

IN DUS TRIAL

Apren
diza
gem



Fundamentos de Banco de Dados

Professor: Hermano Roepke



Contextualização



Lista telefônica



Lista de contatos



Arquivo



Ficheiro

Contextualização



Há 30 anos....

o termo era
“processamento de
dados”, ou seja, o uso
de computadores num
contexto de negócios.



Processamento de
grandes quantidades de
dados.

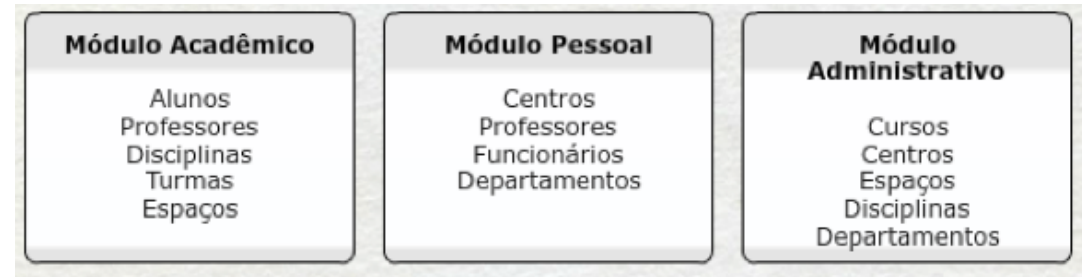
Hoje...

O termo “processamento de
dados” deu lugar a
“tecnologia de informação”.



Extração de informações
relevantes desses dados
processados.

Contextualização



Nosso objetivo não é discutir a evolução dos sistemas de informações propriamente dita. Nosso foco está no armazenamento dos dados. Imagine o cenário informatizado de uma instituição de ensino superior, sem o uso de banco de dados.

Nesse exemplo, podemos identificar que os módulos manipulam dados que poderiam ser compartilhados, porém foram projetados para atender a uma aplicação específica. Como consequência, temos problemas de redundância nos dados, de integridade e de segurança entre muitos outros.

Contextualização



Considerando o mesmo cenário, veja como atualmente são construídos os sistemas tendo como recursos para armazenamento os “bancos de dados”:

Módulo Acadêmico

Módulo Pessoal

Módulo Administrativo



Banco de Dados

Alunos
Turmas
Cursos

Centros
Espaços
Disciplinas

Professores
Funcionários
Departamento

Contextualização



Você já parou para pensar por que “banco de dados”, e não “banco de informações”?





Contextualização

Por vezes podemos entender um **dado** como sendo uma **informação** e uma **informação** como sendo um **dado**. Por exemplo ao responder à pergunta “que dia é hoje?”, sua resposta seria um **dado** ou uma **informação**? Ou seja, em alguns cenários e/ou contextos **dado** e **informação** dão o mesmo significado ao fato em si. Mas no contexto da Tecnologia a informação e da Comunicação eles são termos distintos.



Contextualização

Dado?

É o elemento de partida que serve de base para o tratamento sobre o qual o computador efetua as operações necessárias à tarefa a que foi programado. Podemos dizer que o dado é a representação primária de um fato, conceito ou instrução. Portanto, não representa uma informação ao indivíduo.

Eles podem ser representados de várias formas, como números, letras, imagens, etc. No contexto de bancos de dados, os dados são a matéria-prima que constitui a base de informações mais significativas. Por exemplo, um número "25" por si só é um dado.



Contextualização

Informação?

É um conjunto de dados processados e organizados de tal forma que representam um significado de um fato ou conceito. Pode-se dizer que é uma abstração informal, isto é, que não pode ser formalizada através de uma teoria lógica ou matemática.

Por exemplo, se o número "25" for associado a "idade" e referir-se à idade de uma pessoa, ele se torna uma informação relevante.



Contextualização

Conhecimento?

O conhecimento é um nível mais alto de compreensão e contexto que é derivado da análise e interpretação das informações. Envolve a aplicação de raciocínio, experiência e compreensão para tirar conclusões ou tomar decisões. No contexto de bancos de dados, o conhecimento pode ser extraído da análise de informações em conjunto com outros fatores.

Por exemplo, se a idade "25" estiver associada a uma faixa demográfica específica e essa faixa estiver relacionada a um aumento nas vendas de determinado produto, isso pode levar a um conhecimento útil para planejar estratégias de marketing.

Fundamentos de banco de dados



Mas afinal o que é um banco de dados?





Definição

Banco de dados é um conjunto de dados organizados que representam informações sobre um domínio específico.

Em bancos de dados, a transformação de dados em informações e, eventualmente, em conhecimento é um processo fundamental para a tomada de decisões informadas e a geração de insights significativos.



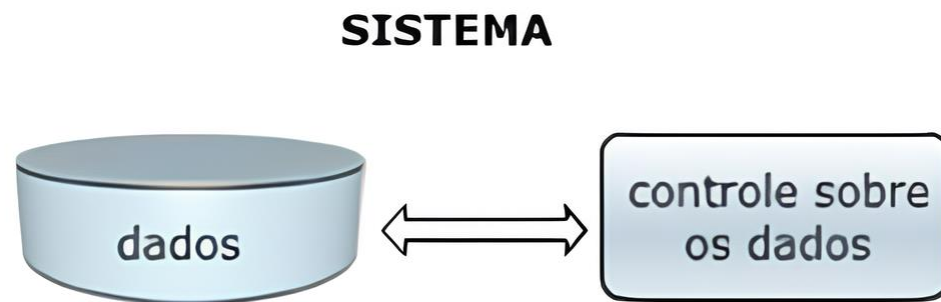
Abordagens de bancos de dados

Podemos classificar em três as diferentes abordagens de armazenamento e gerenciamento de dados. Cada uma delas tem suas particularidades, mas todas têm em comum a necessidade cada vez maior de manter os dados e controlar as operações sobre eles.

1ª Abordagem



A 1ª abordagem das estruturas de bancos de dados é marcada pelo armazenamento dos dados totalmente dependente do programa ou aplicação que os controla. Vejamos uma representação gráfica desse cenário:



Podemos observar que o sistema é responsável pelo armazenamento e controle sobre os dados, ou seja, os dados são acessíveis somente por esse sistema ou aplicação.



1ª Abordagem / exemplos

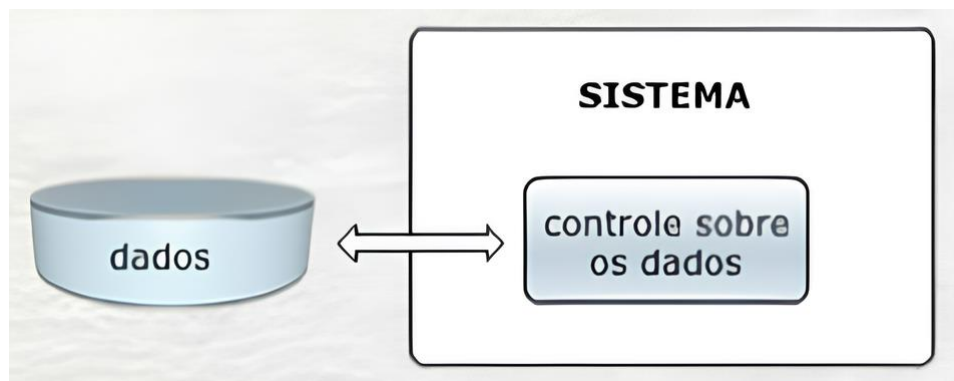
Imagine que você está construindo uma aplicação utilizando a linguagem de programação C para controlar a agenda de contatos de seus amigos. Será necessário armazenar esses dados em um arquivo. Então você vai criar uma estrutura de registro (**struct**) e associá-la a um arquivo. Este arquivo binário que será criado vai armazenar os registros definidos pela sua estrutura.

Agora imagine um aplicativo que você está construindo em Java. Ele vai controlar suas despesas ao longo de um determinado período. Você poderá, por exemplo, definir uma classe “despesa” com respectivos atributos que julgar necessários para o seu propósito. Utilizando uma interface (`java.io.Serializable`), você poderá salvar uma instância dessa classe, ou seja, um objeto em um arquivo.

2ª Abordagem



Já na 2ª Abordagem das estruturas de bancos de dados, temos os dados armazenados de forma independente das estruturas que os controlam. Vejamos a representação gráfica do cenário:



Podemos observar que o sistema é responsável pelo controle sobre os dados. Já a estrutura de armazenamento é independente do sistema, ou seja, os dados podem ser acessados por outros sistemas ou aplicações.



2ª Abordagem / exemplos

Você já deve ter ouvido falar nas estruturas padrão DBF (*Data Base File*) ou simplesmente Dbase, não é? Pois bem, estruturas marcaram esta geração em termos de armazenamento.

A organização dos dados é simples e pode ser acessada por diferentes ambientes de desenvolvimento, assim como aplicações de controle e manutenção.

Para se ter uma ideia, aplicativos que mantêm planilhas eletrônicas permitem o armazenamento e recuperação de dados nessas estruturas.



2ª Abordagem / exemplos

Outro exemplo usual nos dias atuais se refere à persistência de dados em estrutura de arquivo padrão XML (*Extensible Markup Language*).

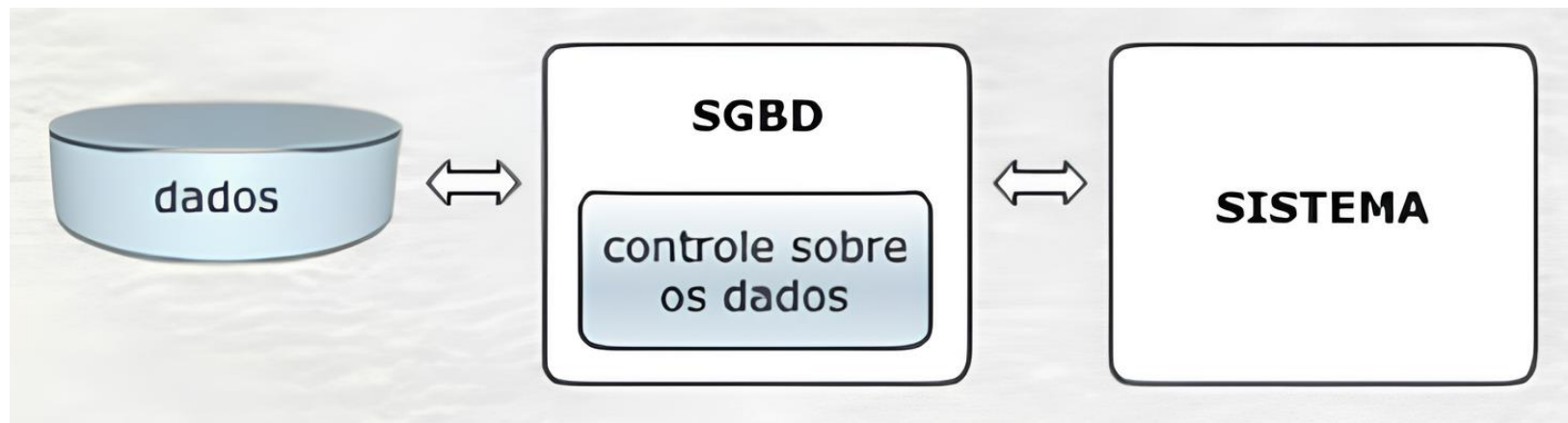
O padrão XML é um formato para a criação de documentos com dados organizados de forma hierárquica, o que geralmente contempla as necessidades das aplicações.

Os arquivos XML podem ser acessados por inúmeras linguagens de programação, bem como por aplicativos de leitura e manipulação.

3ª Abordagem



A 3ª Abordagem das estruturas de bancos de dados é atualmente mais utilizada. Nela temos a inserção do Sistema Gerenciador de Bancos de Dados (SGBD). O SGBD é o responsável pelo acesso e manutenção dos dados armazenados, ou seja, o sistema se comunica com o SGBD e este interage com os dados. Vejamos a representação gráfica desse cenário:





3ª Abordagem / exemplos

ORACLE? Microsoft SQL Server? PostgreSQL? MySQL? MariaDB? FireBird? Estes são alguns exemplos de SGBDs, todos com muitas características em comum. Sua estrutura é complexa, mas seu acesso é simplificado por meio das interações de comunicação que proporcionam o acesso por diferentes ambientes de desenvolvimento, assim como aplicações de controle e manutenção.

É raro você encontrar um ambiente de desenvolvimento ou linguagem de programação de sistemas que não ofereça recursos de acesso e manutenção de estruturas de dados mantidos nos principais SGBDs do mercado. Os SGBDs compreendem um componente fundamental na arquitetura de desenvolvimento de aplicações em camadas.



Bancos de dados NoSQL

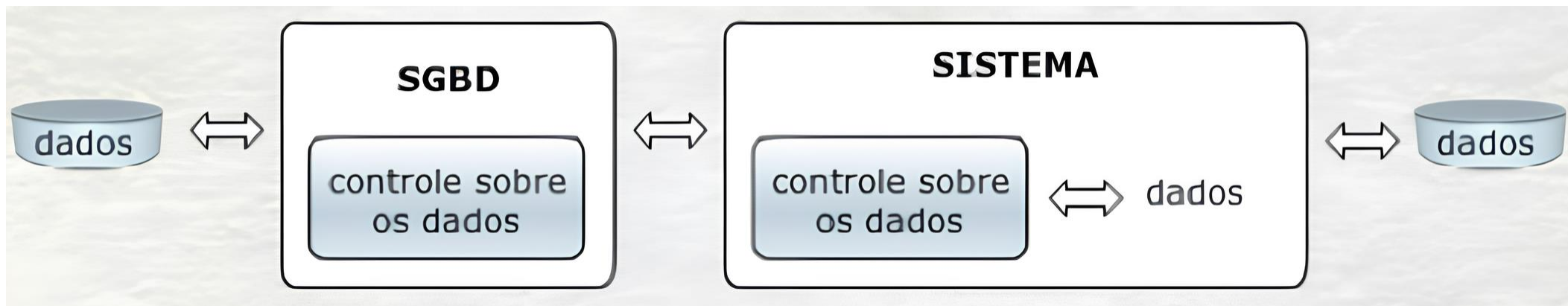
Na terceira geração, com a proliferação de aplicativos da web, redes sociais e big data, os bancos de dados NoSQL surgiram para abordar casos de uso específicos que os sistemas relacionais não podiam tratar de maneira eficaz. Esses bancos de dados adotaram abordagens não estruturadas ou semiestruturadas, como documentos, colunas largas, grafos e chave-valor, para lidar com escalabilidade, flexibilidade e velocidade.

Exemplos de bancos de dados NoSQL incluem MongoDB, Cassandra, Redis, Couchbase e Neo4j. Cada um desses bancos de dados é projetado para atender a diferentes necessidades e cenários de aplicativos.

4ª Abordagem



Você já deve ter ouvido falar que cada vez mais, que nos dias atuais, os sistemas estão mais flexíveis, conectando tudo a todos. Pois bem, isso tem exigido que os sistemas e aplicativos façam uso de estruturas de armazenamento híbridas. Vejamos uma representação gráfica desse cenário:





Atividade

Após conhecer as abordagens das estruturas de bancos de dados e o gerenciamento sobre os dados, qual sua opinião sobre o possível surgimento da 4ª abordagem de bancos de dados?

