



CENTRO DE EDUCAÇÃO PROFISSIONAL TALAL ABU-ALLAN
ADMINISTRADOR DE BANCO DE DADOS

**ANA CRISTINA OLIVEIRA DA SILVA, AUDEANE DE LUCENA, LUIZ FELIPE
PEREIRA**

**IMPLEMENTAÇÃO DE BANCO DE DADOS PARA LOJA DE AUTOPEÇAS NO SQL
SERVER**

BRASÍLIA-DF
2024

**ANA CRISTINA OLIVEIRA DA SILVA, AUDEANE DE LUCENA, LUIZ
FELIPE PEREIRA**

**IMPLEMENTAÇÃO DE BANCO DE DADOS PARA LOJA DE AUTOPEÇAS NO
SQL SERVER**

Projeto integrador para Conclusão do curso,
administrador de banco de dados visando,
promover a interdisciplinaridade, ao estabelecer
a integração dos conhecimentos desenvolvidos
durante todo o curso.

Orientador (a): Prof. Hudson Neves

**Brasília-DF
2024**

SUMÁRIO

INTRODUÇÃO	4
CONCEITO DE BANCO DE DADOS E SUA IMPORTÂNCIA	5
SISTEMAS DE BANCO DE DADOS	6
SGBD: SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS.....	8
MODELOS DE BANCO DE DADOS.....	9
VANTAGENS MER: MODELO ENTIDADE RELACIONAMENTO.....	10
VANTAGENS MER: MODELO ENTIDADE RELACIONAMENTO	12
ARQUITETURAS DE BANCO DE DADOS	13
PADRÕES SQL.....	15
ESTUDO DE CASO	17
AMBIENTE DE DESENVOLVIMENTO USADO E FERRAMENTAS	18
CONSIDERAÇÕES FINAIS	22

INTRODUÇÃO

Bancos de dados se tornaram componentes essenciais no cotidiano da sociedade moderna. No decorrer do dia a dia, nos deparamos com atividades que envolvem alguma interação com banco de dados. Nas transações bancárias, nas reservas de passagens aéreas, nas compras de supermercados, no acesso ao catálogo de uma biblioteca informatizada, aplicações de web e em dispositivos móveis um banco de dados será acessado. O uso de banco de dados facilita o dia a dia de quem está envolvido com esse tipo de transação, pois torna-as mais rápidas, eficientes e seguras, e claro, importando em menos trabalho braçal. Essas interações são exemplos de acesso a aplicações tradicionais de banco de dados, nas quais a informação armazenada é essencialmente numérica ou textual.

Nos últimos anos a proliferação das redes sociais requereram a criação de bancos de dados enormes para armazenar informações não tradicionais, como imagens e vídeos. Novos tipos de sistemas de bancos de dados foram criados para lidar com essa demanda, os sistemas de armazenamento de big data ou sistemas NoSQL.

CONCEITO DE BANCO DE DADOS E SUA IMPORTÂNCIA

Conceitos básicos: Um banco de dados (BD) é uma coleção de dados relacionados. Por dados, entende-se fatos conhecidos que podem ser armazenados e têm um significado implícito. Por exemplo, considere os nomes, telefones e endereços das pessoas que você conhece. Atualmente, esses dados estão armazenados nos celulares, que têm o seu próprio software para gerenciar esses dados.

Um banco de dados tem as seguintes propriedades implícitas:

- Representa algum aspecto do mundo real;
- É uma coleção de dados logicamente coerente;
- É projetado, construído e populado para um propósito específico.

Bancos de dados podem ser de qualquer tamanho ou complexidade. Por exemplo, a lista de contatos mencionada anteriormente pode conter algumas centenas de registros, cada qual com uma estrutura simples. Em contrapartida, um exemplo de banco de dados grande e complexo é o da Amazon.com, que contém dados de 60 milhões de usuários ativos, milhões de livros, jogos, aparelhos eletrônicos e outros itens. Esse banco de dados ocupa mais de 42 terabytes e é armazenado em centenas de computadores.

Um sistema de gerenciamento de banco de dados (SGBD) é um sistema computadorizado que permite ao usuário criar e manter um banco de dados. Um SGBD é um software de propósito geral que facilita o processo de definir, construir, manipular e compartilhar bancos de dados entre vários usuários e aplicações. A definição de um banco de dados envolve especificar os tipos de dados, estruturas e restrições nos dados a serem armazenados. A definição dos dados e sua informação descritiva é também armazenada no SGBD na forma de um catálogo do banco de dados, chamado metadados.

A construção do banco de dados é o processo de armazenar os dados em algum meio controlado pelo SGBD. A manipulação do banco de dados inclui funções como consultar, atualizar e gerar relatórios sobre o banco de dados. O compartilhamento de um banco de dados permite que múltiplos usuários e programas acessem o banco de dados simultaneamente. Um programa de aplicação acessa o banco de dados enviando consultas ou requisições de dados ao SGBD. Uma consulta tipicamente seleciona alguns dados para serem recuperados; uma transação pode levar a alterações nos dados de um banco de dados. Para completar as definições iniciais, o banco de dados juntamente com seu sistema gerenciador é chamado de sistema de banco de dados.

Importância: sua importância na era moderna é indiscutível, por serem essenciais em praticamente todos os aspectos do nosso cotidiano. Por sua capacidade de gerenciar e armazenar grandes volumes de dados e permitir acesso rápido e confiável a essas informações.

Além de garantir uma alta eficiência e a produtividade em diversas áreas, também garantem a integridade, a consistência e a segurança das informações nele contidas.

SISTEMAS DE BANCO DE DADOS

Um sistema de banco de dados é composto por um banco de dados e um software gerenciador de banco de dados. Esse sistema está dividido em módulos específicos, de modo que todas as necessidades desse sistema possam ser atendidas. Algumas das funções essenciais do sistema de banco de dados podem ser oferecidas pelo sistema operacional e o banco de dados é então construído nessa base. Deve-se, portanto, ao construir o banco de dados, considerar a interface entre o sistema de banco de dados e o sistema operacional. Os componentes funcionais de um sistema de banco de dados podem ser divididos em componentes de processamento de consultas e componentes de administração de memória. Segue abaixo a descrição de cada componente.

Componentes de processamento de consultas: Compilador DML Traduz os comandos DML (Data Manipulation Language) da linguagem de consulta em instruções de baixo nível, inteligíveis ao componente de execução de consultas. Além disso, o compilador DML tenta transformar a solicitação do usuário em uma solicitação equivalente, mas mais eficiente, buscando assim uma boa estratégia para a execução da consulta.

Pré-compilador para comandos DML onde são inseridos em programas de aplicação, que convertem comandos DML em chamadas de procedimentos normais da linguagem hospedeira. O pré-compilador precisa interagir com o compilador DML de modo a gerar o código apropriado.

Interpretador DDL Interpreta os comandos DDL (Data Definition Language) e registra-os em um conjunto de tabelas que contém metadados (dados contendo informações sobre outros dados).

Componentes para o tratamento de consultas executam instruções de baixo nível geradas pelo compilador DML.

Componentes de autorização e integridade: Gerenciamento de autorização e integridade testam o cumprimento das regras de integridade e a permissão ao usuário no acesso aos dados.

Gerenciamento de transações: Garante que o banco de dados permanecerá em estado consistente (correto) a despeito de falhas no sistema e que transações concorrentes serão executadas sem conflito em seus procedimentos.

Administração de arquivos: Gerencia a alocação do espaço no armazenamento em disco e as estruturas de dados usadas para representar estas informações armazenadas.

Administração de buffer: Responsável pela intermediação de dados no disco para a memória principal e pela decisão de quais dados colocar em memória cache. A vantagem principal na utilização de memória cache consiste em evitar o acesso ao dispositivo de armazenamento, que pode ser demorado, armazenando os dados em meios de acesso mais rápidos. Além dos componentes de processamento de consulta e

de administração de memória, algumas estruturas de dados são exigidas como parte de implementação física do sistema.

Essas estruturas são:

- Arquivo de dados: Armazena o próprio banco de dados.
- Dicionário de dados: Armazena os metadados relativos à estrutura do banco de dados. O dicionário de dados é muito usado. Portanto grande ênfase é dada ao desenvolvimento de um bom projeto com uma implementação eficiente no dicionário.
- Índices: Proporcionam acesso rápido aos itens de dados que são associados a valores determinados.
- Estatísticas de dados: Armazena informações estatísticas relativas aos dados contidos no banco de dados. Essas informações são usadas pelo processador de consultas para seleção de meios eficientes para execução de uma consulta. Para que tudo que foi dito acima seja possível, ou seja, para criar e gerenciar um banco de dados torna-se necessário um Software Gerenciador de Banco de Dados.

SGBD: SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS

Um SGBD é uma coleção de programas que permite ao usuário criar e manter um banco de dados. Ele facilita os processos de definição, onde ocorre a especificação do tipo de dados e das restrições dos dados que serão armazenados, de construção, etapa de armazenamento dos dados em alguma mídia controlada pelo SGBD, de manipulação do banco de dados, onde pudesse incluir várias funções como por exemplo a pesquisa para recuperar um dado específico, a atualização do banco de dados, a geração de relatórios de dados, e outras atividades de acordo com a necessidade do usuário, e finalmente no processo de compartilhamento dos dados entre usuários, permitindo acesso de forma concorrente.

Um SGBD funciona como uma interface entre o banco de dados e os usuários. Com isso, é possível armazenar, recuperar, atualizar e gerenciar o modo como as informações são organizadas e otimizadas.

Os SGBD oferecem uma variedade de funcionalidades e recursos, incluindo:

1. Modelagem de dados: Permitindo definir a estrutura dos dados e os relacionamentos entre eles usando diferentes modelos de dados, como o modelo relacional, o modelo hierárquico e o modelo de rede.
2. Gerenciamento de transações: Garantindo a atomicidade, consistência, isolamento e durabilidade (ACID) das transações, para garantir que as operações de banco de dados sejam executadas de forma confiável, mesmo em caso de falha do sistema.
3. Controle de concorrência: Gerenciando o acesso simultâneo de múltiplos usuários ao banco de dados para garantir que as transações sejam executadas de forma consistente e sem conflitos.
4. Segurança: Protegendo os dados contra acesso não autorizado por meio de autenticação, autorização e criptografia, garantindo que apenas usuários autorizados possam acessar e manipular os dados.
5. Otimização de consultas: Analisando consultas SQL e identificando a melhor forma de executá-las para garantir um desempenho otimizado do sistema.
6. Backup e recuperação: Realizando backups regulares dos dados e fornecendo mecanismos de recuperação para restaurar o banco de dados em caso de falha ou corrupção dos dados.
7. Replicação e sincronização de dados: Permitindo replicar dados entre diferentes servidores para garantir alta disponibilidade e tolerância a falhas.

MODELOS DE BANCO DE DADOS

Um modelo de banco de dados mostra a estrutura lógica de um banco de dados, incluindo as relações e restrições que determinam como os dados podem ser armazenados e acessados. Modelos de banco de dados individuais são projetados com base nas regras e nos conceitos do modelo de dados que os designers adotam. A maioria dos modelos de dados pode ser representada por um diagrama de banco de dados acompanhante.

Graças à tecnologia de banco de dados, empresas e organizações conseguem manter registros de transações, gerenciar grandes volumes de informações de clientes, e realizar análises de dados para tomadas de decisão estratégicas. Sem esses sistemas sofisticados, o manuseio da imensa quantidade de dados gerados diariamente seria, sem dúvida, uma tarefa gigantesca.

Adicionalmente, os bancos de dados não se limitam apenas ao armazenamento de dados. Eles são também essenciais para garantir a integridade, segurança e recuperação dos dados em casos de falhas ou perdas inesperadas. São, portanto, o suporte essencial para aplicações que exigem uma administração e manipulação de dados confiável e eficiente.

Há vários tipos de modelos de dados.

Alguns dos mais comuns são:

Modelo relacional

o modelo relacional, classifica dados em tabelas, também conhecidas como relações, cada uma das quais consiste em colunas e linhas. Cada coluna lista um atributo da entidade em questão, como preço, código postal ou data de nascimento. Juntos, os atributos em uma relação são chamados de domínio. Um determinado atributo ou combinação de atributos é escolhido como uma chave primária que pode ser consultada em outras tabelas, quando é chamada de chave estrangeira.

Cada linha, também chamada de tupla, inclui dados sobre uma instância específica da entidade em questão, como um determinado colaborador.

O modelo também explica os tipos de relações entre essas tabelas, incluindo relações uma para uma, uma para muitas e muitas para muitas. Dentro do banco de dados, as tabelas podem ser normalizadas ou levadas a cumprir as regras de normalização que tornam o banco de dados flexível, adaptável e redimensionável. Quando normalizado, cada dado é atômico, ou dividido em pequenos pedaços úteis.

Os bancos de dados relacionais são tipicamente escritos em SQL (Structured Query Language). O modelo foi introduzido por E. F. Codd em 1970.

Modelo hierárquico

O modelo hierárquico organiza dados em uma estrutura do tipo árvore, onde cada registro tem um único "pai" ou raiz. Registros "irmãos" são classificados em uma ordem específica. Essa ordem é usada como a ordem física para armazenar o banco de dados. Este modelo é bom para descrever muitas relações do mundo real.

Modelo de rede

O modelo de rede se baseia no modelo hierárquico, permitindo relações muitas para muitas entre registros vinculados, implicando em vários registros "pai". Baseado na teoria de conjuntos matemáticos, o modelo é construído com conjuntos de registros relacionados. Cada conjunto consiste em um registro proprietário, ou "pai", e um ou mais registros de membro, ou "filho". Um registro pode ser um membro, ou "filho", em vários conjuntos, permitindo que esse modelo transmita relações complexas. Foi mais popular nos anos 70, depois de ter sido formalmente definido pela Conferência sobre Linguagens de Sistemas de Dados (CODASYL).

Modelo de banco de dados orientado para objetos

Este modelo define o banco de dados como uma coleção de objetos, ou elementos de software reutilizáveis, com recursos e métodos associados. Há vários tipos de bancos de dados orientados para objetos:

Um banco de dados multimídia

incorpora mídia, como imagens, que não podem ser armazenadas em um banco de dados relacional.

Um banco de dados de hipertexto

permite que qualquer objeto seja vinculado a qualquer outro objeto. É útil para organizar lotes de dados diferentes, mas não é ideal para a análise numérica.

O modelo de banco de dados orientado a objetos é o modelo de banco de dados pós-relacional mais conhecido, uma vez que ele incorpora tabelas, mas não se limita a elas. Tais modelos também são conhecidos como modelos de bancos de dados híbridos.

Modelo relacional-objetos

Este modelo de banco de dados híbrido combina a simplicidade do modelo relacional com algumas das funcionalidades avançadas do modelo de banco de dados orientado a objetos. Em essência, ele permite que os designers incorporem objetos na estrutura da tabela familiar.

As interfaces de linguagem e chamadas incluem SQL3, linguagens de fornecedor, ODBC, JDBC e interfaces de chamada proprietárias que são extensões das linguagens e interfaces usadas pelo modelo relacional.

Modelo entidade-relacionamento

Este modelo capta as relações entre entidades do mundo real, de forma parecida com o modelo de rede, mas não está diretamente ligado à estrutura física do banco de dados. Em vez disso, ele é frequentemente usado para projetar um banco de dados conceitualmente.

Aqui, pessoas, lugares e coisas sobre quais pontos de dados são armazenados e referidos como entidades, cada uma das quais possui certos atributos que, em conjunto, compõem seu domínio. A cardinalidade, ou relações entre entidades, também são mapeadas.

Uma forma comum do diagrama ER é o esquema em estrela, no qual uma tabela de fatos central se conecta a tabelas multidimensionais.

VANTAGENS MER: MODELO ENTIDADE RELACIONAMENTO

Ao utilizar o MER Banco de Dados, é possível organizar os dados de maneira clara e precisa, o que permite que você possa encontrar e utilizar as informações de maneira eficiente. O MER permite que você crie um modelo conceitual do seu banco de dados, que representa as entidades envolvidas, seus relacionamentos e atributos.

Com esse modelo conceitual, você pode visualizar e entender a estrutura do seu banco de dados de forma mais clara e organizada. Isso ajuda a evitar erros de organização de dados, o que pode prejudicar o desempenho do sistema como um todo.

Outra vantagem do uso do MER Banco de Dados é que ele facilita a comunicação entre os membros da equipe que trabalham no projeto de banco de dados. Com um modelo conceitual claro e preciso, fica mais fácil para todos os envolvidos entenderem a estrutura do banco de dados e colaborarem na criação e manutenção do sistema.

Além disso, o MER Banco de Dados permite que você faça alterações no modelo conceitual com facilidade. Por exemplo, se você precisar adicionar uma nova entidade ao seu banco de dados, pode fazê-lo facilmente sem afetar todo o sistema. Isso ajuda a manter a flexibilidade e escalabilidade do banco de dados ao longo do tempo.

Ao organizar seus dados com o MER Banco de Dados, você também pode melhorar a segurança do sistema. Isso ocorre porque a organização eficiente dos dados permite que você controle melhor o acesso aos diferentes tipos de informações armazenadas no banco de dados, garantindo que apenas pessoas autorizadas possam acessá-las.

O uso do MER Banco de Dados é uma maneira eficiente de organizar seus dados. Ele ajuda a garantir que o sistema de banco de dados seja organizado, escalável, flexível e seguro, o que é essencial para o sucesso de qualquer empresa ou organização.

ARQUITETURAS DE BANCO DE DADOS

A arquitetura de dados é um conjunto de práticas e tecnologias que permitem que as empresas gerenciem seus dados de forma eficiente e segura. Ela envolve a organização, armazenamento, processamento e análise de dados, com o objetivo de fornecer informações valiosas para a tomada de decisões.

Uma arquitetura de dados é composta por vários componentes, incluindo hardware, software, redes e serviços em nuvem. Esses componentes trabalham juntos para garantir que os dados sejam coletados, armazenados e processados de forma eficiente e segura.

Ela permite que as empresas colem e analisem dados de várias fontes, incluindo clientes, fornecedores e parceiros de negócios. Com esses dados, as empresas podem tomar decisões informadas sobre estratégias de negócios, marketing, vendas e muito mais.

Principais aspectos de como a arquitetura de dados funciona:

Coleta de dados

Diversas fontes: grandes empresas geralmente coletam dados de uma variedade de fontes, como transações comerciais, aplicativos, sensores, mídias sociais entre outras.

Integração de dados: os dados de várias fontes precisam ser integrados e consolidados para formar uma visão única e consistente das informações.

Armazenamento de dados

Data Warehouses: As empresas utilizam data warehouses para armazenar grandes volumes de dados estruturados. Isso permite análises de dados históricos e relatórios detalhados.

Data Lakes: Para dados não estruturados e semiestruturados, os data lakes oferecem flexibilidade e escalabilidade, permitindo o armazenamento de grandes quantidades de dados brutos.

Processamento de dados

ETL (Extract, Transform, Load): processos ETL são usados para extrair dados de diversas fontes, transformá-los em um formato útil e carregá-los nos sistemas de armazenamento.

Processamento em tempo real: em muitos casos, o processamento em tempo real é essencial para tomar decisões rápidas. Isso pode envolver o uso de tecnologias como o streaming de dados e o processamento de eventos complexos.

Acesso e visualização

Painéis e relatórios: a arquitetura de dados suporta a criação de painéis interativos e relatórios personalizados que ajudam os tomadores de decisão a acessar informações relevantes de maneira rápida e eficiente.

APIs e integrações: os dados também são disponibilizados por meio de APIs para que outras aplicações possam acessá-los e incorporá-los.

Segurança e privacidade

Políticas de acesso: existem rigorosas políticas de controle de acesso para garantir que apenas pessoal autorizado possa acessar dados sensíveis.

Criptografia e auditoria: a criptografia e a auditoria são usadas para proteger dados confidenciais e rastrear atividades de acesso.

Governança de Dados

Metadados: a gestão de metadados ajuda a documentar e controlar os dados, incluindo definições, origens e propriedades.

Padronização: a padronização de nomenclaturas e classificações de dados é essencial para garantir a consistência e a compreensão dos dados em toda a empresa.

Evolução Contínua

Escalabilidade: a arquitetura de dados precisa ser escalável para atender às crescentes demandas de dados em uma grande empresa.

Adaptação: à medida que as necessidades de negócios evoluem, a arquitetura de dados deve ser adaptada para incorporar novas tecnologias e metodologias.

Conformidade regulatória

As organizações precisam cumprir regulamentações de privacidade de dados, como o RGPD na União Europeia ou a Lei de Proteção de Dados no Brasil (LGPD), e a arquitetura de dados desempenha um papel crucial na conformidade.

PADRÕES SQL

O SQL é uma linguagem padrão para trabalhar com bancos de dados relacionais, amplamente utilizada por profissionais em diversas áreas, desde cientistas de dados até pessoas usuárias de Excel.

SQL é a sigla para Structured Query Language, que em português significa Linguagem de consulta estruturada.

A linguagem SQL é relativamente semelhante entre os principais Sistemas Gerenciadores de Banco de Dados (SGBDs) do mercado, como: Oracle, MySQL, MariaDB, PostgreSQL, Microsoft SQL Server, entre outros.

No entanto, é muito importante destacar que cada um deles têm suas características e particularidades distintas. Por exemplo, o MySQL e o PostgreSQL são extremamente notáveis por oferecerem versões gratuitas e de código aberto, tornando-se extremamente populares por essa razão.

Um grande destaque do SQL é ser uma linguagem fundamental em várias profissões, especialmente para pessoas que querem se sobressair no mercado.

Com esses dados, as empresas podem tomar decisões informadas sobre estratégias de negócios, marketing, vendas e muito mais.

Linguagem de consulta: a Query Language permite ao usuário interagir diretamente com o software de banco de dados, a fim de executar as tarefas de processamento de informações, usando dados em um banco de dados. É normalmente uma linguagem de computador fácil de usar, que se baseia em palavras básicas, tais como SELECT, DELETE ou ALTER. Usando linguagem de consulta e um teclado de computador, o usuário digita comandos que instruem o SGBD para recuperar dados de uma base de dados ou a atualização de um banco de dados.

Linguagem estruturada: Structured Query Language é um tipo de linguagem de consulta que é amplamente utilizada para executar operações usando bancos de dados relacionais. 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 consulta SQL contém três elementos chaves:

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

O padrão SQL também possui outras capacidades, umas das quais é a de ser capaz de atualizar e revisar um banco de dados relacional. Os usuários podem descobrir a necessidade de adicionar ou excluir colunas em um banco de dados.

Funções do SQL:

- Executar consultas em um banco de dados;
- Selecionar e recuperar dados a partir de um banco de dados;
- Inserir registros em um banco de dados;
- Atualizar registros em um banco de dados;
- Criar bancos de dados;
- Criar tabelas;
- Criar procedimentos armazenados (stored procedures);

- Criar visões (views);
- Definir permissões em tabelas, procedures e views.

A linguagem do SQL pode ser dividida em três partes:

- DDL – linguagem de definição de dados (Data Definition language) que 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;
- DML – Linguagem de manipulação de dados (Data Manipulation language) permite a inserção de dados e exclusão dos dados das tabelas;
- DCL- Linguagem de controle de dados (Data Control Language) atribui permissões aos objetos através dos comandos Grant, revok e Deny. A linguagem de controle de dados é subconjunto do Lanaguge SQL (Strucrured Query) que permite que os administradores de banco de dados para configurar o acesso de segurança para bancos de dados relacionais.

ESTUDO DE CASO

Implementação de Banco de Dados Para Loja de Autopeças no SQL Server

A empresa Aalu, focada na otimização e agilidade na pesquisa de dados, solicitou o desenvolvimento de um sistema de banco de dados para o controle de estoque de peça, que será disponibilizado para os funcionários da empresa, o que, até então, era realizado manualmente.

Esse sistema terá como função principal a de controlar a entrada e saída de peças do estoque, além de atender aos seguintes requisitos:

- A empresa possuir muitas peças a serem controladas;
- O sistema deverá solicitar o saldo inicial da peça uma única vez a cada vez que o sistema é iniciado;

O sistema deverá ter três tipos de entrada de dados:

- Compra (entrada) de peças;
- Venda (saída) de peças;
- Atualização de cadastro;

Para criação desse sistema de cadastro, você, enquanto programador(a) responsável, deverá solucionar os seguintes desafios:

- Descrever a sequência de passos lógicos necessários para criação do sistema solicitado pela empresa.
- Aplicar no sistema SQLServer.

AMBIENTE DE DESENVOLVIMENTO USADO E FERRAMENTAS

Inicialmente o sistema seria desenvolvido para o MySQL, porém devido a dificuldades encontradas em relação à infraestrutura dos equipamentos, optou-se por modificar o código para adaptá-lo ao SQL Server. Após vários testes o código abaixo mostrou-se favorável ao solicitado:

/*Criação do código para implementação de BD de uma loja de autopeças*/

```
CREATE TABLE Contato (  
    IDcontato INT PRIMARY KEY,  
    Telefone CHAR(15),  
    Endereco VARCHAR(100) NOT NULL,  
    CEP CHAR(9),  
    Email VARCHAR(60) NOT NULL  
);  
  
CREATE TABLE Veiculo (  
    IDVeiculo INT PRIMARY KEY,  
    Marca_veic CHAR(50) NOT NULL,  
    Modelo_veic CHAR(22) NOT NULL,  
    Placa_veic VARCHAR(15) NOT NULL  
);  
  
CREATE TABLE Marca (  
    IDmarca INT PRIMARY KEY,  
    Descricao VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE Cliente (  
    IDCliente INT PRIMARY KEY,  
    Nome VARCHAR(100) NOT NULL,  
    Data_Nasc DATE NOT NULL,  
    Contato_idcontato INT,  
    FOREIGN KEY (Contato_idcontato) REFERENCES Contato(IDcontato)  
);
```

```

CREATE TABLE Pessoa_Fisica (
    IDPessoa INT PRIMARY KEY,
    Nome VARCHAR(100) NOT NULL,
    RG VARCHAR(20) NOT NULL,
    CPF CHAR(14) NOT NULL,
    Data_Nasc DATE NOT NULL,
    Contato_idcontato INT,
    FOREIGN KEY (Contato_idcontato) REFERENCES Contato(IDcontato)
);

```

```

CREATE TABLE Pessoa_Juridica (
    IDPessoa INT PRIMARY KEY,
    Nome_fantasia VARCHAR(50) NOT NULL,
    Razao_social VARCHAR(50) NOT NULL,
    CNPJ CHAR(18) NOT NULL,
    IE VARCHAR(20),
    Contato_idcontato INT,
    FOREIGN KEY (Contato_idcontato) REFERENCES Contato(IDcontato)
);

```

```

CREATE TABLE Fornecedor (
    IDFornecedor INT PRIMARY KEY,
    Nome_fantasia VARCHAR(50) NOT NULL,
    Razao_social VARCHAR(50) NOT NULL,
    CNPJ CHAR(18) NOT NULL,
    IE VARCHAR(20),
    Contato_idcontato INT,
    FOREIGN KEY (Contato_idcontato) REFERENCES Contato(IDcontato)
);

```

```

CREATE TABLE Produto (
    IDProduto INT PRIMARY KEY,
    Nome VARCHAR(50) NOT NULL,
    Preco DECIMAL(10,2) NOT NULL,

```

```

        Descricao VARCHAR(255) NOT NULL,
        Qtde_estoque INT NOT NULL,
        Qtde_min INT NOT NULL,
        Modelo_idmod INT,
        FOREIGN KEY (Modelo_idmod) REFERENCES Modelo(IDmod)
    );

CREATE TABLE Modelo (
    IDmod INT PRIMARY KEY,
    Descricao VARCHAR(100) NOT NULL,
    Marca_idmarca INT,
    FOREIGN KEY (Marca_idmarca) REFERENCES Marca(IDmarca)
);

CREATE TABLE Venda (
    IDvenda INT PRIMARY KEY,
    DTVenda DATE,
    Cliente_idcliente INT,
    FOREIGN KEY (Cliente_idcliente) REFERENCES Cliente(IDCliente)
);

CREATE TABLE Itens_venda (
    IDvenda INT,
    IDproduto INT,
    Preco DECIMAL(10,2) NOT NULL,
    Qtde INT NOT NULL,
    PRIMARY KEY (IDvenda, IDproduto),
    FOREIGN KEY (IDvenda) REFERENCES Venda(IDvenda),
    FOREIGN KEY (IDproduto) REFERENCES Produto(IDProduto)
);

ALTER TABLE Veiculo
ADD Cliente_idcliente INT NOT NULL;

```

```
SP_HELP Veiculo;
```

```
ALTER TABLE Veiculo
```

```
ADD CONSTRAINT fk_cliente
```

```
FOREIGN KEY (Cliente_idcliente)
```

```
REFERENCES Cliente(IDCliente);
```

CONSIDERAÇÕES FINAIS

O desenvolvimento desse trabalho possibilitou o aprendizado amplo sobre alguns tipos de sistema de gerenciamento de banco de dados. Passamos inicialmente pelo MySQL, após isso aprendemos o SQL Server, em sequência o PostgreSql e concluímos com MongoDB. A partir desse leque tivemos maturidade para de fato escolher o que melhor se adapta ao nosso gosto, mesmo sabendo que no mercado de trabalho, atuaremos com o que estiver disponível nas empresas, o que nos incentiva a sempre buscarmos atualização de nossos conhecimentos e habilidades.

Cada um rumará para aquela área que melhor apetece a seus sentidos ou não. Mas o que não se pode negar é a capacidade do instrutor em nos mostrar os caminhos que devemos tomar para alcançar nossos objetivos profissionais e conseqüentemente o sucesso que almejamos.

Agradecemos ao Professor Hudson Neves por nos proporcionar a oportunidade de ampliar nosso interesse na área de banco de dados, por não desistir de nós e por ser mais que um instrutor, mas sim um mestre que nos guia e nos aconselha.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 Tutorial_DB_2010_12_16.pdf
https://www.telecom.uff.br/pet/petws/downloads/tutoriais/db/Tutorial_DB_2010_12_16.pdf.
- 2 2020-2_apostila_cdd003.pdf
http://professor.ufop.br/sites/default/files/george/files/2020-2_apostila_cdd003.pdf.
- 3 blog.vibetecnologia.com
<https://www.blog.vibetecnologia.com/banco-de-dados#:~:text=Trata%2Dse%20de%20uma%20tecnologia,onde%20as%20informa%C3%A7%C3%B5es%20ficam%20armazenadas>.
- 4 4infra.com.br
<https://4infra.com.br/vantagens-do-sqbd-sistema-de-gerenciamento-de-banco-de-dados#:~:text=Como%20funciona%20o%20SGBD%3F,informa%C3%A7%C3%B5es%20s%C3%A3o%20organizadas%20e%20otimizadas>.
- 5 lucidchart.com
<https://www.lucidchart.com/pages/pt/o-que-e-um-modelo-de-banco-de-dados>.
- 6 awari.com.br
<https://awari.com.br/mer-banco-de-dados#:~:text=O%20MER%20permite%20que%20voc%C3%AA,forma%20mais%20clara%20e%20organizada>.
- 7 objective.com.br
<https://www.objective.com.br/insights/arquitetura-de-dados/>.
- 8 ehgomes.com.br
<https://ehgomes.com.br/banco-de-dados/sql-o-que-e-guia-explicativo-e-conciso#:~:text=O%20SQL%20%C3%A9%20uma%20linguagem,software%20de%20banco%20de%20dados>.