



Sistema de banco de dados

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

Prof. Tomás de Aquino Tinoco Botelho

1. Itens iniciais

Objetivos

- Reconhecer o histórico dos bancos de dados e suas tecnologias.
- Identificar as características dos sistemas de banco de dados (SBD).
- Descrever a arquitetura dos sistemas de gerência de banco de dados (SGBD).

Introdução

Olá! Antes de começarmos, assista ao vídeo sobre sistema de banco de dados.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Histórico

Você certamente já leu sobre o termo **banco de dados** em algum contexto técnico ou geral, na mídia tradicional ou na internet. Mas o que é um banco de dados?

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

Confira um breve histórico dessa tecnologia, em que apresentaremos a evolução dos sistemas de informação em computador.



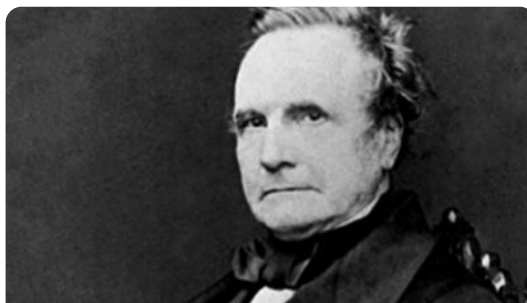
Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Arquitetura do computador

Historicamente, o computador, inventado na década de 1940 ao fim da Segunda Guerra Mundial, era usado primordialmente como uma máquina para cálculos matemáticos complexos, a exemplo da máquina diferencial de **Charles Babbage**, do século XIX.

Charles Babbage (1791-1871) é um ícone no mundo da computação. As suas notáveis contribuições para a área fizeram dele o pioneiro dos computadores, com a Máquina Diferencial N.1, a primeira calculadora automática, reconhecida pela precisão que já apresentava na época apresentava (PPLWARE, 2017).



Charles Babbage



John von Neumann

Logo se percebeu que, graças à arquitetura criada pelo seu inventor, John von Neumann, baseada em uma unidade central de processamento que armazena programas e dados, o computador também serve para o processamento de dados, e não apenas para cálculos.

John von Neumann (1903-1957) foi um matemático húngaro, naturalizado norte-americano, considerado o pai da Teoria dos Jogos. Seus interesses abrangiam lógica, assuntos militares, computação, economia, entre outros.

À época, esse dispositivo – atualmente conhecido como disco rígido e pela sigla HD (hard disk) – apresentou a capacidade de leitura de dados externos à unidade central de processamento de forma direta, sem a necessidade de leitura sequencial, como em fitas magnéticas.



HD (hard disk)

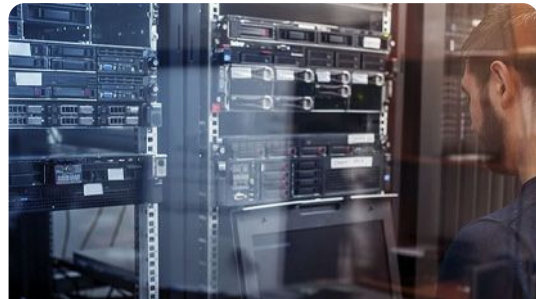
Essa utilidade do computador foi impulsionada com a invenção do disco magnético pela IBM, em 1957, que o denominou de dispositivo de armazenamento de acesso direto, do inglês *direct access storage device* (DASD).

Era do processamento de dados por computador

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

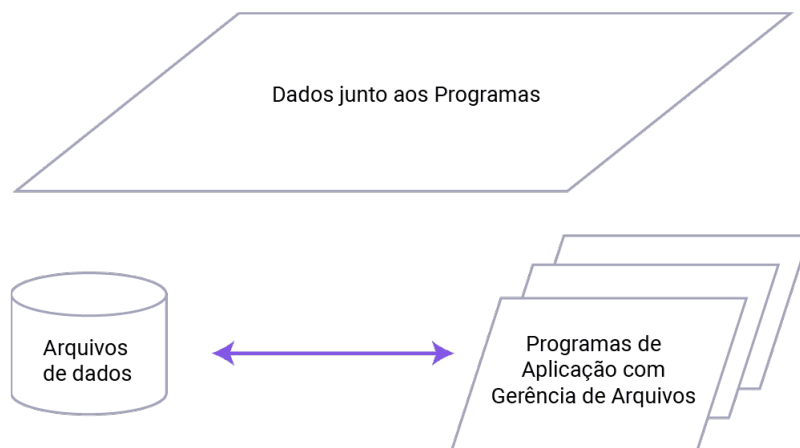
Você já ouviu falar em centro de processamento de dados (CPD), denominação que ainda persiste em organizações tradicionais?

Nessa era, os programas de aplicação, desenvolvidos em uma linguagem de programação (usualmente COBOL, em aplicações empresariais, ou Fortran, em aplicações científicas), manipulavam dados armazenados em arquivos hospedados em disco magnético, utilizados pelo sistema operacional e formando o que se denomina **sistema de arquivos**.



Servidor de arquivos

Vejamos agora a evolução obtida ao possibilitar que os programas acessassem os dados externamente em arquivos no disco (configurando um avanço em relação aos programas que continham internamente os próprios dados) para execução em lotes (batch), na fase inicial do uso do computador.



Sistemas de arquivos vs. banco de dados

Esse modelo de processamento de dados com sistema de arquivos foi largamente utilizado no início do emprego do computador em sistemas de informação empresariais, após o advento do disco magnético, persistindo até 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 manutenção em sistemas da Administração Pública do governo dos EUA.



Saiba mais

Para saber mais sobre o aumento da procura por programadores COBOL durante a pandemia de covid-19, não deixe de verificar a indicação feita no Explore +.

Conceito de banco de dados

O conceito de banco de dados como uma coleção de dados relacionados sempre existiu como componente central dos sistemas de informação. Estes, por sua vez, sempre existiram desde que a humanidade se organizou como sociedade. Segundo afirmam Melo, Silva e Tanaka (1998), o que tem mudado rapidamente ao longo da história é a tecnologia que permite a sua implementação e que se confunde com o próprio conceito de sistemas de informação.

Assim, antes da existência do computador, bancos de dados existiam sob a forma de registros físicos em papel, organizados em pastas dentro de armários, que formavam os arquivos dos sistemas de informação, operados manualmente pelos seus usuários. Será que ainda existem sistemas de informação desse tipo em pleno século XXI?

Atividade 1

Questão 1

A arquitetura de computadores é a forma como os diversos componentes de um computador são organizados, determinando aspectos relacionados à qualidade, ao desempenho e à aplicação para a qual o dispositivo vai ser orientado. Nesse contexto, a arquitetura de von Neumann é caracterizada pela(o):

- ☒ A possibilidade de uma máquina digital armazenar seus programas no mesmo espaço de memória que os dados, podendo assim manipular tais programas.
- ☐ B advento do armazenamento externo em disco rígido.
- ☐ C execução em lotes dos dados.
- ☐ D coleção de dados relacionados como componente central dos sistemas de informação.
- ☐ E advento dos sistemas de bancos de dados.

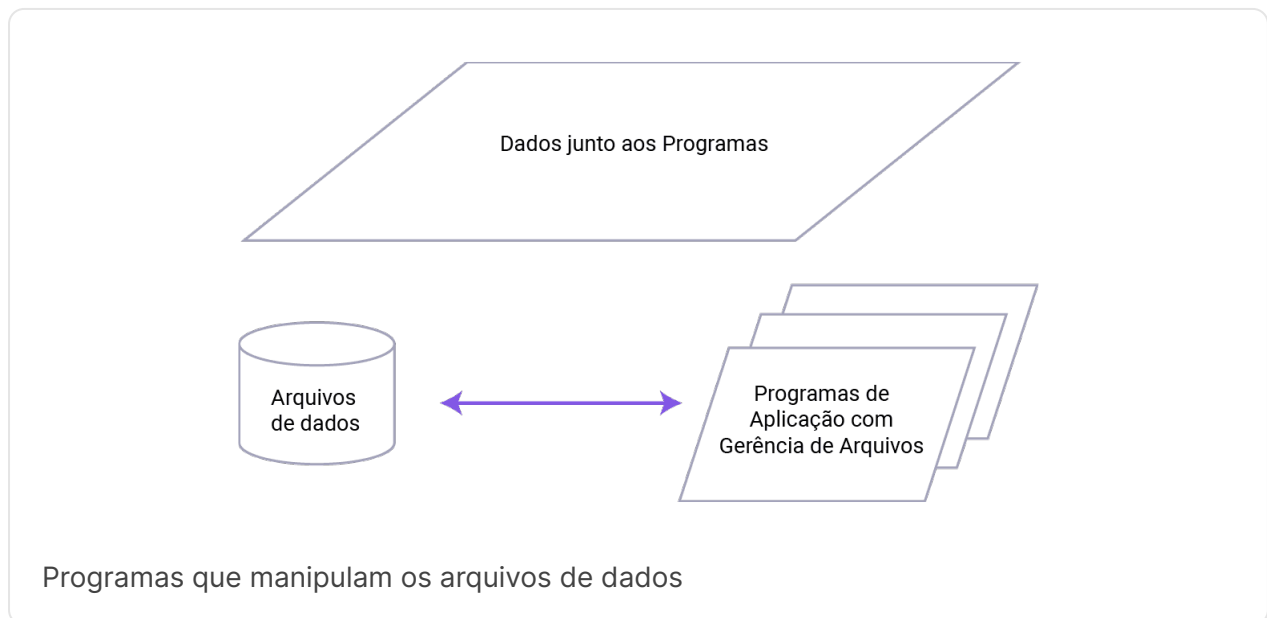


A alternativa A está correta.

Esta arquitetura é um projeto modelo de um computador digital de programa armazenado que utiliza uma unidade de processamento (CPU) e uma de armazenamento (memória) para comportar, respectivamente, instruções e dados.

Evolução dos sistemas de informação

Seguindo na história, o advento dos bancos de dados foi uma evolução natural dos sistemas de arquivos. Podemos observar que os **programas que manipulam os arquivos de dados**, além de implementarem a lógica da aplicação, têm de conter um módulo para a gerência dos arquivos de dados. Esse módulo deve ser repetido em todos os programas que precisam acessar e manipular o mesmo arquivo de dados.



Neste vídeo, entenderemos a evolução dos sistemas de informação em computador.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Primórdios dos sistemas de banco de dados

Os sistemas de banco de dados (SBD) vieram para mitigar esse problema, a partir de 1960, tirando dos programas de aplicação a responsabilidade de gerenciar os arquivos de dados. Essa tarefa passou a ser delegada a um software intermediário, denominado de sistema de gerência de banco de dados (SGBD), como mostrado a seguir.



Essa propriedade dos sistemas de banco de dados é denominada de independência entre dados e programas, uma diferença primordial em relação aos sistemas de arquivos.

Em outras palavras, ocorreu uma modularização do sistema de informação, com a distribuição de responsabilidades entre os programas de aplicação e o SGBD. Os programas de aplicação passaram a se

ocupar exclusivamente das funcionalidades da aplicação propriamente dita, deixando as tarefas de acesso e manipulação dos dados armazenados em disco para o SGBD, um software tipicamente auxiliar, de bastidores ou, como se costuma dizer no jargão do mercado, um serviço de back end.



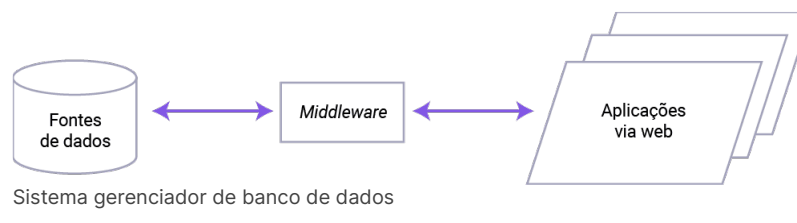
Comentário

Perceba a diferença entre o sistema de banco de dados (SBD) e o sistema de gerência de banco de dados (SGBD). O SBD é mais amplo, englobando o SGBD, os próprios programas de aplicação e os bancos de dados manipulados por eles.

Estágio atual dos sistemas de informação (na Web)

Antes de discorrer sobre o histórico dos SBD, vale completar a evolução dos sistemas de informação até os dias atuais, fortemente influenciada pela revolução tecnológica causada pela World Wide Web no final do século XX.

Com a popularização da interface web no desenvolvimento das aplicações, surgiram novas linguagens de programação e novas formas de armazenamento e acesso a dados em fontes com diferentes formatos. Assim, o SGBD das aplicações tradicionais pode ser considerado atualmente como um gênero de software básico, com papel intermediário, que denominamos como middleware, em que se incluem servidores de aplicações das diferentes linguagens e ambientes de desenvolvimento web. Confira!



Essa imagem resume o atual estágio dos sistemas de informação na web, em que as fontes de dados não se restringem a dados estruturados, como em bancos de dados tradicionais, admitindo volumes gigantescos em diversos formatos, localizações e velocidade de produção, características marcantes do conceito de big data. Igualmente, as aplicações via web são desenvolvidas em uma diversidade de plataformas digitais, de smartphones a supercomputadores, que têm em comum a conexão com a internet e, em consequência, a computação em nuvem (cloud computing).

Atividade 2

Considerando as descrições dos modelos de computação em sistemas de informação, qual modelo é o mais eficiente para uma aplicação na mesma linguagem de programação, com o mesmo volume de dados, sob a mesma infraestrutura de hardware?

A Modelo monolítico com os dados junto dos programas.

B Modelo de sistema de arquivos, com os programas acessando os dados por meio de módulos internos de gerência de arquivos.

C Modelo de sistema de banco de dados, com os programas acessando os dados por meio de um SGBD.

D

Modelo de sistema de informação na Web, com os programas acessando os dados na nuvem.

E

Modelo monolítico com os programas acessando os dados na nuvem.



A alternativa A está correta.

Consideradas as mesmas condições de aplicação, volume de dados e infraestrutura, conforme o enunciado, o modelo monolítico será mais eficiente, pois todo o processamento será realizado na velocidade de processamento junto à memória principal (RAM), que é muito mais rápida do que o acesso à memória secundária ou à nuvem.

Evolução dos sistemas de banco de dados

Podemos dizer que tudo começou com registros de bibliotecas, empresas em geral, registros policiais, fichas de pacientes em clínicas e hospitais e todas as informações armazenadas de maneira impressa para consultas posteriores.

Na década de 1960, os computadores se tornaram parte efetiva das empresas, juntamente com o crescimento da capacidade de armazenamento. Foram desenvolvidos dois principais modelos de dados: modelo em rede (*committee for data systems language* - CODASYL) e o modelo hierárquico (*information management system* - IMS).

Assista ao vídeo para compreender a evolução dos sistemas de banco de dados em computador.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Bancos de dados navegacionais

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

- **Integrated Data Systems (IDS):** Criado por Charles Bachman (1924-2017) no âmbito de um comitê que padronizou a linguagem COBOL (Committee On Data Systems Languages - CODASYL).
- **Information Management System (IMS):** Criado pela IBM na esteira do sucesso da invenção do disco magnético anos antes.

O IDS e o IMS tinham em comum a característica 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 característica, atualmente aqueles SGBDs e outros que seguiram a mesma abordagem são denominados de navegacionais.

Agora, observe a diferença entre eles:

IDS

Usava a estrutura de dados de grafos ou redes, daí a denominação de network databases.



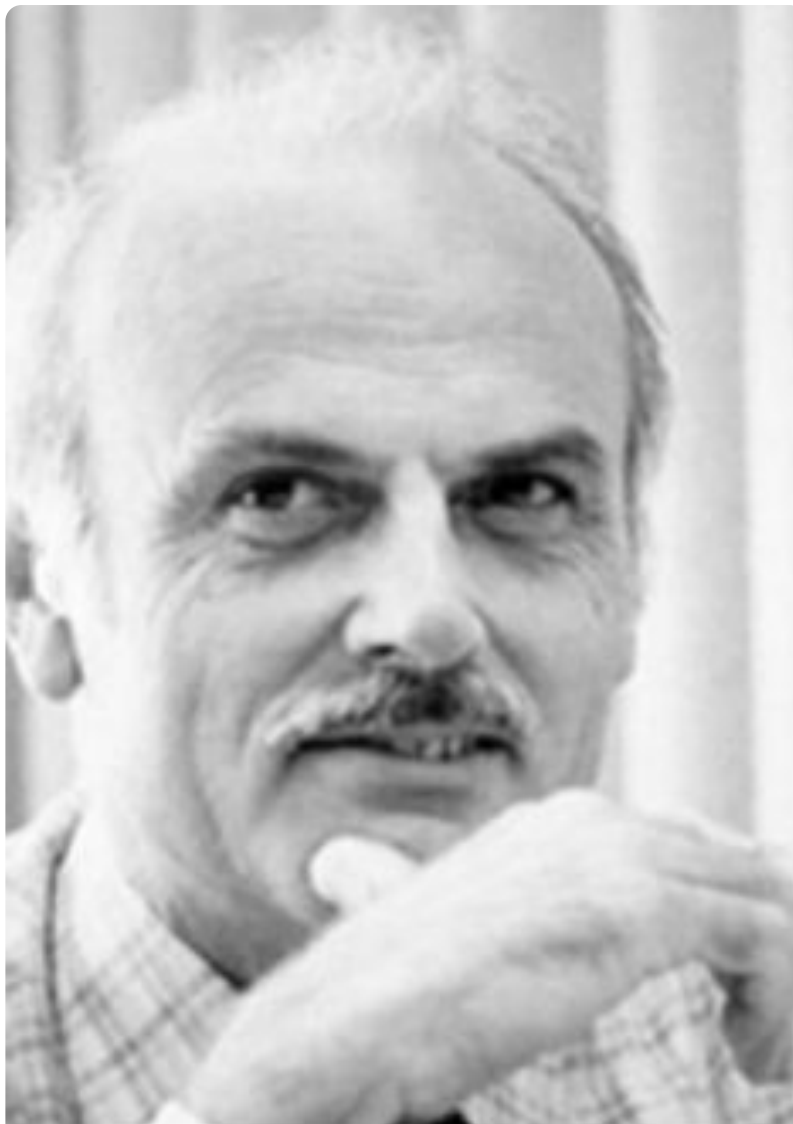
IMS

Adotava a estrutura de dados de árvores, que é um tipo de grafo mais restrito do que as redes, baseado em hierarquias, originando a denominação hierarchical databases.

Vários SGBD foram implementados com variantes desses modelos de banco de dados, como o Data Management System (DMS) e o Integrated Database Management System (IDMS). Vale lembrar que muitos sistemas de informação legados daquela época ainda utilizam esses SGBD navegacionais, a exemplo da demanda por programadores COBOL em meio à pandemia de covid-19.

Modelo relacional de banco de dados

A grande revolução na história dos bancos de dados ocorreu na virada das décadas de 1960 e 1970, com a publicação do artigo seminal do matemático 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.



Edgar Frank Codd (1923-2003)

Foi um cientista da computação e matemático americano que inventou o modelo de dados relacionais, que levou à criação do banco de dados relacional, um método padrão de recuperação e armazenamento de dados do computador. Fonte: Encyclopædia Britannica.



Curiosidade

O artigo de Codd, uma das obras mais citadas na comunidade da computação 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, é uma função matemática denominada relação.

Codd criou uma álgebra relacional e um cálculo relacional, nos quais baseou toda a teoria matemática das relações em que fundamentou o modelo relacional. Apesar da base teórica do modelo, a estrutura de dados subjacente tem o mérito de ser muito simples, pois uma relação nada mais é do que uma tabela formada por

colunas e linhas, em cujas células estão armazenados os dados, conceito compreensível pelo senso comum de qualquer leigo em matemática ou computação. Veja!

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

Exemplo de tabela para armazenagem de dados

A solidez da fundamentação matemática do modelo relacional disparou uma série de iniciativas de implementação em empresas, como a própria IBM, e no meio acadêmico, principalmente nas universidades do estado da Califórnia, onde se localizava o centro de pesquisas da IBM. A partir de então, a IBM patrocinou o projeto System R (de relational), enquanto a Universidade da Califórnia em Berkeley (UCB) deu início à implementação acadêmica de um SGBD relacional denominado de Interactive Graphics Retrieval System (Ingres).

Atividade 3

Assinale qual o nome do modelo de banco de dados cuja estrutura de dados, diferentemente dos grafos dos bancos de dados navegacionais, é fortemente baseada em uma função matemática denominada relação.

A Rede

B Hierárquico

C Relacional

D Árvore

E Orientado a objetos

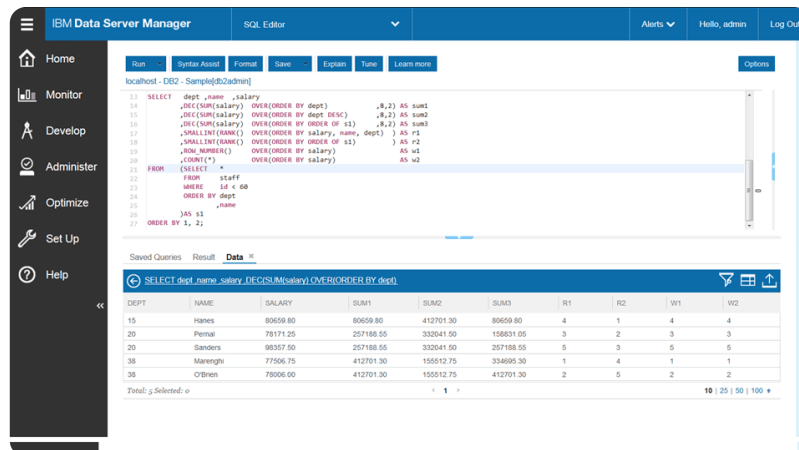


A alternativa C está correta.

O modelo relacional representa os dados em um banco de dados como uma coleção de relações e seus relacionamentos. Cada relação contém um nome e um conjunto de atributos com seus respectivos nomes. Informalmente, as relações do modelo relacional são também chamadas de tabelas pela maioria dos desenvolvedores.

Principais SGBD relacionais

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



Sistema gerenciador de banco de dados DB2

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

O vídeo a seguir apresenta os principais SGBD relacionais e suas características.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Oracle

No âmbito comercial, também despontou, como decorrência do sucesso do modelo relacional, o desenvolvimento de um SGBD pela empresa inicialmente denominada Software Development Laboratories (SDL), depois renomeada Relational Software Inc. (RSI) e, finalmente, Oracle Corporation, nome pelo qual é atualmente reconhecido o SGBD líder do mercado global de banco de dados.



Logotipo Oracle

MySQL

Em 2010, com a aquisição 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 liderança na comunidade de desenvolvimento de sistemas para a Web. O SGBD MySQL, associado ao sistema operacional Linux, ao servidor Web Apache e à linguagem de programação PHP, formou o quarteto de software conhecido pela sigla LAMP, de grande sucesso no desenvolvimento de aplicações web até os dias atuais.



Logotipo MySQL

SQL Server

Na frente acadêmica, o projeto Ingres, da Universidade da Califórnia (UCB), deu origem a versões comunitárias mediante licença livre da própria universidade junto com seu sistema operacional Unix, denominado Berkeley Software Distribution (BSD). O projeto acadêmico originou um produto comercial de mesmo nome, Ingres DBMS, que concorreu diretamente com o Oracle e o SQL/DS nos primórdios dos SGBD relacionais.



Logotipo Microsoft SQL Server

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

PostgreSQL

A continuidade do projeto Ingres deu frutos também na área acadêmica com a evolução para um modelo de dados além do relacional, estendido com conceitos da programação orientada a objetos, denominado Postgres (de post ingres).



Logotipo PostgreSQL



Saiba mais

Após a incorporação da linguagem SQL na década de 1990, o Postgres foi rebatizado como PostgreSQL, atualmente reconhecido como o mais avançado SGBD open source do mundo. Para saber mais sobre o PostgreSQL, não deixe de verificar a indicação feita no Explore +.

O PostgreSQL e o MySQL são os SGBD mais utilizados no aprendizado dos bancos de dados relacionais pela sua popularidade e pelo fato de disponibilizarem versões com licença e documentação livres.

Atividade 4

Qual é o nome da linguagem de consulta e manipulação de banco de dados que se tornou sinônimo do modelo relacional de banco de dados?

A PostgreSQL

B MySQL

C SQL

D SQL Server

E Oracle



A alternativa C está correta.

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

Outros modelos de SGBD

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

Cabe observar que esse ranking não se trata exclusivamente de SGBD do modelo relacional de dados. Os próprios SGBD relacionais, mencionados como líderes de mercado, são classificados no ranking como “multimodelos”, pois implementam funcionalidades que vão além do modelo relacional. Veja!

Banco de Dados	Relacional	Documentos	Grafos	RDF
Oracle	X	X	X	X
MySQL	X	X		
SQL Server	X	X	X	
PostgreSQL	X	X		
IBM DB2	X	X		X

Ranking de multimodelos
Tanaka, 2018, s.p.

O ranking é atualizado mensalmente e você pode saber mais sobre bancos de dados acessando o site db-engines.com.

Assista ao vídeo a seguir, que apresenta outros modelos de SGBD.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Banco de dados não relacionais

Não resta dúvida de que o modelo relacional se firmou no mundo corporativo, sendo utilizado na grande maioria dos sistemas de informação empresariais pela sua popularidade e pela robustez dos produtos disponíveis ao longo de décadas de desenvolvimento, bem como pela padronização e pelo uso da linguagem de consulta e manipulação de dados SQL.

Entretanto, existem aplicações em sistemas de informação que requerem muito mais recursos de armazenamento e manipulação de dados do que as tabelas do modelo relacional, em especial aplicações web e de cunho científico que processam grandes quantidades de dados em formatos diversos, com as atuais tendências como big data, internet of things e data science.



Representação ilustrativa de armazenamento e manipulação de dados

Assim, vários modelos de banco de dados não relacionais vêm surgindo no mercado, sendo denominados de NoSQL, termo traduzido como “não SQL” ou “não somente SQL” (de not only SQL).

São, de fato, bancos de dados que não adotam o modelo relacional de dados e, portanto, não usam a linguagem SQL, embora alguns possuam implementações do comando SELECT da SQL para fins de compatibilidade de linguagem de consulta com os bancos de dados relacionais.



Comentário

O estudo de bancos de dados NoSQL está fora do escopo deste estudo, constituindo-se em um tema à parte pela diversidade dos seus conceitos e de suas tecnologias. É interessante, porém, conhecer a importância dessa tendência dos SGBDs, como demonstra o site DB-Engines Ranking, que apresenta mais de uma dúzia de modelos de bancos de dados NoSQL com seus principais produtos, muitos deles multimodelos.

Multimodelos de banco de dados

Listamos a seguir os multimodelos, confira!

1

Chave-Valor

Redis, Amazon DynamoDB, Microsoft Azure CosmosDB.

2

Documentos

MongoDB, Amazon DynamoDB, Microsoft Azure CosmosDB.

3

Séries temporais

InfluxDB, KDB+, Prometheus.

4

Grafos

Neo4J, Microsoft Azure CosmosDB, ArangoDB.

5

Orientado a objetos

InterSystems Caché, Versant Object Database, ObjectStore.

6

Motores de busca

Elasticsearch, Splunk, Solr.

7

RDF (Resource Description Framework)

Marklogic, Apache Jena, Virtuoso.

8 Colunar

Cassandra, HBase, Microsoft Azure CosmosDB.

9

Multivalores

Adabas, UniData/UniVerse, jBASE.

10

XML nativo

Marklogic, Oracle Berkeley DB, Virtuoso.

11

Eventos

Event Store, IBM DB2 Event Store, NEventStore.

12

Conteúdos

JackRabbit, ModeShape.

13

Navegacional

IMS, IDMS.

O modelo navegacional é exatamente aquele dos primórdios dos sistemas de banco de dados, da década de 1960, antes do advento do modelo relacional, cujos produtos ainda continuam sendo utilizados, principalmente em sistemas de informação legados daquela época.

Atividade 5

Os SGBD relacionais, mencionados como líderes de mercado, são classificados no ranking como “multimodelos” porque implementam funcionalidades que vão além do modelo relacional. Qual dos SGBD a seguir apresenta, além das funcionalidades de relacional e documentos, as de grafos e RDF?

A

PostgreSQL

B

MySQL

C

Oracle

D

SQL Server

E

MongoDB



A alternativa C está correta.

O SGBD Oracle implementa as funcionalidades dos modelos relacional, documentos, de grafos e RDF.

Evolução dos sistemas de informação em computador

Verificamos que uma característica primordial que distingue os sistemas de banco de dados dos sistemas baseados de arquivos é a independência dos dados em relação a programas. Essa, na verdade, é apenas uma espécie da característica dos sistemas de banco de dados denominada independência de dados.

Neste vídeo, explicaremos a evolução dos sistemas de informação em computador.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Arquitetura de três esquemas

O conceito de independência de dados deriva da arquitetura de três esquemas do banco de dados que separa as aplicações dos usuários finais do banco de dados físico armazenado.

Vejamos uma adaptação da arquitetura ANSI/SPARC, que considera três níveis de esquemas:

Nível externo

Contém o esquema conceitual externo, integrando as diferentes visões dos usuários das aplicações.



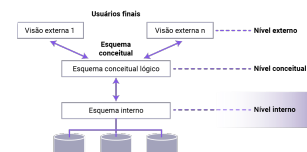
Nível conceitual

Contém o esquema conceitual lógico ou de implementação, descrevendo a estrutura lógica do banco de dados.



Nível interno

Contém o esquema interno ou físico do banco de dados armazenado em disco.



Essa arquitetura define dois tipos de independência de dados, além da independência entre dados e programas:

Independência lógica de dados

Consiste na capacidade de se alterar o esquema conceitual lógico, por exemplo, acrescentando um item de dado, sem alterar o esquema conceitual externo, isto é, as visões externas dos usuários por meio dos programas de aplicação.

Independência física de dados

Consiste na capacidade de se alterar o esquema interno, por exemplo, reorganizando os arquivos físicos que armazenam os dados, sem alterar o esquema conceitual lógico e, em consequência, o esquema conceitual externo.

Sistemas de banco de dados x Sistemas de arquivos

Vejamos agora, algumas outras características importantes, entre muitas, que diferenciam os sistemas de bancos de dados dos sistemas de arquivos.

1

Natureza autocontida

Significa que, além dos dados, o SBD contém a descrição completa de suas estruturas e restrições. Como se verá na arquitetura do SGBD, a descrição da estrutura e das restrições de dados, conhecida como metadados, isto é, dados que descrevem dados, é armazenada no catálogo do sistema de banco de dados.

2

Abstração de dados

Permite a representação conceitual dos dados por meio de modelos de dados que ocultam detalhes de armazenamento e implementação, os quais não interessam aos diferentes usuários, dando suporte a múltiplas visões lógicas dos dados e à independência de dados.

3

Suporte ao compartilhamento de dados e processamento de transações concorrentes

Permite que múltiplos usuários acessem o banco de dados simultaneamente.

Modelos de dados

Suportam a abstração de dados e permitem sua independência lógica e física são classificados como:

Modelos físicos

Descrevem como os dados são armazenados no computador mediante informações como tipos de arquivos, formatos e ordenação de registros, caminhos de acesso. São as várias formas de estruturas de arquivos que dependem do SGBD e do sistema operacional em que estão instaladas.

Modelos lógicos

Representam, de maneira abstrata, a implementação dos bancos de dados, ocultando detalhes de como os dados são armazenados e acessados no disco. O modelo lógico mais tradicional, largamente utilizado na grande maioria dos sistemas de informação organizacionais, é o criado por Edgar Codd. A estrutura de dados do modelo relacional é a tabela (relação matemática) e deve ser conferida em profundidade em estudo específico, assim como as implementações de SGBD relacional. Como vimos, há outros modelos lógicos de implementação do banco de dados entre aqueles que atualmente se denominam de modelos não relacionais ou NoSQL.

Modelos conceituais

Representam a visão dos dados do ponto de vista do usuário final, no nível de abstração mais próximo do mundo real. Dentre esses, destaca-se o modelo de entidades e relacionamentos, criado pelo pesquisador Peter Chen (1976). O modelo ER, segundo o autor, foi criado para prover 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.

Destacamos que a modelagem conceitual de dados realizada com o modelo ER foi incorporada na linguagem UML (*Unified Modeling Language*), criada no final da década de 1990 para uniformizar a modelagem orientada a objetos sob a forma do modelo de classes, em que as entidades são representadas pelas classes de objetos e os relacionamentos pelas associações entre as classes. Na prática atual do processo de desenvolvimento de sistemas de informação, é comum a utilização do modelo de classes da UML como substituto do modelo ER nas fases iniciais do projeto de banco de dados.

Atividade 1

Analise as seguintes afirmações e responda qual alternativa corresponde a características que distinguem os sistemas de banco de dados dos sistemas de arquivos.

I. Natureza autocontida dos dados.

II. Volatilidade dos dados.

III. Abstração de dados.

A Somente as afirmações I e II.

B Somente as afirmações I e III.

C Somente as afirmações II e III.

D Todas as afirmações estão corretas.

E Somente a afirmativa III.



A alternativa B está correta.

A natureza autocontida dos dados, com o armazenamento junto aos metadados, e a abstração de dados, permitindo a independência dos dados, são características que diferenciam os SBD dos sistemas de arquivos. Já a volatilidade dos dados não é uma característica dos SBD.

Vantagens e desvantagens da abordagem de banco de dados

A abordagem de SBD possui funcionalidades que conferem vantagens adicionais em relação a sistemas sem banco de dados, além das características que a diferenciam da abordagem de sistemas de arquivos.

Neste vídeo, apresentaremos as vantagens e desvantagens da abordagem de banco de dados.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Funcionalidades do SBD

Listamos a seguir algumas funcionalidades do SBD. Confira!

Controle da redundância de dados

Previne a possibilidade de inconsistência dos dados, a duplicação de esforço para manter os dados atualizados e o desperdício de espaço de armazenamento. O SGBD permite controlar o trade off entre o armazenamento em um único local no banco de dados *versus* a redundância forçada para melhorar o desempenho das consultas.

Compartilhamento de dados

Sendo os SBD multiusuários, realizam controle de concorrência de acesso aos dados compartilhados para garantir as propriedades de atomicidade, consistência, isolamento e durabilidade (propriedades ACID) das transações de banco de dados.

Controle de acesso a dados

Compreende mecanismos de segurança e autorização, como senhas para usuários e para grupos de usuários; restrição de acesso a partes do banco de dados; proibição de executar certas operações privilegiadas; acesso de usuário restrito apenas a transações autorizadas; proibição de uso de software privilegiado, como o software de administração do SBD.

Múltiplas interfaces de usuários e aplicações

Abrange diferentes linguagens de consulta para usuários casuais; linguagens de programação para programadores de aplicações; interfaces gráficas com formulários (telas) e menus para usuários de aplicações; interfaces para administração do banco de dados; interfaces de linguagem natural.

Representação de relacionamento entre os dados

Permite representar os relacionamentos existentes entre dados no mundo real, mediante mecanismos que dependem do modelo lógico de implementação do SBD.

Cumprimento de restrições de integridade dos dados

Previne a violação de restrições como tipo de dado ou domínio de valores admissíveis; unicidade de itens de dados por meio de chaves únicas; integridade referencial entre dados relacionados e restrições derivadas da semântica dos dados. Aqui também o SGBD permite controlar o trade off entre garantir o cumprimento automático das restrições ou deixar a sua especificação para os programas de aplicação.

Capacidade de backup e recuperação de dados

Em caso de ocorrência de falhas de hardware ou de software, os mecanismos de cópia de segurança (backup) e posterior restauração (recovery) garantem a consistência de estado do banco de dados antes e depois da falha.

Compartilhamento de dados entre múltiplos usuários simultâneos

Característica primordial do SBD que tem como resultado o controle de concorrência das transações.

Propriedades das transações – ACID

É responsabilidade do SGBD garantir as propriedades das transações, conhecidas pela sigla ACID. Vejamos mais a seguir!

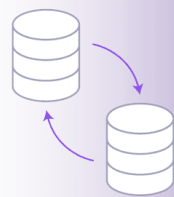
Atomicidade (atomicity)

Cada transação é tratada como uma unidade composta de uma sequência de operações, de modo que deve executar completamente com sucesso ou falhar completamente.



Consistência (consistency)

Uma transação só pode levar o banco de dados de um estado válido para outro, de acordo com suas regras de integridade.



Isolamento (isolation)

Cada transação é isolada das demais, isto é, essa propriedade assegura que transações executadas concorrentemente levem o banco de dados ao mesmo estado que chegaria se as transações fossem executadas sequencialmente.



Durabilidade (durability)

Uma vez que a transação é aceita (committed), o que significa que seu resultado foi gravado em memória não volátil, esse resultado permanecerá válido mesmo em caso de falhas do sistema.



Vantagens e desvantagens do SBD

Veremos agora as vantagens e desvantagens do SBD!

Vantagens

As vantagens decorrentes dessas funcionalidades do SBD são diversas: potencial para o estabelecimento de padrões de uso dos dados na organização; redução do tempo de desenvolvimento de aplicações; flexibilidade na manutenção dos dados; disponibilidade dos dados atualizados no âmbito de toda a organização; economia de escala, entre outras.



Desvantagens

A presença do SGBD como um software intermediário entre as aplicações e os dados armazenados provoca uma sobrecarga no desempenho do sistema como um todo. Além disso, há de se levar em conta o custo e o esforço adicional na capacitação e no oferecimento de funcionalidades sofisticadas como as já citadas anteriormente.

Assim, pode-se dizer que é preferível **não usar a abordagem de SBD** em algumas aplicações, tais como: aplicações muito simples com dados estáticos e bem definidos, aplicações de tempo real com requisitos rígidos os quais não possam ser atendidos com o overhead causado pelo SGBD, sistemas embarcados, com poucos dados e com requisitos estritos de tempo real (exemplo: piloto automático), sistemas monousuários de uso sem concorrência.

Poderiam ser enquadradas nessa lista as aplicações que requerem dados volumosos com dinâmica não processável por SGBDs tradicionais, como a Internet das Coisas (IoT, de Internet of Things), porém o advento de bancos de dados NoSQL, com modelos de dados específicos para tratamento de big data, busca atender a esses tipos de aplicações incompatíveis com os modelos tradicionais de SGBD.

Atividade 2

Qual das alternativas a seguir não é uma funcionalidade dos sistemas de banco de dados?

A Representação de relacionamentos entre os dados.

B Controle de redundância de dados.

C Compartilhamento de dados.

D Armazenamento de dados estáticos.

E

Múltiplas interfaces de usuários e aplicações.



A alternativa D está correta.

O armazenamento de dados estáticos é uma das situações em que não é recomendável o uso da abordagem de banco de dados, pois não se trata de uma funcionalidade típica dos SBDs, como as demais alternativas.

Papéis em sistemas de banco de dados

Um banco de dados, para funcionar, deve possuir um conjunto de quatro componentes básicos, são eles: dados, hardware, software e usuários. Vejamos agora o detalhamento desses componentes!

Neste vídeo, apresentaremos um detalhamento dos papéis em sistemas de banco de dados.

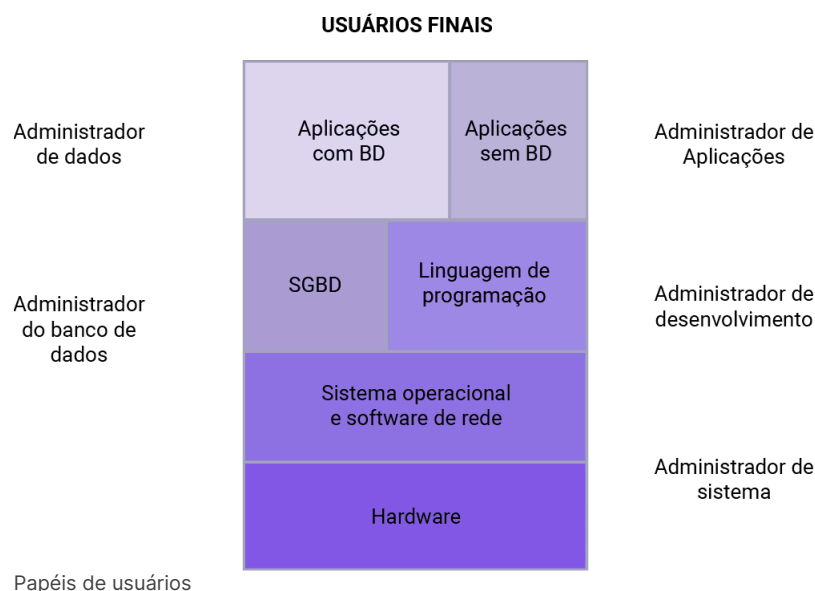


Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Camadas de um sistema de computação

A seguir, observe as camadas de um sistema de computação desde o hardware, nos quais são armazenados os arquivos com programas e dados, até as aplicações disponíveis aos usuários finais, que podem ser desenvolvidas com ou sem a utilização de um SGBD. Também é possível observar os diferentes papéis de usuários ao longo das camadas de software.



Papéis desempenhados

Em um sistema de computação corporativo de grandes organizações, é possível separar os diferentes papéis desempenhados. Vejamos!

1 Administrador de aplicações

Função técnico-gerencial responsável pela manutenção dos sistemas de aplicação e pelo suporte aos seus usuários, podendo ser exercida por vários administradores. Exemplo: administrador do sistema integrado de gestão empresarial, conhecido pela sigla ERP (*enterprise resource planning*).

2

Administrador de desenvolvimento

Função técnica que pode ser desdobrada em equipes de desenvolvimento do sistema de aplicação, dependendo do porte e da complexidade do sistema, que se utilizam das ferramentas disponíveis no ambiente de desenvolvimento de sistemas.

3

Administrador de dados

Função gerencial responsável pelo ambiente de dados da organização, que define políticas e responsabilidades sobre os recursos de dados, assim como as regras do negócio e os padrões de dados a serem seguidos no desenvolvimento.

4

Administrador do banco de dados

Função técnica responsável pela criação e manutenção dos bancos de dados no SGBD, dando suporte às equipes de desenvolvimento no tocante aos objetos dos bancos de dados.

5

Administrador de sistema

Responsável por manter no ar o sistema de computação 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 papéis de administração podem ser exercidos por uma única pessoa, mas uma pessoa pode exercer mais de um papel. A distribuição desses papéis por indivíduos e equipes dependerá do porte da organização e do sistema de computação, variando desde uma única pessoa exercendo todos os papéis de administração até todo um departamento de TI encarregado da administração e do suporte aos diversos sistemas em operação.

Atividade 3

Analise as seguintes afirmações e responda qual alternativa corresponde aos papéis beneficiários e ao responsável por definir políticas e responsabilidades sobre os recursos de dados:

A

Usuários finais e administrador de dados.

B

Administrador de aplicações e administrador do banco de dados.

C

Usuários finais e administrador do banco de dados.

D

Usuários finais e administrador de aplicações.

E

Administrador de dados e administrador de aplicações.



A alternativa A está correta.

Os usuários finais são os beneficiários dos sistemas que rodam no sistema de computação e o administrador de dados é o responsável pelo ambiente de dados da organização, que define políticas e responsabilidades sobre os recursos de dados, assim como as regras do negócio e os padrões de dados a serem seguidos no desenvolvimento.

Componentes de um sistema de banco de dados

Sistema de banco de dados é a interação entre dados, humanos, hardware e software, o que permite dar “sentido” e tornar os dados acessíveis, compreensíveis e utilizáveis.

Neste vídeo, apresentaremos os componentes de um sistema de banco de dados.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Ambiente de banco de dados

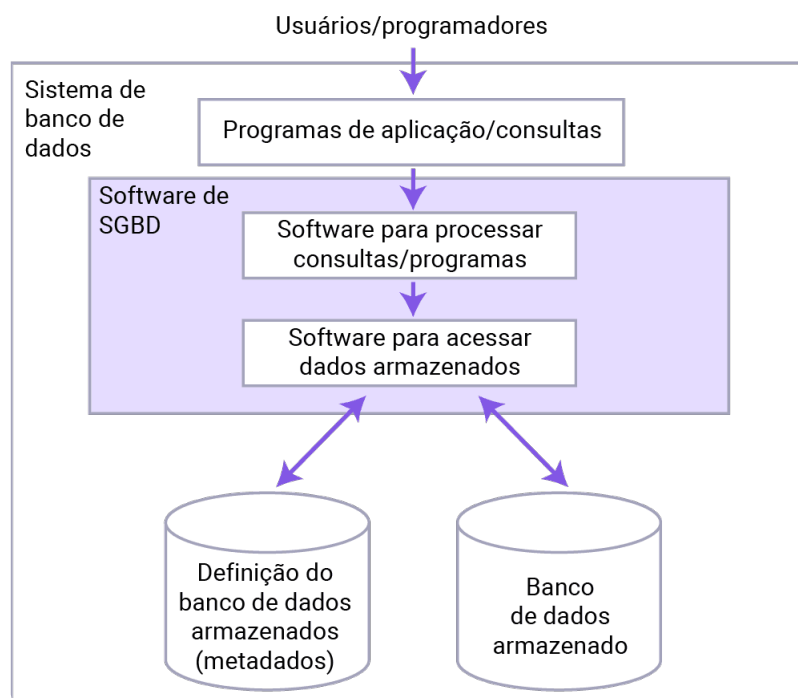
Vamos mostrar uma visão simples do ambiente de banco de dados, compreendendo programas de aplicação ou consultas de usuários que acessam os dados e metadados armazenados em disco por meio do SGBD. Este, por sua vez, compõe-se, simplificada, de um software para processar consultas e programas e outro para acessar os dados e metadados armazenados.



Representação ilustrativa de armazenamento de dados

Nota-se que não se confundem o SBD e o SGBD, visto que o SGBD é um componente daquele que engloba também os programas e as consultas, bem como os dados e metadados armazenados. Confira!

Nota-se que não se confundem o SBD e o SGBD, visto que o SGBD é um componente daquele que engloba também os programas e as consultas, bem como os dados e metadados armazenados. Confira!



A fronteira do SBD engloba os programas que implementam as aplicações, bem como as consultas provenientes de usuários com acesso a linguagens e interfaces de consulta. O SGBD é o software intermediário. A fronteira do SBD envolve os metadados armazenados no catálogo (às vezes chamado de dicionário de dados) e o próprio conteúdo armazenado no banco de dados.



Comentário

Costuma-se definir metadados como o esquema do banco de dados, estruturado de acordo com o modelo lógico de implementação. Assim, modelar um banco de dados consoante ao modelo lógico de implementação equivale a esquematizar o banco de dados conforme os construtores desse modelo. Por exemplo, no modelo relacional, o esquema é composto por tabelas e suas colunas. Cada comando de definição de dados, criando, alterando ou removendo uma tabela, provoca uma mudança no esquema do banco de dados.

Chama-se de estado ou instância o conteúdo do banco de dados armazenado em um momento. Cada manipulação no banco de dados mediante comandos de inserção, atualização ou remoção de dados provoca uma mudança de estado, gerando uma nova instância do banco de dados.

Curiosidades sobre BD

Como vimos, o termo **banco de dados** é correntemente usado com o sentido do original, em inglês, *database*, cuja origem, segundo o *Oxford English Dictionary*, remonta a 1962 em um relatório de uma empresa na Califórnia. Raras são as referências ao termo *data bank* como, por exemplo, no mencionado artigo de Edgar Codd sobre o modelo relacional (CODD, 1970).



Representação ilustrativa de acesso ao banco de dados

Em espanhol, usa-se *base de datos*, no francês *base de données*, enquanto em alemão se diz *datenbank*, e em italiano *banca dati*.

Em português, existem os dois termos, sendo banco de dados usado no sentido geral do ambiente que engloba o sistema, enquanto base de dados tem o sentido mais restrito do conteúdo do banco, isto é, corresponde ao estado ou à instância do banco de dados. Por exemplo, costuma-se referir à base de dados da Receita Federal como o conjunto de dados armazenados sobre os contribuintes e não como o sistema que gerencia os dados.

Atividade 1

Qual componente de um SBD provê uma descrição concisa a respeito dos dados, ou seja, são informações que acrescem aos dados e que têm como objetivo informar-nos sobre eles para tornar mais fácil a sua organização?

A

SGBD

B Metadados

C Banco de dados

D Programas de aplicação

E Usuários de banco de dados



A alternativa B está correta.

Metadados são dados que descrevem dados. Os metadados provêm uma descrição concisa a respeito dos dados. Os dados podem ser documentos, coleção de documentos, gráficos, tabelas, imagens, vídeos, entre tantos outros.

Sistema de gerência de banco de dados

Um sistema de gerenciamento de banco de dados (SGBD) — do inglês *data base management system* (DBMS) — é responsável pelo gerenciamento de um ou mais bancos de dados. Seu principal objetivo é retirar da aplicação cliente a responsabilidade de gerenciar o acesso, a persistência, a manipulação e a organização dos dados.

Neste vídeo, identificaremos um sistema de gerência de banco de dados.

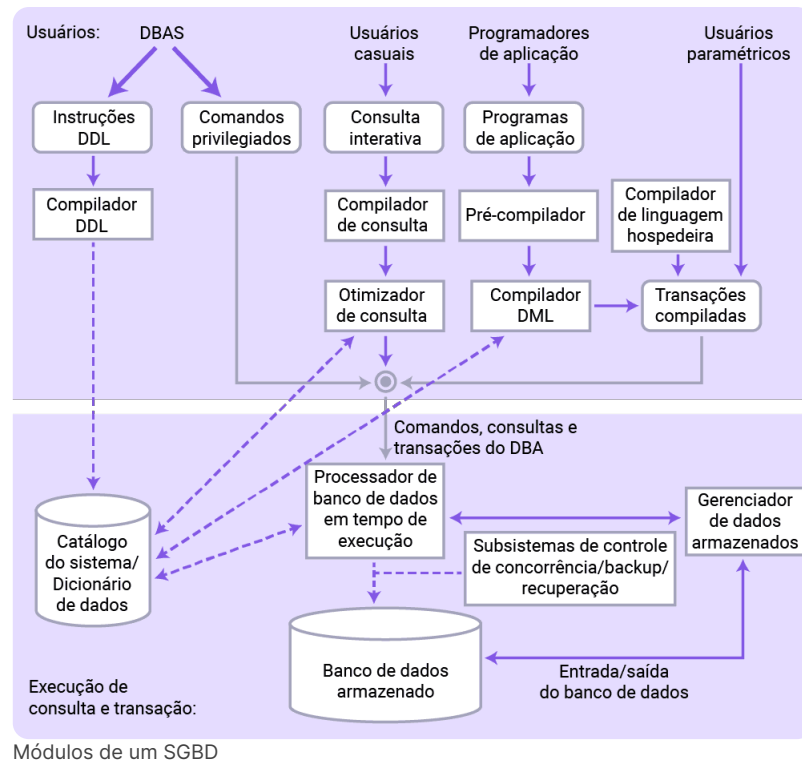


Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Módulos de um SGBD

A seguir, ilustramos, em detalhes, os diversos módulos que compõem um SGBD.



Módulos de um SGBD

Vamos conferir a descrição desses módulos!

Usuários

Da esquerda para a direita, vemos os usuários, começando com os administradores de banco de dados (ABD) do inglês *database administrator* (DBA).

ABD

O ABD usa comandos da linguagem de definição de dados (LDD), do inglês *data definition language* (DDL), para criar, alterar ou remover objetos do banco de dados (comandos CREATE, ALTER, DROP no padrão SQL), os quais ficam armazenados no catálogo do sistema, que contém os metadados.

O ABD também possui privilégio para executar comandos de controle de dados, conhecidos por alguns autores como linguagem de controle de dados (LCD), do inglês *data control language* (DCL), para conceder e revogar permissões de acesso aos dados (comandos GRANT e REVOKE no padrão SQL), entre outros. Note que o destino desses comandos privilegiados é um nó central do SGBD, por onde passam todos os comandos destinados ao processador de banco de dados em tempo de execução, denominado de processador de runtime ou runtime engine.

Usuários casuais

Seguindo para a direita, vemos os usuários casuais, os quais fazem consultas interativas por meio de uma interface para consultas *ad hoc*, ou seja, consultas não programadas previamente. As consultas são feitas tipicamente com o comando SELECT no padrão SQL do modelo relacional e provido de modo similar em outros modelos de implementação de banco de dados. Esses comandos são compilados por um compilador da linguagem de consulta e passam por um otimizador de consulta antes de chegar ao nó central do SGBD.

Programadores de aplicações

Mais à direita, aparecem os programadores de aplicações que escrevem os programas em uma linguagem de programação hospedeira, como, por exemplo, Java, PHP ou Python, nos quais estão embutidos comandos de consulta (SELECT), inserção, atualização e exclusão de dados em uma linguagem de manipulação de dados (LMD), do inglês *data manipulation language* (DML).

No padrão SQL, esses comandos são, respectivamente, INSERT, UPDATE e DELETE. Note que eles manipulam dados, diferentemente dos comandos da linguagem de definição de dados (CREATE, ALTER, DROP), que manipulam metadados.

Programas de aplicação

Os programas de aplicação, portanto, possuem comandos híbridos, da linguagem hospedeira e da linguagem de consulta e manipulação de dados.

Pré-compilador

Os programas de aplicação são processados, inicialmente, por um pré-compilador, responsável por separar os comandos e os repassar para os compiladores das respectivas linguagens.

Compiladores

Cabe a esses compiladores produzir o código das aplicações sob a forma de transações executáveis, que ficam à disposição dos usuários paramétricos.

Usuários paramétricos

Os usuários paramétricos são assim chamados porque interagem com o sistema por meio de parâmetros passados em interfaces apropriadas. Por exemplo, um agente de viagens faz uma reserva de passagem aérea passando ao sistema os dados do passageiro, data e hora da viagem, número do voo, número do assento e outros parâmetros necessários para efetivar a reserva.

Transações compiladas

Uma vez executadas as **transações compiladas**, assim como os demais comandos provenientes de usuários ou aplicações, são passadas ao nó central do SGBD para posterior processamento do processador de runtime.

Atenção! Antes de prosseguir na parte de baixo da imagem, note que alguns processos da parte de cima, como o otimizador de consulta e o compilador da LMD, estão ligados ao catálogo por linhas tracejadas, que denotam fluxos de controle, enquanto as linhas cheias representam fluxos de dados, além de controle. Isso é necessário porque as referências a objetos do banco de dados existentes no catálogo devem ser consistentes com os objetos, criados e mantidos pelo ABD mediante comandos da LDD.

Processamento do acesso a dados armazenados

Prosseguindo para a parte do processamento do acesso aos dados armazenados, observamos que o processador de runtime é o coração do SGBD. De fato, o diferencial dos produtos de SGBD reside na eficiência e na funcionalidade desse processador, segredo industrial em muitos dos SGBD proprietários, a ponto de exigir rigorosos termos de confidencialidade das pessoas que têm acesso ao seu código.

Acesso à base de dados

O processador de runtime também precisa acessar e, por vezes, modificar o catálogo, dependendo da natureza dos comandos, das consultas ou transações que estiver processando. Ele acessa a base de dados diretamente, sob controle de subsistemas de controle de concorrência, backup e recovery. Quando precisa realizar operações de entrada e saída (gravação e leitura) na base de dados, o processador de runtime se vale de um gerenciador de dados armazenados.

Atividade 2

Qual é o principal módulo componente de um SGBD pelo qual passam todas as consultas e transações de usuários, sejam administradores, usuários casuais ou paramétricos?

A Compilador de linguagem de manipulação de dados.

B Catálogo de metadados.

C Otimizador de consulta.

D Processador de run time.

E Transações compiladas.



A alternativa D está correta.

O processador do banco de dados em tempo de execução (runtime) recebe todas as requisições dos usuários e as processa para acessar o catálogo e a base de dados.

Exemplo de SGBD Relacional: PostgreSQL

O PostgreSQL originou-se do projeto Ingres da Universidade da Califórnia, tendo sido sucessivamente renomeado de Postgres, depois Postgres 95 e, finalmente, PostgreSQL.

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 concorrência direta com a SQL da IBM, até que o ANSI (American National Standard Institute) e depois a ISO (International Standard Organization) reconheceram a SQL como a linguagem padrão para o banco de dados relacional. A partir de então, a SQL foi implementada no Postgres, passando a ser chamada de PostgreSQL.

Neste vídeo, identificaremos as principais características do SGBD Relacional: PostgreSQL.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Características básicas

O PostgreSQL, além de ser reconhecido como *the world's most advanced open source relational database*, possui uma completa documentação igualmente reconhecida como referência global para os conceitos de SGBD relacional. Não é sem razão que o PostgreSQL, por ser open source e ter licença livre, é vastamente utilizado no ensino de banco de dados, além de ser adotado comercialmente em grandes organizações ao redor do mundo.



Logotipo do PostgreSQL

O PostgreSQL é um típico SGBD relacional-objeto, isto é, um SGBD cuja estrutura de dados básica é a relação (tabela), porém contempla extensões de tipos de dados e características próprias da orientação a objetos. Assim, possui suporte em grande parte ao padrão SQL e implementa extensões úteis como: consultas complexas, gatilhos (triggers), visões materializadas atualizáveis, controle de concorrência multiversionado. Além disso, pode ser estendido de muitas formas, por exemplo, gerando:

- novos tipos de dados;
- funções genéricas e agregadas;
- operadores;
- métodos de indexação;
- linguagens procedurais.

Em jargão de sistemas, o PostgreSQL usa um modelo de computação cliente/servidor, assim como a maioria dos SGBD relacionais empresariais. Uma sessão PostgreSQL consiste dos seguintes processos cooperativos (programas):

1

Back end

Um processo servidor (back end), responsável por gerenciar os arquivos do banco de dados, aceitar conexões por aplicações clientes e executar ações sobre o SBD em nome dos clientes. O programa servidor é chamado postgres.

Front end

As aplicações clientes de usuários (front end) que desejam executar operações no banco de dados. Aplicações clientes podem ser diversas em natureza, como uma ferramenta orientada a texto, uma aplicação gráfica, um servidor web que acessa o banco de dados para exibir páginas web, ou uma ferramenta especializada de manutenção. Algumas aplicações clientes são fornecidas com a distribuição PostgreSQL (psql, pgadmin).

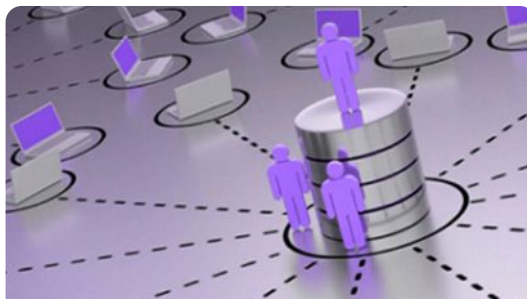
O PostgreSQL roda em todos os sistemas operacionais importantes, incluindo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) e Windows. Possui todas as propriedades ACID das transações (atomicidade, consistência, isolamento e durabilidade) e tem completo suporte para chaves estrangeiras, junções, visões, funções, triggers, e procedimentos armazenados em múltiplas linguagens de programação, incluindo Java, Perl, Python, Ruby, Tcl, C/C++, e sua própria PL/pgSQL, similar à PL/SQL do Oracle.

Características avançadas

Como banco de dados de classe empresarial, PostgreSQL possui características sofisticadas como controle de concorrência multiversão (*multi version concurrency control*), recuperação point in time recovery (PITR), tablespaces, replicação assíncrona, transações aninhadas (savepoints), on-line/hot backups, um sofisticado planejador/otimizador de consultas, e write ahead logging (WAL) para tolerância a falhas.

O PostgreSQL é altamente escalável tanto na quantidade de dados que pode gerenciar como no número de usuários concorrentes que pode acomodar. Existem sistemas PostgreSQL em ambientes de produção que gerenciam centenas de usuários simultâneos com dezenas de terabytes de dados armazenados.

Além de possuir um catálogo de sistema totalmente relacional, capaz de suportar múltiplos schemas por database, o catálogo do PostgreSQL é também acessível através do information schema, um conjunto de visões de metadados definido no padrão SQL.



Representação ilustrativa de gerenciamento de banco de dados



Comentário

PostgreSQL suporta índices compostos, parciais e funcionais que podem usar seus métodos de armazenamento B-tree, R-tree, hash ou GiST. Este último é um sofisticado framework que serve de base para muitos projetos públicos que usam PostgreSQL, como o OpenFTS (Open Source Full Text Search engine), que provê indexação online de dados e ranking de relevância para pesquisa em bancos de dados, e o PostGIS, um projeto que adiciona suporte para objetos geográficos, permitindo seu uso como banco de dados espacial para Sistemas Geográficos de Informação.

Veja, agora, outras características avançadas do PostgreSQL!

1 Herança de tabelas

Coloca um "sabor" orientado a objeto na criação de tabelas, permitindo a projetistas de banco de dados derivar novas tabelas de outras tabelas, tratando-as como classe base. Esse esquema suporta tanto herança simples como múltipla.

2

Sistema de regras

Também chamado de sistema de reescrita de consultas, permite ao projetista de banco de dados criar regras que identificam operações específicas para uma dada tabela ou visão e dinamicamente transformá-las em operações alternativas quando são processadas (cláusula `INSTEAD OF`).

3

Sistema de eventos

Sistema de comunicação interprocessos em que mensagens e eventos podem ser transmitidos entre clientes usando os comandos `LISTEN` e `NOTIFY`, permitindo tanto comunicação simples peer to peer como avançadas coordenações entre eventos de banco de dados. Clientes PostgreSQL podem monitorar eventos de banco de dados, como atualizações, inserções e exclusões em tabelas, enquanto eles acontecem.

O código fonte do PostgreSQL é disponível sob a licença open source BSD. Essa licença dá a liberdade de usar, modificar e distribuir o PostgreSQL da forma que se desejar, com código aberto ou fechado. Quaisquer modificações, melhoramentos ou mudanças que você faça serão seus para fazer o que quiser. Por isso, o PostgreSQL não é somente um poderoso SGBD capaz de rodar o negócio da empresa. Também é uma plataforma de desenvolvimento sobre a qual se pode desenvolver produtos de software in-house que requeiram um suporte adequado a banco de dados.



Curiosidade

Com a grande aceitação do PostgreSQL no mercado, desde 2004, um grupo de contribuidores do seu desenvolvimento fundou a empresa Enterprise DB, que fornece uma versão empresarial de mesmo nome, com consideráveis extensões ao PostgreSQL.

Atividade 3

Qual é o modelo de computação utilizado pelo PostgreSQL, assim como pela maioria dos SGBD relacionais empresariais?

A

Centralizado

B

Pessoal

☒ C Cliente/Servidor

☐ D Stand alone

☐ E Monolítico



A alternativa C está correta.

O PostgreSQL usa um modelo de computação cliente/servidor, em que uma sessão consiste de um processo servidor (back end), que gerencia os arquivos do banco de dados, aceita conexões por aplicações clientes e executa ações sobre o banco de dados em nome dos clientes, e aplicações clientes de usuários (front end) que desejam executar operações no banco de dados.

Considerações finais

O que você aprendeu neste conteúdo?

- Introdução aos sistemas de banco de dados, resumindo sua evolução histórica desde a invenção do disco magnético, que possibilitou o início da era de processamento de dados por computador.
- As principais características do sistema de banco de dados (SBD), com ênfase naquelas que o diferenciam dos sistemas de arquivos, bem como nas suas funcionalidades, vantagens e desvantagens.
- A descrição da arquitetura de um sistema de gerência 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.

Explore +

- Para reforçar o conteúdo visto, todo bom livro-texto de fundamentos de banco de dados possui uma introdução ao assunto. 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 introdução a banco de dados, formada pelos capítulos: 1 – Bancos de Dados e Usuários 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 classificação dos SGBDs mais populares, nos diversos modelos de dados, com breves explicações sobre cada modelo e informações sobre cada produto.
- Como dissemos, a documentação do PostgreSQL, disponível em website de mesmo nome, é reconhecidamente um material de referência para o modelo relacional-objeto em geral, servindo como excelente ponto de partida para o entendimento do tema. Sugerimos iniciar pela seção About e prosseguir fazendo o Tutorial, que inclui Getting Started, The SQL Language e Advanced Features.
- Leia a matéria **Crise da COVID-19 aumenta a procura por programadores de COBOL**, de Claudio Yuge.
- Como dissemos, o PostgreSQL é reconhecido como o mais avançado SGBD open source do mundo. Acesse o website do PostgreSQL, onde você encontrará a possibilidade de baixá-lo e saber sobre as últimas atualizações.

Referências

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. ACM Transactions on Database Systems., v. 1, n. 1, 1976.

CODD, E. F. **A Relational Model of Data for Large Shared Data Banks**. Communications of the ACM, v. 13, n. 6, 1970.

DB-ENGINES. **DB-Engines Ranking of Relational DBMS**. Consultado na internet em: 3 jun. 2020.

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

IBM. **SQL**: The language of Db2. IBM, 2 maio 2022. Consultado na internet em: 27 jun. 2023.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 9075-1:2016**: Information technology - Database languages - SQL - Part 1: Framework (SQL/Framework), 2016.

KLUG, A.; TSICHRITZIS, D. **The ANSI/X3/SPARC DBMS framework**: report of the study group on database management systems. Information Systems. Data: creations, management and utilization. Elsevier Journal, v.3, 1978.

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

PPLWARE. **Charles Babbage** – O pioneiro dos computadores. PLLware, 28 set. 2017.

POSTGRESQL. **PostgreSQL**: o banco de dados relacional de código aberto mais avançado do mundo. Consultado na internet em: 3 jun. 2020.

TANAKA, A. **Notas de Aula sobre Banco de Dados do professor Asterio Tanaka**. Disponível sob licença Creative Commons BR Atribuição – CC BY, 2018.

WIKIPEDIA. **Comparison of relational database management systems**. Consultado na internet em: 6 jun. 2020.

YUGE, C. **Crise da COVID-19 aumenta a procura por programadores de COBOL**. Canaltech, 13 abr. 2020.