pois representa a visão global e integrada de toda a base de dados para uma comunidade de usuários.
- **O nível interno** tem um esquema interno, que **descreve a estrutura do armazenamento físico do banco de dados**. O esquema interno usa um **modelo de dados físico** e descreve os detalhes completos do armazenamento de dados e caminhos de acesso para o banco de dados.

Vale ressaltar que a **maioria dos SGBDs não separa os três níveis** de forma completa e explícita, mas dá suporte a eles de alguma forma. Além disso, os esquemas são apenas descrições dos dados, os **dados em si estão apenas no nível físico**.

Outro destaque é que os **processos de transformação** de requisições e os resultados **entre os níveis** são chamados de **mapeamentos**. Temos então:

- **Mapeamento externo/conceitual**: define a correspondência entre uma visão particular externa e o esquema conceitual do banco de dados.
- **Mapeamento conceitual/interno**: especifica como a estrutura conceitual é armazenada fisicamente.

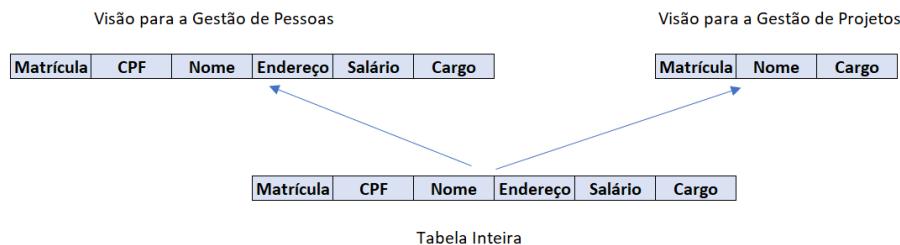
EXEMPLIFICANDO!!!

Para ajudar na compreensão da diferença entre os níveis, vamos pensar em uma situação concreta. Imagine um banco de dados de uma empresa que trabalha com projetos. Esse banco de dados irá armazenar todas as informações úteis para auxiliar na gestão da empresa, incluindo informações sobre os projetos, funcionários e muitas outras. A empresa possui um setor de Gestão de Pessoas e outro de Gestão de Projetos.

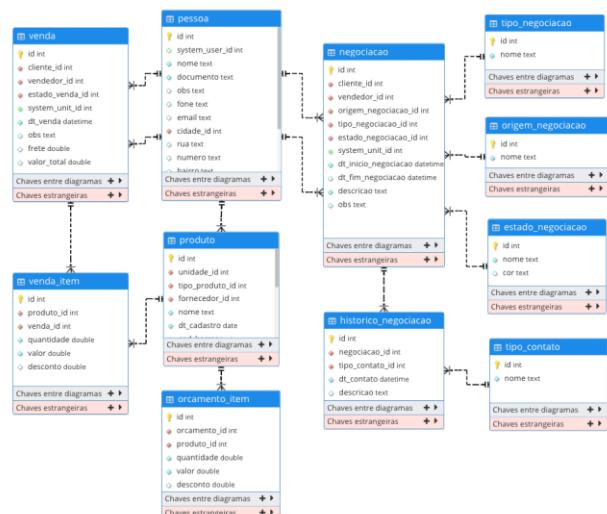
Dito isto, vamos supor que tanto o pessoal da Gestão de Pessoas quanto o pessoal da Gestão de Projetos precisem de informações sobre os funcionários, sendo que a primeira precisa para questões de controle de pessoal e pagamentos e a segunda para alocação nos projetos.

Suponha que tenhamos as seguintes informações sobre os funcionários: matrícula, CPF, nome, endereço, salário e cargo. Nem todas essas informações podem ou precisam ser divulgadas a todos os usuários dos bancos de dados. Por exemplo, o pessoal da Gestão de Projetos não precisa saber o CPF, endereço e nem o salário dos funcionários, bastando saber as matrículas, nomes e cargos para realizar a alocação nos projetos. Como resolver isso?

Para resolver essa situação, basta criar dois esquemas de visão distintos: um para o pessoal da Gestão de Pessoas, que pode apresentar todas as informações dos funcionários; e outro para o pessoal da Gestão de Projetos, apresentando somente a matrícula, nome e cargo. Dessa forma, os dois grupos terão visões diferentes sobre os dados. Esse é o **nível externo ou de visão**. A figura a seguir ilustra essa situação:

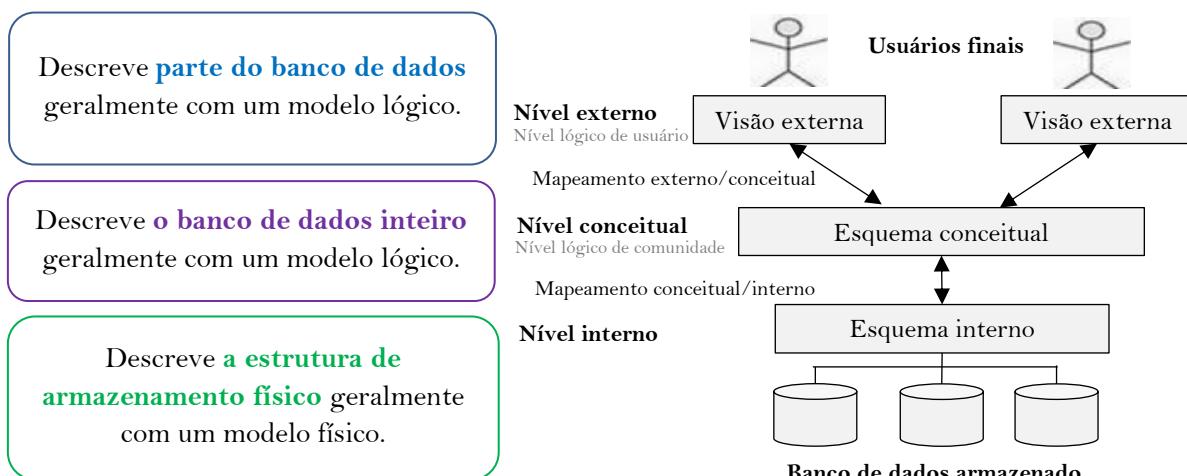


Já quando temos a representação do banco de dados inteiro, isto é, todas as estruturas que compõem o banco de dados, temos o **nível conceitual**. A seguir, um diagrama representando essa situação para um pequeno banco de dados:



No **nível interno**, há o detalhamento das estruturas de armazenamento.

Vamos fixar a arquitetura com um esquema:



Esquema 10 – Arquitetura de 3 esquemas.

18- (CESPE - 2024 – MPE-TO – Analista Ministerial) A respeito de arquitetura de bancos de dados, julgue o item a seguir.

Na arquitetura de três esquemas, o nível externo apresenta uma série de visões do usuário.

Resolução:

Perfeitamente, é o nível externo da arquitetura de 3 esquemas que apresenta as visões do usuário. **No nível externo ou de visão**, cada esquema externo **descreve a parte do banco de dados em que um usuário em particular está interessado e oculta o restante do banco de dados** do grupo de usuários. Os outros dois níveis são o **conceitual**, **descreve a estrutura do banco de dados inteiro para uma comunidade de usuários**. E o **interno**, que **descreve a estrutura do armazenamento físico do banco de dados**.

Gabarito: Certo.

19- (FGV - 2024 – CM-SP – Consultor Técnico Legislativo) Com relação aos níveis da arquitetura ANSI/SPARC dos Sistemas Gerenciadores de Bancos de Dados (SGBD) relacionais, assinale (V) para a afirmativa verdadeira e (F) para a falsa.

I. O nível interno é o mais próximo do meio de armazenamento físico, é uma representação de baixo nível de todo o banco de dados, ele se ocupa do modo como os dados são fisicamente armazenados dentro do SGBD.

II. O nível externo, também conhecido como o nível lógico de comunidade, é o mais próximo dos usuários finais ou programadores de aplicação, é aquele que se ocupa do modo como os dados são vistos pelos usuários do sistema.

III. O nível conceitual, também conhecido nível lógico de usuário, é um nível indireto entre os outros dois níveis e representa todo o conteúdo do banco de dados de uma forma um tanto abstrata em comparação como os dados são armazenados logicamente.

As afirmativas são, respectivamente,

- a) F – V – V.
- b) F – F – V.
- c) F – V – F.
- d) V – V – F.
- e) V – F – F.

Resolução:

Vamos analisar cada um dos itens:

- I. (V) Perfeitamente, é o **nível interno** que se ocupa dos detalhes de armazenamento.
- II. (F) O nível externo, também conhecido como o nível lógico de **comunidade usuário**, é o mais próximo dos usuários finais ou programadores de aplicação, é aquele que se ocupa do modo como os dados são vistos pelos usuários do sistema.
- III. (F) O nível conceitual, também conhecido nível lógico de **usuário comunidade**, é um nível indireto entre os outros dois níveis e representa todo o conteúdo do banco de dados de uma forma um tanto abstrata em comparação como os dados são armazenados logicamente.

No nível conceitual, há a representação do banco de dados inteiro de forma global e integrada, sendo visível a toda a comunidade de usuários. Por isso, falamos em **nível lógico de comunidade**.

Gabarito: Letra E.

20- (CESPE - 2022 – FUB – Técnico) A respeito de arquitetura de bancos de dados, julgue o item a seguir.

O nível de abstração de dados em que é definida detalhadamente a maneira pela qual os dados de um sistema estão estruturalmente armazenados e como podem ser computacionalmente acessados corresponde ao nível de visão do usuário.

Resolução:

O nível que trata de detalhar **como os dados são armazenados** é o **nível interno** e não o nível de visão. O **nível de visão** representa uma parte do banco de dados para um usuário ou grupo de usuários, **ocultando o restante do banco de dados** do grupo de usuários.

Gabarito: Errado.

21- (CESPE - 2017 - TRE-TO - Técnico Judiciário - Programação de Sistemas) A respeito da arquitetura de três esquemas para banco de dados, assinale a opção correta.

- a) Uma das desvantagens da arquitetura de três esquemas é a impossibilidade de aplicar a independência de dados.
- b) Um dos objetivos da arquitetura de três esquemas é aproximar o banco de dados físico das aplicações.
- c) O nível conceitual serve para descrever a estrutura do banco de dados para um conjunto de usuários.
- d) Mapeamentos são as transformações que dados brutos armazenados sofrem para se tornar informações inteligíveis.
- e) O nível interno inclui uma série de visões do usuário utilizadas para descrever partes do banco de dados.

Resolução:

Vamos analisar cada um dos itens:

- a) **Incórrito:** Uma das ~~des~~vantagens da arquitetura de três esquemas é a ~~im~~possibilidade de aplicar a independência de dados.
- b) **Incórrito:** Um dos objetivos da arquitetura de três esquemas é ~~aproximar~~ **separar** o banco de dados físico das aplicações.
- c) **Correto:** O nível conceitual serve para descrever a estrutura do banco de dados para um conjunto de usuários.

Cuidado com este item, pois você poderia ser levado a crer que temos aqui a descrição do nível externo, pois ele fala de conjunto de usuários. Contudo, perceba que o item fala na descrição da estrutura do banco de dados e não de apenas parte dele. O conjunto de usuários pode ser entendido como a comunidade geral que usa esse banco e não apenas um conjunto específico de usuários.

- d) **Incórrito:** Mapeamentos são as transformações ~~que dados brutos armazenados sofrem para se tornar informações inteligíveis entre os níveis da arquitetura de três esquemas, isto é, transformação externo/conceitual ou conceitual/físico.~~
- e) **Incórrito:** O nível **interno** **externo** inclui uma série de visões do usuário utilizadas para descrever partes do banco de dados. **O nível interno descreve as estruturas de armazenamento.**

Gabarito: Letra C.

1.5.4 Independência lógica e física de dados

A arquitetura de três esquemas pode ser usada para explicar melhor o conceito de **independência de dados**, que é a **capacidade de alterar o esquema em um nível do sistema de banco de dados sem ter de alterar o esquema no nível mais alto**.

Temos dois tipos de independência de dados:

- A **independência lógica de dados** é a **capacidade de alterar o esquema conceitual sem ter de alterar os esquemas externos ou de programas de aplicação**. Podemos alterar o esquema conceitual para expandir o banco de dados (acrescentando um tipo de registro), para alterar restrições ou para reduzir o banco de dados (removendo algum tipo de registro). Depois que o esquema conceitual passa por uma reorganização lógica, os programas de aplicação que referenciam as construções do esquema externo devem trabalhar da mesma forma que antes.
- A **independência física de dados** é a **capacidade de alterar o esquema interno sem ter de alterar o esquema conceitual** e, por consequência, sem ter que alterar os esquemas externos. Mudanças no esquema interno podem ser necessárias porque alguns arquivos físicos foram reorganizados para melhorar o desempenho da recuperação ou atualização.

EXEMPLIFICANDO!!!

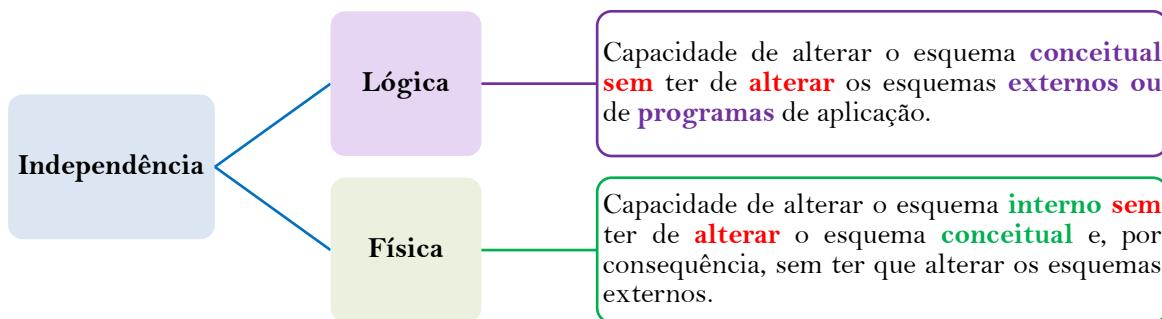
Suponha um sistema de banco de dados de uma biblioteca digital que armazena informações sobre livros, autores e usuários.



Imagine que você decide adicionar um novo tipo de informação ao banco de dados, como a categoria dos livros (por exemplo, ficção, ciência, drama, etc.). O atributo “Categoria” pode ser adicionado na tabela “Livros”, o que representa uma alteração no esquema conceitual. Agora imagine um bibliotecário que usa um software para gerenciar empréstimo de livros. Esse software não precisa ser alterado para continuar funcionando após a adição da coluna “Categoria”, a menos que se queira incluir a funcionalidade de busca por categoria. Temos então um exemplo de **independência lógica de dados**, pois a modificação no esquema conceitual não exigiu mudanças nos programas de aplicação existentes.

Agora imagine que você decide reorganizar os arquivos físicos que armazenam os dados dos livros, mudando para um sistema de armazenamento em nuvem. Logo, a forma como os dados são armazenados (esquema interno) foi modificada. Contudo, a estrutura do banco de dados (tabelas, relacionamentos) permanecerá a mesma e os programas de aplicação irão continuar funcionando sem necessidade de alteração. Essa situação ilustra a **independência física de dados**, pois a alteração no esquema interno não afeta o esquema conceitual nem os programas de aplicação.

Esquematizando os conceitos de independência:



Esquema 11 – Independência lógica e física de dados.

22- (FGV - 2024 - DNIT - Analista Administrativo) A arquitetura de três camadas, tradicional nos SGBDs relacionais, pode ser usada para explicar o conceito de independência de dados.

A esse respeito, avalie se as afirmativas a seguir são verdadeiras (V) ou falsas (F).

() Existem três tipos de independência de dados no SGBDs relacionais: a independência lógica, a semântica e a física.

() A independência lógica dos dados é a capacidade de alterar o esquema conceitual sem ter que alterar esquemas externos ou programas aplicativos. É possível alterar o esquema conceitual para expandir o banco de dados, para alterar restrições ou para reduzir o banco de dados (removendo um tipo de registro ou itens de dados).

() A independência física dos dados é a capacidade de alterar o esquema interno sem ter que alterar o esquema conceitual. No entanto, os esquemas externos também precisam ser alterados.

As afirmativas são, respectivamente,

- a) V – V – F.
- b) F – V – F.
- c) V – F – F.
- d) F – V – V.
- e) F – F – V.

Resolução:

Vamos analisar cada um dos itens:

(F) Existem **três** **dois** tipos de independência de dados no SGBDs relacionais: a independência lógica, a semântica e a física.

(V) Perfeitamente. A **independência lógica de dados** é a **capacidade de alterar o esquema conceitual sem ter de alterar os esquemas externos ou de programas de aplicação**. Podemos alterar o esquema conceitual para expandir o banco de dados (acrescentando um tipo de registro), para alterar restrições ou para reduzir o banco de dados (removendo algum tipo de registro).

(F) A independência física dos dados é a capacidade de alterar o esquema interno sem ter que alterar o esquema conceitual. ~~No entanto, os esquemas externos também precisam ser alterados. Por consequência, não deve ser necessário alterar os esquemas externos.~~

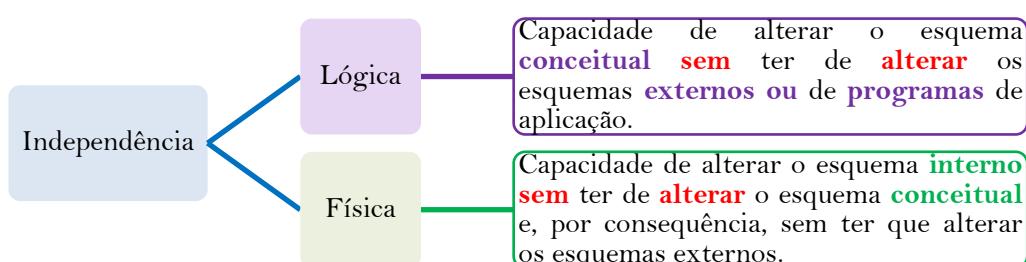
Gabarito: Letra B.

23- (CESPE / CEBRASPE - 2021 - TCE-RJ - Analista de Controle Externo - Especialidade: Tecnologia da Informação) Julgue o item a seguir, acerca dos conceitos de administração de banco de dados.

A independência de dados é caracterizada pelo fato de os dados não dependerem do esquema físico e não precisarem ser reescritos se o esquema físico do banco de dados for alterado.

Resolução:

A **independência de dados** é a capacidade de **alterar um esquema inferior sem necessitar alterar o esquema superior**. Nesse contexto, temos dois tipos:



Gabarito: Certo.

1.5.5 Modelo de dados x arquitetura de três esquemas

Caros alunos, é importante que vocês tenham bastante cuidado para não confundir os modelos de dados com os níveis da arquitetura de três esquemas. É muito fácil confundir, pois além de possuírem a mesma quantidade de níveis, temos inclusive nomes iguais para representar níveis desses modelos.

Tenha as seguintes noções:

- **Modelos de dados** são usados para **representar os conceitos usados para descrever a estrutura** dos bancos de dados.
 - **Modelo conceitual ou de alto nível.**
 - **Modelo lógico, representativo ou de implementação.**
 - **Modelo físico.**

- **Níveis do esquema** representam as **camadas de interação do usuário com o banco de dados**. Esses níveis são inclusive representados por modelos de dados.
 - **Nível externo ou de visão.**
 - **Nível conceitual.**
 - **Nível interno.**

Então, essas classificações representam coisas diferentes. A principal confusão é causada por conta do termo CONCEITUAL que existe tanto na hierarquia de modelos quanto na arquitetura de esquemas. Tenha em mente que o **MODELO CONCEITUAL** é usado para **representar conceitos facilmente entendidos pelo usuário**, enquanto o **ESQUEMA EM NÍVEL CONCEITUAL** é usado para **descrever o banco de dados inteiro**.

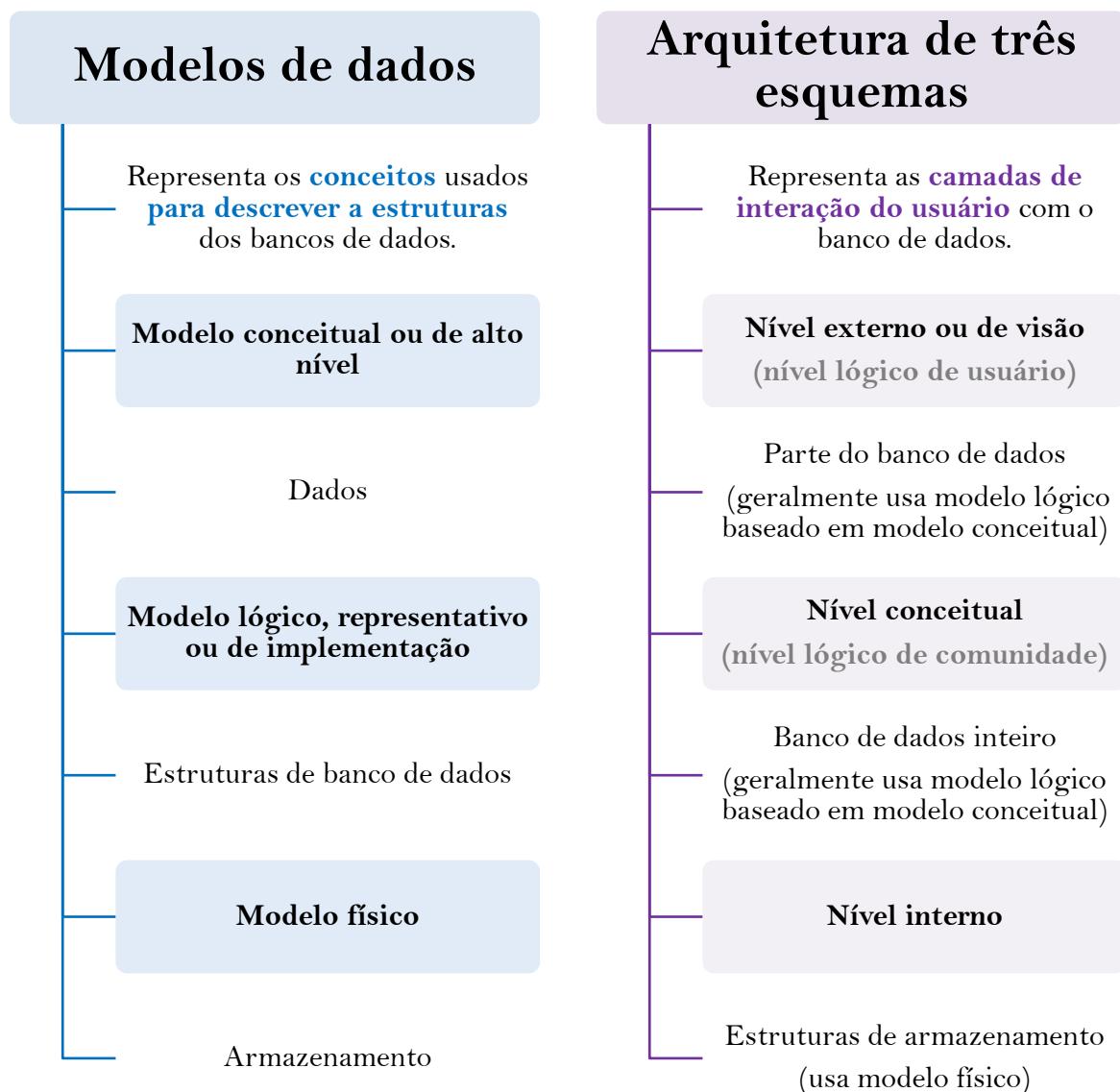
ESCLARECENDO!!!

Como se classifica o banco de dados relacional em relação aos modelos e aos níveis da arquitetura de 3 esquemas?

O **modelo relacional** é classificado como um **modelo lógico, representativo ou de implementação**, pois representa os conceitos utilizados com base em estruturas específicas de bancos de dados dependentes de um modelo específico, as tabelas.

Contudo, em uma arquitetura de três esquemas, o **modelo relacional** pode ser utilizado para representar mais de um esquema. Assim, se for utilizado para descrever apenas parte de um banco de dados, está representando um **esquema externo ou de visão**. Caso seja utilizado para descrever um banco de dados inteiro, então está representando um **esquema conceitual**.

Vejamos um esquema matador para consolidar a diferença entre modelo de dados e arquitetura de três esquemas:



Esquema 12 – Modelos de Dados X Arquitetura de três esquemas.

1.6 Principais modelos de dados (modelos lógicos)

Os SGBDs são classificados conforme o modelo utilizado, sendo principais modelos:

- **O modelo relacional** representa um banco de dados como uma **coleção de tabelas** bidimensionais (linhas x colunas), onde cada tabela pode ser armazenada como um arquivo separado. As tabelas (relações) representam tanto os dados como os relacionamentos entre esses dados.

EXEMPLIFICANDO!!!

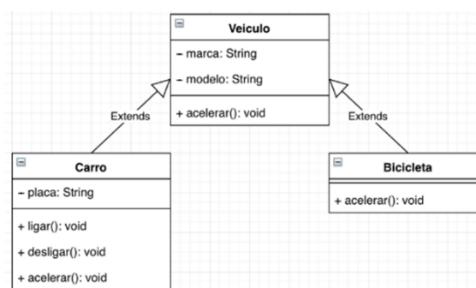
Veja um trecho de uma tabela do modelo relacional com 4 colunas e 2 linhas.

| Código do cliente | Nome do cliente | CPF | Endereço |
|-------------------|----------------------------|----------------|---------------|
| 134512 | João Aprovado dos Santos | 123.123.123-12 | Rua A, nº 123 |
| 251218 | Maria Concursada Fernandes | 456.456.456-45 | Rua B, nº 456 |

- **O modelo de objetos** define um banco de dados em termos de **objetos, suas propriedades e operações**. Objetos com a mesma estrutura e comportamento pertencem a uma **classe**. As operações de cada classe são especificadas com procedimentos predefinidos, chamados **métodos**. As classes podem ser organizadas em **hierarquias (heranças)**, caso em que uma classe filha (**subclasse**) herda as propriedades e métodos da classe pai (**superclasse**). Nesse modelo, há o **encapsulamento**, que é o processo de esconder os detalhes de um objeto ou de restringir o acesso direto a alguns componentes de um objeto. Podemos ainda citar o **polimorfismo**, que é a característica na qual os mesmos atributos ou métodos podem ser utilizados por objetos distintos e com implementações distintas.

EXEMPLIFICANDO!!!

A figura a seguir ilustra um trecho de um modelo de objetos:



Nesse modelo, temos 3 classes em uma **hierarquia de classes (herança)**, sendo “Veiculo” uma **superclasse** “Carro” e “Bicicleta” como suas **subclasses**. Esse modelo está representando gêneros de objetos que compartilham as mesmas características. Por exemplo, “Carro” é uma **classe** para representar todos os carros. Cada carro é específico, mas é um **objeto**. Logo, um carro Hyundai HB20 de placa XXX é um objeto.

As **propriedades** são representadas na parte intermediária da caixinha, em que “marca” e “modelo” como propriedades para “Veiculo” e “placa” para “Carro”. Contudo, pela herança, as subclasses herdam as propriedades e métodos da superclasse e, portanto, tanto “Carro” quanto “Bicicleta” também possuem os atributos “marca” e “modelo”.

CONTINUA...

CONTINUAÇÃO...

Os **métodos ou operações** são representados na parte inferior da caixinha. Assim, “Veiculo” possui um método “acelerar()”, “Bicicleta” possui um método “acelerar()” e “Carro” possui os métodos “ligar()”, “desligar()” e “acelerar()”. Essas são as assinaturas dos métodos. Dentro do método, vai seu corpo que é a implementação, isto é, o código que declara o comportamento que deverá ser realizado.

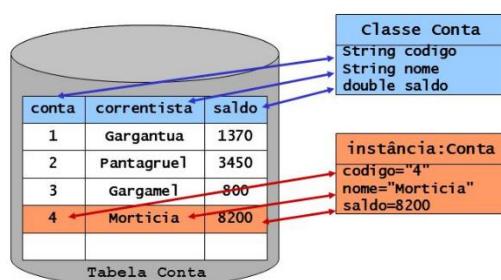
Observe que o método “acelerar()” está presente nas três classes. Isso é um exemplo de **polimorfismo**, pois a implementação desses métodos pode ser diferente em cada classe, ainda que a assinatura do método seja a mesma. A aceleração de um carro é diferente da de uma bicicleta, certo? Então, a aplicação poderá simplesmente invocar o método “acelerar()” e, a depender do objeto que receber a solicitação, ele irá acionar uma implementação diferente: no carro serão acionados todos os procedimentos mecânicos e/ou elétricos necessários; na bicicleta, é força nas pernas.

Só pra fechar o nosso entendimento dos conceitos relacionados ao modelo de objetos, suponha que uma aplicação deseja acessar a placa de um “Carro”. De acordo com o **encapsulamento**, essa aplicação não poderá fazer isso diretamente. Então, deverá existir um método que permita esse acesso (ex.: `obterPlaca()`). Assim, a aplicação irá chamar esse método as invés de simplesmente tentar acessar “placa” diretamente. Por que isso é importante? Porque podem ser feitas verificações e tratamentos nesse método para garantir a consistência das informações e a segurança nos dados.

- No **modelo objeto-relacional**, os **SGBDs relacionais** têm estendido seus modelos para **incorporar conceitos de bancos de dados de objeto** e outras funcionalidades.

EXEMPLIFICANDO!!!

A figura a seguir ilustra um trecho de um modelo objeto-relacional:



Perceba que há tanto uma tabela do modelo relacional (Tabela Conta), quanto uma classe do modelo de objetos (Classe Conta), com o devido mapeamento entre elas.

Nesse caso, as informações da conta são armazenadas na Tabela Conta no banco de dados, mas a manipulação no programa é feita por meio das classes e objetos.

Suponha que eu queria cadastrar uma conta, então eu vou acessar o aplicativo, vou preencher as informações da conta, estas serão manipuladas pelos objetos e, quando, eu finalizar o cadastro, essas informações serão armazenadas no banco de dados. Essa comunicação objeto-tabela é realizada pelo programa com auxílio de frameworks de mapeamento objeto-relacional.

- **O modelo XML** utiliza **estruturas de árvore hierárquicas** e combina conceitos de banco de dados com conceitos dos modelos de representação de documentos. Os dados são representados como elementos; com o uso de tags (etiquetas), os dados podem ser aninhados para criar estruturas hierárquicas complexas.

EXEMPLIFICANDO!!!

A trecho de código a seguir ilustra um arquivo XML:

```

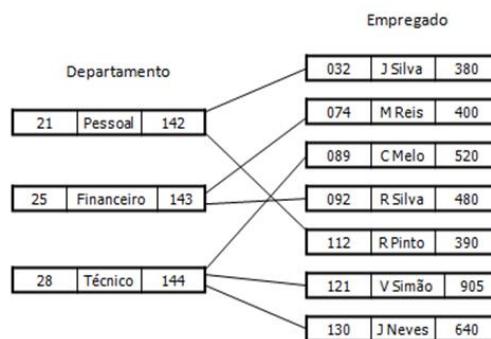
<?xml version="1.0" encoding="ISO-8859-1"?>
<receita nome="pão" tempo_de_preparo="5 minutos" tempo_de_cozimento="1 hora">
  <titulo>Pão simples</titulo>
  <ingredientes>
    <ingrediente quantidade="3" unidade="xícaras">Farinha de Trigo</ingrediente>
    <ingrediente quantidade="7" unidade="gramas">Fermento</ingrediente>
    <ingrediente quantidade="1.5" unidade="xícaras" estado="morna">Água</ingrediente>
    <ingrediente quantidade="1" unidade="colheres de chá">Sal</ingrediente>
  </ingredientes>
  <instrucoes>
    <passo>Misture todos os ingredientes, e dissolva bem.</passo>
    <passo>Cubra com um pano e deixe por uma hora em um local morno.</passo>
    <passo>Misture novamente, coloque numa bandeja e asse num forno.</passo>
  </instrucoes>
</receita>
  
```

Perceba o uso de estruturas hierárquicas com o uso de tags. `<receita>` é a tag raiz e possui como “filhas” as tags `<titulo>`, `<ingredientes>` e `<instrucoes>`. Cada uma dessas tags representa um elemento do modelo.

- **O modelo de rede** é um modelo legado (mais antigo) que representa os **dados como tipos de registro e também representa um tipo limitado de relacionamento 1:N**, chamado de tipo de conjunto. Um relacionamento 1:N (um-para-muitos) relaciona uma instância de um registro a muitas instâncias de registros usando algum mecanismo de ligação com ponteiros nesses modelos.

EXEMPLIFICANDO!!!

A figura a seguir ilustra um modelo em rede, relacionando departamentos aos seus empregados.

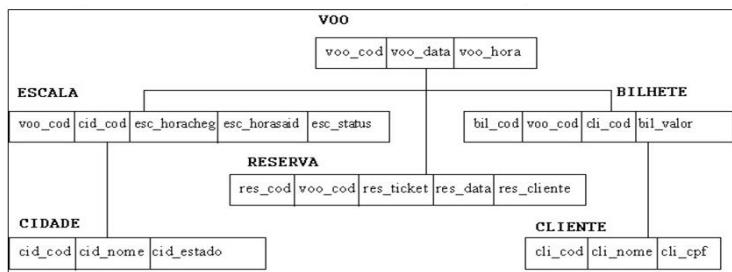


Note que cada “Departamento” pode se relacionar com vários “Empregados”, mas cada “Empregado” só está ligado a um “Departamento”. Por isso, temos uma relação do tipo um-para-muitos (1:N).

- **O modelo hierárquico** representa os dados como **estruturas de árvore hierárquicas**. Cada hierarquia simboliza uma série de registros relacionados em uma estrutura de “pais” e “filhos”. O registro principal é chamado de **raiz (root)**. Cada registro tem **um único “pai”**, mas **pode ter vários “filhos”**. Um registro que não seja “pai” em nenhum relacionamento é denominado “folha” (leaf). **Não existe uma linguagem padrão** para o modelo hierárquico, mas uma linguagem hierárquica popular é a DL/1, que foi um padrão de fato na indústria por muito tempo.

EXEMPLIFICANDO!!!

Veja um modelo hierárquico a seguir:

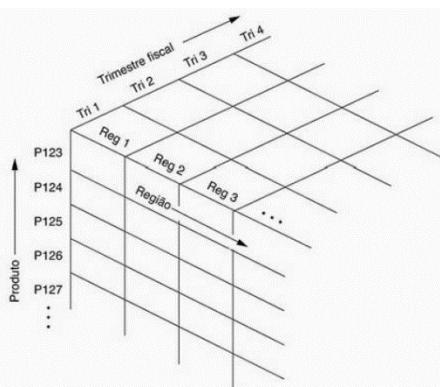


Nesse modelo, temos uma estrutura hierárquica, cujo elemento **raiz (root)** é “VOO”. “CIDADE”, “RESERVA” e “CLIENTE” são **folhas**, pois não são “pais” de nenhum elemento. Note que nenhum elemento é “filho” de dois “pais”, mas um mesmo elemento pode ser “pai” de mais de um “filho”.

- **O modelo dimensional (ou multidimensional)** representa os dados em **matrizes multidimensionais (cubos de dados)** que **combinam tabelas de fatos e de dimensões**. As **tabelas de fatos** armazenam os dados mensuráveis, enquanto as **tabelas de dimensões** fornecem contexto e descrição para as entidades do modelo. Esse modelo é usado em data warehouses e sistemas de apoio à decisão, facilitando a realização de consultas complexas e análises detalhadas.

EXEMPLIFICANDO!!!

Vejamos um exemplo de representação do modelo multidimensional por um cubo de dados tridimensional que organiza os dados de vendas de produtos por trimestres fiscais e por região.



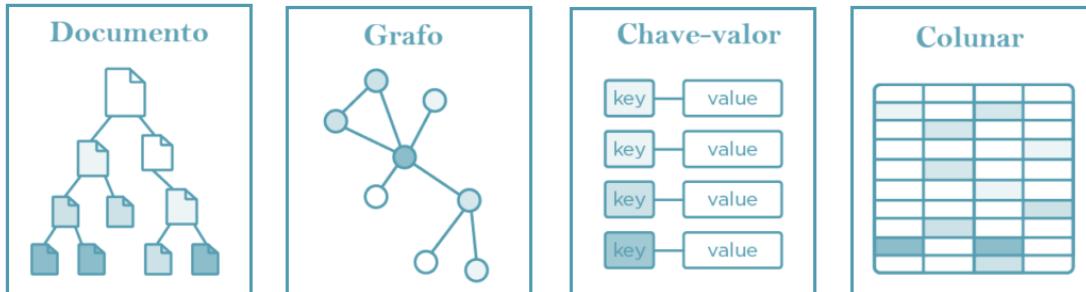
Cada célula neste modelo representa dados de um produto específico, em um trimestre fiscal específico e para uma determinada região.

As consultas neste modelo são realizadas diretamente a partir da combinação das dimensões, por exemplo, podemos consultar o total de vendas de um produto P123 em uma determinada região Reg1.

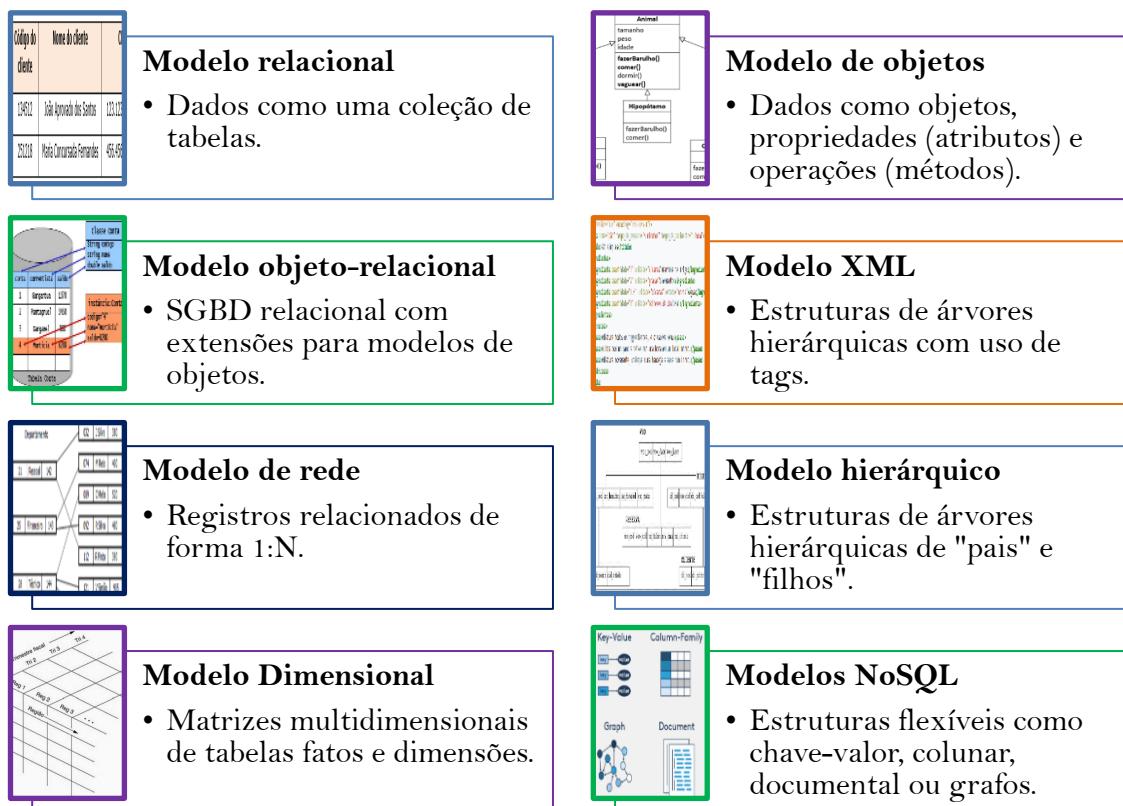
- Os **modelos NoSQL (Not Only SQL)** utilizam **estruturas flexíveis, sem um esquema rígido**, para representar dados. Eles incluem subcategorias como **chave-valor, colunar, documental e grafos**. Criados para gerenciar grandes volumes de dados semiestruturados ou não estruturados, oferecem alta disponibilidade e escalabilidade, tornando-os ideais para aplicações de Big Data.

EXEMPLIFICANDO!!!

As estruturas utilizadas podem ser várias conforme o modelo, mas normalmente são dos quatro tipos ilustrados a seguir:



O esquema a seguir resume os principais modelos de dados:



Esquema 13 – Modelos de dados (modelos lógicos).

24- (CESPE - 2024 – CAU-BR – Analista) No que se refere a banco de dados relacional e orientado a objeto, julgue o item a seguir.

No modelo de dados orientados a objetos, a estratégia de ocultar a implementação interna de uma classe para permitir acesso aos dados dessa classe por meio de interfaces públicas é denominada encapsulamento.

Resolução:

Isso mesmo, em um modelo orientado a objetos, o **encapsulamento** é o processo de **esconder detalhes de uma classe ou objeto ou de restringir o acesso direto**, permitindo o acesso por meio de interfaces.

Considere, por exemplo, uma conta bancária onde o saldo é um dado sensível. Utilizando o encapsulamento, o saldo é declarado como privado e não pode ser acessado diretamente. Em vez disso, são fornecidos métodos públicos para depositar e sacar dinheiro, garantindo que essas operações sejam feitas de forma controlada. Além disso, há um método público que permite visualizar o saldo sem modificá-lo. Isso protege a integridade do saldo e previne acessos e modificações não autorizadas, escondendo os detalhes internos e oferecendo uma interface segura.

Gabarito: Certo.

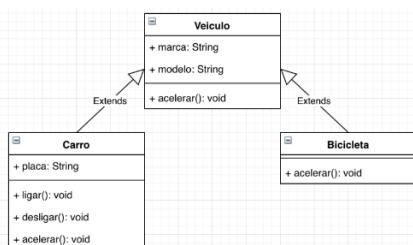
25- (CESPE - 2024 – CAU-BR – Analista) No que se refere a banco de dados relacional e orientado a objeto, julgue o item a seguir.

No modelo de dados orientados a objetos, os dados são geralmente organizados em tabelas com linhas e colunas, o que facilita o seu armazenamento.

Resolução:

Assertiva trata na verdade de um modelo de dados relacional.

O **Modelo de dados de objeto** define um banco de dados em termos de **objetos, suas propriedades e operações**. Objetos com a mesma estrutura e comportamento pertencem a uma **classe**. As operações de cada classe são especificadas com procedimentos predefinidos, chamados **métodos**.



Por sua vez, no **modelo relacional** os dados são **armazenados em tabelas**.

| Código do cliente | Nome do cliente | CPF | Endereço |
|-------------------|----------------------------|----------------|---------------|
| 134512 | João Aprovado dos Santos | 123.123.123-12 | Rua A, nº 123 |
| 251218 | Maria Concursada Fernandes | 456.456.456-45 | Rua B, nº 456 |

Gabarito: Errado.

26- (CESGRANRIO - 2024 - IPEA – Técnico de Planejamento e Pesquisa)
Considere os diferentes paradigmas de modelagem de dados: relacional, orientada a objetos e mapeamento objeto- relacional.

Sobre esses paradigmas, verifica-se que

- a) a modelagem orientada a objetos permite a representação de entidades complexas com atributos e comportamentos, refletindo mais de perto o mundo real.
- b) a modelagem relacional é mais flexível do que a orientada a objetos, pois permite uma representação mais dinâmica dos dados.
- c) o mapeamento objeto-relacional é um método exclusivo da modelagem relacional, permitindo a conversão direta de objetos em tabelas.
- d) o mapeamento objeto-relacional é uma técnica exclusiva da modelagem orientada a objetos, não sendo aplicável a sistemas baseados em modelagem relacional.
- e) os dados, na modelagem orientada a objetos, são representados como tabelas com linhas e colunas, semelhante à abordagem relacional.

Resolução:

Vamos analisar cada um dos itens:

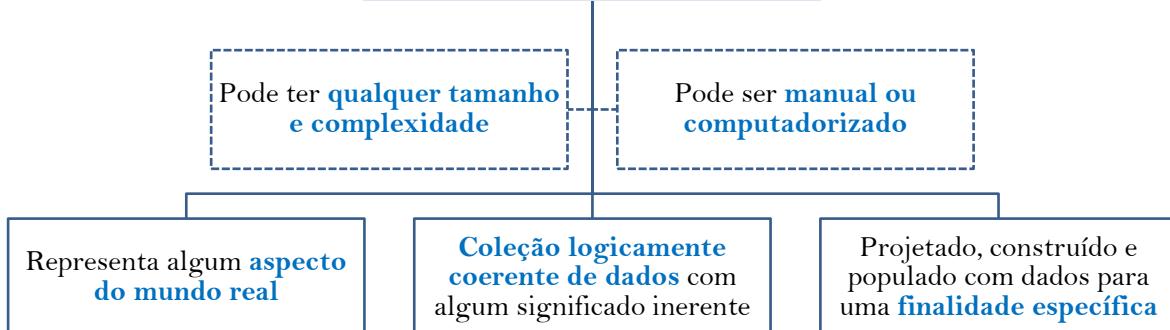
- a) **Correto:** isso mesmo, o mundo pode ser modelado mais facilmente com base em objetos. **O modelo de objetos** define um banco de dados em termos de **objetos, suas propriedades e operações**.
- b) **Incórrito:** a modelagem **relacional orientada a objetos** é mais flexível do que a **orientada a objetos relacional**, pois permite uma representação mais dinâmica dos dados.
- c) **Incórrito:** o mapeamento objeto-relacional é um método **exclusivo da que mescla a** modelagem relacional **com a orientação a objetos**, permitindo a conversão direta de objetos em tabelas.
- d) **Incórrito:** o mapeamento objeto-relacional é uma técnica **exclusiva da** modelagem orientada a objetos, **não sendo aplicável a em conjunto com** sistemas baseados em modelagem relacional.
- e) **Incórrito:** os dados, na modelagem orientada a objetos, são representados como **tabelas com linhas e colunas objetos, semelhante à diferente da** abordagem relacional, **que representa os dados como tabelas**.

Gabarito: Letra A.

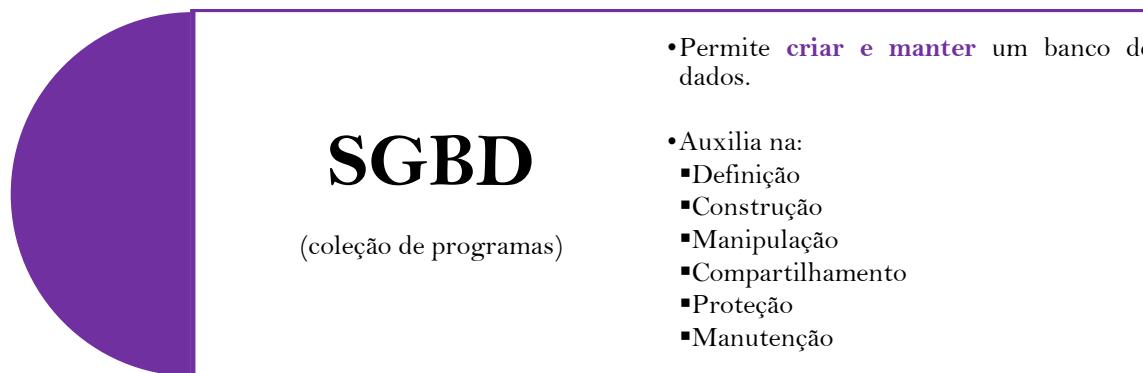
2. ESQUEMAS DE AULA

Banco de Dados (BD)

Banco de Dados (repositório, depósito ou base)



Sistema Gerenciador de Banco de Dados (SGBD)



Sistema de Banco de Dados (SBD)



Características dos Banco de Dados

Elsmari e Navathe

(quatro principais **características** dos bancos de dados)

- Natureza de autodescrição dos dados.
- Isolamento entre programas e dados; abstração de dados.
- Suporte a múltiplas visões.
- Compartilhamento de dados e processamento de transação multiusuário.

Elsmari e Navathe

(**vantagens** de usar a abordagem **SGBD**)

- Controle de redundância.
- Restrição de acesso não autorizado.
- Armazenamento persistente para objetos do programa.
- Estruturas de armazenamento e técnicas de pesquisa para o processamento eficiente de consulta.
- Backup e recuperação.
- Múltiplas interfaces do usuário.
- Representação de relacionamentos complexos entre dados.
- Restrições de integridade.
- Dedução e ação usando regras.
- Potencial para garantir padrões.
- Tempo reduzido para o desenvolvimento de aplicações.
- Flexibilidade.
- Disponibilidade de informações atualizadas.
- Economias de escala.

Date

(**benefícios** da abordagem de **BD**)

- O dado pode ser compartilhado.
- A redundância pode ser reduzida.
- Inconsistências podem ser evitadas.
- Pode-se utilizar o suporte a transações.
- A integridade pode ser mantida.
- A segurança pode ser aperfeiçoada.
- Requisitos conflitantes podem ser balanceados.
- Padrões podem ser utilizados.

Sylberchatz, Korth e Sudarshan

(**desvantagens** de usar **sistema de arquivo**)

- Redundância e inconsistência de dados.
- Dificuldade de acesso a dados.
- Isolamento dos dados.
- Problemas de integridade.
- Problemas de atomicidade.
- Anomalias de acesso concorrente.
- Problemas de segurança.

Desvantagens da abordagem de SGBD

Custos adicionais

Alto investimento inicial
em hardware, software e treinamento

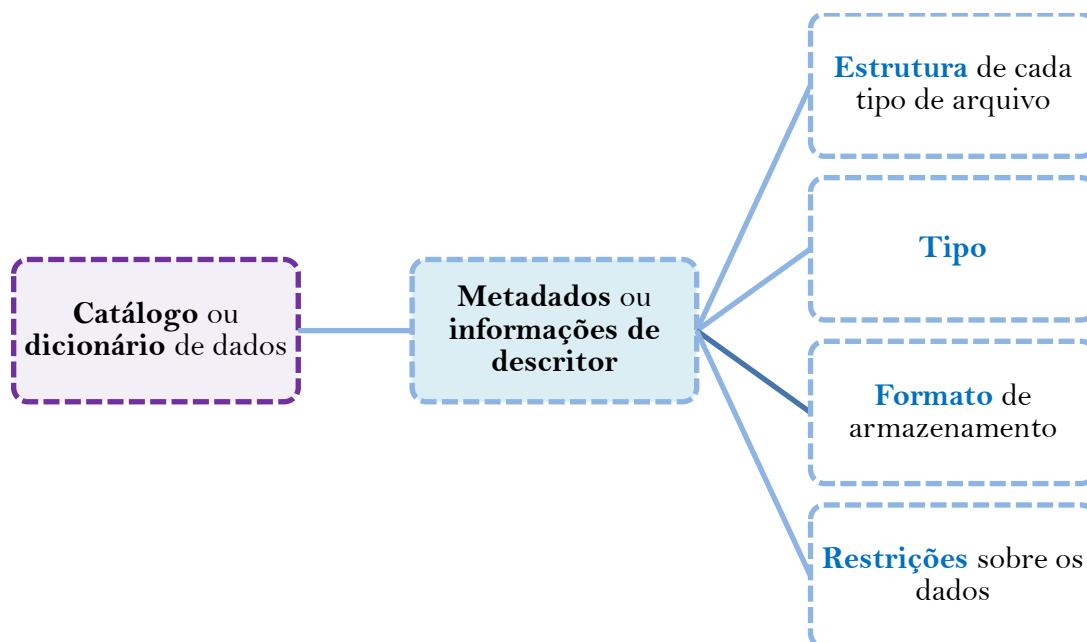
Generalidade para definição e processamento de dados

Esforço adicional para funções de segurança, concorrência, recuperação e integridade

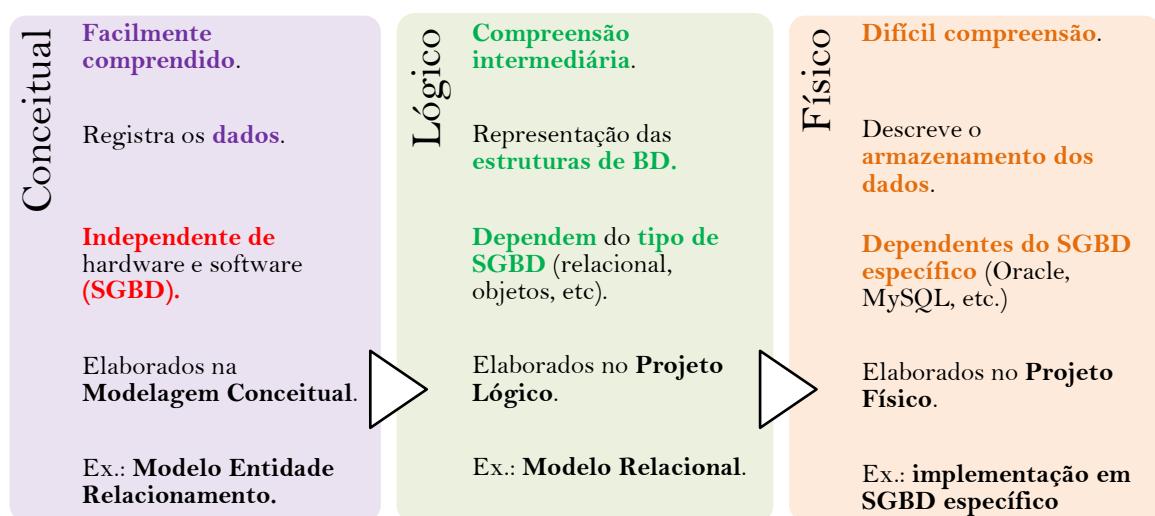
Propriedades das transações

| Atomicidade | Consistência | Isolamento | Durabilidade |
|---|---|--|--|
| <p>Conceito: unidade de processamento atômica.</p> <p>Transação deve ser realizada em sua totalidade ou não deve ser realizada de forma alguma.</p> | <p>Conceito: transação deve levar o banco de um estado consistente para outro.</p> <p>Regras e restrições respeitadas.</p> | <p>Conceito: transação deve parecer executar isolada das demais.</p> <p>Não interferência entre transações paralelas.</p> | <p>Conceito: mudanças realizadas devem ser persistidas no banco de dados.</p> <p>Mudanças não devem ser perdidas por falhas posteriores.</p> |

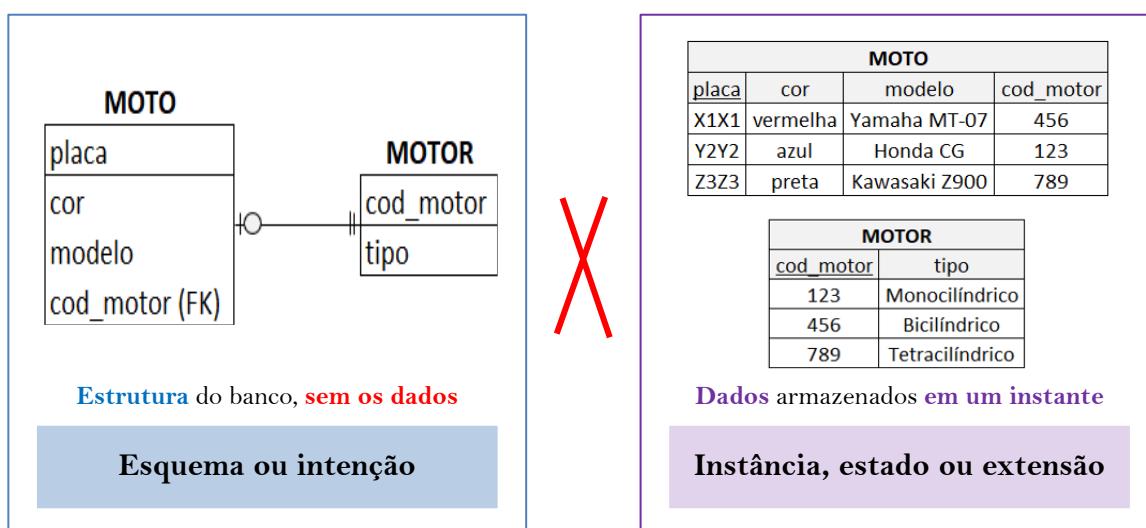
Catálogo ou dicionário de dados



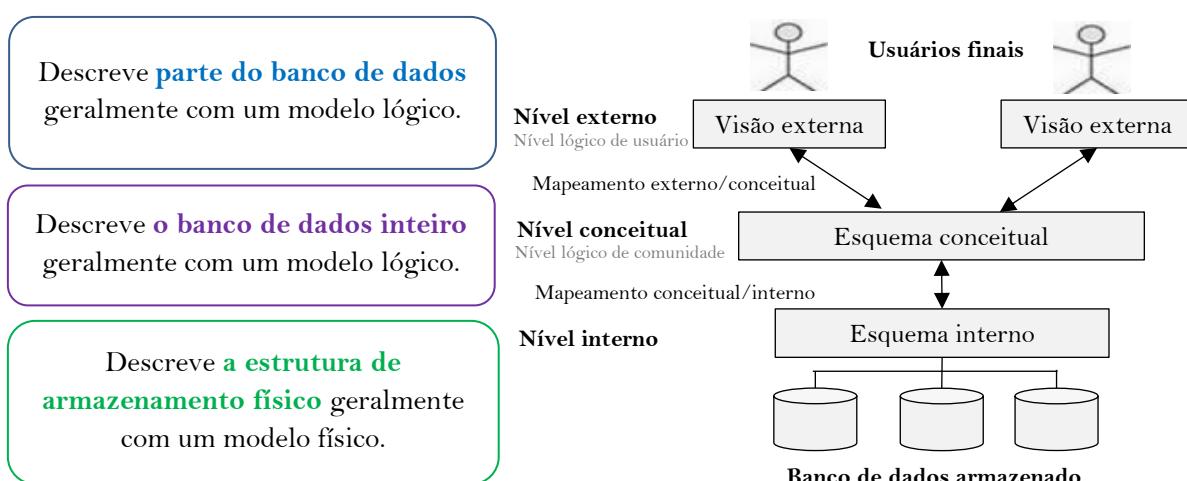
Modelo de bancos de dados

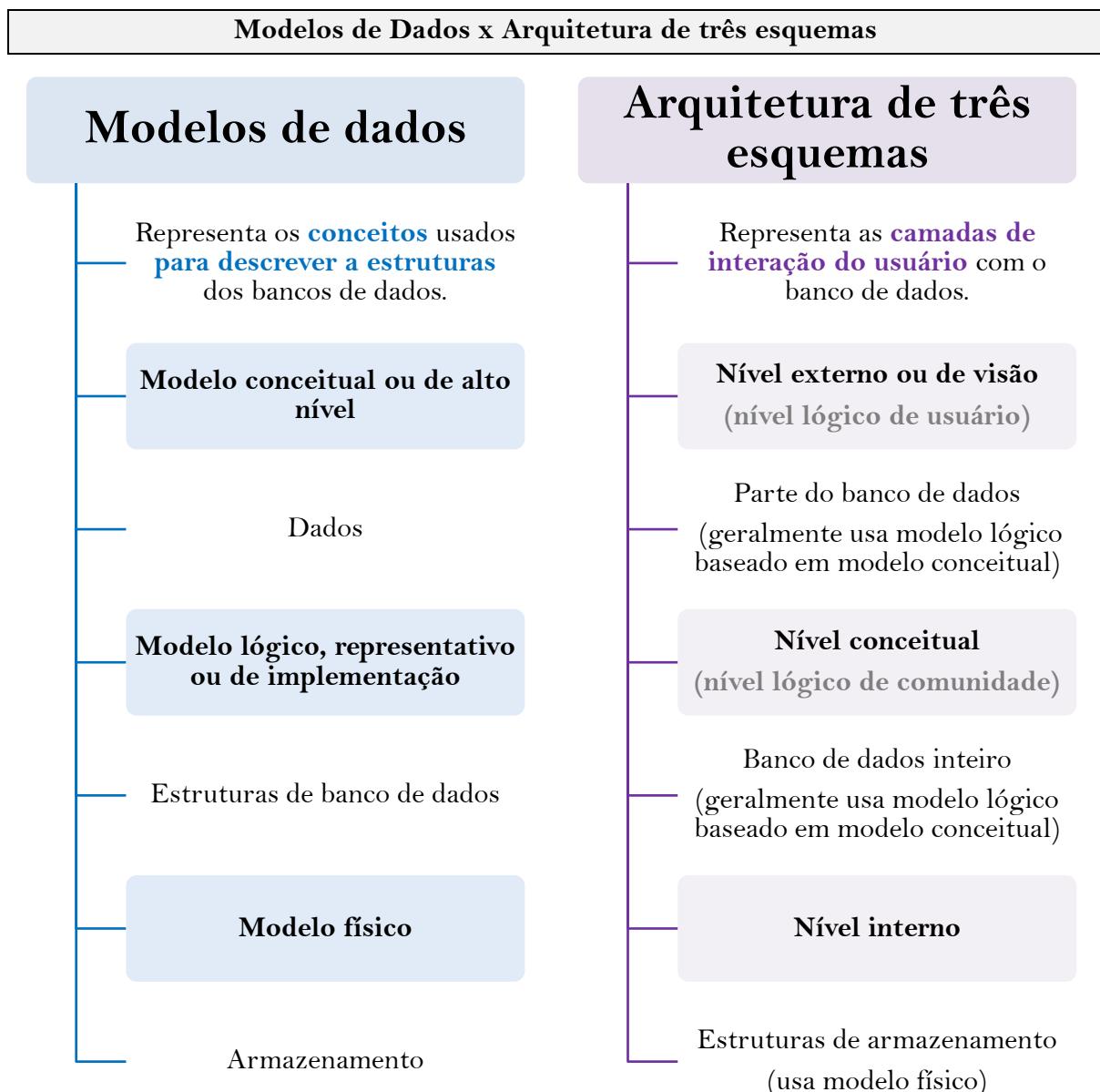
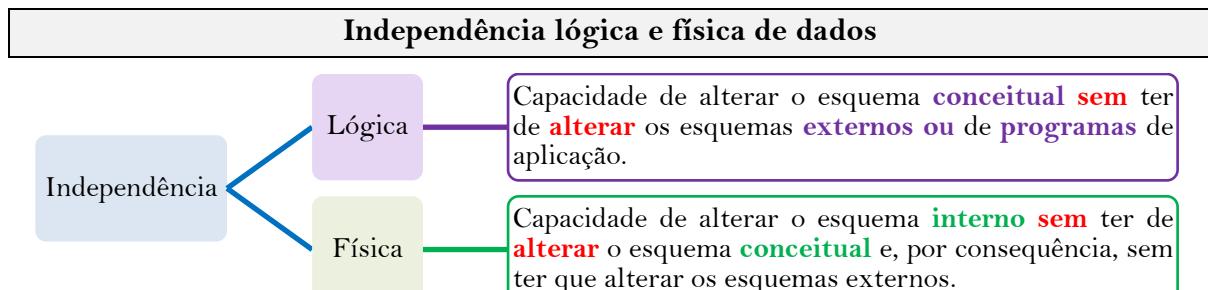


Esquema x Instância



Arquitetura de 3 esquemas



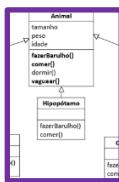


Modelos de dados (modelos lógicos)



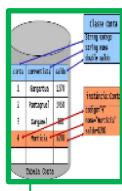
Modelo relacional

- Dados como uma coleção de tabelas.



Modelo de objetos

- Dados como objetos, propriedades (atributos) e operações (métodos).



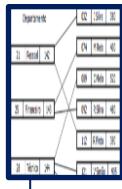
Modelo objeto-relacional

- SGBD relacional com extensões para modelos de objetos.



Modelo XML

- Estruturas de árvores hierárquicas com uso de tags.



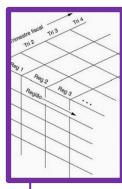
Modelo de rede

- Registros relacionados de forma 1:N.



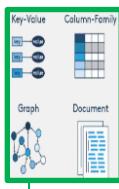
Modelo hierárquico

- Estruturas de árvores hierárquicas de "pais" e "filhos".



Modelo Dimensional

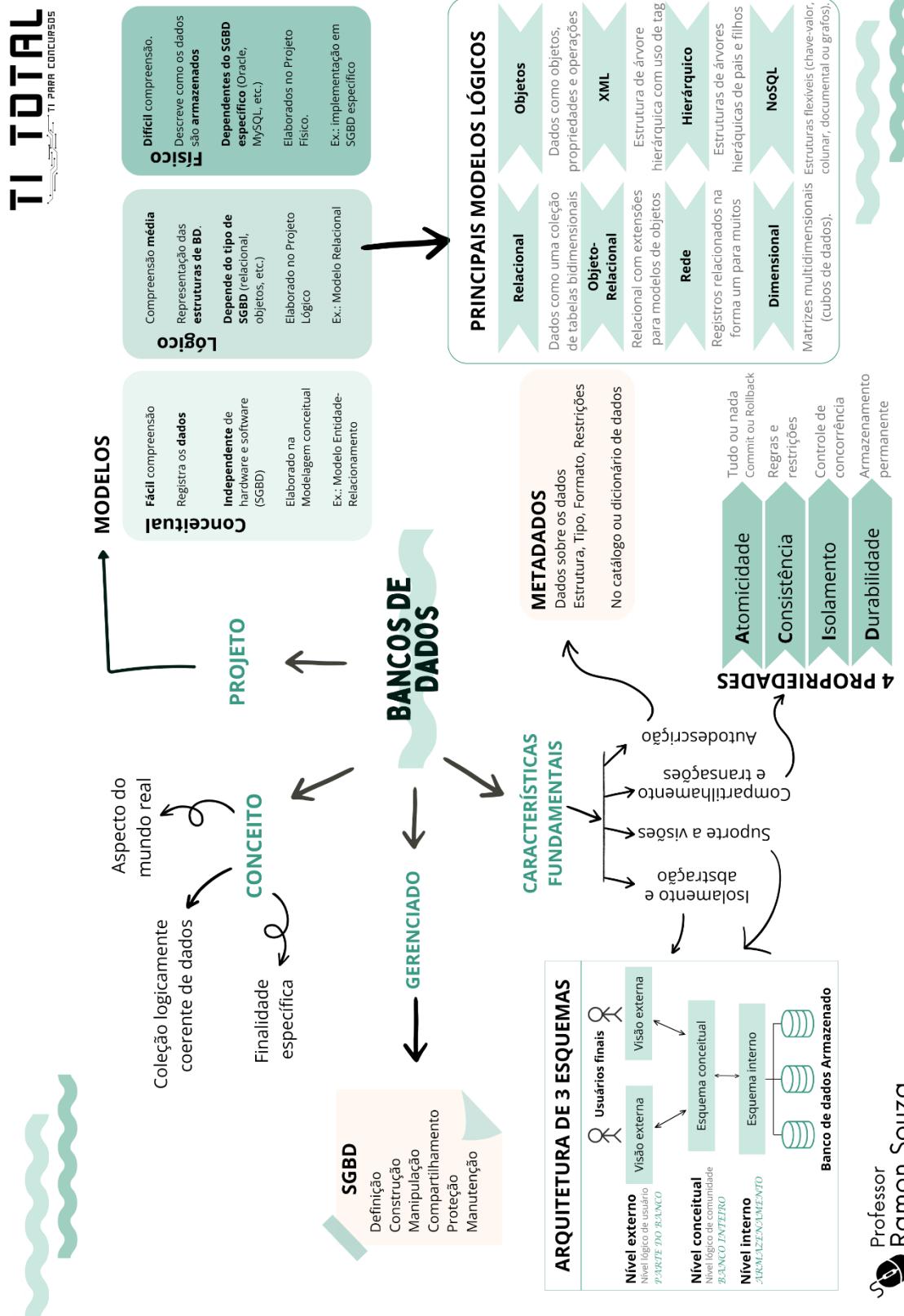
- Matrizes multidimensionais de tabelas fatos e dimensões.



Modelos NoSQL

- Estruturas flexíveis como chave-valor, colunar, documental ou grafos.

3. MAPA MENTAL



4. REFERÊNCIAS

DATE, Christopher J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Elsevier Brasil, 2003.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistema de Banco de Dados**. 7ed. São Paulo: Pearson Addison Wesley, 2019.

MICROSOFT. **Níveis de isolamento da transação (ODBC)**. Disponível em: <<https://learn.microsoft.com/pt-br/sql/odbc/reference/develop-app/transaction-isolation-levels?view=sql-server-ver16>>. Acesso em: 24 jul. 2024.

SYLBERCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Database System Concepts**. 6th. New York: McGraw-Hill, 2011.