



# Modelagem de Dados

---

*Modificado em 16/04/2014*

**Bianca e Letícia**

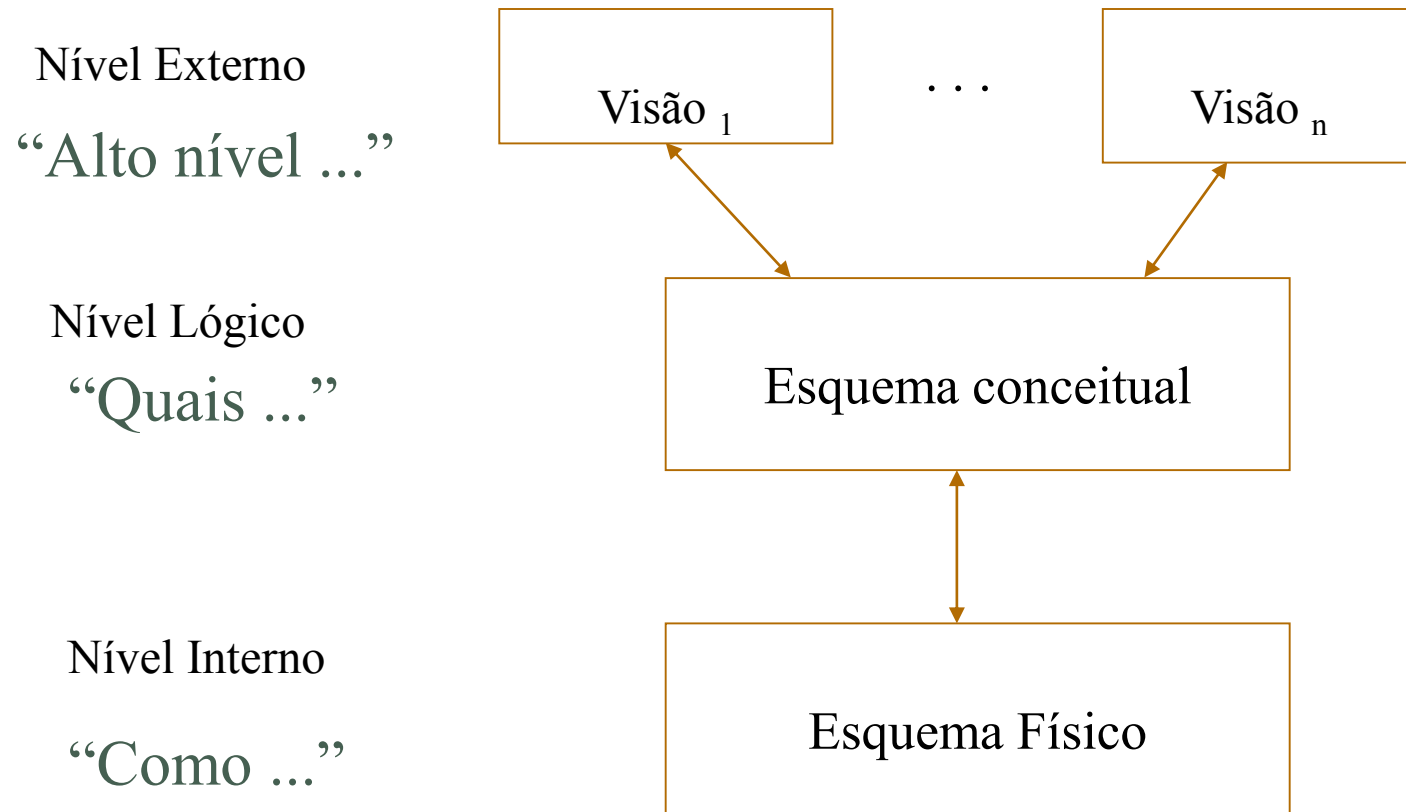
# Conteúdo

---

- ✓ Níveis de Abstração em Banco de Dados
- ✓ Modelo Entidade-Relacionamento
- ✓ Modelos Relacional
- ✓ Transformação entre Modelos
- ✓ Modelo Entidade-Relacionamento Estendido
- ✓ Diferentes Notações

# Níveis de Abstração dos Dados

---



# Modelo Entidade-Relacionamento

---

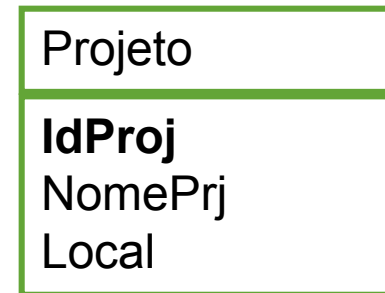
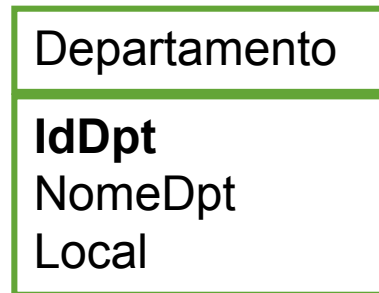
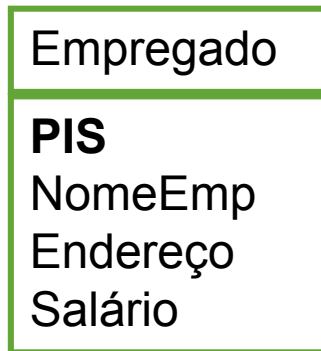
- ✓ Notação criada em 1976 por Peter Chen para representar o projeto conceitual de um BD
- ✓ Popularmente chamada de MER ou DER
- ✓ Elementos principais são as entidades e seus relacionamentos

# Modelo Entidade-Relacionamento

---

## ✓ Entidade

- um objeto com existência física (pessoa, carro, casa) ou conceitual (empresa, universidade, curso), composto por propriedades que o descreve, chamadas atributos.

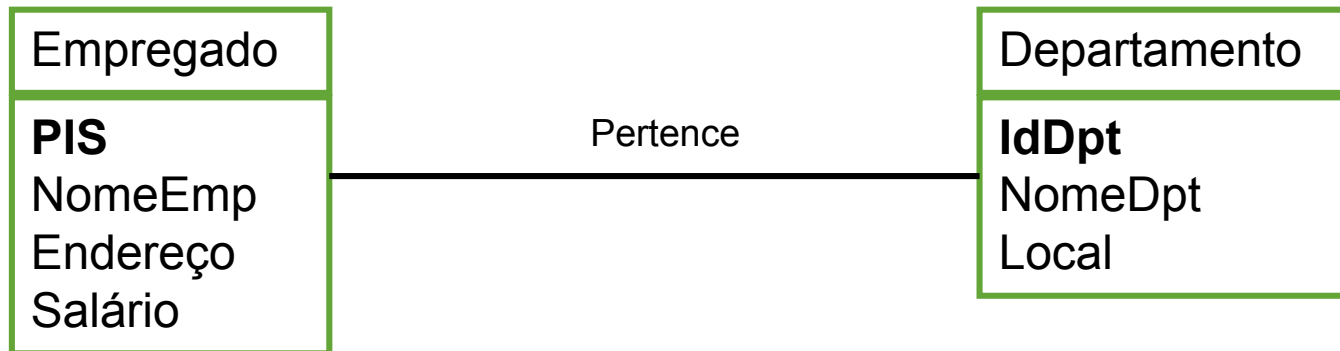


# Modelo Entidade Relacionamento

---

## ✓ Relacionamento

- associação entre duas ou mais entidades

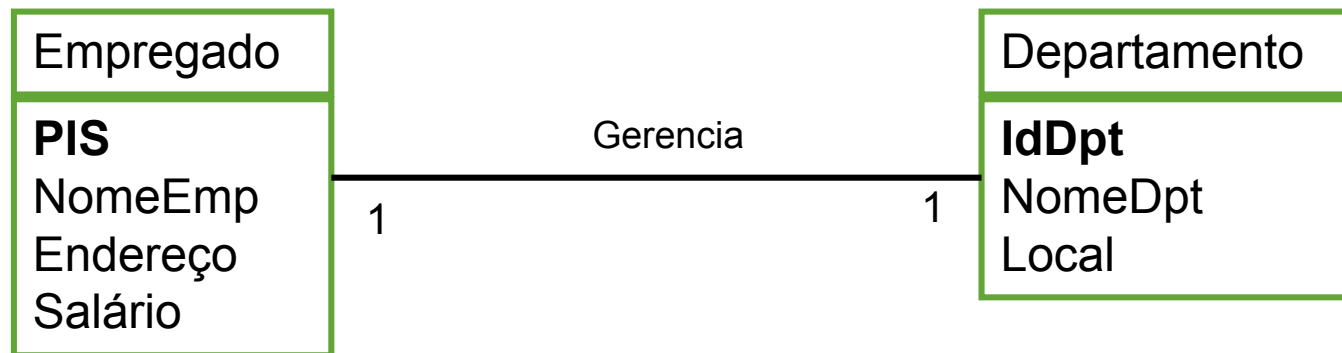


# Cardinalidade dos relacionamentos

---

## ✓ 1 para 1 (um para um)

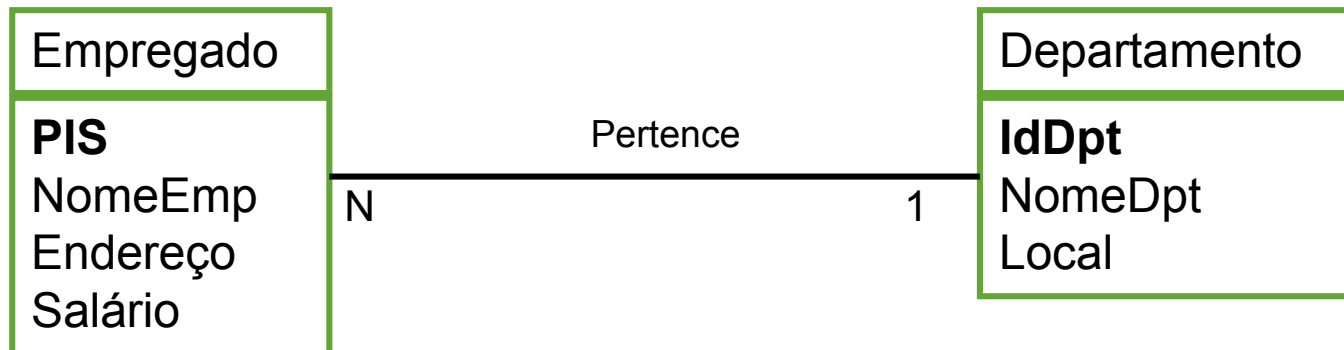
- Cada elemento de uma entidade relaciona-se com apenas um elemento da outra.
- Exemplo: cada departamento é gerenciado por um único funcionário. Um funcionário gerencia apenas um departamento.



# Cardinalidade dos relacionamentos

## ✓ 1 para N (um para muitos)

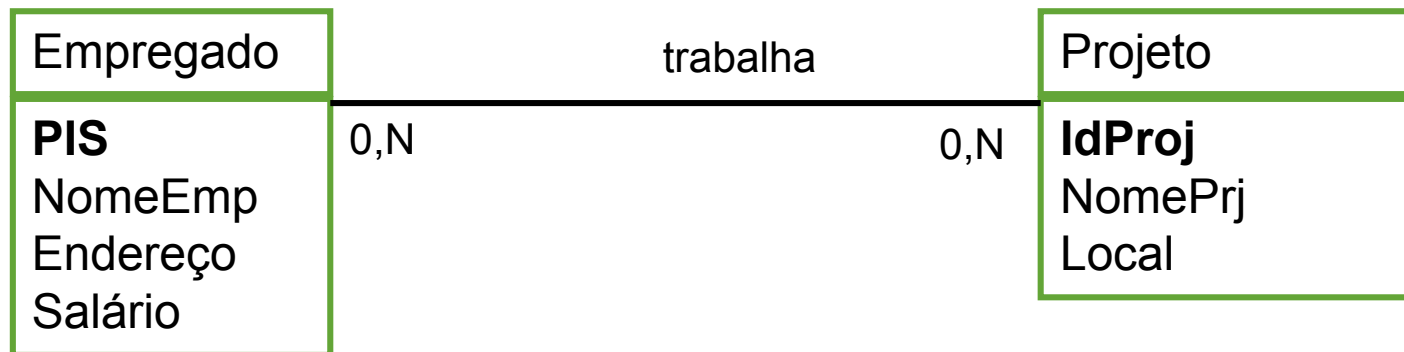
- Cada elemento de uma entidade relaciona-se com vários elementos da outra.
- Exemplo: cada departamento possui vários empregados, mas cada empregado está lotado em apenas um departamento





# Cardinalidade dos relacionamentos

- ✓ N para N (muitos para muitos)
  - Cada elemento de uma entidade relaciona-se com vários elementos da outra e vice-versa.
  - Exemplo: um empregado pode participar de vários projetos e um projeto pode ter a participação de vários empregados.



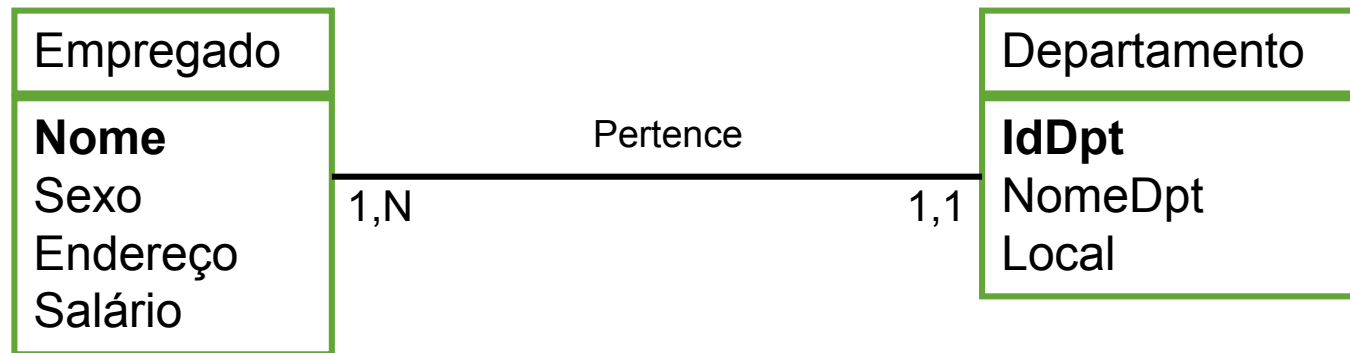
# Cardinalidade com participação

- ✓ Para representar todas as restrições de um BD, não é suficiente dizer apenas em que número (1 ou N) uma entidade aparece em um relacionamento. É necessário dizer se sua participação é opcional (0) ou obrigatória (1).
- ✓ Para isto, representa-se a cardinalidade através de um par ( min, max), onde:
  - Min indica a participação da entidade no relacionamento. Pode ser obrigatória (1) ou opcional(0).
  - Max indica o número de vezes que a entidade aparece no relacionamento. Pode ser 1 ou N(muitos).
- ✓ Possibilidades de cardinalidade:

	Um	Muitos
Opcional	(0,1)	(0,N)
Obrigatória	(1,1)	(1,N)

# Cardinalidade com Participação

- ✓ Todos os departamentos possuem pelo menos um empregado. Todo empregado trabalha para algum departamento.



# Ferramentas CASE

---

- ✓ CASE = Computer Aided Software Engineering
- ✓ Neste curso usaremos duas ferramentas case para projetos de BD:



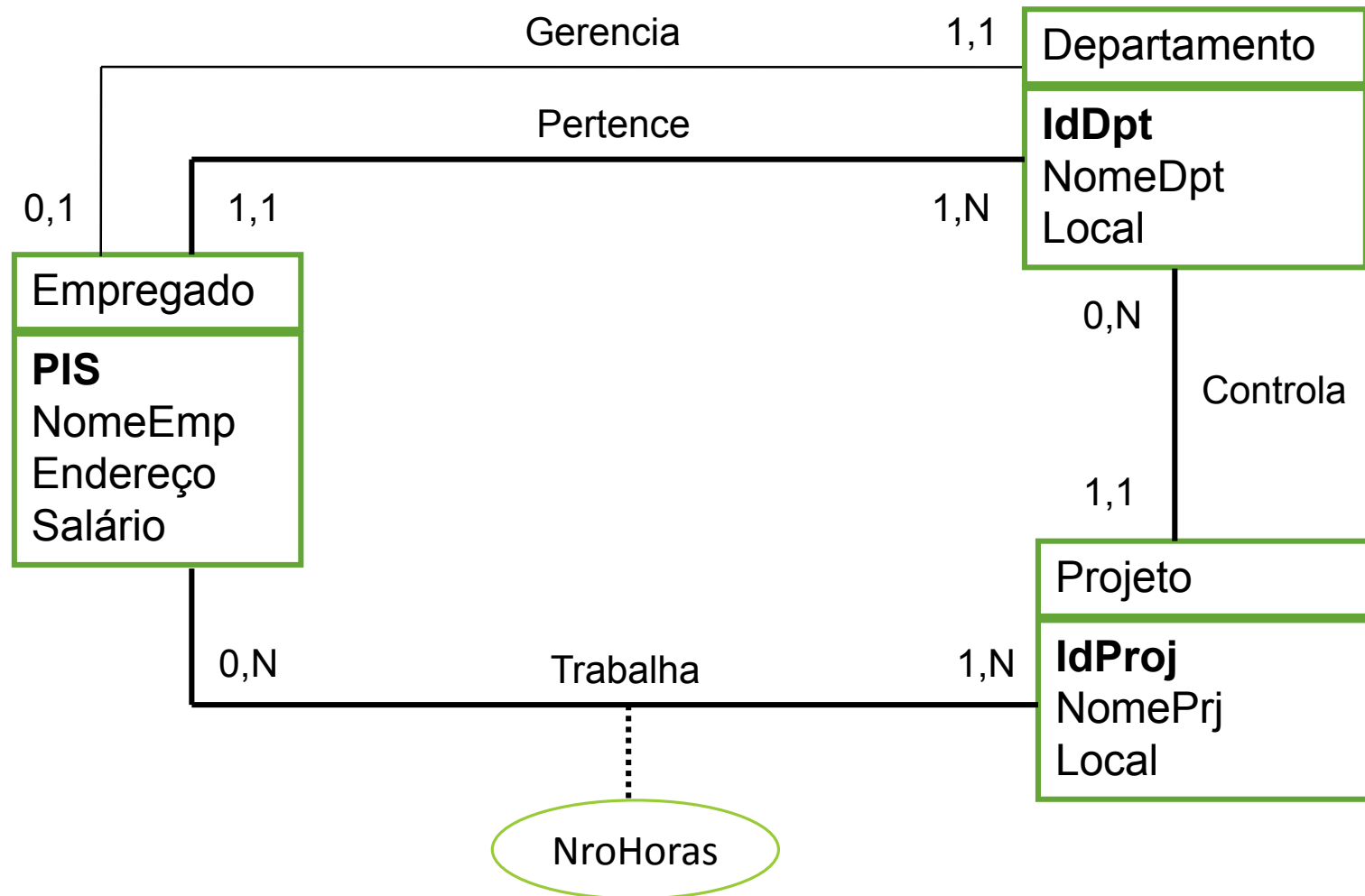
- ✓ Uma ferramenta CASE para banco de dados deve ser capaz de:
  - Fazer mapeamento do modelo ER para modelo relacional
  - Gerar scripts SQL
  - Fazer engenharia reversa

Para download das

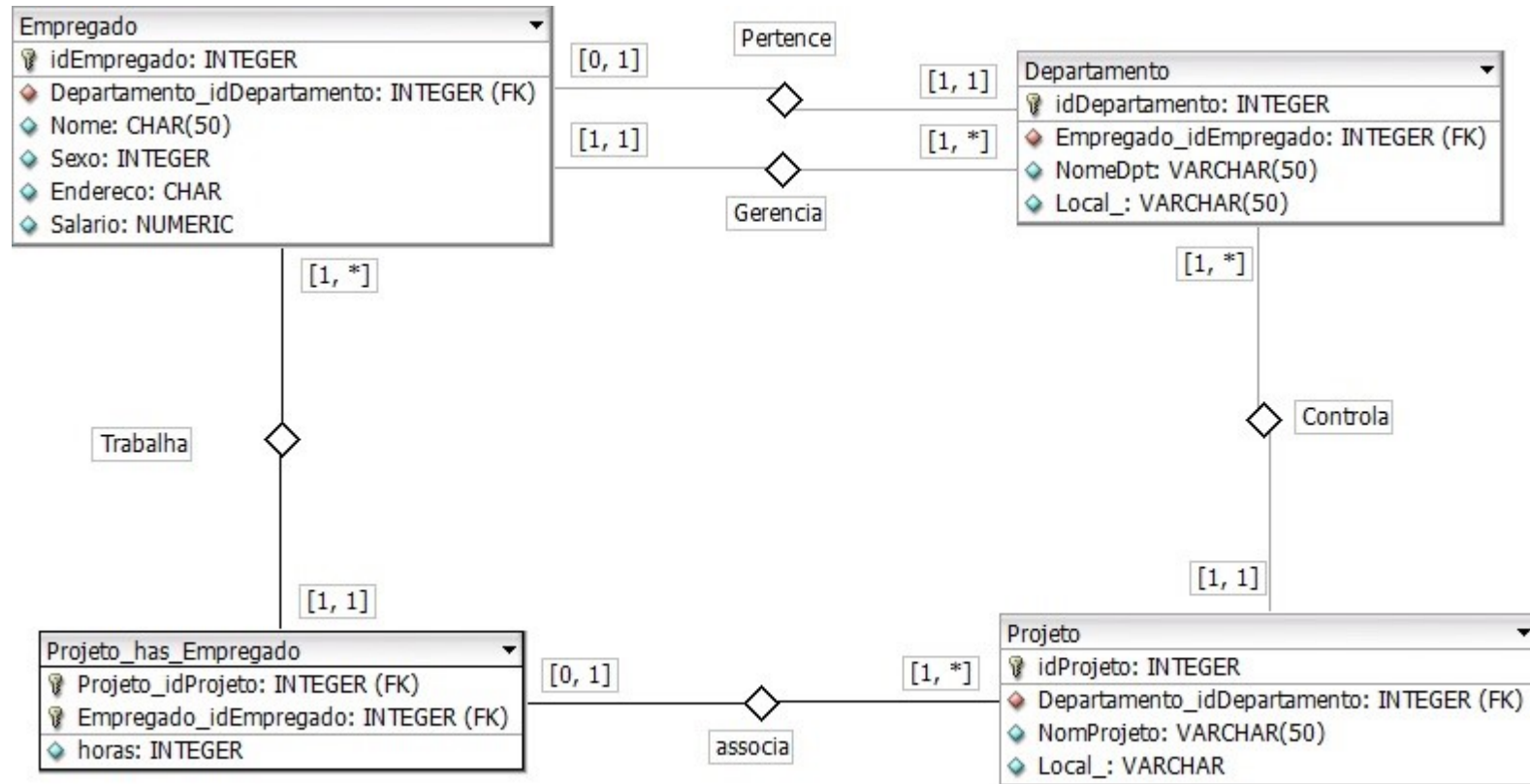
# Exemplo

---

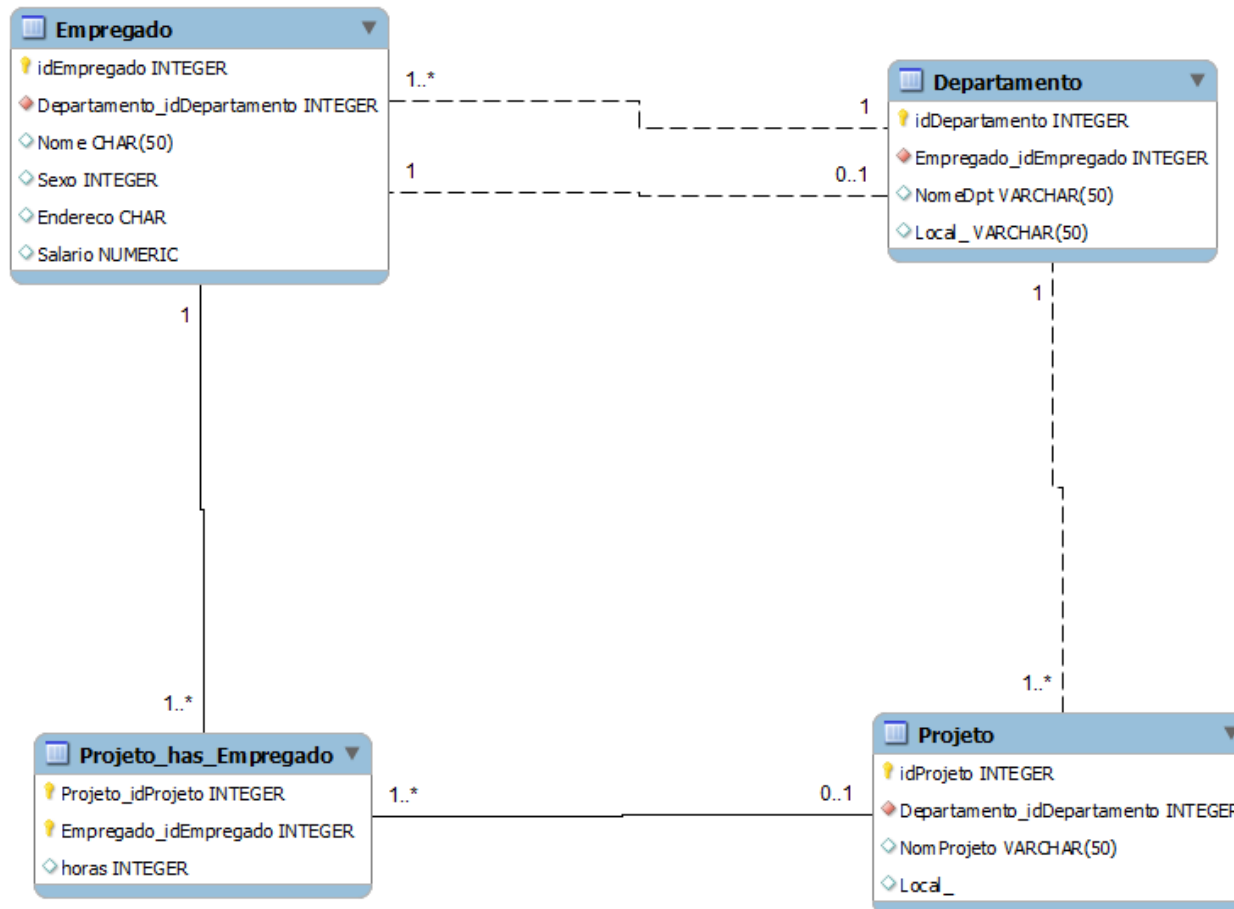
- ✓ Uma companhia deseja um BD para armazenar seus empregados, departamento e projetos. Para tal, considere a seguinte descrição:
  - A companhia é organizada em departamentos. Cada **departamento** tem um único nome, um único nro e um determinado empregado que gerencia o departamento.
  - Um departamento controla um nro de **projetos**, cada um deles tem um único nome, um único nro e uma localização.
  - É armazenado o nome, pis, endereço , salário, sexo e data de nascimento de um empregado. Um **empregado** é associado a um departamento mas pode trabalhar em vários projetos, que não são necessariamente controlados pelo mesmo departamento. Além dessas informações, deve ser armazenado o nro de horas que cada empregado trabalha em um projeto.



# DER – DbDesigner



# DER – Workbench (UML)





# DER: Várias Notações

---

CHEN

MERISE

PÉ-DE-GALINHA

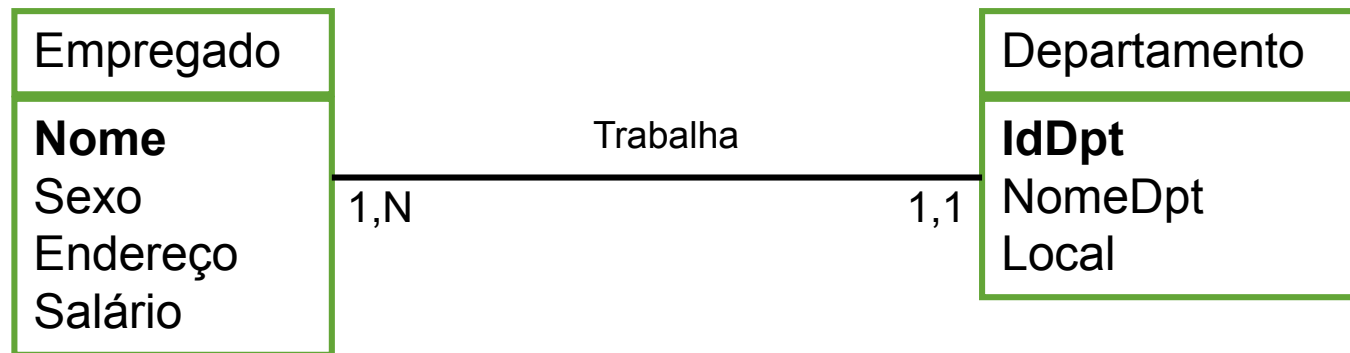
UML

A solid green horizontal bar spanning the entire width of the slide at the bottom.

# Chen

---

- ✓ Peter Chen foi um dos primeiros autores a lançar um livro sobre abordagem E-R, por isso que sua notação é a mais utilizada
- ✓ Ex:



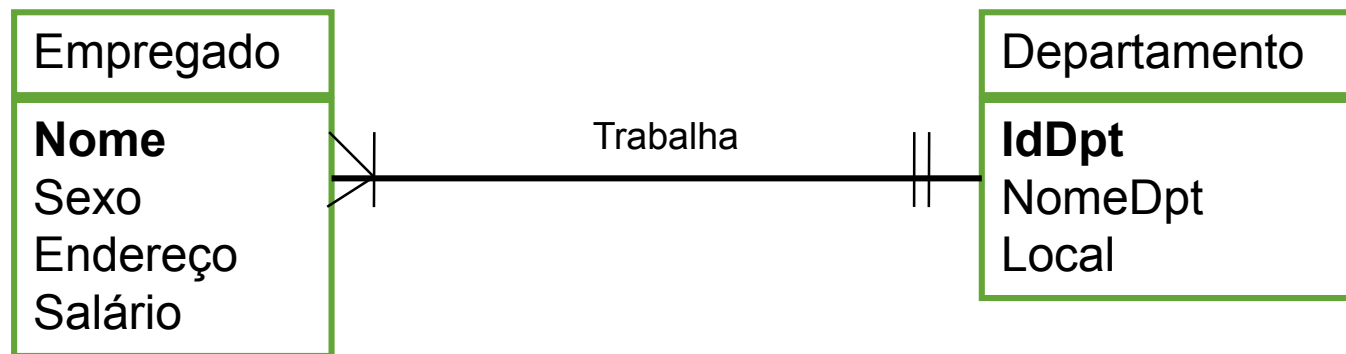
1,N é a cardinalidade do Departamento

1,1 é a cardinalidade do Empregado

# Pé-de-galinha

---

- ✓ Chamada também de notação Engenharia de Informações ou notação James Martin
- ✓ Para a Engenharia de informação (método de desenvolvimento de sistemas de informação), foi definida a seguinte notação gráfica
- ✓ Ex:

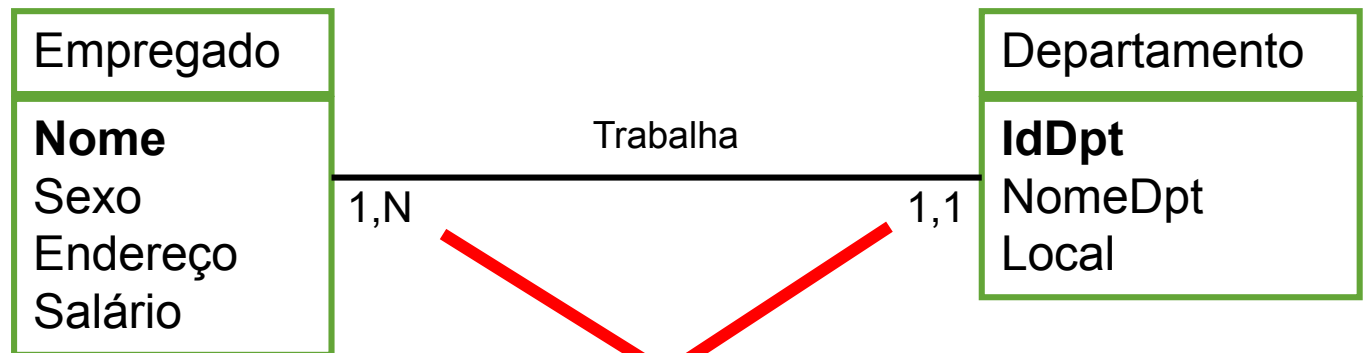
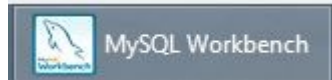


# Merise

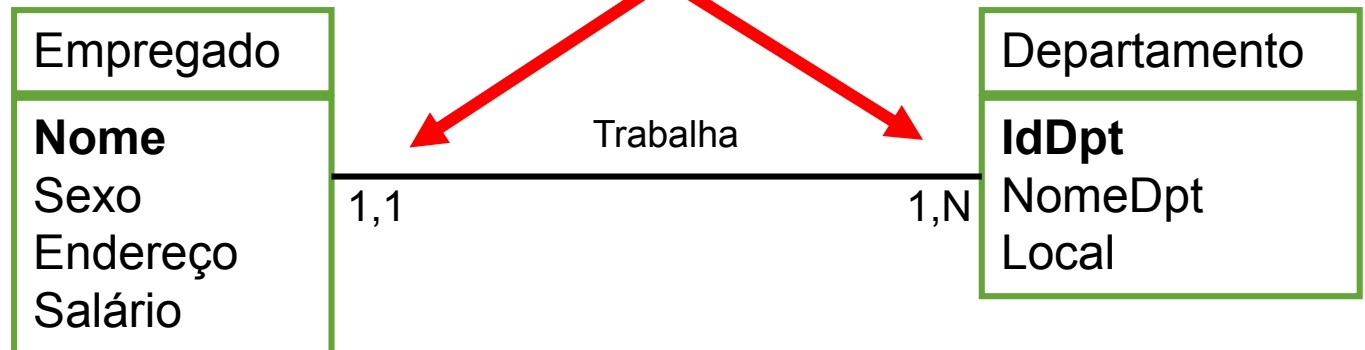
- ✓ Muito utilizado em ferramentas CASE. Coloca-se a cardinalidade do lado da entidade a que ela se refere (contrário do Chen).

✓ Ex:

- Chen

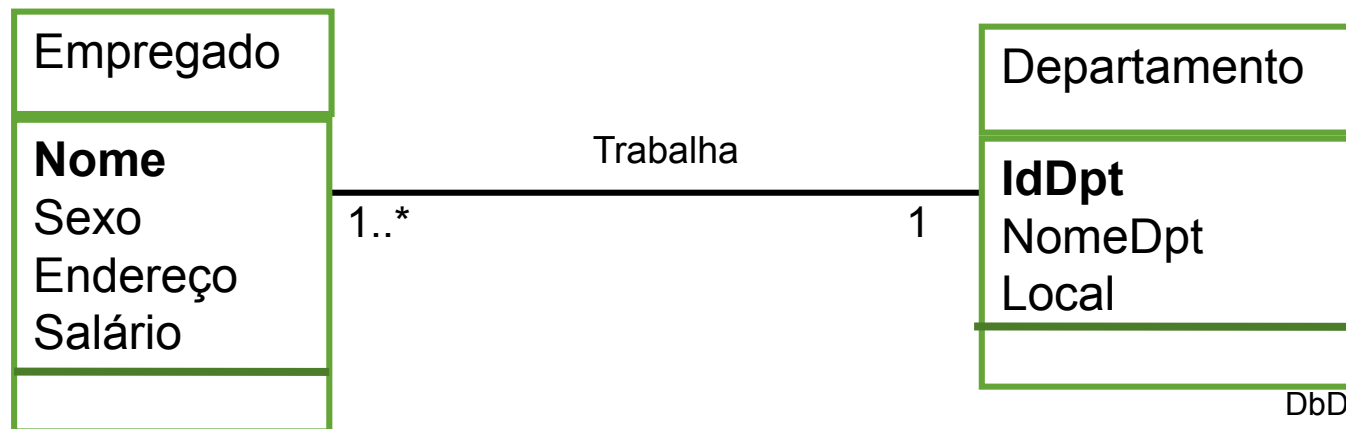


- Merise




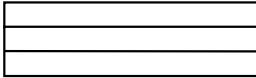
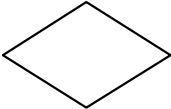

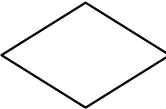

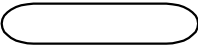
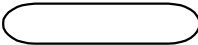
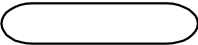
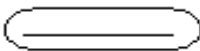
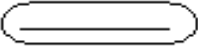
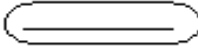
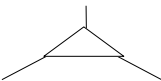

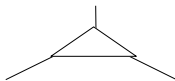
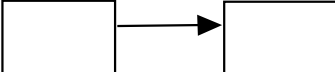
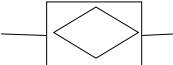
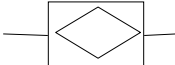
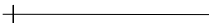






# UML

- ✓ UML (*Unified Modeling Language*) é uma linguagem utilizada em projetos de software orientados a objetos
- ✓ O DER á la UML é baseado no Diagrama de classes da UML
- ✓ A representação de Mínimo e Máximo é chamado de multiplicidade e essas são especificadas na forma *MIN..MAX* onde *\** indica que não há limite. As multiplicidades são colocadas nas extremidades opostas do relacionamento em comparação. Um único *\** representa *0..\**, um único *1* representa *1..1*



# Tabela comparativa

Conceito	Símbolos:	DER		UML
	CHEN	Pé-de- galinha	Merise	
Entidade				
Relacionamento				
Atributos				São representados na classe
Atributo identificador				Nada consta
Generaliza ção				
Entidade Associativa		<b>X</b>		<b>X</b>
Cardinalidade	<b>1:1</b> <b>1:N</b> <b>N:N</b> <b>0:1</b> <b>0:N</b>	    	<b>1:1</b> <b>1:N</b> <b>N:N</b> <b>0:1</b> <b>0:N</b>	<b>1</b> <b>1..*</b> <b>*</b> <b>0..1</b> <b>0..*</b>

Fonte: Sistemas de Banco de Dados, Fundamentos e Aplicações, Elmasri, Ed. LTC

# Modelo Relacional

---

# Modelo Relacional

---

- ✓ Modelo de dados onde o BD consiste em uma coleção de tabelas
- ✓ Cada tabela é chamada relação porque corresponde a este conceito matemático
- ✓ Cada linha de uma tabela recebe o nome de tupla
- ✓ Cada coluna de uma tabela recebe o nome de atributo



# Modelo Relacional

---

Relação

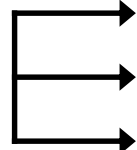


Empregado

Atributos



Tuplas



E_Nome	E_PIS	E_Endereço	E_DataNasc
Pedro	9670000	r. XV de No...	5/5/1965
João	9711111	r. 13 de Maio...	6/7/1977
Maria	9598765	r. 7 de Setem...	3/5/1975

# Restrições do Modelo Relacional

---

## ✓ Domínio

- Todo atributo deve ter um valor atômico (indivisível). Não é possível a existência de valores compostos ou multi-valorados.

## ✓ Chave

- Toda tupla tem que ser distinta. Duas tuplas não podem ter a mesma combinação de valores para todos os seus atributos. Um atributo chave distingue apenas uma tupla em uma relação

# Chaves

---

Chave Candidata

Chave Primária

Chave Estrangeira

Nome	PIS	Rg	Depto
Jose Silva	123	987	D1
Jose Silva	235	789	D2
Joao Sa	252	126	D1

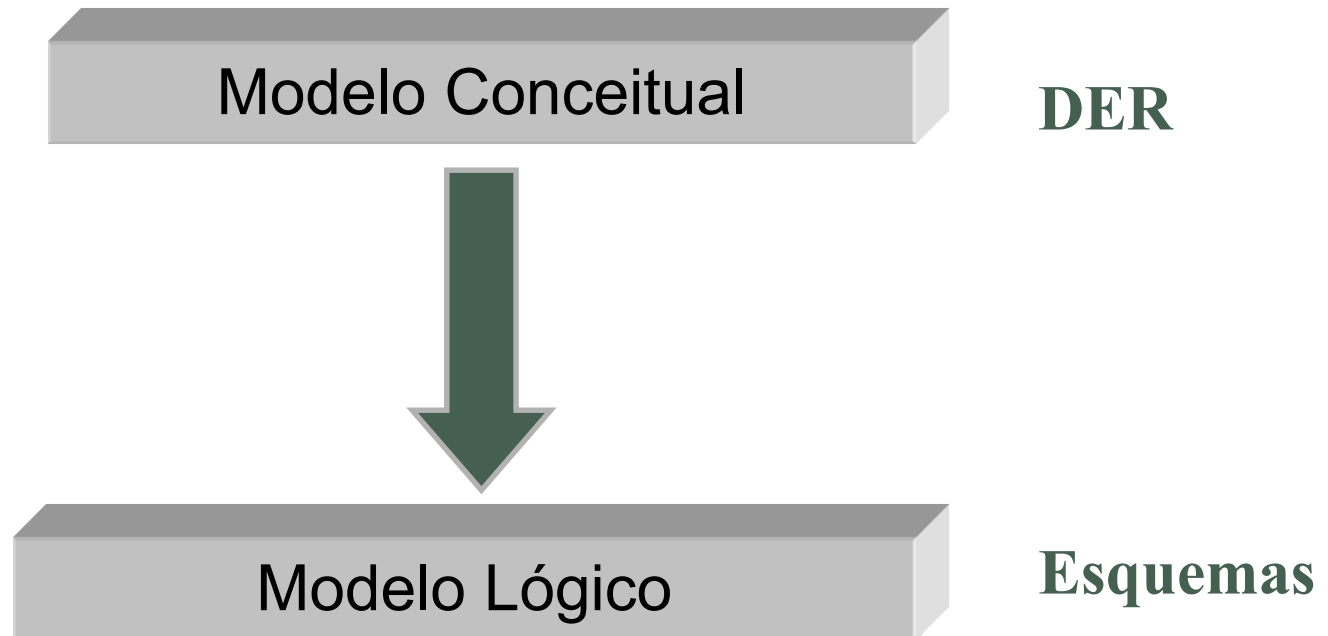
Chave Primária

Super Chave

Código	Nome	Local
D1	RH	Campinas
D2	Vendas	São Paulo

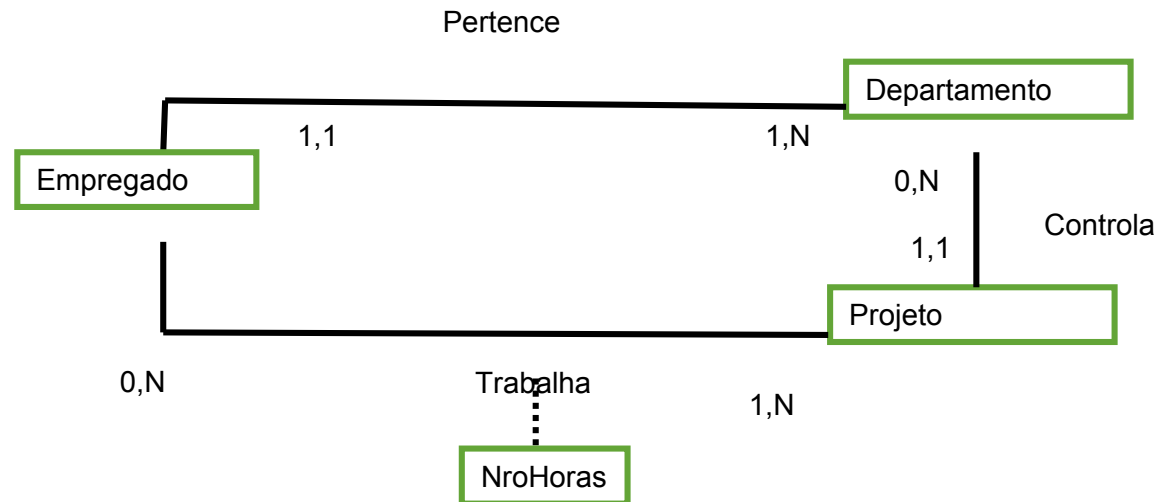
# Transformação Entre Modelos

---



# Mapeamento ER->Relacional (Regras)

## 1. Para cada entidade criar uma tabela



Empregado(PIS, NomeEmp, Salário, Endereço)

Departamento(IdDpt, NomeDpt, Local)

Projeto(IdPrj, NomePrj, Local)



# Mapeamento de relacionamentos

## *Implementação de relacionamentos 1:1*

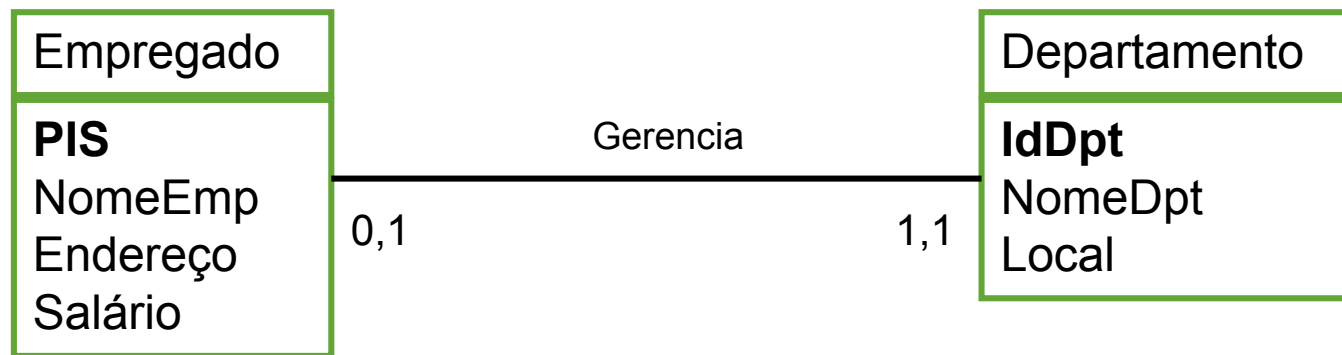
Tipo de relacionamento		Regra de implementação		
		Tabela própria	Adição coluna	Fusão tabelas
(0,1)		±	✓	×
(0,1)		×	±	✓
(1,1)		×	×	✓

✓ Alternativa preferida

± Pode ser usada

× Não usar

# Mapeamento Relacionamentos 1:1



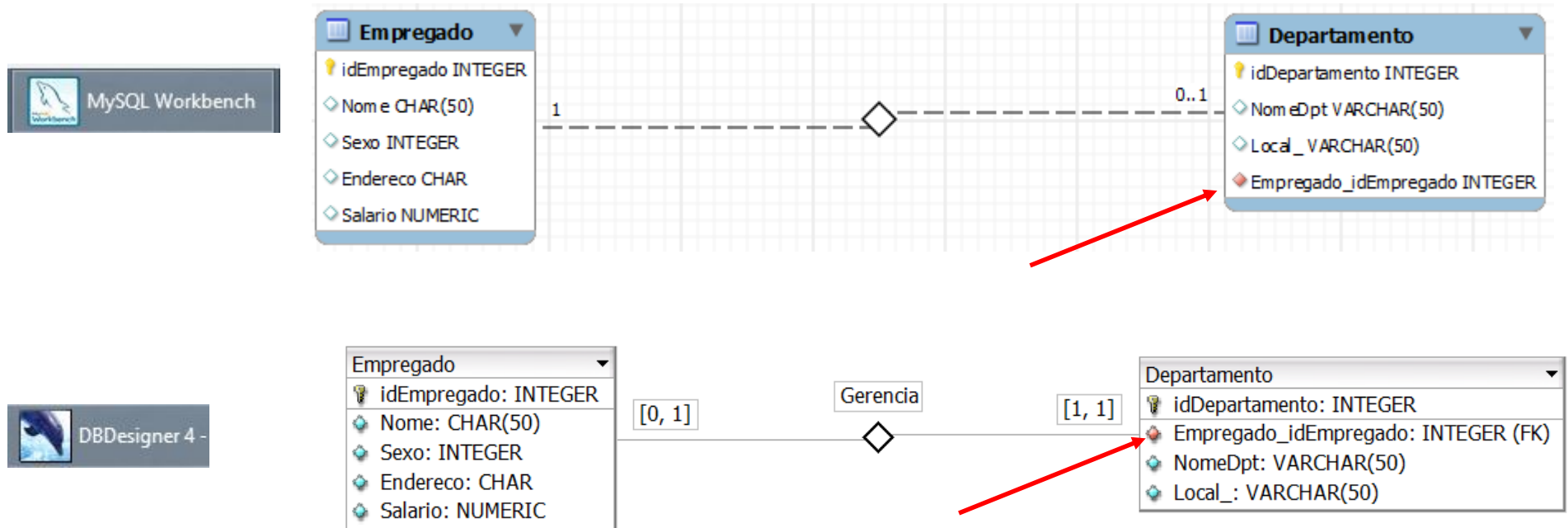
Empregado(PIS, NomeEmp, Endereço, Salário)

chave estrangeira

Departamento(IdDpt, Nome, Local, PIS\_Gerente)







# Mapeamento Relacionamentos 1:1





# Mapeamento Relacionamentos 1:N

## Relacionamentos 1:n

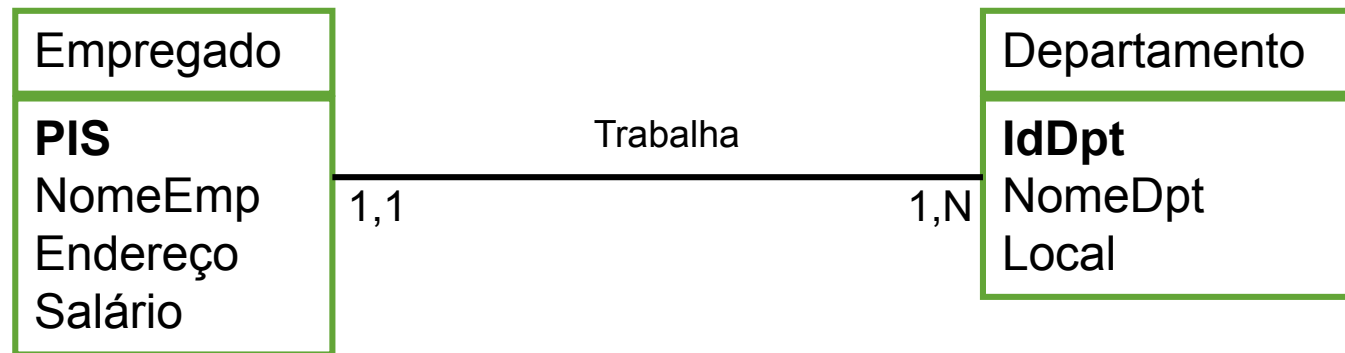
Tipo de relacionamento		Regra de implementação		
		Tabela própria	Adição coluna	Fusão tabelas
(0,1) 	(0,n)	±	✓	×
(0,1) 	(1,n)	±	✓	×
(1,1) 	(0,n)	×	✓	×
(1,1) 	(1,n)	×	✓	×

✓ Alternativa preferida

± Pode ser usada

× Não usar

# Mapeamento Relacionamentos 1:N



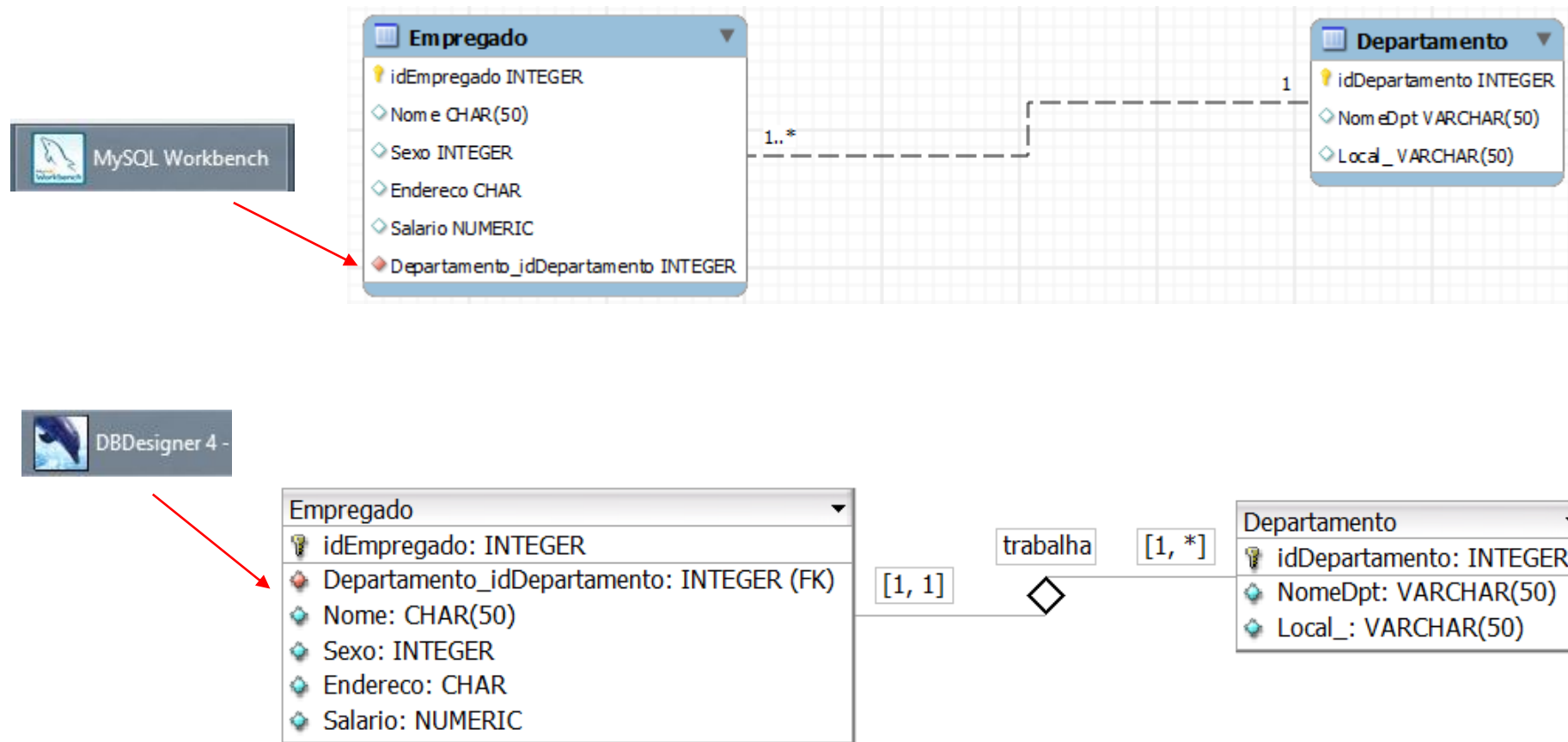
Empregado(PIS, NomeEmp, Endereço, Salário, IdDepto)

chave estrangeira

Departamento(IdDpt, NomeDpt, Local)


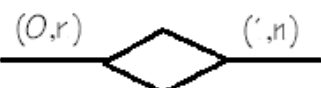



# Mapeamento Relacionamentos 1:N



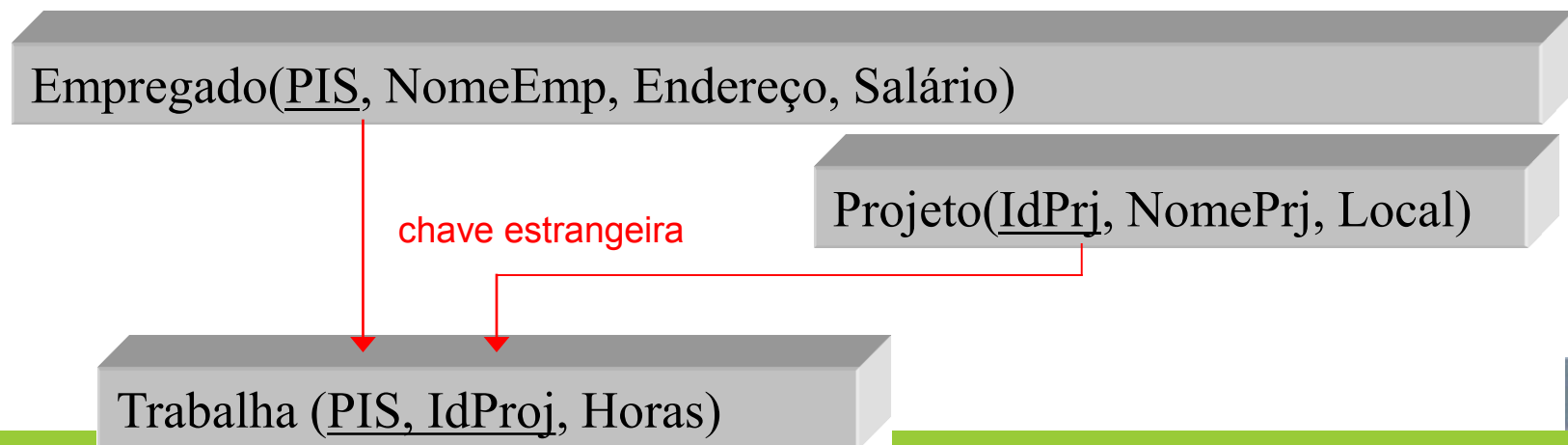
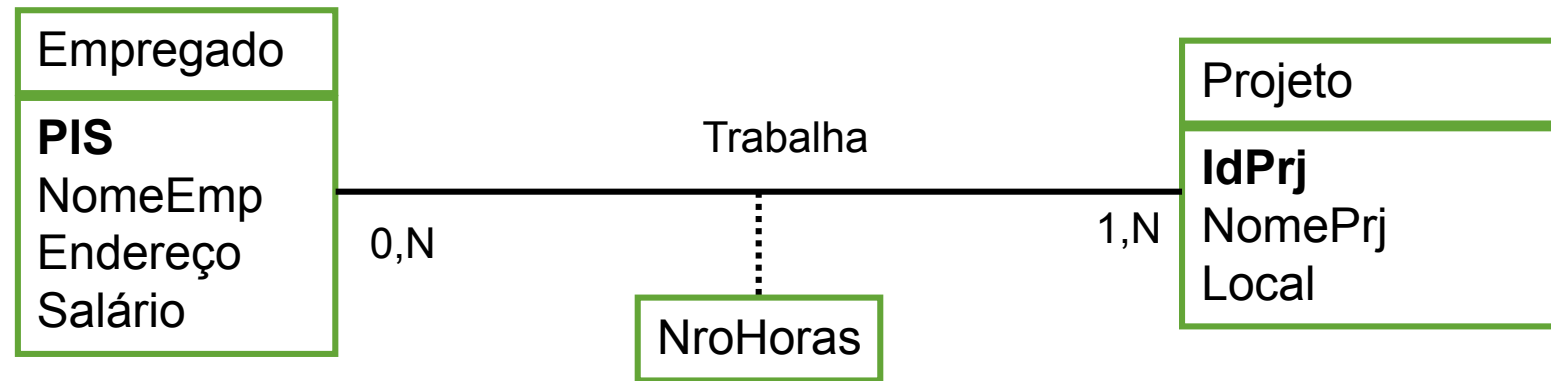
# Mapeamento Relacionamentos N:N

## *Relacionamentos n:n*

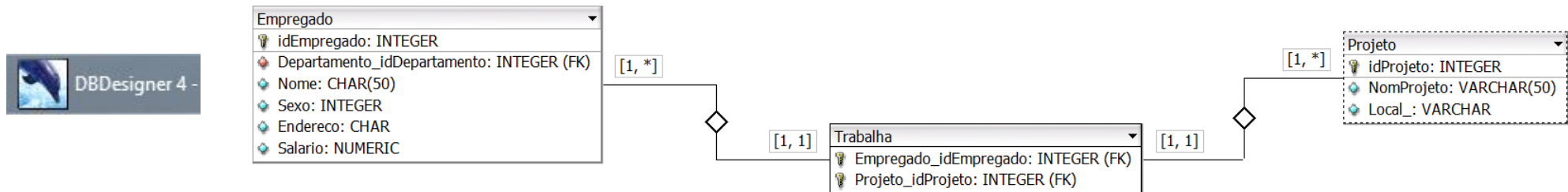
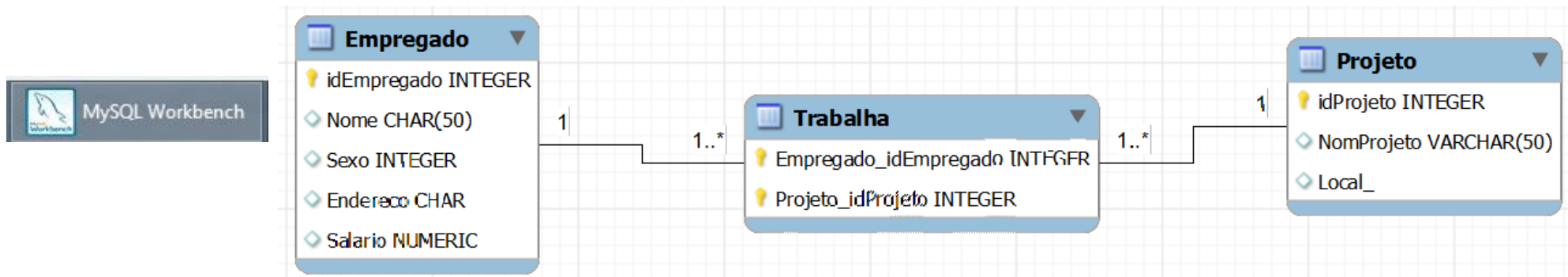
Tipo de relacionamento		Regra de implementação		
		Tabela própria	Adição coluna	Fusão tabelas
	$(O,r)$ $(O,n)$	✓	×	×
	$(O,r)$ $(',n)$	✓	×	×
	$(1,n)$ $(',n)$	✓	×	×

✓ Alternativa preferida      × Não usar

# Mapeamento Relacionamentos N:N



# Mapeamento Relacionamentos N:N

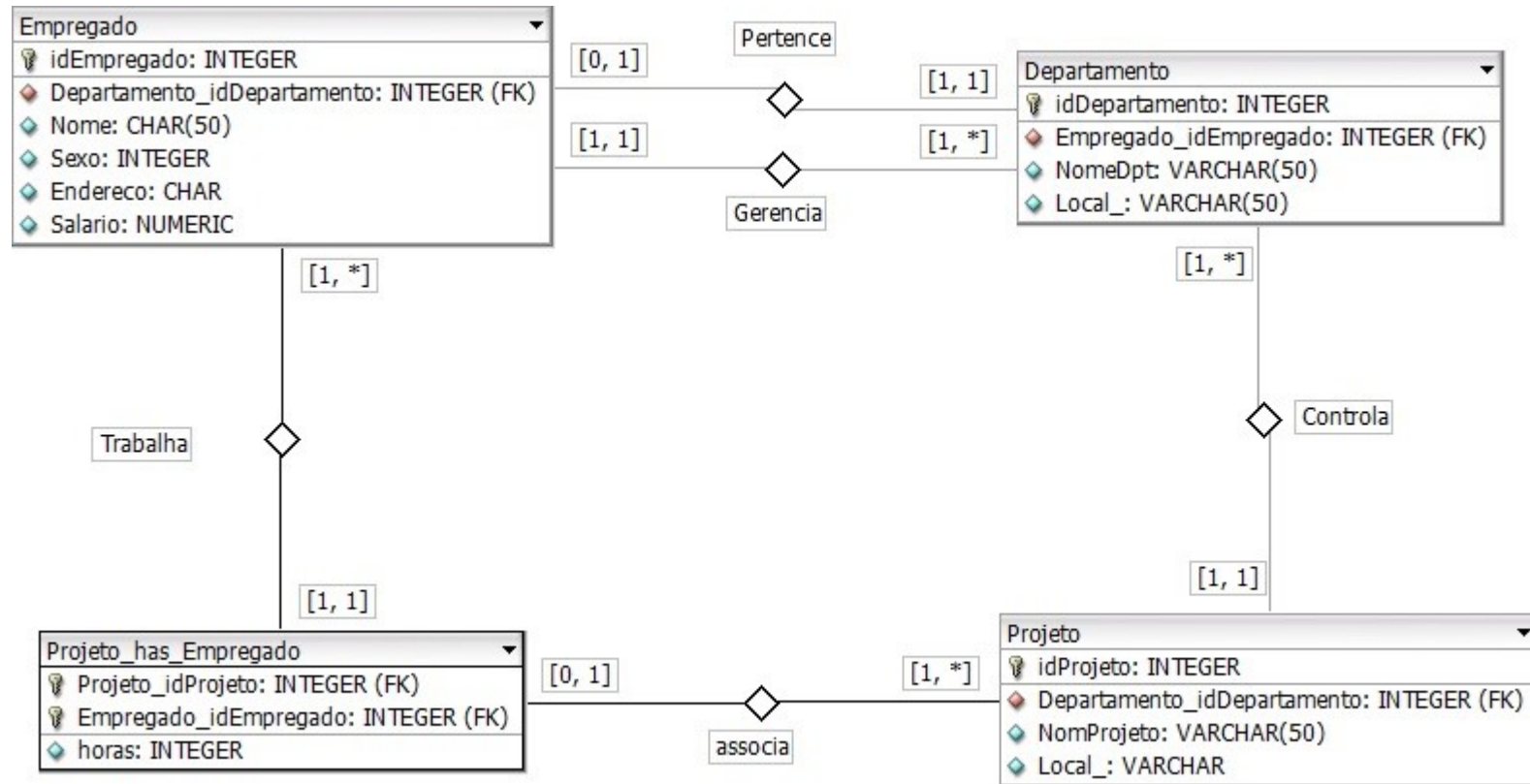


# Resumo das Regras de Mapeamento ER-> Relacional

---

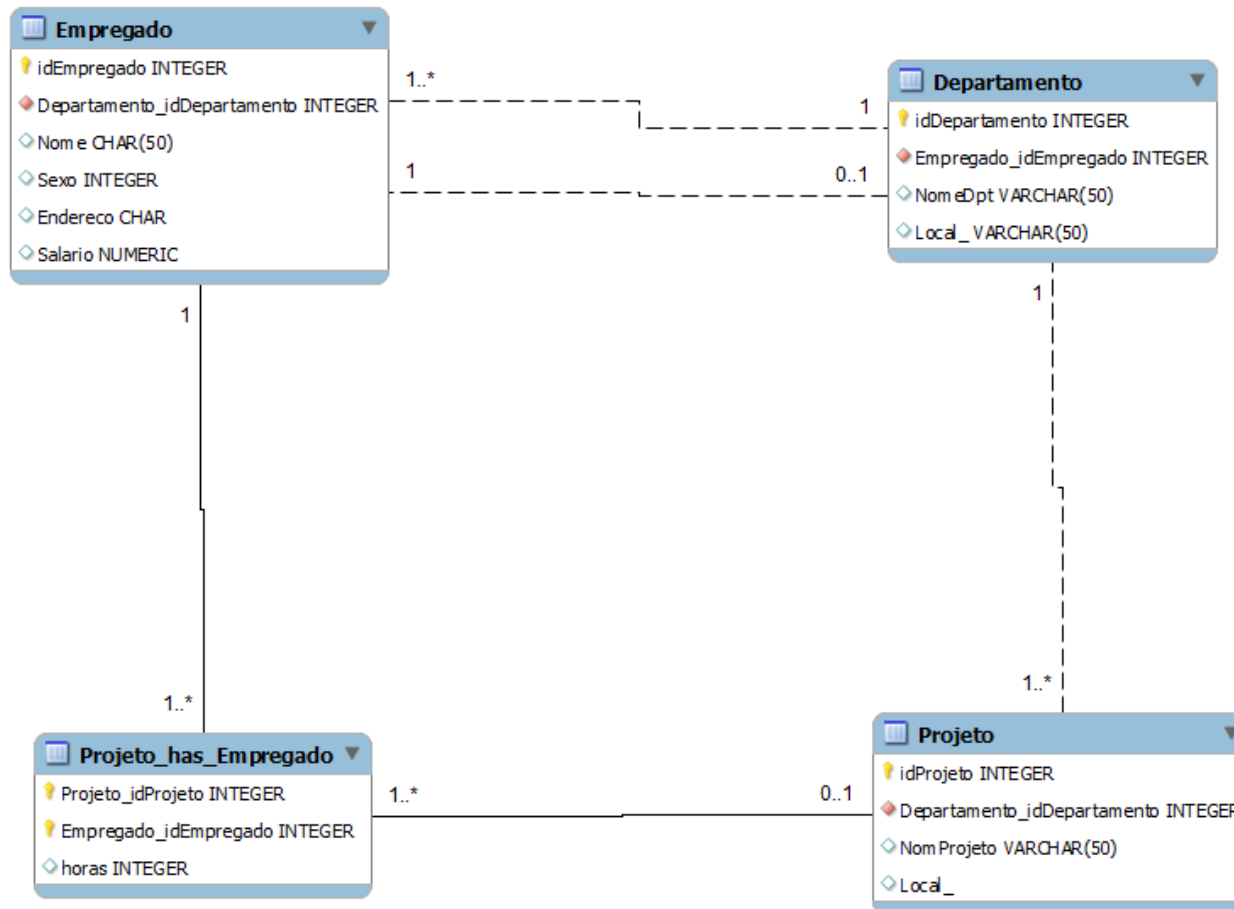
1. Para cada entidade criar uma tabela
2. Para cada relacionamento 1:1 juntar tabela ou adicionar FK
3. Para cada relacionamento 1:N adicionar FK
4. Para cada relacionamentos N:N criar uma tabela com chaves primárias compostas

# DER – DbDesigner





# DER – Workbench (UML)



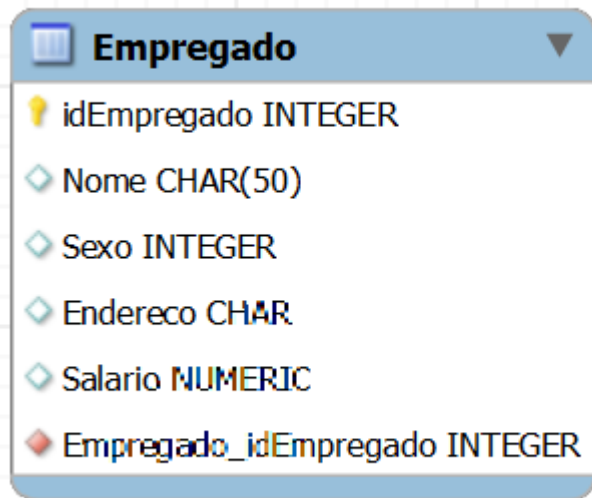
# Outros tipos de relacionamentos

---

# Auto-relacionamento

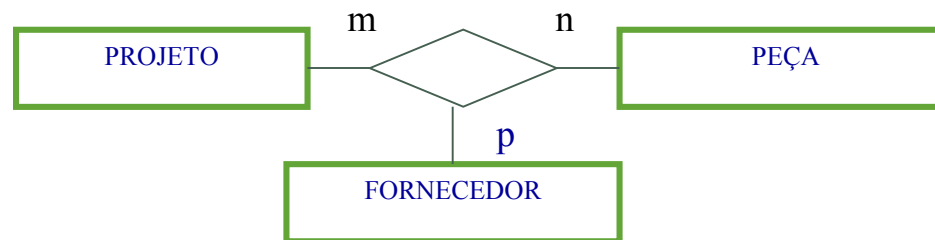
---

- ✓ Relacionamento de uma entidade com ela mesma
- ✓ Exemplo Empregado supervisiona empregado



# Relacionamentos Ternários

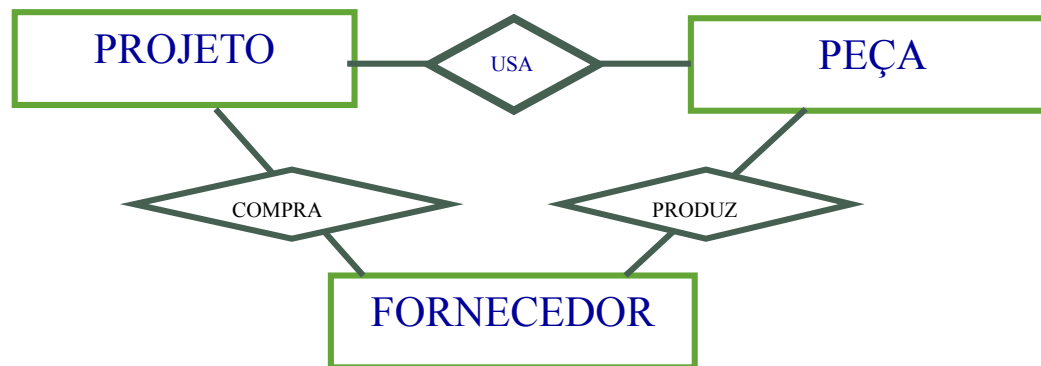
- ✓ Relacionamento que associa três entidades.
  - Ex. Vários fornecedores fornecem determinadas peças a alguns projetos:



NroForneced	NroPeça	NroProj
1	100	1
1	200	1
1	200	2
2	100	1

# Relacionamentos Ternários

- ✓ Um relacionamento ternário **NÃO** corresponde a três relacionamentos binários



compra

NroProj	NroFornec
1	1
1	2
2	1
2	2

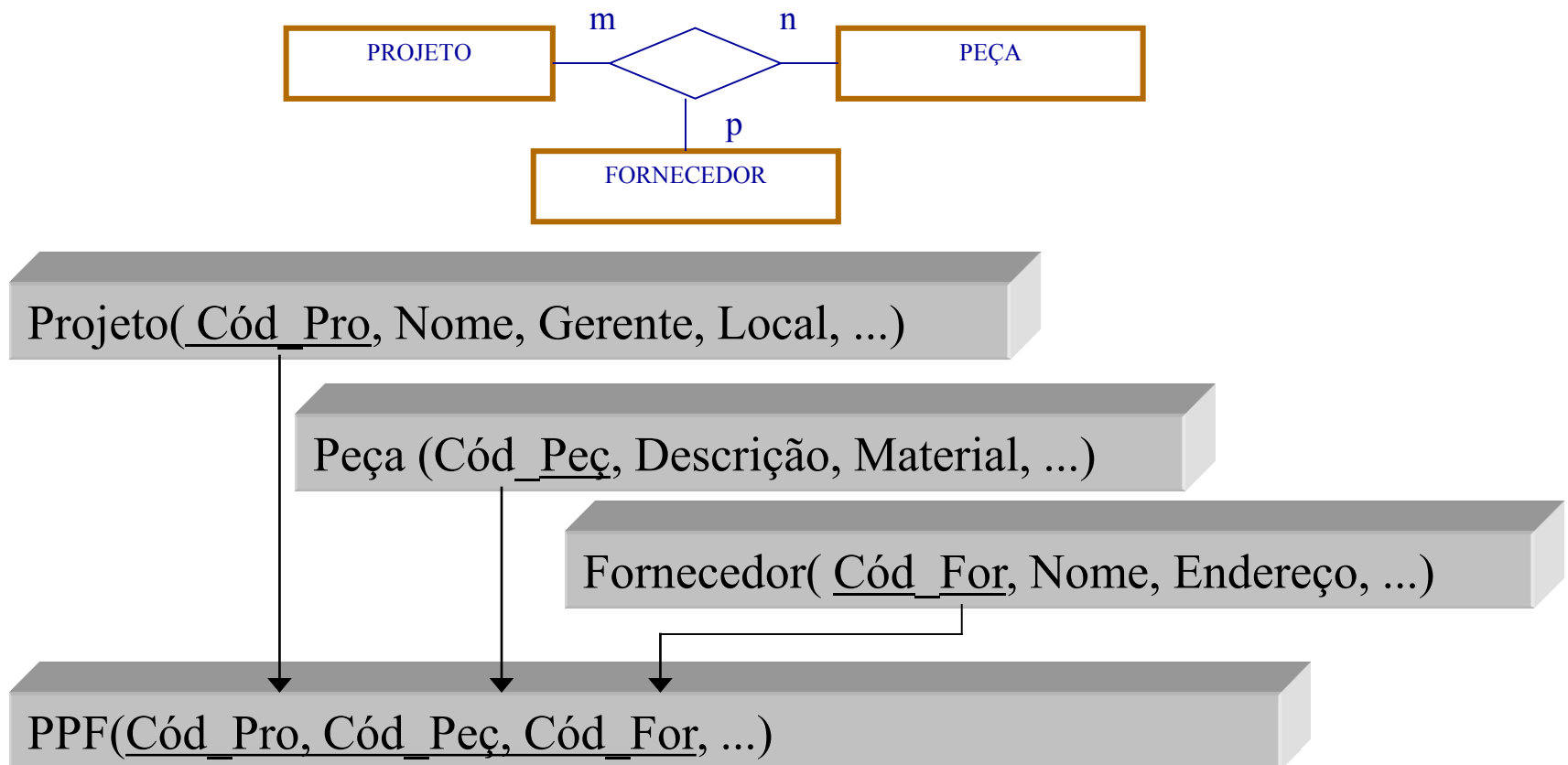
usa

NroProj	NroPeça
1	100
2	100
1	200

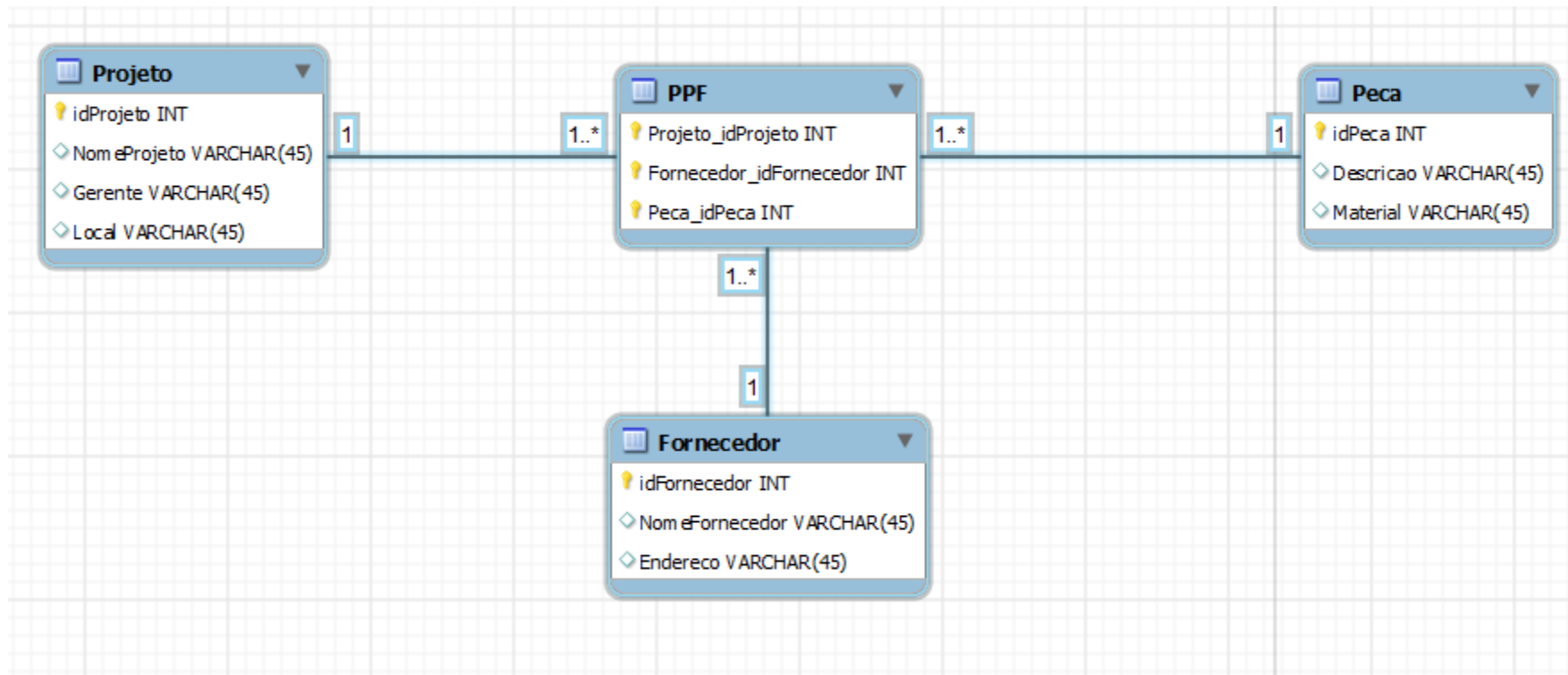
produz

NroForneced	NroPeça
1	100
2	100
1	200

# Mapeamento Relacionamentos Ternários



# Relacionamento ternário



# Modelo Entidade-Relacionamento Estendido

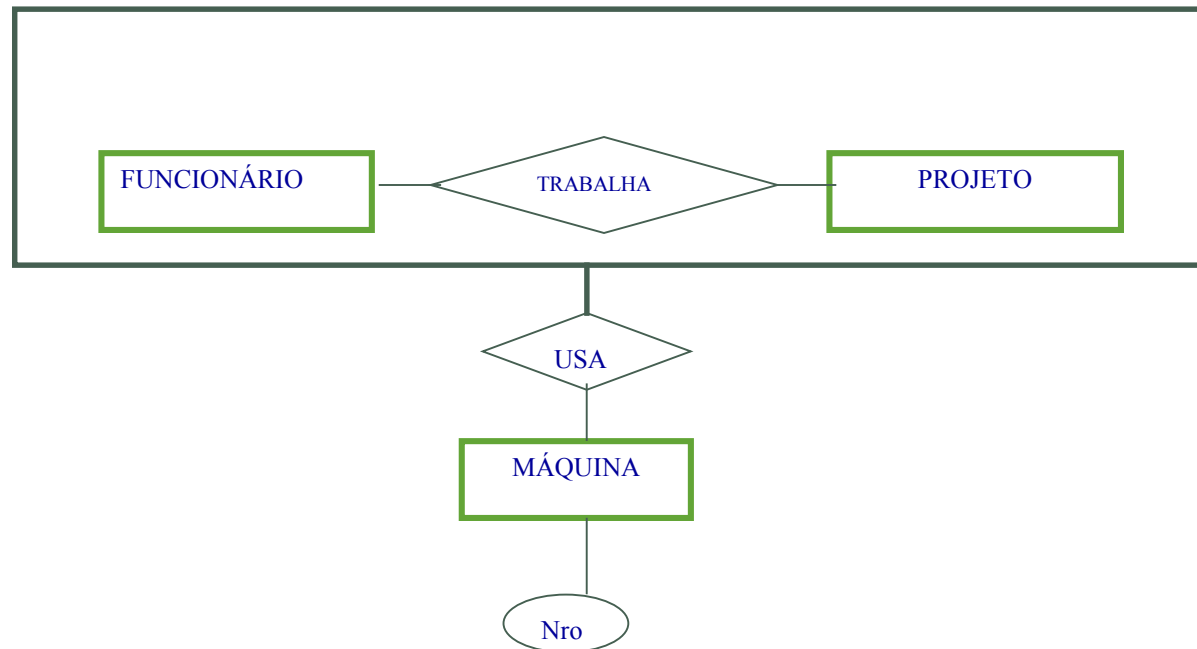
---



# Agregação

---

- ✓ No modelo E-R não é possível expressar relacionamentos entre relacionamentos.

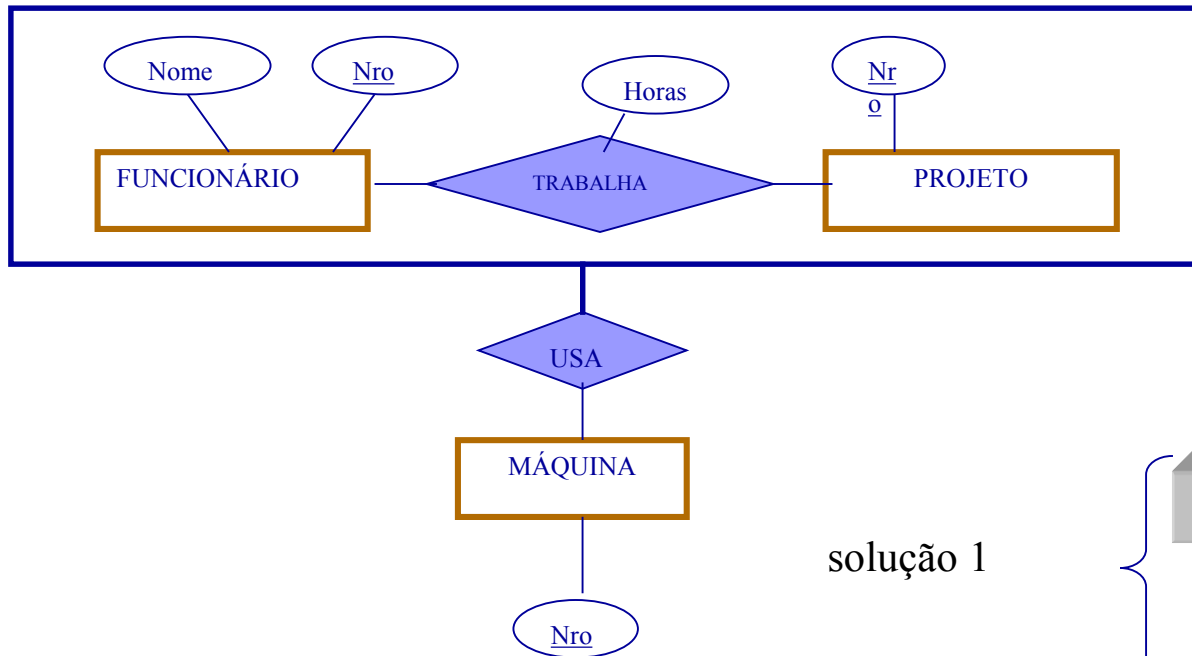


# Mapeamento de Agregações

---

- ✓ Para mapear Agregações duas soluções são possíveis:
  - Cria-se uma relação para cada relacionamento.
  - Cria-se uma única relação para os dois.

# Mapeamento de Agregações



solução 1

Trabalha(Nrofunc, nroProj, horas)

Usa(Nrofunc, NroProj, NroMaq)

solução 2

Trab\_Usa(Nrofunc, NroProj, NroMaq, Horas)

Funcionário(Nome, Nro,...)

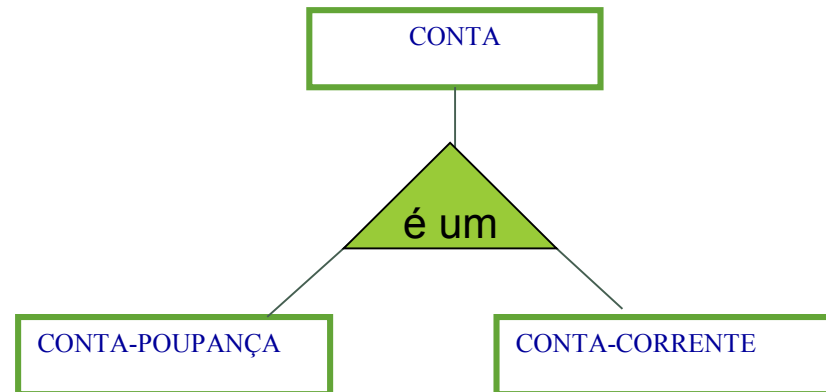
Projeto(Nro, Nome, local,...)

Máquina(Nro, ...)

# Generalização

---

- ✓ Expressa a semelhança entre entidades através de um relacionamento de conteúdo entre um conjunto entidade de nível superior e um ou mais conjuntos entidade de nível inferior.



# Mapeamento

## Generalização/Especialização

---

- ✓ Para casos de Generalização/Especialização duas soluções podem ser adotadas:
  - Criar uma relação para cada entidade. As relações correspondentes as entidades não principais contêm a chave da relação principal
  - Criar relações apenas para as entidades não principais.

# Mapeamento Generalização/Especialização

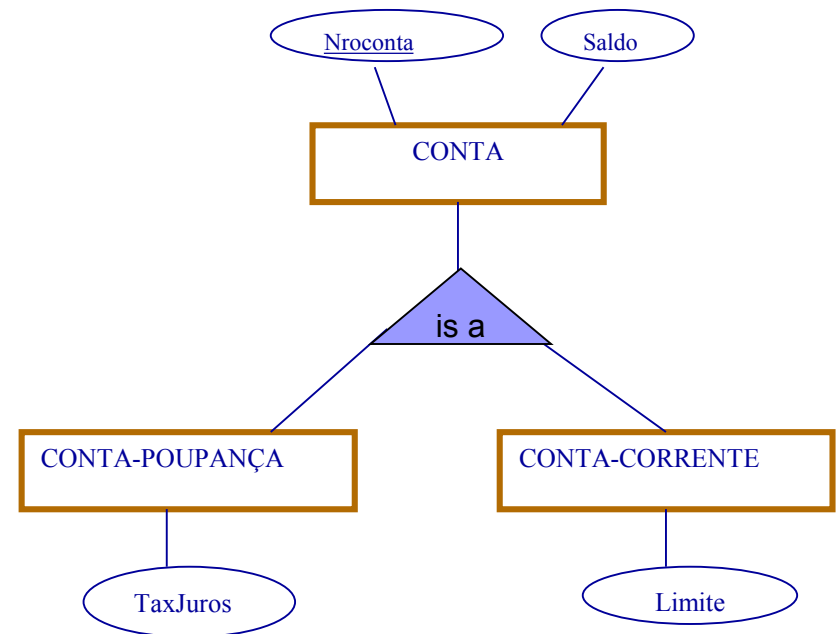
## ✓ Solução 1:

- Conta(Nroconta, Saldo)
- ContaPoup(NroConta, Txjuros)
- ContaCorr(NroConta, Limite)

## Solução 2:

ContaPoup(NroConta, Saldo, Txjuros)

ContaCorr(NroConta, Saldo, Limite)



# Generalização em DbDesigner

---

# Bibliografia

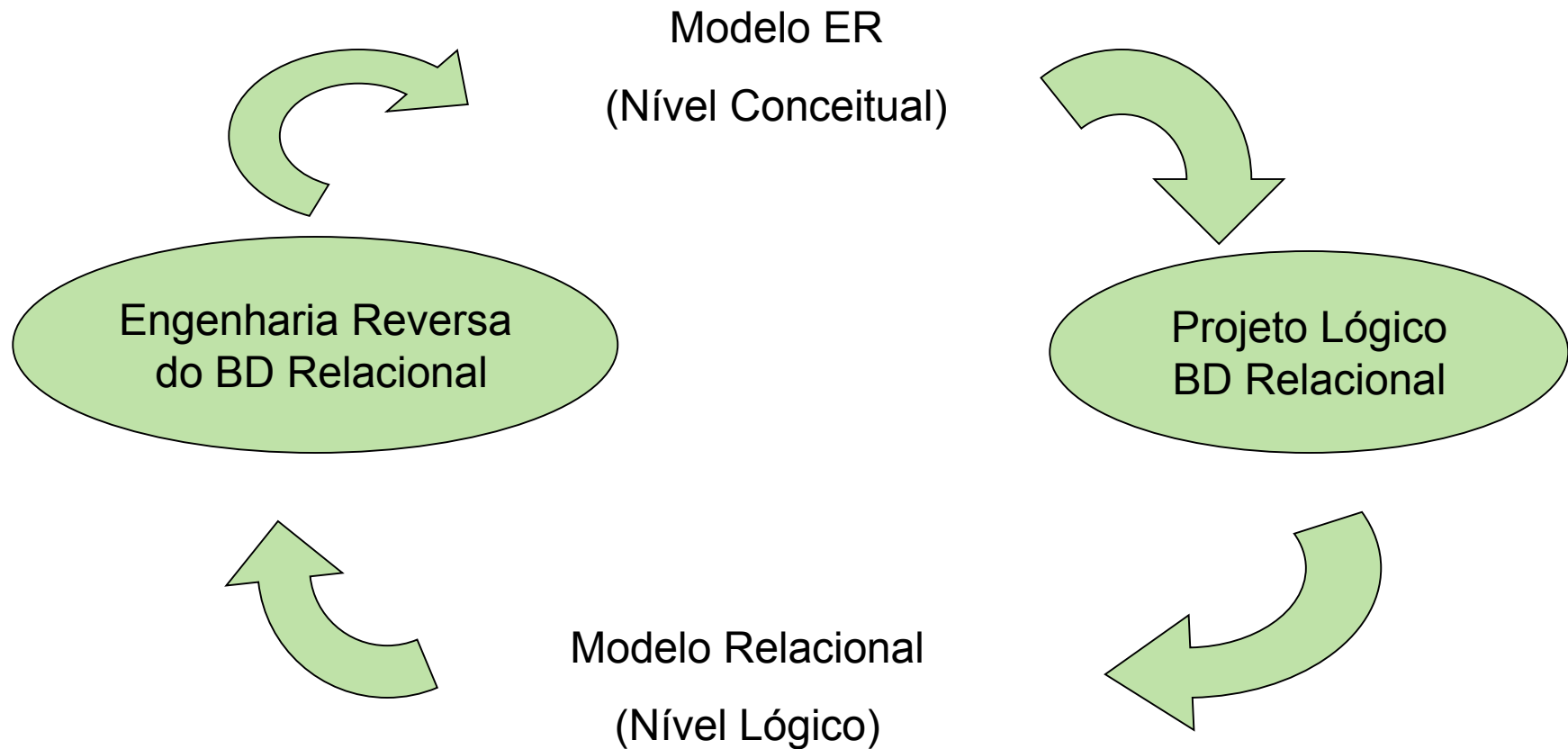
---

- ✓ Heuser C. A, Projeto de Banco de Dados, 3a ed, Editora Sagra Luzzatto, 2000.
- ✓ Elmasri & Navathe, Sistemas de Banco de Dados Fundamentos e Aplicações, 3a. ed., LTC, 2002.
- ✓ Korth, h.; Silberschatz, A. Sistemas de Banco de Dados. Makron, 3a ed. 1999.
- ✓ Date, C. J., Introdução a Sistemas de Banco de Dados. Campus, 7a ed. 2000.
- ✓ Oliveira, C.H. P. SQL Curso Prático, Novatec, 2002.
- ✓ Patrick, J.J. SQL Fundamentos, Berkeley Brasil, 2002.
- ✓ Taylor, A. G. SQL para Dummies, Campus, 2001.



# Engenharia Reversa do BD Relacional

---



# Engenharia Reversa de Modelos Relacionais

---

- ✓ Um Processo de Engenharia reversa pode ser definido como um processo de abstração, que parte de um modelo de implementação e resulta em um modelo conceitual que descreve abstratamente a implementação em questão.
- ✓ Ponto de Partida é o Modelo lógico de um BD Relacional
- ✓ O Resultado é o Modelo conceitual (DER)

# Passos para Engenharia Reversa

---

- ✓ Identificação da construção ER correspondente a cada tabela;
- ✓ Definição de relacionamentos 1:n e 1:1;
- ✓ Definição de atributos;
- ✓ Definição de identificadores de entidades e relacionamentos;

# Esquema do BD de um Sistema Acadêmico

---

- ✓ Disciplina(idDisc, NomeDisc)
- ✓ Curso(idCur, NomeCur)
- ✓ Curriculo(idCur, idDisc, nrocreditos)  
idCur referencia Curso  
idDisc referencia Disciplina
- ✓ Sala(idPredio, idSala, Capacidade)  
idPredio referencia Prédio
- ✓ Prédio(idPredio, Endereco)
- ✓ Turma(AnoSem, idDisc, SiglaTur, NroAlunos, idPredio, idSala)  
idDisc referencia Disciplina  
idPredio, idSala referencia Sala
- ✓ Laboratório (idPredio, idSala, Equipamento)  
idPredio, idSala referencia Sala

# Possibilidades de uma tabela

---

- ✓ Uma tabela pode ser:
  - Uma entidade
  - Um relacionamento n:n
  - Uma entidade especializada

# Identificação da construção ER das tabelas

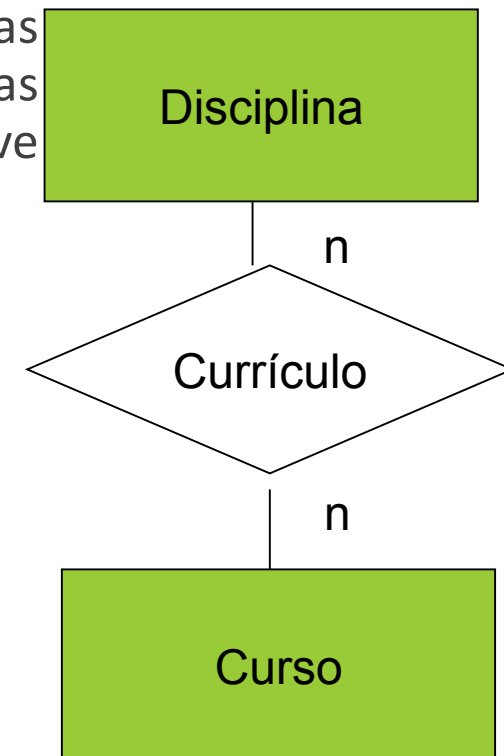
## ✓ Regra 01

A tabela que possui uma chave primária composta por múltiplas chaves estrangeiras implementa um relacionamento n:n entre as entidades correspondentes às tabelas referenciadas pela chave estrangeira

- Disciplina(idDisciplina, NomeDisc)
- Curso (idCurso, NomeCurso)
- Currículo(idDisciplina, idCurso, nrocreditos, ... )

idDisciplina refere-se a Disciplina

idCurso refere-se a curso

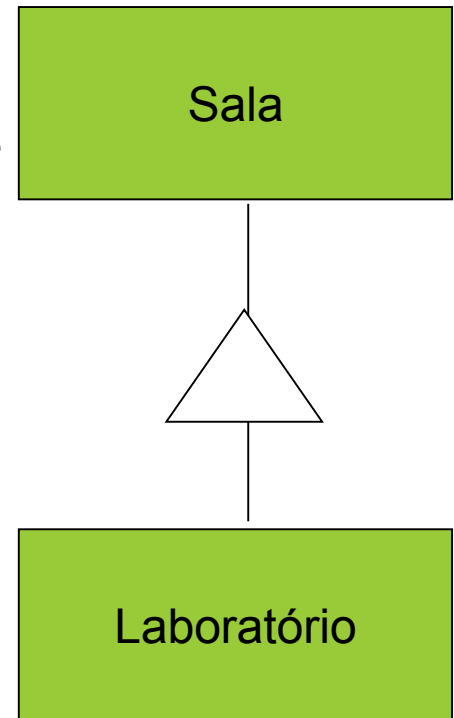


# Identificação da construção ER das tabelas

## ✓ Regra 02

A tabela cuja chave primária é toda ela **uma** chave estrangeira representa uma entidade que forma uma especialização da entidade correspondente à tabela referenciada pela chave estrangeira

- Sala(idPredio, idSala, Capacidade)  
idPredio referencia Prédio
- Laboratório(idPredio, idSala, área, ... )  
idPredio, idSala refere-se a Sala



# Identificação da construção ER correspondentes as tabelas

---

## ✓ Regra 03

- Quando a chave primária da tabela não for composta por múltiplas chaves estrangeiras (regra 1), nem for toda **uma** chave estrangeira (regra 2), a tabela representa uma entidade.

- Prédio(idPredio, Endereco)
- Disciplina(idDisciplina, NomeDisc)
- Curso (idCurso, NomeCurso)

Prédio

Disciplina

Curso

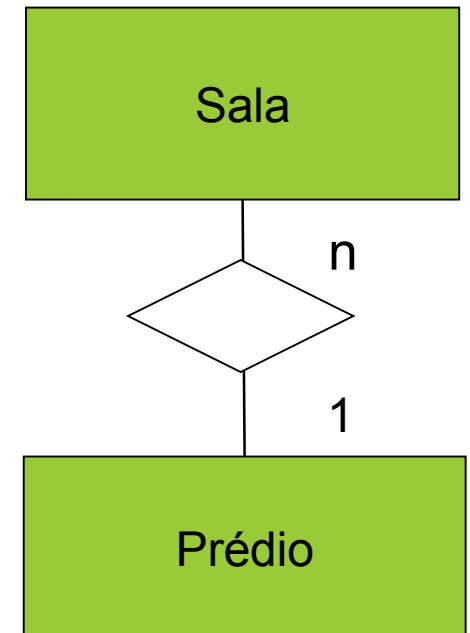


# Identificação de relacionamentos 1:n e 1:1

Toda chave estrangeira que não se enquadra nas regras 1 e 2, ou seja, que não faz parte de uma chave primária composta por múltiplas chaves estrangeiras, nem é toda ela uma chave primária, representa um relacionamento 1:n ou 1:1.

- Prédio (idPredio, endereço)
- Sala (idPredio, idSala, Capacidade)

idPredio referencia Prédio



Neste caso, como o idSala se repete, por isto optou-se pela representação de um relacionamento 1:N identificado, que é aquele tipo de relacionamento em que a chave estrangeira criada na tabela apropriada será definida, também, como chave primaria.

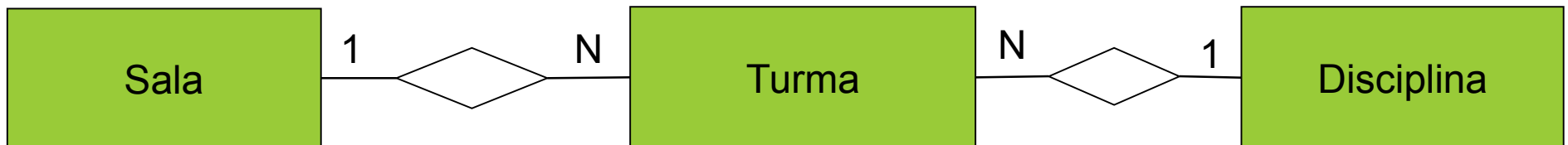
# Identificação de relacionamentos 1:n

✓ Toda chave estrangeira que não se enquadra nas regras 1 e 2, ou seja, que não faz parte de uma chave primária composta por múltiplas chaves estrangeiras, nem é toda ela uma chave primária, representa um relacionamento 1:n ou 1:1.

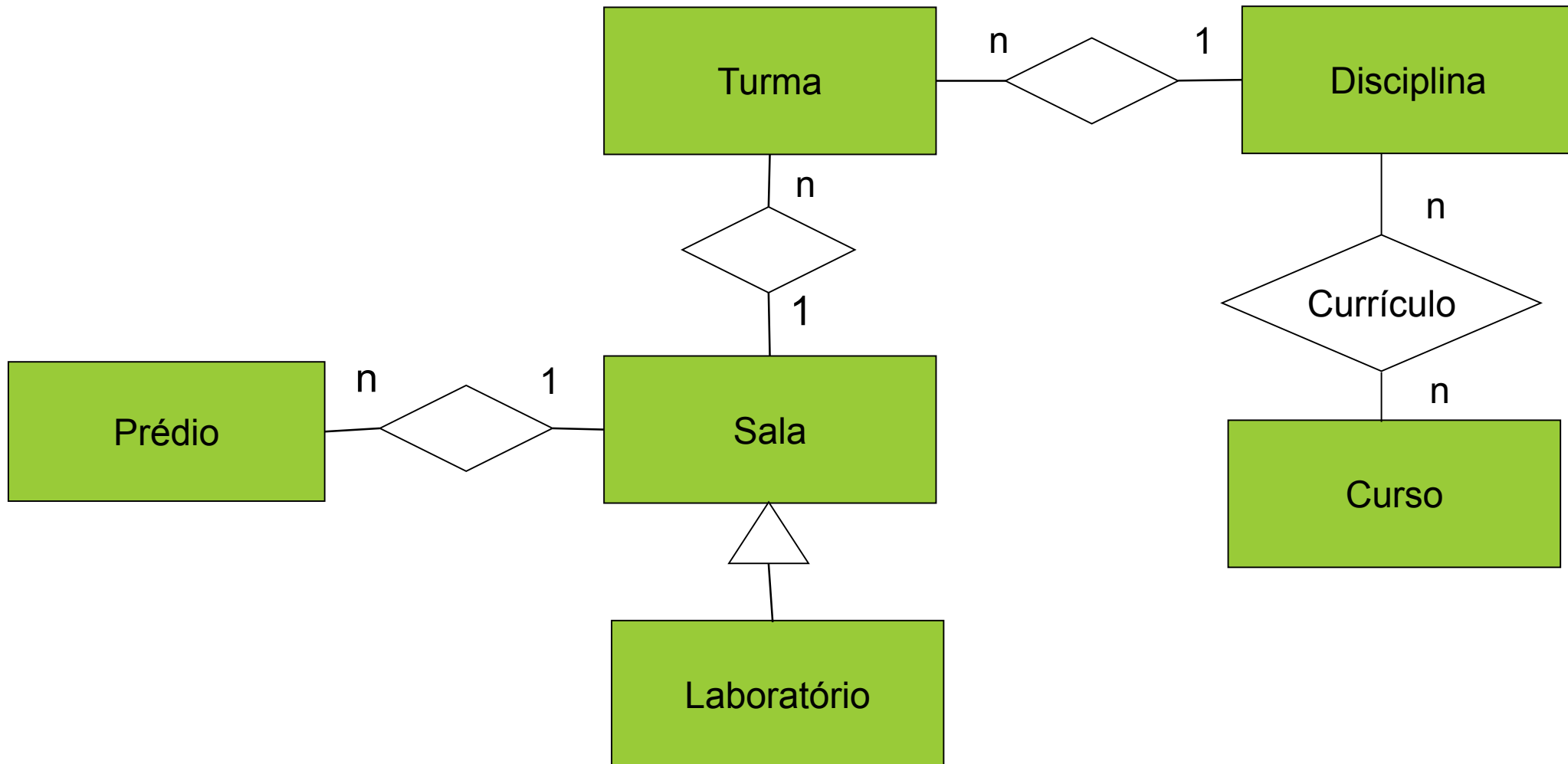
- Disciplina (idDisciplina, nomeDisciplina)
- Sala (idPredio, idSala, capacidade)
- Turma(AnoSem, idDisciplina, SiglaTur, NroAlunos, idPredio, idSala)

idDisciplina referencia Disciplina

idPredio, idSala referencia Sala



# Engenharia Reversa de Modelos Relacionais

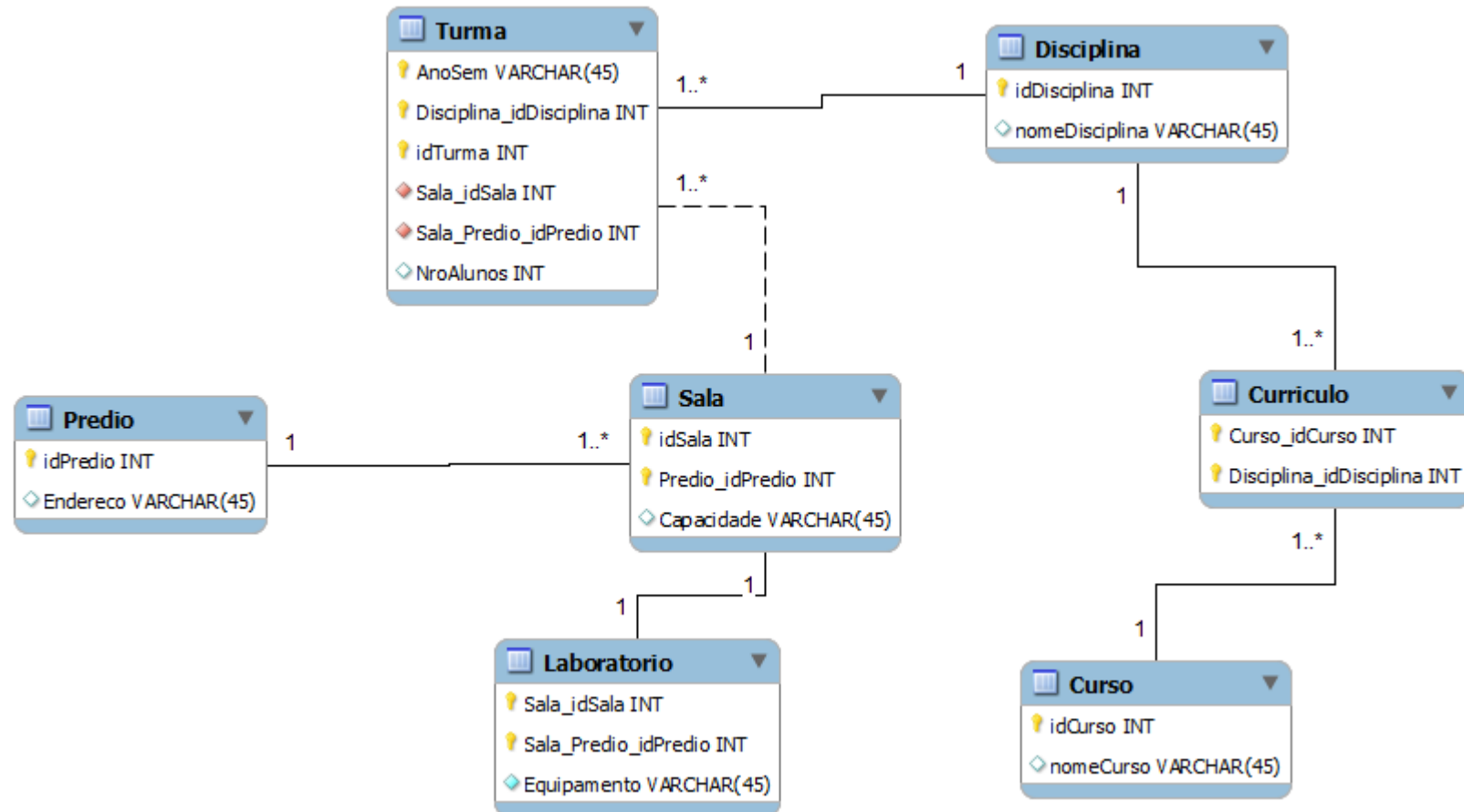


# Definição de Atributos

---

- ✓ Cada coluna de uma tabela que não seja chave estrangeira, corresponde a um atributo na ER correspondente à tabela.
- ✓ As colunas chave estrangeira não correspondem a atributos no ER mas sim a relacionamentos.

# Definição de Atributos



# Definição de Identificadores

---

- ✓ Toda coluna da chave primária e que não é chave estrangeira corresponde a um atributo identificador.
- ✓ Toda coluna que faz parte da chave primária e que é chave estrangeira corresponde a um identificador externo