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

Um banco de dados relacional (RDBMS - Relational Database Management System) é um tipo de sistema de gerenciamento de banco de dados que organiza dados em tabelas, onde cada tabela é composta por linhas (registros) e colunas (atributos).

## Exemplos

Banco de dados de uma empresa

Sistema de gerenciamento de biblioteca

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

Bancos de dados não relacionais (NoSQL) são sistemas de armazenamento de dados que não seguem o modelo relacional tradicional, onde os dados são organizados em tabelas com linhas e colunas

## Exemplos de bancos de dados NoSQL

MongoDB

**DynamoDB**

**RavenDB**

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

## Bancos de Dados Relacionais (RDBMS)

### Modelo de Dados:

* Baseado em tabelas, com linhas e colunas.
* Dados são organizados em esquemas predefinidos com tabelas que podem ser relacionadas entre si.

### Estrutura:

* Estrutura rígida com esquemas fixos.
* Mudanças no esquema podem exigir alterações significativas na estrutura do banco de dados.

### SQL:

* Utiliza Structured Query Language (SQL) para consultas e manipulação de dados.
* SQL é padrão e bastante expressivo para operações complexas.

### Transações:

* Suporte robusto para transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade), garantindo integridade e consistência dos dados.

### Escalabilidade:

* Escalabilidade vertical (aumento de capacidade de uma única máquina). Escalar horizontalmente (distribuir em várias máquinas) pode ser mais complexo.

### Exemplos:

* MySQL, PostgreSQL, Oracle, Microsoft SQL Server.

## Bancos de Dados Não Relacionais (NoSQL)

### Modelo de Dados:

* Modelos variados: chave-valor, documentos, colunares, grafos, entre outros.
* Estrutura flexível e geralmente mais adaptável a diferentes tipos de dados e formatos.

### Estrutura:

* Esquemas dinâmicos ou ausentes. Dados podem ser armazenados sem um esquema fixo, permitindo maior flexibilidade.

### Consultas:

* Não utiliza SQL. Cada tipo de banco NoSQL tem suas próprias formas de consulta e manipulação de dados.

### Transações:

* Suporte variável para transações ACID. Alguns bancos NoSQL focam mais em escalabilidade e disponibilidade do que em consistência estrita.

### Escalabilidade:

* Geralmente projetados para escalabilidade horizontal (distribuição em várias máquinas) de forma mais natural e eficiente.

### Exemplos:

* MongoDB (documentos), Redis (chave-valor), Cassandra (colunar), Neo4j (grafos).

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

## **Situações para Bancos de Dados Relacionais (RDBMS):**

### **Aplicações com Dados Estruturados e Relacionamentos Complexos:**

* **Exemplo:** Sistemas de gestão empresarial (ERP), sistemas de gerenciamento de clientes (CRM), sistemas bancários.
* **Justificativa:** A estrutura rígida e as relações bem definidas entre tabelas são ideais para modelar dados inter-relacionados e garantir integridade.

### **Aplicações que Exigem Integridade e Consistência dos Dados:**

* **Exemplo: Sistemas financeiros, sistemas de controle de estoque.**
* **Justificativa: Bancos de dados relacionais garantem a consistência dos dados por meio de transações ACID, o que é crucial para evitar erros e garantir a integridade das informações.**

### **Relatórios e Análises Complexas:**

* **Exemplo: Ferramentas de BI (Business Intelligence) que realizam análises complexas e cruzamento de dados.**
* **Justificativa: SQL permite consultas complexas e agregações que são bem suportadas em bancos de dados relacionais.**

### **Aplicações com Esquemas Bem Definidos e Estáveis:**

* **Exemplo: Aplicações corporativas com um modelo de dados que não muda com frequência.**
* **Justificativa: A estrutura de esquema fixo dos bancos relacionais funciona bem quando os dados são bem definidos e estáveis.**

## **Situações para Bancos de Dados Não Relacionais (NoSQL):**

### **Aplicações com Grande Volume de Dados e Requisitos de Escalabilidade Horizontal:**

* **Exemplo: Redes sociais, plataformas de e-commerce com alta demanda de leitura e escrita.**
* **Justificativa: Bancos NoSQL são projetados para escalabilidade horizontal, permitindo distribuir dados entre várias máquinas para lidar com grandes volumes e picos de tráfego.**

### **Dados Sem Estrutura Fixa ou Dados Semi-Estruturados:**

* **Exemplo: Aplicações de gerenciamento de conteúdo, blogs, dados de sensores IoT.**
* **Justificativa: Bancos NoSQL como os de documentos (e.g., MongoDB) podem lidar com dados sem esquema fixo, adaptando-se facilmente a mudanças no formato dos dados.**

### **Alta Disponibilidade e Tolerância a Falhas:**

* **Exemplo: Aplicações distribuídas globalmente, serviços de streaming.**
* **Justificativa: Muitos bancos NoSQL são projetados para alta disponibilidade e tolerância a falhas, garantindo que o sistema continue funcionando mesmo em caso de problemas com partes do banco de dados.**

### **Desempenho em Leitura/Escrita de Alta Velocidade:**

* **Exemplo: Aplicações de caching em tempo real, jogos online.**
* **Justificativa: Bancos de dados NoSQL como Redis (chave-valor) são otimizados para operações de leitura e escrita de alta velocidade, o que é essencial para aplicações que precisam de respostas rápidas.**

### **Dados com Relações Simples ou Sem Relações:**

* **Exemplo: Dados de sessão de usuário, dados de histórico de navegação.**
* **Justificativa: Para dados que não exigem complexas relações entre diferentes entidades, um banco NoSQL pode ser mais eficiente e mais fácil de adaptar às mudanças nos requisitos.**