

Digital
College

SQL para Devs: Construindo a Base do seu Back End

Prof. David Wanderson



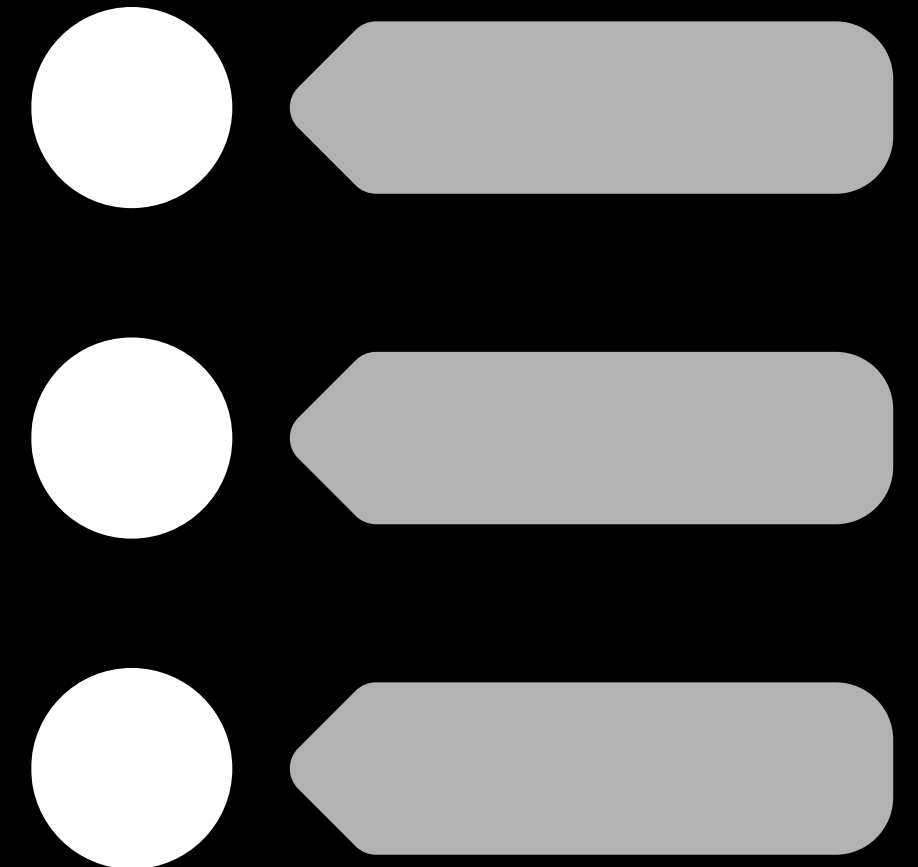
Analista de Sistemas, Especialista em Engenharia de Software. MBA em Gestão de TI, Inteligência Artificial e Machine Learning. Atualmente sou Consultor de TI Sênior na Geplant Tecnologia Florestal, e professor há mais de 11 anos. Meus principais conhecimentos e experiências são com as stacks: Python, SQL, JavaScript, HTML/CSS, NodeJS, ReactJS e análise de dados.

Estrutura Geral

Tema: SQL para Devs: Construindo a Base do seu Back End

Duração: 3h (online, ao vivo e prática)

Público: Devs iniciantes/intermediários em Back End

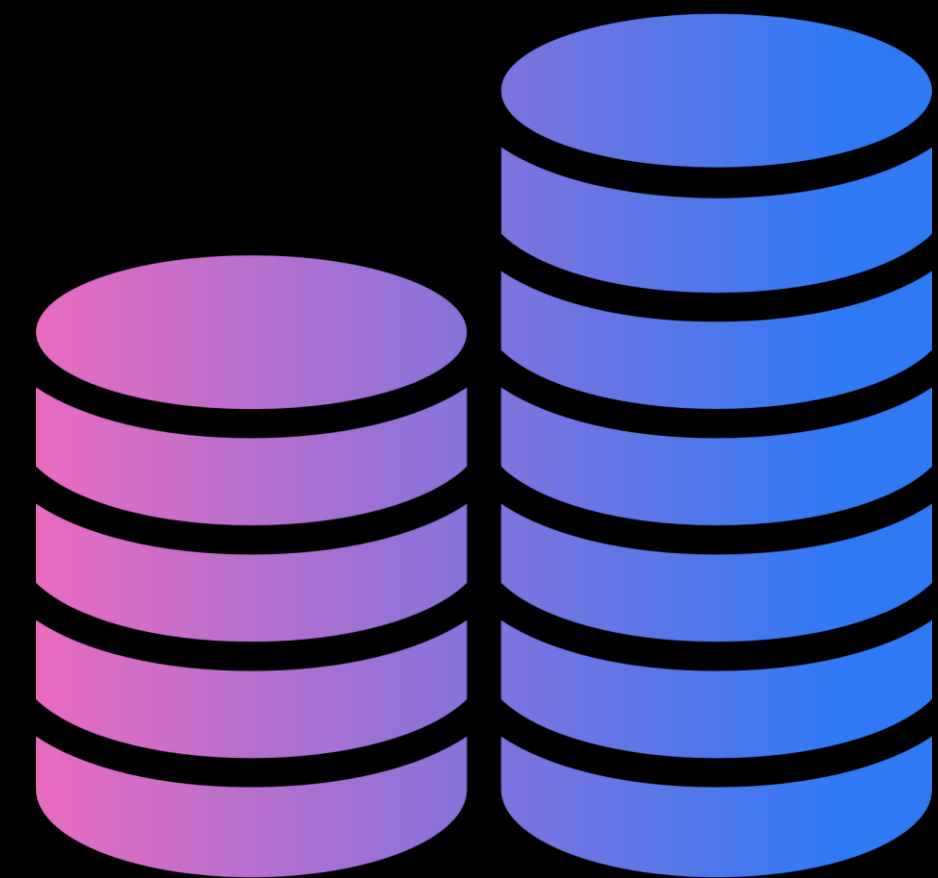


Introdução ao Banco de Dados

O que é um banco de dados?

Um banco de dados é um conjunto organizado de informações que podem ser armazenadas, acessadas e manipuladas de forma eficiente por um sistema computacional.

Ele serve para guardar dados de forma estruturada, permitindo que aplicações — como sites, sistemas e APIs — busquem, insiram, alterem e excluam informações rapidamente.

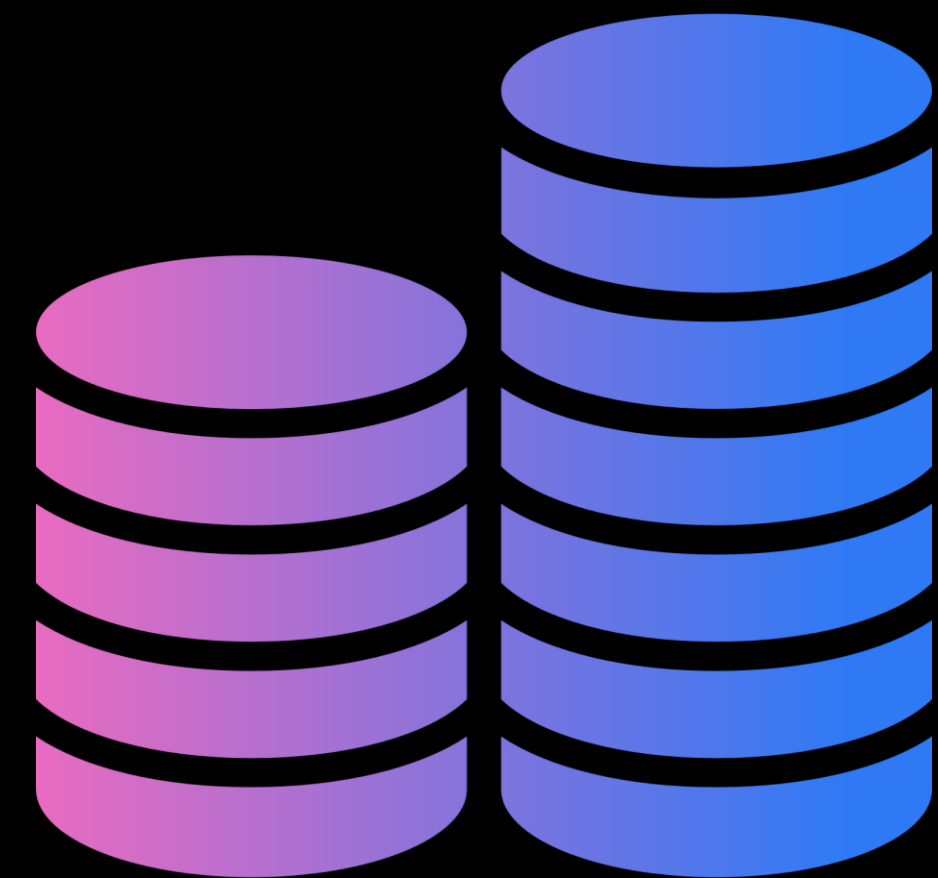


Introdução ao Banco de Dados

Banco relacional x Banco não relacional

Bancos Relacionais: São bancos que organizam os dados em tabelas, com linhas (registros) e colunas (campos), assim como uma planilha.

Bancos Não Relacionais (NoSQL): Armazenam dados de forma flexível e não tabular, geralmente em formato documento (JSON), chave-valor, grafo ou colunas.



Introdução ao Banco de Dados



Let the game begin



Criando nosso Banco

Vamos escolher uma opção de banco de dados em nuvem e gratuito

Exemplos:

- Supabase
- Neon

Tipos de dados SQL

INTEGER (INT)	Números inteiros
SERIAL	Inteiro autoincremental (geralmente utilizado para id)
DECIMAL (p,s)	Número decimal preciso (p = total de dígitos, s = casas decimais)
REAL	Número de ponto flutuante (menos preciso)Número de ponto flutuante (menos preciso)
CHAR (n)	Texto de tamanho fixo.
VARCHAR(n)	Texto de tamanho variável até 'n' caracteres
DATE	Apenas data (AAAA-MM-DD)
TIMESTAMP	Data e hora (AAAA-MM-DD HH:MM:SS)

Comandos SQL

CREATE TABLE:

```
CREATE TABLE usuarios (  
  id SERIAL PRIMARY KEY,  
  nome VARCHAR(100) NOT NULL,  
  email VARCHAR(100) UNIQUE,  
  data_nasc DATE,  
  cpf CHAR(11) UNIQUE  
);
```

Comandos SQL

INSERT INTO

```
INSERT INTO usuarios (nome, email, data_nasc, cpf) VALUES  
('João Silva', 'joao.silva@email.com', '1990-05-10', '12345678901')
```

Comandos SQL

SELECT

```
SELECT * FROM usuarios;
```

ou

```
SELECT nome, email FROM usuarios  
WHERE data_nasc > '1990-01-01';
```

Comandos SQL

UPDATE

```
UPDATE usuarios  
SET nome = 'João da Silva Júnior'  
WHERE id = 1;
```

DELETE

```
DELETE FROM usuarios  
WHERE id = 1;
```

Estrutura básica API

```
digital-api/
├── .env
├── package.json
├── README.md
├── src/
│   ├── index.js           # servidor express principal
│   ├── app.js             # configura express, middlewares e rotas
│   ├── config/
│   │   └── database.js    # configuração do Sequelize
│   ├── models/
│   │   └── usuario.model.js # definição do modelo Usuario
│   ├── controllers/
│   │   └── usuarios.controller.js
│   ├── routes/
│   │   └── usuarios.routes.js
│   └── utils/
│       └── handleErrors.js # helpers para erros (opcional)
```