

Modelo Relacional

Prof. Dr. Carina F. Dorneles
dorneles@inf.ufsc.br
INE/CTC

INE 5423 – Banco de Dados I

Projeto de Banco de Dados

▶ Modelagem conceitual

- ▶ Descrição mais abstrata da base de dados.
- ▶ Não contém detalhes de implementação.
- ▶ Independente de tipo de SGBD usado.
- ▶ Ponto de partida do projeto da base de dados.

Pode ser representado por:

- ER
- UML

▶ Projeto lógico

- ▶ Descrição da base de dados como vista pelos usuários do SGBD (programadores, usuários que tem acesso ao BD).
- ▶ Dependente de SGBD.
- ▶ Não contém detalhes físicos de implementação (índices, etc.): SGBD oferece abstração de dados, independência de dados.

Pode ser representado por:

- Relacional
- Orientado a Objetos
- Objeto-relacional
- XML

▶ Projeto físico (interno)

- ▶ Descrição da base de dados como armazenado internamente (ajuste de performance).
- ▶ Tendência em produtos modernos é cada vez mais esconder o modelo

Não existe representação formal

Modelo Lógico

- ▶ Representação do modelo conceitual no Banco de Dados
 - ▶ Define como os objetos representados no modelo conceitual pode ser representados logicamente no Banco de Dados
 - ▶ Possui a visão daqueles que usam o do Banco de Dados
- ▶ Podem ser representado como
 - ▶ Relacional
 - ▶ Exemplo: Oracle, Postgresql, Mysql, Firebird... (maioria dos SGBDs)
 - ▶ Objeto-Relacional
 - ▶ Exemplo: Oracle, BD2, SQL Server, Postgresql
 - ▶ Orientado a Objetos
 - ▶ Exemplo: O2, DB4O, Caché
 - ▶ XML
 - ▶ Exemplo: eXist, Tamino

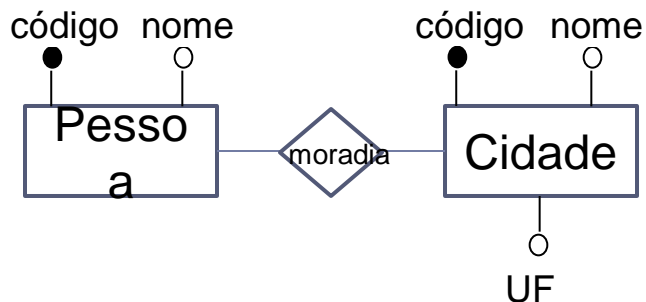
Modelo Relacional

- ▶ Dados são guardados em tabelas (ou relações)
- ▶ Definição teórica baseada nas
 - ▶ Lógica de predicados
 - ▶ Teoria dos conjuntos
- ▶ Criado por **Edgar Frank Codd** em 1970
- ▶ Modelos anteriores a ele
 - ▶ modelo hierárquico
 - ▶ modelo em rede ou *CodasyI*
 - ▶ modelo de listas invertidas



Tabelas

- ▶ O modelo relacional representa o banco de dados como uma coleção de tabelas (ou relações)
- ▶ Basicamente, cada entidade representada no modelo conceitual é uma tabela no modelo relacional.



<i>Pessoa</i>	
codigo	nome

<i>Cidade</i>		
codigo	nome	UF

Tabelas

- ▶ Cada **linha** representa um objeto da coleção de dados
- ▶ Cada **coluna** representa um atributo do objeto
- ▶ Na terminologia do modelo relacional formal:
 - ▶ Cada **linha** (da tabela) é uma **tupla** (da relação)
 - ▶ Cada **coluna** (da tabela) é um **atributo** (da relação)

<i>Cidade</i>		
codigo	nome	UF
1	Florianópolis	SC
2	Joinville	SC
3	Palhoça	SC
4	Porto Alegre	RS

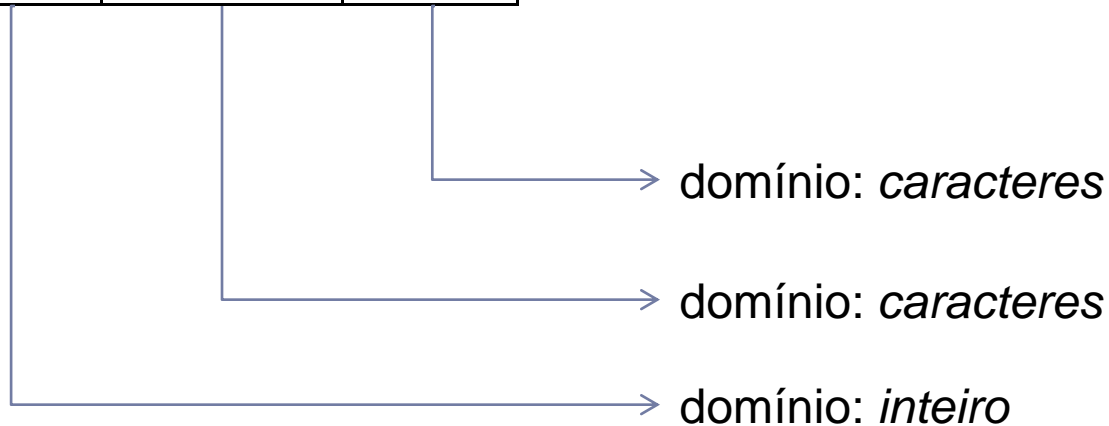


Tipos de Dados

- ▶ Cada atributo aceita um conjunto de valores:
domínio

Cidade

codigo	nome	UF
1	Florianópolis	SC
2	Joinville	SC
3	Palhoça	SC
4	Porto Alegre	RS



Valor NULL

- ▶ Um atributo pode possuir valor NULL
 - ▶ **Inexistência** de valor (*não é o mesmo que vazio*)

Cidade

codigo	nome	UF
1	Florianópolis	NULL
2	Joinville	SC
3	Palhoça	SC
4	Porto Alegre	RS



Chave primária

- **Atributo**, ou combinação de atributos, usado para **distinguir** o valor de **uma linha** dos valores das demais linhas de uma tabela

Cidade

codigo	nome	UF
1	Florianópolis	SC
2	Joinville	SC
3	Palhoça	SC
4	Porto Alegre	RS
5	Pedra Branca	CE
6	Pedra Branca	SE

O único campo capaz de distinguir o valor de uma linha dos valores das demais é **codigo**

codigo é **chave primária** (PK) de cidade

Qual atributo escolher?

- Valor **Único** (sem repetição)
- Valor **Não Nulo** (deve existir)
- Valor que **nunca** (ou rarissimamente), **muda**

Chave estrangeira

- Usada para definir os relacionamentos

<i>Pessoa</i>		
codigo	nome	codCid
1	Ana	1
2	Pedro	1
3	Juca	2
4	Lia	2

<i>Cidade</i>		
codigo	nome	UF
1	Florianópolis	SC
2	Joinville	SC
3	Palhoça	SC
4	Porto Alegre	RS
5	Pedra Branca	CE
6	Pedra Branca	SE

São **chaves primárias** de outras tabelas

Resumindo

- ▶ Tabelas são compostas por linhas e colunas
 - ▶ Relações são compostas por tuplas e atributos
- ▶ Colunas possuem domínios, que definem o conjunto de valores permitidos
- ▶ Toda tabela deve ter uma chave primária
 - ▶ Valor único
 - ▶ Valor não nulo
- ▶ Os relacionamentos são representados por chaves Estrangeiras
 - ▶ Chaves primárias de outras tabelas



Mapeamento ER → Relacional

Prof. Dr. Carina F. Dorneles
dorneles@inf.ufsc.br
INE/CTC

INE 5613 – Banco de Dados I

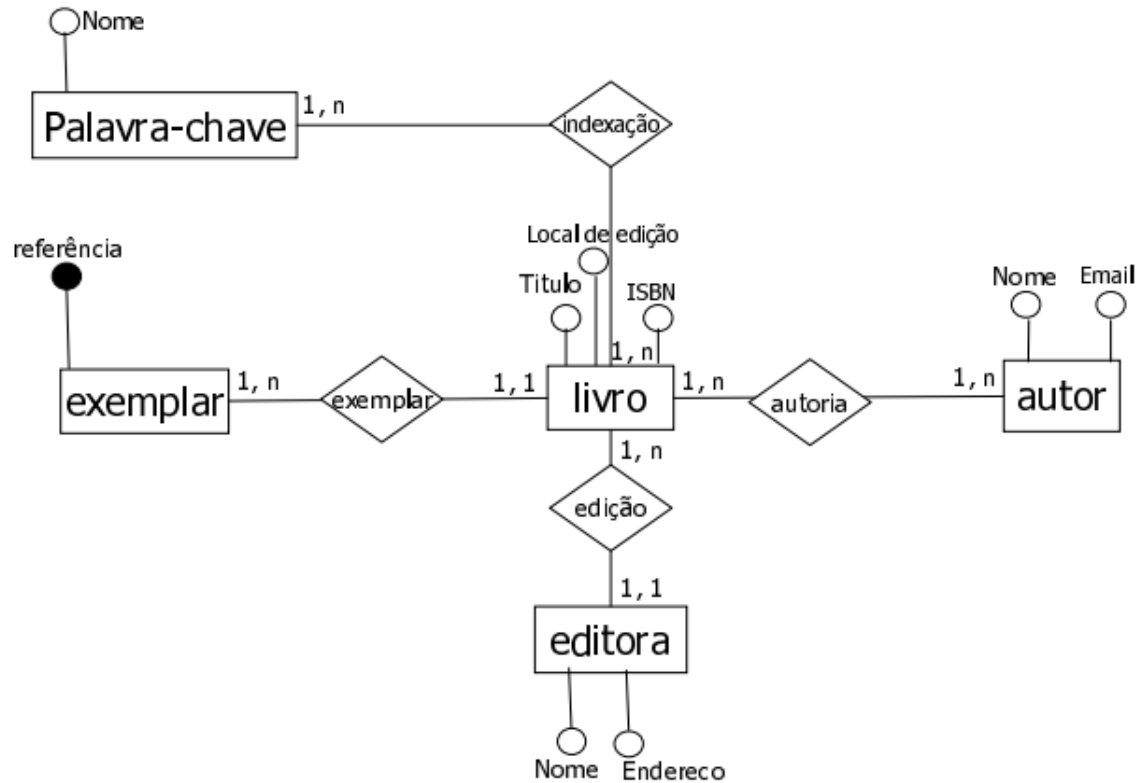
Mapeamento

► Cinco regras básicas de mapeamento

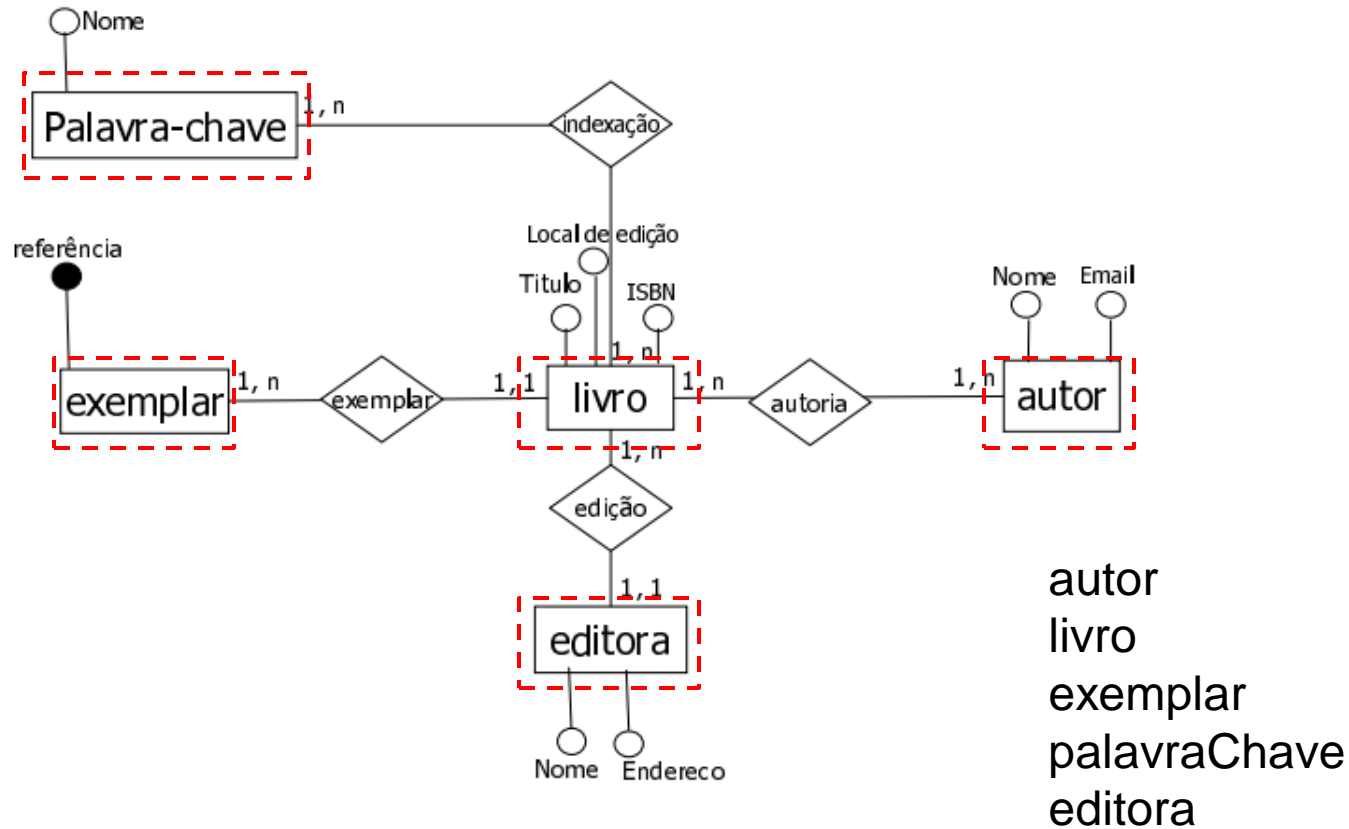
1. Entidade → criar uma tabela
2. Atributos → criar os colunas das tabelas
3. Rel. 1_n → entidade lado n recebe chave primária da entidade lado 1 (será chave estrangeira)
4. Rel. n_n → criar nova relação com as chaves primárias das entidades relacionadas (serão chaves estrangeiras)
5. Rel. 1_1 → duas alternativas:
 1. escolher qual das tabelas recebe a chave estrangeira da outra
 2. colocar todos os atributos de uma tabela na outra

- Sempre que houver a criação de uma chave estrangeira, deve-se fazer referência à tabela referenciada
-

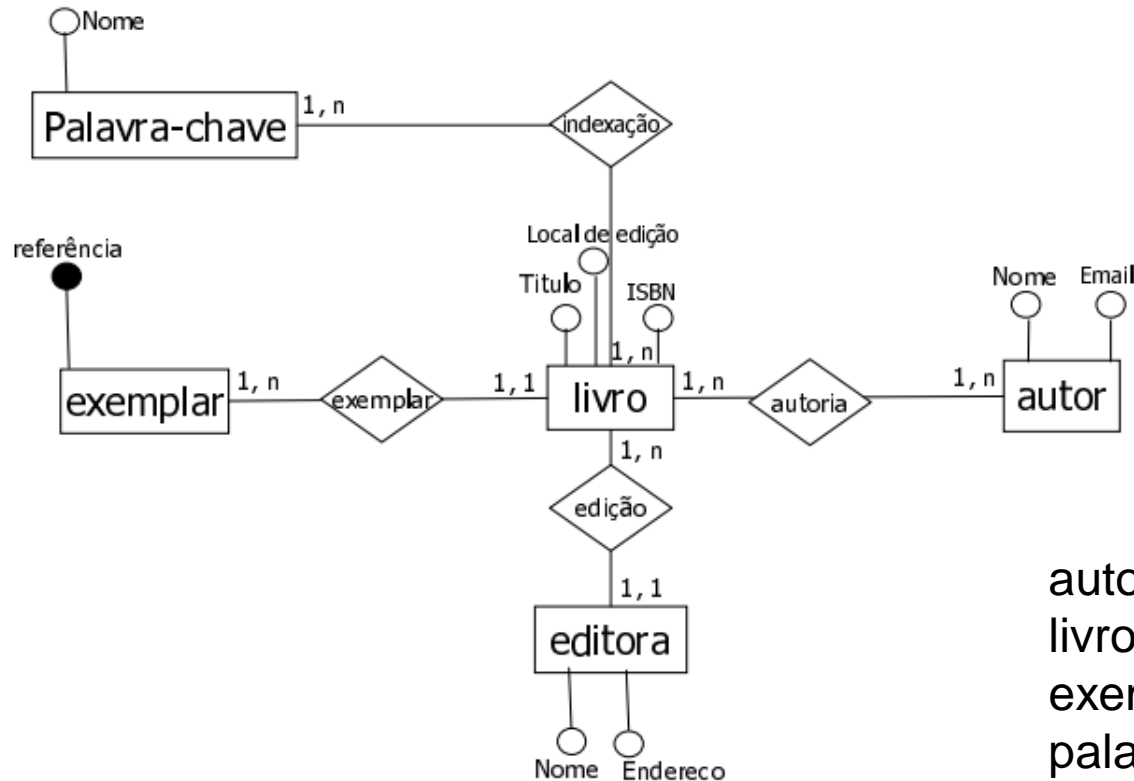
Mapeamento - exemplo



Regra 1: Entidade → tabela

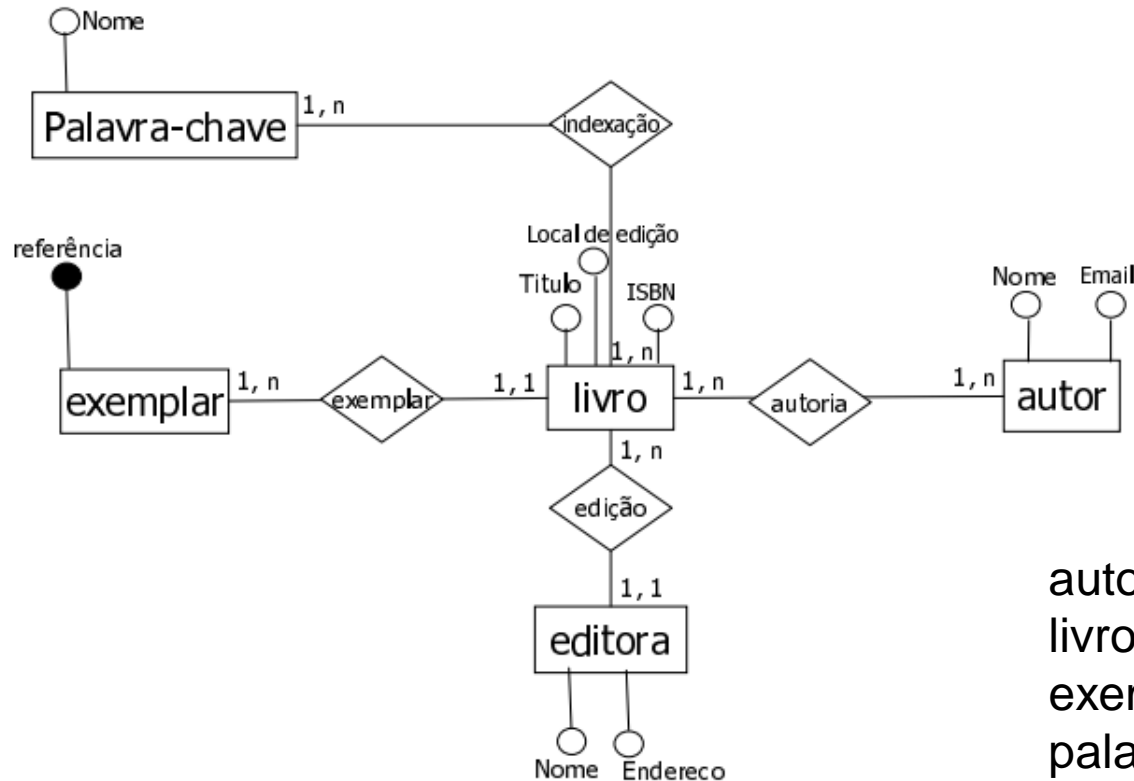


Regra 2: Atributos → colunas das tabelas



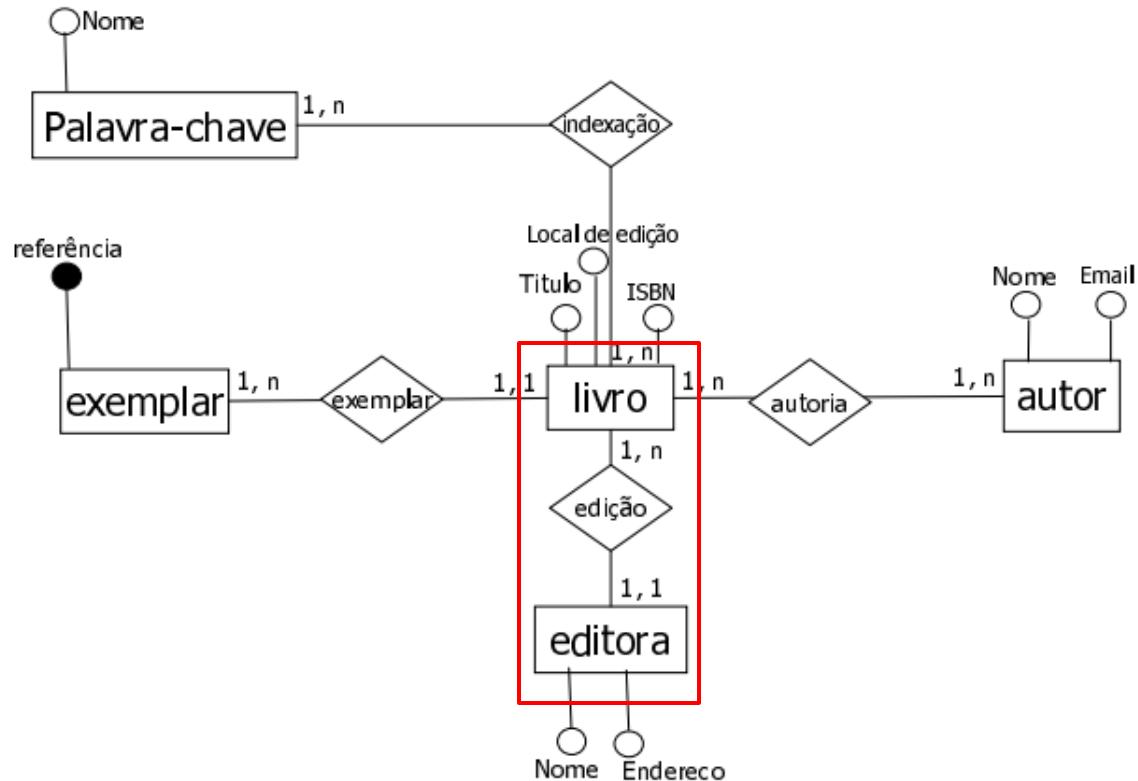
autor (nome, email)
livro (isbn, titulo, localEdicao)
exemplar (referencia)
palavraChave (nome)
editora (nome, endereco)

Antes da Regra 3: definir chaves primárias

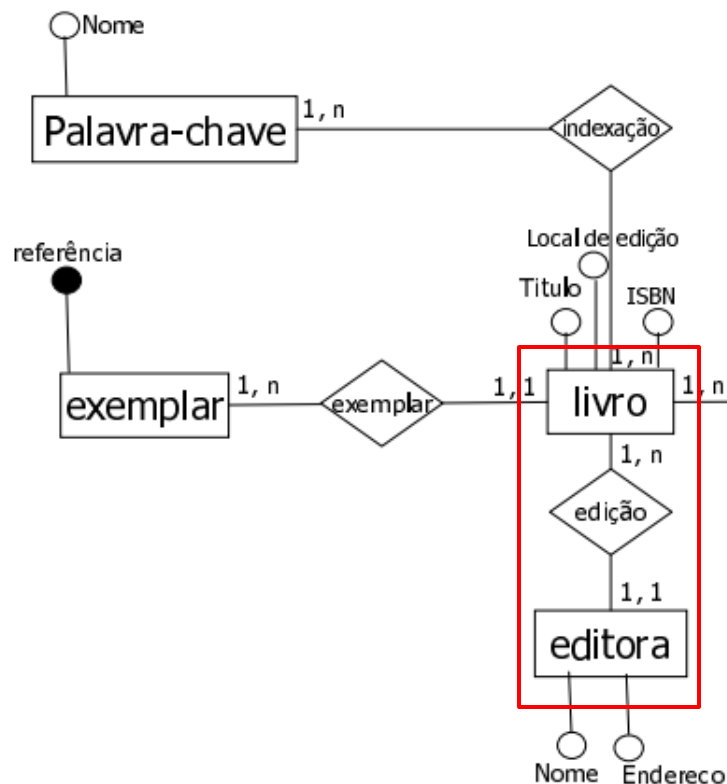


autor (codigo, nome, email)
livro (isbn, titulo, localEdicao)
exemplar (referencia)
palavraChave (codigo, nome)
editora (codigo, nome, endereco)

Regra 3: Relacionamentos 1_n



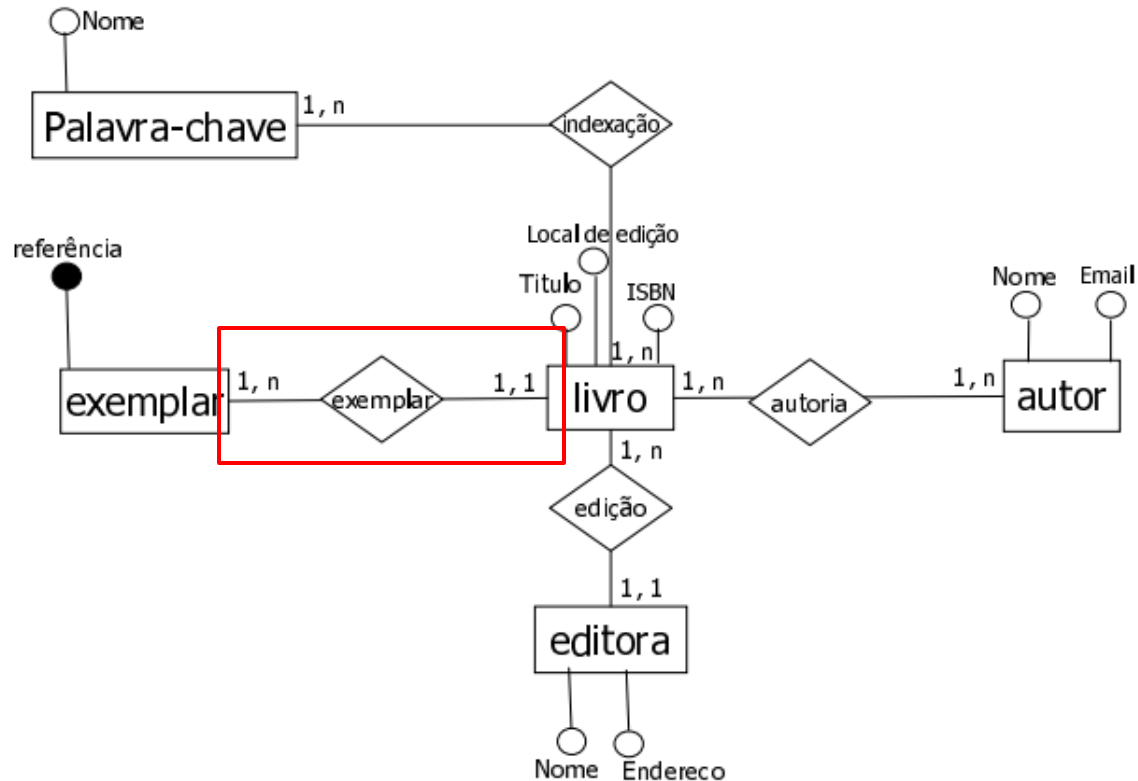
Regra 3: Relacionamentos 1_n



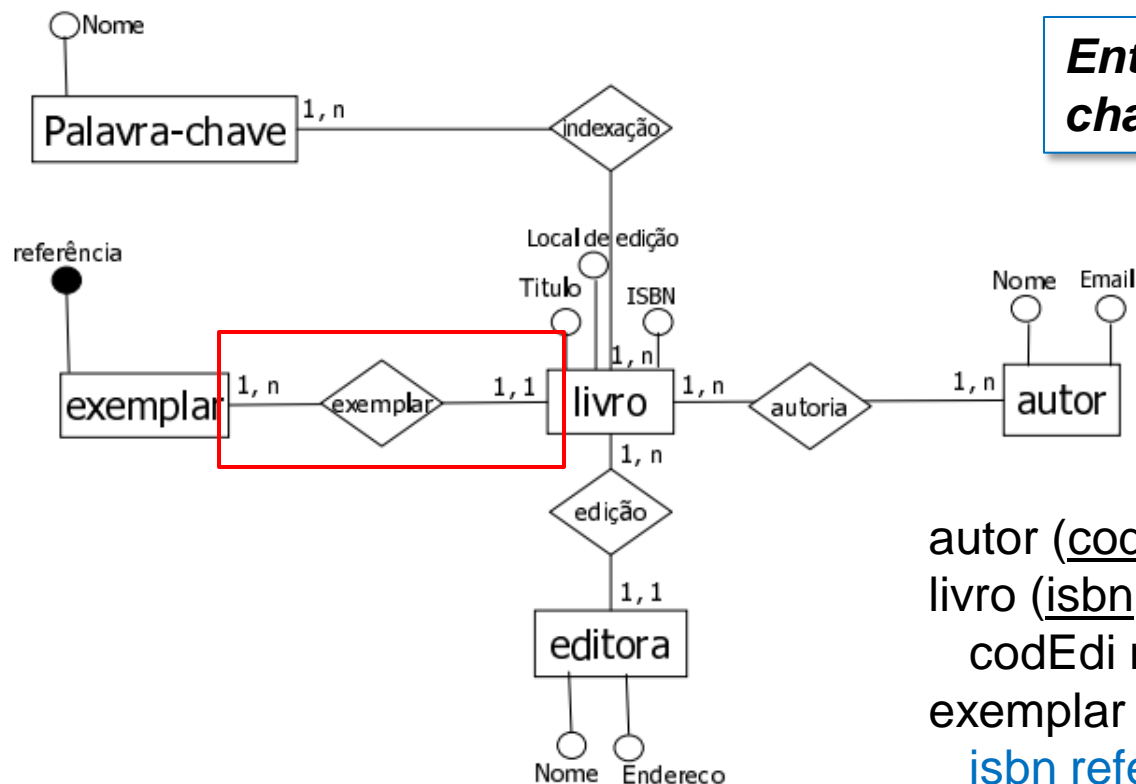
Entidade do lado n recebe a chave primária da outra

autor (codigo, nome, email)
livro (isbn, titulo, localEdicao, **codEdi#**)
codEdi referencia editora (codigo)
exemplar (referencia)
palavraChave (codigo, nome)
editora (codigo, nome, endereco)

Regra 3: Relacionamentos 1_n



Regra 3: Relacionamentos 1_n

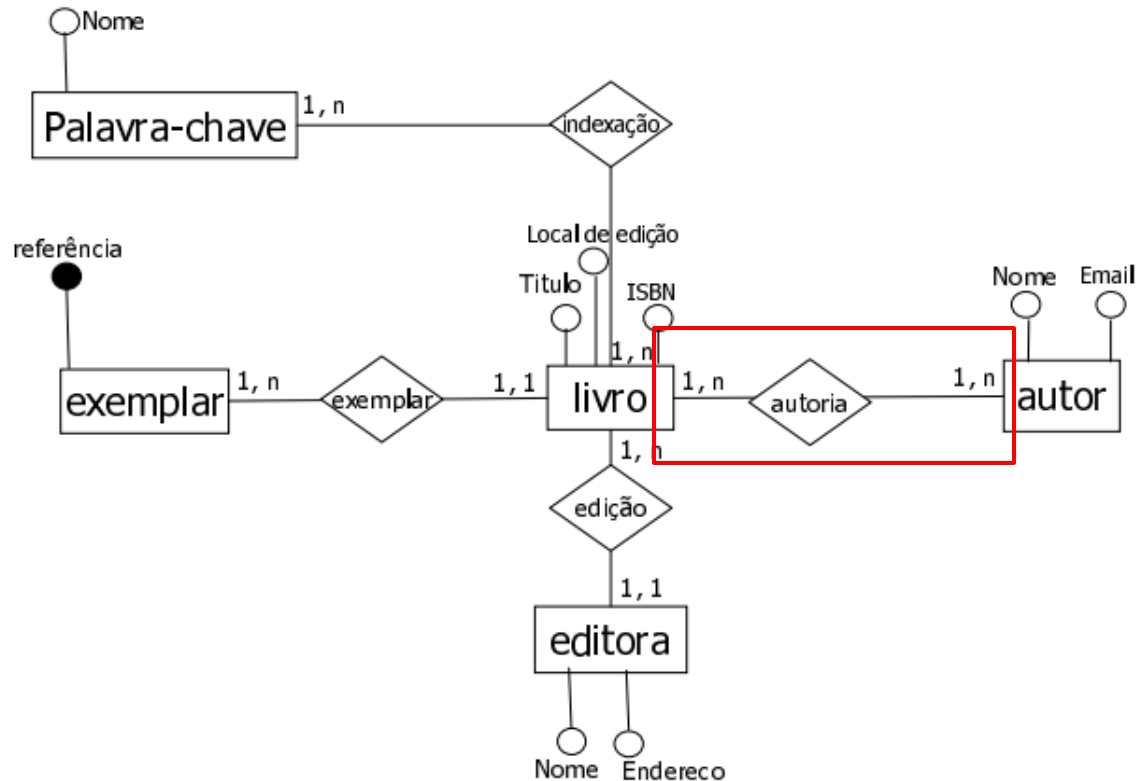


Entidade do lado n recebe a chave primária da outra

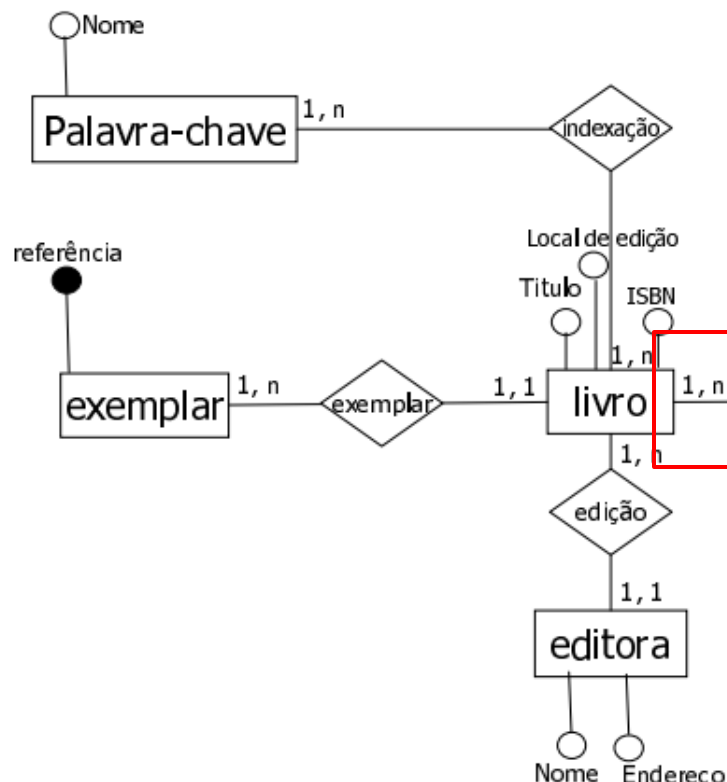
autor (codigo, nome, email)
livro (isbn, titulo, localEdicao, codEdi#)
codEdi referencia editora (codigo)
exemplar (referencia, isbn)
isbn referencia livro (isbn)
palavraChave (codigo, nome)
editora (codigo, nome, endereco)

Como exemplar é entidade fraca,
isbn passa a ser PK também

Regra 4: Relacionamentos n_n



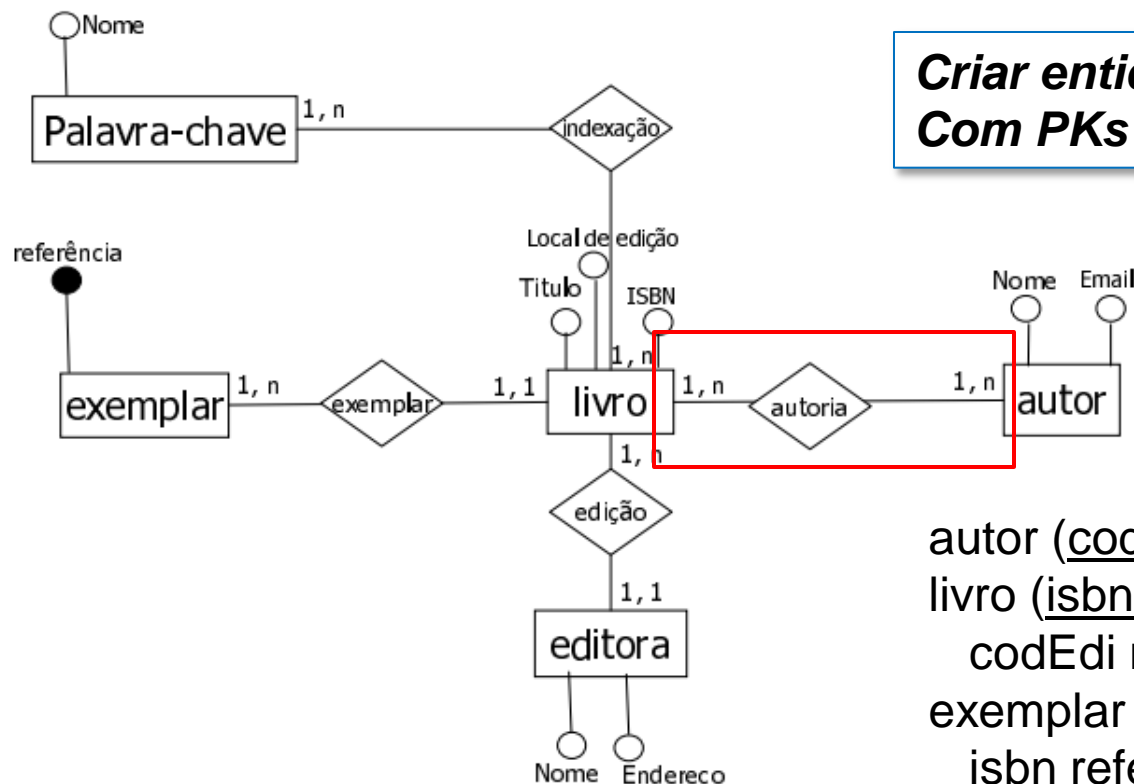
Regra 4: Relacionamentos n_n



***Criar entidade para o relacionamento
Com PKs das entidades relacionadas***

autor (codigo, nome, email)
livro (isbn, titulo, localEdicao, codEdi#)
 codEdi referencia editora (codigo)
exemplar (referencia, isbn)
 isbn referencia livro (isbn)
palavraChave (codigo, nome)
editora (codigo, nome, endereco)
autoria (codLiv#, codAut#)

Regra 4: Relacionamentos n_n

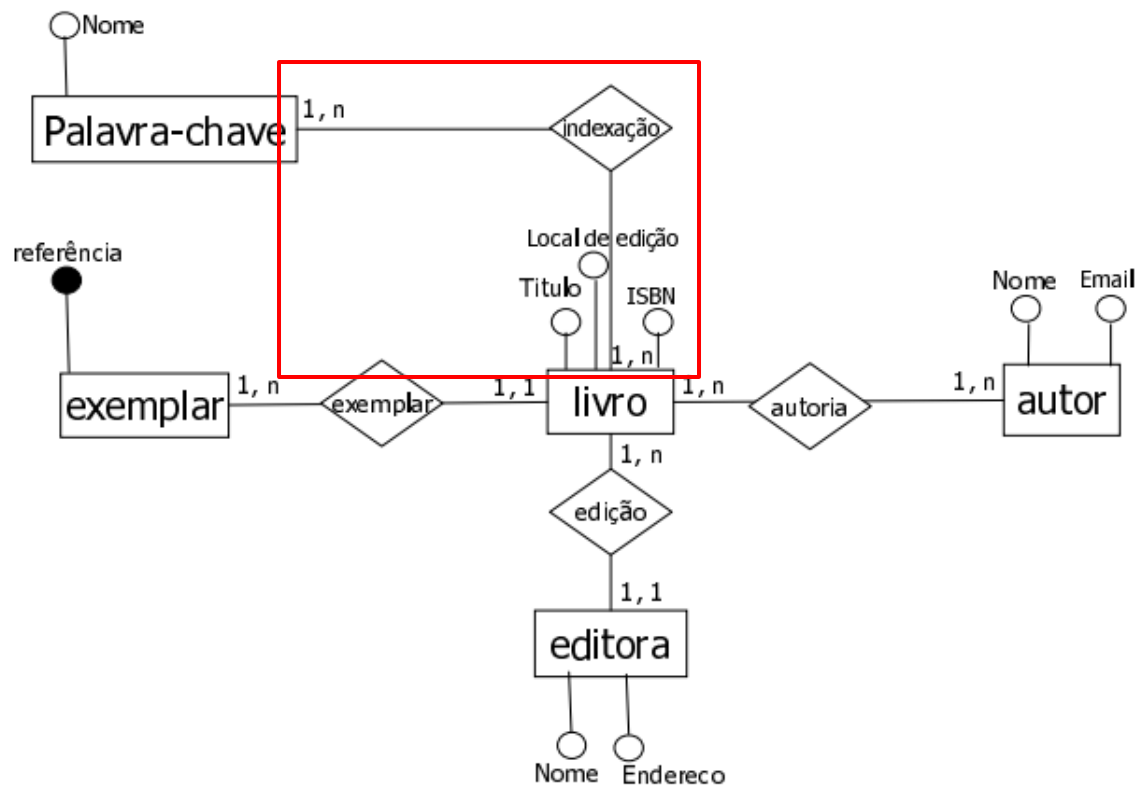


**Criar entidade para o relacionamento
Com PKs das entidades relacionadas**

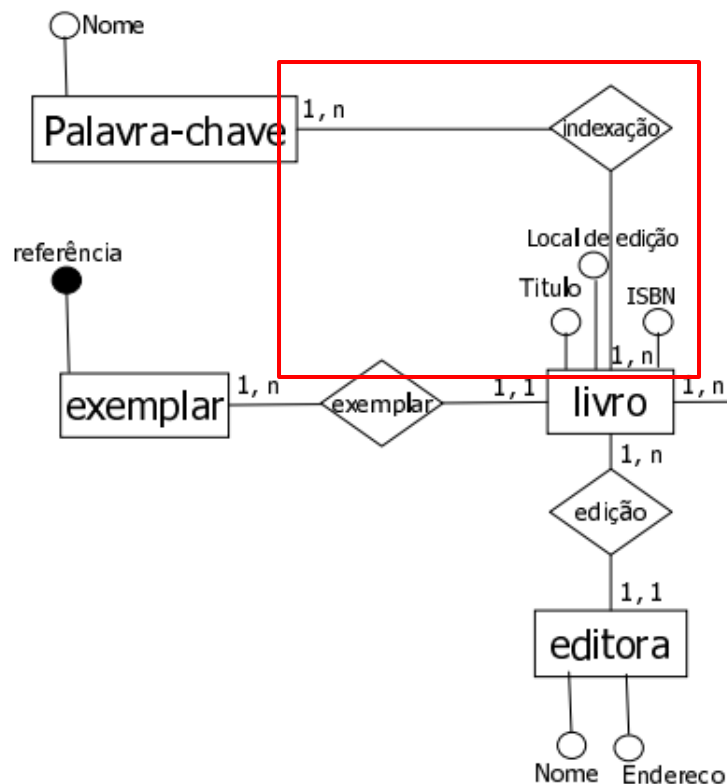
autor (codigo, nome, email)
livro (isbn, titulo, localEdicao, codEdi#)
codEdi referencia editora (codigo)
exemplar (referencia, isbn)
isbn referencia livro (isbn)
palavraChave (codigo, nome)
editora (codigo, nome, endereco)
autoria (codLiv#, codAut#)
codLiv referencia livro (isbn)
codAut referencia autor(codigo)

Ambas serão PKs da
nova entidade

Regra 4: Relacionamentos n_n



Regra 4: Relacionamentos n_n



***Criar entidade para o relacionamento
Com PKs das entidades relacionadas***

autor (codigo, nome, email)
livro (isbn, titulo, localEdicao, codEdi#)
 codEdi referencia editora (codigo)
exemplar (referencia, isbn)
 isbn referencia livro (isbn)
palavraChave (codigo, nome)
editora (codigo, nome, endereco)
autoria (codLiv#, codAut#)
 codLiv referencia livro (isbn)
 codAut referencia autor(codigo)
indexacao (codPala#, codLiv#)
 codPala referencia palavraChave(codigo)
 codLiv referencia livro(isbn)

Porque a tabela a mais?

Livro

Codigo	titulo	...	Autores
111	XML e Web	...	233, 344
222	Banco de Dados	...	122, 344
333	Aprenda PHP	...	122, 233, 344

Não é possível armazenar coleções de valores no modelo relacional



Porque a tabela a mais?

Livro

Codigo	titulo	...	Autores
111	XML e Web	...	233, 344
222	Banco de Dados	...	122, 344
333	Aprenda PHP	...	122, 233, 344

Não é possível armazenar coleções de valores no modelo relacional

Livro

Codigo	Título	ISBN
111	XML e Web	999555
222	Banco de Dados	555222
333	Aprenda PHP	111222

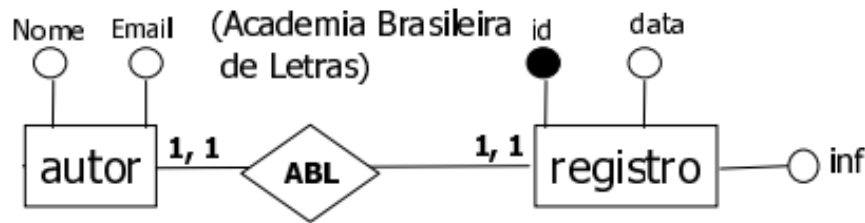
Autor

Codigo	Nome
122	Carlos Heuser
233	Alon Halevy
344	Dan Suciu

Autoria

codLiv	CodAut
111	233
111	344
222	122
222	344

Regra 5: relacionamento 1_1



registro (id, data, inf)
autor (codigo, nome, email, idReg#)
idReg referencia registro (id)

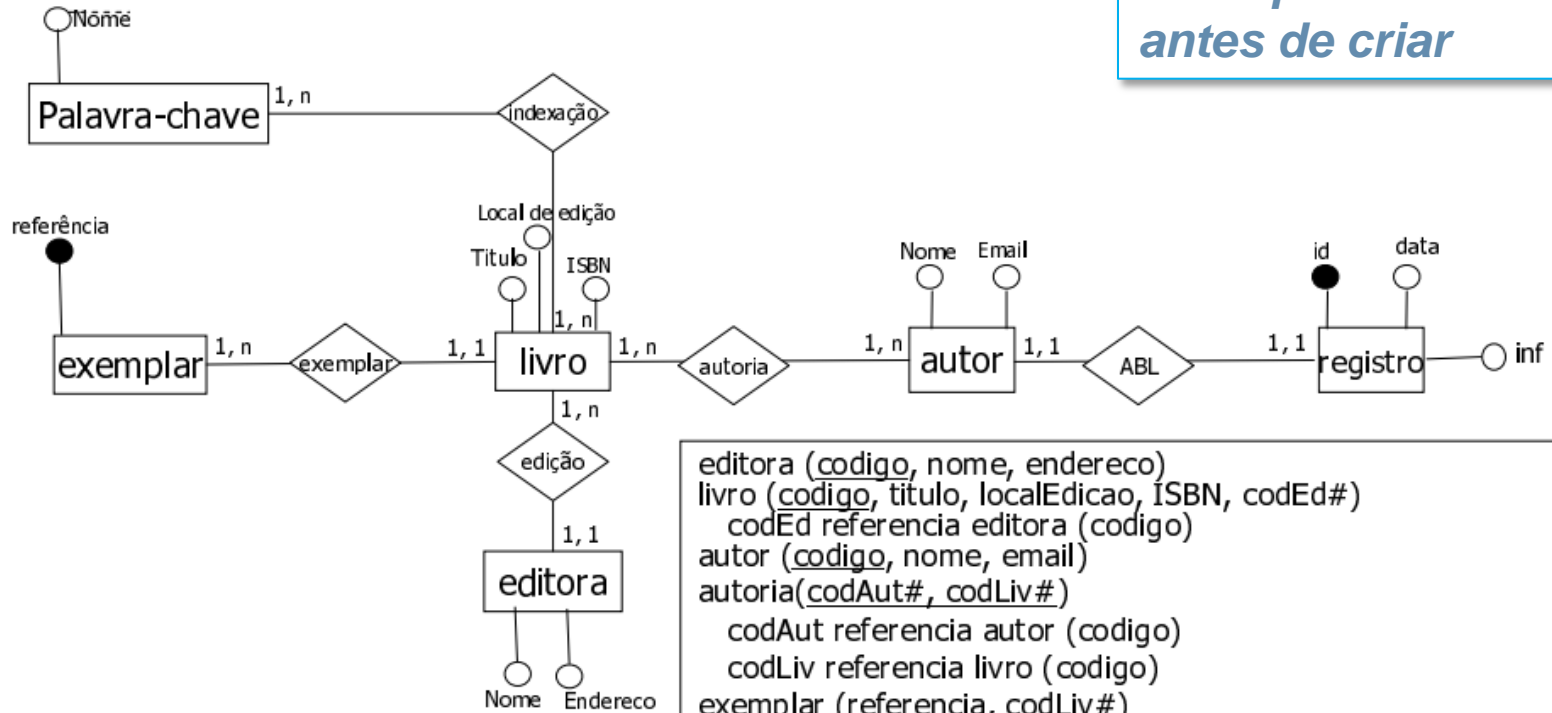
Ou

registro (id, data, inf, codAutor#)
codAutor referencia autor (codigo)



Finalmente – ordenar!!!!!!

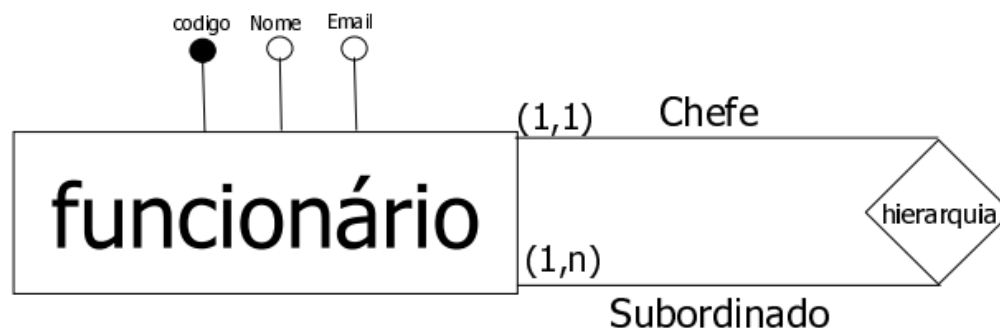
- Não pode referenciar antes de criar



editora (codigo, nome, endereco)
 livro (codigo, titulo, localEdicao, ISBN, codEd#)
 codEd referencia editora (codigo)
 autor (codigo, nome, email)
 autoria(codAut#, codLiv#)
 codAut referencia autor (codigo)
 codLiv referencia livro (codigo)
 exemplar (referencia, codLiv#)
 codLiv referencia livro (codigo)
 palavraChave (codigo, nome)
 indexacao (codLiv#, codPal#)
 codLiv referencia livro (codigo)
 codPal referencia palavraChave(codigo)
 registro (id, data, inf, codAut#)
 codAut referencia autor (codigo)

Mapeamento

► Auto-relacionamento

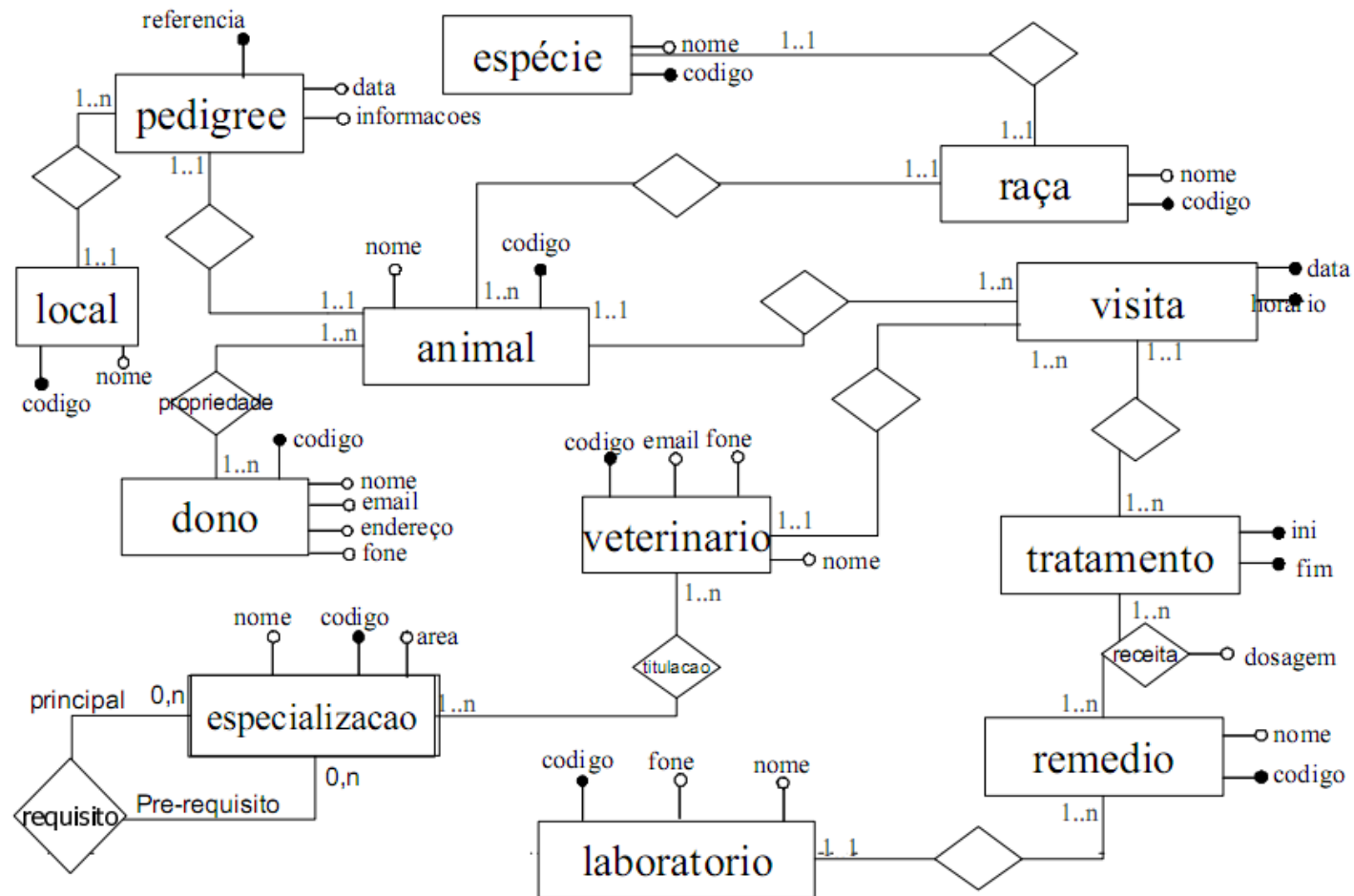


Lado n (subordinado) recebe a chave do lado 1 (chefe)

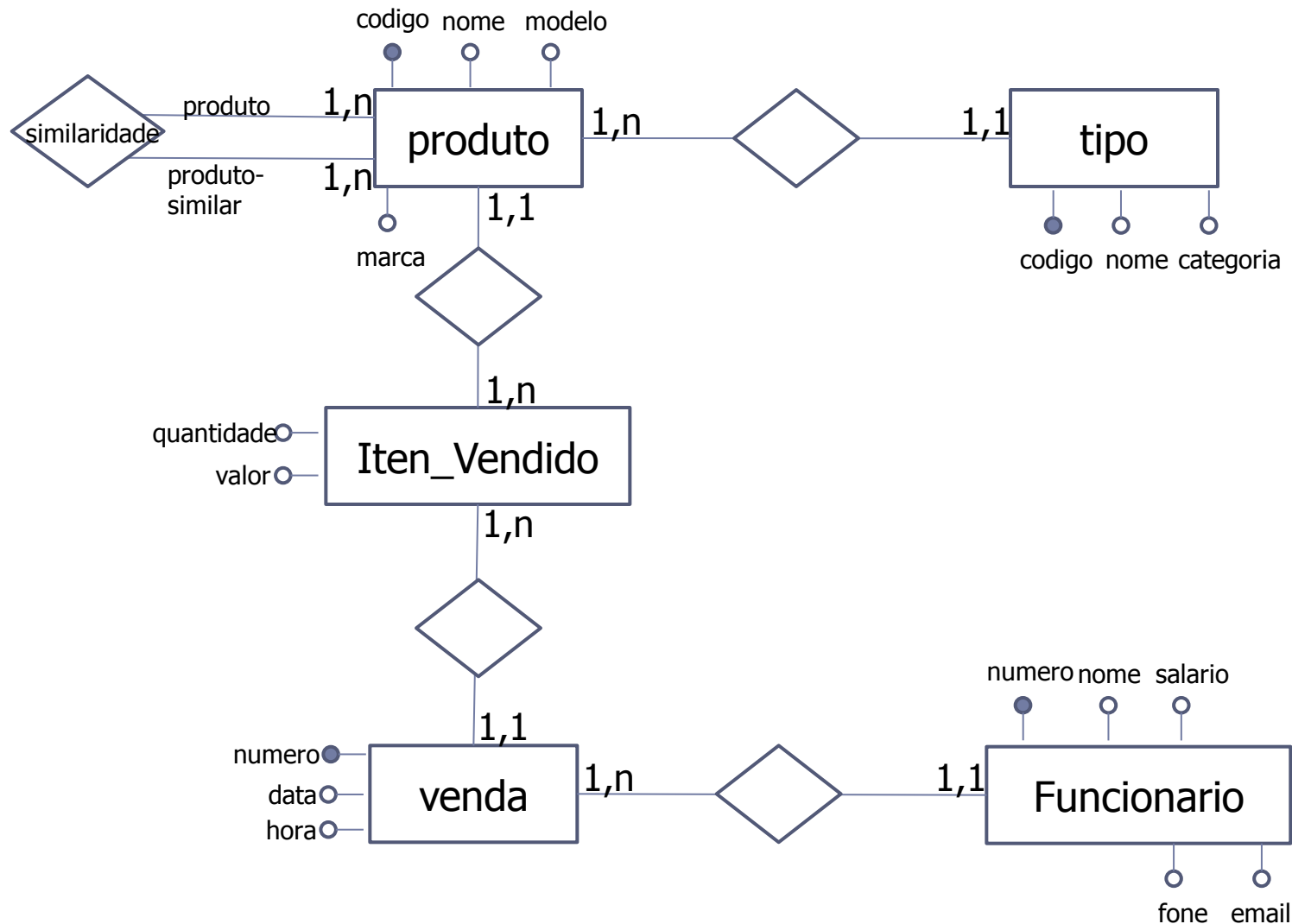
funcionário (codigo, nome, email, codChefe#)
codChefe referencia funcionario (codigo)



Exercício 1



Exercício 2



Exercício 3

