

# Modelagem de Banco de Dados

---

CHAVE ESTRANGEIRA



# Tópicos

Definição;

Compartilhamento de Dados;

Sistemas de Gerenciamento de Banco de Dados;

Modelos de Banco de Dados

Modelo conceitual

Modelo lógico



## Tipos de chaves

- Chave primária;
- **Chave estrangeira;**
- Chave alternativa.

# O que são Formas Normais?

---

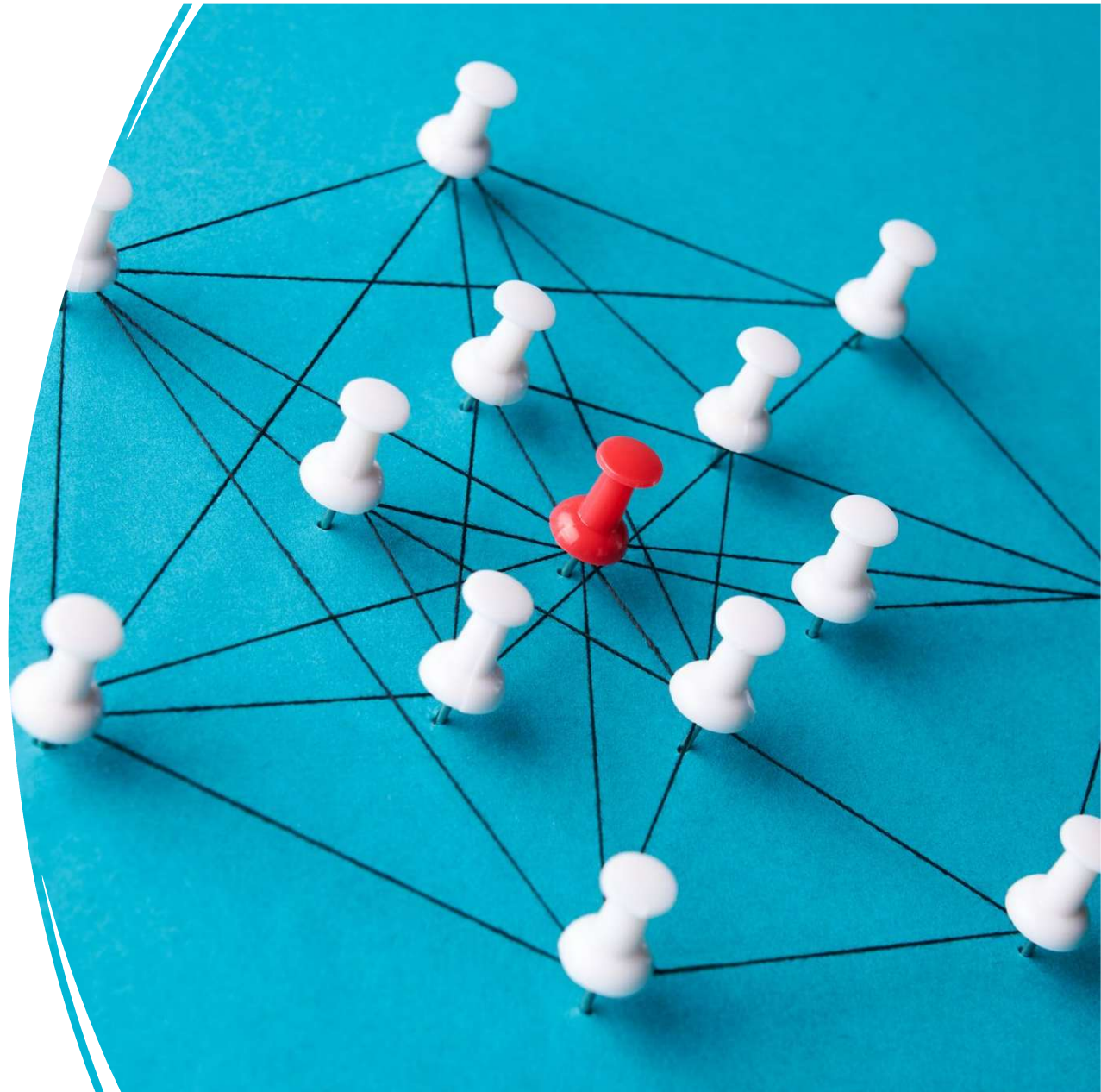
As formas normais são regras que ajudam a estruturar um banco de dados relacional para reduzir a redundância e melhorar a integridade dos dados.



# Objetivos:

---

- Eliminar dados duplicados.
- Garantir a dependência funcional.
- Facilitar a manutenção e a atualização dos dados.



# Principais Formas Normais:

---

Primeira  
Forma  
Normal (1FN)

Segunda  
Forma  
Normal (2FN)

Terceira  
Forma  
Normal (3FN)



## Primeira Forma Normal (1NF)

Definição:

Uma tabela está na 1NF se:

1. Todos os atributos contêm **valores atômicos** (indivisíveis).
2. Cada coluna contém valores do **mesmo tipo**.
3. Cada linha é **única** (não pode haver duplicatas).
4. A ordem dos dados não deve ter importância.

## Tabela não normalizada:

empregado

id	nome	telefone
1	Ana	999991234, 999991111
2	Maria	999997777
3	Pedro	988884567
4	José	988115566

## Tabela normalizada:

empregado

id	nome	telefone
1	Ana	999991234
2	Ana	999991111
3	Maria	999997777
4	Pedro	988884567
5	José	988115566





## Segunda Forma Normal (2NF)

Definição:

Uma tabela está na 2NF se:

1. Está na 1NF.
2. Todos os atributos não-chave são totalmente dependentes da chave primária (sem dependências parciais).

## Tabela na 1FN :

empregado

Id_empregado	nome	telefone
1	Ana	999991234
2	Ana	999991111
3	Maria	999997777
4	Pedro	988884567
5	José	988115566

## Tabela na 2FN :

empregado

id_empregado	nome
1	Ana
2	Maria
3	Pedro
4	José

telefone

id_telefone	id_empregado	numero
1	1	999991234
2	1	999991111
3	2	999997777
4	3	988884567
5	4	988115566



## Terceira Forma Normal (3NF)

Definição:

Uma tabela está na 3NF se:

1. Está na 2NF.
2. Não há dependências transitivas (um atributo não-chave não deve depender de outro atributo não-chave).



## Chave estrangeira

Uma chave estrangeira é uma coluna ou uma combinação de colunas, cujos valores aparecem necessariamente na chave primária de uma tabela.



# Tabela na forma incorreta

empregado

id	nome	sexo	cpf	telefone	endereco
1	Ana	F	12345678900	999991234	Av. Brasil, 1234
2	Maria	F	45678912344	999997777	Rua São Paulo, 100
3	Pedro	M	78912345677	988884567	Rua Três, 4567
4	José	M	98765432199	988115566	Av. Tiradentes, 5577

# Utilizando as formas normais:

empregado

id	nome	sexo	cpf	telefone	endereço
1	Ana	F	12345678900	999991234	Av. Brasil, 1234
2	Maria	F	45678912344	999997777	Rua São Paulo, 100
3	Pedro	M	78912345677	988884567	Rua Três, 4567
4	José	M	98765432199	988115566	Av. Tiradentes, 5577

empregado

id_empregado	nome	sexo	cpf
--------------	------	------	-----

# Utilizando as formas normais:

empregado

id	nome	sexo	cpf	telefone	endereco
1	Ana	F	12345678900	999991234	Av. Brasil, 1234
2	Maria	F	45678912344	999997777	Rua São Paulo, 100
3	Pedro	M	78912345677	988884567	Rua Três, 4567
4	José	M	98765432199	988115566	Av. Tiradentes, 5577

empregado

id_empregado	nome	sexo	cpf
--------------	------	------	-----

telefone

id_telefone	id_empregado	tipo	numero
-------------	--------------	------	--------

# Utilizando as formas normais:

empregado

id	nome	sexo	cpf	telefone	endereço
1	Ana	F	12345678900	999991234	Av. Brasil, 1234
2	Maria	F	45678912344	999997777	Rua São Paulo, 100
3	Pedro	M	78912345677	988884567	Rua Três, 4567
4	José	M	98765432199	988115566	Av. Tiradentes, 5577

empregado

id_empregado	nome	sexo	cpf
--------------	------	------	-----

telefone

id_telefone	id_empregado	tipo	numero
-------------	--------------	------	--------

endereço

id_endereco	id_empregado	logradouro	numero	bairro	cidade
-------------	--------------	------------	--------	--------	--------



# Principais Formas Normais:

empregado

id_empregado	nome	sexo	cpf
1	Ana	F	12345678900
2	Maria	F	45678912344
3	Pedro	M	78912345677
4	José	M	98765432199

telefone

id_telefone	id_empregado	tipo	numero
1	1	Celular	999991234
2	2	Celular	999997777
3	3	Celular	988884567
4	4	Celular	988115566
5	2	Residencial	3722-2222

endereco

id_endereco	id_empregado	logradouro	numero	cidade	estado
1	1	Av. Brasil	1234	Franca	SP
2	2	Rua São Paulo	100	Franca	SP
3	3	Rua Três	4567	Franca	SP
4	4	Av. Tiradentes	5577	Franca	SP



Abrindo o arquivo:

**modelagemBasica3.sql**

Copie o arquivo do **Teams**  
e abra em um editor de  
sua preferência.

# Alterando as tabelas para as formas normais

```
/* Criando a tabela cliente */  
CREATE TABLE pessoa(  
    id_pessoa INT auto_increment PRIMARY KEY,  
    nome VARCHAR(30),  
    genero CHAR(1),  
    cpf VARCHAR(11) unique  
);
```

# Alterando as tabelas para as formas normais

```
/* Criando a tabela telefone */  
CREATE TABLE telefone(  
    id_telefone INT auto_increment PRIMARY KEY,  
    id_pessoa INT,  
    tipo VARCHAR(20),  
    numero VARCHAR(12),  
    FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa)  
);
```

# Alterando as tabelas para as formas normais

```
/* Criando a tabela endereco */  
CREATE TABLE endereco(  
    id_endereco INT auto_increment PRIMARY KEY,  
    id_pessoa INT,  
    logradouro VARCHAR(100),  
    numero INT,  
    bairro VARCHAR(50),  
    cidade VARCHAR(50),  
    estado CHAR(2),  
    FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa)  
);
```

# Selecionando dados das tabelas

```
/* Função inner join */
SELECT pessoa.nome, endereco.logradouro, endereco.numero,
endereco.bairro, endereco.cidade, endereco.estado
FROM pessoa
INNER JOIN endereco ON pessoa.id_pessoa = endereco.id_pessoa;

/* utilizando alias */
SELECT p.nome, e.logradouro, e.numero, e.bairro, e.cidade,
e.estado
FROM pessoa p
INNER JOIN endereco e ON p.id_pessoa = e.id_pessoa;
```

# Selecionando dados das tabelas

```
/* Utilizando USING no lugar de ON
   Somente utilizado quando o campo possui o mesmo nome nas
   duas tabelas.
*/
SELECT p.nome, e.logradouro, e.numero, e.bairro, e.cidade,
e.estado
FROM pessoa p
INNER JOIN endereco e USING(id_pessoa);
```

# Selecionando dados das tabelas

```
/* Selecionar o nome, tipo de telefone, numero do telefone,  
logradouro, numero, bairro, cidade e estado */  
SELECT p.nome, t.tipo, t.numero, e.logradouro, e.numero,  
e.bairro, e.cidade, e.estado  
FROM pessoa p  
INNER JOIN telefone t ON p.id_pessoa = t.id_pessoa  
INNER JOIN endereco e ON p.id_pessoa = e.id_pessoa;
```



# Selecionando dados das tabelas

```
/* Selecionar os dados de toda tabela da esquerda e os dados  
da tabela da direita que tem correspondência */  
SELECT p.nome, p.cpf, t.tipo, t.numero  
FROM pessoa p  
LEFT JOIN telefone t ON p.id_pessoa = t.id_pessoa;
```

```
/* Selecionar os dados de toda tabela da direita e os dados  
da tabela da esquerda que tem correspondência */  
SELECT p.nome, p.cpf, t.tipo, t.numero  
FROM pessoa p  
RIGHT JOIN telefone t ON p.id_pessoa = t.id_pessoa;
```

# Inserindo dados das tabelas

```
/* Inserindo dados nas tabelas */
START TRANSACTION;
INSERT INTO pessoa (nome, genero, cpf) VALUES ('Pedro', 'M',
'31300002111');
SET @id_pessoa = LAST_INSERT_ID();
INSERT INTO telefone (id_pessoa, tipo, numero) VALUES
(@id_pessoa, 'Celular', '+977775555');
INSERT INTO endereco (id_pessoa, logradouro, numero, bairro,
cidade, estado) VALUES (@id_pessoa, 'Rua São Paulo', 1234,
'Cidade Nova', 'Franca', 'SP');
COMMIT;
```

# Exercício:

Copiar o arquivo  
“**exercicio4.sql**” no  
TEAMS e fazer o  
passo a passo.

