



# Banco de Dados - SQL/NoSQL






Conceitos, estrutura e aplicação

**De Alunos Para Alunos**

Por: Pedro Henrique



## This is Pedro...

-  Comecei na jornada tech aos 16 anos, trabalhei como Suporte Técnico N1, N2 e como Software Tester com foco em Testes Automatizados.
-  Estudante de Web Development (WEB)
-  Alguns apelidos: **DON**, **PH**, **PEDRIIN**
-  Finalizando o bootcamp da TripleTen
-  Apaixonado por RAP, basquete e Assassins Creed

💬 “Nada é verdade, tudo é permitido!”



## Conteúdo:

1. Conceitos Básicos
2. Exemplos de SGBDs
3. Relacional vs Não-Relacional
4. Tipos de relacionamento
5. CRUD
6. Exemplos práticos
7. Q&A



- [illegible]



## Conceitos Básicos

### - O QUE SÃO DADOS?

Dados do latim DATUM - Peça única de informação.

Quando organizamos esses dados, damos contexto, e extraímos sentido, eles viram informação.

Exemplos:

- Nome
- Idade
- Altura



## Conceitos Básicos

### - BANCO DE DADOS

Um banco de dados é um sistema criado para armazenar, organizar, proteger e recuperar dados de forma eficiente.

Sem banco de dados, todo sistema moderno seria “amnésico”

Na década de 70, a IBM criou o modelo relacional, e junto dele, a linguagem **SQL** — Structured Query Language — que se tornaria o padrão de mercado.

A revolução provocada pela criação do **SQL**, permitiu que dados fossem armazenados em tabelas, conectados por relações lógicas e consultados com precisão.

Mas quando a era da web 2.0 explodiu, lá nos anos 2000, com redes sociais, vídeos e big data, os bancos relacionais começaram a mostrar limites.

Foi nesse cenário que surgiu o movimento **NoSQL**, com novas propostas de armazenar dados de forma mais flexível, distribuída e escalável.



## Conceitos Básicos

### - SGBD ou DBMS

- SGBD (Sistemas de Gerenciamento de Banco de Dados)
- DBMS (Database Management System)

As siglas significam a mesma coisa, porém em idiomas diferentes.

Empresas precisavam lidar com milhões de registros, controlar acesso, garantir segurança, organizar estrutura, evitar perdas e permitir consultas rápidas e confiáveis.

Pense no SGBD como um super gerente.

Ele sabe exatamente onde cada informação está, quem pode acessá-la, quando pode ser alterada, e como garantir que o sistema não quebre mesmo com milhões de acessos simultâneos.

## Exemplos de SGBD

### RELACIONAL

- SQL Server
- PostgreSQL
- Oracle



### NÃO-RELACIONAL

- MongoDB
- Redis
- Cassandra



redis







## RELACIONAL vs NÃO-RELACIONAL

### RELACIONAL

Tudo tem seu lugar, regras rígidas, cada dado mora numa tabela, com colunas bem definidas. Exige que você defina a estrutura antes de colocar qualquer informação

- Consistência forte (ACID)
- Ideal para relacionamentos complexos
- Ferramentas maduras e bem documentadas

### NÃO-RELACIONAL

Não exige um esquema fixo. Você pode jogar dados, documentos, grafos, colunas... como quiser.

A linguagem de consulta depende do banco.

Exemplo: o MongoDB usa uma sintaxe baseada em JavaScript.



# Relacionamento entre tabelas - SQL

Bancos relacionais organizam dados em tabelas que se conectam através de **CHAVES (KEYS)**, criando relações lógicas entre informações.

Vamos explorar os conceitos fundamentais:

- Chave PRIMÁRIA
- Chave ESTRANGEIRA
- Tipos de Relacionamento

# Chave Primária (Primary Key)

É tipo o CPF do dado.

Cada linha tem que ter o seu,  
não pode repetir e nem estar vazio.

```
• CREATE TABLE clientes (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    telefone VARCHAR(20)  
);
```

	id	nome	email	telefone
▶	1	Mayza Ynara	mayza@email.com	1111111
	2	Georgia Cavallaro	georgia@email.com	2222222
	3	Thiago Muniz	thiago@email.com	3333333
	4	Marcio Martinazzo	marcio@email.com	4444444
	5	Bruno Alves	bruno@email.com	5555555
	6	Tupac Shakur	tupac@email.com	6666666
	7	Mano Brown	mano@email.com	7777777
*	NULL	NULL	NULL	NULL

# Chave Estrangeira (Foreign Key)

Um campo numa tabela que aponta para a **PRIMARY KEY** de outra tabela, criando um relacionamento.

```
CREATE TABLE endereco_entrega (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  cliente_id INT UNIQUE NOT NULL, -- Garante o relacionamento 1:1  
  logradouro VARCHAR(200) NOT NULL,  
  numero VARCHAR(20) NOT NULL,  
  complemento VARCHAR(100),  
  cidade VARCHAR(100) NOT NULL,  
  estado CHAR(2) NOT NULL,  
  cep VARCHAR(10) NOT NULL,  
  FOREIGN KEY (cliente_id) REFERENCES CLIENTES(id) ON DELETE CASCADE  
);
```

	id	cliente_id	logradouro
▶	1	1	Avenida Paulista
	2	2	Rua do Ouvidor
	3	3	Avenida Afonso Pena
	4	4	Rua da Praia
	5	5	Avenida Beira Mar
*	NULL	NULL	NULL

# Tipos de relacionamento

Relacionamento 1:N - Um para Muitos

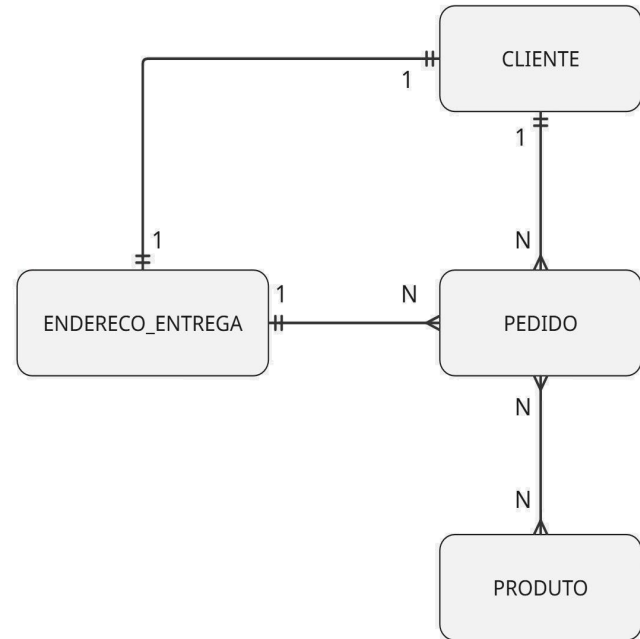
Um **cliente** pode ter vários **pedidos**, mas cada **pedido** pertence a um único **cliente**.

Relacionamento 1:1 - Um para Um

Um **cliente** tem um único cadastro de **endereço de entrega** do pedido.

Relacionamento N:N - Muitos para Muitos

Um **pedido** pode ter vários **produtos** e um **produto** pode estar presente em vários **pedidos**.





# Exercícios de CRUD

## CREATE (Criar)

- Função: Adiciona novos dados ao banco.
- Características:
  - Pode ser em lote (INSERT múltiplo) ou unitário.
  - Geralmente valida regras antes de inserir.

## READ (Ler)

- Função: Recupera dados existentes
- Características:
  - Pode ser simples (SELECT \*) ou complexo (JOIN, WHERE)
  - Não altera os dados originais.

## UPDATE (Atualizar)

- Função: Modifica dados já existentes
- Características:
  - Sempre deve usar WHERE para evitar acidentes.
  - Pode atualizar um ou múltiplos campos.

## DELETE (Excluir)

- Função: Remove registros permanentemente.
- Características
  - Operação irreversível sem backup
  - Em produção, prefira “soft delete” (Marcar como inativo)

**tripleten**



**Q & A**



## THE END

🙏 Obrigado pela atenção!

💬 Qualquer dúvida, só dar um salve!

📞 Meus contatos:

🔗 [LinkedIn](#)

💻 [GitHub](#)

🎮 Discord: @phendges7

🚀 Um dia você percebe que não tem que chegar em lugar nenhum... Você tem que se manter indo!

🟡 Spoiler:

Você conhece Git e GitHub?

Você sabia que o GitHub está diretamente ligado a empregabilidade?

➡ A resposta vem na próxima aula, com a Mayza, fechando com chave de ouro essa missão que nos foi dada e trazendo um surpresinha extra. 😊

Special thanks to Karen Silva for challenging me and improving the quality of this content.