**A diferença entre bancos de dados relacionais e não relacionais:**

**O que são bancos de dados relacionais? Cite exemplos.**

Bancos de dados relacionais (RDBMS) é o tipo mais comum de banco de dados, que se armazena em tabelas, normalmente compostas por colunas e linhas. São consideras muito eficientes para armazenar e processar dados que são estruturados, como vendas, dados de clientes ou produtos.

Temos como exemplos MySQL que é open source (código aberto), e é conhecido pela facilidade de uso e simplicidade. O Microsoft SQL server que é mais usado para aplicações corporativas. Oracle que é um banco comercial, sendo uma boa opção para aplicações de grande volume de dados, entre outros.

**O que são bancos de dados não relacionais (NoSQL)? Cite exemplos.**

O banco de dado não relacional (NoSQL) tem uma maior flexibilidade na hora de armazenar os dados, pois não são apenas tabelas com colunas e linhas. Seu modelo de armazenamento é otimizado, se adaptando para a necessidade de cada dado.

Alguns exemplos são Apache CouchDb, que é um baseado em documentos em JSON (é um formato leve e de fácil leitura para a troca de informações entre sistemas computacionais). Elasticsearsch, que é baseado nos documentos que já contém um mecanismo completo para pesquisas de texto. O Couchbase, da a possibilidade de criar aplicativos flexíveis para computar em nuvem, móvel e edge computing.

**Quais as principais diferenças entre os dois tipos de bancos de dados?**

A sua principal diferença é o tipo de armazenamento de dados de ambos. O SQL utiliza a tabela de colunas e linhas, já o NoSQL utiliza um modelo de documento, gráfico e chave-flor, por essa diferença o SQL requer os dados organizados na tabela com um esquema já pré-definido, enquanto o NoSQL, é mais flexível, podendo armazenar em documentos sem seguir um esquema.

Outra grande diferença é a sua escalabilidade. O NoSQL é distribuído em vários servidores pela sua alta escalabilidade, e o SQL não tem essa possibilidade pelas restrições de tamanho.

**Cite situações onde seria mais vantajoso usar um banco de dados relacional e situações onde um banco de dados não relacional seria mais adequado.**

O SQL é recomendado em situações onde exige uma estrutura consistente e organizada, como em estruturas de dados definidos, quando a integridades dos seus dados é crucial, como em estoques, finanças ou registros de pacientes, em consultas complexas que necessitam de filtragem avançada, cálculos agregados, agrupamento e junção de tabelas é mais adequada por tem a segurança e conformidade maior.

O NoSQL é recomendado em aplicações que trabalham com cache onde o desempenho de gravação e leitura é necessário, como um sistema de dados que é acessado de forma frequente e rápida em tempo real, são muito utilizadas pela sua alta velocidade no acesso e recuperação, pode ser recomendado também se requer um sistema de catálogos ou estruturas mais flexíveis, é uma boa escolha pela sua capacidade de conter informações em um único documento.

**Perguntas:**

**Como a informação é organizada em cada tipo de banco?**

SQL é em tabelas de colunas e linhas, já NoSQL em documentos.

**Qual tipo de banco oferece maior flexibilidade para lidar com diferentes tipos de dados?**

NoSQL.

**Qual tipo de banco é mais adequado para lidar com grandes volumes de dados (Big Data)?**

NoSQL.

**Que tipos de aplicações utilizam cada tipo de banco de dados?**

SQL: Finanças, comércio eletrônico, gerenciamento de conteúdo e registros.

NoSQL: Redes sociais, sistemas de recomendação e análise de dados.

**Fontes:**

<https://escolakoru.com.br/blog/banco-de-dados-relacional-e-nao-relacional/>;

<https://www.ibm.com/br-pt/topics/nosql-databases#:~:text=Exemplos%20de%20bancos%20de%20dados%20NoSQL&text=Apache%20CouchDB%2C%20banco%20de%20dados,completo%20de%20pesquisas%20de%20texto>.;

<https://www.impacta.com.br/blog/diferencas-entre-sql-e-nosql-comparando-bancos-de-dados-relacionais-e-nao-relacionais/#:~:text=A%20principal%20diferen%C3%A7a%20entre%20SQL,%2C%20chave%2Dvalor%20ou%20gr%C3%A1fico>.;

<https://www.alura.com.br/artigos/sql-nosql-bancos-relacionais-nao-relacionais>;