**Banco de dados**

**O que é**

é uma coleção organizada de dados que pode ser facilmente acessada, gerenciada e atualizada. Ele é utilizado para armazenar informações de forma estruturada, permitindo consultas eficientes e manipulação dos dados. Os bancos de dados permitem que aplicações façam operações como inserir, atualizar, excluir e consultar dados. Eles são fundamentais para a maioria das aplicações modernas, desde sistemas de gestão até websites.

Os bancos de dados permitem que aplicações façam operações como inserir, atualizar, excluir e consultar dados. Eles são fundamentais para a maioria das aplicações modernas, desde sistemas de gestão até websites.

Um banco de dados é uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um [sistema de gerenciamento de banco de dados (DBMS)](https://www.oracle.com/br/database/what-is-database/#WhatIsDBMS). Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados.

Os dados nos tipos mais comuns de bancos de dados em operação atualmente são modelados em linhas e colunas em uma série de tabelas para tornar o processamento e a consulta de dados eficientes. Os dados podem ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados. A maioria dos bancos de dados usa a linguagem de consulta estruturada (SQL) para escrever e consultar dados.

**Banco de dados relacional**

Um banco de dados relacional é um tipo de banco de dados que organiza dados em tabelas estruturadas, onde cada tabela é composta por linhas e colunas. As principais características de um banco de dados relacional incluem:

1. **Estrutura em Tabelas**: Os dados são organizados em tabelas (ou relações), onde cada tabela tem um nome único e é composta por registros (linhas) e atributos (colunas).
2. **Chaves Primárias e Estrangeiras**: Cada tabela possui uma chave primária que identifica de forma única cada registro. Chaves estrangeiras são usadas para relacionar tabelas diferentes, estabelecendo vínculos entre os dados.
3. **SQL (Structured Query Language)**: A linguagem padrão para interagir com bancos de dados relacionais. Com o SQL, é possível realizar operações como consultas, inserções, atualizações e exclusões de dados.

SQL é uma linguagem de programação usada por quase todos [os bancos de dados relacionais](https://www.oracle.com/br/database/what-is-database/#relational) para consultar, manipular e definir dados e fornecer controle de acesso. O SQL foi desenvolvido pela primeira vez na IBM nos anos 1970, com a Oracle como principal contribuinte, o que levou à implementação do padrão SQL ANSI; o SQL estimulou muitas extensões de empresas como IBM, Oracle e Microsoft. Embora o SQL ainda seja amplamente usado hoje em dia, novas linguagens de programação estão começando a aparecer.

1. **Integridade dos Dados**: Os bancos de dados relacionais garantem a integridade dos dados por meio de regras como restrições de unicidade, integridade referencial e validações.
2. **Transações**: Suportam operações que podem ser agrupadas em transações, garantindo que as alterações nos dados sejam feitas de maneira segura e consistente.

Exemplos populares de bancos de dados relacionais incluem MySQL, PostgreSQL, Oracle Database e Microsoft SQL Server. Esses sistemas são amplamente utilizados em aplicações empresariais, sites e serviços que requerem gerenciamento eficiente de dados estruturados.

Armazenam dados em tabelas, com linhas e colunas, e seguem um esquema rígido de organização. São ideais para aplicações que exigem consistência e integridade de dados, e que precisam de análises complexas.

**Banco de dados não relacional**

Um banco de dados não relacional, também conhecido como NoSQL, é um tipo de banco de dados que não segue o modelo relacional tradicional de tabelas, linhas e colunas. Em vez disso, ele utiliza diferentes estruturas de armazenamento que permitem maior flexibilidade e escalabilidade. Aqui estão algumas características principais:

1. **Estruturas Diversas**: Os bancos de dados não relacionais podem armazenar dados em várias formas, como documentos (JSON, BSON), pares chave-valor, colunas ou grafos. Exemplos incluem:
   * **Document Stores**: Armazenam dados em documentos. Exemplo: MongoDB.
   * **Key-Value Stores**: Armazenam dados como pares chave-valor. Exemplo: Redis.
   * **Column Family Stores**: Armazenam dados em colunas. Exemplo: Cassandra.
   * **Graph Databases**: Focados em armazenar e consultar dados em forma de grafos. Exemplo: Neo4j.
2. **Escalabilidade Horizontal**: Facilita a adição de mais servidores para lidar com aumentos de carga, permitindo que as aplicações escalem de forma mais eficiente.
3. **Flexibilidade de Esquema**: Não requer um esquema fixo, o que permite que os dados sejam armazenados de forma mais dinâmica. Isso é útil em cenários onde os dados podem mudar frequentemente.
4. **Desempenho e Velocidade**: Muitos bancos de dados não relacionais são otimizados para leituras e gravações rápidas, tornando-os adequados para aplicações que requerem alto desempenho.
5. **Consistência Eventual**: Em muitos sistemas NoSQL, a consistência dos dados pode ser eventual, o que significa que, em vez de garantir que todas as réplicas dos dados estejam sempre sincronizadas, o sistema permite algumas divergências temporárias.

Os bancos de dados não relacionais são amplamente utilizados em aplicações que lidam com grandes volumes de dados, dados não estruturados ou em ambientes onde a escalabilidade e a flexibilidade são essenciais, como redes sociais, e-commerce e análise de dados em tempo real.

* São mais flexíveis e permitem armazenar dados de forma não estruturada ou semiestruturada, como imagens, vídeos e documentos. São mais adequados para aplicações que precisam de alta escalabilidade e flexibilidade no esquema de dados, e que trabalham com grandes volumes de dados.

Aqui estão as principais diferenças entre MySQL, PostgreSQL e Oracle Database:

### 1. **Licença e Acesso**

* **MySQL**: Licença GPL, mas possui versões comerciais. Controlado pela Oracle Corporation.
* **PostgreSQL**: Licença de código aberto (PostgreSQL License), semelhante ao MIT, o que permite uso livre, modificação e distribuição.
* **Oracle Database**: Licença comercial com custos significativos. Existe uma versão gratuita limitada (Oracle Database Express Edition).

### 2. **Recursos**

* **MySQL**: Fornece recursos básicos para gerenciamento de dados e é fácil de usar, mas com suporte limitado para tipos de dados avançados e consultas complexas.
* **PostgreSQL**: Conhecido por sua extensibilidade e suporte a recursos avançados, como tipos de dados personalizados, funções, transações complexas e extensões (como PostGIS para dados geoespaciais).
* **Oracle Database**: Oferece um conjunto robusto de recursos empresariais, incluindo suporte a replicação, partição de dados, segurança avançada e análise de dados.

### 3. **Desempenho**

* **MySQL**: Geralmente rápido em operações de leitura, mas pode apresentar limitações em operações complexas e grandes volumes de dados.
* **PostgreSQL**: Altamente otimizado para consultas complexas e operações de leitura e escrita, especialmente em ambientes com alta concorrência.
* **Oracle Database**: Projetado para lidar com grandes volumes de dados e operações críticas em ambientes empresariais, oferecendo desempenho superior em cenários complexos.

### 4. **Comunidade e Suporte**

* **MySQL**: Tem uma comunidade ativa, mas o suporte é mais voltado pela Oracle, com um modelo de suporte comercial.
* **PostgreSQL**: Possui uma comunidade robusta e ativa, com muitas contribuições e documentação extensa, além de suporte comercial disponível.
* **Oracle Database**: Suporte comercial forte, com documentação detalhada e serviços de suporte, mas a comunidade é menos centralizada.

### 5. **Usabilidade**

* **MySQL**: Considerado amigável para iniciantes, com uma curva de aprendizado suave.
* **PostgreSQL**: Pode ter uma curva de aprendizado mais acentuada devido à sua riqueza de recursos e flexibilidade, mas oferece uma interface poderosa.
* **Oracle Database**: Complexidade elevada, pode ter uma curva de aprendizado acentuada, especialmente em sua configuração e gerenciamento.

### 6. **Casos de Uso**

* **MySQL**: Ideal para aplicações web, sites e projetos de pequeno a médio porte.
* **PostgreSQL**: Preferido em cenários que requerem manipulação de dados complexos, análise avançada e ambientes científicos.
* **Oracle Database**: Comumente utilizado em grandes empresas e aplicações críticas que exigem robustez, escalabilidade e alta disponibilidade.

Aqui está um resumo das principais diferenças entre MySQL, PostgreSQL e Oracle Database:

### MySQL

* **Tipo:** Relacional, baseado em servidor.
* **Uso:** Amplamente utilizado em aplicações web e sistemas que exigem alta performance de leitura.
* **Recursos:**
  + Suporte a replicação e clustering.
  + Transações ACID (com o mecanismo de armazenamento InnoDB).
  + Boa performance em operações de leitura.
* **Licença:** Licença dual (open-source e comercial).
* **Facilidade de uso:** Interface amigável e fácil de configurar, com uma comunidade grande.

### PostgreSQL

* **Tipo:** Relacional, baseado em servidor.
* **Uso:** Ideal para aplicações que requerem complexidade em consultas e robustez em integridade de dados.
* **Recursos:**
  + Suporte a tipos de dados avançados (JSON, XML).
  + Extensibilidade (ex.: funções personalizadas e operadores).
  + Transações ACID e suporte a consultas complexas.
* **Licença:** Licença open-source permissiva.
* **Desempenho:** Excelente em operações complexas e análises de dados.

### Oracle Database

* **Tipo:** Relacional, baseado em servidor.
* **Uso:** Usado em grandes empresas para aplicações críticas que exigem alto desempenho e escalabilidade.
* **Recursos:**
  + Ferramentas avançadas de segurança, recuperação e backup.
  + Suporte a PL/SQL, particionamento de dados e várias opções de replicação.
  + Transações ACID com alta integridade de dados.
* **Licença:** Licença comercial, com versões gratuitas limitadas (Oracle Express).
* **Desempenho:** Muito robusto em ambientes de alta carga, mas pode ser mais caro e complexo de gerenciar.

### Resumo

* **MySQL** é ideal para aplicações web de alto desempenho e fácil configuração.
* **PostgreSQL** é a escolha preferida para aplicações que exigem complexidade e robustez em dados.
* **Oracle Database** é voltado para grandes empresas que precisam de soluções robustas e escaláveis, mas pode ser mais caro e complexo.