

# Banco de Dados

Jordana S. Salamon

[jssalamon@inf.ufes.br](mailto:jssalamon@inf.ufes.br)

[jordanasalamon@gmail.com](mailto:jordanasalamon@gmail.com)

DEPARTAMENTO DE INFORMÁTICA

CENTRO TECNOLÓGICO

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

# Introdução

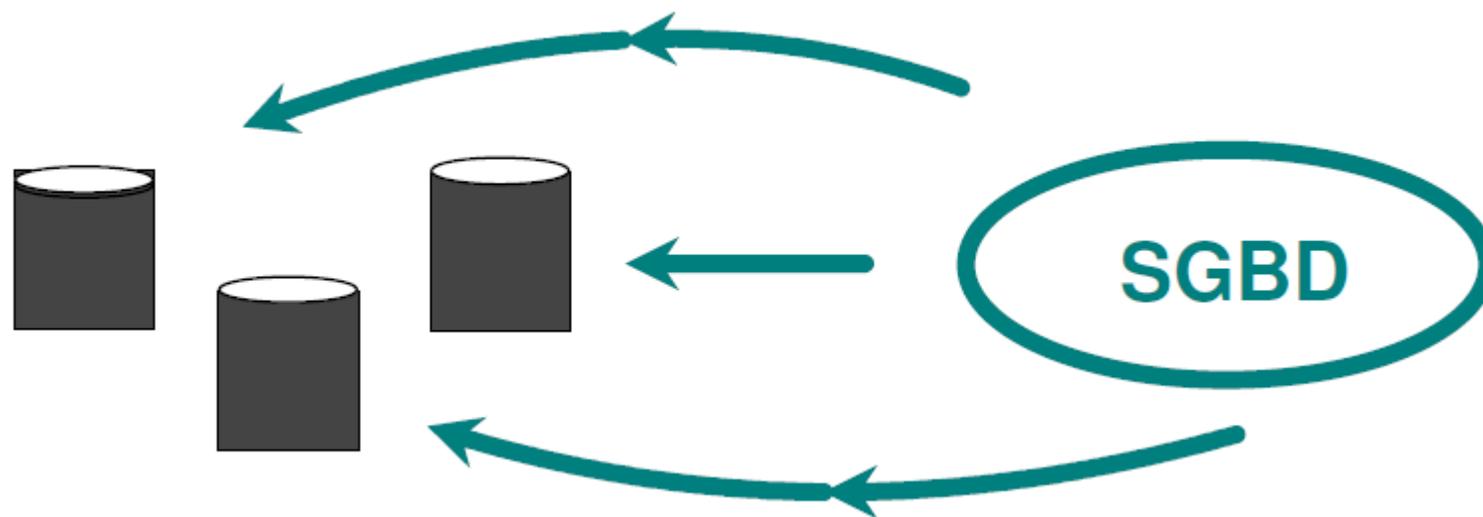
- ▶ Bancos de Dados: Essenciais em todas os ramos de negócios;
  - ▶ Aplicações Tradicionais: dados textuais e/ou numéricos;
  - ▶ “Novas” Aplicações: BD multimídia, BD de Informações Geográficas, Data Warehouses, BD ativos, BD temporais, BD móveis, BD Web ...
- ▶ A tecnologia de Banco de Dados se desenvolveu ao longo de várias décadas;

# Introdução

- ▶ O sucesso dos bancos de dados nos dias de hoje deve-se aos recursos incorporados em um software especializado denominado Sistema de Gerência de Bancos de Dados.
- ▶ Um SGBD é uma ferramenta complexa que tem por objetivo criar e gerenciar grandes quantidades de dados, de forma eficiente, permitindo que estes dados persistam por longo tempo com segurança.

# Introdução

- ▶ Um SGBD é uma coleção de programas que permitem aos usuários definirem, construírem e manipularem uma base de dados para o uso das diversas aplicações.



# Evolução das Aplicações ao Longo dos Anos

- ▶ Inicialmente, baseadas em Sistemas de Arquivos;
- ▶ Duplicação de dados (redundância);
- ▶ Incoerência de uma mesma informação armazenada (inconsistência);
- ▶ Aumento do grau de complexidade na construção e manutenção das aplicações;
- ▶ Duplicação de tarefas feitas para cada aplicação;
- ▶ Aumento do custo de manutenção.



# Sistemas de Arquivos

## ▶ Principais Problemas:

- ▶ Cada aplicação tem seus próprios arquivos;
- ▶ Dados repetitivos, causando informações inconsistentes;
- ▶ Subordinação de programas a arquivos;
- ▶ Falta de integridade dos dados;
- ▶ Dificuldade no desenvolvimento de novos sistemas;
- ▶ Analista “DONO” dos sistemas;
- ▶ Manutenção difícil e cara;
- ▶ Quase não existe segurança dos dados.



# Sistemas de Arquivos Integrados

- ▶ Principais Problemas:
  - ▶ Pouca maleabilidade para mudanças;
  - ▶ Alterações comprometem vários sistemas;
  - ▶ Tornam-se estáticos no tempo;
  - ▶ Problemas que surgem em um sistema podem paralisar um conjunto de procedimentos;
  - ▶ Continua a subordinação de programas a arquivos;
  - ▶ Frágil “Visão do Usuário”;
  - ▶ Integridade e segurança de dados ainda difícil de adotar.

# Sistemas de Bancos de Dados

- ▶ Vantagens:
  - ▶ Flexibiliza o desenvolvimento de sistemas;
  - ▶ Integra dados a nível empresarial;
  - ▶ Atende às exigências de informações de todas as aplicações;
  - ▶ Transparência de dados;
  - ▶ Utiliza “Visões do Usuário”;
  - ▶ Viabiliza a integridade de dados;
  - ▶ Facilita a adoção de segurança.



# Abordagem de Banco de Dados

## ▶ Exigências:

- ▶ Requer um Sistema de Gerência de Banco de Dados (SGBD).
- ▶ Evidencia a necessidade de possuir um bom dicionário de dados.

## ▶ Necessita:

- ▶ Boa Gerência da Definição de Dados (Administração de Dados)
- ▶ Boa Gerência do Projeto de Tabelas (criação de dados) (Administração de Banco de Dados )



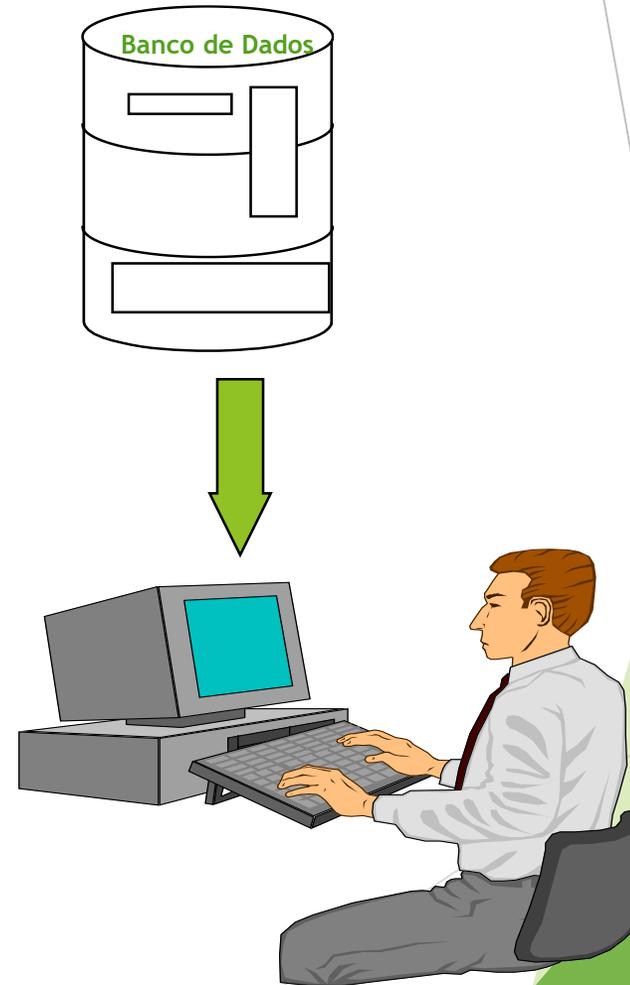
# Sistema de Gerência de Banco de Dados (SGBD)

- ▶ Sistema de Gerência de Banco de Dados (SGBD) ou Database Management System (DBMS) é um sistema (software) com a finalidade de prover um ambiente que seja *adequado* e *eficiente* para armazenar e recuperar informações de bancos de dados.
- ▶ Permite a inclusão, alteração, exclusão, consulta e manipulação dos dados e do esquema, através de linguagens de alto nível e de mecanismos de controle de acesso.
- ▶ Neste ambiente, o usuário diz o *que fazer* e não *como* fazer.



# Sistema de Gerência de Banco de Dados - SGBD

- ▶ Objetivo:
  - ▶ Armazenar, manter e recuperar informações, de forma adequada e eficiente.
- ▶ Componentes:
  - ▶ Dados
  - ▶ Hardware
  - ▶ Software
  - ▶ Usuários



# Sistema de Gerência de Banco de Dados (SGBD)

- ▶ Dados armazenados em um ou mais Bancos de Dados.
- ▶ Os Bancos de Dados permitem *Integração e Compartilhamento*.
  - ▶ *Integração*: eliminação parcial ou total de redundância.
  - ▶ *Compartilhamento*: dados podem ser compartilhados entre diversos usuários de forma concorrente.



# O Que é Um Banco de Dados?

- ▶ Um Banco de Dados é uma coleção de *Dados Operacionais* armazenados, usados pelos sistemas de aplicações de uma empresa.
- ▶ **Exclui-se:**
  - ▶ Dados de Entrada;
  - ▶ Dados de Saída;
  - ▶ Dados transientes.
- ▶ É um acervo de informações segundo certo critério de organização.



# Por Que Banco de Dados?

## ▶ **Objetivos:**

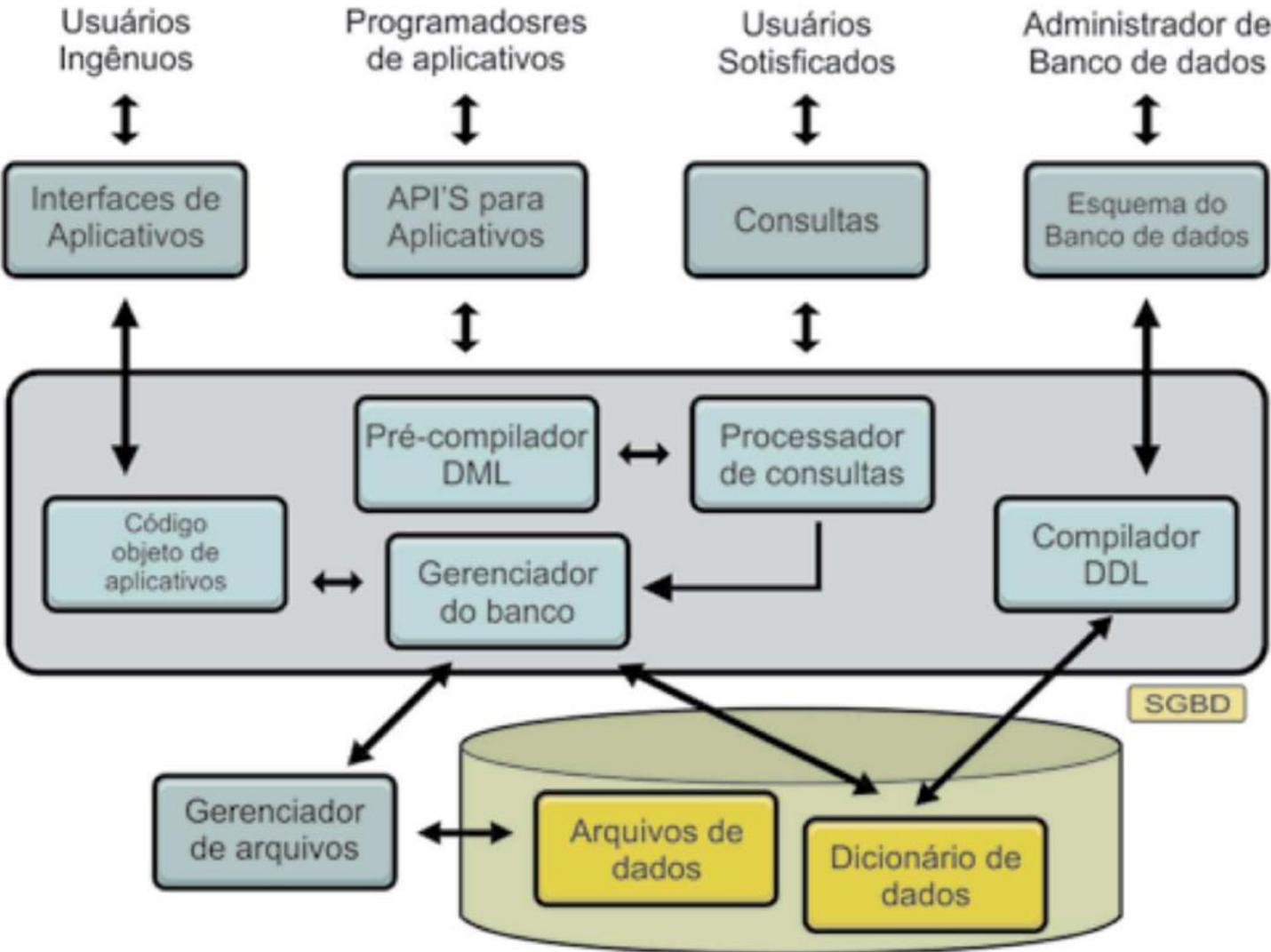
- ▶ Controle de redundância de dados;
  - ▶ Garantia de segurança e integridade dos dados;
  - ▶ Facilidade de criação de novas aplicações;
  - ▶ Controle automático de relacionamento entre registros;
  - ▶ Compartilhamento de dados;
  - ▶ *Independência de Dados;*
  - ▶ Otimização da utilização do espaço de armazenamento;
  - ▶ Controle de concorrência;
  - ▶ Recuperação de falhas;
  - ▶ Proporciona à empresa um controle centralizado de seus dados operacionais.
- 
- ▶ Responsável: Database Administrator - DBA  
(Administrador de Banco de Dados)



# Independência de Dados

- ▶ Imunidade das aplicações a mudança na estrutura de armazenamento ou na estratégia de acesso.
- ▶ Independência Lógica:
  - ▶ Alteração da Visão Global (nível conceitual ou lógico) dos dados sem que as aplicações existentes tenham que ser alteradas.
- ▶ Independência Física:
  - ▶ Alteração na representação física das estruturas de dados (nível interno), de acordo com os requisitos de novas aplicações, sem alteração nas já existentes.

# Arquitetura de um Sistema de Banco de Dados

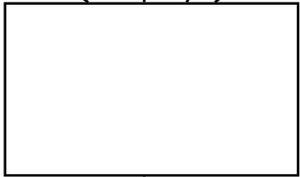


# Arquitetura de um Sistema de Banco de Dados

**Nível Externo**  
(Visão do Usuário Individual)



**Nível Conceitual**  
(Visão do Conjunto de Usuários)



**Nível Interno**  
(Visão do Armazenamento)



# Os Três Níveis da Arquitetura

- ▶ O nível **interno** é o mais próximo do armazenamento físico. Relaciona-se à forma como os dados estão realmente armazenados.
- ▶ O nível **externo** é o mais próximo aos usuários, descrevendo apenas parte do banco de dados, ou seja, como os dados são vistos pelos usuários individuais.
- ▶ O nível **conceitual** é a visão lógica global do banco de dados, como sendo a integração das visões externas. Descreve quais dados estão no banco de dados e as relações entre eles.



# O Nível Externo

- ▶ É o mais próximo do usuário.
- ▶ Tipos de usuários:
  - ▶ Programador de Aplicações
  - ▶ Usuário Final
  - ▶ Administrador de Banco de Dados (DBA)
- ▶ Cada tipo de usuário possui uma linguagem específica à sua disposição.
- ▶ Linguagens:
  - ▶ Hospedeiras
  - ▶ Sub-linguagem de SGBD para:
    - ▶ Definição de Dados (DDL)
    - ▶ Manipulação de Dados (DML)



# O Nível Conceitual

- ▶ Representação de todo o conteúdo de informações do Banco de Dados.
- ▶ Integração das visões externas.
- ▶ Esquema conceitual de toda a semântica do Banco de Dados.
- ▶ Representação conceitual do problema.



# O Nível Interno

- ▶ Voltado para a forma como os dados são realmente armazenados.
- ▶ É a visão do registro armazenado (arquivos).
- ▶ Descreve:
  - ▶ Tipos de registros armazenados;
  - ▶ Organização dos registros;
  - ▶ Índices de acessos;
  - ▶ Estruturas de armazenamento e recuperação (Pilhas, filas, árvores, grafos, listas, ... ).



# Profissionais Envolvidos com Sistemas de Bancos de Dados

- ▶ **Administrador de Banco de Dados:**
  - ▶ responsável por autorizar o acesso ao banco de dados e coordenar e monitorar seu uso.
- ▶ **Projetista do Banco de Dados:**
  - ▶ responsável pela identificação dos dados a serem armazenados no banco de dados; por escolher estruturas apropriadas para representar e armazenar tais dados.
- ▶ **Usuários Finais:**
  - ▶ Profissionais que precisam ter acesso ao banco de dados para consultar, modificar e remover dados.
- ▶ **Analistas de Sistemas e Programadores de Aplicação:**
  - ▶ Os analistas desenvolvem especificações das transações que atendem aos requisitos dos usuários, e os programadores implementam estas especificações produzindo os programas. Devem estar familiarizados com todas as capacidades do SGBD.

# Propriedades de SGBDs

- ▶ Consistência de dados:
  - ▶ O SGBD avalia os dados recebidos através de regras de integridade, garantindo que sempre estejam corretos.
- ▶ Validade:
  - ▶ Os dados são válidos quando pertencem ao domínio de valores possíveis naquele caso.
- ▶ Completude:
  - ▶ Todos os dados que precisam ser conhecidos estão disponíveis.
- ▶ Consistência:
  - ▶ Sempre que a mesma informação é gravada, mesmo que em locais diferentes, ela tem o mesmo valor.

# Propriedades de SGBDs - Exemplo

Nome	Endereco	Telefone
José de Almeida	R. Tulipas 89	0162761112
Jose da Silva	Av. Clóvis Penteado 98	Residencial
Aparecido Navarro	R. Marechal Deodoro	0192445618

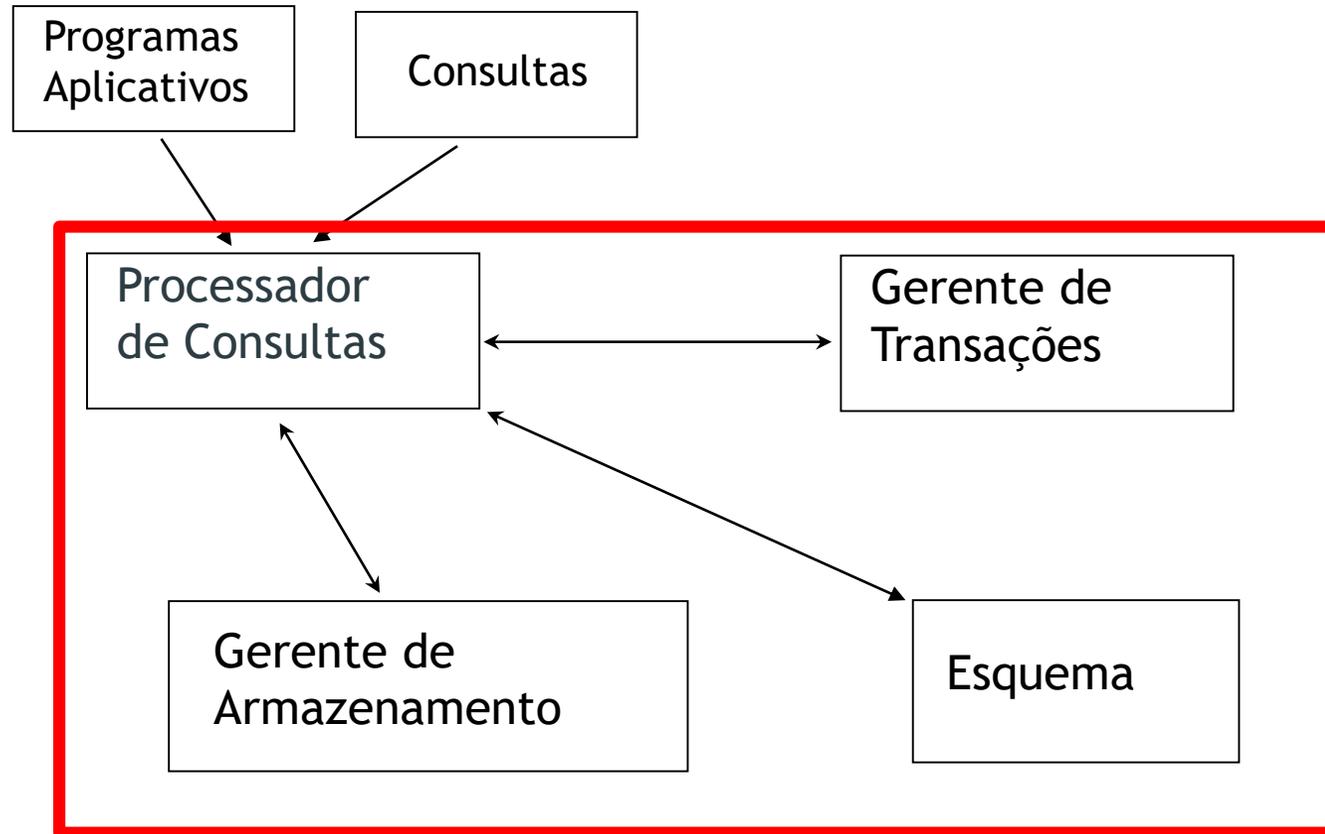
Dado inválido !

# Estrutura Geral de um SGBD



nemo

# SGBD: Módulos Componentes



# Processador de Consultas

- ▶ Tem como objetivo responder com o menor tempo possível as consultas sobre a base de dados.
- ▶ Compreende:
  - ▶ análise sintática;
  - ▶ análise semântica;
  - ▶ geração e otimização do plano de execução;
  - ▶ execução da consulta.



# Gerente de Armazenamento

- ▶ É responsável por armazenar, recuperar, manter e atualizar dados no banco de dados;
- ▶ Abrange o sub-sistema de gerência de memória;
- ▶ Faz a interface entre o sistema e os arquivos;
- ▶ Inclui ainda os índices e o dicionário de dados.



# Gerente de Transação

- ▶ Componente responsável pela gerência de acesso multi-usuário ao SGBD;
- ▶ Inclui o controle de concorrência;
- ▶ Provê mecanismos de recuperação de falhas;
- ▶ Garante as propriedades ACID das transações.

# Esquema

- ▶ É o grande dicionário do SGBD. O esquema armazena a descrição dos objetos do banco de dados;
- ▶ É acessado durante a criação, alteração e consultas a objetos do banco de dados;
- ▶ Contém também as descrição das visões e autorizações de acesso.

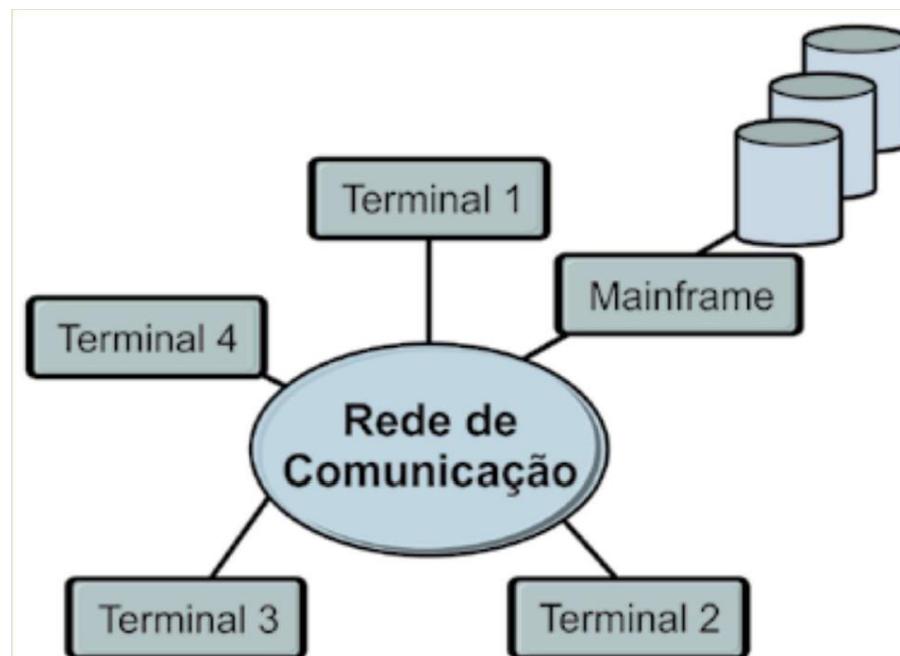


# Arquiteturas de Bancos de Dados

- ▶ Um Sistema de Bancos de Dados pode ser organizado de várias formas:
  - ▶ Centralizada;
  - ▶ Cliente-Servidor
  - ▶ Distribuída;

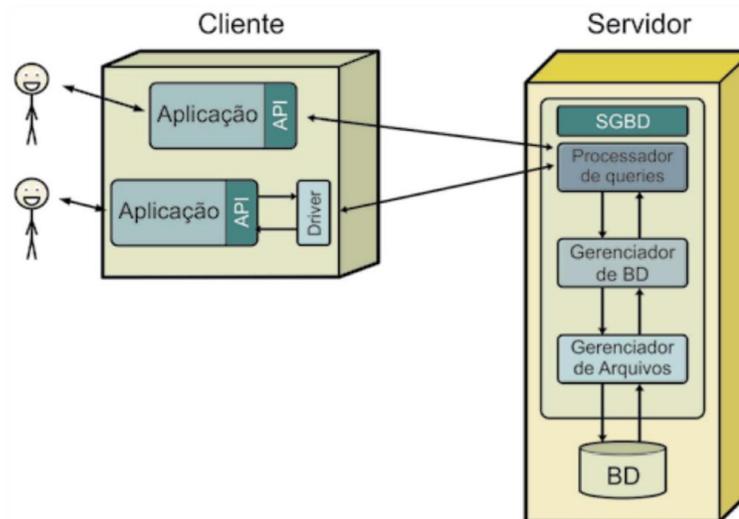
# Arquitetura Centralizada

- ▶ Na arquitetura centralizada, existe um computador com grande capacidade de processamento, o qual é o hospedeiro do SGBD, e emuladores para os vários aplicativos. Esta arquitetura tem como principal vantagem a de permitir que muitos usuários manipulem grande volume de dados. Sua principal desvantagem está no seu alto custo, pois exige ambiente especial para mainframes e soluções centralizadas.



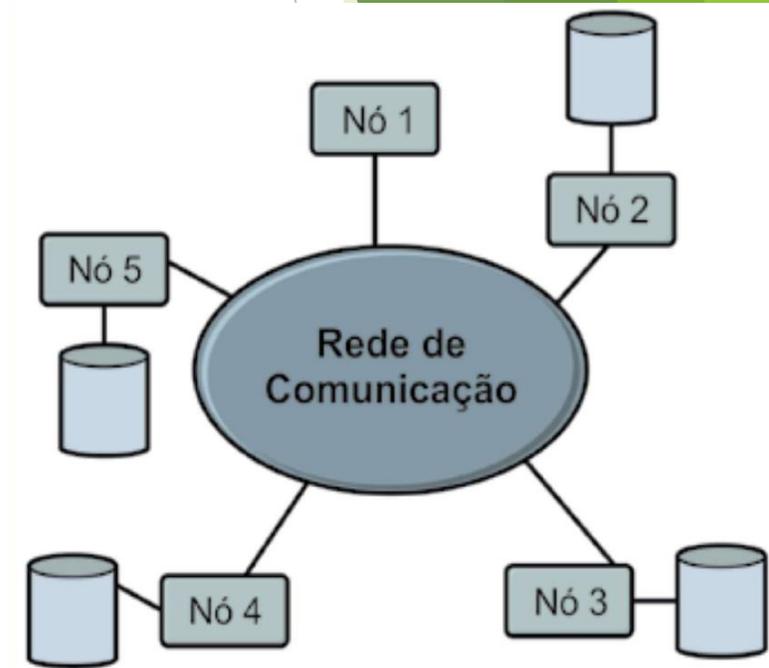
# Arquitetura Cliente-Servidor

- ▶ O cliente (front\_end) executa as tarefas do aplicativo, ou seja, fornece a interface do usuário (tela, e processamento de entrada e saída). O servidor (back\_end) executa as consultas no SGBD e retorna os resultados ao cliente. Apesar de ser uma arquitetura bastante popular, são necessárias soluções sofisticadas de software que possibilitem: o tratamento de transações, as confirmações de transações (commits), desfazer transações (rollbacks), linguagens de consultas (stored procedures) e gatilhos (triggers). A principal vantagem desta arquitetura é a divisão do processamento entre dois sistemas, o que reduz o tráfego de dados na rede.



# Arquitetura Distribuída

- ▶ A informação está distribuída em diversos servidores.
- ▶ Cada servidor atua como no sistema cliente-servidor, porém as consultas oriundas dos aplicativos são feitas para qualquer servidor indistintamente. Caso a informação solicitada seja mantida por outro servidor ou servidores, o sistema encarrega-se de obter a informação necessária, de maneira transparente para o aplicativo, que passa a atuar consultando a rede, independente de conhecer seus servidores.
- ▶ Exemplos típicos são as bases de dados corporativas, em que o volume de informação é muito grande e, por isso, deve ser distribuído em diversos servidores. A característica básica é a existência de diversos programas aplicativos consultando a rede para acessar os dados necessários, porém, sem o conhecimento explícito de quais servidores dispõem desses dados.



# Tipos de SGBDs

- ▶ Existem duas categorias de bancos de dados: **banco de dados relacionais e não relacionais**.
- ▶ **Bancos de dados relacionais** são fundamentados no paradigma da orientação a conjuntos. Seus dados são armazenados em estruturas denominadas tabelas. Cada tabela é composta por colunas (atributos e linhas), tuplas ou registros. Os bancos relacionais são a opção ideal para sistemas em que é necessária uma grande consistência de dados.
- ▶ Sua linguagem é o SQL (Structured Query Language) e seus principais representantes são Oracle, SQL Server, MySQL e PostgreSQL.

# Tipos de SGBDs

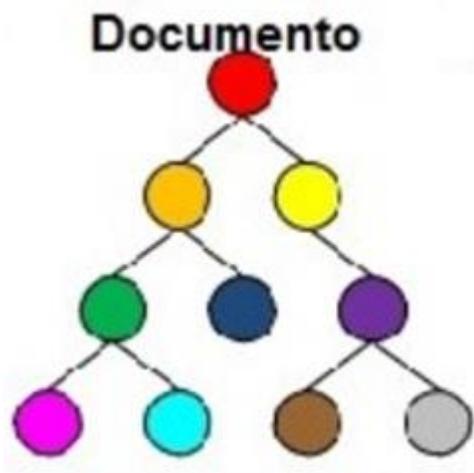
- ▶ Os **bancos de dados não relacionais** são soluções para situações nas quais os bancos relacionais não atendem. Um exemplo são os ambientes com dados mistos (imagens, mapas e tabelas), que não podem ser tabulados em linhas e colunas. Também é utilizado em grandes soluções baseadas em nuvem.
- ▶ Eles são conhecidos como NoSQL (Not Only SQL, ou em português, não apenas SQL). Buscam consistência nas informações armazenadas, disponibilidade do banco de dados e tolerância ao particionamento das informações.
- ▶ Seus bancos mais conhecidos são MongoDB, Redis e Cassandra.

# Tipos de SGBDs - Não Relacionais

- ▶ Existem vários tipos de bancos de dados NoSQL.
- ▶ Há grandes diferenças no que se refere à a forma de armazenamento e conceitos de modelagem. Há 4 tipos básicos de bancos de dados NoSQL, a saber:
  - ▶ NoSQL orientado a documento;
  - ▶ NoSQL Key-Value;
  - ▶ NoSQL representado por Grafos;
  - ▶ NoSQL modelo colunar;

# Tipos de SGBDs - Não Relacionais

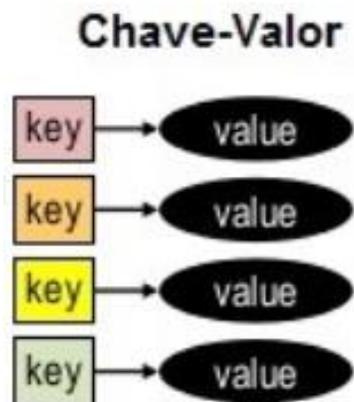
- **NoSQL orientado a documento:** consiste em uma estrutura baseada em uma coleção de documentos, sendo um documento um objeto que contém um código único com um conjunto de informações, podendo ser strings, documentos aninhados ou ainda listas. Exemplos destes são o MongoDB e CouchBase.



```
{
  "id": 49,
  "Pais": "Alemanha",
  "Regiao": "Europa",
  "Populacao":
  "PrincipaisCidades": [
    {
      "NomeCidade": "Berlin",
      "Populacao": 3610156,
    },
    {
      "NomeCidade": "Hamburg",
      "Populacao": 1746342,
    }
  ]
}
```

# Tipos de SGBDs - Não Relacionais

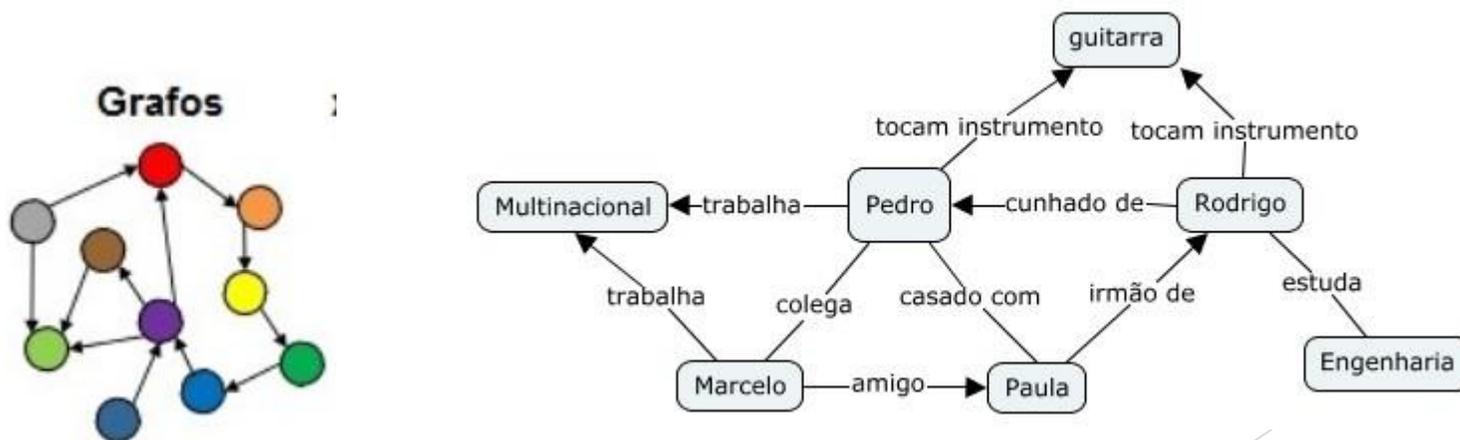
- **NoSQL Key-Value (chave-valor):** consiste em uma modelagem que indexa os dados a uma chave. Ao se armazenar os dados, sua forma de procura se dá por uma base similar a um dicionário, onde estes possuem uma chave. Esta forma de armazenamento é livre de “schema”, permite a inserção de dados em tempo de execução, sem conflitar o banco e não influenciando na disponibilidade, pois seus valores são isolados e independentes entre si. Alguns exemplos são: Oracle NoSQL, Riak, Azure Table Storage, BerkeleyDB e Redis.



Chave	Valor
carro_3345_cor	preto
carro_3345_pneu	17
carro_3365_cor	branco
carro_3365_pneu	15
carro_4560_peso	1215
carro_4715_ano	2016

# Tipos de SGBDs - Não Relacionais

- **NoSQL representado por Grafos:** Este modelo armazenamento utiliza três componentes básicos: um grafo para representar um dado, arestas ou ligações para representar a associação entre os grafos e os atributos (ou propriedades) dos nós e relacionamentos. Modelo altamente usado onde exijam dados fortemente ligados. Este modelo é vantajoso onde há consultas complexas frente aos outros modelos, pois seu diferencial é o ganho de performance. Alguns exemplos são: Neo4J, OrientedDB, GraphBase e InfiniteGraph.



# Tipos de SGBDs - Não Relacionais

- **NoSQL modelo Colunar:** Basicamente consiste em uma Tabela, onde nela possui várias famílias de colunas, e dentro destas famílias, colunas onde estão as propriedades. Neste modelo, as entidades são representadas por tabelas e os dados gravados em disco, o modelo caracteriza-se por indexar um dado por uma Tripla, que consiste em linha, coluna e timestamp, sendo este o que permite verificar as diferentes versões de um dado. Os valores das propriedades das colunas podem ser semelhantes ao modelo “Key-Value”. São bancos de dados indicados para mídias sociais e problemas que envolvem consultas complexas.

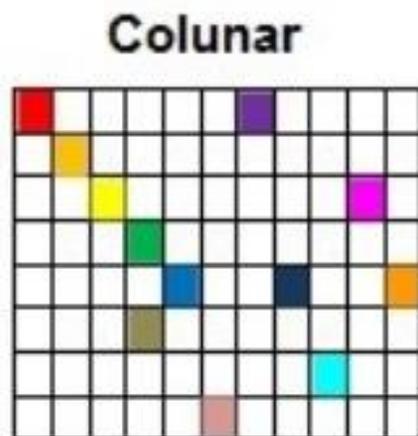


Tabela			
Família de coluna 1		Família de coluna 2	Família de coluna 3
Coluna 1	Coluna 2	Coluna 3	Coluna 4
#1 {Chave: Valor, Chave: Valor}			
#2 {Chave: Valor, Chave: Valor}			

# Tipos de SGBDs - Bancos de Dados OO

- ▶ Um banco de dados orientado a objetos é um banco de dados em que cada informação é armazenada na forma de objetos, ou seja, utiliza a estrutura de dados denominada orientação a objetos. O gerenciador do banco de dados para um orientado a objeto é referenciado por vários como ODBMS ou OODBMS.
- ▶ Existem dois fatores principais que levam à adoção da tecnologia de banco de dados orientados a objetos:
- ▶ A primeira, é que, em um banco de dados relacional, se torna difícil de manipular com dados complexos.
- ▶ O segundo fator é que os dados são geralmente manipulados pela aplicação escrita usando linguagens de programação orientada a objetos, como C++, C#, Java, Python, e o código precisa ser traduzido entre a representação do dado e as tuplas da tabela relacional, o que além de ser uma operação tediosa de ser escrita, consome tempo. Esta perda entre os modelos usados para representar a informação na aplicação e no banco de dados é também chamada de "perda por resistência".

# Tipos de SGBDs - Bancos de Dados OO

- ▶ Num banco de dados orientado a objetos puro, os dados são armazenados como objetos onde só podem ser manipulados pelos métodos definidos pela classe de que estes objetos pertencem. Os objetos são organizados numa hierarquia de tipos e subtipos que recebem as características de seus supertipos. Os objetos podem conter referências para outros objetos, e as aplicações podem conseqüentemente acessar os dados requeridos usando um estilo de navegação de programação.
  
- ▶ Um dos objetivos de um SGBDO (Sistema de Gerenciamento de Banco de Dados de Objeto) é manter uma correspondência direta entre objetos do mundo real e do banco de dados, de modo que objetos não percam sua integridade e identidade e possam facilmente ser identificados e operados. Assim, o SGBDO oferece uma identidade única e imutável a cada objeto armazenado no banco de dados chamado OID (Identificador de objeto).

# Tipos de SGBDs - Bancos de Dados OO

- ▶ A linguagem de consulta utilizada por esses bancos é a OQL (Object Query Language).
- ▶ O acesso aos dados pode ser rápido porque as junções (*join*) geralmente não são necessárias, isto é, porque um objeto pode ser obtido diretamente sem busca, seguindo os ponteiros. As técnicas baseadas em ponteiros são otimizadas para "rotas de pesquisa" ou pontos de vista muito específicos. Entretanto, para o propósito de consultas gerais a mesma informação, técnicas baseadas em ponteiros tenderão a ser mais lentas e mais difíceis de se formular do que as relacionais. Desta maneira, a abordagem navegacional parece simplificar para usos específicos conhecidos às custas do uso geral.
- ▶ Outra coisa que trabalha contra os ODBMS parece ser a perda da interoperabilidade com um grande número de ferramentas/características que são tidas como certas no mundo SQL, incluindo a indústria de padrões de conectividade, ferramentas de relatório, ferramentas de OLAP e backup, e padrões de recuperação. Adicionalmente, banco de dados orientado a objetos perdem o fundamento formal matemático, ao contrário do modelo relacional, e isto às vezes conduz à fraqueza na sustentação da consulta.

# Propriedades de um Banco de Dados

- ▶ Atualmente as aplicações suportam vários usuários e sendo assim o banco de dados tem que garantir a confiabilidade nas transações, haja vista que muitas podem ocorrer forma concorrente.
- ▶ O que é uma transação?
- ▶ Uma transação é uma sequência de operações executadas como uma única unidade lógica de trabalho, ou seja, é um conjunto de procedimentos, executados num banco de dados, que o usuário percebe como uma única ação.
- ▶ ACID é um conceito que se refere às quatro propriedades de transação de um sistema de banco de dados: Atomicidade, Consistência, Isolamento e Durabilidade.

# Propriedades de um Banco de Dados

- ▶ **Atomicidade:** Todas as ações que compõem a unidade de trabalho da transação devem ser concluídas com sucesso, para que seja efetivada. Se durante a transação qualquer ação que constitui unidade de trabalho falhar, a transação inteira deve ser desfeita (rollback). Quando todas as ações são efetuadas com sucesso, a transação pode ser efetivada e persistida em banco (commit).
- ▶ **Consistência:** A transação cria um novo estado válido dos dados ou, em caso de falha, retorna todos os dados ao seu estado anterior ao início da transação.
- ▶ **Isolamento:** Uma transação em andamento mas ainda não validada deve permanecer isolada de qualquer outra operação, ou seja, garantimos que a transação não será interferida por nenhuma outra transação concorrente.
- ▶ **Durabilidade:** Dados validados são registrados pelo sistema de tal forma que mesmo no caso de uma falha e/ou reinício do sistema, os dados estão disponíveis em seu estado correto.

# Referencias

- ▶ [https://pt.wikipedia.org/wiki/Banco\\_de\\_dados\\_orientado\\_a\\_objetos](https://pt.wikipedia.org/wiki/Banco_de_dados_orientado_a_objetos) Acesso em 18/3/2019
- ▶ <http://micreiros.com/tipos-de-bancos-de-dados-nosql/> Acesso em 16/03/2019
- ▶ <https://medium.com/opensanca/o-que-%C3%A9-acid-59b11a81e2c6> Acesso em 18/08/2019
- ▶ [https://pt.wikipedia.org/wiki/Banco\\_de\\_dados](https://pt.wikipedia.org/wiki/Banco_de_dados) Acesso em 18/08/2019

That's all Folks!



nemo