

Introdução a Banco de Dados I

Caros alunos, as videoaulas desta disciplina encontram-se no AVA
(Ambiente Virtual de Aprendizagem).

Unidade 01

Conceitos básicos de banco de dados

Introdução da Unidade

Esta unidade está direcionada a apresentar ao aluno as diferenças entre um banco de dados e arquivos convencionais, abstração e dados, instâncias de esquemas, independência de dados, linguagem de acesso aos dados, gerência de banco de dados e por fim, apresentar as funções de um administrador de banco de dados – DBA.

Objetivos

- Apresentar os conceitos básicos de um banco de dados (SGBD e BD);
- Reportar as diferenças entre Banco de Dados e Arquivos Convencionais;
- Apresentar temas relacionais a abstração de dados, instâncias, esquemas e independência de dados;
- Abordar sobre Gestão de Banco de Dados e sobre a principal função de um Administrador de Banco de Dados – DBA.

Conteúdo programático

Aula 01 – Conceitos e definições relacionadas a SGBD e BD.

Aula 02 – Abordagem entre Banco de Dados e Arquivos Convencionais.



Você poderá, também, **assistir as videoaulas** em seu celular, basta apontar a câmera para os **QR Codes** distribuídos neste conteúdo.



APRESENTAÇÃO

Utilize o QRcode para assistir!



MINICURRÍCULO

Utilize o QRcode para assistir!



Referências

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 4. ed. São Paulo: Addison Wesley, 2005.

GOMES, E. H. **Sistema Gerenciador de Banco de Dados**: uma visão geral de SGBD, seus tipos, usuários e o Modelo Relacional. In: EHGOMES. [S. l.], 2 jan. 2021. Disponível em:

<http://ehgomes.com.br/disciplinas/bdd/sghd.php#>. Acesso em: 3 maio 2021.

HEUSER, C. A. **Projeto de Banco de Dados**. 3. ed. Porto Alegre: Ed. Sagra Luzzato, 2002.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S.; **Sistema de Banco de Dados**. Rio de Janeiro: Elsevier, 2006.

O que é um SGBD? **Oficina da Net**. [S. l.], 9 jun. 2016. Disponível em:

<https://www.oficinadanet.com.br/post/16631-o-que-e-um-sghd>. Acesso em: 3 maio 2021.

Aula 01 - Conceitos e definições relacionadas a SGBD e BD

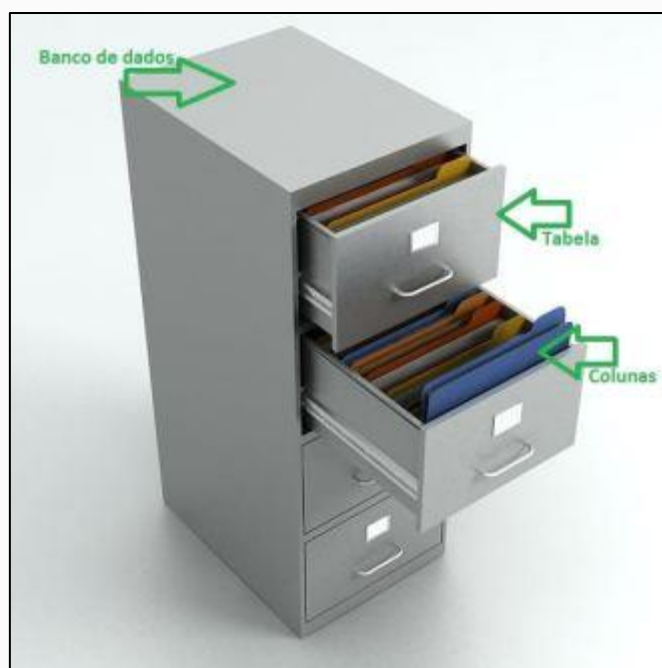
De modo teórico, existe uma definição específica para banco de dados, no qual basicamente consiste em uma coleção de dados inter-relacionados, representando assim informações sobre determinado domínio. Todo sistema, que necessita armazenar dados, deve possuir um banco de dados com informações consolidadas, organizadas e relacionadas. Para que, quando necessário, possa ser usado para extrair a informação para um relatório ou até mesmo para uma busca simplificada.

Exemplo: Determinado *software* é armazenado nos mais diversos dispositivos, quando o usuário executa alguma função, este dado, fica armazenado em uma memória temporária, na medida em que o programa é encerrado, todos os dados são limpos. Já um banco de dados, possui um local onde os dados ficam armazenados, e assim que executado, os dados apresentados são buscados, armazenados ou modificados diretamente do banco de dados.

- Todo sistema possui um banco de dados, composto por tabelas e colunas, onde são armazenados e organizados entre linhas, que podem ou não se relacionar entre si.

Realizando um paralelo com o mundo real, o banco de dados seria um arquivo de escritório, cada gaveta seria uma tabela, cada pasta uma coluna e dentro das pastas os dados referentes a cada pessoa. Este arquivo (banco de dados) pode ser acessado por qualquer pessoa, incluindo novos dados, realizando uma alteração ou até mesmo removendo algum dado. A Figura 1 demonstra esta breve analogia.

Figura 1 - Analogia sobre Banco de Dados



Fonte: Oficina da Net (2016)

Definições – BD e SGBD

BD - Banco de Dados (SILBERSCHATZ, 2006):

- Coleção de dados que contêm informações relevantes à uma empresa.

BD - Banco de Dados (HEUSER, 2002):

- Conjunto de dados inter-relacionados que objetivam atender às necessidades de um conjunto de usuários;
- Sinônimo: Base de Dados (*Database*).

BD - Banco de Dados (NAVATHE, 2005):

- Os dados são fatos que podem ser gravados e que possuem um significado implícito;
- Um BD representa alguns aspectos do mundo real (minimundo);
- Coleção lógica e coerente de dados com algum significado inerente;
- Um BD deve ser projetado, construído e “populado” por dados, atendendo a uma proposta específica.

SGBD – Sistema Gerenciador de Banco de Dados (SILBERSCHATZ, 2006):

- Coleção de dados inter-relacionados e um conjunto de programas para acessar esses dados de forma conveniente e eficiente.

SGBD – Sistema Gerenciador de Banco de Dados (HEUSER, 2002):

- Sistema Gerenciador de Banco de Dados (DBMS);

- *Software* que auxilia na definição, carga, atualização e manutenção de um banco de dados (BD).

SGBD – Sistema Gerenciador de Banco de Dados (NAVATHE, 2005):

- SGBD é uma coleção de programas que permite aos usuários criar e manter um banco de dados;
- Outras funções importantes de um SGBD:
 - Proteção contra falhas;
 - Segurança contra acessos não autorizados;
 - Manutenção dos dados por um longo período de tempo.
- Ainda segundo Silberschatz (2006), um sistema de banco de dados consiste de duas partes:
 - O sistema de gerenciamento de banco de dados (SGBD), programa que organiza e mantém essas listas de informações (BD) e;
 - O aplicativo de banco de dados, que é o programa que permite restaurar, visualizar e atualizar as informações armazenadas pelo SGBD.



VIDEOAULA 01

Utilize o QRcode para assistir!

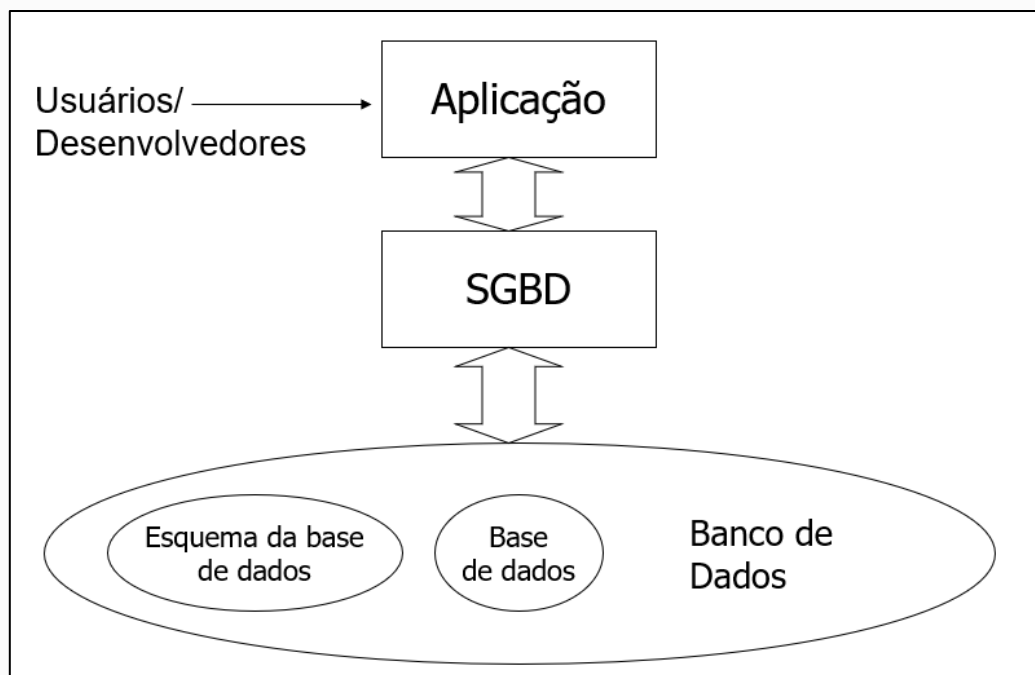
Agora, assista ao vídeo que aborda as definições de SGBD e BD.



Arquitetura de um Sistema de Banco de Dados

A Figura 2, demonstra uma arquitetura simplificada de um banco de dados. Onde, os usuários finais (Desenvolvedores) acessam o SGBD, que é o *software* responsável pelo gerenciamento de um ou mais banco de dados. Com isso, é possível realizar o acesso a base de dados.

Figura 2 - Arquitetura de um sistema de banco de dados



Fonte: o autor (2021)

Incidentes no desenvolvimento de aplicações de BD

- Arquivos são projetados para atender diferentes necessidades, enquanto bancos de dados não:
 - Ex.: Armazenamento de estruturas de dados complexas.
- Um banco de dados é acessado por múltiplos programas:
 - Várias equipes de desenvolvimento podem estar envolvidas;
 - Definição da estrutura da base de dados:
 - Deve ser mantida de forma centralizada;
 - Deve estar disponível para múltiplos usuários.
- Dados devem estar corretos:
 - Programas devem garantir as restrições de integridade (regras que estabelecem quando uma base está correta).
- Banco de dados é acessado por múltiplos usuários:
 - Programas devem implementar controle de acesso concorrente.
- Nem todo usuário pode ter acesso a qualquer informação:
 - Programas devem implementar controle de acesso.
- Os dados são de importância vital e não podem ser perdidos:

→ Mecanismos simples de cópia e *backup* não são suficientes.

- Após falha, o banco de dados deve ser recuperado rapidamente;
- Transações confirmadas ao usuário não podem ser reprocessadas;
- Programas devem implementar mecanismos de tolerância a falhas.

Processamento tradicional de arquivos x SGBD

A Tabela 1, demonstra uma breve comparação entre o processamento de arquivos x um SGBD.

Tabela 1 – Processamento entre arquivos x SGBD

Processamento de arquivos	SGBD – Banco de dados
Definição dos dados é a parte do código dos programas das aplicações	A definição dos dados: são o armazenamento como metadados (dicionário de dados).
Dependência entre aplicações específicas e os dados.	Dados dispostos de uma forma genérica para diversas aplicações. (Compartilhamento de dados)
Representação dos dados no nível físico	Representação conceitual dos dados (maior facilidade de manutenção)
Difícil controle sobre regras que estão distribuídas por diferentes aplicações	Controle de segurança Tolerância a falhas Controle de acesso concorrente Ambiente auto - contido

Fonte: o autor (2021)

Características de um SGBD

- **Integração de Dados:**

→ Controle ou eliminação de redundância: imaginemos um sistema sendo executado em diversos computadores, e este sistema possui os dados armazenados em um único local. O Controle de redundâncias organiza esse fluxo, para que dados inconsistentes não sejam armazenados, uma série de regras são criadas para que os dados sejam armazenados corretamente e não tenha dados duplicados ou inválidos.

→ Fundamental para o crescimento de BDs.

- **Compartilhamento de Dados:**

→ Controle de concorrência (Multiusuário): Todo SGBD deve possuir um controle de concorrência de dados, ou seja, irá garantir a escrita/leitura dos dados sem erros. Pois elas podem estar sendo disponibilizadas a mais de um usuário e não poderá prejudicar ambos.

- **Processamento de Transações:**

- Atomicidade (Tudo ou Nada).
- **Independência de Programa-Dados:**
 - Abstração de Dados + Catálogo (metadados).
- **Integridade:**
 - Regras de Negócio (Simples, Complexas);
 - Regras de Integridade (Validação de Consistência);
 - Restrições implementadas no SGBD: um item que também entra em segurança, o controle da integridade deverá impedir que aplicações ou acessos que possam comprometer a integridade dos dados sejam feitos. Deverá garantir o acesso somente de pessoas autorizadas e ainda de acordo com os níveis de acesso.
- **Suporte a Múltiplas Visões dos Dados:**
 - Diferentes visões de um mesmo BD.
- **Controle de Segurança:**
 - Principal funcionalidade de um SGBD: um SGBD deve possuir regras para restringir e garantir o acesso somente de pessoas autorizadas, com o acesso ao banco e qual o nível de acesso que o usuário irá possuir (Leitura, escrita, alteração, exclusão, etc.). Outro quesito que um SGBD possui e faz parte da segurança, são as cópias e a recuperação de dados em caso de falhas, permitindo que informações sejam recuperadas a partir de outro local. Grandes corporações possuem *backups* de seus bancos de dados até em países diferentes para garantir que nenhuma catástrofe destrua os dados armazenados.
- **Múltiplas Interfaces**
 - Um SGBD deve possuir uma forma de acesso gráfica, em linguagem natural, menus de acesso e em SQL. Podendo ser acessada diretamente, não sendo necessário passar pela aplicação que a utiliza.
- **Serviços de Manutenção:**
 - Backup, Recovery, Conversão de dados e monitoramento de desempenho;
 - Cópias de segurança são imprescindíveis para a segurança de qualquer informação em um SGBD. Pois estamos sujeitos a falhas tanto de *hardware* quanto de *software* e através de mecanismos previamente ajustados podemos recuperar as informações minimizando as perdas.
- **Flexibilidade:**
 - Passível a mudanças de espaço e alocação de memória.
- **Esquematização:**

- Normalmente em um banco de dados, as tabelas se relacionam entre si, um SGBD deve fornecer mecanismos que compreendem estes relacionamentos. Deve compreender por exemplo, um diagrama de classes UML.

- **Economia de Escala:**

- Reduz custos de operação e gerenciamento no servidor e clientes.



VIDEOAULA 02

Utilize o QRcode para assistir!

Agora, assista ao vídeo que aborda as características de um SGBD.



Tipos de um SGBD

Existem SGBDs de muitas formas e tamanhos. Por algumas centenas de dólares ou até mesmo de forma gratuita, você pode comprar um SGBD para o seu computador *desktop*. Para sistemas maiores os SGBDs podem ser muito mais caros. Muitos SGBDs são baseados em *mainframe* e alugados por organizações. SGBDs desta escala são altamente sofisticados e seria extremamente caro para desenvolver a partir do zero. Portanto, é mais barato para uma organização alugar um programa que desenvolvê-lo. Uma vez que há uma variedade de SGBDs disponíveis, você deve conhecer algumas das características básicas, bem como os pontos fortes e fracos, dos principais tipos.

Existem quatro tipos estruturais de sistemas de gerenciamento de banco de dados: hierárquico, rede, relacional e orientado a objetos.

Banco de dados Hierárquico

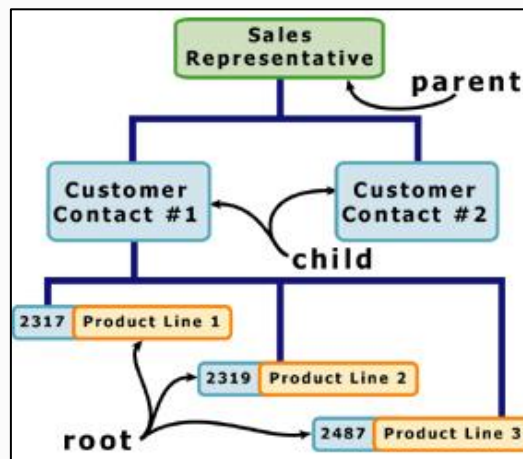
Banco de dados hierárquico, comumente usados em computadores *mainframe*. É um dos mais antigos métodos de organização e armazenamento de dados, e ainda é utilizado por algumas organizações para a realização de reservas de viagens. Um banco de dados hierárquico é organizado em forma de pirâmide, como os galhos de uma árvore. Áreas afins ou registros são agrupados de modo que registros de nível não são mais elevados que outros registros inferiores, assim como os pais em uma árvore genealógica de família que fica acima das crianças.

Com base nesta analogia, o registro principal no topo da pirâmide é chamado registro raiz. Um registro de criança sempre tem apenas um registro pai ao qual ele está ligado, como em uma árvore familiar normal. Em contraste, um registro pai pode ter mais de um registro filho a ele ligado. Bancos de dados hierárquicos trabalham movendo de cima para baixo. A pesquisa de registro é realizada começando pelo topo da pirâmide e indo para baixo por meio da árvore de pai para filho até que o registro da criança apropriada seja encontrado. Além disso, cada uma das crianças também pode ser um dos pais com crianças abaixo dele.

A vantagem de Bancos de Dados hierárquicos é que podem ser acessados e atualizados rapidamente porque a estrutura do tipo árvore e as relações entre os registros são previamente

definidas. No entanto, esse recurso é uma faca de dois gumes. A desvantagem deste tipo de estrutura de banco de dados é que cada criança na árvore pode ter apenas um pai, e os relacionamentos ou ligações entre as crianças não são permitidas, mesmo se elas fazem sentido do ponto de vista lógico. Bancos de dados hierárquicos são tão rígidos em seu projeto que a adição de um novo campo ou registro requer que o banco de dados inteiro seja redefinido. A Figura 3 demonstra um exemplo de Banco de Dados hierárquico.

Figura 3 - Exemplo de banco de dados hierárquico



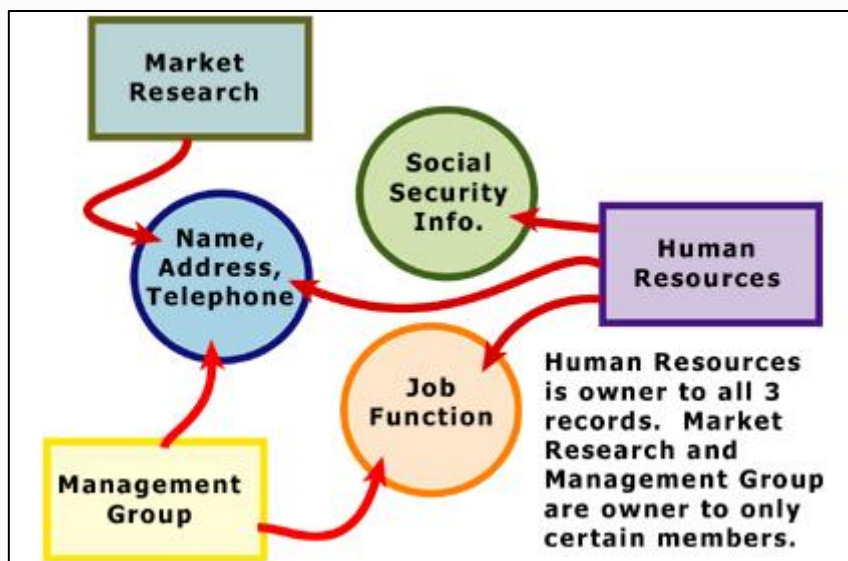
Fonte: Gomes (2021)

Banco de dados de Rede

Bancos de Dados de rede são semelhantes aos Bancos de Dados hierárquicos também com uma estrutura hierárquica. Existem algumas diferenças fundamentais, no entanto. Em vez de olhar como uma árvore de cabeça para baixo, um banco de dados de rede se parece mais com uma teia de aranha ou uma rede interligada de registros. Em bancos de dados de rede, as crianças são chamadas de membros e os pais são chamados proprietários. A diferença mais importante é que cada criança ou membro pode ter mais de um pai (ou dono).

Como bancos de dados hierárquicos, Bancos de Dados de rede são usados principalmente em computadores *mainframe*. Como mais conexões podem ser feitas entre diferentes tipos de dados, bancos de dados de rede são considerados mais flexíveis. No entanto, duas limitações devem ser consideradas quando se utiliza este tipo de banco de dados. Similar aos bancos de dados hierárquicos, Bancos de Dados de rede devem ser definidos com antecedência. Existe também um limite para o número de ligações que podem ser feitas entre os registros. A Figura 4 demonstra um exemplo de Banco de Dados de Rede.

Figura 4 - Exemplo de banco de dados de Rede



Fonte: Gomes (2021)

Banco de dados Relacionais

Em bancos de dados relacionais, a relação entre as tabelas de dados é relacional. Bancos de dados relacionais conectam dados em tabelas diferentes, usando elementos comuns de dados ou um campo chave. Dados em bancos de dados relacionais são armazenados em tabelas diferentes, cada uma com um campo chave que identifica cada linha ou registro. Bancos de dados relacionais são muito mais flexíveis do que as próprias estruturas de dados hierárquicos ou rede. Em bancos de dados relacionais a ligação entre as tabelas são chamadas de relações, as tuplas designam uma linha ou registro, e as colunas são referidas como atributos ou campos.

Bancos de dados relacionais trabalham no princípio de que cada tabela tem um campo chave que identifica unicamente cada linha, e que estes campos chave podem ser usados para ligar uma tabela de dados a outra. Deste modo, uma tabela pode ter uma linha formada por um número de conta de cliente, tal como o campo chave, juntamente com o endereço e número de telefone. O número de conta do cliente nesta tabela pode estar ligada a uma outra tabela de dados que inclui também o número de conta do cliente (um campo de chave), mas, neste caso, contém informações sobre a devolução de produtos, incluindo um número de ordem (um outro campo de chave). Este campo chave pode ser ligado a uma outra tabela que contém números de itens e outras informações do produto, tais como: local de produção, cor e outros dados. Portanto, usando esse banco de dados, as informações dos clientes podem ser ligadas às informações específicas do produto.

O banco de dados relacional se tornou bastante popular, por duas razões principais. Em primeiro lugar, os bancos de dados relacionais podem ser usados com pouca ou nenhuma formação. Segundo, as entradas de banco de dados podem ser modificadas sem redefinir a sua estrutura inteira. A desvantagem de usar um banco de dados relacional é que a busca de dados pode levar mais tempo do que se outros métodos são usados. A Figura 5 demonstra um exemplo de Banco de Dados Relacional.

Figura 5 - Exemplo de Banco de Dados Relacional

Database 1			
	First Name	Last Name	Social Security No.
2	John	Smith	010-22-9432
3	John	Smith	003-63-0037
4	John	Smith	020-45-9326
5	Sally	Smith	
6	Steve	Smith	

Database 2		
	Date of Birth	Social Security No.
2	6/12/82	010-22-9432
3	5/9/40	003-63-0037
4	12/11/57	020-45-9326
5	08/06	289-56-4321
6	07/09	170-54-2334

Database 3		
	Address	Social Security No.
2	321 Byberry Road	010-22-9432
3	268 Monroe Avenue	003-63-0037
4	8120 Venshire Drive	020-45-9326
5	207 Congress Drive	289-56-4321
6	1519 Ashbury Lane	170-54-2334

Fonte: Gomes (2021)

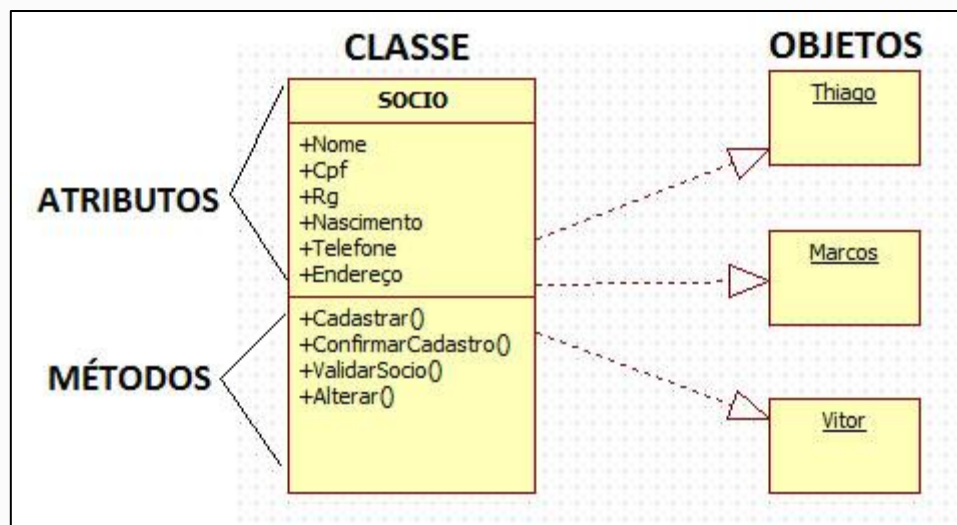
Banco de dados Orientado a Objetos

Capaz de lidar com muitos novos tipos de dados, incluindo gráficos, fotografias, áudio e vídeo. Bancos de dados orientados a objetos representam um avanço significativo sobre seus primos. Por outro lado, uma base de dados de objetos pode ser utilizada para armazenar os dados a partir de uma variedade de fontes de multimídia, tais como: fotografias e texto, e produzir como saída um formato multimídia.

Banco de dados orientado a objetos usam pequenos pedaços reutilizáveis de *software* chamados de objetos. Os próprios objetos são armazenados no banco de dados orientado a objetos. Cada objeto é composto de dois elementos: 1) um pedaço de dados (por exemplo, som, vídeo, texto ou gráfico), e 2) as instruções, ou programas de *software* chamados métodos, para o que fazer com os dados.

Banco de dados orientados a objeto tem duas desvantagens. Em primeiro lugar, eles são mais caros para se desenvolver. Segundo a maioria das organizações estão relutantes em abandonar ou converter esses bancos de dados que eles já investiram dinheiro no desenvolvimento e implementação. No entanto, os benefícios para os bancos de dados orientados a objeto são convincentes. A capacidade de misturar e combinar objetos reutilizáveis fornece uma capacidade multimídia incrível. Organizações de saúde, por exemplo, podem armazenar, controlar e recuperar tomografias, raios-X, eletrocardiogramas e muitas outras formas de dados cruciais. A Figura 6 demonstra um exemplo de Banco de Dados Orientado a Objetos.

Figura 6 - Exemplo de Banco de Dados Orientado a Objetos



Fonte: o autor (2021)



VIDEOAULA 03

Utilize o QRcode para assistir!

Agora, assista ao vídeo que aborda os diferentes tipos de SGBD.



INDICAÇÃO DE LEITURA

Leia o livro “Fundamentos de Banco de Dados: Uma abordagem prático-didática, por Marcio Porto Feitosa” para conhecer mais sobre o assunto introdutório de Banco de Dados.

INDICAÇÃO DE VÍDEO

Agora, assista ao trailer sobre o filme: “O Escândalo da Cambridge Analytica”. Disponível em: <https://www.youtube.com/watch?v=7-xgW4iGw>. Acesso em: 15 abr. 2021.

CURIOSIDADES

No site Perallis Security você encontra mais informações sobre roubo de dados, acesse o link abaixo: Disponível em: <https://www.perallis.com/news/curiosidades-sobre-roubos-de-dados>. Acesso em: 15 abr. 2021.

Aula 02 - Abordagem entre Banco de Dados e Arquivos convencionais

Nesta aula abordaremos os conceitos envolvendo Banco de Dados e armazenamento de arquivos convencionais.

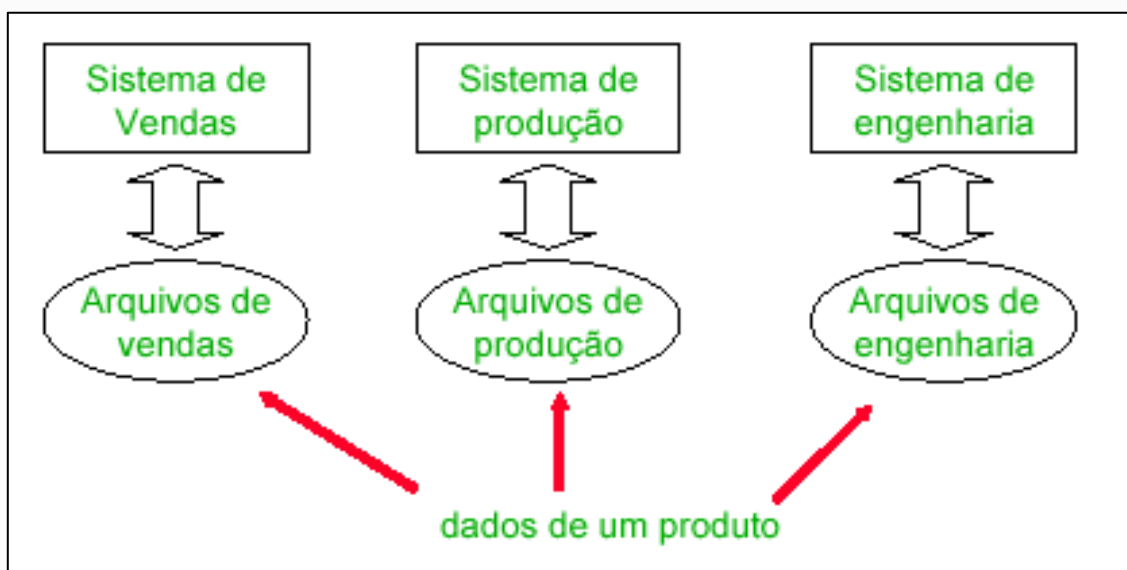
Incidentes envolvendo o armazenamento em arquivos convencionais

1. Redundância e inconsistência de dados:

- O mesmo dado pode estar duplicado em vários lugares e em vários sistemas (redundância);
- Essa redundância leva a altos custos de armazenamento e acesso;
- A redundância pode levar à inconsistência de dados, o que significa que várias cópias do mesmo dado podem estar diferentes umas das outras.
- A redundância ocorre porque:
 - Dados de diferentes aplicações não estão integrados;
 - Informatização gradual das funções nas organizações;
 - Se funções são informatizadas em separado, pode ocorrer a geração de arquivos separados.

A Figura 7 demonstra um exemplo para redundância e inconsistência de dados.

Figura 7 - Esquema de Redundância e inconsistência de dados



Fonte: o autor (2021)

- **Tipos de redundância:**

- Redundância simples:

- Ocorre quando um dado está representado no computador várias vezes.

- Redundância controlada:

- Existe a redundância, mas o *software* tem conhecimento disto e garante a sincronia dos dados entre as diversas representações.

- Redundância não controlada:

- Existe a redundância, mas quem controla a sincronia entre as representações são os usuários.

- **Consequências da redundância:**

- Repetição da mesma informação;

- Inconsistência dos dados;

- Dificuldade de extrair informações (relatórios);

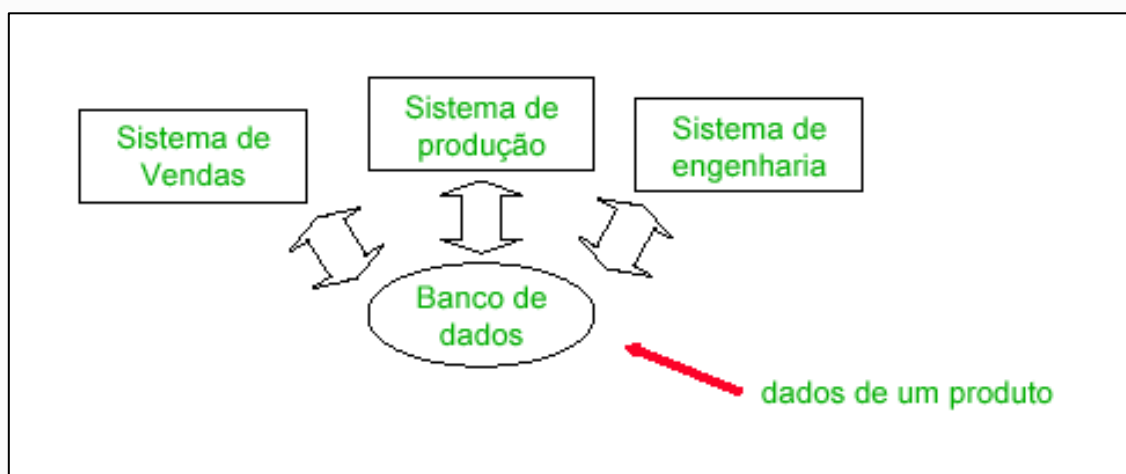
- Dados pouco confiáveis e de baixa disponibilidade (desempenho ruim).

- **Solução para a redundância:**

- Compartilhamento dos dados, formando um conjunto único de arquivos integrados que atendem a um conjunto de sistemas.

A Figura 8 demonstra um exemplo para a solução de redundância.

Figura 8 - Esquema para solução da redundância



Fonte: o autor (2021)

2. Dificuldade no acesso aos dados:

- O modo de armazenamento não permite agilidade na consulta. Ex.: geração de relatórios.

3. Dificuldade no acesso aos dados:

- Dados em sistemas isolados e com formatos diferentes (data, endereço completo), tornam difícil a tarefa de recuperar os dados adequados.

4. Problemas de Segurança:

- Nem todo usuário do sistema deve ter acesso a todos os dados disponíveis;
- Se programas forem adicionados ao sistema de maneira arbitrária, é difícil assegurar as restrições de segurança.

5. Problemas de Integridade

- Como garantir as restrições de consistência. “Regras de negócio”.



VIDEOAULA 01

Utilize o QRcode para assistir!

Agora, assista ao vídeo que aborda os incidentes envolvendo o armazenamento de banco de dados convencionais.



Usuários

Todo agrupamento de bancos de dados possui um conjunto de usuários de banco de dados. Estes usuários são distintos dos usuários gerenciados pelo sistema operacional onde o servidor executa. Os usuários possuem objetos de banco de dados (por exemplo, tabelas), e podem conceder privilégios nestes objetos para outros usuários controlando, assim, quem pode acessar qual objeto.

Administração de Banco de Dados (DBA)

Em um ambiente de banco de dados, o recurso primário é o banco de dados por si só e o recurso secundário o SGBD e os *softwares* relacionados. A administração destes recursos cabe ao Administrador de Banco de Dados, o qual é responsável pela autorização de acesso ao banco de dados e pela coordenação e monitoração de seu uso. Ou seja, ele coordena todas as atividades do sistema de banco de dados; possui boa compreensão dos recursos de informação da empresa e suas necessidades.

Suas funções incluem:

- Definição do esquema;
- Estrutura de armazenamento e definição de acesso aos dados;
- Esquema físico e organização;
- Concede acesso aos usuários;
- Cuida da integridade dos dados;

- Atua como elo com os usuários;
- Acompanha o desempenho, e responde as mudanças exigidas;
- Atividades de manutenção (*Backups*).

Projetista de Banco de Dados

O Projetista de Banco de Dados é responsável pela identificação dos dados que devem ser armazenados no banco de dados, escolhendo a estrutura correta para representar e armazenar dados. Muitas vezes, os projetistas de banco de dados atuam como "*staff*" do DBA, assumindo outras responsabilidades após a construção do banco de dados. É função do projetista também avaliar as necessidades de cada grupo de usuários para definir as visões que serão necessárias, integrando-as, fazendo com que o banco de dados seja capaz de atender a todas as necessidades dos usuários.

Usuários Finais

Existem basicamente três categorias de usuários finais que são os usuários finais do banco de dados, fazendo consultas, atualizações e gerando documentos:

- **Usuários casuais:** acessam o banco de dados casualmente, mas que podem necessitar de diferentes informações a cada acesso; utilizam sofisticadas linguagens de consulta para especificar suas necessidades;
- **Usuários novatos ou paramétricos:** utilizam porções pré-definidas do banco de dados, utilizando consultas pré-estabelecidas que já foram exaustivamente testadas;
- **Usuários sofisticados:** são usuários que estão familiarizados com o SGBD e realizam consultas complexas.

Analistas de Sistemas e Programadores de Aplicações

Os analistas determinam os requisitos dos usuários finais e desenvolvem especificações para transações que atendam estes requisitos, e os programadores implementam estas especificações como programas, testando, depurando, documentando e dando manutenção no mesmo. É importante que, tanto analistas quanto programadores, estejam a par dos recursos oferecidos pelo SGBD.



VIDEOAULA 02

Utilize o QRcode para assistir!

Agora, assista ao vídeo que aborda os tipos de usuários.



Modelos de dados e Abstração de dados

- **Modelo de dados:** conjunto de regras conceituais que podem ser usados para descrever a estrutura de um banco de dados;

- **Abstração de dados:** omissão de detalhes de armazenamento dos dados, que são desnecessários para a maioria dos usuários de bancos de dados (programadores).

O grande objetivo de um sistema de bancos de dados é prover o usuário com uma visão abstrata dos dados. O sistema omite certos detalhes de como os dados são armazenados e mantidos.

A complexidade do banco de dados está oculta dos usuários inexperientes através de diversos níveis de abstração que simplificam a interação do usuário com o sistema.

O usuário final deve ter uma visão abstrata dos dados. As técnicas, detalhes e regras deste armazenamento devem ser de responsabilidade dos analistas e DBAs.

Um SGBD possui a descrição dos tipos de informações que estão armazenadas em seu banco de dados.

- Modelo de dados = descrição dos dados (esquema);
- Banco de Dados (BD) = Dados + Modelo de Dados;

Para construir o modelo de dados, usa-se uma linguagem de modelagem de dados:

- Linguagem textual ou;
- Linguagem gráfica.

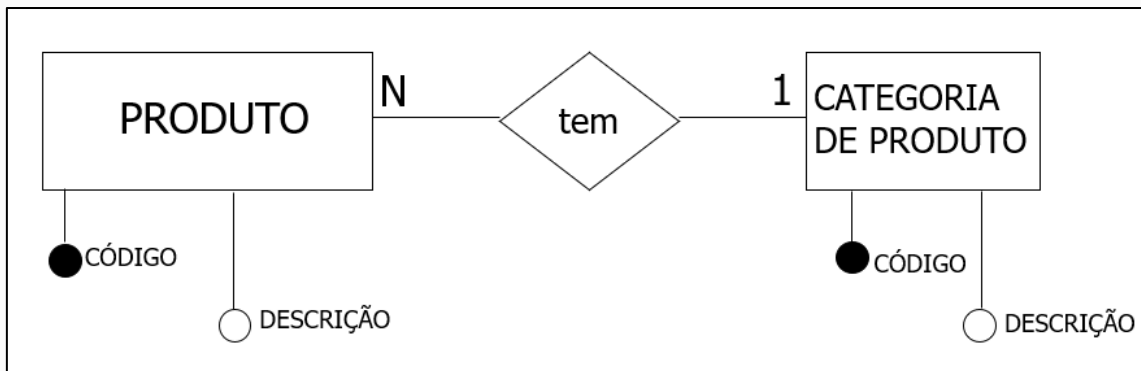
A apresentação do modelo de dados denomina-se esquema do banco de dados. Normalmente, no projeto de banco de dados existem três níveis de abstração para esta apresentação, esses modelos são denominados:

- Conceitual;
- Lógico;
- Físico.

Modelo conceitual

- É a descrição mais abstrata do banco de dados, independentemente do tipo de SGBD a ser utilizado;
- Registra os dados que podem aparecer no banco, mas não como estes dados estão armazenados;
- É o ponto de partida do projeto de banco de dados;
- O modelo mais difundido é a abordagem entidade-relacionamento (E-R) desenvolvido por Peter Chen;
- Representado na forma de entidades e relacionamentos. Exemplo na figura 9.

Figura 9 - Demonstração do funcionamento do Esquema Conceitual



Fonte: o autor (2021)

Modelo Lógico

- Descrição de um banco de dados no nível de abstração visto pelo usuário;
- É dependente do tipo de banco de dados utilizado;
- No caso do modelo relacional, os dados estão organizados na forma de tabelas;
- Modelo lógico para o exemplo anterior:
 - TipoDeProduto (CodTipoProduto, DescrTipoProd);
 - Produto (CodProd, DescrProd, PreçoProd, CodTipoProd).
- Detalhes de armazenamento interno não fazem parte do modelo lógico, e são representados no modelo físico.

Modelo Físico

- Descrição de como a base de dados é armazenada internamente no banco de dados (dependente do SGBD);
- O modelo físico é utilizado buscando-se otimizar a performance;
- As linguagens para o modelo físico variam de produto a produto (padrão é o SQL-ANSI).

Instâncias e Esquemas

- É importante a distinção entre a descrição do banco de dados e o banco de dados de fato:
 - O esquema do Banco de Dados é definido na fase de projeto;
 - Espera-se que um esquema de banco de dados não seja alterado frequentemente;
 - Os dados no banco de dados podem ser alterados frequentemente;
 - O estado de um banco de dados é definido pelos seus dados em um determinado momento (*snapshot*);
 - Os dados em um determinado momento em um banco de dados podem-se denominar ocorrência ou instância;

- Cada vez que dados são incluídos, alterados ou excluídos em um banco, muda-se de um estado para outro.
- Quando um BD é criado, diz respeito ao esquema de banco e ele passa ao seu primeiro estado quando seus dados são populados ou carregados pela primeira vez.

Resumo:

- Instância: “Fotografia” momentânea do banco de dados;
- Esquema - o projeto total do banco de dados (projeto lógico e projeto físico);

Independência de Dados

- Independência Lógica de Dados - Capacidade de modificar o esquema conceitual/lógico sem a necessidade de reescrever os programas.
 - Ex.: adicionar tabelas.
- Independência Física de Dados - Capacidade de modificar o esquema físico sem a necessidade de reescrever os programas.
 - Ex.: mudar tabela de disco ou diretório.
- A independência lógica de dados é mais difícil de ser alcançada do que a independência física, pois os programas são bastantes dependentes da estrutura lógica dos dados que eles acessam.

Structured Query Language (SQL - linguagem de consulta estruturada)

É um tipo de linguagem de consulta que é amplamente utilizada para executar operações usando bancos de dados relacionais. Lembre-se que os bancos de dados relacionais são compostos por tabelas com linhas e colunas. A Linguagem SQL pode ser usada para recuperar informações de tabelas relacionadas em um banco de dados ou para selecionar e recuperar informações de linhas e colunas específicas em uma ou mais tabelas. Uma das chaves para a compreensão de como funciona o SQL em um banco de dados relacional é perceber que cada tabela e coluna tem um nome específico associado a ele. Para consultar uma tabela, o usuário especifica o nome da tabela (indicando as linhas a ser exibido) e os nomes das colunas a serem exibidas.

Uma consulta SQL típica contém três elementos-chave:

- **SELECT** (os nomes das colunas a serem exibidos);
- **FROM** (indica o nome da tabela da qual o nome das colunas será derivado);
- **WHERE** (descreve a condição para a consulta).

Para ilustrar a aplicação deste tipo de consulta, vamos supor que um determinado usuário deseje consultar um banco de dados relacional contendo informações sobre os doadores para uma organização de caridade. Se o usuário quiser saber o nome e endereço de todos os indivíduos que doam US\$ 100 ou mais, a seguinte consulta poderá ser usada:

- **SELECT** Nome, Endereço
- **FROM** Lista-de-Doadores
- **WHERE** Doacao > 100

Uma vez que este comando foi executado, o computador irá exibir uma lista de doadores que atenda aos critérios pré-definidos. Neste caso, todos os dados são extraídos a partir de uma única tabela. Consultas semelhantes podem ser feitas para extrair dados de várias tabelas. Tal estratégia pode ser usada para analisar as informações do cliente, envolvendo dados de faturamento e dados de ordem, usando duas tabelas separadas.

O que pode fazer o SQL?

- SQL pode executar consultas em um banco de dados;
- SQL pode selecionar e recuperar dados a partir de um banco de dados;
- SQL pode inserir registros em um banco de dados;
- SQL pode atualizar registros em um banco de dados;
- SQL pode excluir registros de um banco de dados;
- SQL pode criar novos Bancos de Dados;
- SQL pode criar novas tabelas em um banco de dados;
- SQL pode criar procedimentos armazenados (*stored procedures*) em um banco de dados;
- SQL pode criar visões (*views*) em um banco de dados;
- SQL pode definir permissões em tabelas, procedimentos e visões.

SQL é um padrão

Embora o SQL seja um padrão ANSI (*American National Standards Institute*), há muitas versões diferentes dessa linguagem. No entanto, para ser compatível com o padrão ANSI, a linguagem deve possuir pelo menos, os comandos principais (como SELECT, UPDATE, DELETE, INSERT) de maneira semelhante.

Nota: a maioria dos programas de banco de dados SQL também têm suas próprias extensões proprietárias, além do padrão SQL!

Importante:

- SQL não é *case sensitive* (não é sensível a maiúsculas);
- Alguns sistemas de banco de dados requerem um ponto e vírgula no final de cada instrução SQL.

DML, DDL e DCL

A linguagem SQL pode ser dividida em três partes:

- **DDL - linguagem de definição de dados** (*Data Definition Language*) permite a criação, eliminação e alteração da estrutura física da Base de Dados. Ela também define os índices (chaves), especifica as ligações entre as tabelas, e impõe restrições entre tabelas. As declarações mais importantes DDL em SQL são:
 - CREATE DATABASE - cria um novo banco de dados;
 - ALTER DATABASE - altera um banco de dados;
 - CREATE TABLE - cria uma nova tabela;
 - ALTER TABLE - modifica uma tabela;
 - DROP TABLE - apaga uma tabela;

- **CREATE INDEX** - cria um índice (chave de busca);
- **DROP INDEX** - exclui um índice.
- **DML - linguagem de manipulação de dados** (*Data Manipulation Language*) permite a inserção, edição e exclusão dos dados das tabelas. A consulta e os comandos de atualização formam a parte DML de SQL que são:
 - **SELECT** - extrai dados de um banco de dados;
 - **UPDATE** - atualiza os dados em um banco de dados;
 - **DELETE** - exclui dados de um banco de dados;
 - **INSERT** - insere novos dados em um banco de dados.
- **DCL - Linguagem de Controle de Dados** (*Data Control Language*) atribui permissões aos objetos através dos comandos Grant, revoke e Deny. A Linguagem de Controle de Dados (DCL) é um subconjunto do SQL (*Structured Query Language*) que permite que os administradores de banco de dados configurem o acesso de segurança para bancos de dados relacionais. DCL é o mais simples dos subconjuntos SQL, visto que consiste em apenas três comandos: GRANT, REVOKE e DENY. Combinados, esses três comandos oferecem aos administradores a flexibilidade para definir e remover permissões de banco de dados de forma extremamente granular.
 - **GRANT** é utilizado para adicionar novas permissões para um usuário;
 - **REVOKE** é utilizado para remover o acesso de banco de dados de um usuário;
 - **DENY** é utilizado para impedir explicitamente que um usuário receba uma permissão especial.

Tipos de Dados

Antes de criar uma tabela você precisa entender as diferenças entre os tipos de dados que você pode utilizar em uma coluna. Os tipos de dados SQL se classificam em 13 tipos de dados primários e de vários sinônimos válidos reconhecidos por tais tipos de dados. Os tipos de dados primários são:

Tabela 1 – Tipos de dados primários

Tipos de dados	Longitude	Descrição
BINARY	1 byte	Para consultas sobre tabela anexa de produtos de banco de dados que definem um tipo de dados Binário.
BIT	1 byte	Valores Sim/Não ou <i>True/False</i> .
BYTE	1 byte	Um valor inteiro entre 0 e 255.
COUNTER	4 bytes	Um número incrementado automaticamente (de tipo <i>Long</i>)

CURRENCY	8 bytes	Um inteiro escalável entre 922.337.203.685.477,5808 e 922.337.203.685.477,5807
DATETIME	8 bytes	Um valor de data ou hora entre os anos 100 e 9999.
SINGLE	4 bytes	Um valor em ponto flutuante de precisão simples com uma classificação de - 3.402823*10 ³⁸ a -1.401298*10 ⁻⁴⁵ para valores negativos, 1.401298*10 ⁻⁴⁵ a 3.402823*10 ³⁸ para valores positivos, e 0.
DOUBLE	8 bytes	Um valor em ponto flutuante de dupla precisão com uma classificação de - 1.79769313486232*10 ³⁰⁸ a - 4.94065645841247*10 ⁻³²⁴ para valores negativos, 4.94065645841247*10 ⁻³²⁴ a 1.79769313486232*10 ³⁰⁸ para valores positivos, e 0.
SHORT	2 bytes	Um inteiro curto entre -32,768 e 32,767.
LONG	4 bytes	Um inteiro longo entre -2,147,483,648 e 2,147,483,647.
LONGTEXT	1 byte por caractere	De zero a um máximo de 1.2 gigabytes.
LONGBINARY	Segundo se necessite	De zero 1 gigabyte. Utilizado para objetos OLE.
TEXT	1 byte por caractere	De zero a 255 caracteres.

Fonte: o autor (2021)



VIDEOAULA 03

Utilize o QRcode para assistir!

Agora, assista ao vídeo que aborda a Linguagem SQL.



INDICAÇÃO DE LEITURA

Leia o livro “Fundamentos de Banco de Dados: uma abordagem prático-didática, por Marcio Porto Feitosa” para conhecer mais sobre o assunto introdutório de Banco de Dados.

INDICAÇÃO DE VÍDEO

Agora, assista ao trailer do filme “Snowden – Herói ou Traidor”. Disponível em: <https://www.youtube.com/watch?v=cJkp4bIUyl8>. Acesso em: 15 abr. 2021.

Encerramento

Como vimos, os SGBDs são de várias formas. As diferentes estruturas e definição de SGBDs foram abordadas e contrastadas num esforço para ajudar a demonstrar seus pontos fortes e fracos. Como um trabalhador do conhecimento, você pode um dia ser convidado a selecionar e tomar decisões sobre um SGBD. Esta lição fornece um ponto de partida para a compreensão das questões envolvidas.

Usuários confiam na linguagem de consulta para manipular dados em um banco de dados. O SQL, por exemplo, é largamente utilizado para gerir Bancos de Dados relacionais. É importante entender o propósito de linguagens de consulta. Talvez um dia você terá a oportunidade de usar essas linguagens para ajudar sua organização a manipular e extrair o significado de um dos seus muitos bancos de dados.

Hoje em dia, os novos sistemas de base de dados são quase exclusivamente do tipo relacional.

Databases relacionais substituíram os modelos anteriores porque esses sistemas têm valiosos atributos que a distinguem como superior. Provavelmente o mais importante é que você pode mudar a estrutura de dados sem alterações nas aplicações. Suponha, por exemplo, que você adicione uma ou mais colunas numa tabela. Você não precisa alterar nenhum aplicativo que o sistema vai continuar a processar. Claro, se você remover uma coluna que uma aplicação existente utiliza, você vai ter problemas.

Agora que você completou esta unidade, você deve ser capaz de:

- Comparar e contrastar a estrutura dos diferentes sistemas de gerenciamento de banco de dados;
- Definir Bancos de Dados hierárquicos;
- Definir os bancos de dados de rede;
- Definir Bancos de dados orientados a objeto;
- Definir a linguagem de consulta;
- Descrever um exemplo de uma linguagem de consulta estruturada;
- Discutir as funções e capacidades de uma linguagem de consulta estruturada;
- Definir Bancos de Dados relacionais;
- Discutir as funções e capacidades de uma Base de Dados Relacional;
- Definir os objetos de um Banco de Dados Relacional.

Esperamos que este guia o tenha ajudado compreender a organização e o funcionamento de seu curso. Outras questões importantes relacionadas ao curso serão disponibilizadas pela coordenação.
Grande abraço e sucesso!

