

# SGBD

Mesmo havendo outros modelos de Db, o modelo Relacional é o mais utilizado.

- SGBDR ( RDBMS ) Sistema de Banco de Dados Relacionais, composto por entidades, chamadas de tabelas, que agrupam conjunto de informações, e as linhas onde ficam essas informações são chamadas de registros ou tuplas
  - Cada linha precisa ter uma classificação, no caso, uma coluna.
  - É fundamental em um SGBDR ter uma Chave Primária e Chave Estrangeira.

Antes de criar um Db relacional é preciso criar um modelo que pode ser:

## 1. Modelo conceitual - MER ( Modelo Entidade Relacionamento )

- É uma abstração de uma tabela de dados.
- Ele pode ser representado por meio do DER ( Diagrama Entidade Relacionamento ). Nele mostra as entidades e o relacionamento entre elas.

## 2. Modelo Lógico - Implementação

- 
- Entidades Fortes são entidades que não dependem de outra para existir.
  - Entidade Fraca já são entidades que dependem de outra para existir.
  - No lado da entidade mostra a relação entre uma entidade com outra.

Existem 3 graus de relacionamento:

1. Relacionamento de Um para Um (1:1)
2. Relacionamento de Um para Muitos ou Muitos para Um (1:N)
3. Relacionamento de Muitos para Muito (N:M)

**Obs: O relacionamento ( cardinalidade ) é lido da entidade que se refere a cardinalidade para a outra.**

- O relacionamento assume o estado de Entidade Associativa caso as Entidades A e B tenham uma relação de N:M

---

Normalização é o processo de análise de uma entidade para organizar ela e reduzir redundância.

- Existem 6 formas no processo de normalização, mas, a 1,2 e 3° formas são as mais usadas. As outras são mais específicas e exigem mais processamento. As formas são sequenciais, sendo assim, se está na 3° forma então já passou pela 1° e 2°.

## 1° Forma Normal:

- Possui chave primária.
- Não possui grupos repetitivos.
- Todos os atributos são atômicos, não precisa ser decomposto em outras tabelas.

Sendo assim, determinar a chave primária, tornar os atributos atômicos, e transformar o grupo repetitivo em uma nova tabela criando assim uma ligação entre ele e a tabela que estava antes.

✗ Tabela (cod\_cliente, nome\_cliente, tel1, tel2, endereco, cod\_produto, nome\_produto, preco, quantidade)

✓ Cliente (cod\_cliente, nome\_cliente, tel1, tel2, rua, bairro, cidade, estado)

✓ Produto (cod\_cliente, cod\_produto, nome\_produto, preco, quantidade)

## 2° Forma Normal:

- Está na 1° forma.
- Não possui dependências parciais na primary key.

Se a chave primária da tabela é simples já se encontra na 2° forma normal. Se for composta deve-se verificar se todos os atributos da relação dependem dos atributos que compõem a chave primária.

✓ Cliente (cod\_cliente, nome\_cliente, tel1, tel2, rua, bairro, cidade, estado)

(não possui chave primária composta)

✗ Produto (cod\_cliente, cod\_produto, nome\_produto, preco, quantidade)

cod\_produto -> nome\_produto, preco (dependência parcial)

cod\_cliente, cod\_produto -> quantidade (dependência total)

✓ Resp: Produto (cod\_produto, nome\_produto, preco)

✓ Resp: Compra (cod\_cliente, cod\_produto, quantidade)

Por criar uma única tabela para a quantidade, isso não vai acabar agravando os outros atributos de Produto.

**Obs: O conceito de Entidade Associativa se aplica aqui**

3° Forma Normal:

- Está na 2° forma
- Os atributos não chave não dependem de outro atributo não chave.

✗ Carro (placa, modelo, km\_rodados, cod\_fabricante, nome\_fabricante)

Placa, modelo → km\_rodados

Placa, modelo → cod\_fabricante

Placa, modelo → nome\_fabricante

Cod\_fabricante → nome\_fabricante

✓ Carro (placa, modelo, kmRodados, cod\_fabricante) ✓

✓ Fabricante (cod\_fabricante, nome\_fabricante)