**UNIVERSIDADE SENAI JOINVILLE**

**Felipe Rafael Rothbarth**

**OT14 - Introdução a banco de dados**

**JOINVILLE**

**FICHAMENTOS**

Definição:

Um banco de dados é uma coleção organizada de dados que podem ser facilmente acessados, gerenciados e atualizados. Ele armazena informações de forma estruturada para permitir a recuperação eficiente, manipulação e análise dos dados. Bancos de dados são essenciais em sistemas de informação para armazenar dados de maneira persistente.

Principais Modelos/Tipos de Banco de Dados

**Banco de Dados Relacional (RDBMS)**

Definição: Organiza os dados em tabelas (linhas e colunas) e usa SQL (Structured Query Language) para manipulação e consulta dos dados.

Vantagens:

Integridade e consistência: Através de chaves primárias e estrangeiras, além de restrições de integridade.

Escalabilidade vertical: Capacidade de aumentar os recursos do hardware para melhorar o desempenho.

Suporte robusto e maturidade: Vasto suporte de ferramentas e comunidades ativas.

Desvantagens:

Escalabilidade horizontal limitada: Dificuldade em distribuir dados em vários servidores.

Rigidez no esquema: Alterações no esquema (estrutura das tabelas) podem ser complexas e demoradas.

Exemplos: MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server.

**Banco de Dados NoSQL**

Definição: Projetado para lidar com grandes volumes de dados não estruturados ou semi-estruturados, escalando horizontalmente. Não utiliza o modelo de tabelas, e sim documentos, grafos, colunas ou pares chave-valor.

Vantagens:

Escalabilidade horizontal: Facilmente distribuído em vários servidores.

Flexibilidade de esquema: Modelos de dados flexíveis que permitem alterações rápidas.

Desempenho: Alta performance em operações de leitura e escrita em grandes volumes de dados.

Desvantagens:

Menor consistência: Pode comprometer a consistência dos dados em favor da disponibilidade e partição.

Falta de padronização: Diferentes sistemas NoSQL têm diferentes modelos e query languages.

Exemplos: MongoDB (documentos), Cassandra (colunas), Redis (chave-valor), Neo4j (grafo).

**Banco de Dados Orientado a Objetos (OODBMS)**

Definição: Armazena dados na forma de objetos, como na programação orientada a objetos. Integra perfeitamente com linguagens de programação orientadas a objetos.

Vantagens:

Modelo natural: Integração direta com linguagens de programação orientadas a objetos, sem a necessidade de mapeamento entre tabelas e objetos.

Herança e polimorfismo: Suporte a características da orientação a objetos, como herança, encapsulamento e polimorfismo.

Desvantagens:

Complexidade: Pode ser mais difícil de aprender e usar em comparação com RDBMS.

Menor adoção: Não tão amplamente utilizado quanto RDBMS ou NoSQL.

Exemplos: db4o, ObjectDB.

**Banco de Dados em Nuvem**

Definição: Fornece banco de dados como serviço (DBaaS) através de provedores de nuvem. Pode ser relacional ou NoSQL.

Vantagens:

Escalabilidade automática: Pode crescer ou diminuir de acordo com a demanda.

Acessibilidade: Acesso a partir de qualquer lugar com conexão à internet.

Manutenção reduzida: O provedor de nuvem cuida das atualizações, backups e manutenção.

Desvantagens:

Custo: Pode ser mais caro no longo prazo, especialmente com uso intensivo.

Dependência de terceiros: A disponibilidade e segurança dependem do provedor de nuvem.

Exemplos: Amazon RDS, Google Cloud Spanner, Microsoft Azure SQL Database

**Banco de Dados Relacional**

**Definição:**

Um Banco de Dados Relacional (RDBMS - Relational Database Management System) é um sistema de gerenciamento de banco de dados baseado no modelo relacional, introduzido por Edgar F. Codd em 1970. Ele organiza os dados em tabelas que podem ser relacionadas umas às outras através de chaves. As tabelas contêm dados organizados em linhas (registros) e colunas (campos). O SQL (Structured Query Language) é a linguagem padrão utilizada para a interação com bancos de dados relacionais.

**Tabelas**

Definição: Em um banco de dados relacional, os dados são armazenados em estruturas chamadas tabelas. Uma tabela é composta por linhas e colunas, onde cada linha representa um registro e cada coluna representa um campo de dados específico. Cada tabela armazena informações sobre uma entidade específica (como "clientes", "produtos", etc.).

**Colunas/Campos**

Definição: As colunas, também conhecidas como campos, representam os atributos de uma entidade dentro de uma tabela. Cada coluna tem um nome e um tipo de dado (como inteiro, texto, data, etc.). Por exemplo, em uma tabela "clientes", as colunas poderiam ser "ID", "nome", "email", etc.

**Registros/Tuplas**

Definição: Um registro, ou tupla, é uma única linha em uma tabela, contendo dados para todos os campos dessa tabela. Cada registro representa uma instância única da entidade que a tabela descreve. Por exemplo, em uma tabela "clientes", um registro conteria os dados de um cliente específico.

**Chave Primária**

Definição: A chave primária é um ou mais campos em uma tabela que identificam de forma única cada registro dentro dessa tabela. Ela deve ser única para cada registro e não pode ser nula. A chave primária é fundamental para garantir a integridade dos dados e é frequentemente usada para relacionar tabelas entre si.

**Chave Estrangeira**

Definição: A chave estrangeira é um campo ou combinação de campos em uma tabela que cria uma ligação entre duas tabelas. A chave estrangeira na tabela "filha" referencia a chave primária na tabela "pai". Isso estabelece uma relação entre os dados em diferentes tabelas, permitindo consultas que integrem informações de ambas as tabelas.

**Linguagem SQL:**

**SQL: Definição e Visão Geral**

SQL (Structured Query Language) é uma linguagem de programação utilizada para gerenciar e manipular bancos de dados relacionais. SQL é essencial para a execução de tarefas como criar, modificar, consultar, e excluir dados em um banco de dados.

**Subconjuntos do SQL**

O SQL é dividido em vários subconjuntos, cada um com uma finalidade específica:

**DDL (Data Definition Language):** Conjunto de comandos usados para definir a estrutura do banco de dados e dos objetos que ele contém.

Exemplos de comandos: CREATE, ALTER, DROP, TRUNCATE.

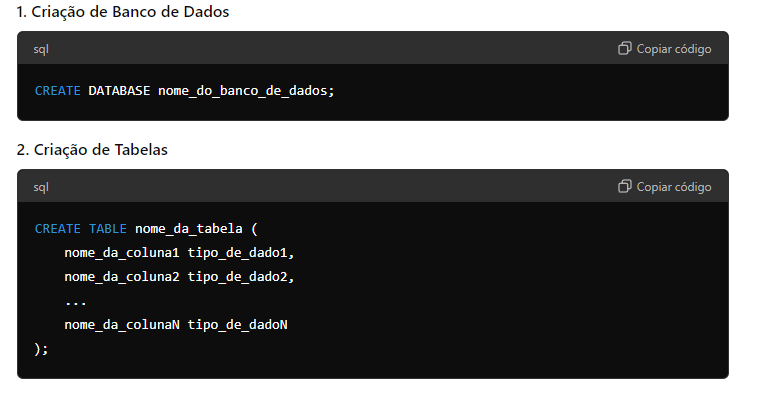
**DML (Data Manipulation Language):** Conjunto de comandos usados para manipular os dados dentro das tabelas.

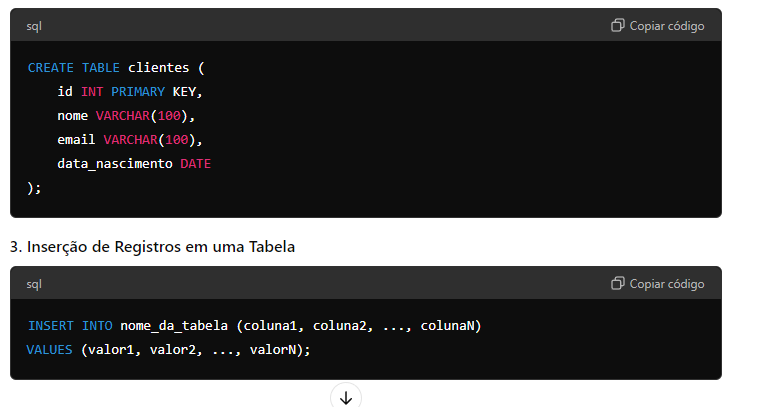
Exemplos de comandos: INSERT, UPDATE, DELETE.

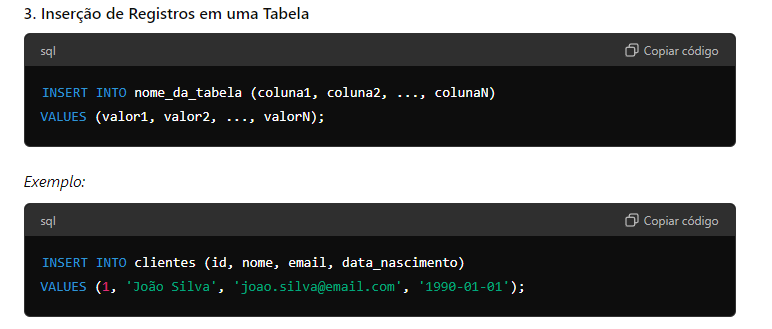
**DQL (Data Query Language):** Conjunto de comandos usados para consultar ou recuperar dados de uma tabela.

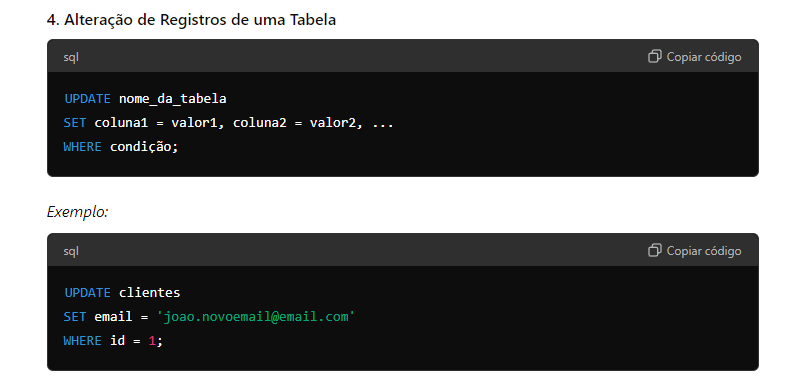
Exemplos de comandos: SELECT.

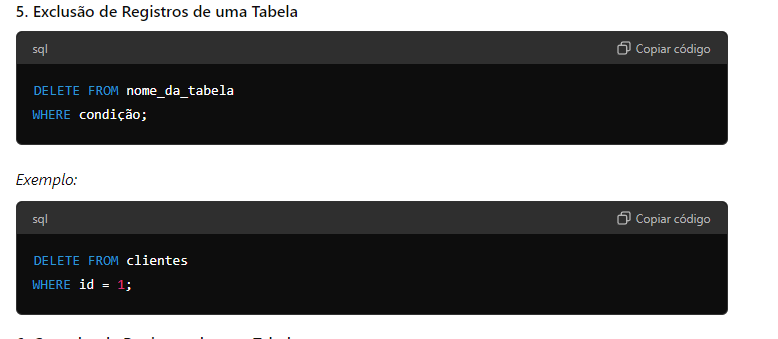
**Sintaxe dos Comandos**

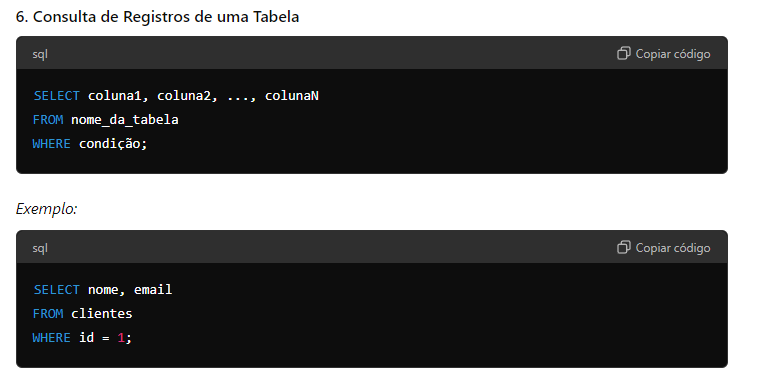












Essa estrutura básica cobre as operações mais comuns realizadas em um banco de dados utilizando SQL.

**Sistemas Gerenciadores de Banco de Dados (SGBD)**

**Definição**

Um Sistema Gerenciador de Banco de Dados (SGBD) é um software que permite criar, gerenciar, e manipular bancos de dados. Ele serve como uma interface entre os usuários e os dados, facilitando operações como inserção, atualização, exclusão e consulta de dados de forma eficiente e segura.

**Como Funcionam**

Os SGBDs funcionam organizando os dados em tabelas que seguem uma estrutura definida por um esquema de banco de dados. Eles gerenciam o armazenamento, recuperação e atualização dos dados, garantindo a integridade e a consistência das informações através de mecanismos como transações e controle de concorrência.

**Principais componentes de um SGBD:**

Motor de armazenamento: Responsável por como os dados são armazenados fisicamente.

Processador de consultas: Interpreta e executa comandos SQL.

Gerenciador de transações: Garante que as operações de banco de dados sejam realizadas de forma atômica, consistente, isolada e durável (propriedades ACID).

Controle de concorrência: Gerencia o acesso simultâneo ao banco de dados por múltiplos usuários.

**MySQL**

**Definição**

MySQL é um dos sistemas gerenciadores de banco de dados relacionais mais populares do mundo. Ele é de código aberto e é amplamente utilizado para o desenvolvimento de aplicações web, devido à sua confiabilidade, facilidade de uso e capacidade de lidar com grandes volumes de dados.

**Principais características:**

Suporte a múltiplas plataformas (Windows, Linux, macOS).

Conformidade com o padrão SQL.

Suporte a diferentes motores de armazenamento, como InnoDB e MyISAM.

Capacidade de trabalhar com grandes quantidades de dados.

Comunidade ativa e extensa documentação.

**MySQL Workbench**

MySQL Workbench é uma ferramenta visual integrada que permite projetar, desenvolver e administrar bancos de dados MySQL. Ela oferece um ambiente gráfico para realizar operações SQL, modelagem de dados, gerenciamento de servidores e criação de relatórios.

**Principais funcionalidades:**

Modelagem de dados: Permite criar diagramas de entidade-relacionamento (ERD) e gerar esquemas de banco de dados.

Desenvolvimento SQL: Inclui um editor de SQL com destaque de sintaxe, auto-completar e execução de scripts SQL.

Administração de servidor: Ferramentas para gerenciar usuários, backups, e configurações de servidores MySQL.

Migração de banco de dados: Auxilia na migração de outros bancos de dados (como SQL Server ou PostgreSQL) para MySQL.

O MySQL Workbench é amplamente utilizado por desenvolvedores e administradores de banco de dados para facilitar a gestão e o desenvolvimento de bancos de dados MySQL.

**FICHAMENTO BIBLIOGRAFICO**

DATE, C. J. An Introduction to Database Systems. 8th ed. Boston: Addison Wesley, 2003.

EL MASRI, Ramez; NAVATHE, Shamkant B. Fundamentals of Database Systems. 7th ed. Boston: Pearson, 2016.

GARCIA-MOLINA, Hector; ULLMAN, Jeffrey D.; WIDOM, Jennifer. Database Systems: The Complete Book. 2nd ed. Upper Saddle River: Pearson, 2008.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. Database System Concepts. 6th ed. New York: McGraw-Hill, 2010.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. Sistemas de banco de dados. 6. ed. Rio de Janeiro: LTC, 2011.

W3Schools. SQL Tutorial. Disponível em: https://www.w3schools.com/sql/. Acesso em: 26 ago. 2024.