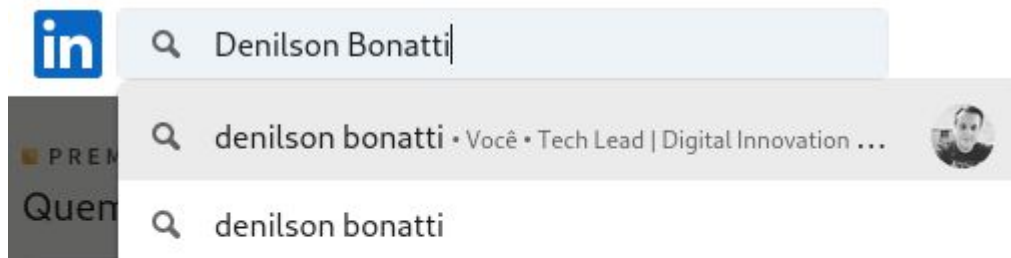


As principais diferenças entre banco de dados SQL e NoSQL

Denilson Bonatti

Tech Lead – DIO

Mais sobre mim



O que é um banco de dados relacional?

Um banco de dados relacional 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 (SGBD).

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.

The diagram shows a table with 5 columns and 3 rows. Red boxes highlight the first column (labeled 'Coluna'), the first two rows (labeled 'Linha (registro)'), and the cell containing 'da Graça' (labeled 'Campo').

ID	Nome	Sobrenome	Cidade	Estado
1	José	da Silva	São Paulo	SP
2	Maria	da Graça	São Paulo	SP

Qual é a diferença entre um banco de dados e uma planilha?

Bancos de dados e planilhas (como o Microsoft Excel) são modos convenientes de armazenar informações. As principais diferenças entre os dois são:

- Como os dados são armazenados e manipulados
- Quem pode acessar os dados
- Quantos dados podem ser armazenados

As planilhas foram originalmente projetadas para um usuário e suas características refletem isso. São ótimos para um único usuário ou um pequeno número de usuários que não precisam fazer manipulações de dados extremamente complicadas.

Bancos de dados, por outro lado, são projetados para conter coleções muito maiores de informações organizadas, quantidades enormes, às vezes. Os bancos de dados permitem que vários usuários, ao mesmo tempo, acessem e consultem com rapidez e segurança os dados usando lógica e linguagem altamente complexas.

O que é um SGBD?

Um Sistema de Gerenciamento de Banco de Dados (SGBD) é o conjunto de programas de computador (softwares) responsáveis pelo gerenciamento de uma base de dados. O SGBD disponibiliza uma interface para que os clientes do Banco de dados possam incluir, alterar ou consultar dados previamente armazenados.

383 systems in ranking, February 2022

Rank			DBMS	Database Model	Score		
Feb 2022	Jan 2022	Feb 2021			Feb 2022	Jan 2022	Feb 2021
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1256.83	-10.05	-59.84
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1214.68	+8.63	-28.69
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	949.05	+4.24	-73.88
4.	4.	4.	PostgreSQL + ⓘ	Relational, Multi-model ⓘ	609.38	+2.83	+58.42
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	488.64	+0.07	+29.69
6.	6.	↑ 7.	Redis +	Key-value, Multi-model ⓘ	175.80	-2.18	+23.23
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model ⓘ	162.88	-1.32	+5.26
8.	8.	8.	Elasticsearch	Search engine, Multi-model ⓘ	162.29	+1.54	+11.29
9.	9.	↑ 11.	Microsoft Access	Relational	131.26	+2.31	+17.09
10.	10.	↓ 9.	SQLite +	Relational	128.37	+0.94	+5.20
11.	11.	↓ 10.	Cassandra +	Wide column	123.98	+0.43	+9.36
12.	12.	12.	MariaDB +	Relational, Multi-model ⓘ	107.11	+0.69	+13.22
13.	13.	13.	Splunk	Search engine	90.82	+0.37	+2.28
14.	14.	↑ 15.	Microsoft Azure SQL Database	Relational, Multi-model ⓘ	84.95	-1.37	+13.67
15.	↑ 17.	↑ 35.	Snowflake +	Relational	83.18	+6.36	+64.96
16.	↓ 15.	↓ 14.	Hive +	Relational	81.88	-1.57	+9.56
17.	↓ 16.	17.	Amazon DynamoDB +	Multi-model ⓘ	80.36	+0.50	+11.21
18.	18.	↓ 16.	Teradata +	Relational, Multi-model ⓘ	68.57	-0.56	-2.33
19.	19.	↑ 20.	Solr	Search engine, Multi-model ⓘ	58.53	+0.00	+7.84
20.	20.	↓ 19.	Neo4j +	Graph	58.25	+0.21	+6.08

Restrição de integridade

Restrições de integridade são usados para garantir a exatidão e a consistência dos dados em um Banco de dados relacional. Ou seja, garantir que dados representem assertivamente a realidade modelada. A integridade dos dados é tratada nas bases de dados através do conceito de integridade relacional e é garantida pelo próprio SGBD.

ID	Nome	Sobrenome	Cidade	Estado
1	José	da Silva	São Paulo	SP
2	Maria	da Graça	Sao Paulo	sp
2	Carlos	Pereira	Saõ Paulo	S.P.
4	Denilson	dos Anjos	Brasília	DF
4	Mariana	Peres	Brasilia	D.F.
null	Roberto	Carlos		

A chave primária, ou Primary key (PK) é o identificador único de um registro na tabela. Pode ser constituída de um campo (chave simples) de tal maneira que não existam dois registros com o mesmo valor de chave primária. A Primary Key , não permite valores nulos e impõe a exclusividade de linhas.

ID	Nome	Sobrenome	Cidade	Estado
1	José	da Silva	São Paulo	SP
2	Maria	da Graça	Sao Paulo	sp
3	Carlos	Pereira	Saõ Paulo	S.P.
4	Denilson	dos Anjos	Brasília	DF
5	Mariana	Peres	Brasilia	D.F.

No contexto dos banco de dados, o conceito de chave estrangeira ou chave externa se refere ao tipo de relacionamento entre distintas tabelas de dados do banco de dados.

Uma chave estrangeira é chamada quando há o relacionamento entre duas tabelas.

Cidade_ID	Cidade
1	São Paulo
2	Brasília

ID	Nome	Sobrenome	Cidade_ID	Estado_ID
1	José	da Silva	1	1
2	Maria	da Graça	1	1
3	Carlos	Pereira	1	1
4	Denilson	dos Anjos	2	2
5	Mariana	Peres	2	2

Estado_ID	Estado
1	São Paulo
2	Brasília

SQL

SQL é uma linguagem de programação usada por quase todos os bancos de dados relacionais 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. Embora o SQL ainda seja amplamente usado hoje em dia, novas linguagens de programação estão começando a aparecer.

Banco de dados não relacionais (NoSQL)

NoSQL é um termo genérico que representa os bancos de dados não relacionais. Uma classe definida de banco de dados que fornecem um mecanismo para armazenamento e recuperação de dados que são modelados de **formas** diferentes das relações tabulares usadas nos bancos de dados relacionais.

Banco de dados não relacionais (NoSQL)

Bancos de dados NoSQL são criados para modelos de dados específicos e têm esquemas flexíveis para a criação de aplicativos modernos. Os bancos de dados NoSQL são amplamente reconhecidos por sua facilidade de desenvolvimento, funcionalidade e performance em escala.

Por que utilizar um banco de dados NoSQL?

Os bancos de dados NoSQL são ideais para muitos aplicativos modernos, como dispositivos móveis, Web e jogos, que exigem bancos de dados flexíveis, escaláveis, de alta performance e altamente funcionais para proporcionar ótimas experiências aos usuários.

Flexibilidade: os bancos de dados NoSQL geralmente fornecem esquemas flexíveis que permitem um desenvolvimento mais rápido e iterativo. O modelo de dados flexível torna os bancos de dados NoSQL ideais para dados semiestruturados e não estruturados.

Escalabilidade: os bancos de dados NoSQL geralmente são projetados para serem escalados horizontalmente usando clusters distribuídos de hardware, em vez de escalá-los verticalmente adicionando servidores caros e robustos. Alguns provedores de nuvem lidam com essas operações nos bastidores como um serviço totalmente gerenciado.

Alta performance: o banco de dados NoSQL é otimizado para modelos de dados específicos e padrões de acesso que permitem maior performance do que quando se tenta realizar uma funcionalidade semelhante com bancos de dados relacionais.

Tipos de bancos de dados

NoSQL

Chave-valor: os bancos de dados de chave-valor são altamente particionáveis e permitem escalabilidade horizontal em escalas que outros tipos de bancos de dados não conseguem alcançar.
Exemplo: Amazon Dynamo DB

Documento: no código do aplicativo, os dados costumam ser representados como um objeto ou um documento do tipo JSON porque esse é um modelo de dados eficiente e intuitivo para os desenvolvedores. Os bancos de dados de documentos facilitam para que os desenvolvedores armazenem e consultem dados usando o mesmo formato de modelo de documento que usam no código do aplicativo.

Exemplo: MongoDB

Em memória: aplicações de jogos e tecnologia de publicidade têm casos de uso como placares de líderes, armazenamentos de sessões e análises em tempo real que exigem tempos de resposta em microssegundos e podem ter grandes picos de tráfego a qualquer momento.

Exemplo: Redis

Banco de dados noSQL

```
{
  "id": 49,
  "Pais": "Alemanha",
  "Regiao": "Europa",
  "Populacao":
  "PrincipaisCidades": [
    {
      "NomeCidade": "Berlin",
      "Populacao": 3610156,
    },
    {
      "NomeCidade": "Hamburg",
      "Populacao": 1746342,
    }
  ]
}
```

NoSQL orientado a documento