



Estrutura e Arquitetura de **Dados**

—

Tipos de Dados

Como vimos anteriormente, os dados são informações que podem ser armazenadas, processadas e transmitidas por meios digitais. No entanto, nem todos os dados têm a mesma forma, estrutura ou finalidade. Dependendo da maneira como os dados são organizados, podemos classificá-los em três categorias principais: dados estruturados, dados não-estruturados e dados semi-estruturados.



Vamos relembrar os conceitos:

- **Dados estruturados** são aqueles que seguem um formato padronizado e pré definido, que facilita a sua manipulação e análise. Exemplos de usos de dados estruturados são: armazenar informações de clientes, produtos ou vendas; realizar consultas, cálculos ou relatórios; e aplicar algoritmos de aprendizado de máquina.
- **Dados não-estruturados** são aqueles que não seguem um formato específico ou padronizado, e que podem variar em tamanho, tipo e conteúdo. Exemplos de usos de dados não-estruturados são: extrair sentimentos, tópicos ou entidades de textos; reconhecer rostos, objetos ou emoções em imagens ou vídeos; e transcrever ou traduzir áudios.
- **Dados semi-estruturados** são aqueles que possuem algum grau de organização, mas que não se encaixam completamente em um formato rígido ou padronizado. Exemplos de usos de dados semi-estruturados são: armazenar e trocar dados entre diferentes sistemas ou aplicações; integrar dados de diversas fontes ou formatos; e indexar ou pesquisar dados na web.

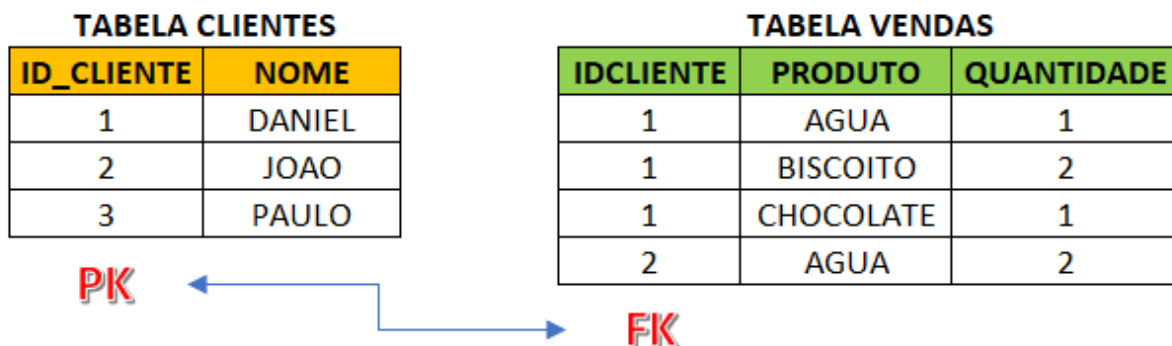
Bancos de dados

Um banco de dados é um conjunto organizado de dados que podem ser armazenados, acessados, manipulados e analisados de forma eficiente. Os tipos de bancos de dados são amplamente agrupados em bancos de dados relacionais e não relacionais.

Bancos de dados relacionais

Sua estrutura usa tabelas, com linhas e colunas, para organizar os dados e compreende uma linguagem de programação chamada SQL (Structured Query Language). Cada linha corresponde a um registro ou entidade, e cada coluna corresponde a um atributo ou propriedade.

Entre os exemplos de bancos de dados relacionais estão o SQL Server, SQL do Azure, MySQL, PostgreSQL e MariaDB.



PK = Primary Key (chave primária); FK = Foreign Key (chave estrangeira)

As chaves primárias e as chaves estrangeiras, que são valores únicos que reconhecem cada registro, permitem relacionar as tabelas entre si. Edgar F. Codd foi quem propôs o modelo relacional em 1970, e ele se tornou o padrão para a maioria dos sistemas de gerenciamento de banco de dados (SGBD).

Bancos de dados relacionais garantem a consistência e integridade dos dados, seguindo as propriedades **ACID** (atomicidade, consistência, isolamento e durabilidade).

Bancos de dados não relacionais (NoSQL)

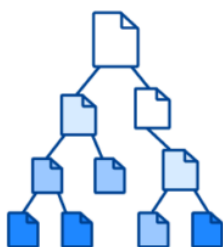
São sistemas que armazenam e recuperam dados (semi estruturados ou não estruturados) de forma diferente do modelo relacional tradicional, que usa tabelas, colunas e linhas. O termo NoSQL significa “not only SQL” (não apenas SQL), indicando que esses sistemas podem usar outras formas de consulta além da linguagem padrão SQL.

Bancos de dados NoSQL priorizam a disponibilidade e a tolerância a falhas, seguindo as propriedades **BASE** (basicamente disponíveis, eventualmente consistentes).

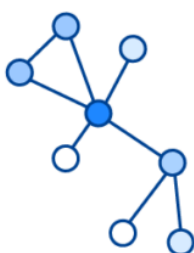
Os bancos de dados NoSQL têm algumas características distintas dos bancos de dados relacionais, tais como:

- **Flexibilidade:** permitem guardar dados de diferentes formatos, estruturas e tipos, sem precisar mudar o esquema;
- **Escalabilidade:** permite aumentar a capacidade de armazenamento e processamento de dados distribuindo-os em vários nós de um cluster, sem afetar o desempenho e a disponibilidade;
- **Performance:** oferecem maior rapidez e eficiência nas operações de leitura e escrita de dados, especialmente quando se trata de grandes quantidades de dados;
- **Simplicidade:** dispensam o uso de modelos complexos e normalizados para representar os dados, simplificando o desenvolvimento e a manutenção das aplicações.

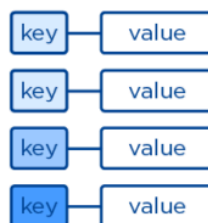
Document



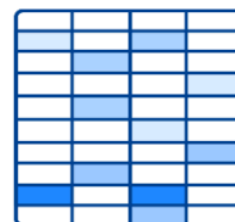
Graph



Key-Value



Wide-column



Há diferentes tipos de bancos de dados NoSQL, cada um com suas próprias características, vantagens e desvantagens. Os principais tipos são:

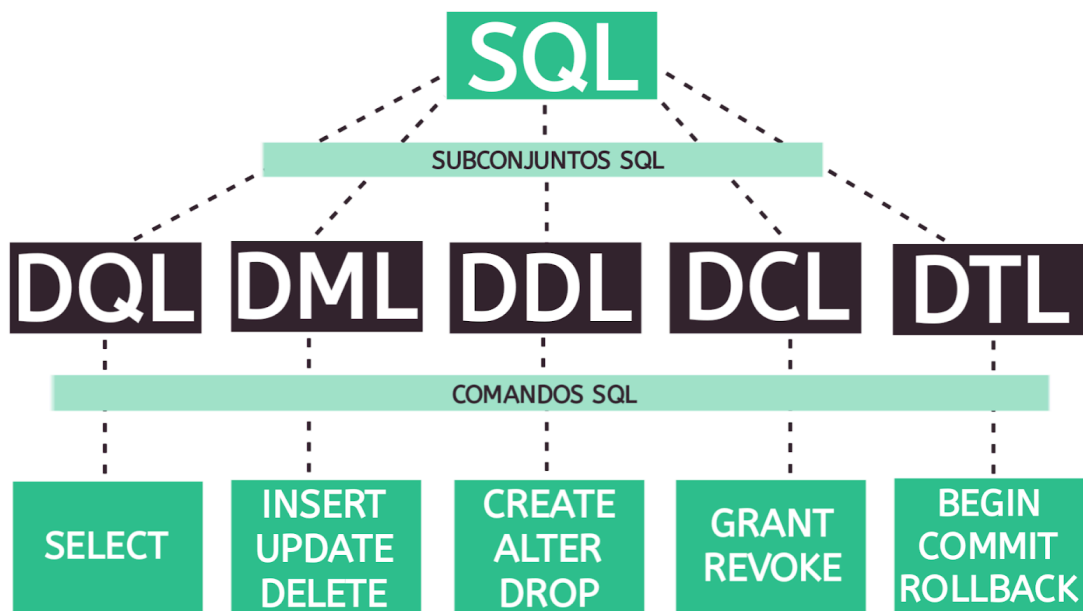
- **Bancos de dados orientados a documentos:** guardam os dados em forma de documentos, geralmente em formatos como JSON ou XML, que contêm pares de chave-valor. Exemplos: MongoDB, CouchDB, Firebase.
- **Bancos de dados orientados a grafos:** guardam os dados em forma de nós e arestas, representando as entidades e as relações entre elas. São ideais para modelar redes complexas e realizar consultas baseadas em padrões e caminhos. Exemplos: Neo4j, OrientDB, ArangoDB.
- **Bancos de dados orientados a chave-valor:** guardam os dados em forma de pares de chave-valor, sendo a chave o identificador único do dado e o valor do conteúdo associado. São os mais simples e rápidos, mas também os mais limitados em termos de funcionalidades. Exemplos: Redis, Memcached, DynamoDB.
- **Bancos de dados orientados a colunas:** guardam os dados em forma de colunas, em vez de linhas, o que permite uma maior compressão e eficiência nas consultas analíticas. Exemplos: Cassandra, HBase, Bigtable.

Os bancos de dados NoSQL não são necessariamente melhores que os bancos de dados relacionais, mas sim apropriados para diferentes situações e necessidades. A escolha do tipo de banco de dados depende de fatores como o volume, a diversidade, a velocidade, a estrutura, a complexidade, a consistência e a confiabilidade dos dados. Portanto, é importante saber as características, vantagens e desvantagens de cada tipo de banco de dados, bem como as aplicações e os casos de uso mais comuns para cada um.

Por que aprender SQL é importante?

SQL é a principal linguagem de manipulação de banco de dados e uma habilidade essencial para profissionais que lidam com dados, como cientistas de dados, analistas de negócios, engenheiros de dados, entre outros. Dominar SQL pode abrir muitas oportunidades de trabalho e crescimento profissional, além de facilitar a comunicação com outras áreas da organização.

A linguagem SQL organiza seus comandos em 5 subconjuntos:



- **DDL - Data Definition Language - Linguagem de Definição de Dados**
 - São os comandos que interagem com os objetos do banco
- **DML - Data Manipulation Language - Linguagem de Manipulação de Dados**
 - São os comandos que interagem com os dados dentro das tabelas
- **DQL - Data Query Language - Linguagem de Consulta de dados**
 - São os comandos de consulta
- **DTL - Data Transaction Language (ou TCL) - Linguagem de Transação de Dados**
 - São os comandos para controle de transação
- **DCL - Data Control Language - Linguagem de Controle de Dados**
 - São os comandos para controlar a parte de segurança do banco de dados

Prepare-se para colocar a mão na massa!

Nas lições a seguir, você vai descobrir como usar SQL e Python para manipular dados de forma eficiente e flexível!