



Modelo Entidade Relacionamento

Herysson R. Figueiredo
herysson.figueiredo@ufn.edu.br



Modelo Entidade Relacionamento

O Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).



Modelo Entidade Relacionamento

Em geral, este modelo representa de forma abstrata a estrutura que possuirá o banco de dados da aplicação. Obviamente, o banco de dados poderá conter várias outras entidades, tais como chaves e tabelas intermediárias, que podem só fazer sentido no contexto de bases de dados relacionais.



Entidades

Os objetos ou partes envolvidas um domínio, também chamados de entidades, podem ser classificados como físicos ou lógicos, de acordo sua existência no mundo real.



Entidades

Entidades físicas: são aquelas realmente tangíveis, existentes e visíveis no mundo real, como um cliente (uma pessoa, uma empresa) ou um produto (um carro, um computador, uma roupa).



Entidades

Entidades lógicas: são aquelas que existem geralmente em decorrência da interação entre ou com entidades físicas, que fazem sentido dentro de um certo domínio de negócios, mas que no mundo externo/real não são objetos físicos (que ocupam lugar no espaço). São exemplos disso uma venda ou uma classificação de um objeto (modelo, espécie, função de um usuário do sistema).



Entidades

As entidades são nomeadas com substantivos concretos ou abstratos que representem de forma clara sua função dentro do domínio. Exemplos práticos de entidades comuns em vários sistemas são Cliente, Produto, Venda, Turma, Função, entre outros.



Entidades

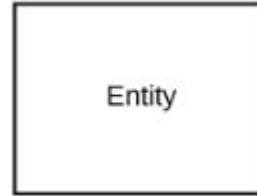
Podemos classificar as entidades segundo o motivo de sua existência:

- Entidades fortes:
- Entidades fracas:
- Entidades associativas:

Entidades

Podemos classificar as entidades segundo o motivo de sua existência:

- Entidades fortes:
- Entidades fracas:
- Entidades associativas:



Strong entity



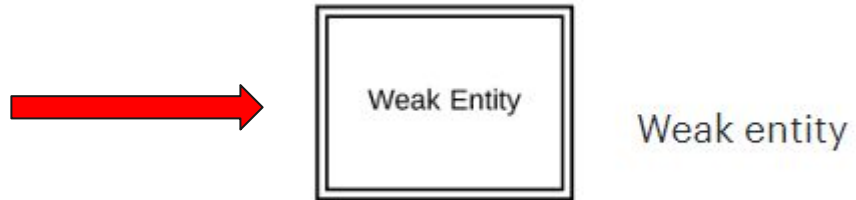
Entidades Fortes

São aquelas cuja existência independe de outras entidades, ou seja, por si só elas já possuem total sentido de existência. Em um sistema de vendas, a entidade produto, por exemplo, independe de quaisquer outras para existir.

Entidades

Podemos classificar as entidades segundo o motivo de sua existência:

- Entidades fortes:
- Entidades fracas:
- Entidades associativas:





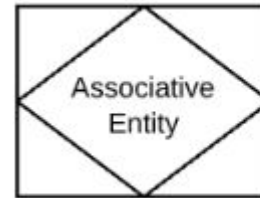
Entidades Fracas

Ao contrário das entidades fortes, as fracas são aquelas que dependem de outras entidades para existirem, pois individualmente elas não fazem sentido. Mantendo o mesmo exemplo, a entidade venda depende da entidade produto, pois uma venda sem itens não tem sentido.

Entidades

Podemos classificar as entidades segundo o motivo de sua existência:

- Entidades fortes:
- Entidades fracas:
- Entidades associativas:



Associative entity



Entidades Associativas

Esse tipo de entidade surge quando há a necessidade de associar uma entidade a um relacionamento existente. Na modelagem Entidade-Relacionamento não é possível que um relacionamento seja associado a uma entidade, então tornamos esse relacionamento uma entidade associativa, que a partir daí poderá se relacionar com outras entidades.



Atributos

Atributos são as características que descrevem cada entidade dentro do domínio. Por exemplo, um cliente possui nome, endereço e telefone.



Atributos

Durante a análise de requisitos, são identificados os atributos relevantes de cada entidade naquele contexto, de forma a manter o modelo o mais simples possível e consequentemente armazenar apenas as informações que serão úteis futuramente.



Atributos

Durante a análise de requisitos, são identificados os atributos relevantes de cada entidade naquele contexto, de forma a manter o modelo o mais simples possível e consequentemente armazenar apenas as informações que serão úteis futuramente.



Atributos

Os atributos podem ser classificados quanto à sua função da seguinte forma:

- Descritivos:
- Nominativos:
- Referenciais:
- Simples:
- Compostos:



Atributos

Os atributos podem ser classificados quanto à sua função da seguinte forma:

- Descritivos:
- Nominativos:
- Referenciais:
- Simples:
- Compostos:

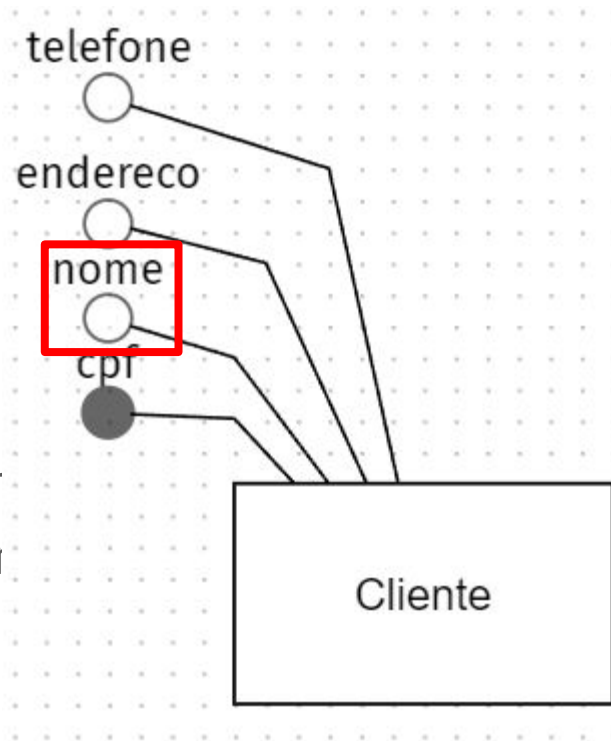


Atributos

Descritivos: representam características intrínsecas de uma entidade, tais como nome ou cor.

Atributos

Descritivos: representam
entidade, tais como nome



instâncias de uma



Atributos

Os atributos podem ser classificados quanto à sua função da seguinte forma:

- Descritivos:
- Nominativos:
- Referenciais:
- Simples:
- Compostos:

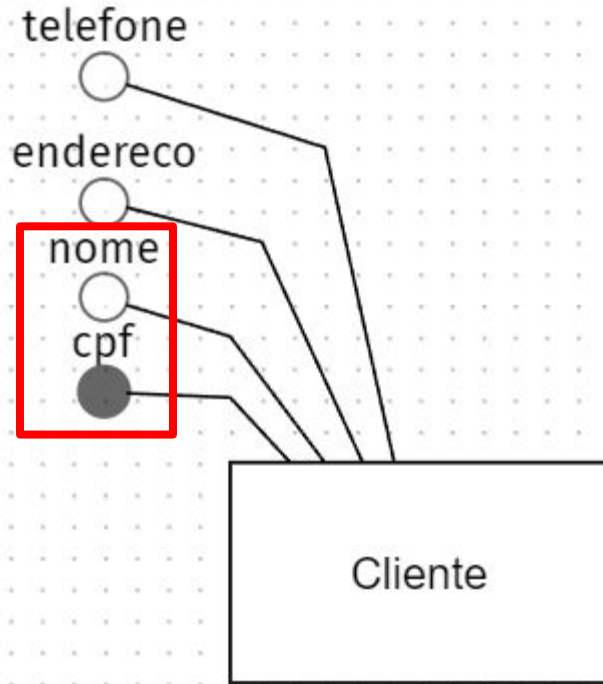


Atributos

Normativos: além de serem também descritivos, estes têm a função de definir e identificar um objeto. Nome, código, número são exemplos de atributos nominativos.

Atributos

Descritivos: além de se
de definir e identificar
exemplos de atributos



estes têm a função
digo, número são



Atributos

Os atributos podem ser classificados quanto à sua função da seguinte forma:

- Descritivos:
- Nominativos:
- Referenciais:
- Simples:
- Compostos:



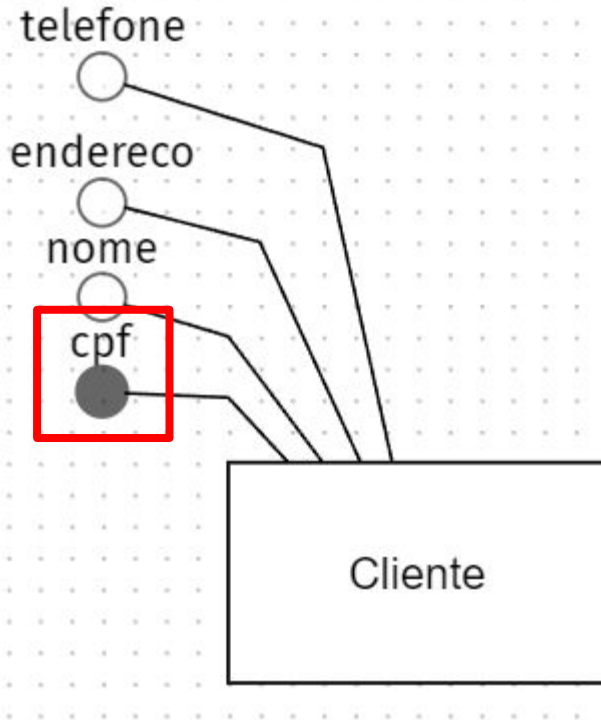
Atributos

Referenciais: representam a ligação de uma entidade com outra em um relacionamento. Por exemplo, uma venda possui o CPF do cliente, que a relaciona com a entidade cliente.

Atributos

Referenciais: representam um relacionamento. F

cliente, que a relaciona



ade com outra em
possui o CPF do



Atributos

Os atributos podem ser classificados quanto à sua função da seguinte forma:

- Descritivos:
- Nominativos:
- Referenciais:
- Simples:
- Compostos:

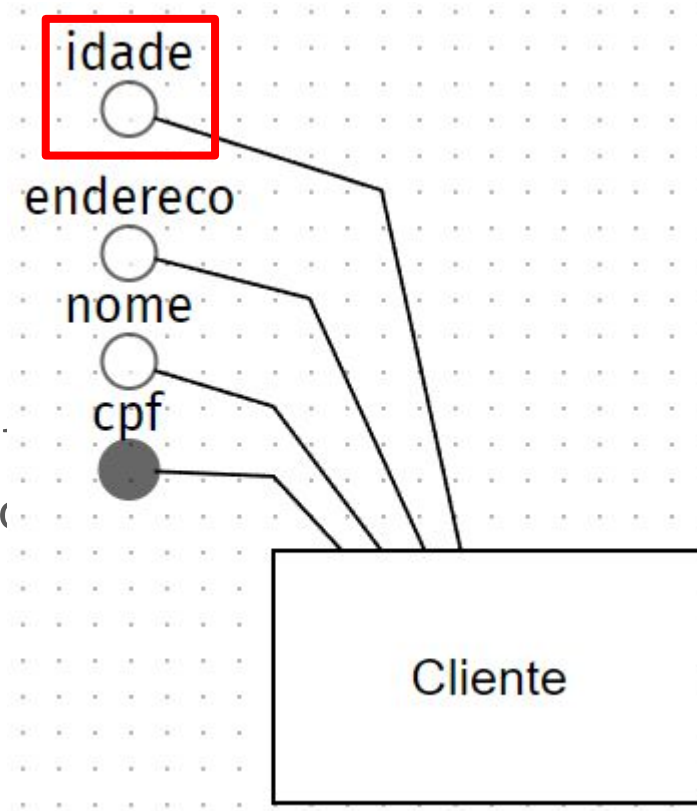


Atributos

Simples: um único atributo define uma característica da entidade.
Exemplos: idade, peso, gênero.

Atributos

Simples: um único a
Exemplos: idade, peso



ica da entidade.



Atributos

Os atributos podem ser classificados quanto à sua função da seguinte forma:

- Descritivos:
- Nominativos:
- Referenciais:
- Simples:
- Compostos:

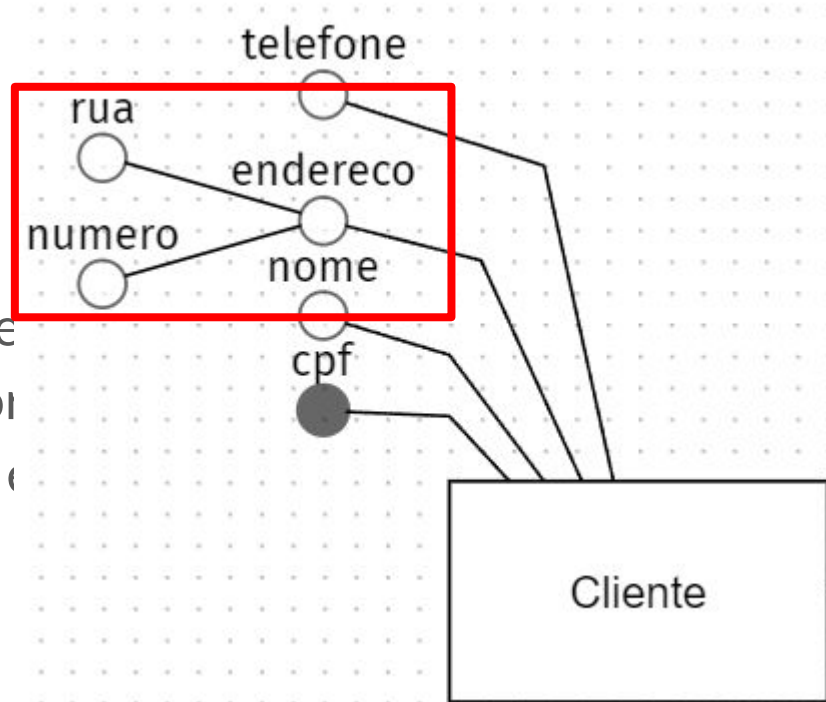


Atributos

Compostos: para definir uma informação da entidade, são usados vários atributos. Por exemplo, o endereço pode ser composto por rua, número, bairro, etc.

Atributos

Compostos: para de
vários atributos. Por
rua, número, bairro, e



são usados
mposto por



Atributos

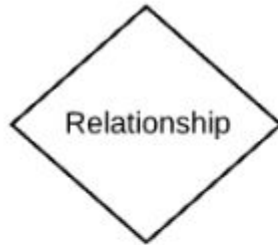
Alguns atributos representam valores únicos que identificam a entidade dentro do domínio e não podem se repetir. Em um cadastro de clientes, por exemplo, esse atributo poderia ser o CPF. A estes chamamos de Chave Primária.



Relacionamentos

Uma vez que as entidades são identificadas, deve-se então definir como se dá o relacionamento entre elas. De acordo com a quantidade de objetos envolvidos em cada lado do relacionamento, podemos classificá-los de três formas:

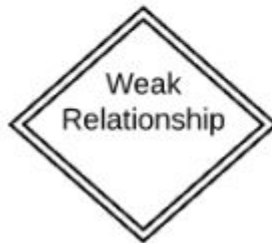
Rela



Relationship

Relationships are associations between or among entities.

Uma
como
quant
pode



Weak
relationship

Weak Relationships are connections between a weak entity and its owner.

r
a
,



Relacionamento / Cardinalidade

Relacionamento 1..1 (um para um): cada uma das duas entidades envolvidas referenciam obrigatoriamente apenas uma unidade da outra. Por exemplo, em um banco de dados de currículos, cada usuário cadastrado pode possuir apenas um currículo na base, ao mesmo tempo em que cada currículo só pertence a um único usuário cadastrado.

Relacionamento / Cardinalidade

Relacionamento 1..1 (um para um): cada uma das duas entidades envolvidas possui uma única instância da outra. Por exemplo, em uma base de dados, cada usuário cadastrado possui um único currículo associado a ele no mesmo tempo em que cada currículo só pertence a um único usuário cadastrado.





Relacionamento / Cardinalidade

Relacionamento 1..n ou 1..* (um para muitos): uma das entidades envolvidas pode referenciar várias unidades da outra, porém, do outro lado cada uma das várias unidades referenciadas só pode estar ligada uma unidade da outra entidade

Relacionamento / Cardinalidade

Relacionamento 1..n ou 1..* (um para muitos): Por exemplo, em um sistema de dependência, cada usuário pode ter uma ou mais dependências, mas cada dependência pertence a um único usuário. Note que o usuário é o principal e a dependência é o dependente. O que muda é a quantidade de unidades/exemplares envolvidas de cada lado.





Relacionamento / Cardinalidade

Relacionamento n..n ou *.* (muitos para muitos): neste tipo de relacionamento cada entidade, de ambos os lados, podem referenciar múltiplas unidades da outra.



Relacionamento / Cardinalidade

Relacionamento n..n ou *.* (muitos para muitos): Por exemplo, em um sistema de biblioteca, um título pode ser escrito por vários autores, ao mesmo tempo em que um autor pode escrever vários títulos. Assim, um objeto do tipo autor pode referenciar múltiplos objetos do tipo título, e vice-versa.

Relacionamento / Cardinalidade

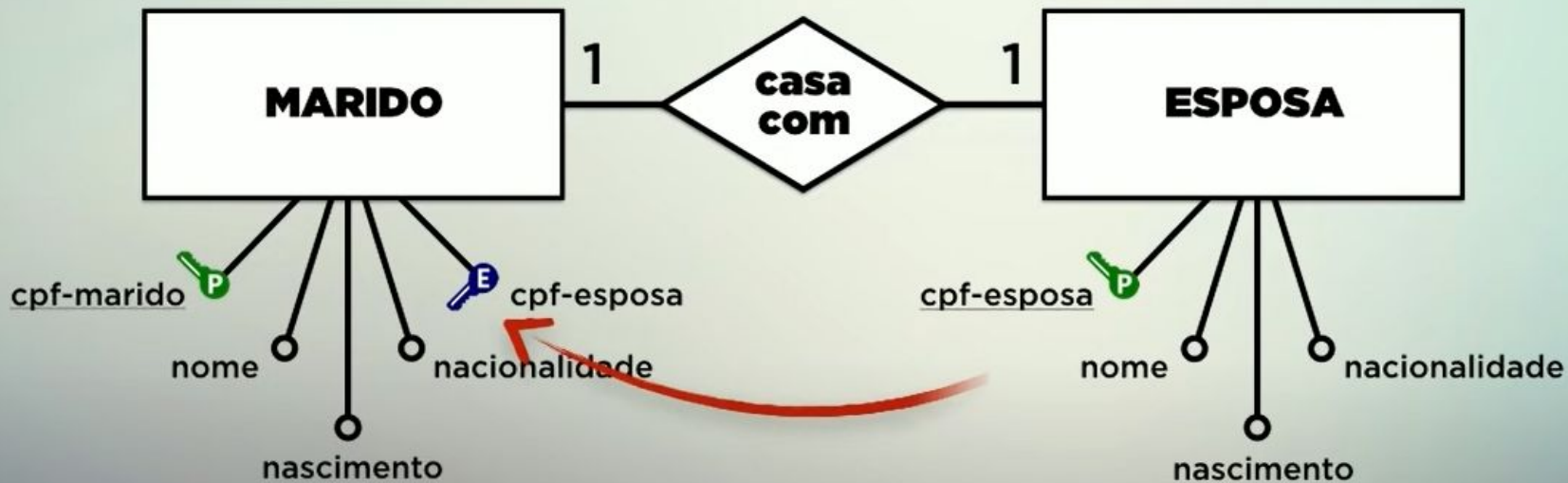
Relacionam um sistema de autores com um sistema de títulos. Assim, um objeto do tipo autor pode referenciar múltiplos objetos do tipo título, e vice-versa.



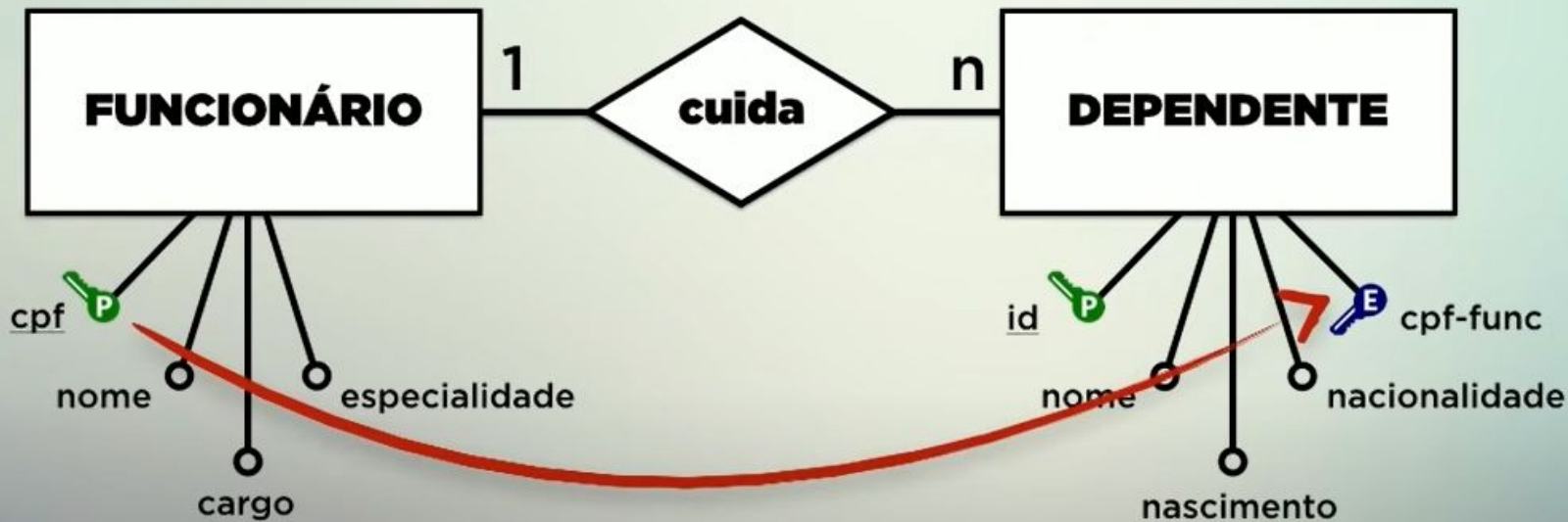


Relacionamento / Cardinalidade

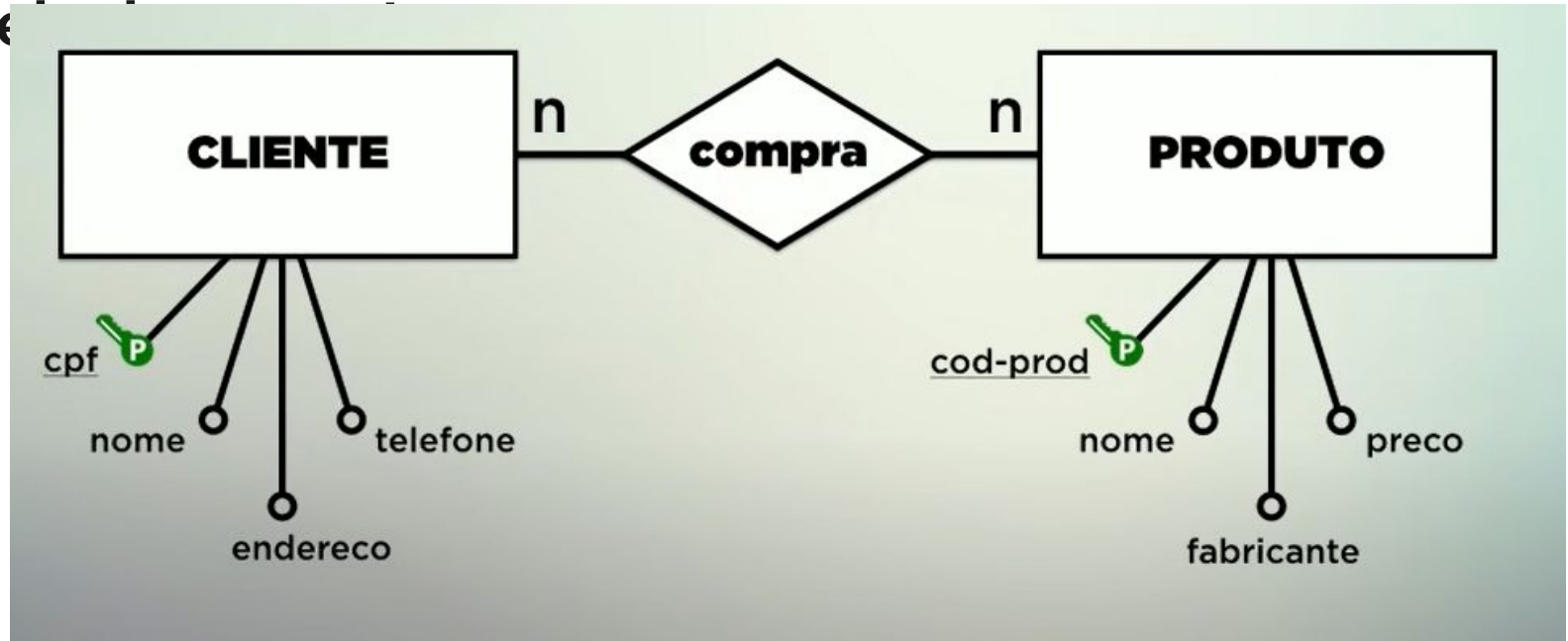
Os relacionamentos em geral são nomeados com verbos ou expressões que representam a forma como as entidades interagem, ou a ação que uma exerce sobre a outra. Essa nomenclatura pode variar de acordo com a direção em que se lê o relacionamento. Por exemplo: um autor escreve vários livros, enquanto um livro é escrito por vários autores.



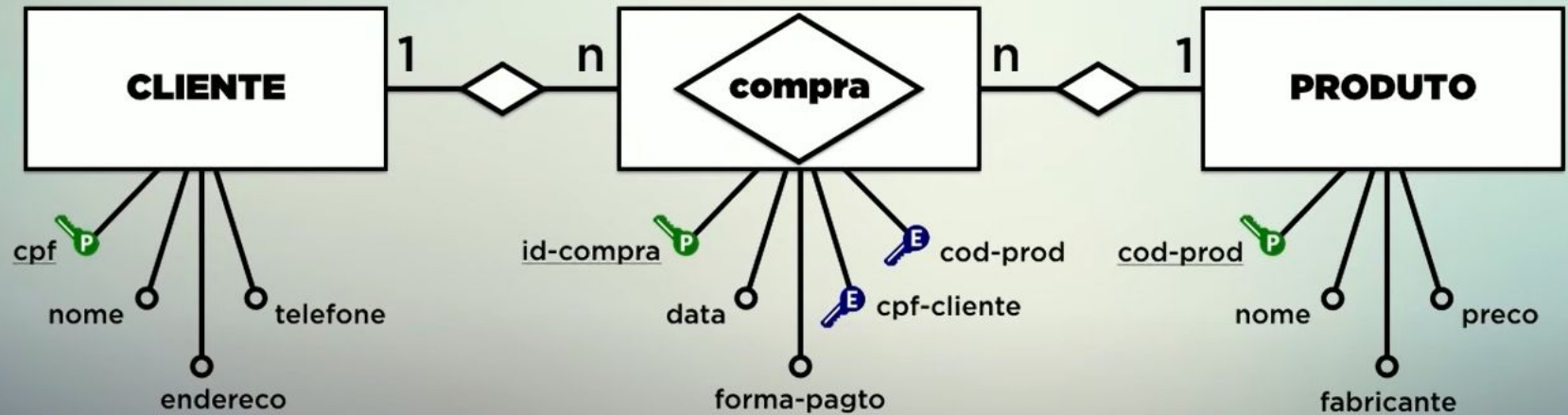
R



Re



Relacionamento n ... n





<div>SUA LOGO</div>		NOME DA EMPRESA	
		(11) 9 0000-0000	(11) 0000-0000
Endereço:		CNPJ:	Inscr. Est.:
Data Emissão ____/____/____		Pedido	
Nome: _____			
End _____ N° _____			
CNPJ _____		Inscr. Est. _____	
Quant.	Descrição das Mercadorias	Unid.	Valor
TOTAL R\$			
Assinatura do Cliente			

Modelo Lógico - Diagrama ER para Banco de Dados



One



Many



One (and only one)



Zero or one

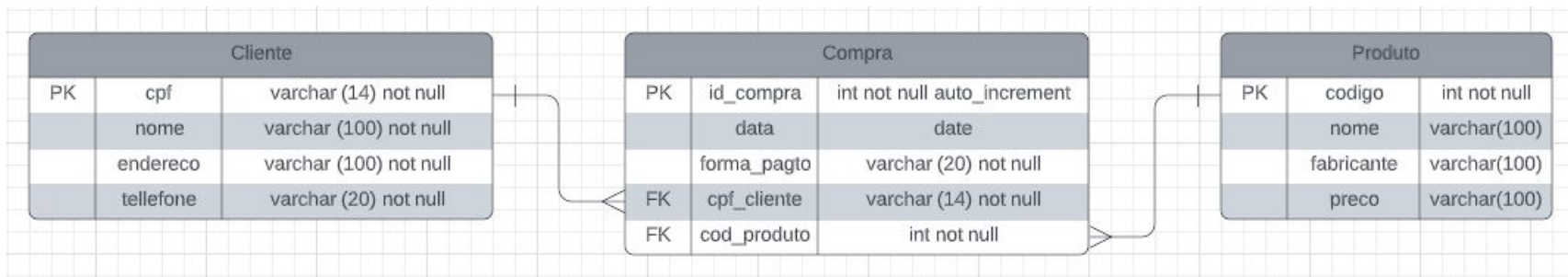


One or many



Zero or many

Modelo Lógico - Diagrama ER para Banco de Dados



Modelo Lógico - Diagrama ER para Banco de Dados

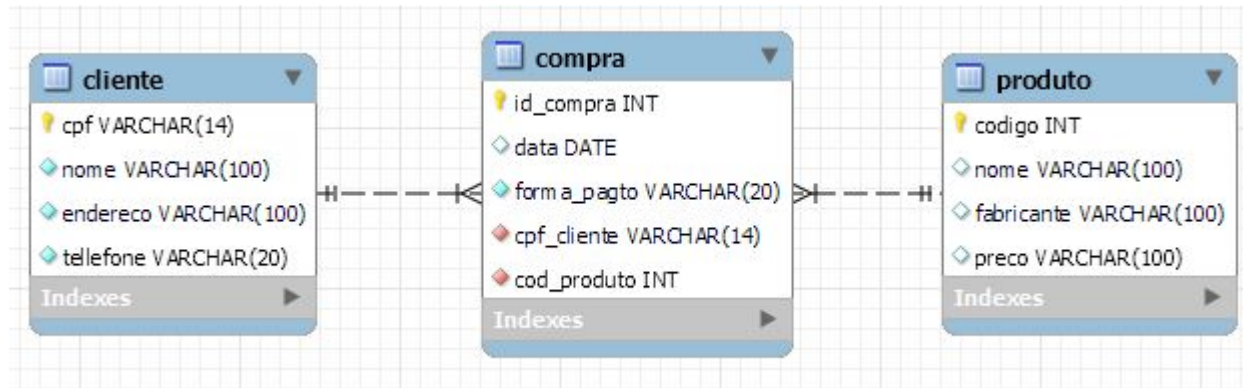




Diagrama ER para Banco de Dados

Criar um modelo conceitual de ER (<https://app.brmodeloweb.com/>) depois utilizando o app (<https://www.lucidchart.com/>) criar um modelo lógico utilizando *Data Base ER Diagram template* do exercício abaixo:

Uma biblioteca quer manter informações sobre seus livros. Deseja-se armazenar para os livros as seguintes características: ISBN, título, ano, editora e autor do livro. Para os autores, deseja-se manter: nome e nacionalidade. Cada livro da biblioteca pertence a uma categoria, então, a biblioteca deseja manter um cadastro de todas as categorias existentes, com informações como: código da categoria e descrição.

Inicialmente a biblioteca conta com os seguintes livros:

Di
Uti
de

ISBN	Título	Ano	Editora	Autor/Nacionalidade	Categoria
8532511015	Harry Potter e A Pedra Filosofal	2000	Rocco	J. K. Rowling (Inglaterra)	Literatura Juvenil
9788578270698	As Crônicas de Nárnia	2009	Wmf Martins Fontes	Clive Staples Lewis (Inglaterra)	Literatura Juvenil
9788577343348	O Espadachim de Carvão	2013	Casa da Palavra	<u>Affonso Solano</u> (Brasil)	Ficção Científica
9788581742458	O Papai É Pop	2015	Belas Letras	Marcos <u>Piangers</u> (Brasil)	Humor
9788582302026	Pior Que Tá Não Fica	2015	Matrix	Ciro Botelho – Tiririca (Brasil)	Humor
9788577345670	Garota Desdobrável	2015	Casa da Palavra	Bianca <u>Mól</u> (Brasil)	Literatura Juvenil
8532512062	Harry Potter e o prisioneiro de Azkaban	2000	Rocco	J. K. Rowling (Inglaterra)	Literatura Juvenil

ICO



Diagrama ER para Banco de Dados

1. Relacione as entidades e seus respectivos atributos para este banco de dados;
2. Crie o modelo conceitual desde banco de dados;
3. Crie o modelo lógico deste banco de dados;
4. Crie o comando SQL para a criação do banco de dados BIBLIOTECA;
5. Crie o comando SQL para a criação das tabelas solicitadas;
6. Crie o comando SQL para inserção dos referidos dados em todas as tabelas;
7. Crie uma consulta para relacionar todos os dados disponíveis de todos os livros existentes na biblioteca em ordem alfabética de título;
8. Crie uma consulta para relacionar todos os dados disponíveis de todos os livros existentes na biblioteca em ordem alfabética de Autor;
9. Crie uma consulta para relacionar todos os dados disponíveis dos livros da categoria de literatura Juvenil em ordem de ano;
10. Crie uma consulta para relacionar todos os dados disponíveis dos livros de humor ou ficção científica com ano entre 2000 e 2010.



Bibliografia

- Christopher John Date. An introduction to database systems. Pearson Education India, 1981 (ver página 27).
- Ramez Elmasri e Sham Navathe. Fundamentals of Database Systems. 7ª edição. Pearson, 2016 (ver páginas 7, 8, 10, 16, 17, 19, 20).
- Nenad Jukic, Susan Vrbsky e Svetlozar Nestorov. Database systems: Introduction to databases and data warehouses. Pearson, 2014 (ver página 23).
- Michael McLaughlin. MySQL Workbench: Data Modeling Development. McGraw Hill Professional, 2013 (ver página 23).
- SQL Tutorial. w3schools, 2022. Disponível em: <https://www.w3schools.com/sql/default.asp>. Acesso em: 20, abril de 2022