

Desenvolvimento para a Web e Dispositivos Móveis

Modelação e Base de Dados

Concepção do
Modelo Lógico de Dados Relacional

Deloitte.

Sumário

- Estratégias de derivação do modelo de dados relacional
- Regras de derivação do modelo de dados relacional a partir do modelo E-R → tratamento de:
 - Relacionamentos binários de cardinalidade 1:1, 1:N e N:M, com participações obrigatórias e opcionais
 - Relacionamentos binários múltiplos
 - Relacionamentos de diversas ordens
 - Atributos multivalor
 - Tipos de entidades: independente, de agregação, de intersecção, fraca e subordinada
 - Entidades fracas e entidades subordinadas
 - Hierarquias de especialização/generalização
 - Malha de especialização/generalização
 - Categorias

Estratégias de concepção do modelo de dados Relacional

- Do particular para o geral (**Bottom-up**)

- 1) Relação universal.
- 2) Análise de dependências funcionais.
- 3) Modelo de dados.

Pequenos projectos
(até 6-8 entidades)

- Do geral para o particular (**Top-down**)

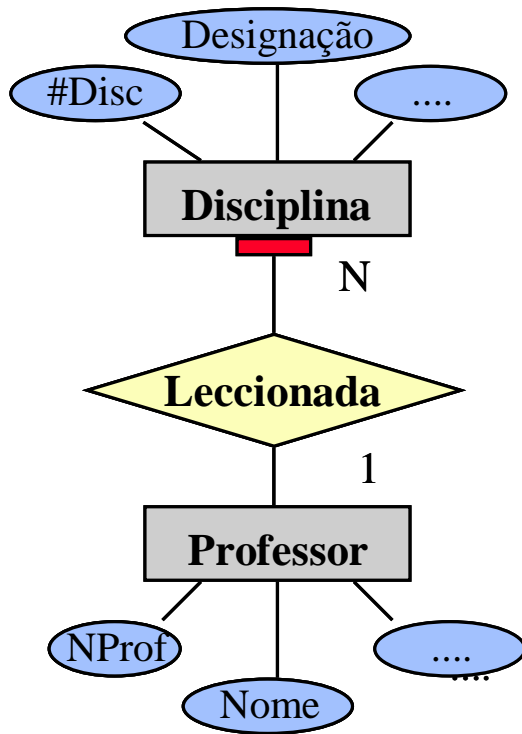
- 1) Modelo conceptual E-R.
- 2) Regras de mapeamento para modelo lógico.
- 3) Modelo de dados.

**Grandes
projectos**

Construção de modelo de dados relacional pelo método Entidade-Relacionamento

- Construir o modelo conceptual de dados:
 - identificar todas as **entidades** e todos os **relacionamentos** importantes para a situação a tratar;
 - construir o **diagrama** de Entidade-Relacionamento (DER);
 - identificar todos os **atributos relevantes** e associa-los a uma das entidades preliminares já definidas (ou a relacionamentos existentes entre estas).
- Derivar o modelo de dados relacional:
 - identificar as **chaves primárias** do conjunto de entidades preliminares;
 - aplicar as **regras** de derivação do modelo relacional;
 - **verificar o resultado** aplicando a teoria da normalização.

Regras de derivação do modelo de dados relacional



Disciplina

<u>#Disc</u>	Designação	...	NProf

Professor

<u>NProf</u>	Nome	...

Modelo conceptual - DER

Modelo lógico relacional – conjunto de tabelas interligadas

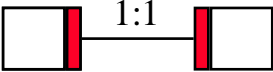
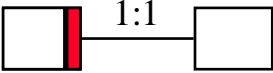
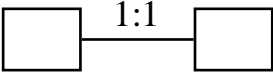
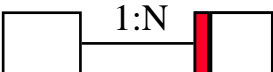
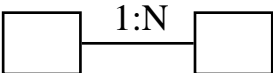
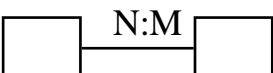
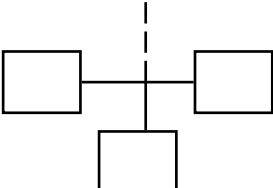
Principais factores com influência nas regras de derivação do ML

- **Cardinalidade** do relacionamento (1:1, 1:N, N:M).
- **Tipo de participação** das entidades no relacionamento:
 - obrigatória;
 - opcional.
- **Tipo de relacionamento:**
 - binário;
 - n_ário;
 - recursivo;
 - etc.
- **Tipo de entidade** (e.g. fraca)

Principais situações a tratar

- Relacionamentos binários
- Relacionamentos binários múltiplos
- Relacionamentos não binários
- Relacionamentos recursivos
- Atributos multivalor
- Entidades Fracas
- Entidades subordinadas e representação de papéis
- Hierarquias e malhas de especialização/generalização
- Categorias

Resumo das principais regras do método ER

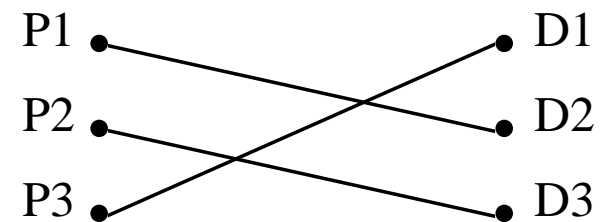
<i>Relacionamento</i>	<i>Nº Tabelas</i>	<i>Observações</i>
	1	A Chave primária pode ser a chave de qualquer das entidades.
	2	A Chave da entidade c/ participação <u>não</u> obrigatória tem de ser atributo na outra.
	3	A tabela do relacionamento terá como atributos as chaves de ambas as entidades
	2	A Chave da entidade do lado 1 tem de ser atributo na entidade do lado N.
	3	A tabela do relacionamento terá como atributos as chaves de ambas as entidades.
	3	A tabela do relacionamento terá como atributos as chaves de ambas as entidades.
	N+1	A tabela do relacionamento terá como atributos as chaves de <u>todas</u> as entidades.

Relacionamento binário 1:1 (2po)

Caso 1 (1:1) - participação obrigatória das **duas** entidades

Exemplo:

- todos os professores **têm** de leccionar uma só disciplina;
- cada disciplina **tem** de ser assegurada por um professor.



Regra 1 - 1:1 e 2po

Professor (NProf, Nome, Tel, #Disc, Prereq)

NProf	Nome	Tel	#Disc	Prereq
1001	Couto	721334	Inf2	Inf1
1662	Nunes	776188	SOC	LP
77	Peixoto	722876	Inf1	Nenhum

Regra 1 - Relacionamento binário de cardinalidade 1:1 e participação obrigatória de ambas as entidades:

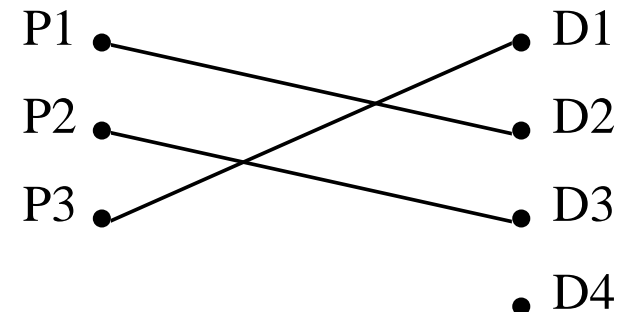
- é apenas necessária **uma tabela** (as duas entidades podem fundir-se numa só);
- a chave primária dessa tabela pode ser a chave primária de qualquer das entidades.

Relacionamento binário 1:1 (1po)

Caso 2 (1:1) - participação obrigatória de **apenas uma** das entidades

Exemplo:

- todos os **professores têm de leccionar** uma só disciplina;
- cada disciplina em funcionamento é assegurada por um professor.



Caso1:1 (1po)

- É viável recorrer a uma só tabela?

NProf	Nome	Tel	#Disc	Prereq
1001	Couto	721334	Inf2	Inf1
?	?	?	IG2	IG1

Regra 2 - **1:1 e 1po** (Professor tem de leccionar)

Professor (NProf, Nome, Tel, #Disc)

NProf	Nome	Tel	#Disc
1001	Couto	721334	Inf2

Disciplina (#Disc, Prereq)

#Disc	Prereq
Inf2	Inf1
IG2	IG1

Regra 2 - Relacionamento binário de cardinalidade 1:1 e participação obrigatória de apenas uma das entidades.

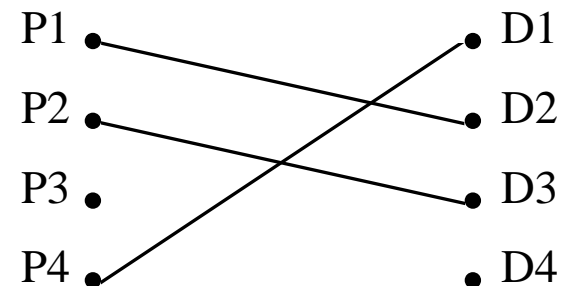
- são necessárias **duas tabelas**;
- a chave primária de cada entidade serve de chave primária na tabela correspondente;
- a chave primária da entidade com participação não obrigatória **tem de ser usada como atributo (FK) na tabela** correspondente à entidade cuja participação é obrigatória.

Relacionamento binário 1:1 (sem po)

Caso 3 (1:1) - **sem** participação obrigatória em ambas as entidades

Exemplo:

- os docentes leccionam uma só disciplina, se não estiverem dispensados do serviço docente;
- cada disciplina é assegurada por um docente, excepto se for opcional e se o número de inscrições for inferior a 15 alunos .



Caso1:1 (sem po) – uma tabela?

NProf	Nome	Tel	#Disc	Prereq
1001	Couto	721334	Inf2	Inf1
1662	Nunes	776188	SOC	LP
?	?	?	IG2	IG1
1056	Martins	734976	?	?

Utilizando uma só tabela, surgem valores nulos:

- quer para as disciplinas que ainda não têm professor;
- quer para os professores que não leccionam nenhuma disciplina

Caso1:1 (sem po) – duas tabelas?

Professor (Prof, Nome, Tel, #Disc)

NProf	Nome	Tel	#Disc
1001	Couto	721334	Inf2
1662	Nunes	776188	SOC
1056	Martins	734976	?

Disciplina (#Disc, Prereq, NDoc)

#Disc	Prereq	NDoc
Inf2	Inf1	1001
SOC	LP	1662
IG2	IG1	?

A subdivisão da entidade em duas tabelas, segundo solução análoga à regra 2, também origina valores nulos.

Regra 3 - 1:1 sem po

Professor

NProf	Nome	Tel
1001	Couto	721334
1662	Nunes	776188
1056	Martins	734976

Disciplina

#Disc	Prereq
Inf2	Inf1
SOC	LP
IG2	IG1

Leccionar

NProf	#Disc
1001	Inf2
1662	SOC



Regra 3 - Relacionamento binário de cardinalidade 1:1 e participação não obrigatória em ambas as entidades.

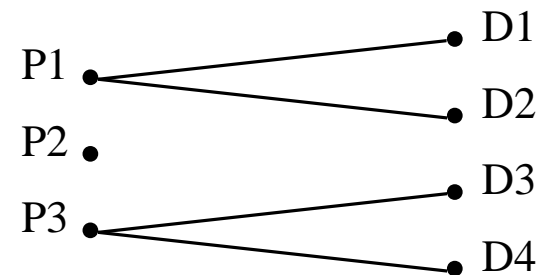
- são necessárias **três tabelas**, uma para cada entidade (e sem FK's) e a terceira para o relacionamento (com FK's);
- a chave primária de cada entidade serve de chave primária na tabela correspondente;
- a tabela correspondente ao relacionamento **terá entre os seus atributos as chaves primárias das duas entidades** (como FK's e frequentemente são também a PK).

Relacionamento binário 1:N (po lado N)

Caso 4 (**1:N**) - participação obrigatória do lado N
(a participação obrigatória no lado 1 não afecta resultado)

Exemplo:

- os professores podem leccionar várias disciplinas;
- cada **disciplina têm de ser assegurada** por um só professor.




Regra 4 - 1:N e po lado N (Disciplina tem de ser leccionada)

Professor (Ndoc, Nome, Tel)

NProf	Nome	Tel
1662	Nunes	776188
1056	Martins	734976

Disciplina (#Disc, Prereq, NProf)

#Disc	Prereq	NProf
SOC	LP	1662
SDP	SOC	1662



Regra 4 - Relacionamento binário de cardinalidade 1:N e participação obrigatória do lado N:

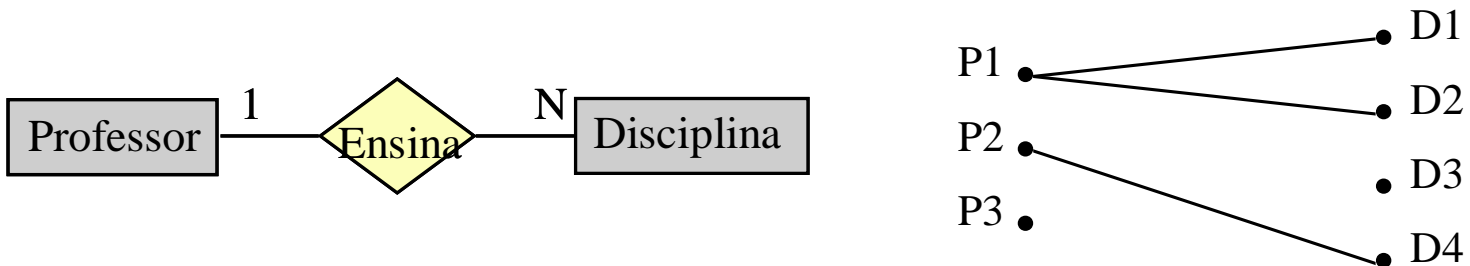
- são necessárias **duas tabelas**;
- a chave primária de cada entidade serve de chave primária na tabela correspondente;
- a chave primária da entidade do lado 1 tem de ser usada como **atributo (FK) na tabela correspondente à entidade do lado N**.

Relacionamento binário 1:N (sem po lado N)

Caso 5 (**1:N**) - participação não obrigatória do lado N

(a participação obrigatória no lado 1 não afecta resultado)

Exemplo:



Regra 5 - 1:N e (sem po lado N)

Professor

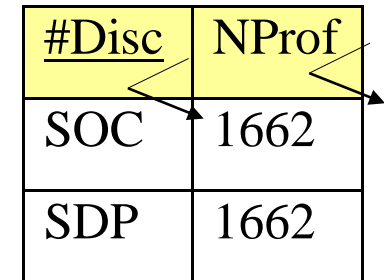
NProf	Nome	Tel
1662	Nunes	776188
1056	Martins	734976

Disciplina

#Disc	Prereq
SOC	LP
SDP	SOC
IG2	IG1

Leccionar

<u>#Disc</u>	NProf
SOC	1662
SDP	1662



Regra 5 - Relacionamento binário de cardinalidade 1:N e participação não obrigatória do lado N.

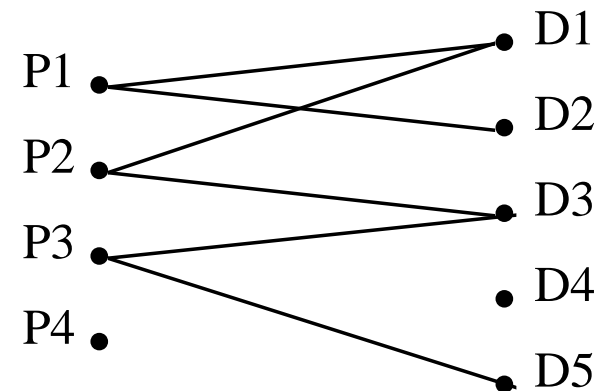
- são necessárias **três tabelas**, uma para cada entidade e a terceira para o relacionamento;
- a chave primária de cada entidade serve de chave primária na tabela correspondente;
- a tabela relativa ao relacionamento terá de ter entre os seus atributos (como FK's) as chaves primárias de cada uma das entidades (tipicamente a FK respeitante ao lado N é usada simultaneamente como PK).

Relacionamento binário N:M

Caso 6 (N:M) - independentemente do tipo de participação

(a participação obrigatória não afecta resultado)

Exemplo:



Regra 6 – cardinalidade N:M

Professor

NProf	Nome	Tel
1001	Couto	721334
1662	Nunes	776188
1033	Reis	716623
1052	Neves	714356
1056	Martins	734976

Disciplina

#Disc	Prereq
Inf2	Inf1
SOC	LP
SDP	SOC
IA	LP
IG2	IG1

Leccionar

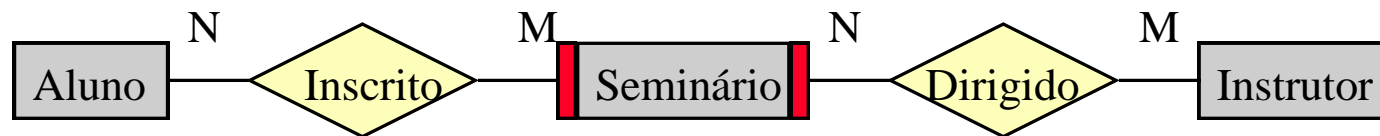
#Disc	NProf
Inf2	1001
SOC	1662
SDP	1662
IA	1033
IA	1052

Regra 6 - Relacionamento binário de cardinalidade N:M:

- são **sempre** necessárias **três tabelas**, uma para cada entidade e uma terceira para o relacionamento;
- a chave primária de cada entidade serve de chave primária na tabela correspondente;
- a tabela relativa ao relacionamento terá de ter entre os seus atributos (FK's) as chaves primárias de cada uma das entidades (usualmente constituem também a PK).

Relacionamentos binários múltiplos (1/4)

- Na maioria dos casos, uma entidade pode ter relacionamentos binários com diversas entidades, ou seja, relacionamentos binários múltiplos.
- Exemplo:
 - um aluno pode inscrever-se em vários seminários;
 - um seminário é dirigido por vários instrutores;
 - um instrutor dirige vários seminários.



3 tabelas

3 tabelas

Aluno (N_aluno,)

Seminário (N_Seminário,)

Inscrição (N_Seminário, N_aluno, ...)



Instrutor (N_Instrutor,)

Seminário (⇐)

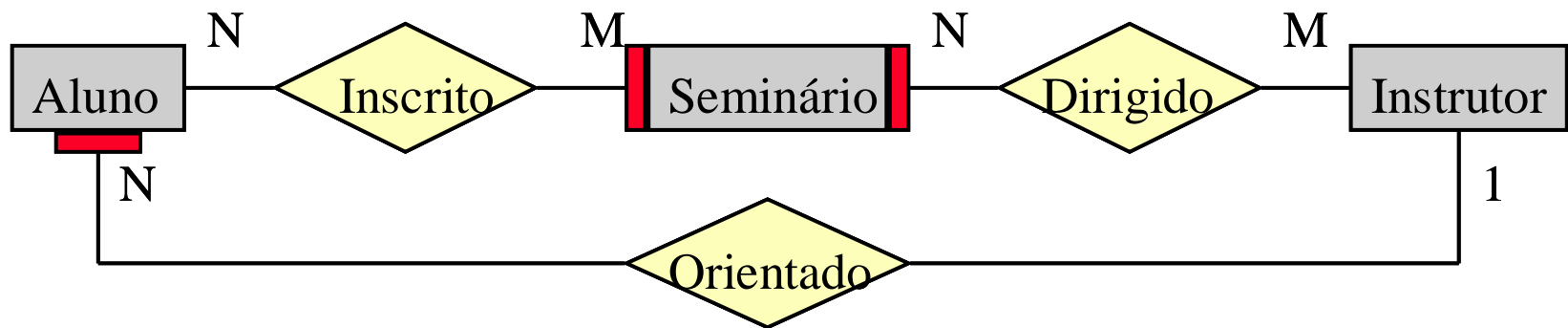
Direcção (N_seminário, N_Instrutor,...)



Relacionamentos binários múltiplos

(2/4)

Supondo que um aluno tem de ser orientado por um instrutor nos vários seminários, obtém-se:



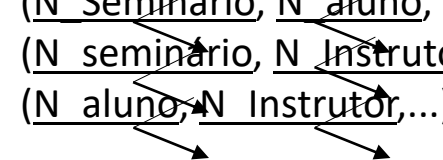
Relacionamentos binários múltiplos

(3/4)

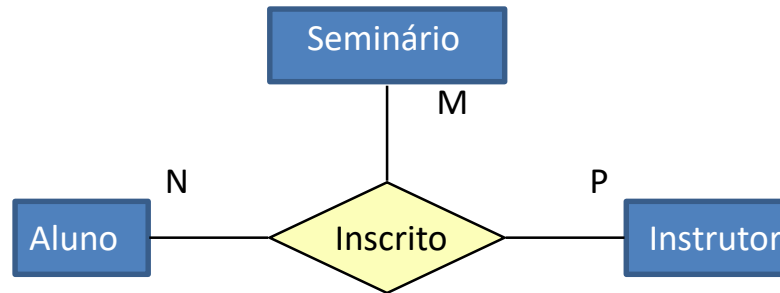
- O novo relacionamento dá origem às seguintes tabelas (mais propriamente, origina uma nova FK numa das tabelas já existentes):
 - **Aluno** (N_aluno, ..., N_instrutor, ...)
 - **Instrutor** (N_Instrutor,)
- O Modelo de dados final seria:
 - **Aluno** (N_aluno, ..., N_instrutor, ...)
 - **Instrutor** (N_Instrutor,)
 - **Seminário** (N_Seminário,)
 - **Inscrição** (N_Seminário, N_aluno,)
 - **Direcção** (N_seminário, N_Instrutor, ...)

Relacionamentos binários múltiplos

(4/4)

- Supondo agora, que:
 - o mesmo aluno pode ter vários instrutores;
 - os instrutores poderão ser diferentes consoante o seminário;
 - ⇒ o relacionamento “orientado” passaria a ser do tipo N:M.
- O modelo de dados final passaria a ser:
 - **Aluno** (N aluno,)
 - **Instrutor** (N Instrutor,)
 - **Seminário** (N Seminário,)
 - **Inscrição** (N Seminário, N aluno,)
 - **Direcção** (N seminário, N Instrutor,...)
 - **Orientação** (N aluno, N Instrutor,...)
- **Questão (já discutida):**
 - quem é(são) o(s) orientador(es) de um aluno num dado seminário ?
 - só é possível determinar quais são os instrutores de um seminário e quais são os orientadores de um dado aluno.

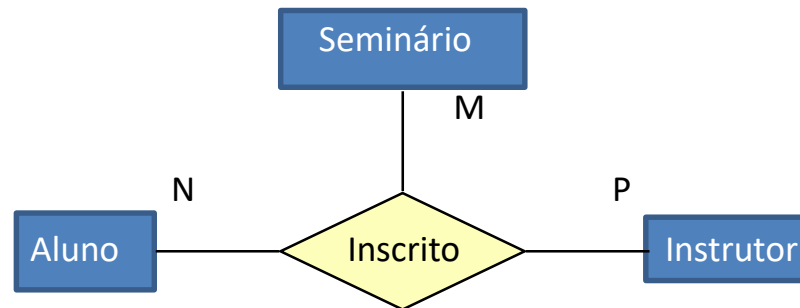
Relacionamento ternário



Regra 7 - Relacionamento ternário (e superior):

- são sempre necessárias **quatro tabelas**, uma para cada entidade e uma quarta para o relacionamento;
- a chave primária de cada entidade serve de chave primária na tabela correspondente;
- a tabela relativa ao relacionamento terá de ter **entre os seus atributos (como FK) as chaves primárias** de cada uma das entidades;
- num relacionamento de **grau n são necessárias n+1 relações**, de modo inteiramente idêntico.

Modelo resultante de relacionamento ternário

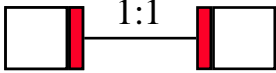
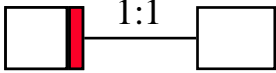
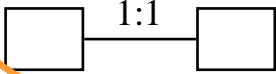
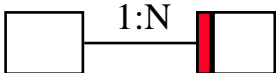
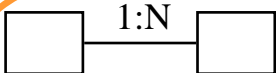
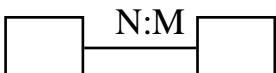
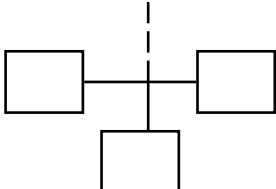


O modelo de dados final passaria a ser:

- **Aluno** (N_aluno,)
- **Instrutor** (N_Instrutor,)
- **Seminário** (N_Seminário,)
- **Inscrição** (N_Seminário, N_aluno, N_instrutor,....)

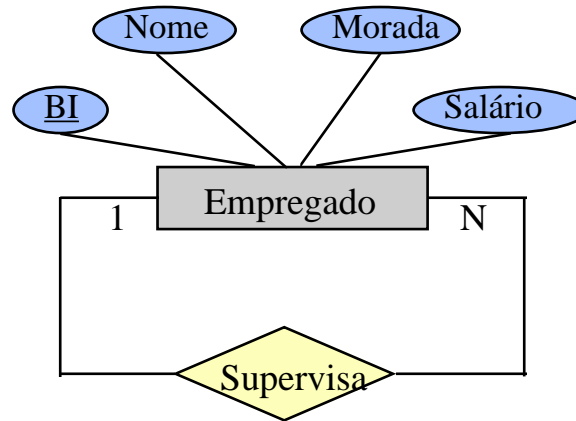
Obs: Se cada aluno tiver um só instrutor num dado seminário, a chave primária da entidade seria somente N_Aluno, N_seminário.

Resumo das principais regras do método ER

Relacionamento	Nº Tabelas	Observações
	1	A Chave primária pode ser a chave de qualquer das entidades.
	2	A Chave da entidade c/ participação <u>não</u> obrigatória tem de ser atributo na outra.
	3	A tabela do relacionamento terá como atributos as chaves de ambas as entidades
	2	A Chave da entidade do lado 1 tem de ser atributo na entidade do lado N.
	3	A tabela do relacionamento terá como atributos as chaves de ambas as entidades.
	3	A tabela do relacionamento terá como atributos as chaves de ambas as entidades.
	N+1	A tabela do relacionamento terá como atributos as chaves de <u>todas</u> as entidades.

Objectivo =>
evitar valores NULOS em FK

Relacionamentos recursivos



Relacionamento não obrigatório do lado N (aplica-se a regra 5)

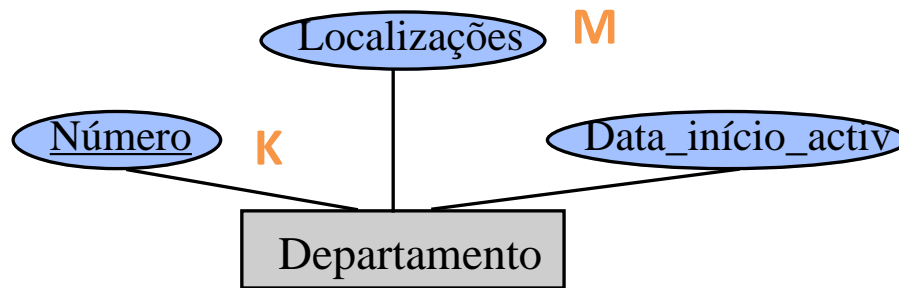
=> 3 tabelas: Empregado, Empregado e “Supervisão” (omitindo a repetição)

- **Empregado** (BI, Nome, Morada, Salário)
- **Supervisão** (BI empregado, BI_Supervisor)

Regra para relacionamentos recursivos

Usam-se as regras já definidas para os relacionamentos não recursivos.

Atributos Multivalor



O atributo localizações é um atributo multivalor, pois um departamento possui várias localizações \Rightarrow

- **Departamento** (Número, Data_início_activ)
- **Local** (Número, Localização)



Regra para atributos multivalor

- um atributo multivalor **M** origina uma nova tabela;
- a nova tabela vai conter esse atributo **M** e a chave estrangeira **K**, sendo **K** o conjunto de atributos que constituem a chave primária da entidade já existente;
- a chave primária da nova tabela será constituída pela combinação de **M** com **K**.

Tipos de entidades

- Entidades independentes
- Entidades de agregação
- Entidades de intersecção
- Entidades dependentes ou fracas
- Entidades subordinadas

Entidades independentes e de agregação

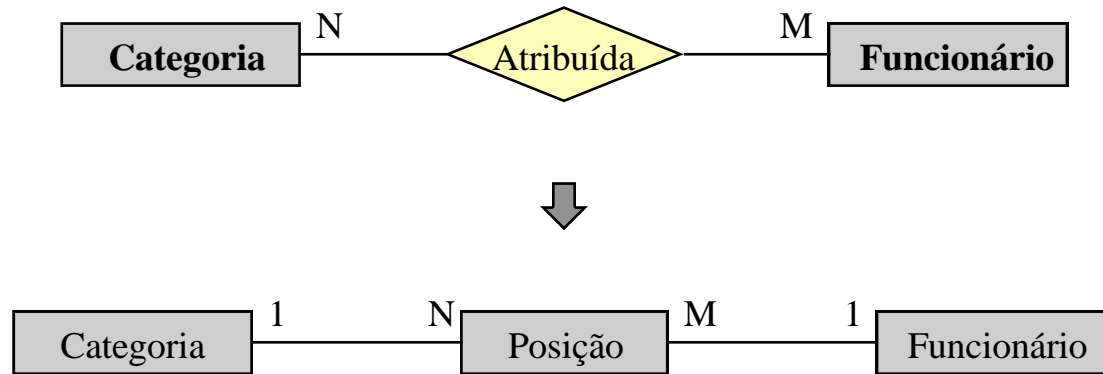
- **Entidades independentes**

- são, frequentemente, entidades **centrais** num modelo de dados;
- possuem **nomes claramente distinguíveis**, pelo facto de ocorrerem no mundo real;
- por norma, possuem **chaves simples** (e.g. código de Departamento ou código de Funcionário).
- Exemplos:
 - **Funcionário** (N_Funcionário, Nome, ...)
 - **Departamento** (N_Departamento, Designação_departamento, ...)

- **Entidades de agregação**

- são criadas quando várias entidades diferentes possuem **atributos similares distinguíveis somente pelos prefixos ou sufixos**;
 - normalmente tornam-se entidades independentes.
 - Exemplo:
 - **Cliente** (#cliente, morada_cliente, telefone_cliente, Fax_cliente)
 - **Fornecedor** (#fornecedor, morada_fornec, telefone_fornec, Fax_fornec.)
- ⇒ **Contactos** (#entidade, *Tipo_entidade*, morada, telefone, fax)

Entidades de intersecção (1/2)



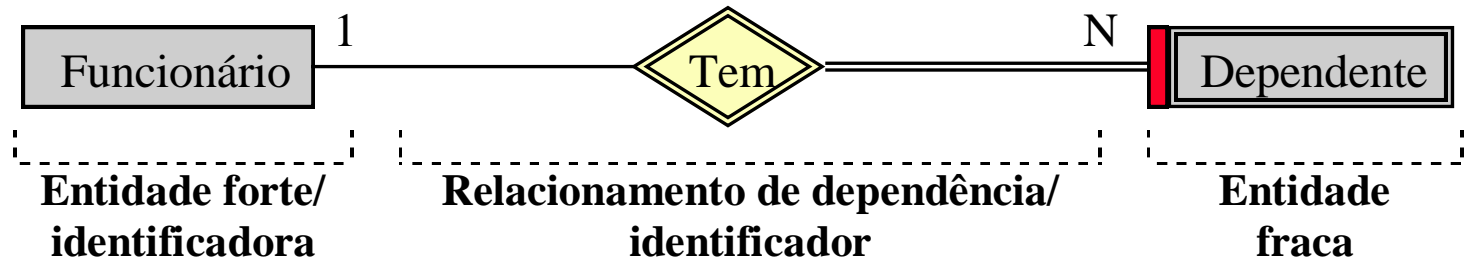
- Resultam de relacionamentos:
 - com cardinalidade N:M;
 - sem participação obrigatória em ambas as entidades.
- Por vezes, estas entidades possuem nomes óbvios pelo facto de ocorrerem no mundo real.
- Em caso contrário, utilizam-se, frequentemente, os nomes das duas entidades (e.g. Categoria_Funcionário).

Entidades de intersecção (2/2)

- Podem representar:
 - Relacionamentos correntes - a entidade inclui nos seus atributos ambas as chaves primárias das entidades iniciais;
 - Relacionamentos históricos - Para além dos atributos que constituem a chave primária das entidades iniciais, a entidade possui atributos de medidas temporais.
- Se a entidade de intersecção tiver uma chave primária própria, esta torna-se uma entidade independente.

Entidades dependentes ou fracas (1/2)


- Depende de outra entidade na sua existência e/ou identificação.
- Exemplo:



- **Características de entidades dependentes ou fracas:**
 - num relacionamento de dependência a **participação da entidade fraca é sempre obrigatória**;
 - se a entidade fraca não possui atributos que possam constituir chaves candidatas => *o conjunto de atributos que permitem identificar univocamente uma ocorrência da entidade fraca, para uma dada ocorrência da entidade identificadora, é a **chave parcial** da entidade fraca.*

Entidades dependentes ou fracas (1/2)

Entidades resultantes do exemplo:

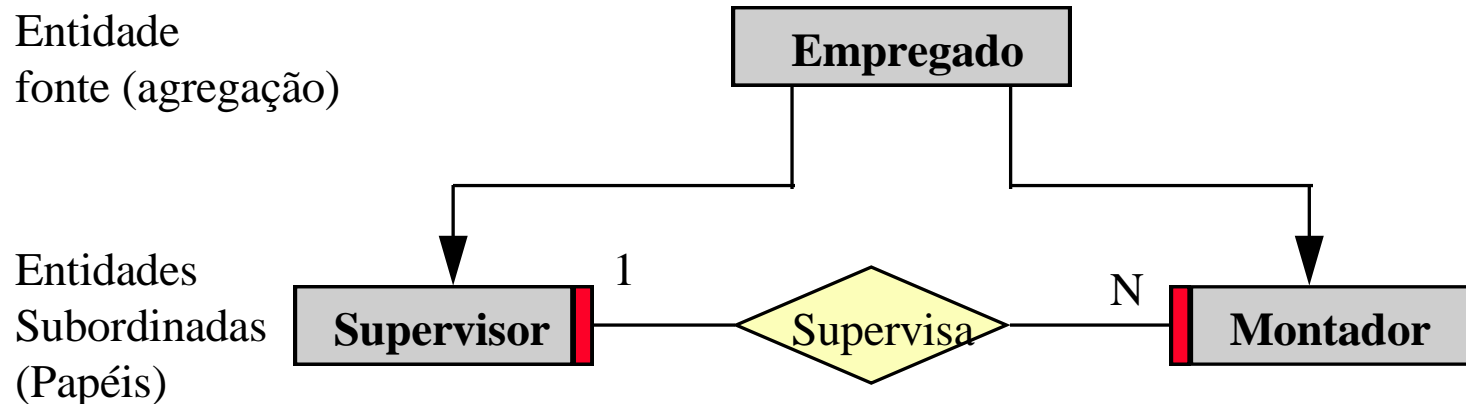
- **Funcionário** (^KN funcionário, Nome_f,)
 - **Dependente** (N funcionário, Nome D, Parentesco,)
- 

Regra para entidades dependentes ou fracas

- a entidade fraca terá de incluir nos seus atributos a chave estrangeira ^K, sendo ^K a chave primária da entidade forte;
- se existe dependência da identificação, a chave primária da entidade fraca é a combinação da sua chave parcial com ^K.

Entidade subordinada e representação de papéis (1/2)

Estas entidades são necessárias quando existem instâncias que possuem atributos específicos não associados com todos os membros de uma entidade



Entidade subordinada e representação de papéis (2/2)

Regra para instâncias com atributos diferentes ou situação de uso de papéis

- a entidade fonte gera uma tabela que agrupa os **atributos comuns** e cuja chave primária é a chave da entidade;
- os atributos específicos são separados em **entidades subordinadas** (papéis) e geram tabelas correspondentes;
- a chave da entidade fonte terá de ser usada como atributo nas tabelas correspondentes às entidades subordinadas (simultaneamente **como FK e PK**);
- as entidades que representam os papéis são tratadas como entidades normais, às quais se aplicam as regras já conhecidas.
- Aplicando a regra 4 e a presente regra, obtém-se:
 - **Empregado** (Ncontrib, Nome, Telef_casa, Morada)
 - **Supervisor** (NcontribS, Telef_trab, Salário, Área)
 - **Montador** (NcontribM, Pagam_hora, #taref, NcontribS)

regra 4
- Conforme já foi referido, **esta representação possui limitações**.

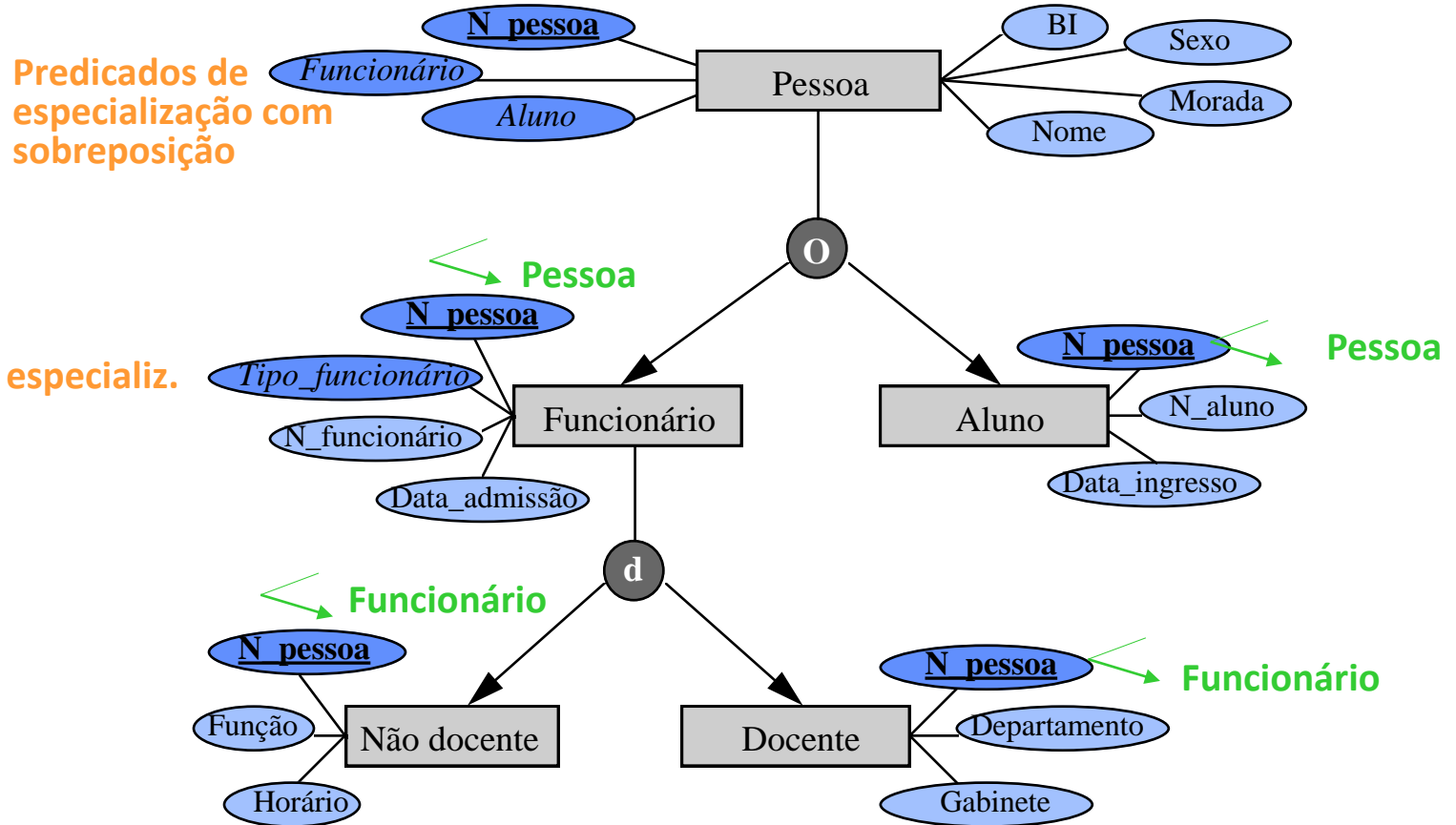
Relacionamentos superentidade e subentidade e hierarquia de especialização ou generalização

- A implementação de relacionamentos superentidade-subentidade (superclasse-subclasse) pode ser representada por tabelas distintas, **ligadas pelo atributo chave em comum** com a superentidade.
- Usualmente, existe uma condição (predicado), baseada no valor de algum atributo, que permite avaliar se um objecto pertence a uma subentidade:
 - no caso de uma disjunção pode usar-se **um atributo que recebe um só valor**;
 - no caso de sobreposição pode usar-se **vários atributos binários**

Regra para relacionamentos Superentidade-subentidade e hierarquias E/G

- a superentidade gera uma tabela com chave primária K;
- cada subentidade gera uma tabela que contém:
 - os atributos específicos;
 - o conjunto de atributos K, que também deve ser a chave primária da tabela (=> simultaneamente como FK e PK).

Exemplo de hierarquia E/G



Considerações adicionais

O modelo relacional **não proporciona suporte nativo** a hierarquias de especialização ou generalização =>

- Ponderar soluções alternativas.
- Decompor em várias tabelas e aplicar a regra em causa quando se justifica:
 - existem relacionamentos específicos relevantes;
 - o número de atributos específicos e comuns são ambos significativos.
- **Assegurar integridade** garantindo regras adicionais.

Algumas alternativas de implementação

- Criar tabelas só para subentidades (sem criar superentidade) =>

Se:

- as subentidades são disjuntas e
 - a especialização é total (cada instância pertence sempre a uma das superentidades)
- ⇒ criar uma **tabela para cada subentidade** (sem criar uma tabela para a superentidade), cada contendo os atributos comuns e os específicos.

OBS:

- no caso de **sobreposição** conduz a **redundância** dos atributos comuns;
- no caso de especialização **parcial** **perdem-se** instâncias.

- Criar apenas a tabela para superentidade (sem criar subentidades)=>

Se existem poucos atributos específicos nas subentidades:

- ⇒ criar **uma tabela** que conterà os atributos comuns, os atributos específicos e um ou vários atributos (respectivamente, no caso de disjunção e sobreposição) para indicar a(s) subentidade(s) a que o tuplo pertence.

Algumas regras para inserções e apagamento

- **Apagar** uma ocorrência da **superentidade** obriga a **apagá-la** automaticamente em todas as **subentidades** a que a ocorrência pertence.
- **Inserir** uma nova ocorrência numa **superentidade** com especialização (obrigatória ou total) obriga a inseri-la em pelo menos uma das **subentidades** (**também é conveniente definir predicado**).
- **Inserir** uma nova ocorrência numa superentidade com especialização definida por **predicados** obriga a inseri-la em todas as **subentidades** que satisfazem o predicado.
- Ao **inserir** uma ocorrência numa superentidade com especialização disjunta é necessário garantir que a ocorrência é **apenas inserida numa** das subentidades.
-

Subentidades partilhadas e malhas de especialização ou generalização

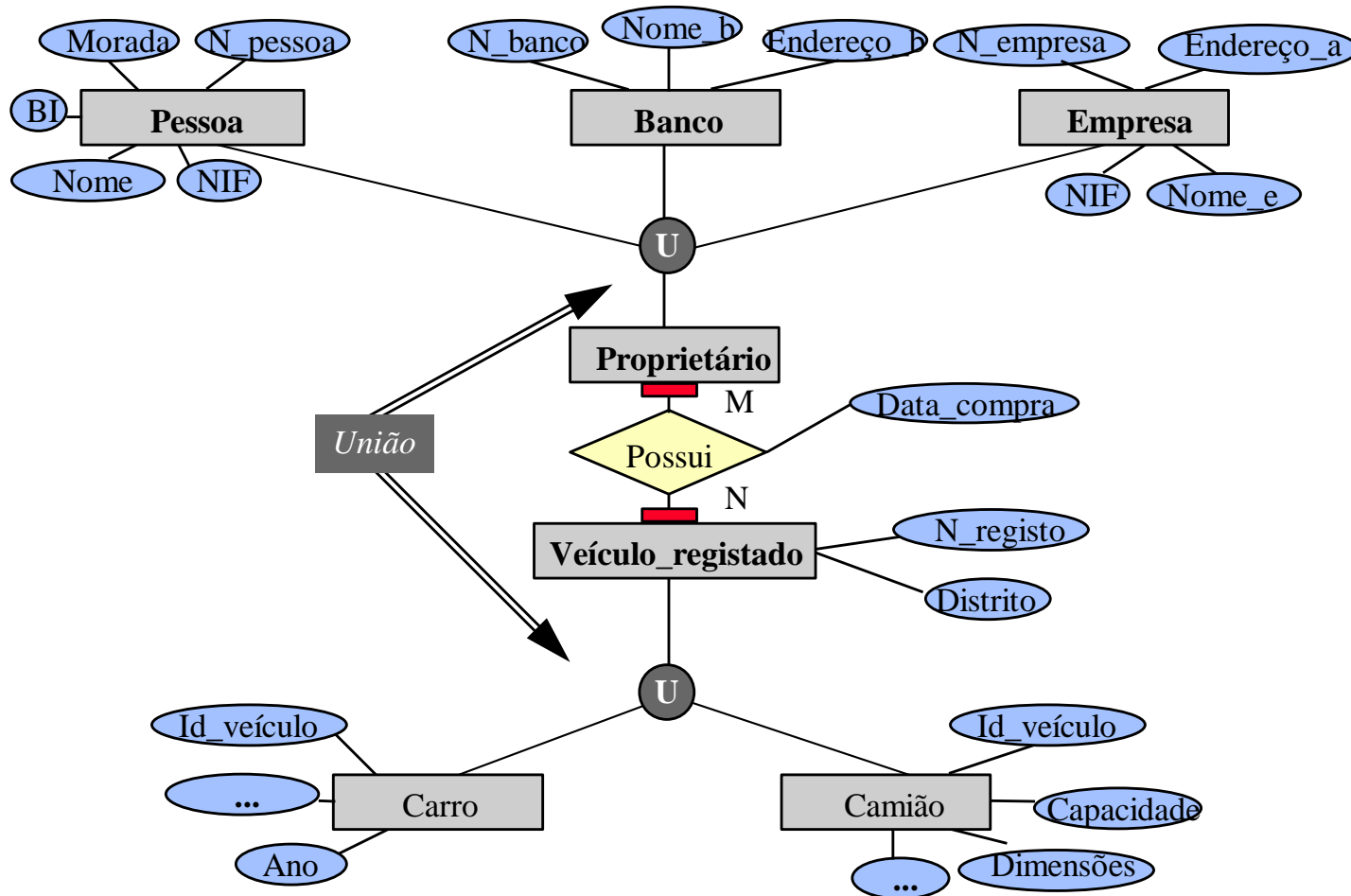
Regra para subentidades partilhadas e malhas de especialização/generalização

Aplicam-se as regras anteriores, pois todas as entidades têm de possuir a mesma chave primária.

⇒ A subentidade partilhada recebe o atributo K comum às suas superentidades.

Categorias

As superentidades de uma categoria podem ter atributos chave diferentes ou iguais



Categorias – PK distintas

Regra para categorias cujas superentidades possuem chaves primárias distintas

- criar uma **chave substituta** S na categoria;
- adicionar a **chave substituta** S (como chave estrangeira) a cada superentidade, para especificar as correspondências em valores entre a chave substituta e a chave de cada superentidade.

Categorias – PK distintas

Regra para categorias cujas superentidades possuem a mesma chave primária

A chave primária das superentidades pode ser utilizada para relacionar as entidades.

Exemplo - modelo de dados resultante

Chave substituta: **Id_proprietário**

Pessoa (N_Pessoa, ..., **Id_proprietário**)

Banco (N_banco, ..., **Id_proprietário**)

Empresa (N_empresa, ..., **Id_proprietário**)

Proprietário (Id_proprietário, ...) categoria cujas superentidades possuem PK distintas

Veículo_registado (Id_veículo, ...) categoria cujas superentidades possuem PK semelhantes

Carro (Id_veículo, ...)

Camião (Id_veículo, ...)

Registo_propriedade (Id_veículo, Id_proprietário, ..., data_compra)