

Apresentação

Atualmente, as empresas acumulam um volume crescente de informações que necessitam de organização meticulosa para garantir a acessibilidade e a utilidade desses dados. De acordo com um relatório da IDC, de 2024, o volume global de dados está projetado para crescer a uma taxa anual composta de 23%, destacando a necessidade de soluções robustas para gerenciamento de informações. Desde relatórios financeiros até registros de clientes, cada pedaço de informação é vital para as operações do negócio e para a tomada de decisões estratégicas.

Nesse contexto, os bancos de dados surgem como ferramentas poderosas, transcendendo as funcionalidades de simples planilhas eletrônicas ou documentos de texto. Dados de um estudo recente da Gartner indicam que 85% das empresas listadas na Fortune 500 utilizam sistemas avançados de banco de dados para suas operações diárias, evidenciando sua capacidade de armazenar, organizar e recuperar informações de forma eficiente. Eles servem como base para sistemas complexos que exigem um manuseio de dados robusto e seguro, garantindo a integridade e a disponibilidade de informações críticas.

Nesta Unidade de Aprendizagem, você irá entender a importância da utilização do banco de dados e a diferença entre os sistemas de banco de dados existentes, bem como suas aplicações mais comuns. Você verá, ainda, os aspectos positivos e negativos de cada opção de uso dos sistemas abordados.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Identificar a importância da utilização do banco de dados no contexto organizacional.
- Reconhecer as principais ferramentas de administração de banco de dados.
- Relacionar os principais ambientes de banco de dados.

Infográfico

Em um âmbito em que os dados são um ativo essencial, assegurar a integridade deles nos bancos de dados é uma tarefa imprescindível para qualquer organização. Negócios de todos os portes e setores dependem da precisão e da confiabilidade das informações armazenadas para realizar operações diárias e tomar decisões estratégicas. Por exemplo, estudos recentes mostram que empresas que investem em segurança de dados têm 40% menos chances de enfrentar problemas relacionados à perda de informações.

Garantir que cada transação seja processada corretamente é um desafio constante para os sistemas de gerenciamento de banco de dados. Eles devem ser capazes de lidar com a complexidade e a demanda de um ambiente empresarial sempre ativo, em que a perda ou a corrupção de informações pode ter consequências severas. De acordo com a Gartner, até 2025, 75% dos bancos de dados serão implementados ou migrados para a nuvem, aumentando a necessidade de robustas medidas de integridade e segurança.

Neste Infográfico, você irá conhecer as quatro propriedades fundamentais das transações em um banco de dados — Atomicidade, Consistência, Isolamento e Durabilidade (ACID) — e entender como elas colaboram para manter a robustez e a segurança das informações.

INTEGRIDADE DAS TRANSAÇÕES EM UM BANCO DE DADOS QUATRO MEDIDAS:

A transação engloba todas as atividades que um SGBD realiza após uma interação que o usuário tem com o próprio sistema. Para manter a integridade dessas atividades, há quatro medidas que devem ser seguidas.

Confira, a seguir, essas quatro medidas.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

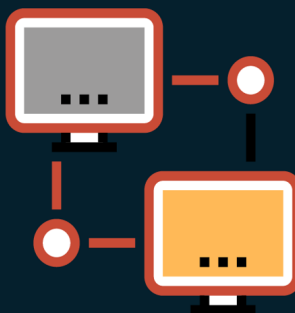


1. ATOMICIDADE:

- > Assegura que todas as etapas de uma transação sejam completadas ou, em caso de falha, nenhuma delas seja concluída.
- > Segundo a IBM, cerca de 80% das transações financeiras utilizam princípios de atomicidade para garantir segurança e precisão.

2. CONSISTÊNCIA

- > Mantém a lógica interna do banco de dados.
- > De acordo com a Oracle, sistemas de reservas que utilizam princípios de consistência reduzem erros em até 90%.



3. ISOLAMENTO:

- > Previne que transações simultâneas se cruzem e causem inconsistências.
- > Estudos da Microsoft indicam que o isolamento adequado de transações pode aumentar a eficiência do sistema em até 50%.

4. DURABILIDADE

- > Certifica que os resultados de uma transação persistirão mesmo diante de interrupções.
- > A AWS destaca que a durabilidade dos dados em seus sistemas é garantida com uma taxa de sucesso de 99,9%.



As propriedades ACID são essenciais para a confiabilidade dos bancos de dados. Sua correta implementação garante que as interações de dados sejam manuseadas com integridade, segurança e eficiência.

Conteúdo do Livro

Manejar vastas quantidades de informações é um desafio central para as organizações modernas. Bancos de dados avançados são essenciais para armazenar e recuperar dados de forma eficiente, proporcionando suporte fundamental para relatórios detalhados e análises críticas que influenciam decisões estratégicas. Por exemplo, a Gartner destaca que empresas que utilizam bancos de dados avançados reportam até 30% mais eficiência em suas operações.

Escolher a tecnologia correta para gerenciamento de dados é fundamental para extrair valor das informações coletadas. Em ambientes em que cada segundo conta, sistemas de banco de dados robustos e bem estruturados superam métodos tradicionais, como planilhas eletrônicas ou documentos de texto, oferecendo maior eficiência e confiabilidade. Dados da Forrester Research indicam que 70% das empresas que adotam tecnologias avançadas de banco de dados observam uma melhoria significativa na qualidade deles.

No capítulo **Conceitos de banco de dados**, base teórica desta Unidade de Aprendizagem, você irá refletir sobre a importância dos sistemas de banco de dados. Você verá as vantagens e práticas recomendadas para modelar e desenvolver bancos de dados, transformando informações em *insights* acionáveis para uma gestão empresarial eficaz.

Boa leitura.



MODELAGEM E DESENVOLVIMENTO DE BANCO DE DADOS

Fabício Felipe
Meleto Barboza

Conceitos de banco de dados

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar a importância da utilização do banco de dados no contexto organizacional.
- Reconhecer as principais ferramentas de administração de banco de dados.
- Relacionar os principais ambientes de banco de dados.

Introdução

Neste capítulo, você vai estudar a necessidade do uso de banco de dados no lugar de outra forma de guardar as informações, como, por exemplo, planilhas eletrônicas ou documentos de textos.

Adicionalmente, você também conseguirá vislumbrar a diferença entre os sistemas de bancos de dados existentes atualmente, suas aplicações mais comumente utilizadas nas corporações em geral e, também, os aspectos positivos e negativos de cada opção de uso dos sistemas abordados.

Necessidade de uso dos bancos de dados

Imagine que você trabalha em uma pequena loja que tem como principal fonte de receita a comercialização de capinhas de celulares. A cada entrada de mercadoria no estoque ou baixa ao realizar uma venda, é registrada a respectiva ação em um caderno de registros. Tome por base que essa lojinha venda uma capinha de celular por dia: ao longo de um mês, haverá 30 registros de venda e de 2 a 3 registros de entrada de mercadoria. Bem fácil controlar na caderneta, certo?

Mas então o dono da lojinha recebeu uma herança e resolveu comprar um espaço no shopping mais movimentado da cidade para expandir os negócios. De 30 capinhas ao mês, agora a venda aumentou para três mil capinhas. O seu trabalho de registro na caderneta começou a ficar complexo e a tomar cada vez mais tempo do seu dia. Para resolver o problema, um sobrinho do dono da loja fez uma planilha eletrônica linda, cheia de fórmulas e que resolve mais rapidamente seu problema de cadastro.

Como um excelente homem de negócios, seu chefe resolveu abrir 20 filiais novas. Temos, então, 60 mil registros de venda de capinhas mais os registros de entrada de estoque por mês. Se colocarmos que a quantidade de linhas que uma planilha eletrônica suporta atualmente é cerca de um milhão, você iria enchê-la em menos de 18 meses (ou um ano e meio). Além disso, essa planilha ocuparia muito espaço no computador e você levaria muito tempo para recuperar as informações inseridas ali em cada momento de venda.

Então você se cansa de trabalhar nessa loja, entra em uma empresa de telefonia multinacional e se pega imaginando quantas planilhas eles possuem para controlar todas as ligações que seus clientes fazem a cada minuto. Segundo a ANATEL (BRASIL, 2018), a quantidade de linhas telefônicas fixas no Brasil já passou a marca de 40 milhões e meio. Coloque nessa equação a quantidade de linhas telefônicas móveis (celulares) e temos o caos das planilhas eletrônicas.

A solução para qualquer um dos casos apresentados, desde a lojinha que vendia 30 capinhas de celular por mês até a grande companhia telefônica com milhões de novos registros que precisam ser processados a cada segundo, é uma só: banco de dados.



Link

Acesse o link a seguir e leia a notícia da ANATEL, que apresenta a enorme quantidade de linhas fixas existentes em todos os estados brasileiros.

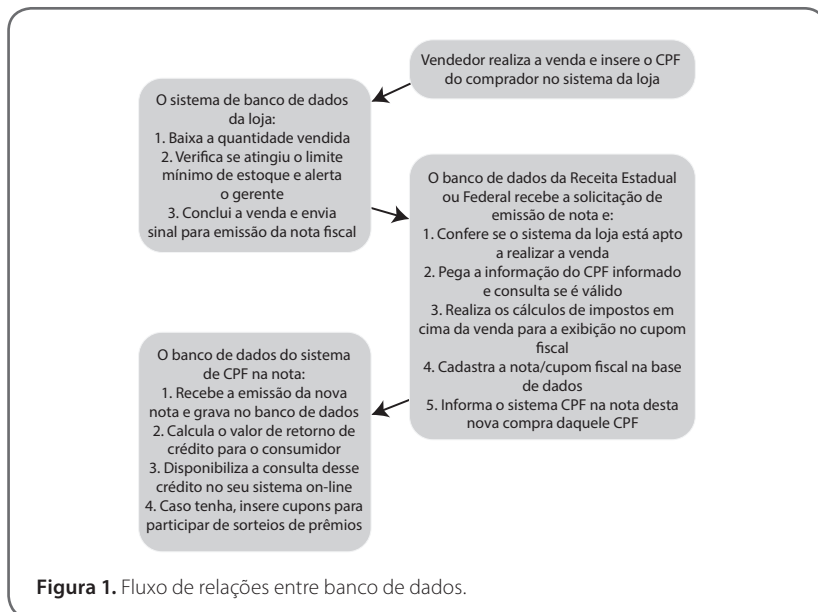
<https://goo.gl/6wcb8D>

Da mesma forma, a atividade pertinente a qualquer área de uma organização requer um poder grande de armazenamento de informações, ordenação, busca e edição. Pense em um sistema on-line popular qualquer, como, por exemplo, o

Facebook. Inúmeros bancos de dados são utilizados para prover a plataforma da rede social, cada qual com seu objetivo: um banco de dados somente para o cadastro dos usuários, outro banco de dados para as postagens dos usuários, um terceiro banco de dados para armazenar as fotos dos usuários, mais outro banco de dados para o *feed* de notícias, um quinto banco de dados para o *chat* da plataforma, etc.

Cada sistema tem sua própria divisão de banco de dados, mas você pode ter certeza de que é necessário um ou mais banco de dados para suportar um *blog*, por mais simplista que possa parecer — até mesmo para grandes plataformas, como Facebook, LinkedIn, Instagram, WhatsApp, Twitter, sistemas bancários e demais sistemas.

Quando você efetua a compra de um produto qualquer para o qual é emitida uma nota fiscal eletrônica, o fluxo de funcionamento daquele ponto de venda (PDV) com a ação do colaborador do estabelecimento ao registrar seu CPF na nota, por exemplo, seria o que está visualmente representado na Figura 1.



Sendo assim, com a utilização de banco de dados, tanto o cadastro de novos registros, como a venda de uma capinha de celular do exemplo anterior, quanto

uma nova ligação por parte do cliente da grande companhia telefônica resulta em um novo registro no próprio banco de dados — portanto, uma nova linha da “planilha”. Como os bancos de dados, em sua maioria, estão conectados aos sistemas, esse registro é feito de forma automática e é garantida a integridade da informação.



Exemplo

Imagine o sistema de uma loja virtual ou sites de compra e venda. O relacionamento entre bancos de dados ocorre por segundo, muitas vezes, para o envio, troca, edição, exclusão e informe de operações. Sites como o do Mercado Livre, por exemplo, precisam de um banco de dados robusto e confiável.

Como administrar um banco de dados

A questão da administração de um banco de dados é ampla e complexa, sendo debatida por diversos autores. Em resumo, você deve ter conhecimento de que tipo de informação está alocada nas tabelas do seu banco de dados e, principalmente, como manipular as informações que ali estão inseridas da forma que necessita.

Um sistema de banco de dados é formado pelo *software* de banco de dados, o repositório de tabelas denominado *database* e as próprias tabelas em si ou *tables*. Dessa forma, um banco de dados para um sistema de agenda telefônica teria o serviço do banco de dados executado, uma *database* e, dentro desta, estariam as tabelas para gravar as informações de nome, endereço, contato telefônico, e-mail, aniversário, foto, etc.

Portanto, para você conseguir realizar uma boa administração do banco de dados, é necessário conhecer o negócio ou o projeto ao qual ele servirá. De nada adianta ter tabelas que não são necessárias, pois isso impacta na performance do serviço. No exemplo da agenda telefônica, se não existisse o campo “contato telefônico”, o sistema ficaria inútil, pois uma agenda telefônica precisa, obrigatoriamente, conter o campo para cadastro do telefone do contato.



Fique atento

No exemplo da agenda telefônica, existem campos obrigatórios e campos opcionais. Os campos “nome” e “contato telefônico” são obrigatórios. Já os campos “endereço”, “e-mail”, “aniversário” e “foto” são opcionais.

Essa diferenciação se deve ao fato de que uma agenda telefônica precisa, no mínimo, gravar o nome e o telefone de contato da pessoa. Portanto, o banco **deve** respeitar essa regra para ser bem construído.

Como passo fundamental na administração correta de um banco de dados, é essencial que você tenha a documentação base do sistema que utilizará do banco de dados sob sua administração. Esse documento contemplará a modelagem do banco de dados, os relacionamentos entre tabelas, como será gravada cada informação, quais campos são obrigatórios e quais são opcionais, a necessidade de indexação para melhorar a velocidade de consulta, etc.

A seguir, na Figura 2, você verá um exemplo simples de uma modelagem de banco de dados, na forma de entidade-relacionamento, para compreensão. Essa forma de ilustração do banco de dados é a mais comumente utilizada em documentações e projetos. Nela, cada retângulo representa uma tabela, cujo título é o seu nome. Para cada tabela, há uma relação de atributos (linhas abaixo do título) e linhas ligando essas tabelas, que representam os relacionamentos entre elas. Por ora, você precisa atentar para o nome e cada atributo das tabelas a seguir, pois iremos estudar esse e outros modelos de banco de dados mais adiante.

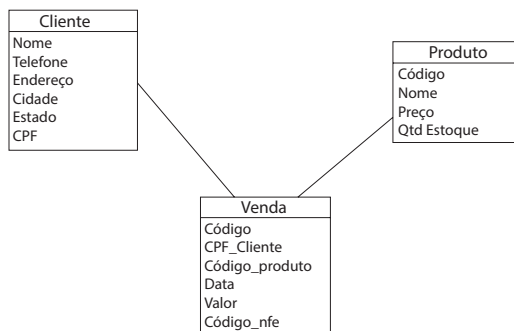


Figura 2. Exemplo de banco de dados.

Note que existem três tabelas ou *tables* no banco de dados da Figura 2: tabela “Cliente”, tabela “Produto” e tabela “Venda”. O relacionamento entre elas é executado quando o colaborador registra uma venda; então, o banco de dados cria um registro na tabela “venda” contendo os dados do produto comprado e os do cliente que comprou. Cada tabela tem atributos próprios, que seriam suas colunas, nas quais ficam as informações correspondentes. Ou seja, pela tabela “Cliente” foi mapeado que o cadastro de cliente terá os campos “Nome”, “Telefone”, “Endereço”, “Cidade”, “Estado” e “CPF”. Já na tabela “Produto”, os atributos (colunas) são “Código”, “Nome”, “Preço” e “Quantidade Estoque”. Por fim, temos a tabela “Venda” com os atributos (colunas) “Código”, “CPF_cliente”, “Código_produto”, “Data”, “Valor” e “Código_nfe”.

Para otimizar a quantidade de registros no banco de dados, existe a chave primária e a chave estrangeira. A chave primária é um campo que é definido como único e não pode repetir-se naquela tabela. Como exemplo de chave primária para a tabela “Cliente”, pode-se utilizar o CPF, que é um número único por pessoa (ninguém no Brasil tem o mesmo CPF que o seu). Para a tabela “Produto”, foi criado um atributo chamado “Código”. Já a tabela “Venda” pode usar o seu campo “Código”.

A chave estrangeira ocorre quando pegamos um atributo de outra tabela que é chave primária nela. Veja novamente a Figura 2 e observe que, na tabela “Venda”, existem duas possíveis chaves estrangeiras: “CPF_cliente”, que vem da tabela “Cliente”, e “Código_produto”, que vem da tabela “Produto”. Isso se dá para que, quando alguém pedir relatório de vendas, ao consultar a tabela “Venda”, o sistema consiga buscar, por meio do campo “CPF_cliente”, os dados do cliente na tabela “Cliente” e os dados do produto por meio do campo “Código_produto” na tabela “Produto”.

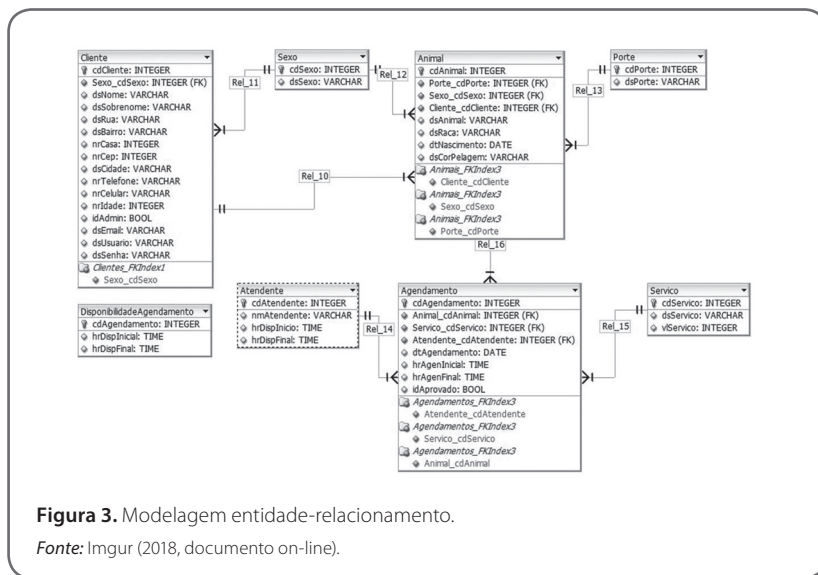
A dúvida que surge é a de por que realizar esses relacionamentos e não simplesmente criar mais atributos na tabela “Venda” para comportar os dados do cliente e dos produtos nessa própria tabela. A resposta é bem simples: em bancos de dados, assim como em qualquer sistema tecnológico, os dados têm complexidade e, quanto mais dados, maior a complexidade para processá-los. Ao registrar apenas o código do produto e o CPF do cliente e, com isso, realizar a busca nas tabelas correspondentes de “Cliente” e “Produto”, você está otimizando recursos computacionais e mantendo o banco de dados totalmente escalável.

Imagine que podem existir milhões de vendas diárias, de maneira que você pouparia muito processamento ao identificar as vendas apenas com os códigos das demais tabelas e, caso necessário, poderia recuperar a informação dessas tabelas em um relatório para o gerente da loja, por exemplo. Outra vantagem é

que, ao fazer uma atualização cadastral de um cliente (telefone, por exemplo), você teria que sair varrendo a tabela de “Venda” também, caso ela tivesse os campos replicados da tabela “Cliente”.

Geralmente, a modelagem de banco de dados ocorre em *times*, como forma de ser o mais acertado possível, visto que um retrabalho em cima da modelagem é muito mais oneroso do que se já estiver correto no início do projeto.

Quando um sistema se torna grande, essa modelagem apresentará centenas de tabelas e milhares de atributos e relacionamentos entre elas. A seguir, na Figura 3, observe o exemplo de um sistema de clínica veterinária com maior número de tabelas, atributos e relacionamentos:



Observe que, na Figura 3, aparecem novas informações após o nome do campo, como *varchar*, *integer*, *date*, *time*, *bool*, etc. Essas informações se referem ao tipo de dado que a coluna relacionada poderá receber. Então, seguindo esse raciocínio, a tabela “Animal”, no atributo “dtNascimento”, que é do tipo DATE, só poderá receber o valor de data válido. Qualquer coisa diferente disso será descartada pelo próprio banco de dados, que não permitirá a inserção. Outro exemplo: na tabela “Cliente”, o atributo “nrCEP” é do tipo INTEGER, de modo que só aceitará caracteres do tipo inteiro e, novamente, caso a entrada seja diferente de 0 até 9, o próprio banco de dados proíbe a entrada.

Pode-se concluir, então, uma relação de tipos de atributos mais comumente utilizados nos bancos de dados atualmente:

- INT ou INTEGER: recebe apenas números inteiros. Exemplo: 45002.
- VARCHAR: recebe caracteres alfanuméricos. Exemplo: João da Silva.
- DATE: recebe apenas datas válidas. Exemplo: 01/01/2018.
- TIME: recebe apenas valores expressos em hora. Exemplo: 16:54.
- BOOL ou BOOLEAN: recebe apenas o número 0 ou o número 1. Serve para indicar se algo é verdadeiro ou falso, positivo ou negativo, etc. Exemplo: 0 (o sistema sabe que, se esse campo for 0, o cliente está verdadeiro para compra a prazo).

Exemplos de sistemas de banco de dados e seus usos na prática

Com toda essa introdução sobre os sistemas de banco de dados, vamos abordar, agora, quais são os tipos de bancos de dados disponíveis e suas características e usos pelos diversos tipos de aplicações e corporações.

Nos dias atuais, quatro sistemas de banco de dados dominam o cenário geral e um quinto está em franco crescimento. São eles:

- MySQL;
- MariaDB;
- Microsoft SQLServer;
- PostgreSQL;
- Oracle;
- MongoDB (franca expansão).

Linguagem SQL

Para realizar a inserção, remoção, edição ou qualquer outra atividade com os dados que irão ou estão no banco de dados, é necessária uma interação com o mesmo. Dessa forma, a linguagem SQL (*Structure Query Language* ou Linguagem de Consulta Estruturada) é a que intermedeia essa interação, criando a ponte necessária entre o que precisa ser feito e o banco de dados efetivamente. Assim, para que algo aconteça, é necessário que a linguagem SQL esteja com a sintaxe correta e os valores desejados também corretos.

Aplicada para gerenciar os bancos de dados do tipo MariaDB (e MySQL), Microsoft SQLServer, PostgreSQL e Oracle, é a linguagem comumente usada por *database administrators* diariamente em suas jornadas de trabalho.

Perceba que o MongoDB, quinto item da lista anterior e marcado como em franca expansão, não utiliza a linguagem SQL para criar suas consultas e manipular os dados necessários. Isso se dá pela forma com que ele organiza suas informações e também como disponibiliza as mesmas de volta ao sistema quando este solicita uma consulta, por exemplo. Assim, ele consegue desempenhar uma velocidade incrível de resposta a consultas se comparado aos bancos de dados mais tradicionais que utilizam a linguagem de consulta estruturada (SQL). Portanto, o MongoDB é um banco de dados do tipo não relacional.

Segundo Elmasri e Navathe (2007), a linguagem SQL nasceu unicamente para ser a linguagem de consulta do sistema System R da IBM. A evolução desta empresa originou a linguagem SQL tão difundida atualmente.



Fique atento

A linguagem SQL tem uma sintaxe bem estruturada. Um exemplo básico seria:

- SELECT atributos (obrigatório);
- FROM tabelas (obrigatório);
- WHERE condições (opcional).

Assim, para retornar o nome e CPF de todos os clientes que moram no Paraná cadastrados na tabela “Cliente” da Figura 2, faríamos:

- SELECT Nome, CPF;
- FROM Cliente;
- WHERE Estado = PR.

Uma resposta para por que a linguagem SQL é tão utilizada e difundida em todo o mundo pode ser compreendida como:

A linguagem possui muitos recursos, tanto para manipulação de dados (DML, *Data Manipulation Language*) como para a definição de dados (DDL, *Data Definition Language*). Por conta disso, SQL é a linguagem mais utilizada em sistemas de bancos de dados (KOMATSU, 2011, p. 20).

Tipos de bancos de dados

A grande maioria dos projetos envolvendo bancos de dados toma por base a questão custo x benefício, ainda mais nos tempos atuais, em que as corporações necessitam de diminuição constante dos valores despendidos em qualquer área. Assim, de início, dividimos a lista dos bancos de dados em duas grandes frentes: os bancos de dados que precisam de uma licença paga e os bancos de dados que não necessitam de uma licença de uso paga.

Quadro 1. Relação de licenças x custo de administração

Banco de dados	Tipo de licença	Custo do administrador do database	Tamanho do projeto
MySQL	Paga	Baixo	Pequeno a médio
MariaDB	Gratuita	Baixo	Pequeno a médio
Microsoft SQLServer Standard ou Enterprise	Paga	Alto	Médio a grande
Microsoft SQLServer Express	Gratuita	Médio	Pequeno
PostgreSQL	Gratuita	Médio	Médio a grande
Oracle	Paga	Alto	Grande
MongoDB	Gratuita	Alto	Médio a grande

Com base nas informações apresentadas no Quadro 1, podemos refletir sobre a necessidade do projeto que precisa ser desenvolvido diante da realidade de custos sucintamente apresentados nessa mesma tabela informativa. Quanto maior o projeto, mais robusto precisa ser o banco de dados e, por consequência, mais cara será a cobrança por parte de quem o administra.



Referências

BRASIL. Agência Nacional de Telecomunicações. *Brasil encerrou o mês de março com 40,46 milhões linhas fixas em serviço*. 08 maio 2018. Disponível em: <www.anatel.gov.br/dados/destaque-1/331-brasil-tem-40-55-milhoes-linhas-fixas-em-operacao-no-mes-de-fevereiro>. Acesso em: 09 maio 2018.

ELMASRI, R.; NAVATHE, S. B. *Fundamentals of database systems*. 4. ed. London: Pearson Education, 2007.

IMGUR. *[Modelagem relacionamento]*. 2018. Disponível em: <<https://i.stack.imgur.com/Y8FiZ.png>>. Acesso em: 18 maio 2018.

KOMATSU, A. S. V. *Interpretador de linguagens de consulta relacionais*. 2011. 44 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade de São Paulo, São Paulo, 2011. Disponível em: <<https://bcc.ime.usp.br/tccs/2011/andre-suzuki/monografia.pdf>>. Acesso em: 09 maio 2018.

Leituras recomendadas

HEUSER, C. A. *Projeto de banco de dados*. 6. ed. Porto Alegre: Bookman, 2008. (Série Livros Didáticos Informática UFRGS, v. 4).

KORT, H. F.; SILBERSCHATZ, A.; SUDARSHAN, S. *Sistemas de bancos de dados*. 6. ed. Rio de Janeiro: Campus, 2012.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS

Dica do Professor

A história dos bancos de dados é marcada por inovações que transformaram profundamente a maneira como as informações são gerenciadas e utilizadas. Desde as primeiras formas de catalogação manual até os complexos sistemas computacionais atuais, o desenvolvimento de bancos de dados acompanhou o ritmo acelerado da evolução tecnológica, tornando-se indispensável para a eficiência operacional das empresas.

Ao longo dos anos, os bancos de dados se transformaram em mais do que simples repositórios de informações; evoluíram para ferramentas estratégicas capazes de fornecer *insights* valiosos e impulsionar decisões de negócios. À medida que as necessidades de armazenamento e análise de dados aumentaram, os sistemas de gerenciamento de bancos de dados (SGBD) avançaram, tornando-se mais adaptáveis, seguros e capazes de lidar com a complexidade e o volume crescente de dados.

Nesta Dica do Professor, você irá acompanhar uma viagem no tempo, explorando o desenvolvimento dos bancos de dados desde seus primórdios até os sistemas atuais que sustentam as operações críticas das organizações em todo o mundo. Você irá descobrir os marcos históricos que moldaram o desenvolvimento das tecnologias de banco de dados e como elas se adaptam continuamente para atender às exigências de uma área cada vez mais digital e interconectada.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Exercícios

- 1) Na escolha entre bancos de dados e planilhas eletrônicas para gerenciamento de informações, é fundamental entender a adequação de cada ferramenta às necessidades da organização.

O que justifica a preferência por bancos de dados em detrimento de planilhas eletrônicas para o armazenamento e a manipulação de dados corporativos?

- A) A utilização de planilhas eletrônicas é mais adequada para a análise de grandes volumes de dados em comparação aos bancos de dados.
- B) Bancos de dados oferecem funcionalidades avançadas de pesquisa e análise de dados, que não são viáveis em planilhas eletrônicas.
- C) Bancos de dados requerem habilidades avançadas de programação para a realização de consultas simples, diferentemente das planilhas.
- D) Planilhas eletrônicas são recomendadas para o armazenamento de dados altamente sensíveis devido à sua superioridade em segurança.
- E) A integração e a automação de processos são mais eficientes em planilhas eletrônicas do que em sistemas de banco de dados.

- 2) Na estruturação de um banco de dados de uma biblioteca, a chave primária de uma tabela de cadastro de livros é um identificador único que permite a busca precisa e a gestão eficaz de cada registro.

Qual atributo é ideal para ser utilizado como chave primária nesse contexto?

- A) Título do livro, pois cada livro tem um título distinto.
- B) Número da prateleira, uma vez que cada livro está localizado em uma única prateleira na biblioteca.
- C) ISBN, já que é um número internacionalmente reconhecido e único para cada edição de um livro.
- D) Nome do capítulo, já que cada livro tem capítulos com nomes únicos.

E) Cor da capa, considerando que os livros geralmente têm cores diferentes de capa.

3) No *design* de um banco de dados, compreender os tipos de atributos que podem ser utilizados nas tabelas é importante para garantir que os dados sejam armazenados e manipulados apropriadamente.

Qual das seguintes opções lista corretamente tipos de atributos de dados comumente utilizados em bancos de dados?

A) VARCHAR, INT, DATE – Tipos para texto, número inteiro e data, respectivamente.

B) BLOB, TIMESTAMP, BOOLEAN – Usados para dados binários grandes, carimbo de data/hora e valores verdadeiro/falso.

C) TEXT, REAL, DATETIME – Para texto longo, números reais e data e hora.

D) CHAR, DOUBLE, TIME – Para caracteres fixos, números com ponto flutuante e tempo.

E) NUMERIC, STRING, BIT – Para valores numéricos precisos, sequências de caracteres e valores binários.

4) Dominar a linguagem SQL é fundamental para efetuar consultas precisas em um banco de dados. Considere a estrutura básica de uma consulta SQL: "SELECT atributo FROM tabela WHERE condição".

Qual das cláusulas é opcional nessa estrutura?

A) As cláusulas SELECT, FROM e WHERE.

B) Somente a cláusula WHERE.

C) As cláusulas SELECT e FROM.

D) Somente a cláusula FROM.

E) As cláusulas FROM e WHERE.

5) Na implementação de um pequeno projeto com recursos financeiros limitados, a escolha do banco de dados é essencial para maximizar a eficiência operacional sem sobrecarregar o orçamento.

Considerando essas condições, qual dos seguintes bancos de dados seria o mais recomendado?

- A) Oracle.
- B) MongoDB.
- C) MySQL.
- D) PostgreSQL.
- E) MariaDB.

Na prática

Analisar e construir tabelas a partir de consultas SQL podem revelar a estrutura subjacente dos dados e a lógica das tabelas envolvidas. Esse processo inverso é essencial para compreender e manipular eficazmente bancos de dados complexos.

Para realizar esse processo, é necessário interpretar cuidadosamente cada componente da consulta SQL, como as cláusulas SELECT, FROM, WHERE e JOIN. A análise detalhada dessas cláusulas ajuda a identificar as relações entre diferentes tabelas e os critérios de seleção aplicados aos dados, permitindo uma visão clara da estrutura do banco de dados.

Neste Na Prática, acompanhe como um profissional aplica esse conhecimento para analisar e recriar eficientemente a estrutura de dados a partir de consultas SQL existentes, uma habilidade fundamental para profissionais de sistemas de informação.

ANÁLISE E RECONSTRUÇÃO DE ESTRUTURA DE DADOS A PARTIR DE CONSULTA SQL

Em um ambiente de TI hospitalar, a capacidade de modelar e entender a estrutura dos bancos de dados é fundamental para manter a eficiência dos processos internos. Profissionais da área frequentemente utilizam consultas SQL para analisar e reconstruir a estrutura de dados, garantindo a integridade e a acessibilidade das informações essenciais.

Confira, a seguir, um caso que envolve a modelagem da estrutura de um banco de dados.

PROJETO EM BANCO DE DADOS HOSPITALAR

Ronaldo, um novo funcionário de uma empresa de TI hospitalar, recebeu do seu supervisor a tarefa de realizar engenharia reversa de um banco de dados utilizando a seguinte consulta SQL.



Engenharia reversa é o processo de desmembrar um sistema para entender sua estrutura, seu funcionamento e seus componentes. No contexto de banco de dados, isso envolve analisar consultas SQL e estruturas existentes para reconstruir a lógica e os dados das tabelas.



Esse método é amplamente utilizado para manutenção, atualização e integração de sistemas complexos, permitindo que profissionais de TI compreendam e otimizem a infraestrutura existente sem acesso direto à documentação original.

PROPÓSITO DO PROJETO

O projeto de realizar engenharia reversa de um banco de dados é parte de uma iniciativa mais abrangente para atualizar e otimizar o sistema de gerenciamento de informações do hospital.

Como parte de uma equipe dedicada à manutenção e à melhoria contínua dos sistemas de TI,



Ronaldo é responsável por entender a estrutura existente dos dados para poder integrá-los com novas funcionalidades e garantir a eficiência operacional.

A consulta deveria resultar:

- > Na especialidade;
- > No nome;
- > Na hora do plantão; e
- > No CRM dos médicos com mais de 5 anos de experiência.



MODELAGEM EFICIENTE DA ESTRUTURA DE DADOS

A partir dessa consulta, Ronaldo identificou a tabela Medico_Cad e os atributos Especialidade, Nome, HoraPlantao, CRM e TempoExperiencia, permitindo a modelagem eficiente da estrutura dos dados.

A modelagem eficiente da estrutura de dados envolve organizá-los de forma que sua utilização, análise e manutenção sejam facilitadas, especialmente em um contexto hospitalar onde a precisão e a acessibilidade são essenciais.

Nesse contexto, a modelagem eficiente permitiu, e permite, uma gestão mais ágil e precisa dos recursos médicos, garantindo que os médicos mais experientes sejam alocados de forma estratégica para melhorar a qualidade do atendimento e a eficiência operacional do hospital.



Compreender a lógica das consultas SQL e aplicar engenharia reversa são habilidades importantes para profissionais de TI, sobretudo em ambientes complexos, como hospitais. Essa prática permite a manutenção eficiente e a otimização dos sistemas de banco de dados, garantindo a precisão e a integridade das operações diárias.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Saiba mais

Para ampliar o seu conhecimento a respeito desse assunto, veja a seguir as sugestões do professor.

Como instalar o MySQL no Windows

A investigação das funcionalidades do MySQL começa pela instalação correta do sistema de gerenciamento de bancos de dados. Neste guia, você irá encontrar instruções detalhadas para instalar o MySQL no Windows.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Criando meu primeiro modelo de dados

A estruturação de um banco de dados inicia com um planejamento eficaz, e criar um diagrama de modelo lógico é uma etapa crucial. Neste vídeo, você vai conferir o passo a passo para criar um Diagrama de Modelo Lógico, fornecendo uma base sólida para o desenvolvimento de suas estruturas de banco de dados.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

SQL INSERT(INTO e SELECT): inserir linhas em tabelas

Para dominar a inserção de dados em tabelas SQL, é necessário entender os comandos INSERT INTO e SELECT. Neste artigo, você irá conferir um guia detalhado para aprimorar suas habilidades em SQL, abrangendo desde a sintaxe básica até exemplos práticos para inserir e selecionar dados eficientemente em um banco de dados.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.