



Melhorando banco de dados

```
Create database cadastro ;
```

Esse comando cria um banco de dados

e serve como ponto de partida para armazenar informações estruturadas. É importante definir tabelas e relações adequadas após a criação do banco de dados.

```
Drop database cadastro;
```

Esse comando apagará o banco de dados.

Deve ser usado com cautela, pois todos os dados contidos serão perdidos permanentemente.

Constraints = parâmetros

```
Not null
```

Esse comando obrigará a digitação de alguma coisa.

No contexto de bancos de dados, NOT NULL é uma **restrição** aplicada a colunas em uma tabela para **garantir que elas não aceitem valores nulos**. Isso significa que, ao inserir ou atualizar um registro, um valor sempre precisa ser fornecido para aquela coluna.

Exemplo:

Imagine uma tabela chamada clientes com as seguintes colunas:

```
id: INT , chave primária
```

```
nome:
VARCHAR ( 100 ) - com restrição NOT NULL
email:
VARCHAR ( 100 ) - sem restrição NOT NULL
```

Aqui, a coluna nome foi definida com a restrição NOT NULL, então não é permitido deixar essa coluna vazia. Já a coluna email, por não ter essa restrição, pode receber valores nulos, ou seja, pode ficar em branco em alguns registros.

Por que usar NOT NULL?

Integridade dos dados:

garante que informações essenciais estejam sempre preenchidas.

Evita erros:

ao não permitir nulos, reduz-se a chance de erros futuros em consultas ou operações.

Melhor desempenho: em alguns sistemas, colunas NOT NULL podem ter um desempenho levemente melhor.

Para aplicar essa restrição, no SQL, você faria algo assim:

```
CREATE TABLE clientes (
  id INT PRIMARY KEY ,
  nome VARCHAR (100) NOT NULL ,
  email VARCHAR (100));
```

Aqui, todo cliente precisará ter um nome, mas o email é opcional.

padrão

o comando `default` permite que se não for digitado nada por padrão será X coisa