

### DEFINIÇÃO

Apresentação da evolução histórica dos bancos de dados, abordando as caracterfsticas dos sistemas de banco de dados (SBD) e a arquitetura dos sistemas de gerência de banco de dados (SGBD).

### PROPÓSITO

Compreender a origem e as características dos SBDs, bem coma suas funcionalidades, vantagens e desvantagens, além de conhecer a arquitetura dos SGBDs e os sistemas mais utilizados em bancos de dados.

### MÓDULO 1

Reconhecer o histórico dos bancos de dados e suas técnologias

#### MÓDULO 2

ldentificar as características dos sistemas de banco de dados (SBD)

#### MÓDULO 3

Descrever a arquitetura dos sistemas de gerência de banco de dados (SGBD)

### INTRODUÇÃO

**MÓDULO 1**

Reconhecer o histórico dos bancos de dados e suas tecnologias

# DEFINIÇÃO DE BANCO DE DADOS

Você certamente ja leu sobre o termo **banco de dados** em algum contexto técnico ou geral na média tradicional ou na internet.

Mas o que é um banco de dados?

O termo banco de dados, no sentido técnico, origina-se de ***database,*** do inglês, e o livro-texto de edição norte-americana mais adotado no mundo o define de maneira simples e direta: "banco de dados e uma coleção de dados relacionados" (ELMASRI; NAVATHE, 2019), em que dados sao **fatos conhecidos que podem ser registrados e possuem significado implicito.**

\* **EXEMPLO**

Nome, data de nascimento, enderego e telefone sao dados relacionados entre si, inerentes a uma pessoa conhecida em certo contexto. Antes de elaboramos mais essa definigao, vamos relembrar um pouco da hist6ria dos bancos de dados.

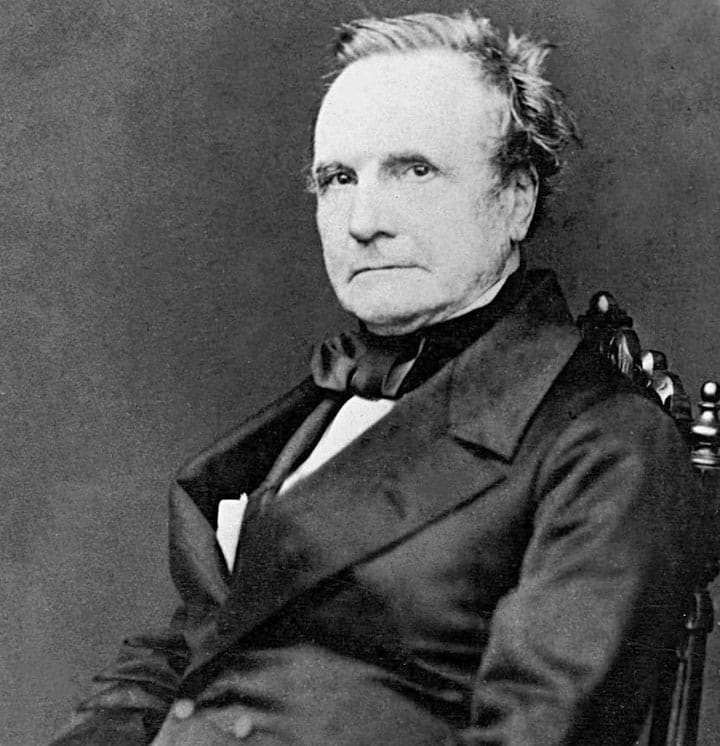


**m Fonte:** Shutterstock I **Por:** fullvector

0 conceito de banco de dados como uma colegao de dados relacionados sempre existiu como componente central dos sistemas de informagao. Estes, por sua vez, sempre existiram desde que a humanidade se organizou como sociedade. Segundo afirmam Melo, Silva e Tanaka (1998), o que tern mudado rapidamente ao longo da hist6ria e a tecnologia que permite a sua implementagao e que se confunde com o pr6prio conceito de sistemas de informagao.

Assim, antes da existencia do computador, bancos de dados existiam sob a forma de registros ffsicos em papel, organizados em pastas dentro de armarios, que formavam os arquivos dos sistemas de informa9ao, operados manualmente pelos seus usuarios. Sera que ainda existem sistemas de informa9ao desse tipo em pleno seculo **XXI?**

### EVOLUCAO DOS SISTEMAS DE INFORMACAO EM COMPUTADOR



Fonte:Shutterstock

**m** Charles Babbage. **(Fonte:** Wikimedia)

#### ERA DO PROCESSAMENTO DE DADOS

Historicamente, o computador, inventado na decada de 1940 ao fim da Segunda Guerra Mundial, era usado primordialmente como uma maquina para calculos matematicos complexos, a exemplo da maquina diferencial de **Charles Babbage,** do seculo XIX.

**CHARLES BABBAGE**

Charles Babbage (1791-1871) e um dos mais celebres icones no mundo da computac;ao. As suas notaveis contribuic;oes para a area fizeram dele o pioneiro dos computadores. A Maquina Diferencial N. 1 foi a primeira calculadora automatica a ser inventada e ainda e considerada uma pec;a Cmica pela precisao que na epoca apresentava.

**Fonte:** Pplware.

Logo se percebeu que, grac;as a arquitetura criada pelo seu inventor, **John von Neumann,**

baseada em uma unidade central de processamento que armazena programas e dados, o computador tambem serve para o processamento de dados e nao apenas para calculos.



Fante:Shutterstock

**m** John van Neumann. **(Fonte:** Los Alamos National Laboratory)

**JOHN VON NEUMANN**

John van Neumann (1903-1957) foi um matematico hungaro, naturalizado norte-americano,

considerado o pai da Teoria dos Jogos. Seus interesses abrangiam I6gica, assuntos militares, computa<;ao, economia, entre outros.



Fante:Shutterstock

**m Fonte:** Shutterstock I **Por:** Lifestyle and Wedding ph

Essa utilidade do computador foi impulsionada com a invengao do disco magnetico pela IBM, em 1957, que o denominou de Dispositivo de Armazenamento de Acesso Direto (DASO, do ingles ***DirectAcess Storage Device).***

Fundamentalmente, o que esse dispositivo - atualmente conhecido coma disco rfgido e pela sigla **HD *(Hard Disk)-*** apresentou de novidade, a epoca, foi a capacidade de leitura de dados externos a unidade central de processamento de forma direta, sem a necessidade de leitura

sequencial, coma em fitas magneticas.

Com o advento do armazenamento externo em disco rfgido, nasceu a era do processamento de dados par computador.

Voce ja ouviu falar em Centro de Processamento de Dados (CPD), denomina9ao que ainda persiste em organiza96es tradicionais?

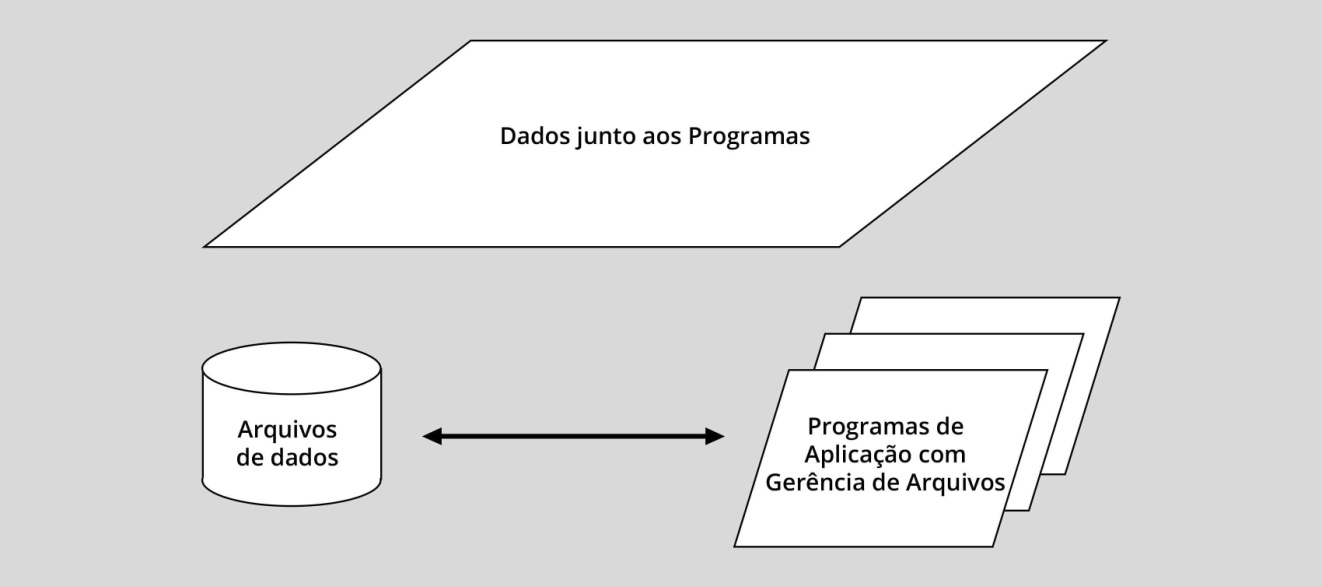
Nessa era, os programas de aplica9ao, desenvolvidos em uma linguagem de programa9ao (usualmente COBOL, em aplica96es empresariais, ou Fortran, em aplica96es cientfficas), manipulavam dados armazenados em arquivos hospedados em disco magnetico, utilizados pelo sistema operacional e formando o que se denomina **sistema de arquivos.**



Fonte:Shutterstock

**m Fonte:** Shutterstock I **Por:** Alexandru Chiriac

A figura a seguir mostra a evolu9ao obtida ao possibilitar que os programas acessassem os dados externamente em arquivos no disco (configurando um avan90 em rela9ao aos programas que continham internamente os pr6prios dados) para execu9ao em lotes ***(batch,*** em ingles), na fase inicial do uso do computador.



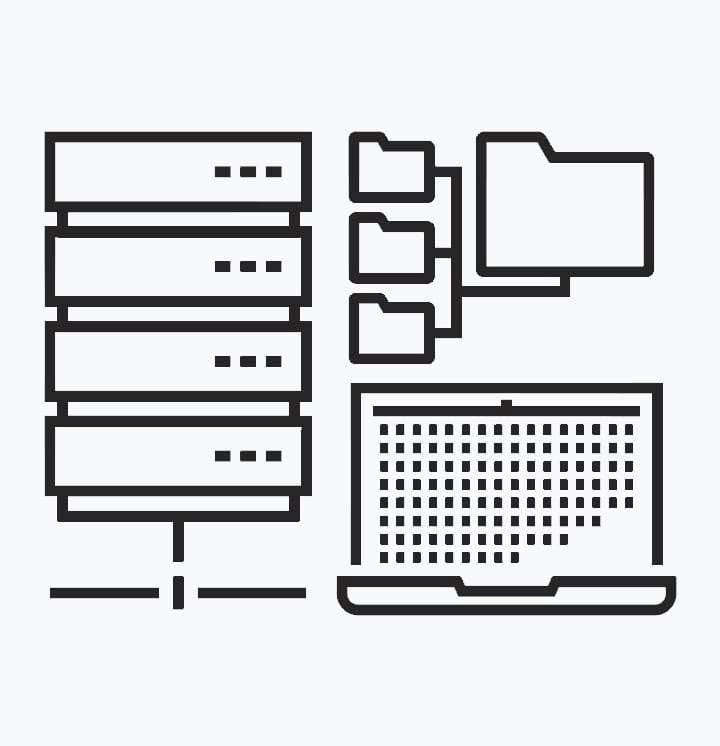
***ii* Fonte:** Tanaka (2018)

Esse modelo de processamento de dados com sistema de arquivos foi largamente utilizado no infcio do emprego do computador em sistemas de informa9ao empresariais, ap6s o advento do disco magnetico, persistindo ate os dias atuais, nos chamados sistemas legados. Exemplo disso, foi a maior demanda por programadores da linguagem COBOL durante a pandemia de COVID- 19 para realizar manuten9ao em sistemas da Administra9ao Publica do governo dos EUA.

#### 0 SAIBAMAIS

Para saber mais sobre o aumento da procura por programadores COBOL durante a pandemia de COVID-19, nao deixe de verificar a indica9ao feita no Explore+ ao fim deste tema.

#### PRIMORDIOS DOS SISTEMAS DE BANCO DE DADOS



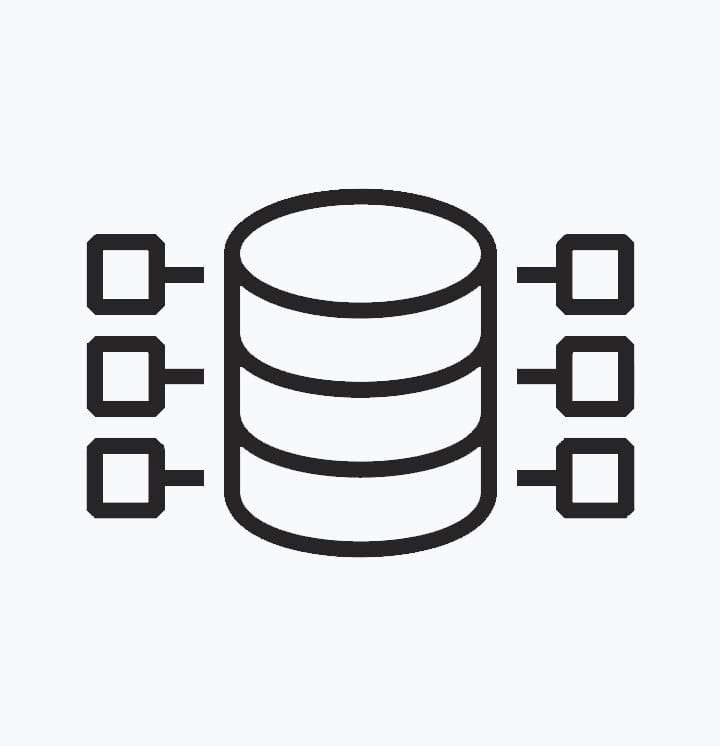
Fonte:Shutterstock

Seguindo na historia, o advento dos bancos de dados foi uma evoluc;ao natural dos sistemas de arquivos. Observe na figura anterior que os programas os quais manipulam os arquivos de dados, alem de implementarem a logica da aplicac;ao, tern de conter um modulo para a gerencia dos arquivos de dados. Esse modulo deve ser repetido em todos os programas que precisam acessar e manipular o mesmo arquivo de dados.



Fonte:Shutterstock

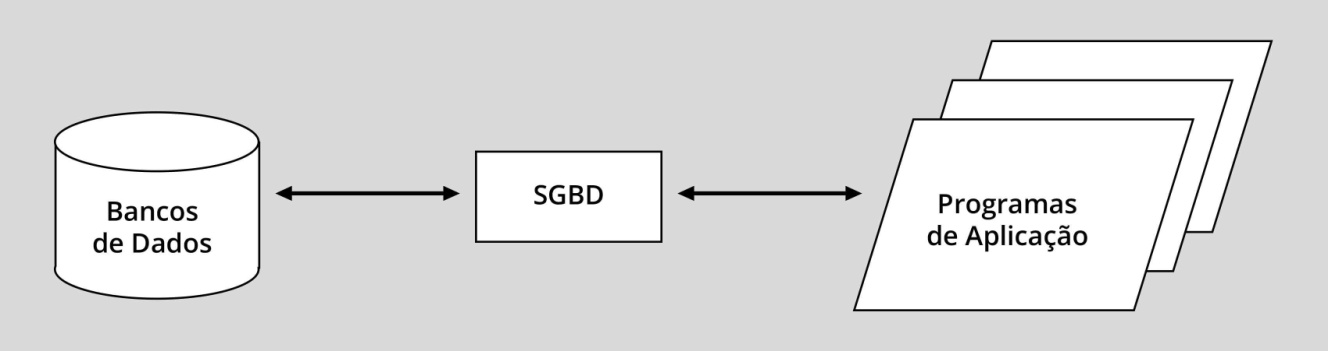
Por exemplo, o departamento de pessoal de uma organizac;ao mantem o arquivo com os dados dos empregados. Suponha que o departamento de produc;ao tambem precise usar dados desse arquivo para alocar empregados em projetos. Nesse caso, os programas de aplicac;ao que atendem aos dois departamentos deverao canter o mesmo modulo de gerencia do arquivo de empregados, causando uma repetic;ao de c6digo de programac;ao e dificultando a sua manutenc;ao.



Fonte:Shutterstock

Assim, os **sistemas de banco de dados** (SBD) vieram para mitigar esse problema, a partir de 1960, tirando dos programas de aplicac;ao a responsabilidade de gerenciar os arquivos de dados, tarefa que passou a ser delegada a um software intermediario, denominado de sistema de gerencia de banco de dados (SGBD), como mostra a figura a seguir.

Essa propriedade dos sistemas de banco de dados e denominada de **independencia entre dados e programas,** uma diferenc;a primordial em relac;ao aos sistemas de arquivos.



***ii* Fonte:** Tanaka (2018)

Em outras palavras, ocorreu uma modularizac;ao do sistema de informac;ao, com a distribuic;ao de responsabilidades entre os programas de aplicac;ao e o SGBD. Os programas de aplicac;ao passaram a se ocupar exclusivamente das funcionalidades da aplicac;ao propriamente dita, deixando as tarefas de acesso e manipulac;ao dos dados armazenados em disco para o SGBD, um software tipicamente auxiliar, de bastidores ou, como se costuma dizer no jargao do mercado, um servic;o de ***back end.***

##### ATENCAO

Perceba a diferenc;a entre o sistema de banco de dados (SBD) e o sistema de gerencia de banco de dados (SGBD), pois o primeiro e mais amplo, englobando o SGBD, os pr6prios programas de aplicac;ao e os bancos de dados manipulados por eles.

Neste ponto, cabe um questionamento importante, cada vez mais valido em face dos avanc;os das tecnologias de hardware de mem6ria, tanto de mem6ria principal (RAM) quanto de mem6ria secundaria (discos magneticos ou ***hard disk drives*** (HOD) e de semicondutores ou ***solid state drives*** (SSD)).

A questao e: qual dos tres modelos de sistemas e o mais eficiente para uma aplicac;ao com o mesmo volume de dados, ou seja, o modelo monolltico com dados junto dos programas; o modelo com sistemas de arquivos; ou o modelo com sistemas de bancos de dados?

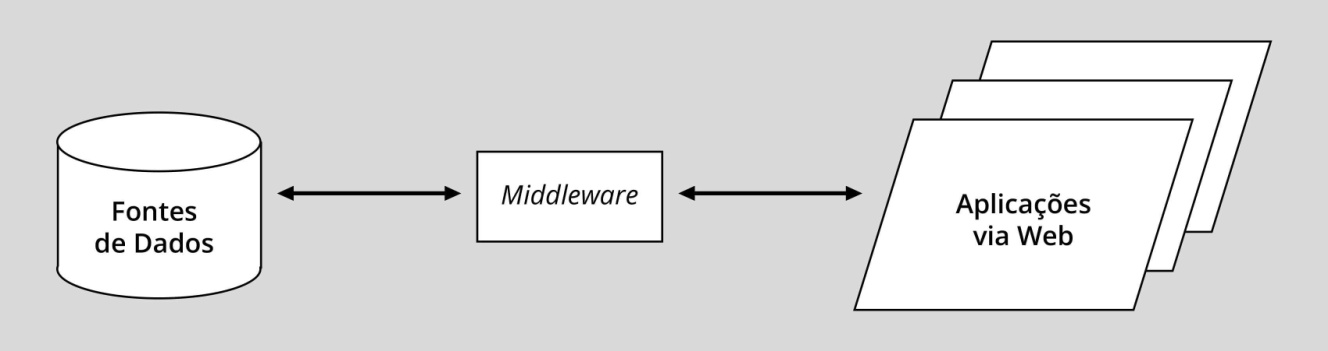
Adiaremos a resposta a essa questao para o final deste modulo.

##### ESTAGIO ATUAL DOS SISTEMAS DE INFORMACAO (NA WEB)

Antes de discorrer sobre o hist6rico dos SBDs, vale completar a evoluc;ao dos sistemas de informac;ao ate os dias atuais, fortemente influenciada pela revoluc;ao tecnol6gica causada pela ***World Wide Web*** no final do seculo XX.

Com a popularizac;ao da interface Web no desenvolvimento das aplicac;oes, surgiram novas linguagens de programac;ao e novas formas de armazenamento e acesso a dados em fontes com diferentes formatos. Assim, o SGBD das aplicac;oes tradicionais pode ser considerado atualmente como um genera de software basico, com papel intermediario, que denominamos na

figura a seguir de ***middleware,*** em que se incluem servidores de aplicac;6es das diferentes linguagens e ambientes de desenvolvimento Web.



***ii* Fonte:** Tanaka (2018)

Essa figura resume o atual estagio dos sistemas de informac;ao na Web, em que as fontes de dados nao se restringem a dados estruturados, como em bancos de dados tradicionais, admitindo volumes gigantescos em diversos formatos, localizac;6es e velocidade de produc;ao, caracterfsticas marcantes do conceito de ***Big Data.*** lgualmente, as aplicac;6es via Web sao desenvolvidas em uma diversidade de plataformas digitais, de ***smartphones*** a supercomputadores, que tern em comum a conexao com a internet e, em consequencia, a computac;ao em nuvem (***Cloud Computing).***

# EVOLUCAO DOS SISTEMAS DE BANCO DE DADOS

#### BANCOS DE DADOS NAVEGACIONAIS

Ha uma controversia sobre qual foi o primeiro SGBD implementado e utilizado comercialmente na decada de 1960. Sabe-se que duas iniciativas independentes ocorreram paralelamente, resultando em dois produtos comerciais:

###### IDS {INTEGRATED DATA SYSTEM)

Criado por Charles Bachman (1924-2017) no ambito de um comite que padronizou a linguagem COBOL (CODASYL, de ***Committee on Data Systems Languages).***

###### IMS (INFORMATION MANAGEMENT SYSTEMS)

Criado pela IBM na esteira do sucesso da invern;ao do disco magnetico anos antes.

0 IDS e o IMS tinham em comum a caracterfstica de que os dados eram acessados por meio de programas que "navegam" de registro em registro pela estrutura dos dados armazenados em disco. Por causa dessa caracterfstica, atualmente aqueles SGBDs e outros que seguiram a mesma abordagem sao denominados de navegacionais.

Observe a **diferenc;a** entre eles:

IDS

Usava a estrutura de dados de grafos ou redes, daf a denomina9ao de ***network databases.***

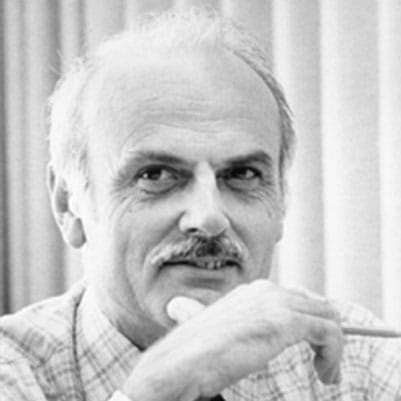
IMS

Adotava a estrutura de dados de arvores, que e um tipo de grafo mais restrito do que as redes, baseado em hierarquias, originando a denomina9ao ***hierarchical databases.***

Varios SGBDs foram implementados com variantes desses modelos de banco de dados, coma o OMS **(Data *Management System)*** e o IDMS ***(Integrated Database Management System).***

Vale relembrar que muitos sistemas de informa9ao legados daquela epoca ainda utilizam esses SGBDs navegacionais, a exemplo da demanda por programadores COBOL em meio a pandemia de COVID-19.

##### 0 MODELO RELACIONAL DE BANCO DE DADOS



Fonte:Shutterstock

**aJ** Edgar F. Codd. **(Fonte:** Wikimedia)

A grande revoluc;ao na hist6ria dos bancos de dados ocorreu na virada das decadas de 1960 e 1970, com a publicac;ao do artigo seminal do matematico pesquisador da IBM, **Edgar Codd,** intitulado *A Relational Model of Data for Large Shared Data Banks,* que introduziu o modelo relacional de banco de dados.

0 artigo de Codd, uma das obras mais citadas na comunidade da computac;ao em todos os tempos, foi o marco do chamado modelo relacional de banco de dados, cuja estrutura de dados, diferentemente dos grafos dos bancos de dados navegacionais, e uma func;ao matematica denominada relac;ao.

**EDGAR CODD**

Edgar Frank Codd (1923-2003) foi um cientista da computac;ao e matematico americano que inventou o modelo de dados relacionais, que levou a criac;ao do banco de dados relacional, um metodo padrao de recuperac;ao e armazenamento de dados do computador.

**Fonte:** Encyclopaedia Britannica.

Codd criou uma Algebra Relacional e um Calculo Relacional, nos quais baseou toda a teoria matematica das rela96es em que fundamentou o modelo relacional. Apesar da base te6rica do modelo, a estrutura de dados subjacente tern o merito de ser muito simples, pois uma rela9ao nada mais e do que uma tabela formada por colunas e linhas, em cujas celulas estao armazenados os dados, conceito compreensfvel pelo senso comum de qualquer leigo em Matematica ou computa9ao, como podemos ver a seguir.

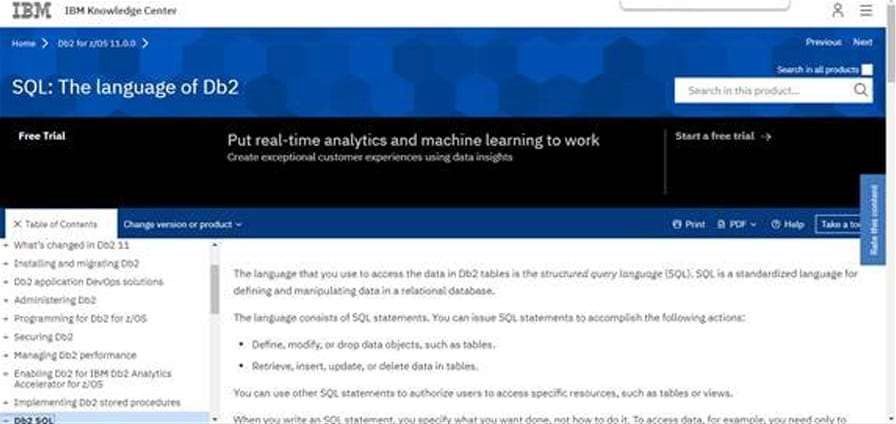
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Linhas/Colunas** | **Nome** | **Data de nascimento** | **Sexo** | **Departamento** |
| Linha 1 | Jorge | 15/01/1999 | Masculino | Produ9ao |
| Linha 2 | Ana | 05/07/1980 | Feminino | Recursos Humanos |
| Linha 3 | Joana | 10/09/1990 | Feminino | Pesquisa |

**Atenc;ao!** Para visualiza9ao completa da tabela utilize a rolagem horizontal

A solidez da fundamenta9ao matematica do modelo relacional disparou uma serie de iniciativas de implementa9ao em empresas, como a pr6pria IBM, e no meio academico, principalmente nas universidades do estado da California, onde se localizava o centro de pesquisas da IBM. A partir de entao, a IBM patrocinou o projeto **System R** (de ***Relational),*** enquanto a Universidade da

California em Berkeley (UCB) deu infcio a implementa9ao academica de um SGBD relacional

denominado de **Ingres *(Interactive Graphics Retrieval* System).**



Fonte:Shutterstock

**m** DB2. Fonte: IBM

##### PRINCIPAIS SGBDS RELACIONAIS

0 projeto **System R** deu origem ao SGBD comercial da IBM, inicialmente denominado SQL/DS ***(Structured Query Language/Data System),*** depois renomeado de DB2, atualmente um dos lfderes no mercado de bancos de dados corporativos, com versoes em diferentes plataformas de hardware/software e na nuvem.

##### 0 SAIBA MAIS

A linguagem SQL, criada pela IBM como uma linguagem de consulta e manipula9ao de dados dos bancos de dados relacionais, passou a ser conhecida como sin6nimo de SGBD relacional, chegando a ser confundida com produtos que levam a sigla em seu nome.

No ambito comercial, tambem despontou, como decorrencia do sucesso do modelo relacional, o desenvolvimento de um SGBD pela empresa inicialmente denominada Software Development Laboratories (SOL), depois renomeada Relational Software Inc. (RSI) e, finalmente, Oracle Corporation, nome pelo qual e atualmente reconhecido como o SGBD lfder do mercado global de banco de dados.



Fonte:Shutterstock

**aJ Fonte:** Shutterstock I **Por:** Anton Garin



Fonte:Shutterstock

**aJ Fonte:** Shutterstock I **Por:** dennizn

Em 2010, com a aquisigao da Sun Microsystems, uma grande empresa de hardware tradicionalmente incentivadora de projetos de software livre, a Oracle incorporou entre seus produtos o MySQL, um SGBD relacional de reconhecida lideranga na comunidade de

desenvolvimento de sistemas para a Web. 0 SGBD MySQL, associado ao sistema operacional Linux, ao servidor Web Apache ea linguagem de programa9ao PHP, formou o quarteto de software conhecido pela sigla LAMP, de grande sucesso no desenvolvimento de aplica96es Web ate os dias atuais.

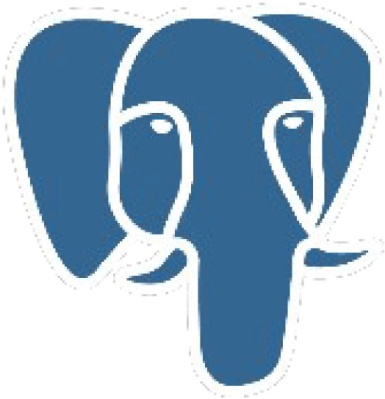
Na frente academica, o projeto Ingres, da UCB (Universidade da California, Berkeley) deu origem a vers6es comunitarias mediante licen9a livre da pr6pria universidade junta com seu sistema operacional Unix, denominado BSD ***(Berkeley Software Distribution).*** 0 projeto academico originou um produto comercial de mesmo nome, Ingres DBMS, que concorreu diretamente com o Oracle e o SQL/DS nos prim6rdios dos SGBDs relacionais.

0 esfor90 de desenvolvimento do Ingres envolveu muitos pesquisadores, professores e estudantes, os quais acabaram levando o seu c6digo livre em linguagem C para implementa9ao em outros produtos comerciais, notadamente o SGBD Sybase que, na decada de 1990, associou-se a Microsoft, dando origem ao SQL Server, atualmente um dos lfderes no mercado de bancos de dados relacionais.



Fonte:Shutterstock

**aJ Fonte:** Wikipedia.



Fonte:Shutterstock

**r.l** Logo Postgre. Fonte: Wikimedia

A continuidade do projeto Ingres deu frutos tambem na area academica com a evolugao para um modelo de dados alem do relacional, estendido com conceitos da programagao orientada a objetos, denominado Postgres (de *Post Ingres).*

##### 0 SAIBA MAIS

Ap6s a incorporagao da linguagem SOL na decada de 1990, o Postgres foi rebatizado como PostgreSQL, atualmente reconhecido como o mais avangado SGBD ***open source*** do mundo. Para saber mais sabre o PostgreSQL, nao deixe de verificar a indicagao feita no Explore+ ao fim deste tema.

0 PostgreSQL e o MySQL sao os SGBDs mais utilizados no aprendizado dos bancos de dados relacionais pela sua popularidade e pelo fato de disponibilizarem versoes com licenga e documentagao livres.

##### OUTROS MODELOS DE SGBDS

No *ranking* de popularidade dos SGBDs, disponibilizado pelo DB-Engines em seu website, destacam-se, entre os que adotam o modelo relacional: Oracle, MySQL, Microsoft SQL Server, PostgreSQL e IBM DB2.

Cabe observar que esse *ranking* nao trata exclusivamente de SGBDs do modelo relacional de dados. Os pr6prios SGBDs relacionais, mencionados como lfderes de mercado, sao

classificados no *ranking* como "multimodelos", porque implementam funcionalidades que vao alem do modelo relacional. Vejamos:

#### ORACLE

Relacional e multimodelo (documentos, grafos e RDF)

#### MYSQL

Relacional e multimodelo (documentos)

#### MICROSOFT SQL SERVER

Relacional e multimodelo (documentos e grafos)

#### POSTGRESQL

Relacional e multimodelo (documentos)

#### IBM DB2

Relacional e multimodelo (documentos e RDF)

#### E O QUE SAO ESSES OUTROS MODELOS DE BANCO DE DADOS, ALEM DO RELACIONAL?

Nao resta duvida de que o modelo relacional se firmou no mundo corporativo, sendo utilizado na grande maioria dos sistemas de informagao empresariais pela sua popularidade e robustez dos produtos disponiveis ao longo de decadas de desenvolvimento, bem como pela padronizagao e pelo uso da linguagem de consulta e manipulagao de dados SOL.

Entretanto, existem aplicagoes em sistemas de informagao que requerem muito mais recursos de armazenamento e manipulagao de dados do que as tabelas do modelo relacional, em especial aplicagoes Web e de cunho cientffico que processam grandes quantidades de dados em formatos diversos, com as atuais tendencias como ***Big Data, Internet of Things*** e ***Data***

***Science.***

Assim, varios modelos de banco de dados nao relacionais vem surgindo no mercado, sendo denominados de NoSOL, termo traduzido como "Nao SOL" ou "Nao somente SOL" (de ***Not Only*** SOL).

##### SAO, DE FATO, BANCOS DE DADOS QUE NAO ADOTAM O MODELO RELACIONAL DE DADOS E, PORTANTO, NAO USAM A LINGUAGEM SQL, EMBORAALGUNS POSSUAM IMPLEMENTACQES DO COMAN DO SELECT DA SQL PARA FINS DE COMPATIBILIDADE DE LINGUAGEM DE CONSULTA COM OS BANCOS DE DADOS RELACIONAIS.

0 estudo de bancos de dados NoSQL esta fora do escopo deste tema, constituindo-se em um tema a parte pela diversidade dos seus conceitos e de suas tecnologias.



**m Fonte:** Shutterstock I **Por:** Andrey Suslov

E importante conhecer a importancia dessa tendencia dos SGBDs, como demonstra o site DB­ Engines Ranking, que apresenta mais de uma duzia de modelos de bancos de dados NoSQL

com seus principais produtos, muitos deles multimodelos.

##### 0 SAIBA MAIS

**Confira a lista de multimodelos**

Chave-Valor: Redis, Amazon DynamoDB, Microsoft Azure CosmosDB.

Documentos: MongoDB, Amazon DynamoDB, Microsoft Azure CosmosDB.

Series temporais: lnfluxDB, KDB+, Prometheus.

Grafos: Neo4J, Microsoft Azure CosmosDB, ArangoDB.

Orientado a objetos: lnterSystems Cache. Versant Object Database, ObjectStore.

Motores de busca: Elasticsearch, Splunk, Solr.

RDF (Resource Description Framework): Marklogic, Apache Jena, Virtuoso.

Colunar: Cassandra, HBase, MicrosoftAzure CosmosDB.

Multivalores: Adabas, UniData/UniVerse, jBASE.

XML nativo: Marklogic, Oracle Berkeley DB, Virtuoso.

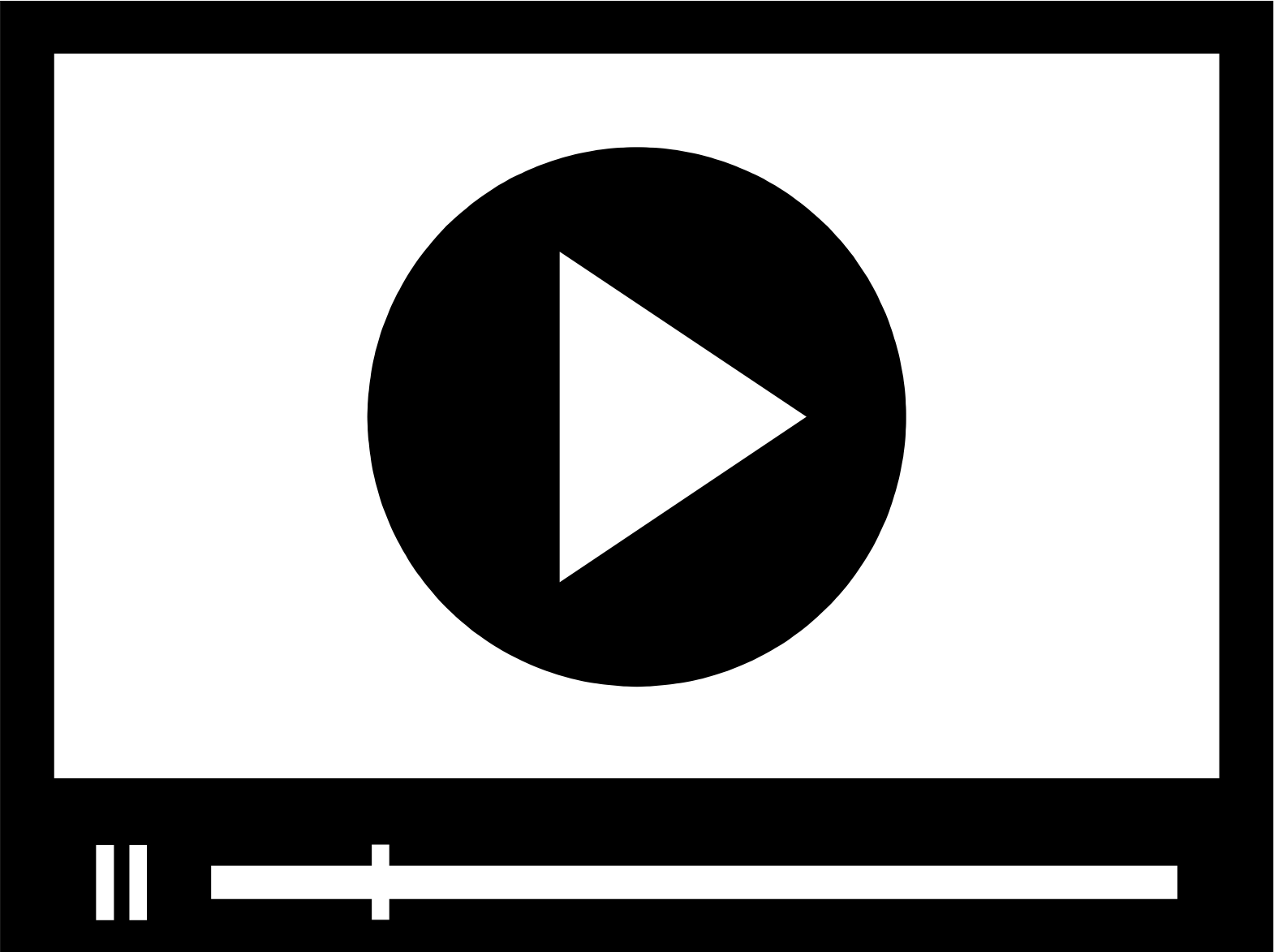
Eventos: Event Store, IBM DB2 Event Store, NEventStore.

Conteudos: JackRabbit, ModeShape.

Navegacional: IMS, IDMS.

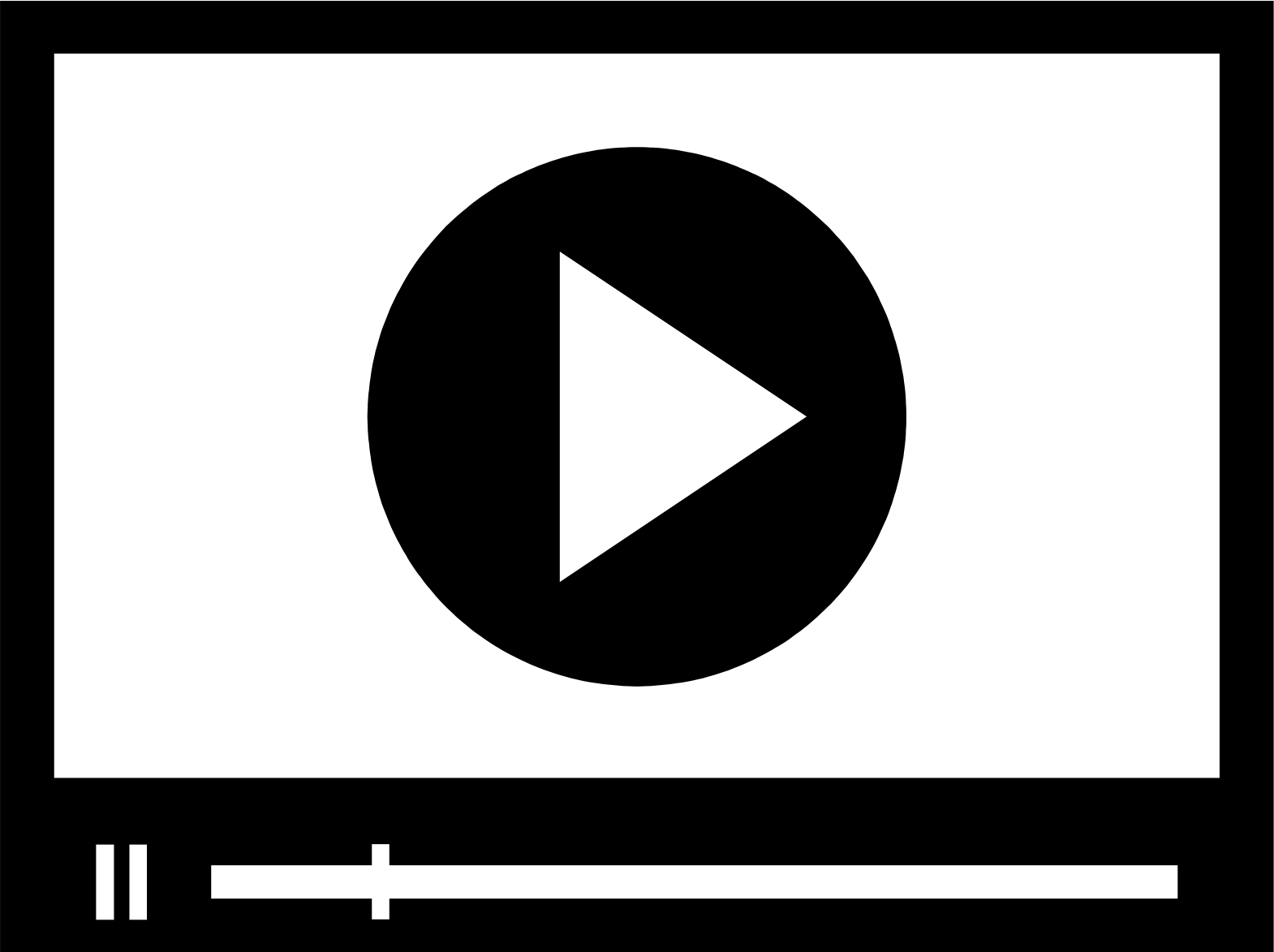
**t9 COMENTARIO**

0 modelo navegacional e exatamente aquele dos prim6rdios dos sistemas de banco de dados, da decada de 1960, antes do advento do modelo relacional, cujos produtos ainda continuam sendo utilizados, principalmente em sistemas de informac;ao legados daquela epoca.



Para fechar o modulo, neste vfdeo o professor Sidney Ventury aprofundara o conteudo falando um pouco mais sabre as vantagens que o Banco de Dados trouxe em relac;ao ao antigo sistema de arquivos.





## 0 MODELO RELACIONAL DE BANCO DE DADOS.



VERIFICANDO O APRENDIZADO

1. AO FINAL DESTE MODULO, E UM BOM MOMENTO PARARETORNARA QUEST A.O LEVANT ADA DURANTE A DESCRICA.O DOS MODELOS DE COMPUTACA.O EM SISTEMAS DE INFORMACA.O. QUAL MODELO E 0 MAIS EFICIENTE PARA UMAAPLICACA.O NA MESMA LINGUAGEM DE

**PROGRAMACAO, COM O MESMO VOLUME DE DADOS, SOB A MESMA INFRAESTRUTURA DE HARDWARE?**

* 1. Modelo monolftico com os dados junta dos programas.
  2. Modelo de sistema de arquivos, com os programas acessando os dados atraves de m6dulos internos de gerencia de arquivos.
  3. Modelo de sistema de banco de dados, com os programas acessando os dados atraves de

**um SGBD.**

* 1. Modelo de sistema de informac;;ao na Web, com os programas acessando os dados na nuvem.

1. QUAL O NOME DA LINGUAGEM DE CONSULT A E MANIPULACA.O DE BANCO DE DADOS QUE SE TORNOU SINONIMO DO MODELO RELACIONAL DE BANCO DE DADOS?
   1. PostgreSQL
   2. MySQL C)SQL
2. SOL Server

GABARITO

* 1. Ao final deste modulo, e um born momento para retornar a questao levantada durante a

descric;ao dos modelos de computac;ao em sistemas de informac;ao. Qual modelo e o mais eficiente para uma aplicac;ao na mesma linguagem de programac;ao, com o mesmo

volume de dados, sob a mesma infraestrutura de hardware?

A alternativa **"A"** esta correta.

Consideradas as mesmas condi<;6es de aplica<;ao, volume de dados e infraestrutura, conforme o

enunciado, o modelo monolftico sera mais eficiente, pois todo o processamento sera realizado na

velocidade de processamento junto a mem6ria principal **(RAM),** que e muito mais rapida do que o acesso a mem6ria secundaria ou a nuvem.

Na verdade, as alternativas dessa questao estao na sequencia de eficiencia dos modelos de computa<;ao sob as mesmas condi96es. lsso nao significa que os bancos de dados sao ineficientes, pois eles se aplicam adequadamente a cenarios com volumes de dados que nao cabem na mem6ria principal, como e o caso de sistemas de informa<;ao corporativos.

* 1. **Qual o nome da linguagem de consulta e manipulac;ao de banco de dados que se tornou sinonimo do modelo relacional de banco de dados?**

A alternativa **"C** " esta correta.

Todas as demais alternativas referem-se a nomes de SGBDs que implementam o modelo relacional e, portanto, implementam a linguagem SQL *(Structured Query Language),* chegando a confundir o nome do produto com a linguagem.

### M6DULO2

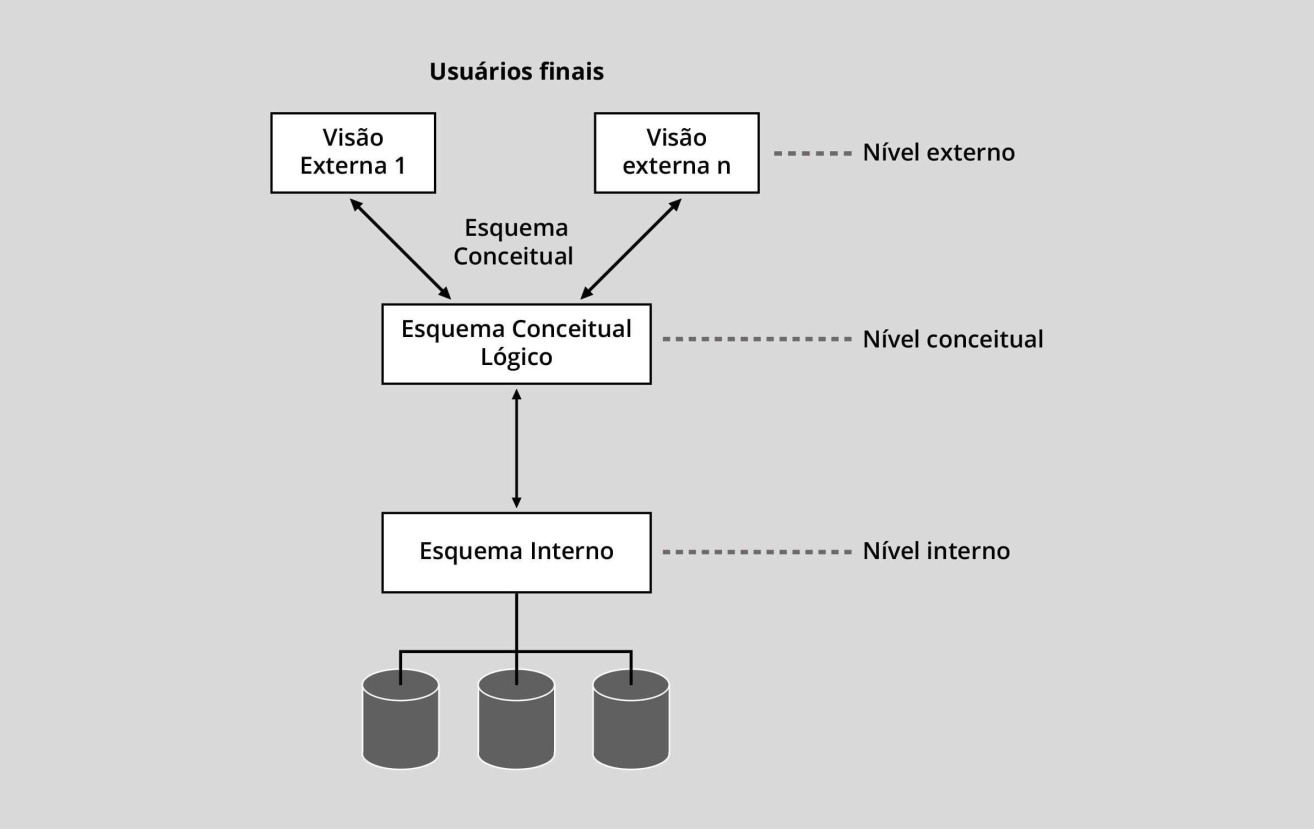
0 ldentificar as caracterfsticas dos sistemas de banco de dados (SBD)

### DIFERENCAS ENTRE SISTEMA DE ARQUIVOS E SISTEMA DE BANCO DE DADOS

No modulo anterior, verificamos que uma caracterfstica primordial que distingue os sistemas de banco de dados dos sistemas baseados de arquivos ea independencia dos dados em rela<;ao a programas. Essa, na verdade, e apenas uma especie da caracterfstica dos sistemas de banco de dados denominada independencia de dados.

0 conceito de independencia de dados deriva da arquitetura de tres esquemas do banco de dados que separa as aplica96es dos usuarios finais do banco de dados ffsico armazenado.

A figura a seguir e uma adapta9ao da arquitetura ANSVSPARC, que considera tres niveis de esquemas: o **nfvel externo,** que contem o esquema conceitual externo, integrando as diferentes vis6es dos usuarios das aplica96es; o **nfvel conceitual,** o qual contem o esquema conceitual I6gico ou de implementa9ao, descrevendo a estrutura I6gica do banco de dados; e o **nfvel interno,** que contem o esquema interno ou ffsico do banco de dados armazenado em disco.



**111 Fonte:** Tanaka (2018)

Essa arquitetura serve para definir dais tipos de independencia de dados, alem da independencia entre dados e programas ja estudada:

lndependencia I6gica de dados

Consiste na capacidade de se alterar o esquema conceitual I6gico, par exemplo, acrescentando um item de dado, sem alterar o esquema conceitual externo, isto e, as vis6es externas dos usuarios atraves dos programas de aplica9ao.

lndependencia ffsica de dados

Consiste na capacidade de se alterar o esquema interno, por exemplo, reorganizando os arquivos ffsicos que armazenam os dados, sem alterar o esquema conceitual 16gico e, em consequencia, o esquema conceitual externo.

Outras caracterfsticas, entre muitas, que diferenciam os sistemas de bancos de dados dos sistemas de arquivos sao descritas a seguir.

NATUREZA AUTOCONTI DA

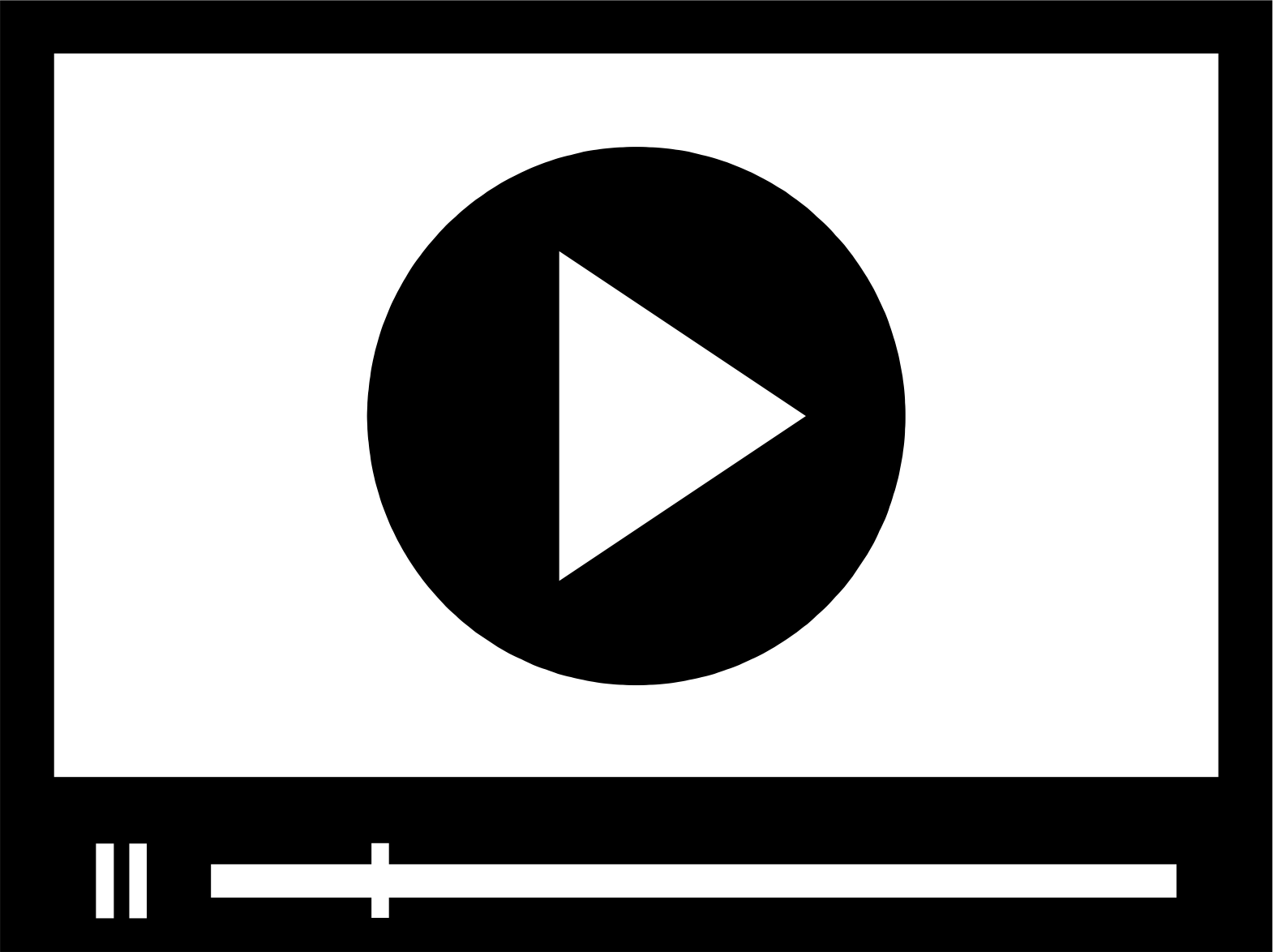
Significa que, alem dos dados, o SBD contem a descric;ao completa de suas estruturas e restric;6es. Como se vera na arquitetura do SGBD, a descric;ao da estrutura e das restric;6es de dados, conhecida como metadados, isto e, dados que descrevem dados, e armazenada no catalogo do sistema de banco de dados.

ABSTRACAO DE DADOS

Permite a representac;ao conceitual dos dados por meio de modelos de dados que ocultam detalhes de armazenamento e implementac;ao, os quais nao interessam aos diferentes usuarios, dando suporte a multiplas vis6es 16gicas dos dados ea independencia de dados.

SUPORTE AO COMPARTILHAMENTO DE DADOS E PROCESSAMENTO DE TRANSACC>ES CONCORRENTES

Permite que multiplos usuarios acessem o banco de dados simultaneamente.



Neste vfdeo, o professor Sidney Ventury refon;ara pontos importantes como os metadados ea arquitetura de tres esquemas.



Os modelos de dados que suportam a abstra9ao de dados e permitem sua independencia 16gica e ffsica sao classificados em modelos ffsicos, 16gicos e conceituais.

##### MODELOS FiSICOS

Descrevem como os dados sao armazenados no computador mediante informa96es como tipos de arquivos, formatos e ordena9ao de registros, caminhos de acesso. Sao as varias formas de estruturas de arquivos que dependem do SGBD e do sistema operacional em que estao instaladas.

##### MODELOS LQGICOS

Sao aqueles que representam, de maneira abstrata, a implementa9ao dos bancos de dados, ocultando detalhes de como os dados sao armazenados e acessados no disco. 0 modelo 16gico mais tradicional, largamente utilizado na grande maioria dos sistemas de informa9ao organizacionais, e o modelo relacional criado por Edgar Codd, mencionado no modulo anterior. A estrutura de dados do modelo relacional ea tabela (rela9ao matematica) e deve ser estudada em profundidade em tema especffico, assim como as implementa96es de SGBD relacional. Como vimos, ha outros modelos 16gicos de implementa9ao do banco de dados entre aqueles que atualmente se denominam de modelos nao relacionais ou NoSQL.

##### MODELOS CONCEITUAIS

Sao aqueles que representam a visao dos dados do ponto de vista do usuario final, no nfvel de abstra9ao mais pr6ximo do mundo real. Dentre esses, destaca-se o modelo de entidades e relacionamentos, criado pelo pesquisador Peter Chen, em 1976. 0 chamado modelo ER, segundo o pr6prio autor, foi criado para prover um melhor entendimento do modelo relacional, de modo a servir como uma etapa inicial no processo de projeto de banco de dados, denominada de modelagem conceitual dos dados.

Vale destacar que a modelagem conceitual de dados realizada com o modelo ER foi incorporada na linguagem UML *(Unified Modeling Language),* criada no final da decada de 1990 para uniformizar a modelagem orientada a objetos sob a forma do modelo de classes, em que as entidades sao representadas pelas classes de objetos e os relacionamentos pelas associac;6es entre as classes. Na pratica atual do processo de desenvolvimento de sistemas de informac;ao, e comum a utilizac;ao do modelo de classes da UML como substituto do modelo ER nas fases iniciais do projeto de banco de dados.

##### ATENCAO

Nesse ponto, vale a pena revisitar a definic;ao de banco de dados citada no infcio do primeiro modulo, complementando-a com as principais caracterfsticas aqui mencionadas, Assim, podemos definir banco de dados coma uma colec;ao autodescritiva de dados relacionados, com significado 16gico inerente, que pode ser manipulada concorrentemente por usuarios com vis6es diferentes.

### VANTAGENS E DESVANTAGENS DA ABORDAGEM DE BANCO DE DADOS

A abordagem de SBD possui funcionalidades que conferem vantagens adicionais em relac;ao a sistemas sem banco de dados, alem das caracterfsticas que a diferenciam da abordagem de sistemas de arquivos. Algumas dessas funcionalidades do SBD sao listadas a seguir e podem ser objeto de estudo em temas sabre implementac;ao e administrac;ao de banco de dados.

##### CONTROLE DA REDUNDANCIA DE DADOS

Previne a possibilidade de inconsistencia dos dados, a duplicac;ao de esfon;o para manter os dados atualizados e o desperdicio de espac;o de armazenamento. 0 SGBD permite controlar o *trade off* entre o armazenamento em um unico local no banco de dados *versus* a redundancia forc;ada para melhorar o desempenho das consultas.

##### COMPARTILHAMENTO DE DADOS

Sendo os SBD multiusuarios, realizam controle de concorrencia de acesso aos dados compartilhados para garantir as propriedades de atomicidade, consistencia, isolamento e durabilidade (propriedades ACID) das transac;oes de banco de dados.

##### CONTROLE DE ACESSO

Mecanismos de seguranc;a e autorizac;ao, como senhas para usuarios e para grupos de usuarios; restric;ao de acesso a partes do banco de dados; proibic;ao de executar certas operac;oes privilegiadas; acesso de usuario restrito apenas a transac;oes autorizadas; proibic;ao de uso de software privilegiado, como o software de administrac;ao do SBD.

##### MULTIPLAS INTERFACES DE USUARIOS E APLICACOES

Prove diferentes linguagens de consulta para usuarios casuais; linguagens de programac;ao para programadores de aplicac;oes; interfaces graficas com formularios (telas) e menus para usuarios de aplicac;oes; interfaces para administrac;ao do banco de dados; interfaces de linguagem natural.

##### REPRESENTACAO DE RELACIONAMENTOS ENTRE OS DADOS

Permite representar os relacionamentos existentes entre dados no mundo real, mediante mecanismos que dependem do modelo 16gico de implementac;ao do SBD.

##### CUMPRIMENTO DE RESTRICOES DE INTEGRIDADE DOS DADOS

Previne a violac;ao de restric;oes como tipo de dado ou domfnio de valores admissfveis; unicidade de itens de dados por meio de chaves unicas; integridade referencial entre dados relacionados e restric;oes derivadas da semantica dos dados. Aqui tambem o SGBD permite controlar o *trade off* entre garantir o cumprimento automatico das restric;oes ou deixar a sua especificac;ao para os programas de aplicac;ao.

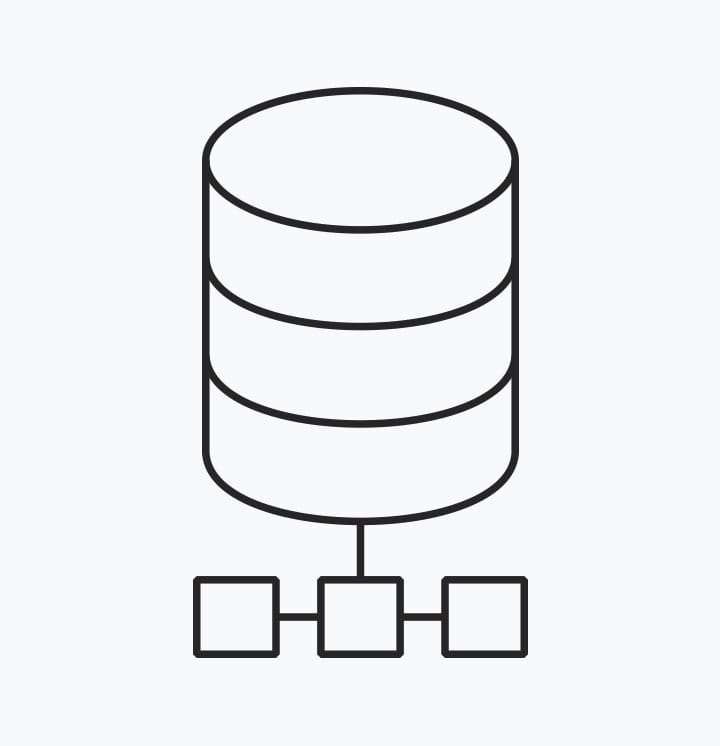
#### CAPACIDADE DE *BACKUP* E RECUPERACAO DE DADOS:

Em caso de ocorrencia de falhas de hardware ou de software, os mecanismos de c6pia de seguranc;a *(backup)* e posterior restaurac;ao *(recovery)* garantem a consistencia de estado do banco de dados antes e depois da falha.

#### COMPARTILHAMENTO DE DADOS ENTRE MULTIPLOS USUARIOS SIMULTANEOS

E uma caracteristica primordial do SBD e tern como resultado o controle de concorrencia das transac;oes.

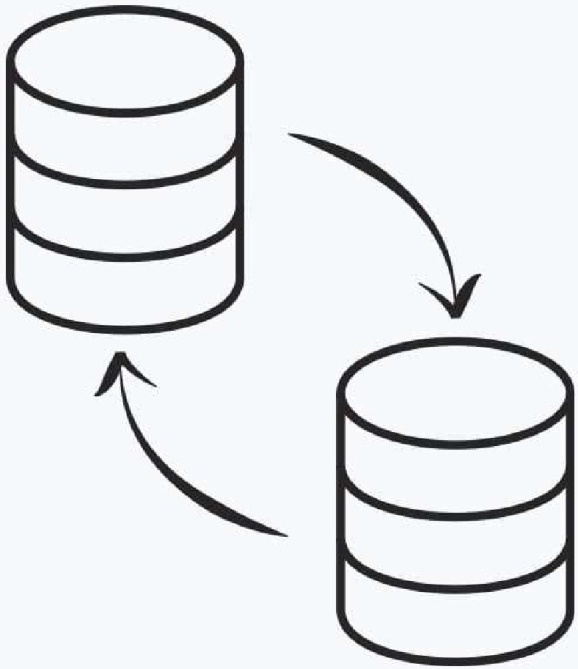
E responsabilidade do SGBD garantir as propriedades das transac;oes, conhecidas pela sigla ACID, relaxando-as quando necessario para manter o desempenho sob seu controle e para nao violar a integridade do banco de dados.



Fonte:Shutterstock

**ATOMICIDADE *(ATOMICITY)***

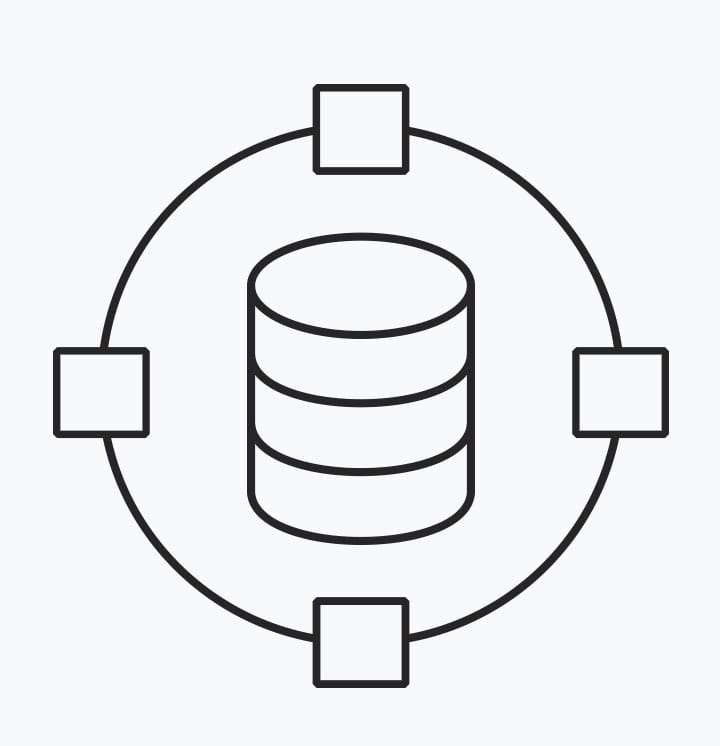
Cada transac;ao e tratada como uma unidade composta de uma sequencia de operac;oes, de modo que deve executar completamente com sucesso ou falhar completamente.



Fonte:Shutterstock

**CONSISTENCIA (*CONSISTENCY)***

Uma transagao s6 pode levar o banco de dados de um estado valido para outro, de acordo com suas regras de integridade.



Fonte:Shutterstock

**ISOLAMENTO *{ISOLATION)***

Cada transac;ao e isolada das demais, isto e, essa propriedade assegura que transac;6es

executadas concorrentemente levem o banco de dados ao mesmo estado que chegaria se as transac;6es fossem executadas sequencialmente.



Fonte:Shutterstock

**DURABILIDADE *(DURABILITY)***

Uma vez que a transa9ao e aceita *(committed),* o que significa que seu resultado foi gravado em mem6ria nao volatil, esse resultado permanecera valido mesmo em caso de falhas do sistema.



autor/shutterstock



autor/shutterstock

##### VANTAGENS

As vantagens decorrentes dessas funcionalidades do SBD sao varias: potencial para o estabelecimento de padroes de uso dos dados na organiza9ao; redu9ao do tempo de desenvolvimento de aplica96es; flexibilidade na manuten9ao dos dados; disponibilidade dos dados atualizados no ambito de toda a organiza9ao; economia de escala, entre outras.



autor/shutterstock

##### DESVANTAGENS

Agora, vejamos as desvantagens da abordagem de banco de dados. Como ja foi observado, a presen9a do SGBD como um software intermediario entre as aplica96es e os dados armazenados provoca uma sobrecarga no desempenho do sistema como um todo. Alem disso, ha de se levar em conta o custo e o esfor90 adicional na capacita9ao e no oferecimento de funcionalidades sofisticadas como as ja citadas anteriormente.

Assim, pode-se dizer que e preferfvel **nao usar a abordagem de SBD** em algumas aplica96es, tais como:

Aplica96es muito simples com dados estaticos e bem definidos, que se espera que nao sejam alterados (exemplo: censo demografico);

Aplica96es de tempo real com requisitos rfgidos os quais nao possam ser atendidos com o

*overhead* causado pelo SGBD (exemplo: controle de trafego aereo);

Sistemas embarcados, com poucos dados e com requisitos estritos de tempo real (exemplo: piloto automatico);

Sistemas monousuarios de uso sem concorrencia, tfpicos de aplica96es em *desktop*

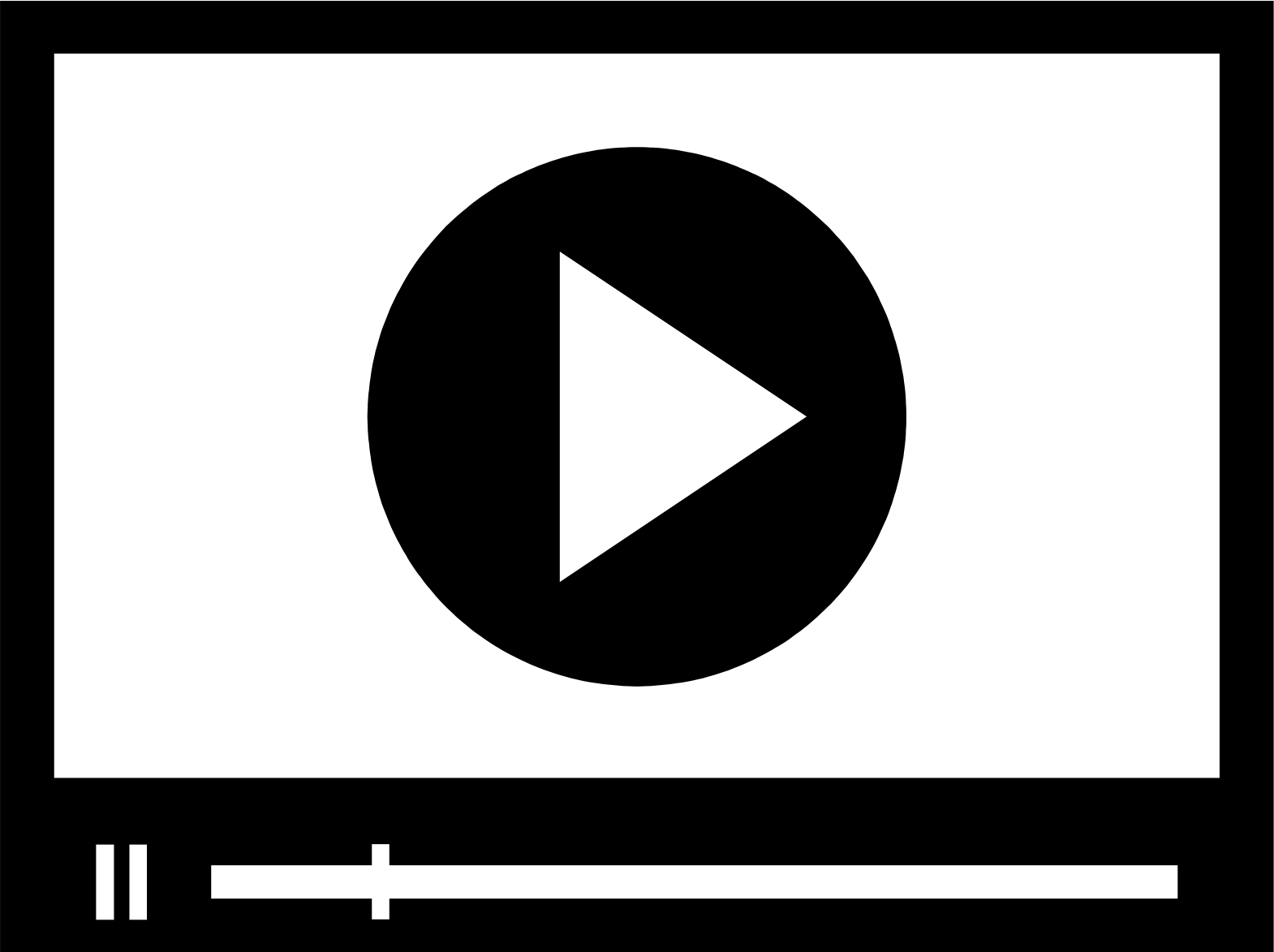
(exemplo: prontuario eletr6nico de um Cmico consult6rio medico).



Fonte:Shutterstock

**m Fonte:** Shutterstock I **Por:** Dmi T

Poderiam ser enquadradas nessa lista as aplica96es que requerem dados volumosos com dinamica nao processavel por SGBDs tradicionais, como a Internet das Coisas (loT, de *Internet of Things),* porem, o advento de bancos de dados NoSQL, com modelos de dados especfficos para tratamento de *Big Data,* busca a atender a esses tipos de aplica96es incompatfveis com os modelos tradicionais de SGBD.

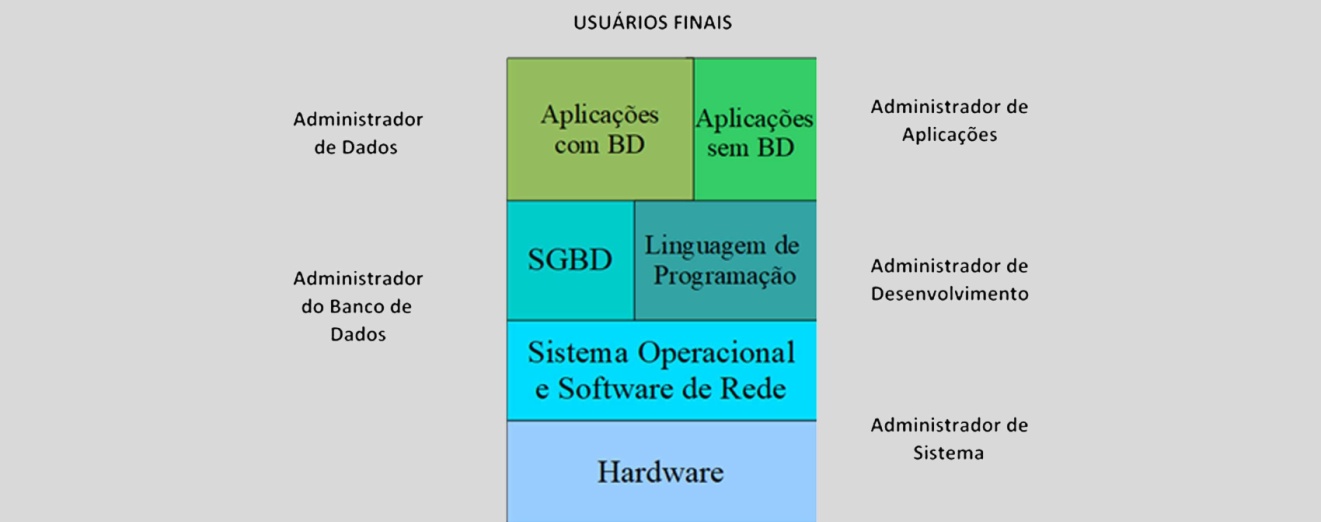


## FUNCIONALIDADES E PROPRIEDADES DAS TRANSACOES DO SISTEMA DE BANCO DE DADOS.



PAPEIS EM SISTEMAS DE BANCOS DE DADOS

A figura a seguir mostra as camadas de um sistema de computac;ao desde o hardware, onde sao armazenados os arquivos com programas e dados, ate as aplicac;oes disponiveis aos usuarios finais, que podem ser desenvolvidas com ou sem a utilizac;ao de um SGBD. Na ilustrac;ao, podemos ver tambem a existencia de diferentes papeis de usuarios ao longo das camadas de software.



***ii* Fonte:** Tanaka (2018)

Em um sistema de computac;ao corporativo de grandes organizac;oes, e possivel separar os diferentes papeis desempenhados como mostra a figura.

##### USUARIOS FINAIS

Beneficiarios dos sistemas que rodam no sistema de computac;ao, seja por aplicac;oes desenvolvidas com a abordagem de banco de dados ou nao.

##### ADMINISTRADOR DE APLICACQES

Func;ao tecnico-gerencial responsavel pela manutenc;ao dos sistemas de aplicac;ao e pelo suporte aos seus usuarios, podendo ser exercida por varios administradores. Exemplo: administrador do sistema integrado de gestao empresarial, conhecido pela sigla ERP *(Enterprise Resource Planning).*

##### ADMINISTRADOR DE DESENVOLVIMENTO

Func;ao tecnica que pode ser desdobrada em equipes de desenvolvimento do sistema de aplicac;ao, dependendo do porte e da complexidade do sistema, que se utilizam das ferramentas disponiveis no ambiente de desenvolvimento de sistemas.

##### ADMINISTRADOR DE DADOS

Func;ao gerencial responsavel pelo ambiente de dados da organizac;ao, que define politicas e responsabilidades sobre os recursos de dados, assim como as regras do neg6cio e os padr6es

de dados a serem seguidos no desenvolvi mento.

##### ADMINISTRADOR DO BANCO DE DADOS

Fungao tecnica responsavel pela criagao e manutengao dos bancos de dados no SGBD, dando suporte as equipes de desenvolvimento no tocante aos objetos dos bancos de dados.

##### ADMINISTRADOR DE SISTEMA

Responsavel por manter no ar o sistema de computagao como um todo, com foco no hardware e no sistema operacional, bem como nas interfaces deste com os demais softwares instalados.

**CABE OBSERVAR QUE NEM TODOS OS PAPEIS DE ADMINISTRACAO PODEM SER EXERCIDOS POR UMA UNICA PESSOA, MAS UMA PESSOA PODE EXERCER MAIS DE UM PAPEL. A DISTRIBUICAO DESSES PAPEIS POR PESSOAS E EQUIPES DEPENDERA DO PORTE DA ORGANIZACAO E DO SISTEMA DE COMPUTACAO, VARIANDO DESDE UMA UNICA PESSOA EXERCENDO TODOS OS PAPEIS DE ADMINISTRACAO ATE TODO UM DEPARTAMENTO DE Tl ENCARREGADO DAADMINISTRACAO E DO SUPORTE AOS DIVERSOS SISTEMAS EM OPERACAO.**

## VERIFICANDO O APRENDIZADO

1. ANALISE AS SEGUINTES AFIRMACOES E RESPONDA QUAL ALTERNATIVA CORRESPONDE A CARACTERiSTICAS QUE DISTINGUEM OS SISTEMAS DE BANCO DE DADOS DOS SISTEMAS DE ARQUIVOS.
2. **NATUREZAAUTOCONTIDADOS DADOS**
3. **VOLATILIDADE DOS DADOS**

Ill. **ABST RA<;AO DE DADOS**

* 1. Somente as afirmac;6es I e II.
  2. Somente as afirmag6es I e Ill.
  3. Somente as afirmac;6es II e Ill.
  4. Todas as afirmag6es estao corretas.

1. QUAL DAS ALTERNATIVAS ABAIXO NAO E UMA FUNCIONALIDADE DOS SISTEMAS DE BANCO DE DADOS?
   1. Representac;ao de relacionamentos entre os dados.
   2. Controle de redundancia de dados.
   3. Compartilhamento de dados.
   4. Armazenamento de dados estaticos.

GABARITO

* + 1. **Analise as seguintes afirma oes e responda qual alternativa corresponde a caracteristicas que distinguem os sistemas de banco de dados dos sistemas de arquivos.**

1. **Natureza autocontida dos dados**
2. **Volatilidade dos dados**

Ill. **Abstra ao de dados**

A alternativa **"B** " esta correta.

A natureza autocontida dos dados, com o armazenamento junta aos metadados, e a abstragao de dados, permitindo a independencia dos dados, sao caracterfsticas que diferenciam os SBD dos sistemas de arquivos. Ja a volatilidade dos dados nao e uma caracterfstica dos SBD.

* + 1. **Qual das alternativas abaixo nao e uma funcionalidade dos sistemas de banco de dados?**

A alternativa **"D** "esta correta.

O armazenamento de dados estaticos e uma das situagoes em que nao e recomendavel o uso da abordagem de banco de dados, pois nao se trata de uma funcionalidade tfpica dos SBDs, coma as demais alternativas.

# M6DULO 3



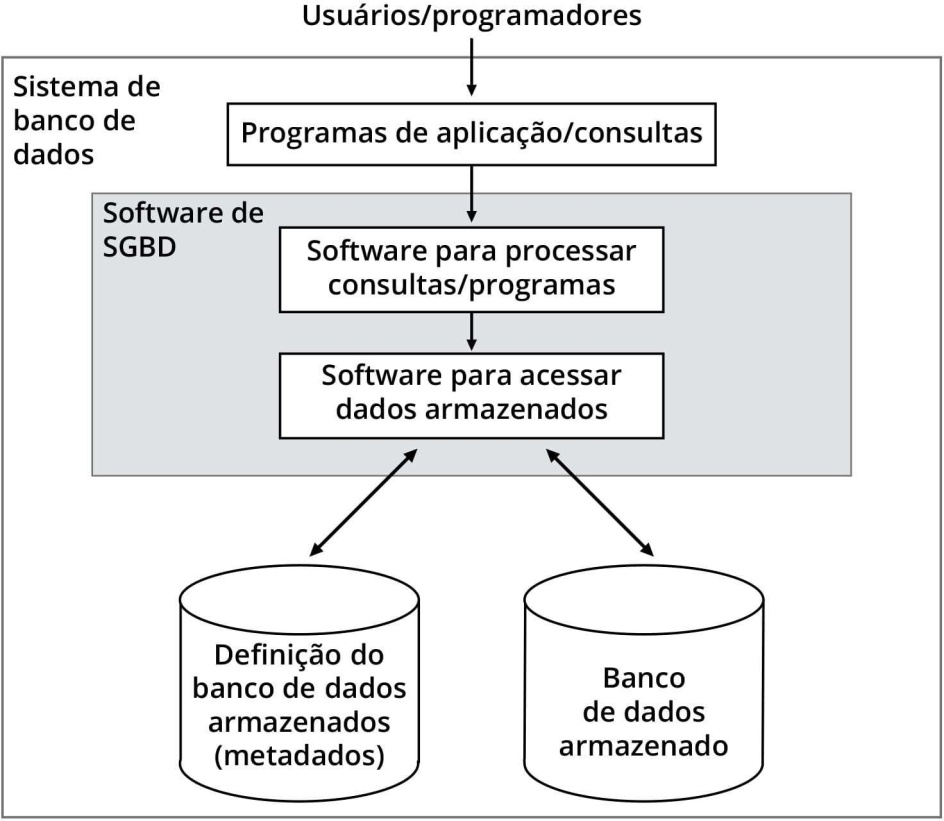
0 Descrever a arquitetura dos sistemas de gerencia de banco de dados (SGBD)

# COMPONENTES DE UM SISTEMA DE BANCO DE DADOS

A figura a seguir mostra uma visao simples do ambiente de banco de dados, compreendendo programas de aplicagao ou consultas de usuarios que acessam os dados e metadados armazenados em disco atraves do SGBD. Este, par sua vez, compoe-se, simplificadamente, de um software para processar consultas e programas e outro para acessar os dados e metadados armazenados.

Nota-se, na figura, que nao se confundem o SBD e o SGBD, vista que este e um componente daquele que engloba tambem os programas e as consultas, bem coma os dados e metadados armazenados.

A fronteira do SBD engloba os programas que implementam as aplicagoes, bem coma as consultas provenientes de usuarios com acesso a linguagens e interfaces de consulta. 0 SGBD e o software intermediario, uma caixa preta a ser aberta no pr6ximo modulo. A fronteira do SBD envolve os metadados armazenados no catalogo (as vezes chamado de dicionario de dados) e o pr6prio conteudo armazenado no banco de dados.



aJ Fonte: Elmasri e Navathe (2019)

COSTUMA-SE DEFINIR METADADOS COMO 0 ESQUEMA DO BANCO DE DADOS, ESTRUTURADO DE ACORDO COM O MODELO LOGICO DE IMPLEMENTACAO. ASSIM, MODELAR UM BANCO DE DADOS CONSOANTE AO MODELO LOGICO DE IMPLEMENTACAO EQUIVALE A ESQUEMATIZAR 0 BANCO DE DADOS CONFORME OS CONSTRUTORES DESSE MODELO. POR EXEMPLO, NO MODELO

RELACIONAL, 0 ESQUEMA E COM POSTO POR

TABELAS E SUAS COLUNAS. CADA COMANDO DE DEFINICAO DE DADOS, CRIANDO, ALTERANDO OU REMOVENDO UMA TABELA, PROVOCA UMA MUDANCA NO ESQUEMA DO BANCO DE DADOS.

Por outro lado, chama-se de estado ou instancia o conteudo do banco de dados armazenado em um momenta. Cada manipula<;ao no banco de dados mediante comandos de inser<;ao, atualiza9ao ou remo9ao de dados provoca uma mudan<;a de estado, gerando uma nova instancia do banco de dados.

#### CURIOSIDADES SOBRE BANCO DE DADOS

Como vimos, o termo **banco de dados** e correntemente usado com o sentido do original, em ingles, *database,* cuja origem, segundo o *Oxford English Dictionary,* remonta a 1962 num

relat6rio de uma empresa na California. Raras sao as referencias ao termo *data bank coma,* por exemplo, no mencionado artigo de Edgar Codd sobre o modelo relacional (CODD, 1970).

Em espanhol, usa-se *base de datos,* no trances *base de donnees,* enquanto em alemao se diz

*Datenbank,* e em italiano *banca dati.*

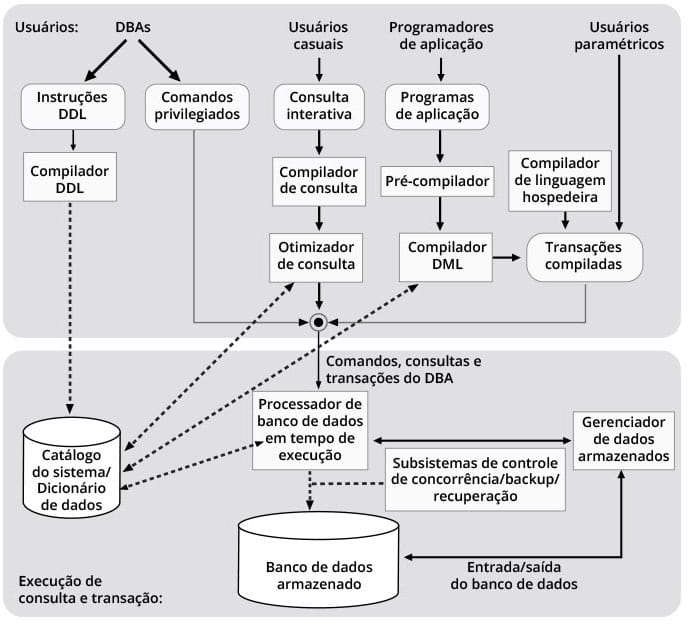
Em portugues, existem os dois termos, sendo banco de dados usado no sentido geral do

ambiente que engloba o sistema, como na figura anterior, enquanto base de dados tern o sentido mais restrito do conteudo do banco, isto e, corresponde ao estado ou a instancia do banco de dados. Por exemplo, costuma-se referir a base de dados da Receita Federal como o conjunto de

dados armazenados sobre os contribuintes e nao como o sistema que gerencia os dados.

# M6DULOS DE UM SISTEMA DE GERENCIA DE BANCO DE DADOS

A figura a seguir ilustra, em detalhes, os diversos m6dulos que compoem um SGBD, sendo a parte superior correspondente ao processamento das consultas e aplica96es, ea parte inferior ao acesso a metadados e dados armazenados (a base de dados).



**rl Fonte:** Elmasri e Navathe (2019)

Vamos conferir a descrii;ao desses m6dulos.

##### USUARIOS

Da esquerda para a direita, vemos as usuarios, comei;ando com as administradores de banco de dados (ABO, do ingles OBA- *Database Administrator).*

# ABD

OABD usa comandos da linguagem de definii;ao de dados (LDD, do ingles DDL- *Data Definition Language)* para criar, alterar au remover objetos do banco de dados (comandos CREATE, ALTER, DROP no padrao SQL), as quais ficam armazenados no catalogo do sistema, que contem as metadados.

0 ABO tambem possui privilegio para executar comandos de controle de dados, conhecidos par alguns autores coma linguagem de controle de dados (LCD, do ingles DCL- *Data Control Language),* para conceder e revogar permiss6es de acesso aos dados (comandos GRANT e REVOKE no padrao SQL), entre outros. Note que o destine desses comandos privilegiados e um n6 central do SGBD, par onde passam todos as comandos destinados ao processador de banco de dados em tempo de execui;ao, denominado de processador de *runtime* au *runtime engine.*

##### USUARIOS CASUAIS

Seguindo para a direita, vemos as usuarios casuais, as quais fazem consultas interativas atraves de uma interface para consultas *ad hoc,* au seja, consultas nao programadas previamente. As consultas sao feitas tipicamente com o comando SELECT no padrao SQL do modelo relacional e

provido de modo similar em outros modelos de implementa<;ao de banco de dados. Esses comandos sao compilados por um compilador da linguagem de consulta e passam por um otimizador de consulta antes de chegar ao n6 central do SGBD.

##### PROGRAMADORES DE APLICACCES

Mais a direita, aparecem os programadores de aplica96es que escrevem os programas em uma linguagem de programa9ao hospedeira, como por exemplo, Java, PHP ou Python, nos quais

estao embutidos comandos de consulta (SELECT), inser9ao, atualiza9ao e exclusao de dados em uma linguagem de manipula<;ao de dados (LMD, do ingles DML- *Data Manipulation Language).*

No padrao SQL, esses comandos sao, respectivamente, INSERT, UPDATE e DELETE. Note que eles manipulam dados, diferentemente dos comandos da linguagem de defini<;ao de dados (CREATE, ALTER, DROP), que manipulam metadados.

##### PROGRAMAS DE APLICACAO

Os programas de aplica9ao, portanto, possuem comandos hibridos, da linguagem hospedeira e da linguagem de consulta e manipula9ao de dados.

##### PRE-COMPILADOR

Os programas de aplica9ao sao processados, inicialmente, por um pre-compilador, responsavel por separar os comandos e os repassar para os compiladores das respectivas linguagens.

##### COMPILADORES

Cabe a esses compiladores produzir o c6digo das aplica96es sob a forma de transa96es executaveis, que ficam a disposi<;ao dos usuarios parametricos.

##### USUARIOS PARAM ETRICOS

Os usuarios parametricos sao assim chamados porque interagem com o sistema atraves de parametros passados em interfaces apropriadas. Por exemplo, um agente de viagens faz uma reserva de passagem aerea passando ao sistema os dados do passageiro, data e hara da viagem, numero do voo, numero do assento e outros parametros necessarios para efetivar a reserva.

##### TRANSACCES COMPILADAS

Lima vez executadas as **transac;oes compiladas,** assim coma os demais comandos provenientes de usuarios ou aplica96es, sao passadas ao n6 central do SGBD para posterior processamento do processador de *runtime.*

**Atenc;ao!** Antes de prosseguir na parte de baixo da figura, note que alguns processos da parte de cima, como o otimizador de consulta e o compilador da LMD, estao ligados ao catalogo por linhas tracejadas, que denotam fluxos de controle, enquanto as linhas cheias representam fluxos de dados, alem de controle. lsso e necessario porque as referencias a objetos do banco de dados existentes no catalogo devem ser consistentes com os objetos, criados e mantidos pelo ABO mediante comandos da LDD.

##### PROCESSAMENTO DO ACESSO AOS DADOS

**ARMAZENADOS**

Prosseguindo para a parte do processamento do acesso aos dados armazenados, observamos que o processador de *runtime* e o corac;ao do SGBD. De fato, o diferencial dos produtos de SGBD reside na eficiencia e na funcionalidade desse processador, segredo industrial em muitos dos SGBDs proprietarios, a ponto de exigir rigorosos termos de confidencialidade das pessoas que tern acesso ao seu c6digo.

##### ACESSO A BASE DE DADOS

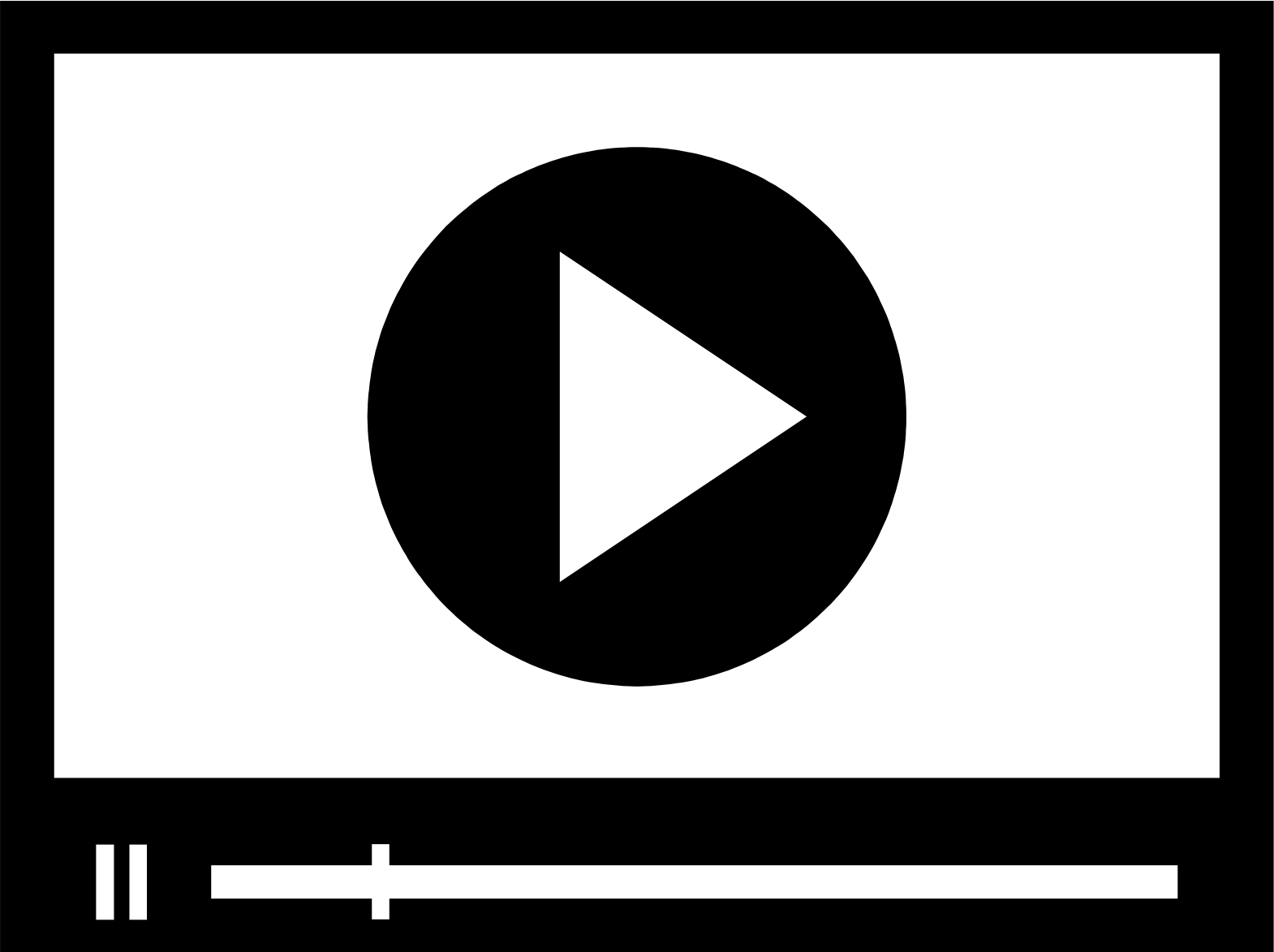
0 processador de *runtime* tambem precisa acessar e, por vezes, modificar o catalogo, dependendo da natureza dos comandos, das consultas ou transac;oes que estiver processando. Como mostra a figura, ele acessa a base de dados diretamente, sob controle de subsistemas de controle de concorrencia, *backup* e *recovery.* Quando precisa realizar operac;oes de entrada e safda (gravac;ao e leitura) na base de dados, o processador de *runtime* se vale de um gerenciador de dados armazenados.

A descric;ao dada, embora simplificada, denota a complexidade de um SGBD, software que evoluiu desde a decada de 1960 a ponto de se tornar um recurso robusto, eficiente e indispensavel para a maioria das organizac;oes privadas ou publicas, principalmente com base no modelo relacional de banco de dados.

Existe uma grande quantidade de SGBDs disponfveis no mercado, como mostrou o DB-Engines Ranking, ao listar mais de 300 SGBDs, sendo cerca de 130 deles do modelo relacional.

##### 0 SAIBA MAIS

A excelente entrada na Wikipedia, intitulada *Comparison of relational database management systems,* faz uma comparac;ao de diversas caracterfsticas de SGBDs relacionais, contendo informac;oes detalhadas sobre mais de 60 produtos.



Neste vfdeo, o professor Sidney Ventury abordara pontos importantes como *runtime,* transa96es compiladas, entre outros.



### UM EXEMPLO DE SGBD RELACIONAL: POSTGRESQL

Como vimos anteriormente na evolugao hist6rica dos SGBDs, o PostgreSQL originou-se do projeto Ingres da Universidade da California em Berkeley, na decada de 1970, tendo sido sucessivamente renomeado de Postgres *(Post Ingres),* depois Postgres 95 e, finalmente, PostgreSQL.

Embora fossem implementac;6es do modelo relacional proposto por Edgard Codd (CODD, 1970), o Ingres e o Postgres originalmente usavam uma linguagem diferente da SQL, adotada pela IBM, conhecida como Quel *(Query Language).* Por muitos anos, a Quel persistiu nesses SGBDs em concorrencia direta com a SQL da IBM, ate que o ANSI *(American National Standard Institute)* e depois a ISO *(International Standard Organization)* reconheceram a SQL como a linguagem padrao para o banco de dados relacional. A partir de entao, a SQL foi implementada no Postgres, passando a ser chamada de PostgreSQL.

0 PostgreSQL, alem de ser reconhecido como *the V1,0r/d's most advanced open source relational database,* possui uma completa documentac;ao igualmente reconhecida como referenda global para os conceitos de SGBD relacional. Nao e sem razao que o PostgreSQL, por ser *open source* e ter licenc;a livre, e vastamente utilizado no ensino de banco de dados, alem de ser adotado comercialmente em grandes organizac;6es ao redor do mundo.

##### 0 POSTGRESQL E UM TiPICO SGBD RELACIONAL­ OBJETO, ISTO E, UM SGBD CUJA ESTRUTURA DE DADOS BASICA EA RELACAO {TABELA), POREM CONTEMPLA EXTENSQES DE TIPOS DE DADOS E CARACTERiSTICAS PROPRIAS DA ORIENTACAO A OBJETOS. ASSIM, POSSUI SUPORTE EM GRANDE PARTE AO PADRAO SQL E IMPLEMENTA EXTENSQES UTEIS COMO: CONSULTAS COMPLEXAS, GATILHOS

**{TRIGGERS), VISQES MATERIALIZADAS ATUALIZAVEIS, CONTROLE DE CONCORRENCIA MULTIVERSIONADO. ALEM DISSO, PODE SER ESTENDIDO DE MUITAS FORMAS, POR EXEMPLO, GERANDO NOVOS TIPOS DE DADOS, FUNCC>ES GENERICAS E AGREGADAS, OPERADORES, METODOS DE INDEXACAO E LINGUAGENS PROCEDURAIS.**

Em jargao de sistemas, o PostgreSQL usa um modelo de computa<;ao cliente/servidor, assim como a maioria dos SGBDs relacionais empresariais. Uma sessao PostgreSQL consiste dos seguintes processos cooperativos (programas):

Back end

Um processo servidor *(back end),* responsavel por gerenciar os arquivos do banco de dados,

aceitar conexoes por aplica96es clientes e executar a96es sobre o SBD em nome dos clientes. 0 programa servidor e chamado postgres.

Front end

As aplica96es clientes de usuarios *(front end)* que desejam executar opera96es no banco de dados. Aplica96es clientes podem ser diversas em natureza, como uma ferramenta orientada a texto, uma aplica9ao grafica, um servidor Web que acessa o banco de dados para exibir paginas Web, ou uma ferramenta especializada de manuten9ao. Algumas aplica96es clientes sao fornecidas com a distribui<;ao PostgreSQL (psql, pgadmin).



Fonte:Shutterstock

***ii* Fonte:** Shutterstock I **Por:** Ton Snoei

0 PostgreSQL roda em todos os sistemas operacionais importantes, incluindo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) e Windows. Possui todas as propriedades ACID das transag6es (atomicidade, consistencia, isolamento e durabilidade) e tern completo suporte para chaves estrangeiras, jung6es, vis6es, fung6es, *triggers,* e procedimentos

armazenados em multiplas linguagens de programagao, incluindo Java, Perl, Python, Ruby, Tel,

*CIC++,* e sua pr6pria PL/pgSQL, similar a PL/SQL do Oracle.

Como banco de dados de classe empresarial, PostgreSQL possui caracterfsticas sofisticadas como controle de concorrencia multiversao *(Multi Version Concurrency Control),* recuperagao *"point in time"* (PiTR- *Point in Time Recovery), tablespaces,* replicagao assfncrona, transag6es aninhadas *(savepoints), online/hot backups,* um sofisticado planejador/otimizador de consultas, e *wite ahead logging* (WAL) para tolerancia a falhas.

0 PostgreSQL e altamente escalavel tanto na quantidade de dados que pode gerenciar como no numero de usuarios concorrentes que pode acomodar. Existem sistemas PostgreSQL em ambientes de produgao que gerenciam centenas de usuarios simultaneos com dezenas de *terabytes* de dados armazenados.

Alem de possuir um catalogo de sistema totalmente relacional, capaz de suportar multiplos *schemas* por *database,* o catalogo do PostgreSQL e tambem acessfvel atraves do *Information Schema,* um conjunto de vis6es de metadados definido no padrao SQL.



Fonte:Shutterstock

**Fonte:** Shutterstock I **Por:** Alexander Supertramp

##### ATENCAO

PostgreSQL suporta fndices compostos, parciais e funcionais que podem usar seus metodos de armazenamento B-tree, R-tree, hash ou GiST. Este ultimo e um sofisticado *frameoork* que serve de base para muitos projetos publicos que usam PostgreSQL, como o OpenFTS (Open Source Full Text Search engine), que prove indexac;ao online de dados e ranking de relevancia para pesquisa em bancos de dados, e o PostGIS, um projeto que adiciona suporte para objetos geograficos, permitindo seu uso como banco de dados espacial para Sistemas Geograficos de lnformac;ao.

Outras caracterfsticas avanc;adas do PostgreSQL incluem heranc;a de tabelas, sistema de regras e eventos de banco de dados.

##### HERANCA DE TABELAS

Caloca um "sabor" orientado a objeto na criac;ao de tabelas, permitindo a projetistas de banco de dados derivar novas tabelas de outras tabelas, tratando-as como classe base. Esse esquema suporta tanto heranc;a simples como multipla.

##### SISTEMA DE REGRAS

Tambem chamado de sistema de reescrita de consultas, permite ao projetista de banco de dados criar regras que identificam operac;6es especfficas para uma dada tabela ou visao e dinamicamente transforma-las em operac;6es alternativas quando sao processadas (clausula INSTEAD OF).

##### SISTEMA DE EVENTOS

E um sistema de comunicac;ao interprocessos em que mensagens e eventos podem ser transmitidos entre clientes usando os comandos LISTEN e NOTIFY, permitindo tanto comunicac;ao

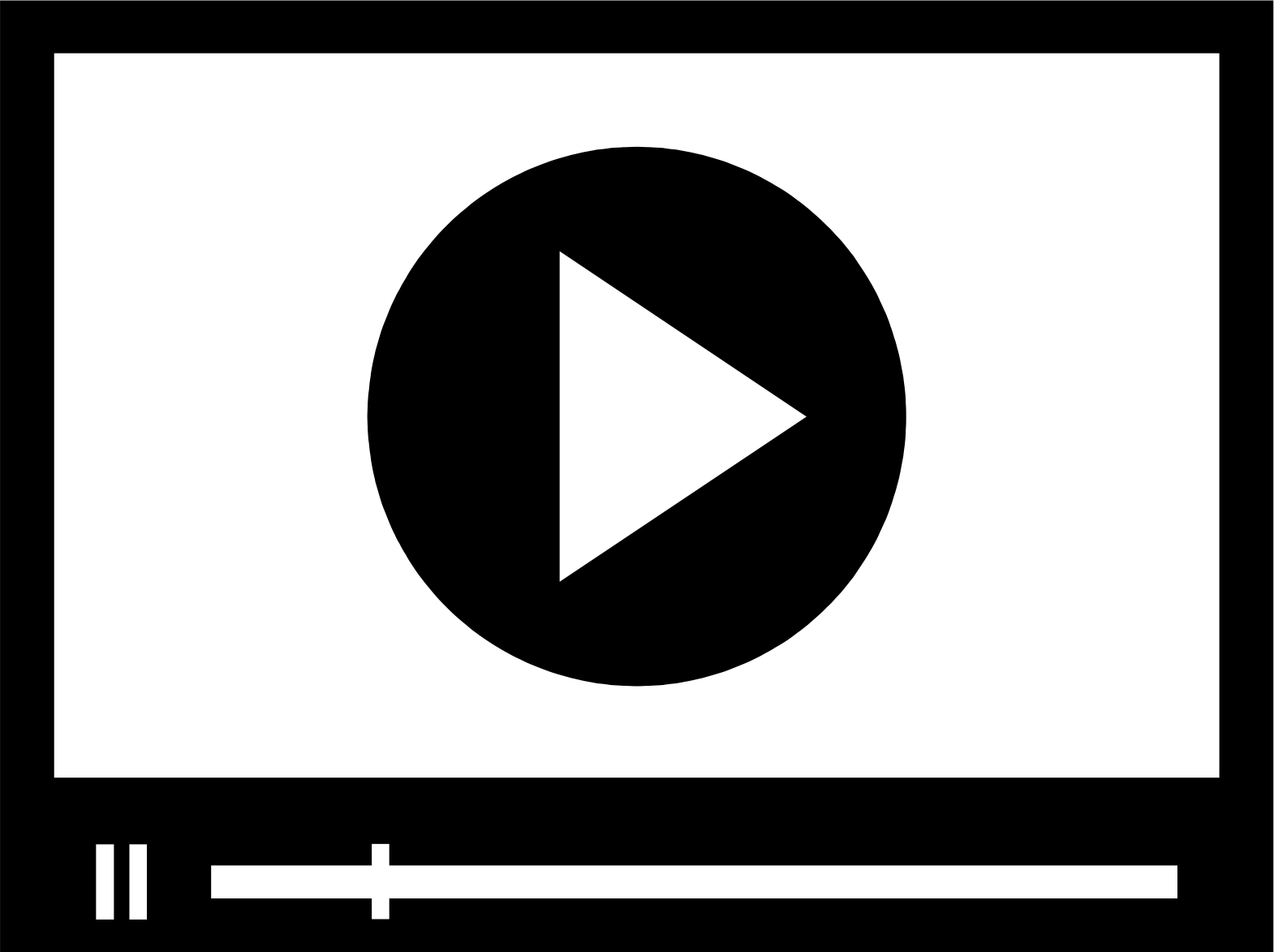
simples *peer to peercomo* avanc;adas coordenac;6es entre eventos de banco de dados. Clientes PostgreSQL podem monitorar eventos de banco de dados, como atualizac;6es, inserc;6es e exclus6es em tabelas, enquanto eles acontecem.

##### ATENCAO

0 c6digo fonte do PostgreSQL e disponfvel sob a licenc;a *open source* BSD. Essa licenc;a da a liberdade de usar, modificar e distribuir o PostgreSQL da forma que se desejar, com c6digo

aberto ou fechado. Quaisquer modifica96es, melhoramentos ou mudan9as que voce fa<;a serao seus para fazer o que quiser. Por isso, o PostgreSQL nao e somente um poderoso SGBD capaz de rodar o neg6cio da empresa. Tambem e uma plataforma de desenvolvimento sobre a qual se pode desenvolver produtos de software *in-house* que requeiram um suporte adequado a banco de dados.

Com a grande aceita9ao do PostgreSQL no mercado, desde 2004, um grupo de contribuidores do seu desenvolvimento fundou a empresa Enterprise DB, que fornece uma versao empresarial de mesmo nome, com consideraveis extens6es ao PostgreSQL.



# UM EXEMPLO DE SGBD RELACIONAL: POSTGRESQL.



### VERIFICANDO O APRENDIZADO

1. QUALE O PRINCIPAL MODULO COMPONENTE DE UM SGBD, PELO QUAL PASSAM TODAS AS CONSULT ASE TRANSACOES DE USUARIOS, SEJAM ADMINISTRADORES, USUARIOS CASUAIS OU PARAMETRICOS?
   1. Compilador de linguagem de manipulac;ao de dados.
   2. Catalogo de metadados.
   3. Otimizador de consulta.
   4. Processador de *run time.*
2. QUALE O MODELO DE COMPUTACAO UTILIZADO PELO POSTGRESQL, ASSIM COMO PELA MAIORIA DOS SGBDS RELACIONAIS EMPRESARIAIS?
   1. Centralizado.
   2. Pessoal.
   3. Cliente/Servidor.
   4. *Stand Alone.*

GABARITO

* + 1. **Quale o principal modulo componente de um SGBD, pelo qual passam todas as consultas e transac;oes de usuarios, sejam administradores, usuarios casuais ou parametricos?**

A alternativa **11D** II esta correta.

0 processador do banco de dados em tempo de execugao *(run time)* recebe todas as requisigoes dos usuarios e as processa para acessar o catalogo ea base de dados.

* + 1. **Qual e o modelo de computac;ao utilizado pelo PostgreSQL, assim como pela maioria dos SGBDs relacionais empresariais?**

A alternativa **11C** II esta correta.

0 PostgreSQL usa um modelo de computagao cliente/servidor, em que uma sessao consiste de um processo servidor *(back end),* que gerencia os arquivos do banco de dados, aceita conexoes par aplicagoes clientes e executa agoes sabre o banco de dados em name dos clientes, e aplicagoes clientes de usuarios *(front end)* que desejam executar operagoes no banco de dados.

# CONCLUSAO

**CONSIDERAC0ES FINAIS**

Este tema abordou uma introdugao aos sistemas de banco de dados, resumindo sua evolugao hist6rica desde a invengao do disco magnetico, que possibilitou o infcio da era de processamento de dados par computador.

Foram examinadas as principais caracterfsticas do sistema de banco de dados (SBD), com enfase naquelas que o diferenciam dos sistemas de arquivos, bem coma nas suas funcionalidades, vantagens e desvantagens.

Finalizamos o tema com a descric;ao da arquitetura de um sistema de gerencia de banco de dados (SGBD), componente central do SBD, terminando com um exemplo de SGBD relacional, amplamente utilizado no ensino e no mercado, o PostgreSQL.

Encerraremos este tema aprofundando um pouco mais sabre sistema de banco de dados.





### REFERENCIAS

BALIEIRO, R. **Banco de Dados.** 1. ed. Rio de Janeiro: SESES, 2015

BOOCH, G; RUMBAUGH, J; JACOBSON, I. **The Unified Modeling Language User Guide.**

Reading: Addison-Wesley. 1998.

CHEN, P. **The Entity-Relationship Model - Toward A Unified View of Data.** /n: ACM Transactions on Database Systems. 1 (1): 9-36, 1976.

CODD, E. F. **A Relational Model of Data for Large Shared Data Banks.** *In:* Communications of the ACM. 13 (6): 377-387, 1970.

DB-ENGINES. **DB-Engines Ranking of Relational DBMS.** Consultado em meio eletronico em:

3 jun. 2020.

ELMASRI, R.; NAVATHE, S. **Sistemas de Banco de Dados.** 7. ed. Sao Paulo: Pearson, 2019.

IBM. **SQL: The language of Db2.**

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 9075-1:2016:** Information

technology- Database languages - SQL - Part 1: Framework (SOL/Framework). Publicado em: dez. 2016.

KLUG, A.; TSICHRITZIS, D. The ANSVX3/SPARC DBMS framework: report of the study group on database management systems. *In:* **Information Systems. Data: creations, management and utilization.** v. 3, 3. Elsevier Journal, 1978.

MELO, R; SILVA, S.; TANAKA, A. **Banco de Dados em Aplica oes Cliente-Servidor.** Rio de Janeiro: lnfobook, 1998.

PPLWARE. **Charles Babbage-O pioneiro dos computadores.** Publicado em: 28 set. 2017.

**POSTGRESQL. PostgreSQL: o banco de dados relacional de c6digo aberto mais avan ado do mundo.** Consultado em meio eletronico em: 3 jun. 2020.

TANAKA, A. **Notas de Aula sobre Banco de Dados do professor Asterio Tanaka.** Disponfvel sob licen<;a Creative Commons BRAtribui9ao - CC BY, 2018.

WIKIPEDIA. **Comparison of relational database management systems.** Consultado em meio eletronico em: 6 jun. 2020.

YUGE, C. **Crise da COVID-19 aumenta a procura por programadores de COBOL.** *In:*

Canaltech. Publicado em: 13 abr. 2020.

### EXPLORE+

Para refor9ar o conteudo visto, todo born livro-texto de fundamentos de banco de dados possui uma introdu9ao ao assunto, nos moldes deste tema. Recomendamos o livro-texto mais adotado mundialmente: *Sistemas de Banco de Dados,* de Ramez Elmasri e Shamkant B. Navathe. Vale a leitura da parte 1, sobre introdu9ao a banco de dados, formada pelos capftulos: 1 - Bancos de Dados e Usuarios de Bancos de Dados; 2 - Conceitos e Arquitetura do Sistema de Banco de Dados

Sugerimos uma visita ao excelente *DB-Engines Ranking,* que possui uma vasta classifica9ao dos SGBDs mais populares, nos diversos modelos de dados, com breves explica96es sabre cada modelo e informa96es sobre cada produto.

Como dissemos, a documenta9ao do PostgreSQL, disponivel em website de mesmo nome, e reconhecidamente um material de referencia para o modelo relacional-objeto em geral, servindo como excelente ponto de partida para o entendimento do tema. Sugerimos iniciar pela se9ao *About* e prosseguir fazendo o *Tutorial,* que inclui *Getting Started, The SQL Language* e *Advanced Features.*

Leia a materia *Crise da COVID-19 aumenta a procura porprogramadores de COBOL,* de Claudio Yuge.

Como dissemos, o PostgreSQL e reconhecido como como o mais avan9ado SGBD *open source* do mundo. Acesse o website do PostgreSQL, onde voce encontrara a possibilidade de baixa-lo e saber sobre as ultimas atualiza96es.

Leia o artigo *The ANSIIX3/SPARC DBMS frameoork report of the study group on database management systems,* de Dennis Tsichritzis e Anthony Klug, e aprofunde seus conhecimentos sobre a arquitetura ANSI/SPARC.

Acesso o site da Wikipedia e busque par *Comparison of relational database management systems,* que traz uma excelente compara9ao de diversas caracteristicas de SGBDs relacionais, contendo informa96es detalhadas sobre mais de 60 produtos.

Acesse o website da ISO *(International Standard Organization)* e busque pela resolu9ao ISO/IEC 9075-1:2016, na qual a SQL e reconhecida como a linguagem padrao para o banco de dados relacional.

### CONTEUDISTA

Nathielly de Souza Campos

,§> **CURRiCULO LATTES**