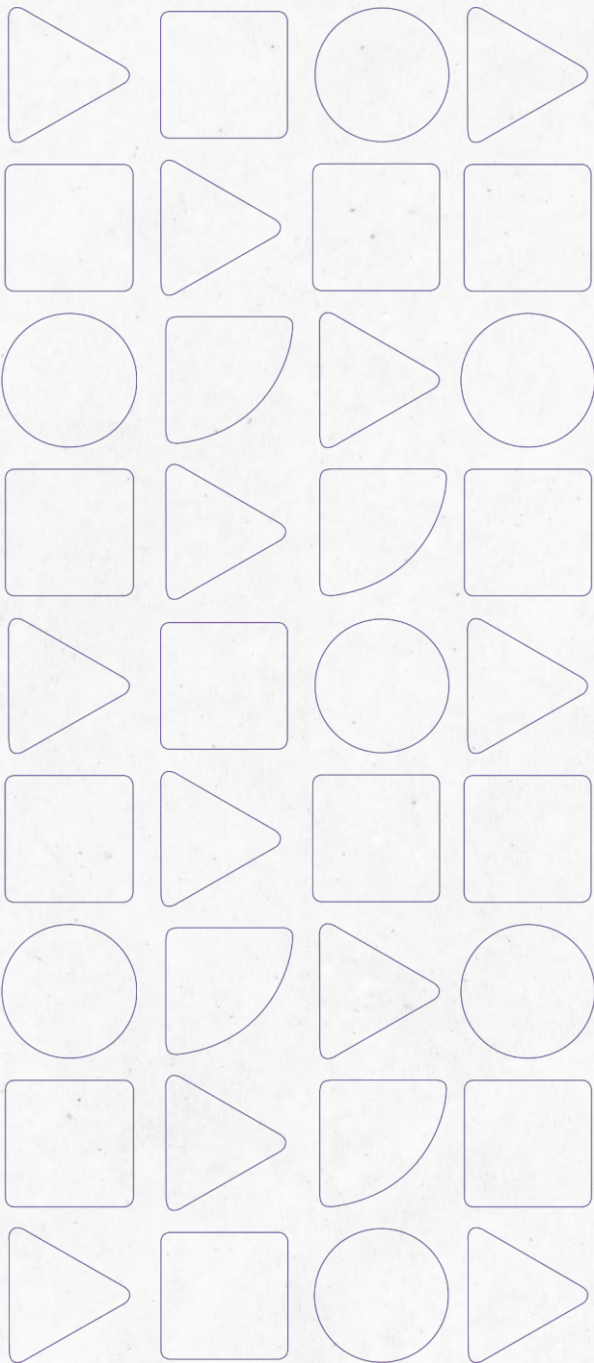


Modelagem Banco de Dados

**Disciplina:** Banco de Dados



## Conteúdos:

Modelagem de Banco de Dados

## Habilidade(s):

- Cardinalidade Banco de Dados.
- Práticas de Modelagem de Dados.

Como você está se sentindo hoje?



1



2



3



4



5



6

- Praticamente tudo o que vamos fazer na vida exige um planejamento, ainda que ele não seja visível, mas até de forma implícita fazemos planejamento o tempo todo.
- Sua finalidade é garantir que o resultado atenda a um propósito inicial e, portanto, precisamos definir as bases, o caminho, os recursos e diversos outros aspectos para que obtenhamos sucesso naquilo que estamos prestes a fazer.
- Com banco de dados a história não é diferente e nesse aspecto é que entra um termo conhecido como modelagem de dados.

**Mais afinal...**

**O que é Modelagem de  
Dados?**



## O QUE É MODELAGEM DE DADOS?



A **modelagem de dados** é uma técnica usada para a especificação das regras de negócios e as estruturas de dados de um banco de dados. Ela faz parte do ciclo de desenvolvimento de um sistema de informação e é de vital importância para o bom resultado do projeto. Modelar dados consiste em desenhar o sistema de informações, concentrando-se nas entidades lógicas e nas dependências lógicas entre essas entidades.

**Modelagem de dados** ou modelagem de banco de dados envolve uma série de aplicações teóricas e práticas, visando construir um modelo de dados consistente, não redundante e perfeitamente aplicável em qualquer SGBD moderno.

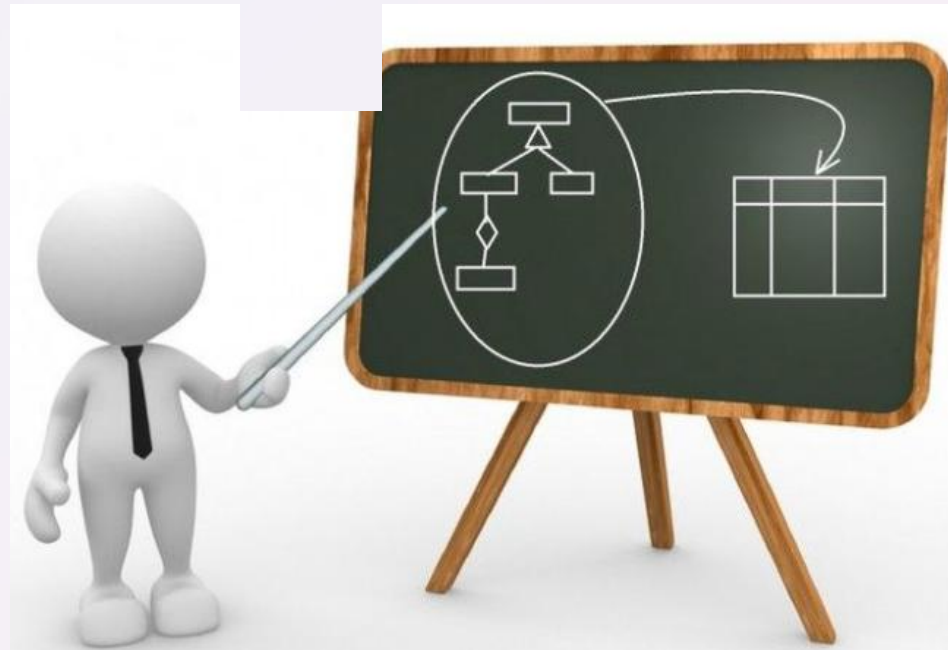


# Modelagem de Dados - Níveis

Classificamos o processo de modelagem de dados em três níveis:

- Modelo Conceitual (alto nível) - Mais próximo da abstração  
MCD
- Modelo Lógico - MLD
- Modelo Físico (baixo nível) - MFD Mais próximo do BD

# MODELO CONCEITUAL, LÓGICO E FÍSICO DE BANCO DE DADOS





# Modelo Conceitual

Esta é a primeira fase da modelagem, onde representaremos o mundo real por meio de uma visão simplificada dos dados e seus relacionamentos. Assim poderemos determinar quais informações serão armazenadas no BD.

Neste nível o projeto é independente de SGBD

# Modelo Conceitual

Neste nível, detalhes da implementação não aparecem, porém é suficientemente detalhado para a ponto de ser possível descrever os tipos de dados requeridos, seus relacionamentos entre si e regras de consistência.

Podemos saber o tipo de informação que se refere por exemplo: a um nome do produto, a quantidade, ao preço do produto e a outras informações que serão recolhidas.

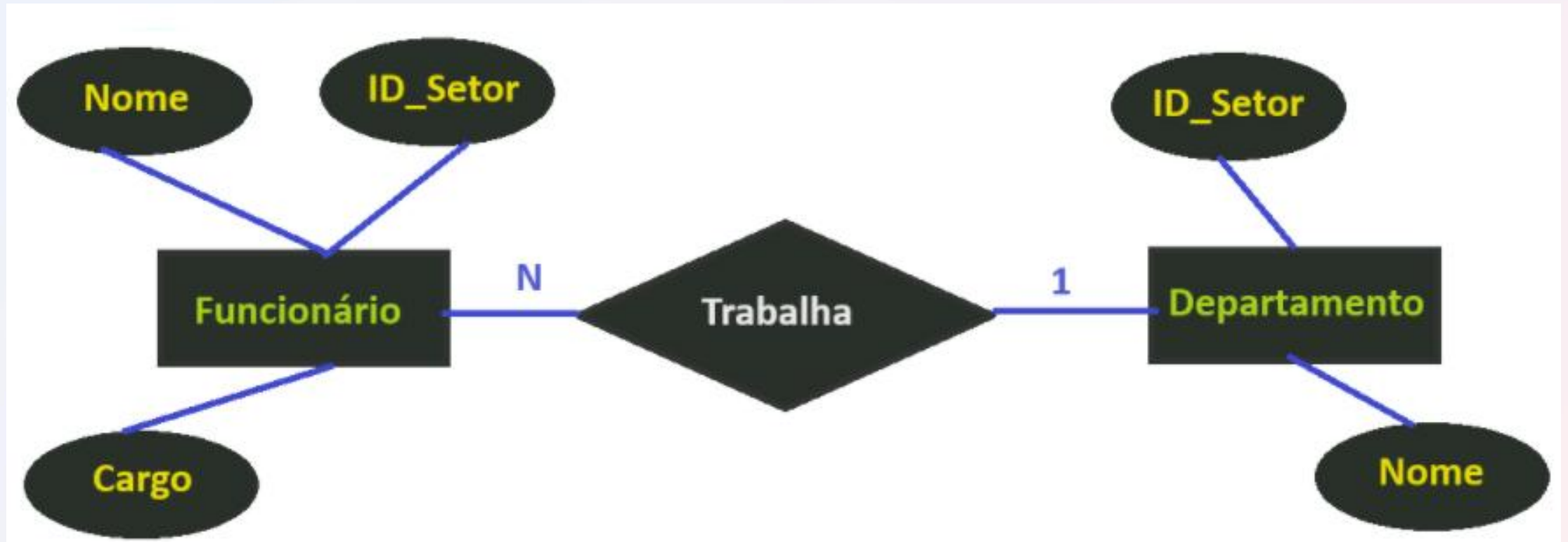
# Modelo Conceitual

- Baseia-se no mais alto nível e deve ser usada para envolver o cliente, pois o foco aqui é discutir os aspectos do negócio do cliente e não da tecnologia.
- Os exemplos de modelagem de dados vistos pelo modelo conceitual são mais fáceis de compreender, já que não há limitações ou aplicação de tecnologia específica.
- O diagrama de dados que deve ser construído aqui é o DER - DIAGRAMA DE ENTIDADE E RELACIONAMENTO, onde deverão ser identificados todas as entidades e os relacionamentos entre elas. Este diagrama é a chave para a compreensão do modelo conceitual de dados.
- Exemplo simples de Diagrama de Entidade e Relacionamento

# Modelo Conceitual

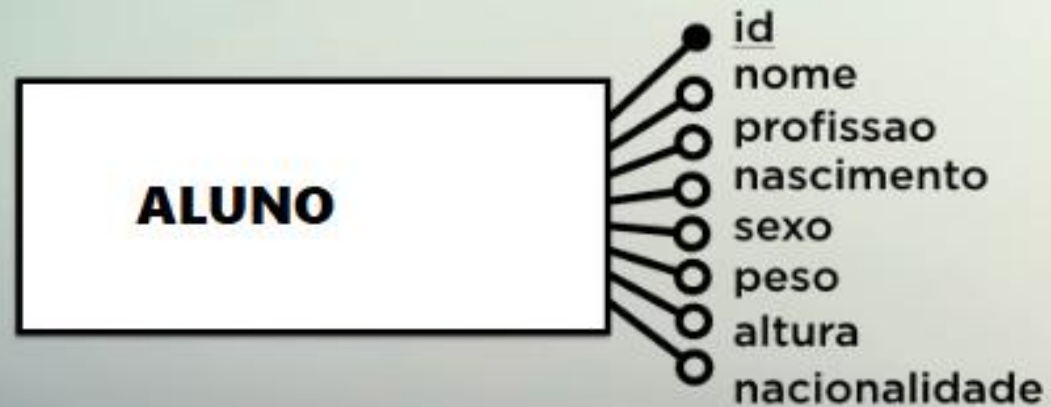


# Modelo Conceitual



# Modelo Conceitual

Tabelas:  
Entidade

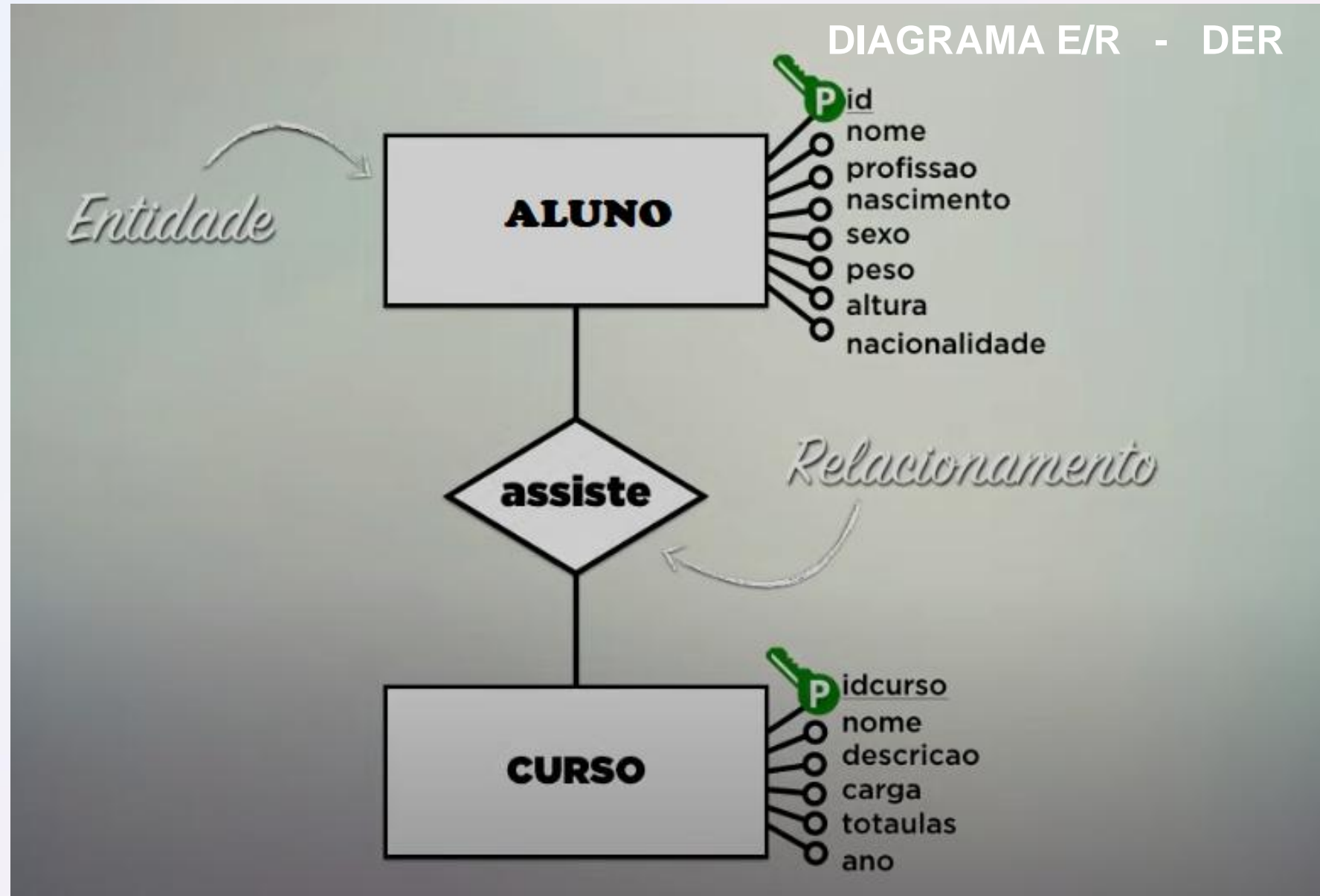


Dados: Atribulos



# Modelo Conceitual

Modelo gráfico  
de como  
o BD vai funcionar



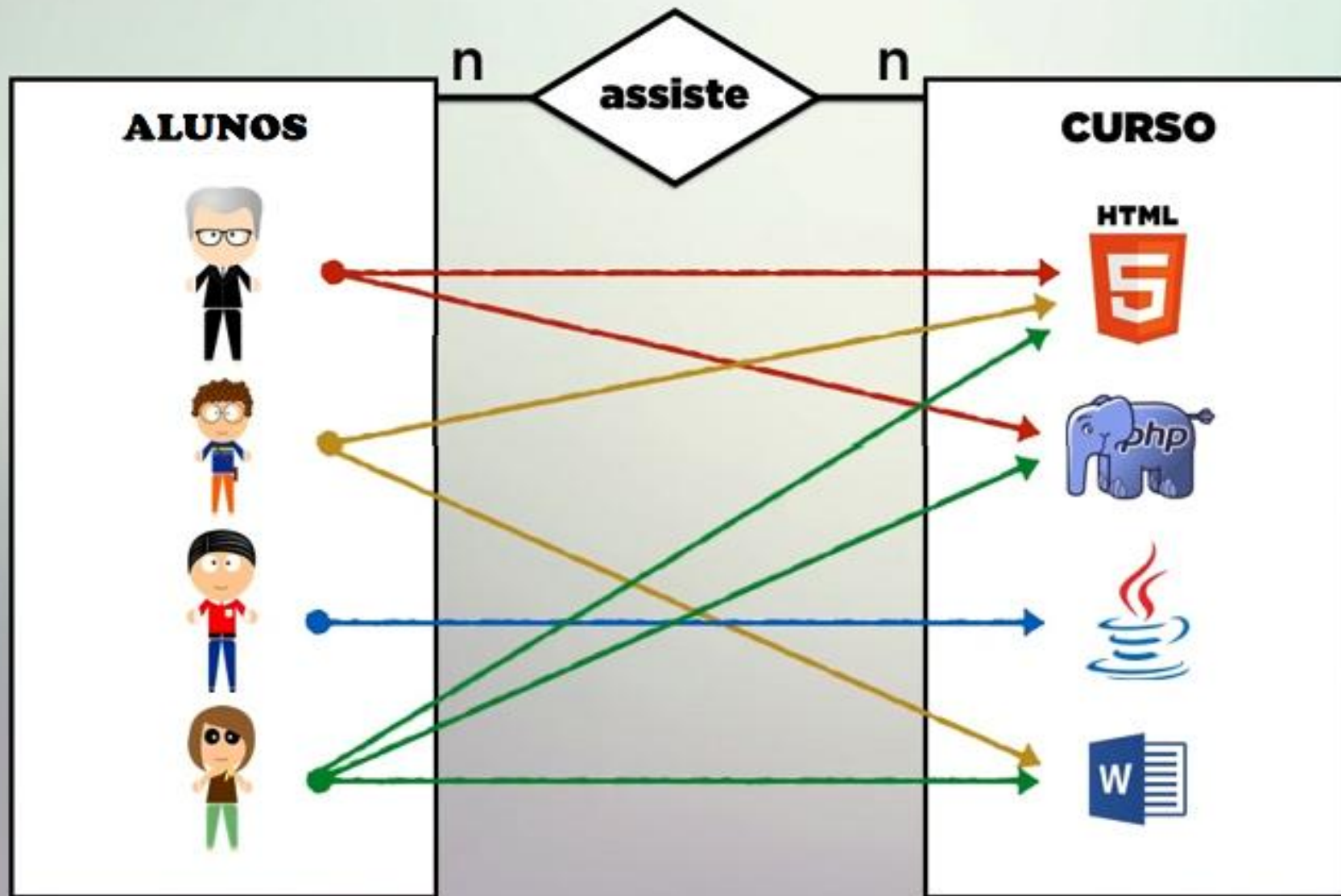
# Cardinalidade

**Cardinalidade**  
**muitos-para-muitos**

Simple ou múltiplas  
Máxima: 1 ou N



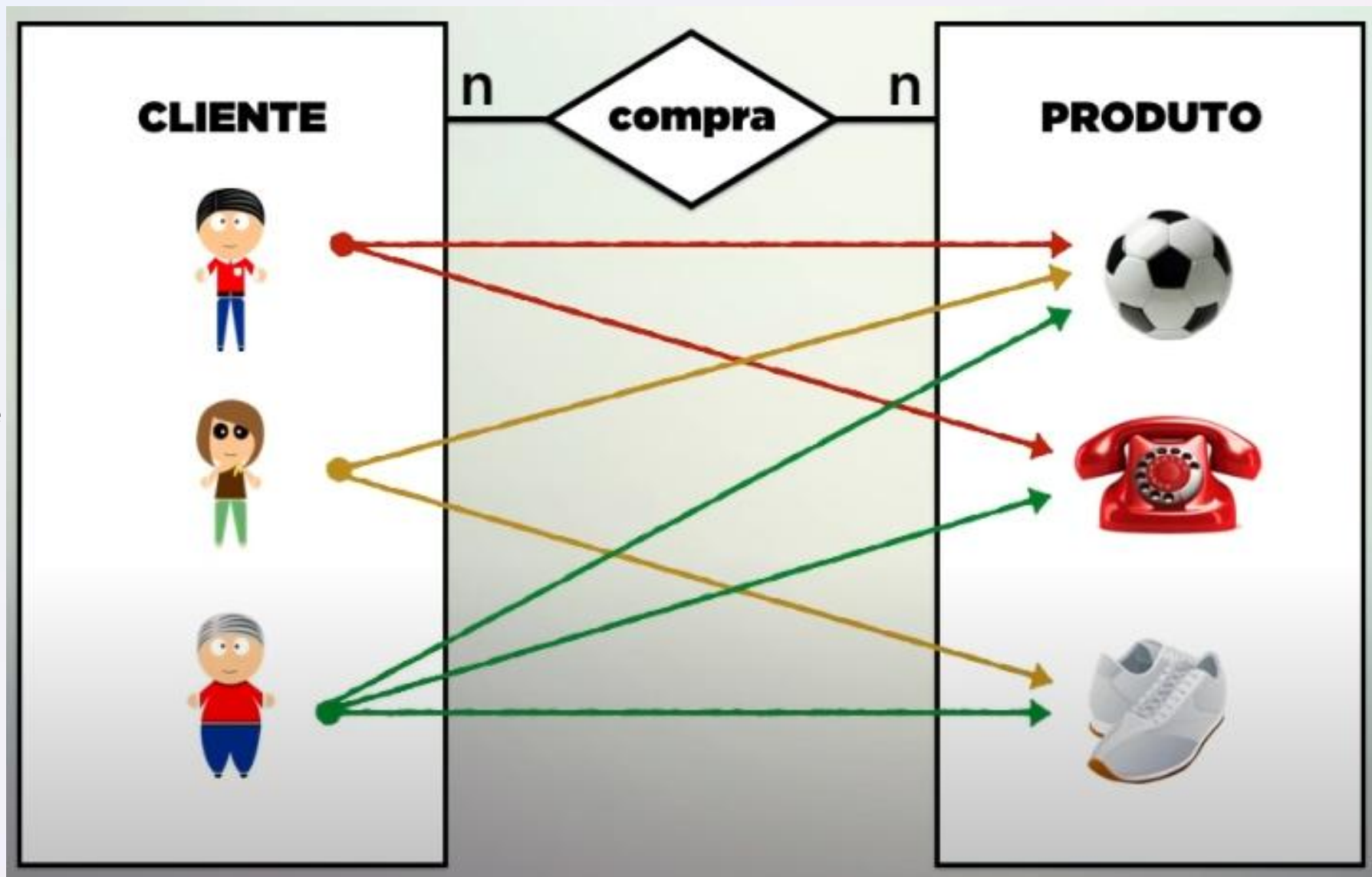
Assistem  
quantos cursos?



Cursos são  
Assistidos  
por quantos?

# Cardinalidade

muitos-para-muitos

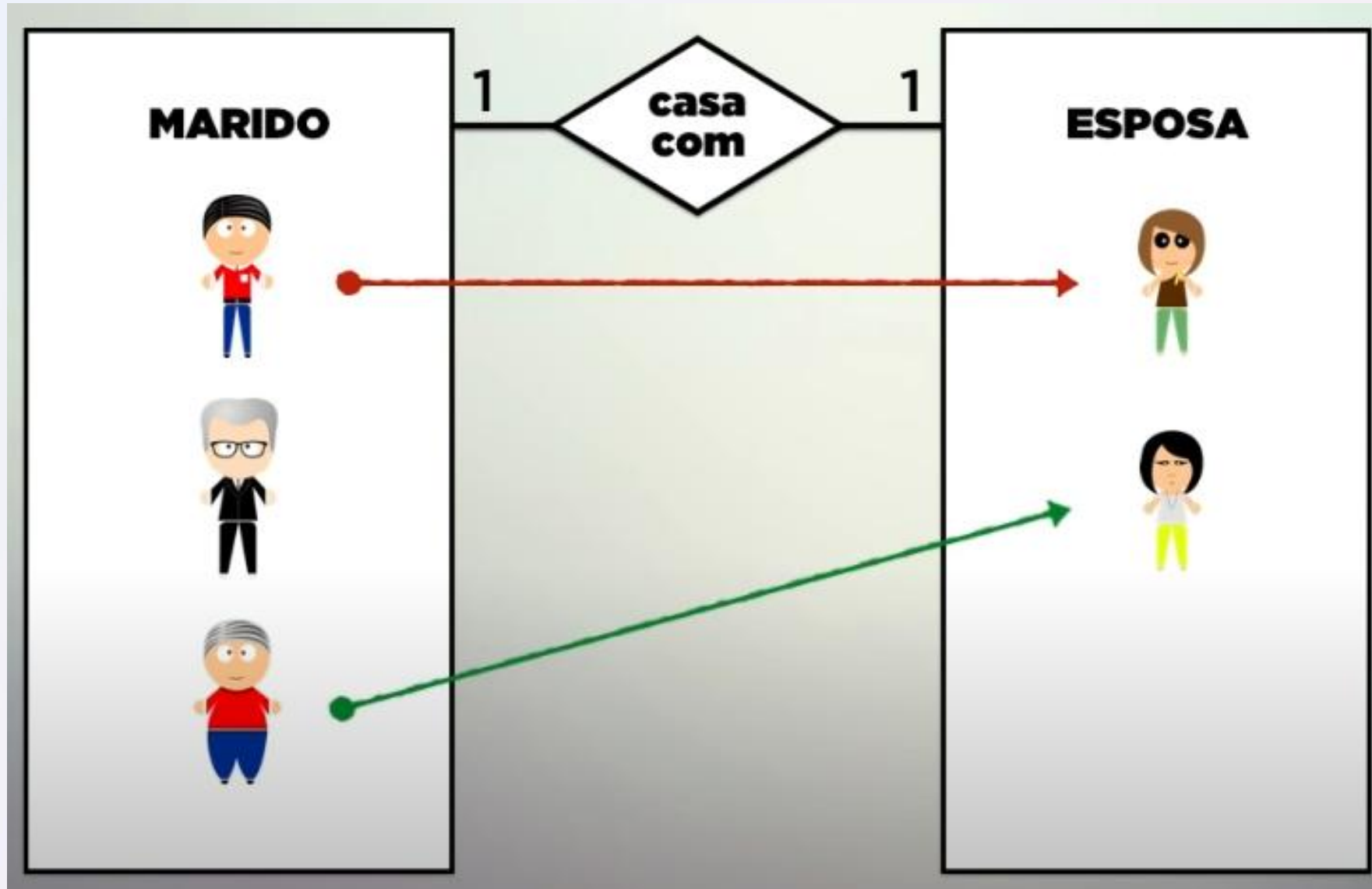


Um cliente  
Compra  
vários  
produtos

Um  
Produto  
pode ser  
comprado  
Por vários  
Clientes.

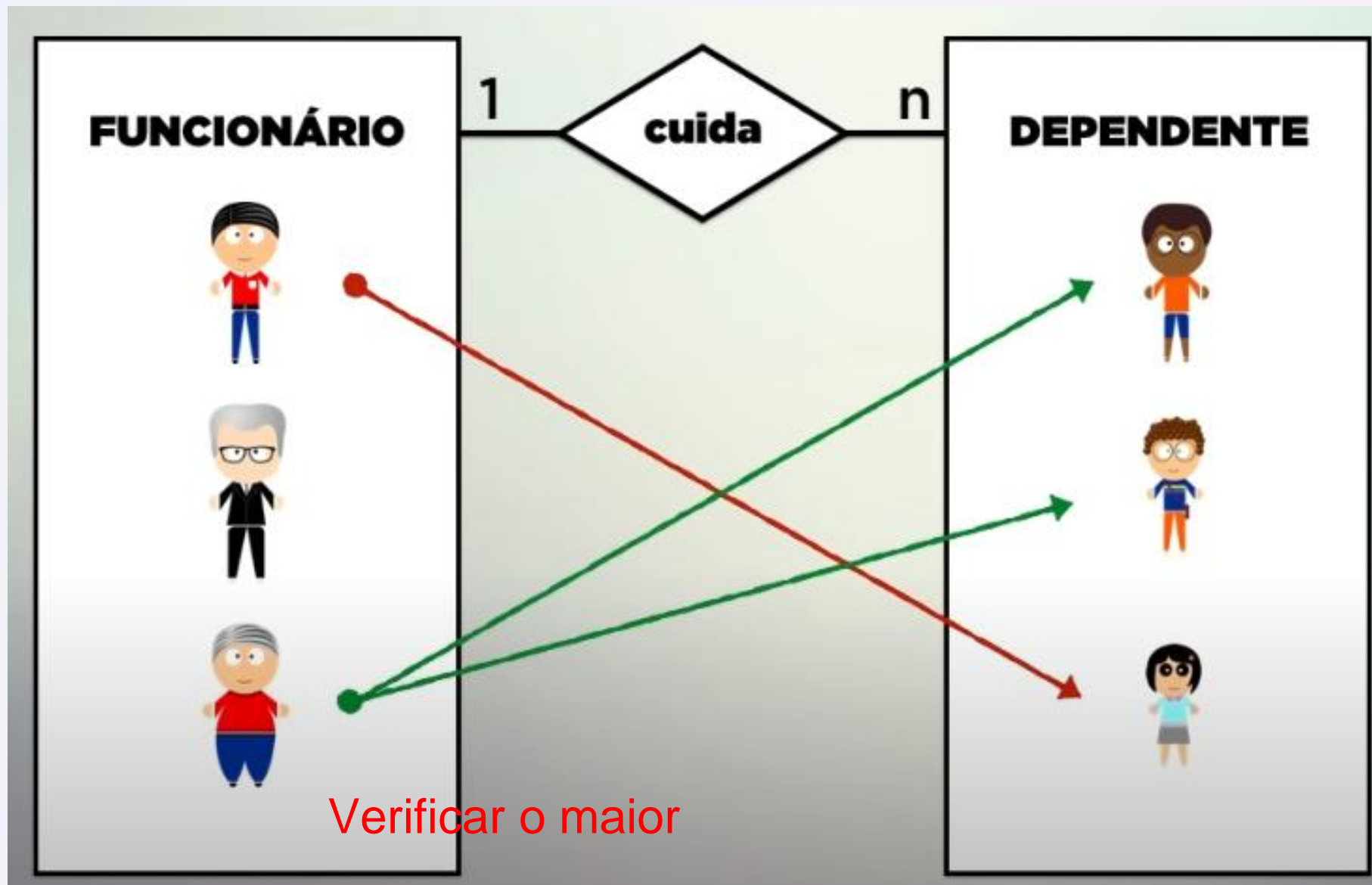
# Cardinalidade

um-para-um



# Cardinalidade

um-para-muitos





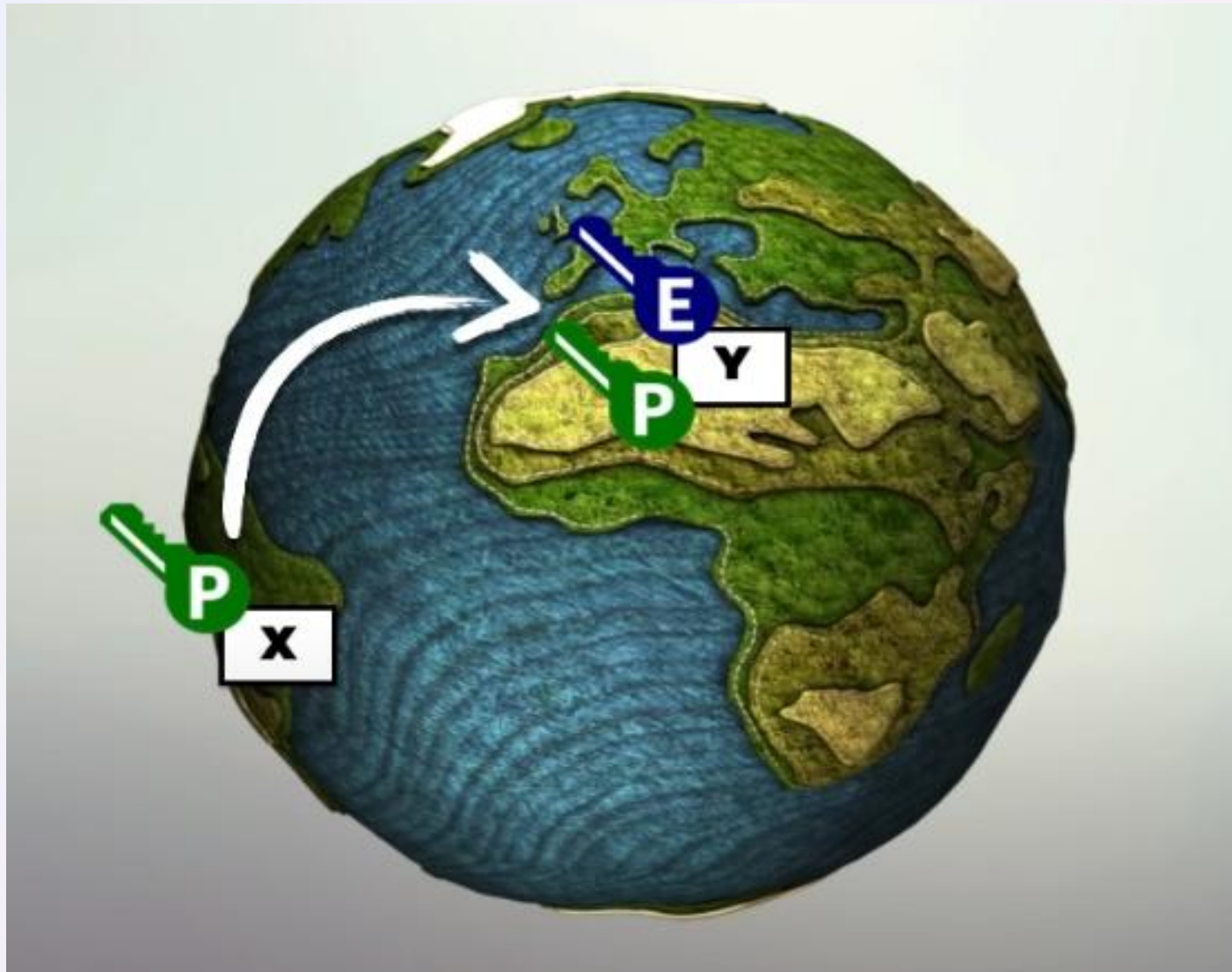
## **Chave Primária**

Atributo específico que identifica as tuplas. Ele é um identificador. Vai fazer com que nenhum atributo seja igual ao outro.  
Ex: Código, CPF, Código de Barra.



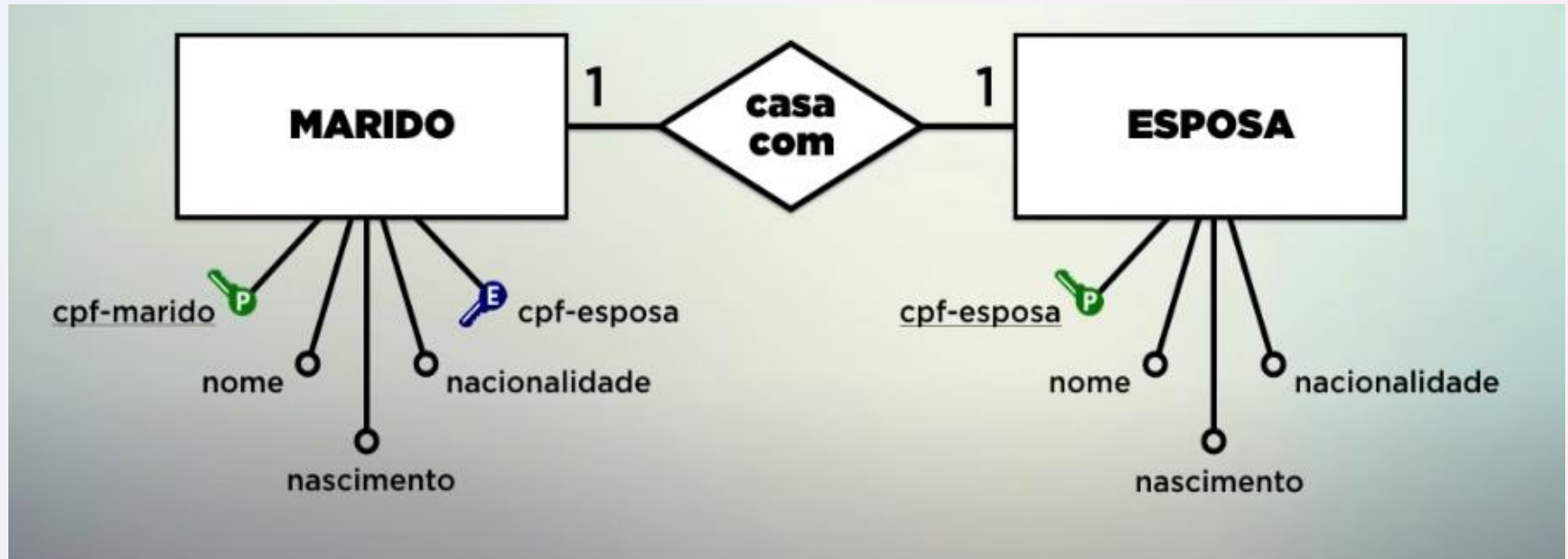


**Chave Estrangeira**  
chave primária que  
veio de outra  
entidade para  
relacionar



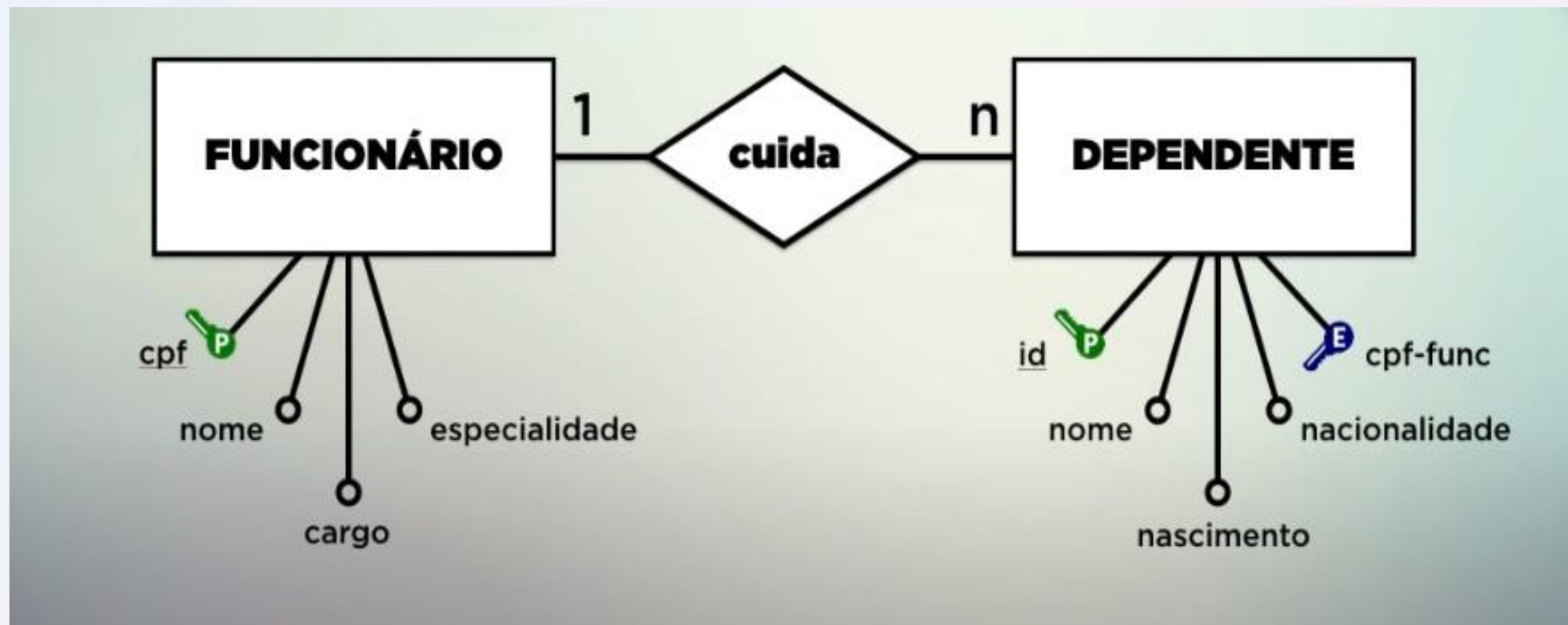
Relacionamento entre tabelas, troca de chaves

Entidade dominante fica a direita

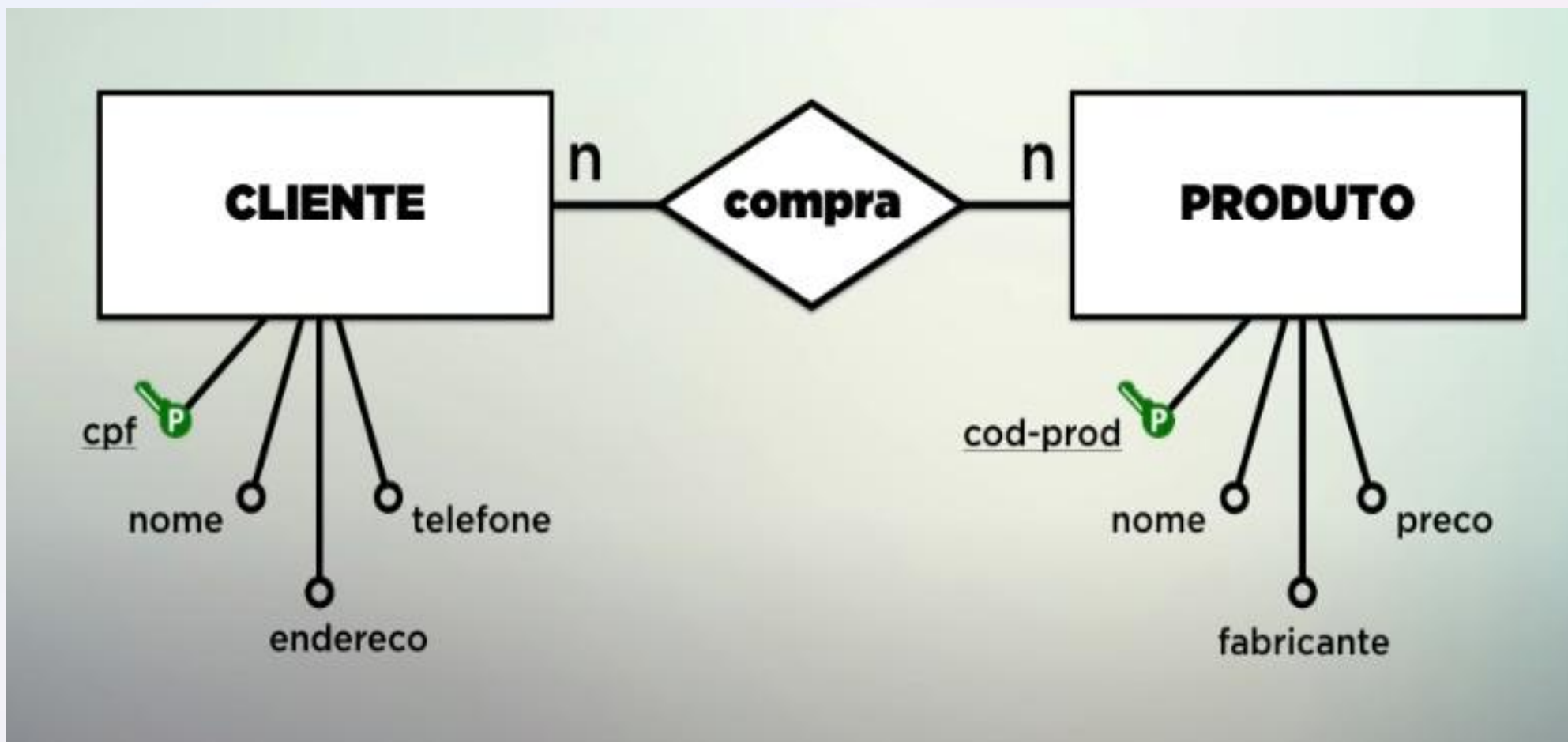


Pega a chave primária da não dominante e coloca na dominante

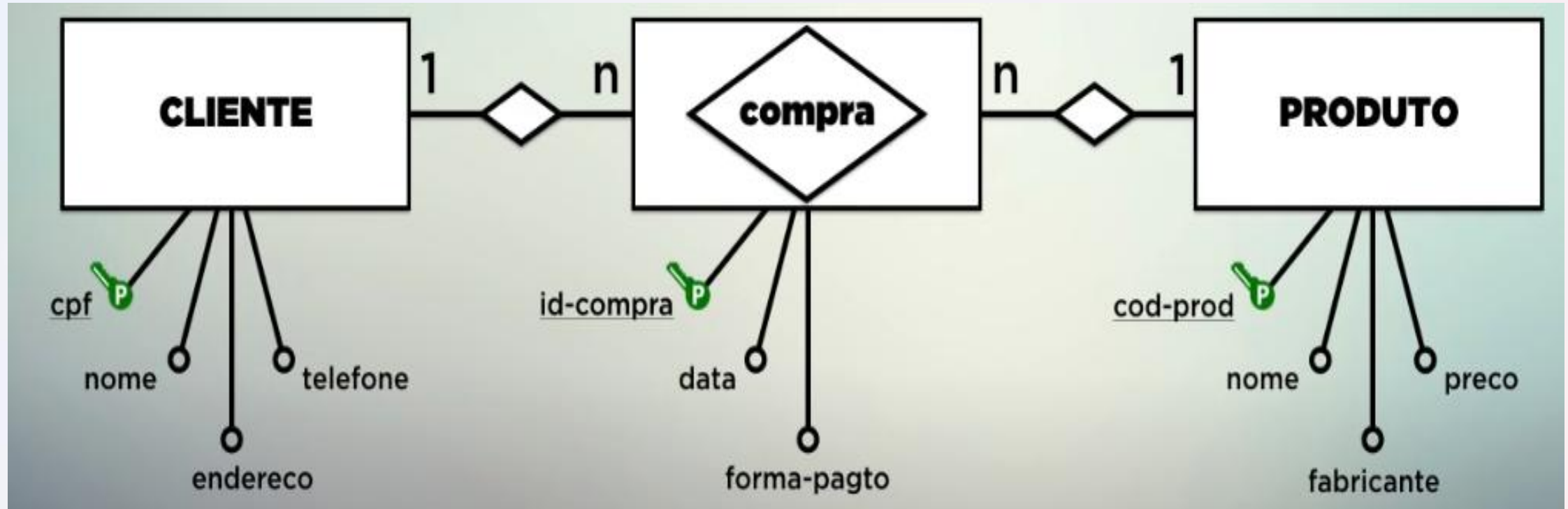
Pegar a chave da entidade 1 e coloca na entidade muitos



O relacionamento cria nova entidade

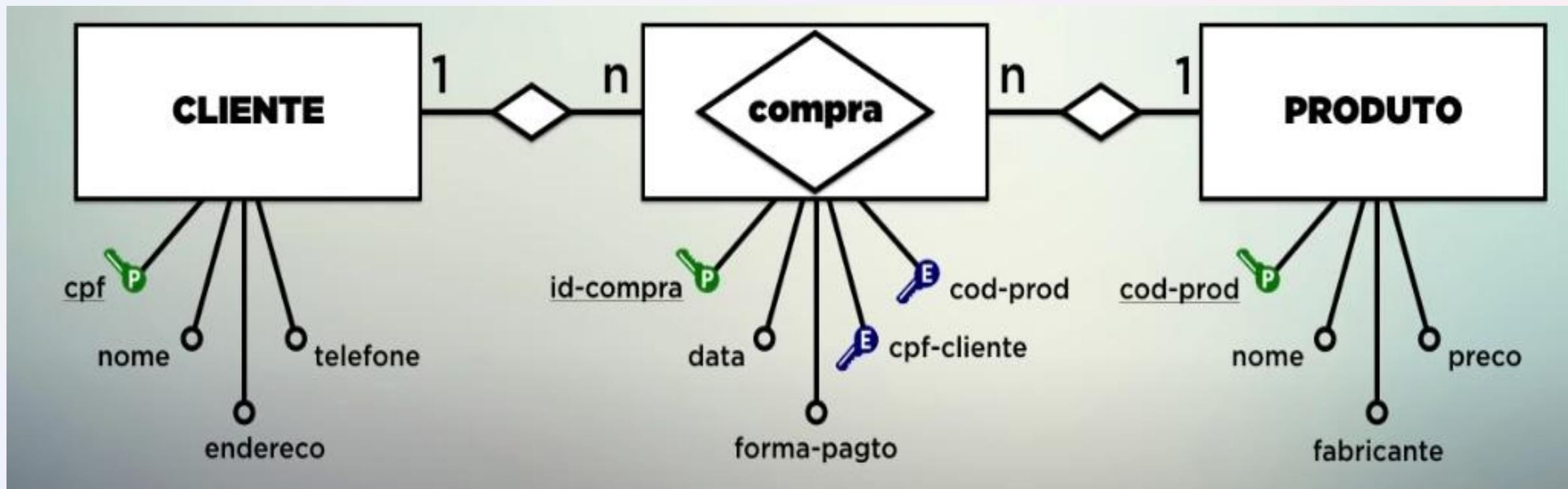


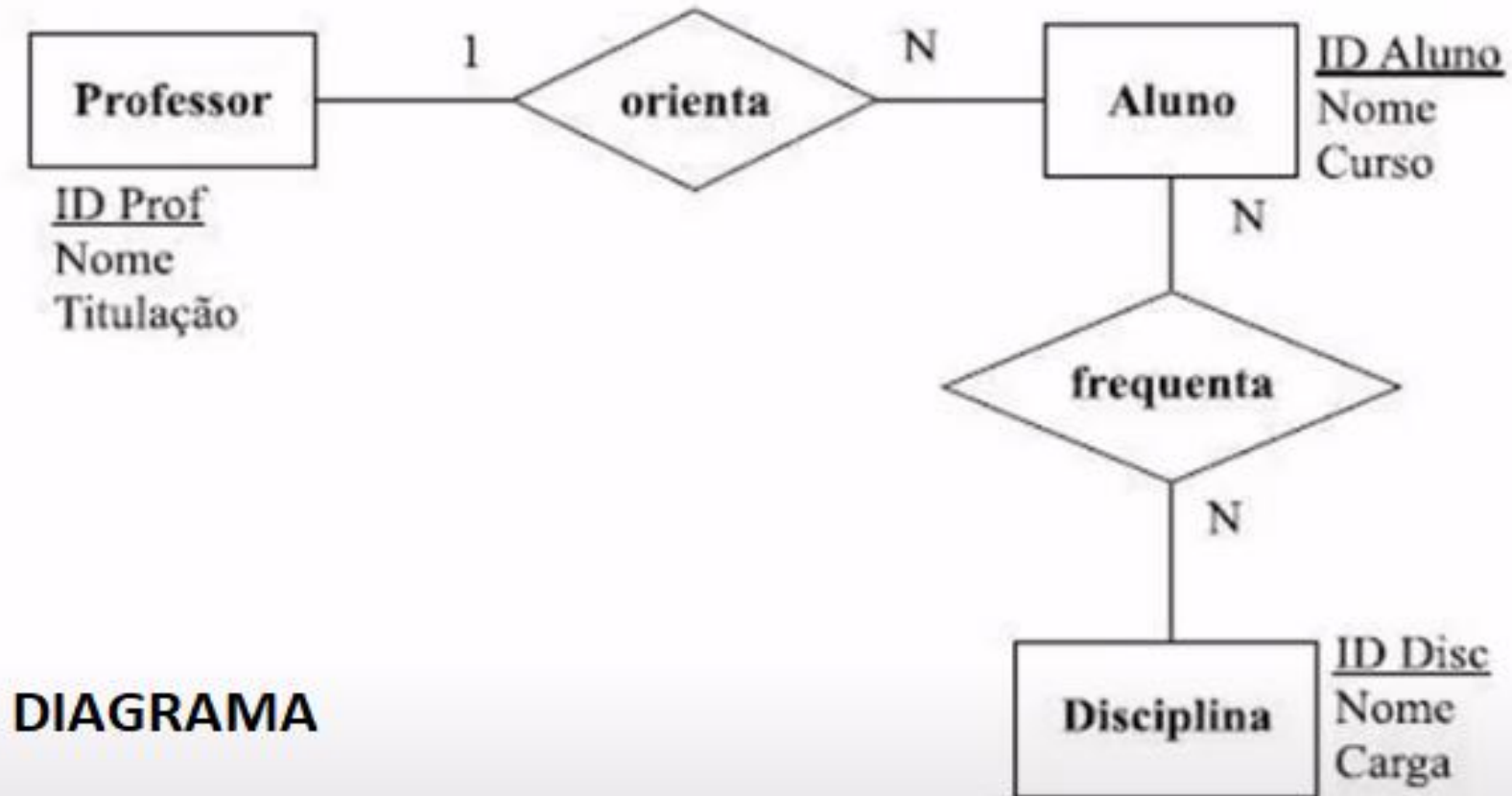
Criou uma nova entidade compra





A entidade gerada herda as chaves primárias das duas entidades como chaves estrangeira





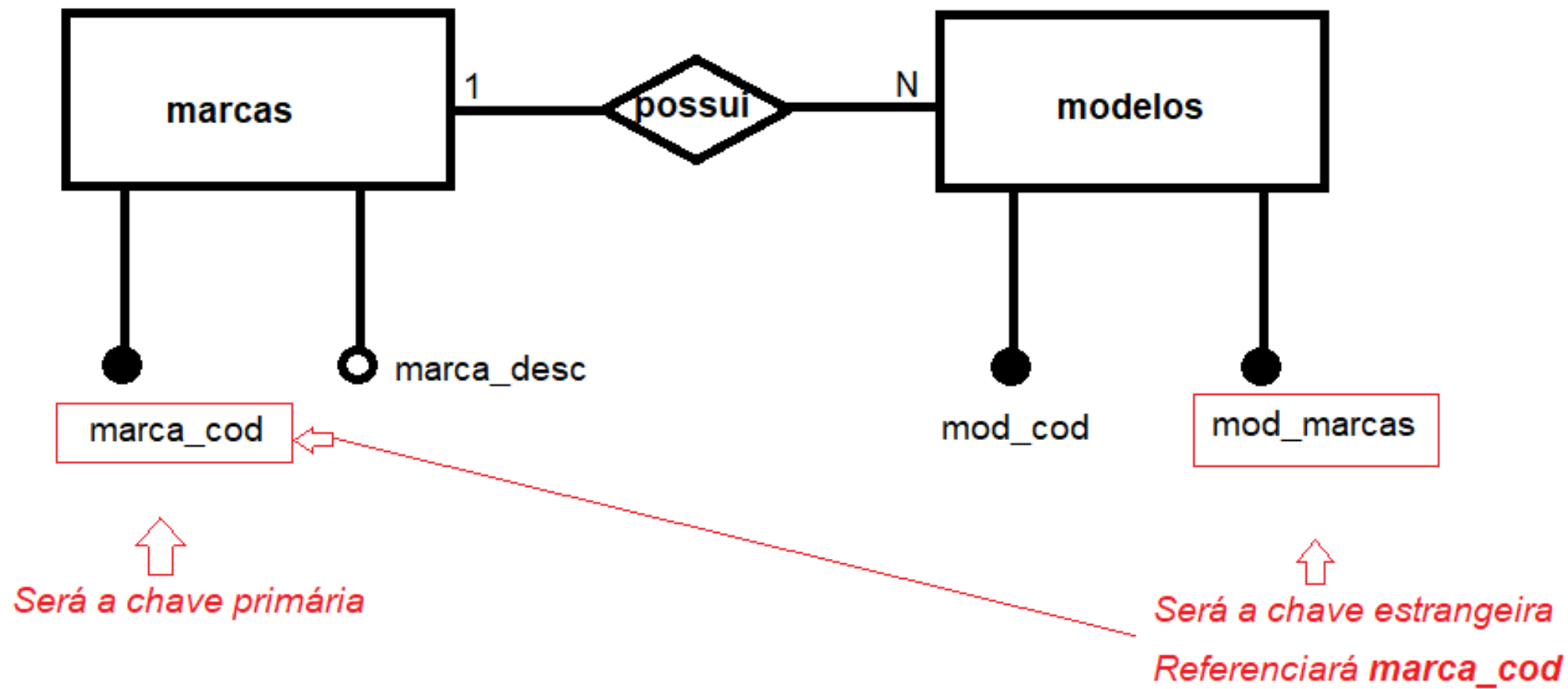
**DIAGRAMA**

[Diagrama online 1](#)

[Referência](#)

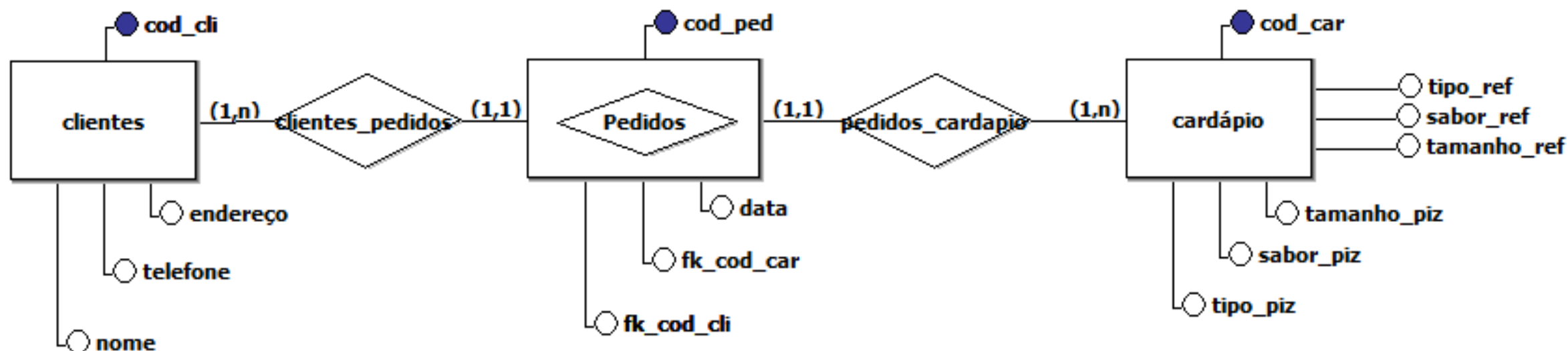
[Diagrama Online 2](#)

# Modelo Conceitual



# Modelo Conceitual

## Projeto Pizzaria



# Modelo Lógico

- Um modelo lógico possui conceitos que os usuários são capazes de entender, ao mesmo tempo em que não está distante do modelo físico do banco de dados.
- Neste nível o projeto é independente de SGBD.
- Consiste na especificação lógica dos dados em um formato adequado ao SGBD escolhido. Os tipos de dados são completamente definidos.

**Aqui o banco de dados começa a delinear. Fazemos especificação lógica dos dados.**

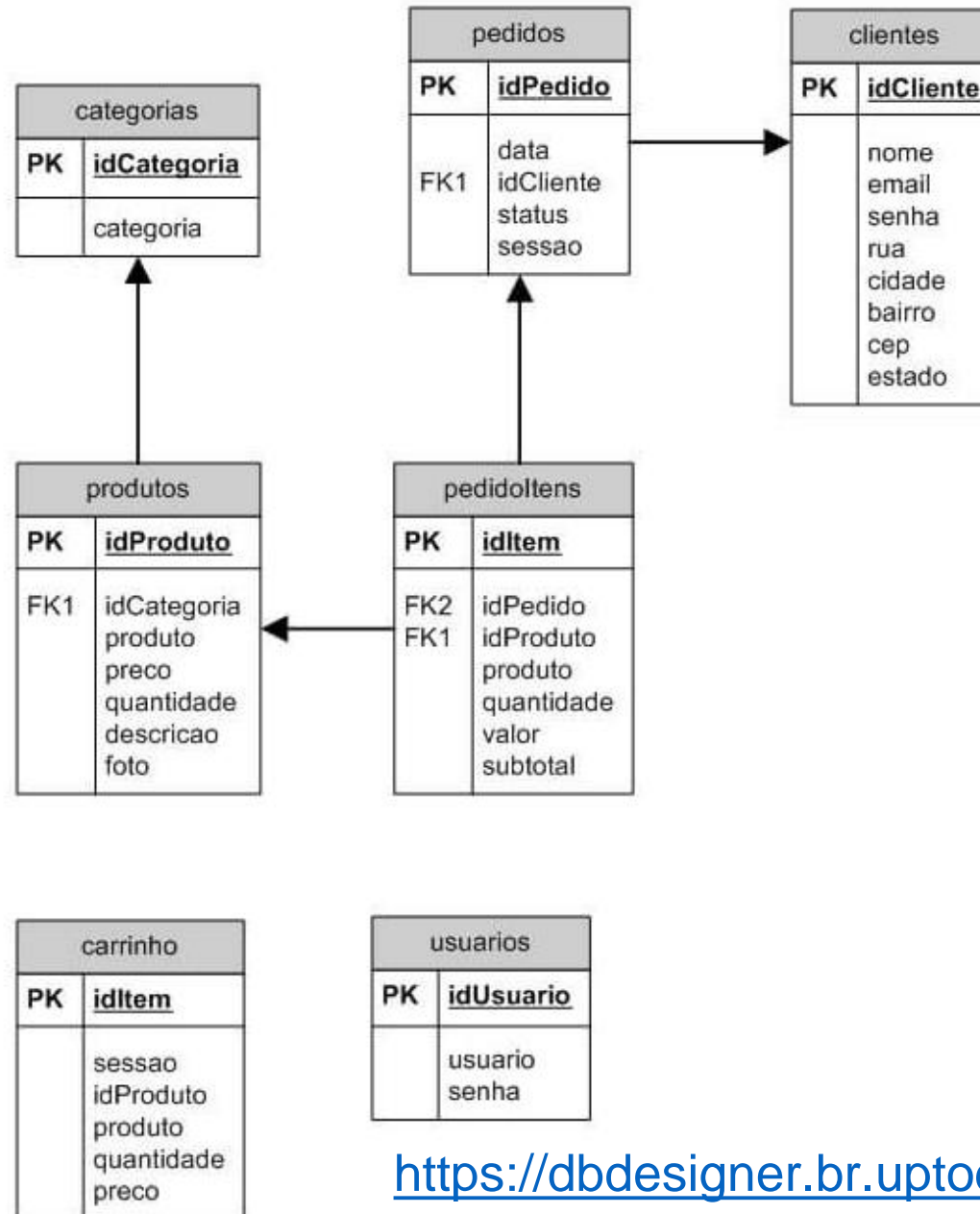
# Modelo Lógico



- O modelo lógico já leva em conta algumas limitações e implementa recursos como adequação de padrão e nomenclatura, define as CHAVES PRIMÁRIAS E ESTRANGEIRAS, normalização, integridade referencial, entre outras.
- Para o modelo lógico deve ser criado levando em conta os exemplos de modelagem de dados criados no modelo conceitual.

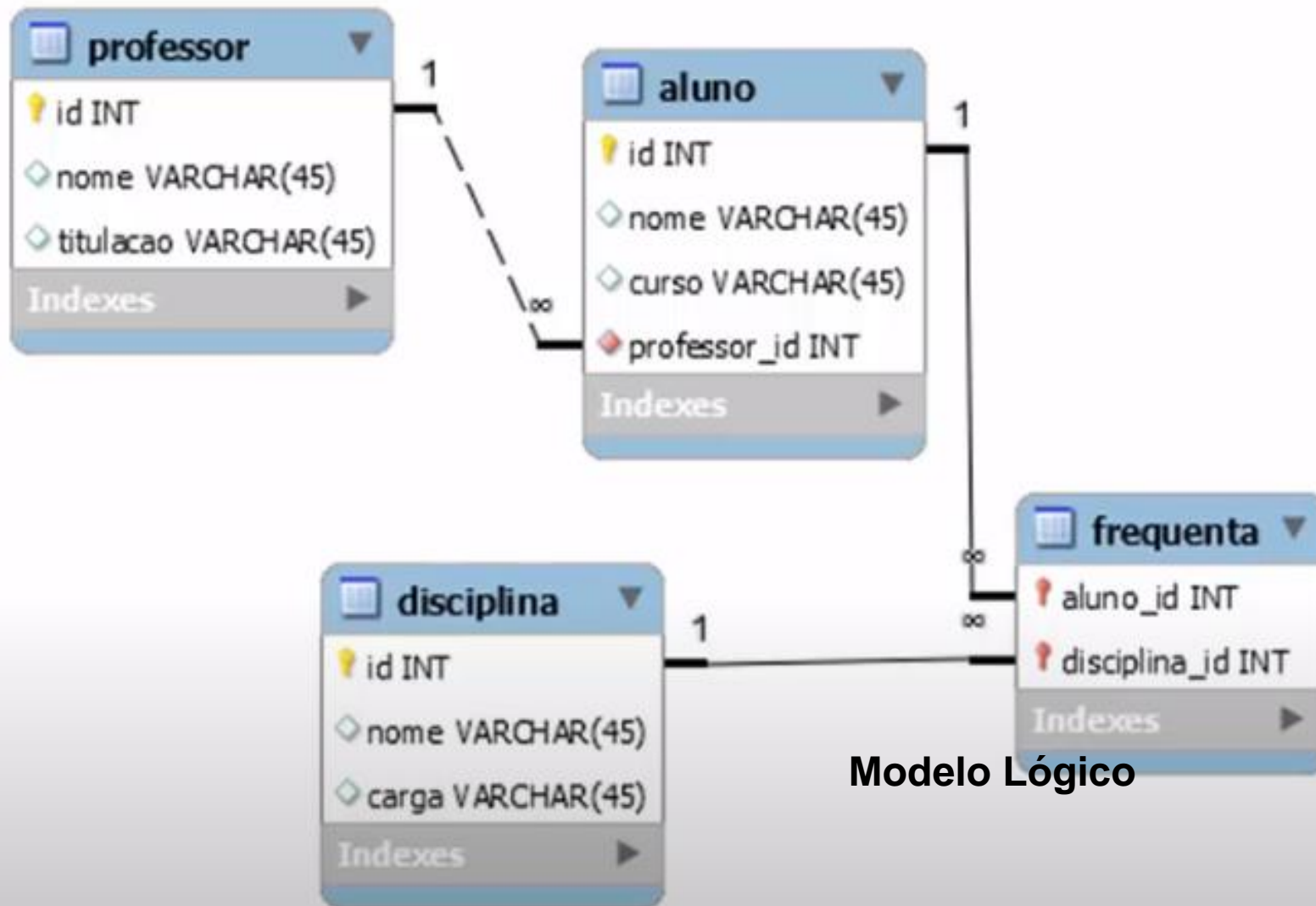


# Modelo Lógico



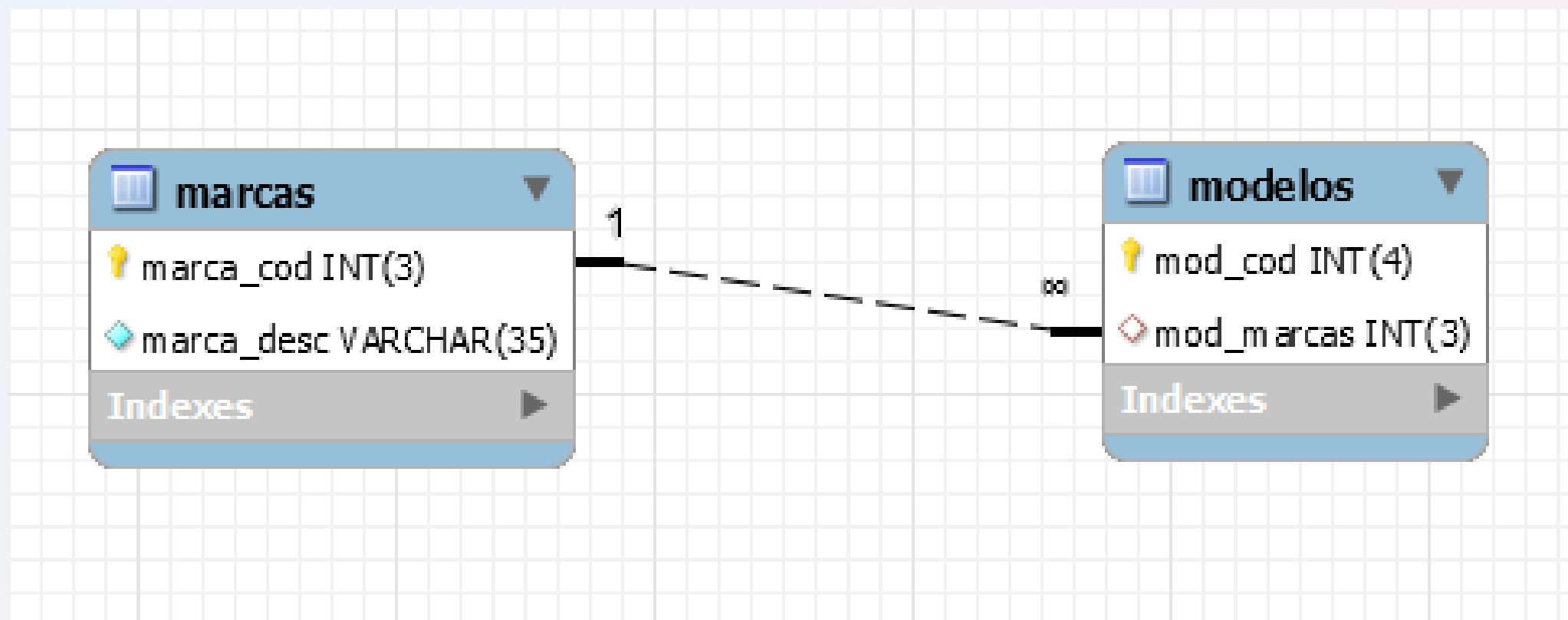
<https://dbdesigner.br.uptodown.com/windows>

# Modelo Lógico



Modelo Lógico

# Modelo Lógico



Desenho gerado no Workbench – Engenharia Reversa

# Modelo Lógico

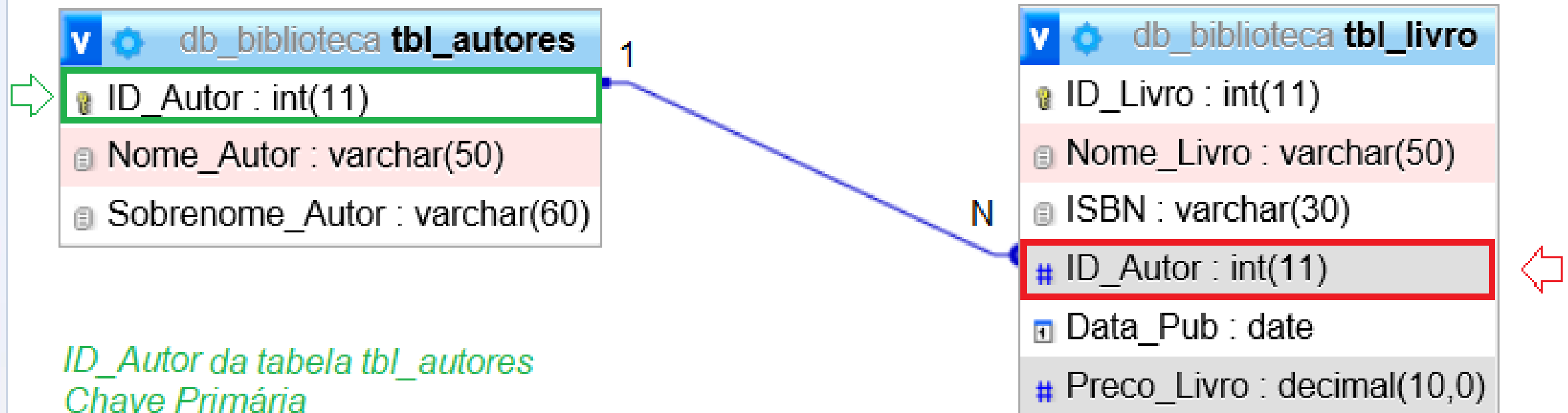
Chave Primária  
Primary key



Chave Estrangeira  
Chave Múltipla  
Foreign Key

Desenho gerado no phpMmyAdmin

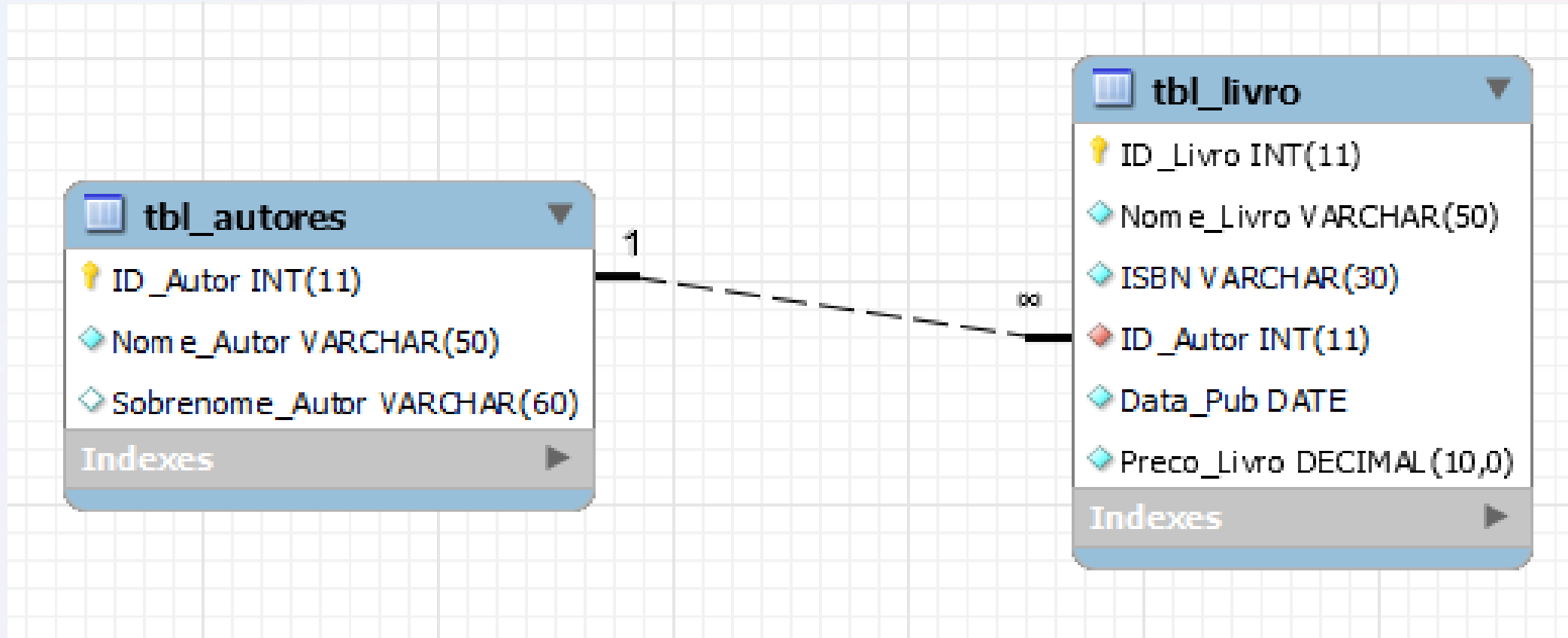
# Modelo Lógico



*ID\_Autor da tabela tbl\_autores  
Chave Primária  
Primary Key*

*ID\_Autor da tabela tbl\_livro  
Chave estrangeira criada  
referencia a chave primária ID\_Autor da tabela tbl\_autores*

# Modelo Lógico



Desenho gerado no Workbench – Engenharia Reversa



# Modelo físico

No modelo físico fazemos a modelagem física do modelo de banco de dados. Neste caso leva-se em conta as limitações impostas pelo SGBD escolhido e deve ser criado sempre com base nos exemplos de modelagem de dados produzidos no item anterior, modelo lógico.

```
create table usuarios
(
  id int AUTO_INCREMENT PRIMARY key,
  nome varchar(40),
  email varchar(40),
  senha varchar(32)
);
```

## Discussões e reflexões



**Assista o vídeo:**

**[Modelagem de dados - modelo conceitual, lógico e físico](#)**



## Recapitulando

Vamos fazer uma breve revisão  
sobre tudo que aprendemos até aqui?



# **MODELO FÍSICO**

## **COMANDOS MYSQL**



## COMANDO STATUS

Para saber o Banco  
de Dados atual

```
MySQL 8.0 Command Line Cli  X  +  v  -  □  X

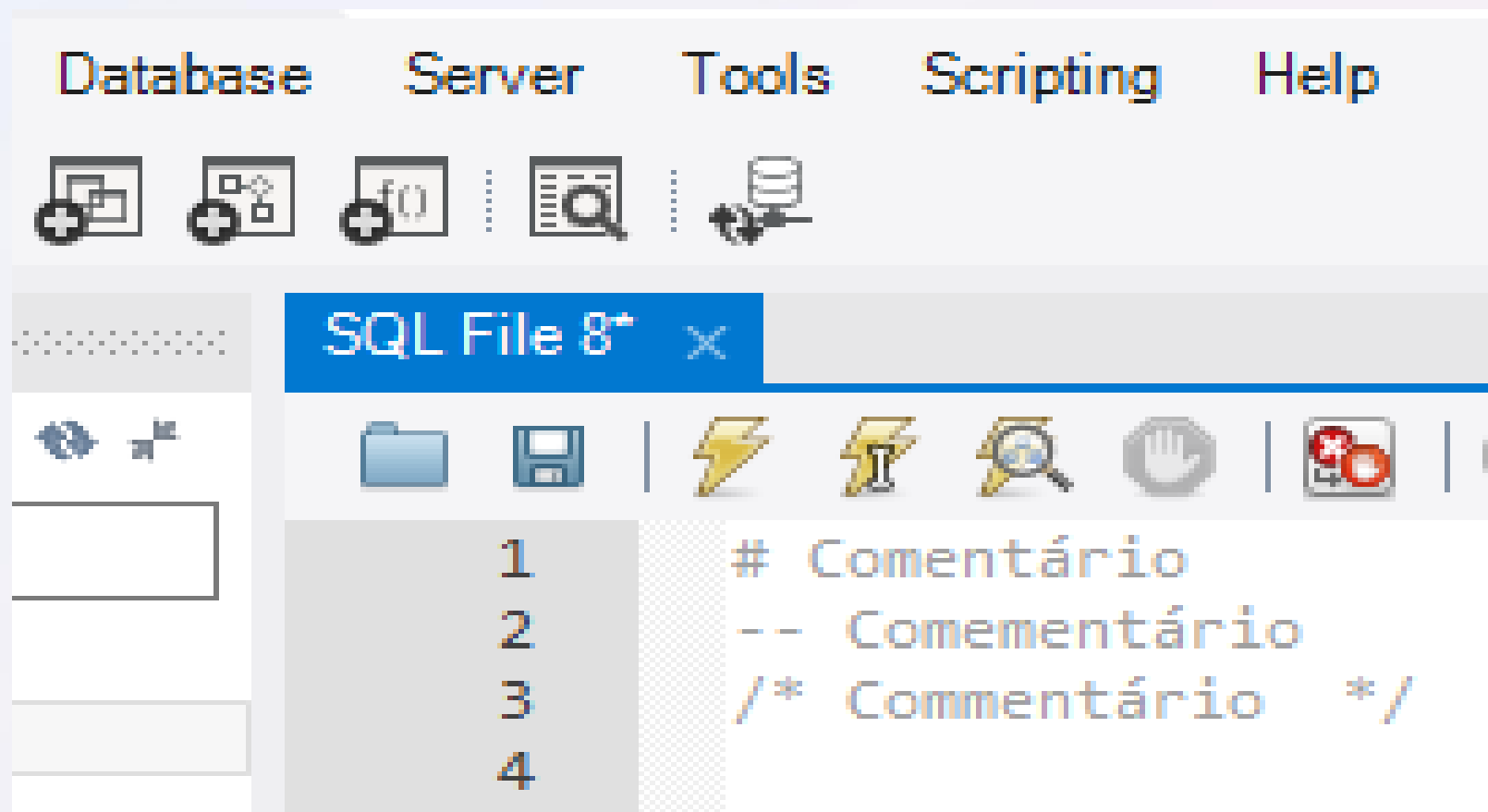
mysql> status
-----
C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe  Ver 8.0.22 for Win64
on x86_64 (MySQL Community Server - GPL)

Connection id:          10
Current database:
Current user:           root@localhost
SSL:                   Cipher in use is TLS_AES_256_GCM_SHA384
Using delimiter:        ;
Server version:         8.0.22 MySQL Community Server - GPL
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:    utf8mb4
Db      characterset:    utf8mb4
Client characterset:    utf8mb4
Conn.  characterset:    utf8mb4
TCP port:               3306
Binary data as:         Hexadecimal
Uptime:                 12 days 19 hours 28 min 19 sec

Threads: 2  Questions: 5  Slow queries: 0  Opens: 115  Flush tables: 3  Ope
n tables: 36  Queries per second avg: 0.000
-----
```



# COMENTÁRIOS NOS EDITORES MYSQL



## CRIANDO BANCO DE DADOS



```
CREATE DATABASE cadastro  
DEFAULT CHARACTER SET utf8mb4  
COLLATE utf8mb4_general_ci;  
  
USE cadastro;
```

```
CREATE CRIA  
USE SELECIONA
```

**DEFAULT CHARACTER SET utf8mb4:** Esta parte do código define o conjunto de caracteres padrão para o banco de dados. Neste caso, está definido como "utf8mb4". O "utf8mb4" é um conjunto de caracteres que suporta uma ampla variedade de caracteres, incluindo aqueles usados em muitos idiomas diferentes, bem como emojis.

É uma escolha comum para bancos de dados que precisam armazenar texto multilíngue e símbolos especiais.

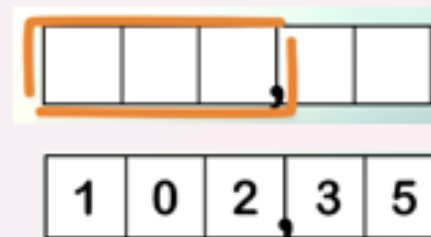
**COLLATE utf8mb4\_general\_ci:** O "COLLATE" é usado para definir a regra de ordenação padrão para o banco de dados. Neste caso, está definido como "utf8mb4\_general\_ci". O "\_ci" no final significa "case-insensitive" (insensível a maiúsculas/minúsculas), o que significa que as consultas SQL neste banco de dados irão tratar letras maiúsculas e minúsculas da mesma forma durante operações de ordenação. Isso é útil para garantir que as consultas sejam insensíveis a maiúsculas/minúsculas, o que é comum em muitos aplicativos.



```
1
2  /* enum Define quais são valores que são aceitos. Forma m
3  DEFAULT 'Brasil' Se ninguém digitar nada, por padrão é
4  chave primária os campos não se repetem. Não haverá dua
5  Não cadastrar idade, mas a data de nascimento
6  */
7
8  CREATE TABLE pessoas(
9      id INT NOT NULL AUTO_INCREMENT,
10     nome VARCHAR(30) NOT NULL,
11     nascimento DATE,
12     sexo ENUM ('M','F'),
13     peso DECIMAL (5,2),
14     altura DECIMAL (3,2),
15     nacionalidade VARCHAR(20) default 'Brasileira',
16     PRIMARY KEY (id)
17 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



decimal



**Default charset** vai aceitar acentos padrão latino

**ENGINE=InnoDB:** Esta parte do código especifica o mecanismo de armazenamento que deve ser usado para a tabela. Neste caso, está definido como "InnoDB". O InnoDB é um dos mecanismos de armazenamento mais populares para bancos de dados MySQL. Ele oferece recursos de transação, conformidade ACID (***Atomicidade, Consistência, Isolamento, Durabilidade***) e é adequado para a maioria das aplicações que exigem integridade de dados, suporte a transações e controle de concorrência.

**DEFAULT CHARSET=utf8mb4:** Aqui, você está definindo o conjunto de caracteres padrão para a tabela. É especificado como "utf8mb4", que é um conjunto de caracteres que suporta uma ampla variedade de caracteres, incluindo caracteres multilíngues e símbolos especiais, como emojis. Essa configuração determina como os dados de texto na tabela serão armazenados e tratados em relação ao conjunto de caracteres.

**Vamos pesquisar!**

*O que é Atomicidade, Consistência, Isolamento, Durabilidade em um banco de dados?*





# Atomicidade, Consistência, Isolamento e Durabilidade



Comumente referidos pelo acrônimo ACID, são um conjunto de propriedades importantes para transações em sistemas de gerenciamento de banco de dados (SGBD).

Essas propriedades garantem que as transações de banco de dados sejam processadas de maneira confiável e segura. Vamos detalhar cada uma dessas propriedades:

## 1. Atomicidade (Atomicity):

Garante que todas as operações dentro de uma transação sejam concluídas com sucesso ou que nenhuma operação seja aplicada ao banco de dados.

- Exemplo: Se você está transferindo dinheiro de uma conta para outra, a transação deve garantir que o débito de uma conta e o crédito na outra sejam concluídos juntos. Se uma das operações falhar, nenhuma delas deve ser aplicada.

## 2 - Consistência (Consistency):



Garante que uma transação leve o banco de dados de um estado consistente a outro estado consistente, preservando todas as regras definidas (como integridade referencial).

- Exemplo: Se uma regra do banco de dados exige que o saldo de uma conta não seja negativo, a consistência garante que, após a transação, essa regra ainda seja cumprida.

### 3. Isolamento (Isolation):

Garante que as operações de uma transação sejam isoladas de outras transações em andamento, evitando interferências e garantindo que o resultado seja o mesmo como se as transações fossem executadas sequencialmente.

- Exemplo: Se dois usuários estão tentando modificar o mesmo dado ao mesmo tempo, o isolamento assegura que uma transação não verá os dados intermediários das operações da outra.

## 4. Durabilidade (Durability):



Garante que, uma vez que uma transação seja confirmada (committed), seus efeitos sejam permanentemente preservados no banco de dados, mesmo em caso de falhas de sistema ou quedas de energia.

- Exemplo: Depois de confirmar uma transferência de dinheiro entre contas, a alteração no saldo das contas deve ser permanente e visível, mesmo após um desligamento inesperado do sistema.

# APRESENTANDO TODOS OS BANCO DE DADOS



DatabaseServerToolsScriptingHelp

SQL File 3\*

SQL File 6\*

SQL File 7\* x

Limit to 1000 rows

1 • show databases;

2

3

4

5

<

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Database
	information schema
	cadastro
	mvsol
	performance schema
	test

SHOW APRESENTA



# APRESENTANDO TODOS AS TABELAS DO BANCO DE DADOS



25

26 • `show tables;`



Result Grid



Filter Rows:

Tables\_in\_cadastro

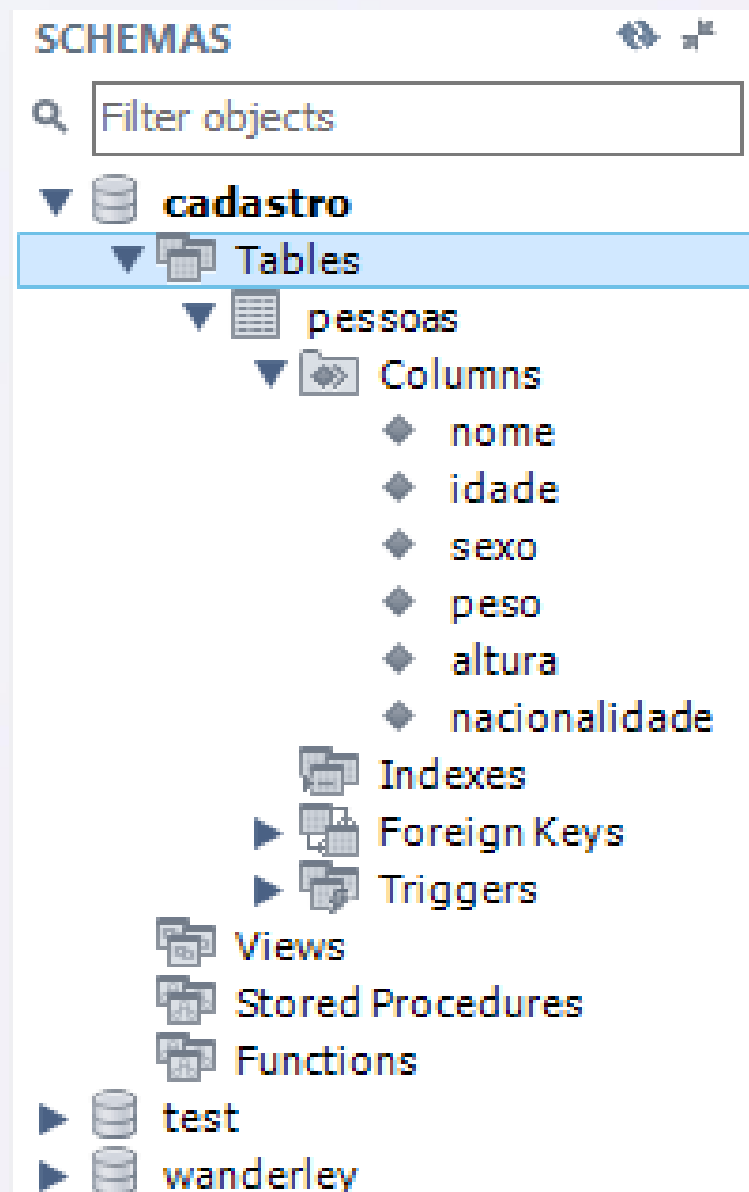


peessoas

```
mysql> show tables;
+-----+
| Tables_in_cadastro |
+-----+
| pessoas            |
+-----+
1 row in set (0.00 sec)
```

**SHOW APRESENTA**

# O BD APARECE ATIVO NO SCHEMAS DO WORKBENCH






Mostra a tabela criada

Mostra os campos criados

# DESCREVER A ESTRUTURA DA TABELA

```
18
19 Describe pessoas;
20
```

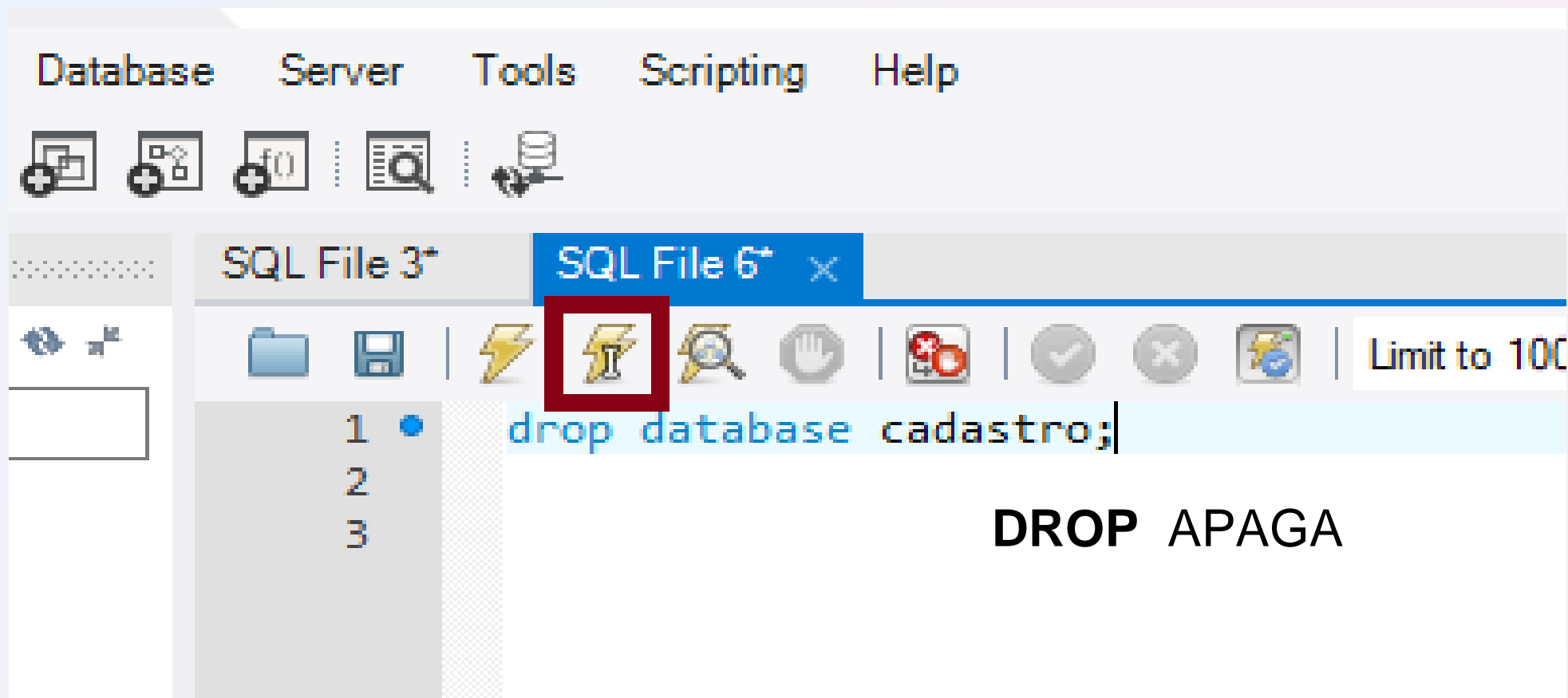
Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	Field	Type	Null	Key	Default	Extra
	nome	varchar(30)	YES		NULL	
	idade	tinyint(3)	YES		NULL	
	sexo	char(1)	YES		NULL	
	peso	float	YES		NULL	
	altura	float	YES		NULL	
	nacionalidade	varchar(20)	YES		NULL	

DESCRIBE DESCRIVER

```
1
2 -- COMANDOS QUE APRESENTAM OS MESMOS RESULTADOS DOS COMANDOS DESC OU
  DESCRIBE:
3
4 -- Apresenta as colunas da tabela
5 SHOW COLUMNS FROM pessoas;
6
7 -- Apresenta os campos da tabela
8 SHOW FIELDS FROM pessoas;
```

# DELETANDO O BANCO DE DADOS CADASTRO



Executar: cursor no final do código. Pode utilizar os comando control + enter ou clicar no ícone do raio acima

# PARA APRESENTAR OS CÓDIGOS QUE GERARAM A TABELA E OS CÓDIGOS QUE GERARAM O BANCO DE DADOS



```
19 • show create table pessoas;
```

```
20
```

< **Result Grid** | Filter Rows:  | Export: | Wrap Cell Content:

	Table	Create Table
▶	pessoas	CREATE TABLE `pessoas` ( `id` int(11) NOT NULL AUTO_INCREMENT, `nome` varchar(30) NOT NULL, `nascimento` date DEFAULT NULL, `sexo` enum('M','F') DEFAULT NULL, `peso` decimal(5,2) D...

```
21 • show create database cadastro;
```

**Result Grid** | Filter Rows:  | Export: | Wrap Cell Content:

	Database	Create Database
▶	cadastro	CREATE DATABASE `cadastro` /*!40100 DEFAULT CHARACTER SET utf8 */

# INSERINDO DADOS NA TABELA

```
peessoas  cadastro.pessoas  cadastro.pessoas  SQL File 9  SQL File
Limit to 1000 rows
/*
PEGAR TODOS OS CAMPOS DA TABELA PESSOAS
(id, nome, nascimento, sexo, peso, altura, nacionalidade)
data no mysql primeiro a ano, traço, mês, traço e depois o dia,
*/
```

INSERT INTO Pessoas INSIRA NA TABELA Pessoas

```
7  insert into pessoas (
8
9      id,  Tirar, ele é auto incremento
10     nome,
11     nascimento,
12     sexo,
13     peso,
14     altura, nacionalidade
15 )
16 values
17
18
19 (
20
21     '1',  Tirar, ele é auto incremento
22     'Wanderley',
23     '1975-02-24',
24     'M',
25     '78.5',
26     '1.83',
27     'Brasil'
28 )
29
30 ;
```

VALUES OS VALORES

# SEM OS CAMPOS AUTO INCREMENTÁVEIS

```
32 • select * from pessoas;
```

Result Grid

id	nome	nascimento	sexo	peso	altura	nacionalidade
1	Wanderlev	1975-02-24	M	78.50	1.00	Brasil
2	Wanderlev	1975-02-24	M	78.50	1.83	Brasil
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Form Editor

Id:

Nome:

Nascimento:

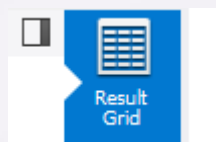
Sexo:

Peso:

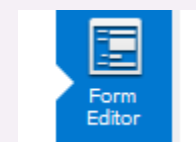
Altura:

Nacionalidade:

Selecionar  
aba direita



Selecionar  
aba direita



```
9 • insert into pessoas (  
10  
11     nome,  
12     nascimento,  
13     sexo,  
14     peso,  
15     altura, nacionalidade  
16  
17 )  
18 values  
19  
20 (  
21  
22     'Wanderley',  
23     '1975-02-24',  
24     'M',  
25     '78.5',  
26     '1.83',  
27     'Brasil'  
28  
29 )  
30 );  
31  
32 • select * from pessoas;  
33
```





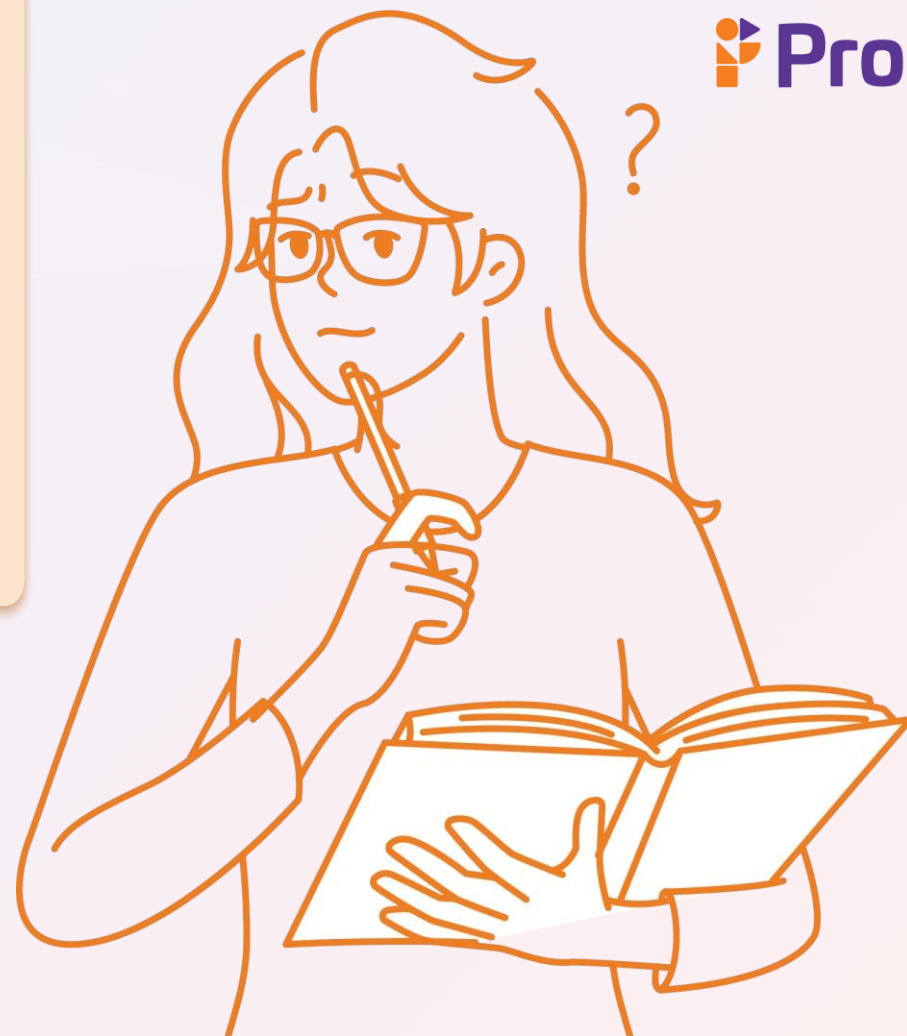
```
create table cursos(  
    id int not null,  
    nome varchar(30),  
    carga time,  
    ano date  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
-- DATE FORMATO padrão: 'aaaa-mm-dd'  
-- TIME FORMATO padrão: 'hh:mm:ss'
```

```
insert into cursos(  
    id,  
    nome,  
    carga,  
    ano  
  
)values(  
    3,  
    'Wanderley Lopes Batista',  
    '02:30:00',  
    '2020-07-03'  
  
);
```

## Exercício:

Inserir mais 10 campos  
nessa tabela e verificar  
o resultado



```

9  •  insert into pessoas (
10      id,
11      nome,
12      nascimento,
13      sexo,
14      peso,
15      altura, nacionalidade
16  )
17  values
18
19
20  (
21      DEFAULT,
22      'Creuza',
23      '1920-12-30',
24      'F',
25      '50.2',
26      '1.65',
27      DEFAULT
28  );
29
30
31
32  •  select * from pessoas;

```

## INSERINDO DEFAULT VALORES PADRÃO



## Verificando o resultado

[illegible]

```

9 • insert into pessoas (
10     id,
11     nome,
12     nascimento,
13     sexo,
14     peso,
15     altura, nacionalidade
16 )
17     values
18     (
19         DEFAULT,
20         'Adalgiza',
21         '1930-12-02',
22         'F',
23         '63.2',
24         '1.75',
25         Irlanda
26     );

```

Pode tirar, na mesma ordem.

```

9 • insert into pessoas values
10     (
11         DEFAULT,
12         'Adalgiza',
13         '1930-12-02',
14         'F',
15         '63.2',
16         '1.75',
17         'Irlanda'
18     );
19
20 • select * from pessoas;

```



**INSERINDO DADOS  
SEM OS CAMPOS NO  
INSERT E VERIFICANDO**

Result Grid							
Filter Rows:							
	id	nome	nascimento	sexo	peso	altura	nacionalidade
	1	Wanderlev	1975-02-24	M	78.50	1.00	Brasil
	2	Wanderlev	1975-02-24	M	78.50	1.83	Brasil
	3	Maria	1999-12-30	F	55.20	1.65	Portual
	4	Creuza	1920-12-30	F	50.20	1.65	Brasil
	5	Adalgiza	1930-12-02	F	63.20	1.75	Irlanda
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Resultset

# INSERINDO MAIS DE UM CADASTRO AO MESMO TEMPO



```
3 • insert into pessoas values
4
5 (DEFAULT, 'Ana', '1975-12-22', 'F', '52.3', '1.45', 'EUA'),
6 (DEFAULT, 'Maria', '1999-05-30', 'F', '75.8', '1.70', 'Portugal'),
7 (DEFAULT, 'Pedro', '2000-07-15', 'M', '52.3', '1.45', 'Brasil');
8
```

```
9
10 • insert into pessoas values (DEFAULT, 'Evandro', '1984-06-21', 'M', '90.6', '1.5', 'Chile'),
11                               (DEFAULT, 'Siviana', '2001-07-04', 'F', '90.4', '1.4', 'Grécia'),
12                               (DEFAULT, 'Carlos', '2007-04-09', 'F', '89.6', '1.7', 'Argentina'),
13                               (DEFAULT, 'Henrique', '2009-08-24', 'F', '76.8', '1.9', 'Brasil');
14
15
```

```
select * from pessoas;
```

Filter Rows: <input type="text"/>							Edit:			Export/1
id	nome	nascimento	sexo	peso	altura	nacionalidade				
	Wanderlev	1975-02-24	M	78.50	1.00	Brasil				
	Wanderlev	1975-02-24	M	78.50	1.83	Brasil				
	Maria	1999-12-30	F	55.20	1.65	Portugal				
4	Creuza	1920-12-30	F	50.20	1.65	Brasil				
5	Adalaiza	1930-12-02	F	63.20	1.75	Irlanda				
6	Ana	1975-12-22	F	52.30	1.45	EUA				
7	Maria	1999-05-30	F	75.80	1.70	Portugal				
8	Pedro	2000-07-15	M	52.30	1.45	Brasil				

Verificando os cadastros inseridos

```
create database if not exists db_biblioteca
```

```
default character set utf8
```

```
default collate utf8_general_ci;
```

```
use db_biblioteca;
```

```
CREATE TABLE IF NOT EXISTS tbl_autores(  
ID_Autor INT NOT NULL AUTO_INCREMENT  
PRIMARY KEY,
```

```
Nome_Autor VARCHAR(50) NOT NULL,
```

```
Sobrenome_Autor VARCHAR(60)
```

```
)ENGINE = InnoDB DEFAULT CHARSET =  
utf8;
```

```
CREATE TABLE IF NOT EXISTS tbl_livro(  
ID_Livro INT AUTO_INCREMENT PRIMARY KEY,  
Nome_Livro varchar(50) NOT NULL,  
ISBN Varchar (30) NOT NULL,  
ID_Autor INT NOT NULL,  
Data_Pub DATE NOT NULL,  
Preco_Livro decimal NOT NULL  
)ENGINE = InnoDB CHARSET = utf8;
```

De acordo com a Câmara Brasileira do **Livro**, “o **ISBN** (International Standard Book Number/ Padrão Internacional de Numeração de **Livro**) é um padrão numérico criado com o objetivo de fornecer uma espécie de “RG” para publicações monográficas, como **livros**, artigos e apostilas.



# RELACIONAMENTO ENTRE TABELAS



1º

```
1 create database if not exists db_biblioteca
2 default character set utf8
3 default collate utf8_general_ci;
```

```
use db_biblioteca;
```

3º

```
CREATE TABLE IF NOT EXISTS tbl_livro(
3 ID_Livro INT AUTO_INCREMENT PRIMARY KEY,
4 Nome_Livro varchar(50) NOT NULL,
5 ISBN Varchar (30) NOT NULL,
6 ID_Autor INT NOT NULL,
7 Data_Pub DATE NOT NULL,
8 Preco_Livro decimal NOT NULL
9 )ENGINE = InnoDB CHARSET = utf8;
```



Será transformada em  
PK depois

10

2º

```
3 CREATE TABLE IF NOT EXISTS tbl_autores(
4 ID_Autor INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
5 Nome_Autor VARCHAR(50) NOT NULL,
6 Sobrenome_Autor VARCHAR(60)
7 )ENGINE = InnoDB DEFAULT CHARSET = utf8;
```



Chave estrangeira

```
1 CREATE DATABASE IF NOT EXISTS db_carros
2 DEFAULT CHARACTER SET utf8mb4
3 COLLATE utf8mb4_general_ci;
```

```
1 USE db_revenda;
```

```
3 CREATE TABLE marcas(
4     marca_cod INT(3) NOT NULL AUTO_INCREMENT,
5     marca_desc VARCHAR(35) NOT NULL,
6     PRIMARY KEY (marca_cod)
7 ) ENGINE = INNODB CHARSET = utf8mb4;
```

RELACIONAMENTO

```
1 CREATE TABLE IF NOT EXISTS modelos(
2     mod_cod INT(4) NOT NULL AUTO_INCREMENT,
3     mod_marcas INT(3),
4     PRIMARY KEY (mod_cod),
5     FOREIGN KEY (mod_marcas) REFERENCES marcas (marca_cod)
6 ) ENGINE = INNODB CHARSET = utf8mb4;
```

```
CREATE DATABASE IF NOT EXISTS db_carros  
DEFAULT CHARACTER SET utf8mb4  
COLLATE utf8mb4_general_ci;
```

```
USE db_carros;
```

```
CREATE TABLE marcas(  
marca_cod INT(3) NOT NULL AUTO_INCREMENT,  
marca_desc VARCHAR(35) NOT NULL,  
PRIMARY KEY (marca_cod)  
)ENGINE=InnoDB DEFAULT CHARSET = utf8mb4;
```

```
CREATE TABLE IF NOT EXISTS modelos(  
mod_cod INT(4) NOT NULL AUTO_INCREMENT,  
mod_marcas INT(3),  
PRIMARY KEY (mod_cod),  
FOREIGN KEY (mod_marcas) REFERENCES marcas (marca_cod)  
)ENGINE=InnoDB DEFAULT CHARSET = utf8mb4;;
```

## OUTRO EXEMPLO DE RELACIONAMENTO

```
1 • CREATE DATABASE IF NOT EXISTS INFOX
2   DEFAULT CHARACTER SET utf8mb4
3   COLLATE utf8mb4_general_ci;
4
5 • USE INFOX;
6
7 • CREATE TABLE tbclientes(
8   idcli INT PRIMARY KEY AUTO_INCREMENT,
9   nomcli VARCHAR(50) NOT NULL,
10  endcli VARCHAR(100),
11  foncli VARCHAR(14) NOT NULL,
12  emacli VARCHAR(30)
13 )ENGINE=InnoDB DEFAULT CHARSET = utf8mb4;
14
```

```
• CREATE TABLE tbos( -- tabela de Ordem de Serviço
os INT PRIMARY KEY AUTO_INCREMENT,
-- TIMESTAMP Um dado desse tipo é armazenado com quantidde de segundos que passou desde
-- uma data de referência, no caso do Mysql é 1970-01-01 00:00:00 formato time: "HH:MM:SS"
data_os TIMESTAMP,
equipamento VARCHAR(150) NOT NULL,
defeito VARCHAR(150) NOT NULL,
tecnico VARCHAR(30),
valor DECIMAL(10,2),
idcli INT NOT NULL,
FOREIGN KEY (idcli) REFERENCES tbclientes(idcli)

)ENGINE=InnoDB DEFAULT CHARSET = utf8mb4;
```

```
-- Geração de Modelo físico  
-- Sql ANSI 2003 - brModelo.
```

```
CREATE TABLE PRODUTO (  
  cod_prod VARCHAR(10) PRIMARY KEY,  
  nome VARCHAR(10),  
  fabricante VARCHAR(10),  
  preco VARCHAR(10)  
)
```

```
CREATE TABLE CLIENTE (  
  nome VARCHAR(10),  
  cpf VARCHAR(10) PRIMARY KEY,  
  endereco VARCHAR(10),  
  telefone VARCHAR(10)  
)
```

```
CREATE TABLE COMPRA (  
  cod_prod VARCHAR(10),  
  cpf VARCHAR(10),  
  FOREIGN KEY(cod_prod) REFERENCES PRODUTO (cod_prod),  
  FOREIGN KEY(cpf) REFERENCES CLIENTE (cpf)  
)
```