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

Os conjuntos de dados que são relacionais têm relações pré-definidas entre eles. Por exemplo, um banco de dados que inclui informações do cliente para uma empresa também pode incluir dados de transação individuais anexados a cada conta.

o MySQL ou o Oracle.

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

Os bancos de dados do NoSQL são projetados para vários padrões de acesso aos dados que incluem aplicações de baixa latência. Os bancos de dados de pesquisa NoSQL são projetados para análise de dados semiestruturados. O modelo relacional normaliza dados em tabelas, compostas por linhas e colunas. Exemplo:

Apache CouchDB, banco de dados baseado em documentos em JSON em código aberto que utiliza JavaScript como linguagem de consulta. Elasticsearch, banco de dados baseado em documentos que contém um mecanismo completo de pesquisas de texto.

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

Você pode vincular as tabelas em um banco de dados relacional para obter insights mais profundos sobre a interconexão entre diversos pontos de dados. Por outro lado, bancos de dados não relacionais (ou bancos de dados NoSQL) usam uma variedade de modelos de dados para acessar e gerenciar dados.

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

Banco de dados relacionaloperam com base no modelo relacional, organizando os dados em tabelas. Utilizam SQL para manipulação dos dados. E tem como vantagem a Integridade e segurança dos dados e Padronização e ampla adoção da SQL

Os bancos de dados relacionais são ideais para aplicações que exigem consistência e integridade de dados, enquanto os bancos de dados não relacionais são mais adequados para aplicações que exigem alta escalabilidade e flexibilidade no esquema de dados..

Os bancos de dados NoSQL são projetados para superar as limitações dos sistemas relacionais, especialmente em termos de escalabilidade e flexibilidade de esquema. Eles se dividem em diversos modelos, cada um com características próprias, adequados para diferentes casos de uso.

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

Os bancos de dados relacionais normalmente usam tabelas com dados organizados em linhas (contendo entidades) e colunas (contendo atributos de entidade). Esse processo é conhecido como normalização. Cada linha contém um identificador ou uma chave exclusivo que vincula tabelas para estabelecer uma relação.

Um banco de dados não relacional é um banco de dados que não usa o esquema de tabela de linhas e colunas encontrado na maioria dos sistemas de banco de dados tradicionais. Em vez disso, os bancos de dados não relacionais usam um modelo de armazenamento otimizado para os requisitos específicos do tipo de dados que está sendo armazenado. Por exemplo, os dados podem ser armazenados como pares chave/valor simples, como documentos JSON ou como um gráfico que consiste em bordas e vértices.

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

Banco de Dados NoSQLOs bancos de dados NoSQL oferecem escalabilidade horizontal, alta disponibilidade e flexibilidade na modelagem de dados. Eles são amplamente usados em aplicativos da web, onde a velocidade e a capacidade de lidar com dados em constante mudança são essenciais.

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

Para lidar com um fluxo muito grande, típico do que chamamos de Big Data, é preciso contar com um banco de dados não relacional. Os não relacionais não apresentam esquemas, nem demandam relações entre os dados. Geralmente, usam chave-valor, grafos, colunas ou documentos como forma de organização dos dados.

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

Os bancos de dados relacionais são ideais para aplicações que exigem consistência e integridade de dados, enquanto os bancos de dados não relacionais são mais adequados para aplicações que exigem alta escalabilidade e flexibilidade no esquema de dados