

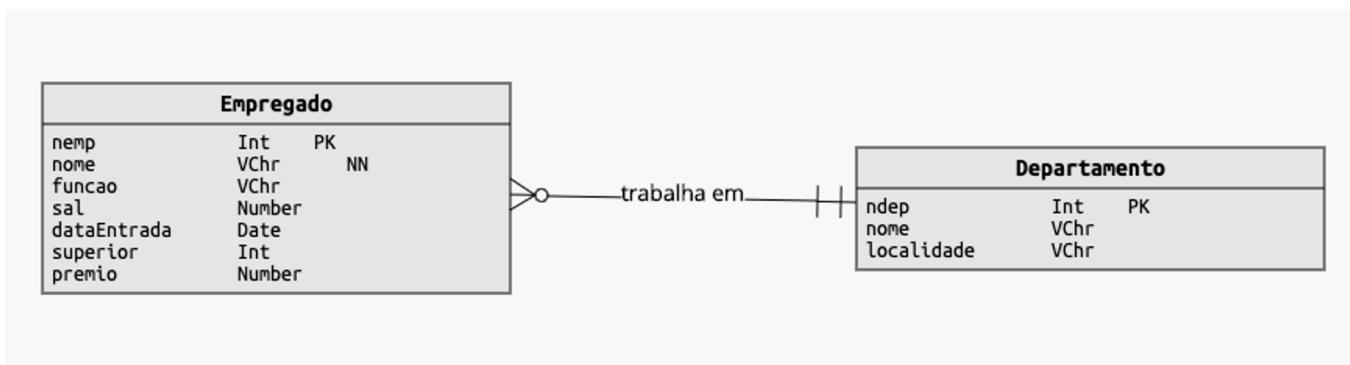
Bases de Dados e Análise de Informação

Pedro Furtado

Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia da Universidade de Coimbra
2019->

Tradução ER-fisico

- Porquê?



Duas tabelas

Emp: (nemp, Nome, **ndep**)
 1,Pedro, **10**
 2, Manel, **10**
 3, Tiago, **20**
 4,Jorge

Dep (ndep, nome)
10,Marketing
20,Engeneering
30,Financas



Tradução ER-fisico

- Porque é que agora deu diferente? Haveria alternativa?

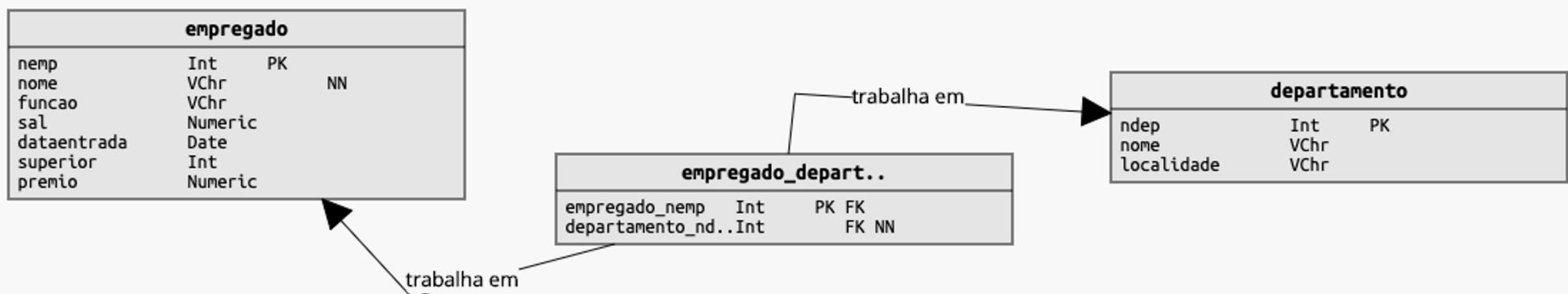
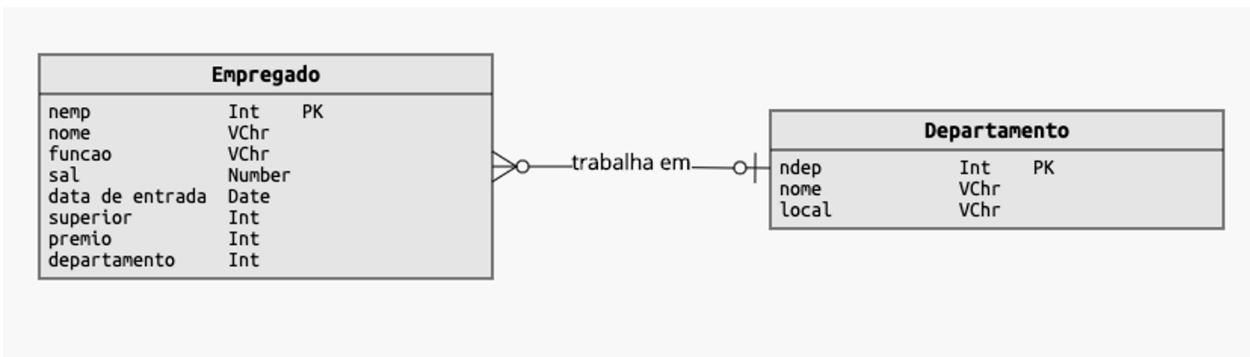


Tabela no meio

Emp: (nemp, Nome)

1,Pedro
2, Manel
3, Tiago
4,Jorge

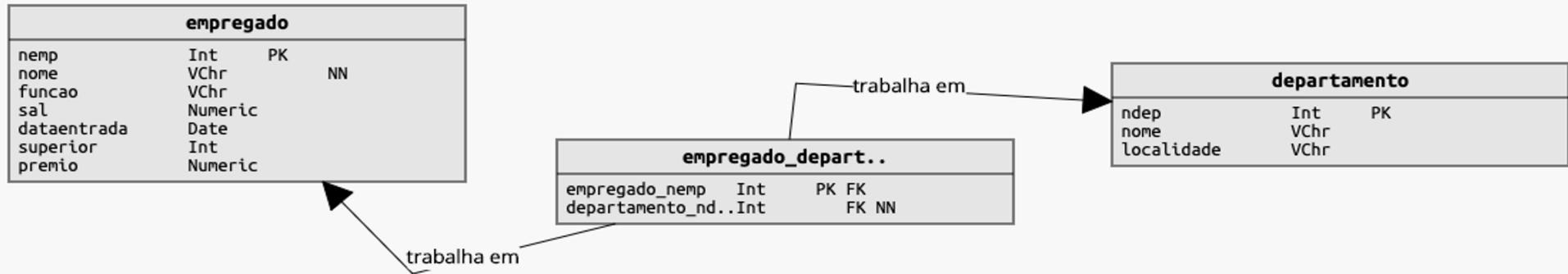
Empdep (nemp FK, ndep FK)

1,10
2,10
3,20

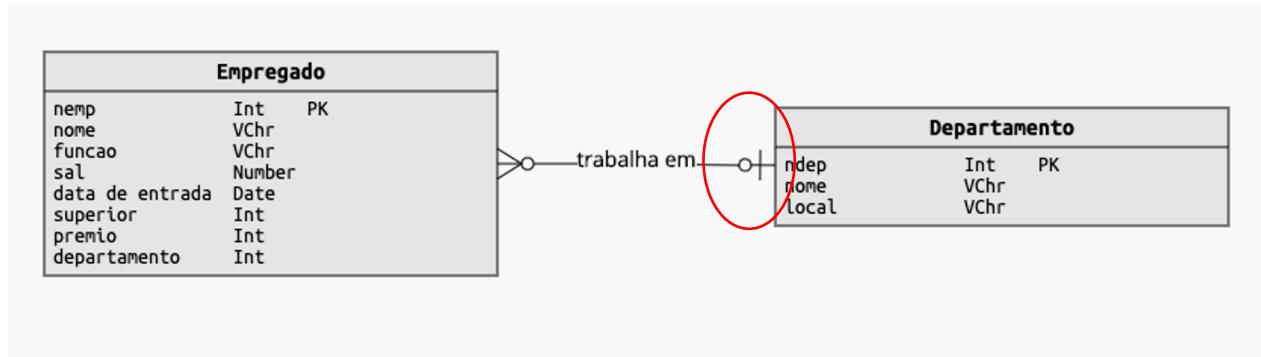
Dep (ndep, nome)

10,Marketing
20,Engeneering
30,Financas

UC

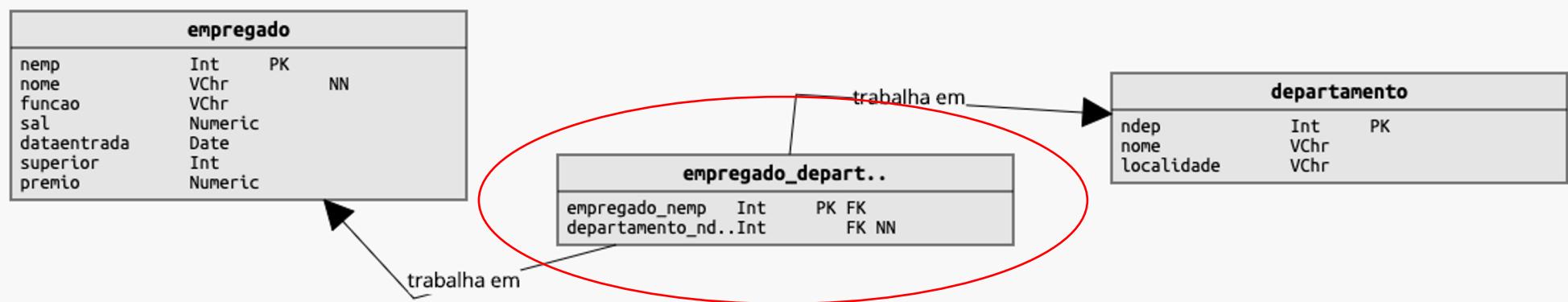


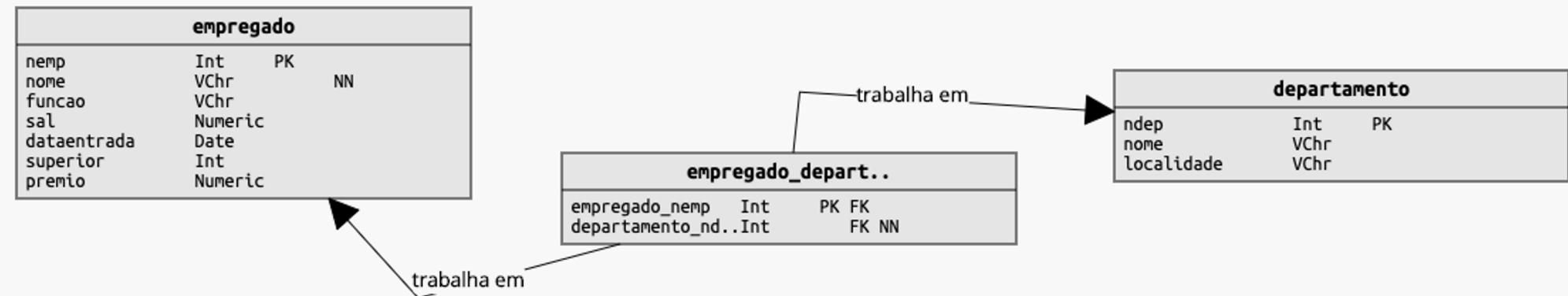
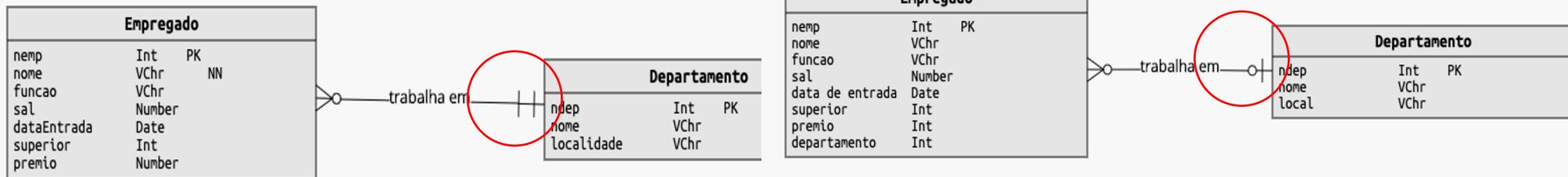
Why is there a third relationship table?



UC

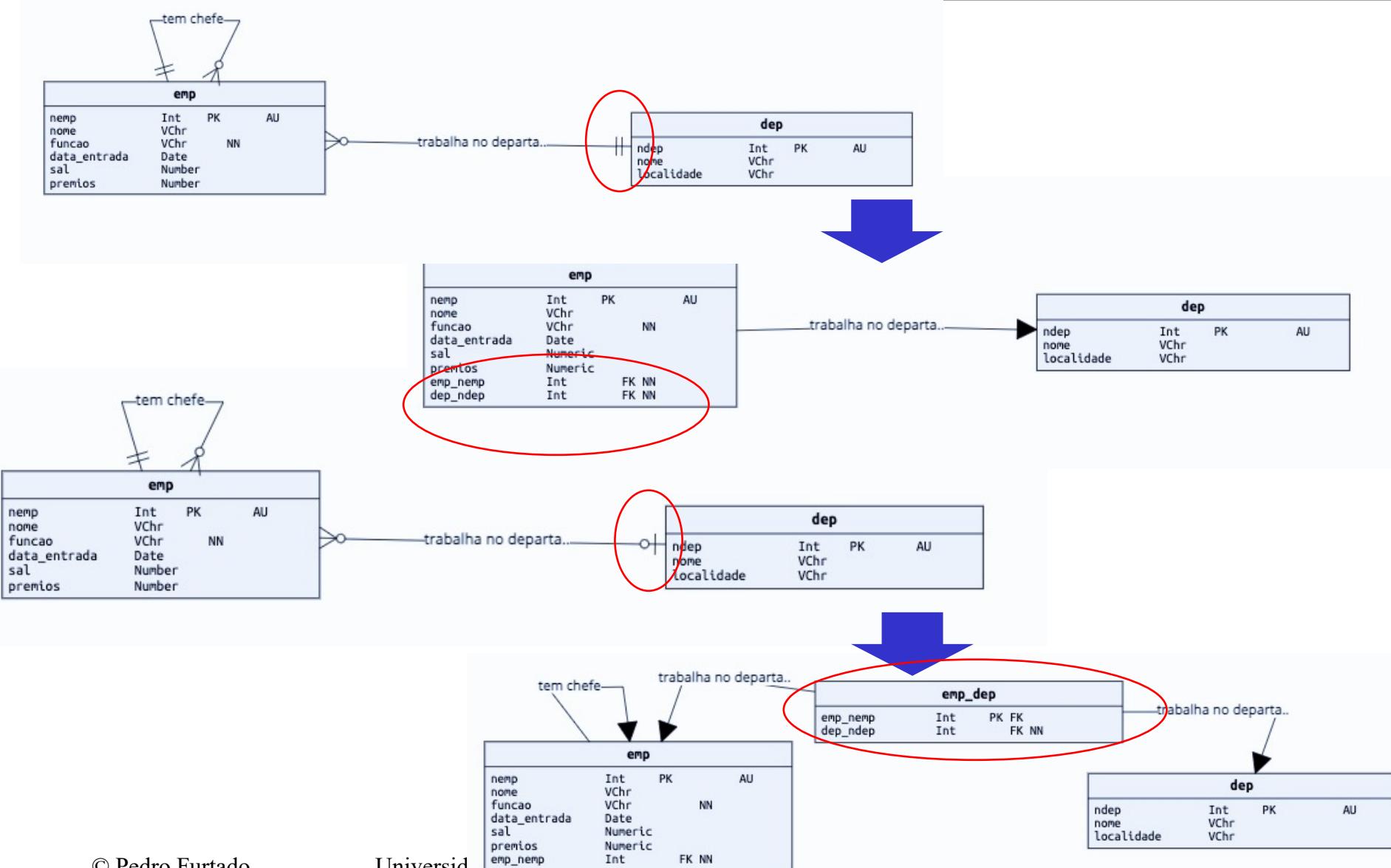
Because it avoids null foreign key values...



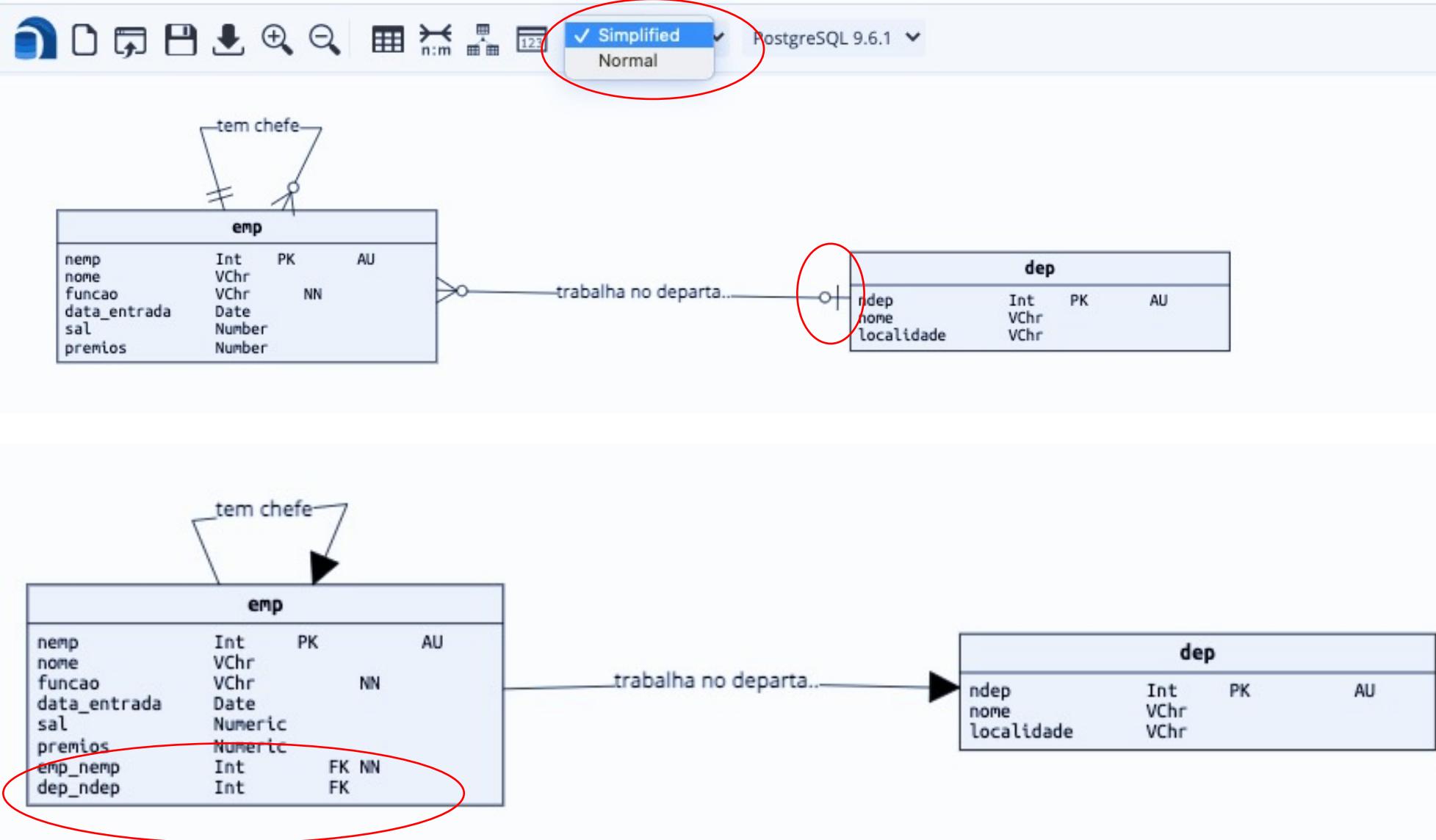


- In this case, we prefer not to have any tables in the middle if possible...
- ... Easier to deal with.

But there is a (better) alternative...



SIMPLE...

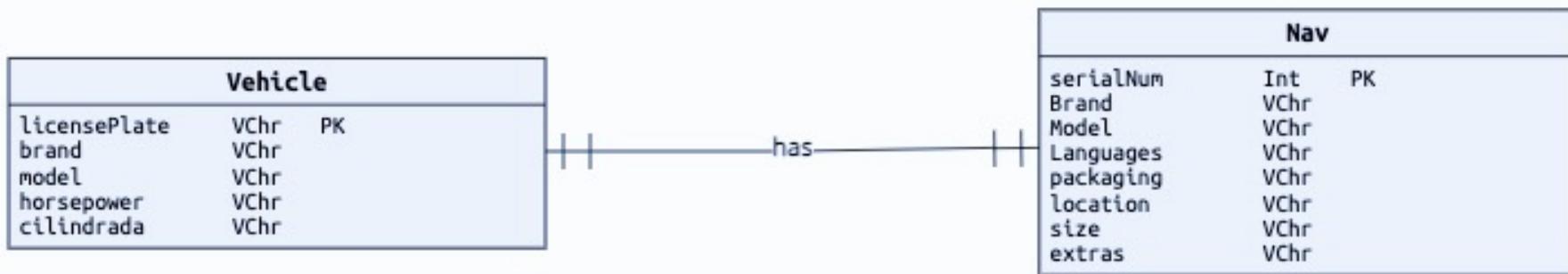


Tradução ER-Relacional

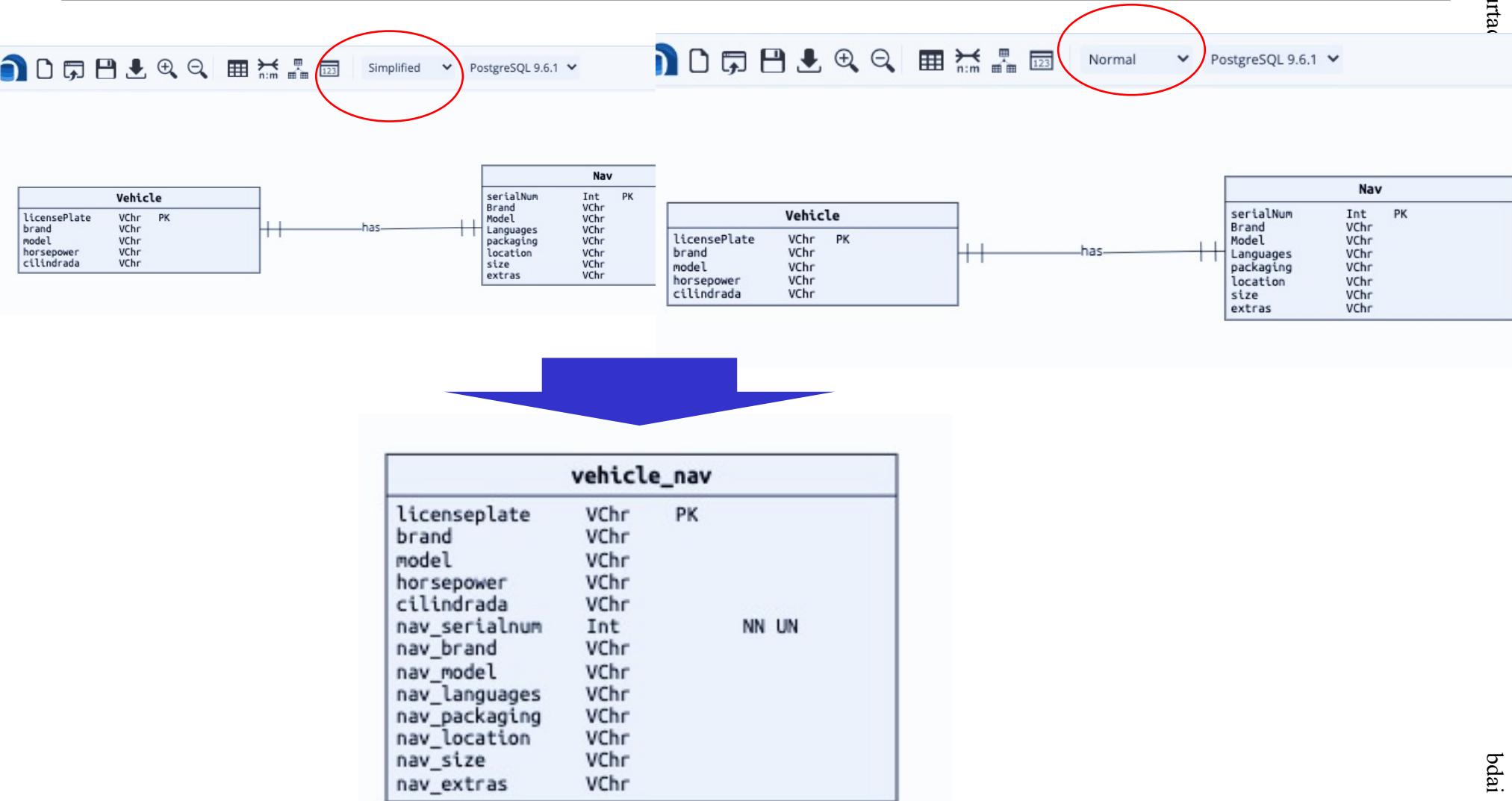
There are two options

- **NORMAL** (default)
 - Fk attributes cannot be null (empty values, not point at any PK value)
 - Leads to more tables, undesirable effect
- **SIMPLIFIED** (preferred)
 - FK attributes can be null (empty values)
 - Decreases the number of relationship tables dramatically

1:1 (mandatory:mandatory)



1:1 mandatory (SIMPLIFIED OR NORMAL)



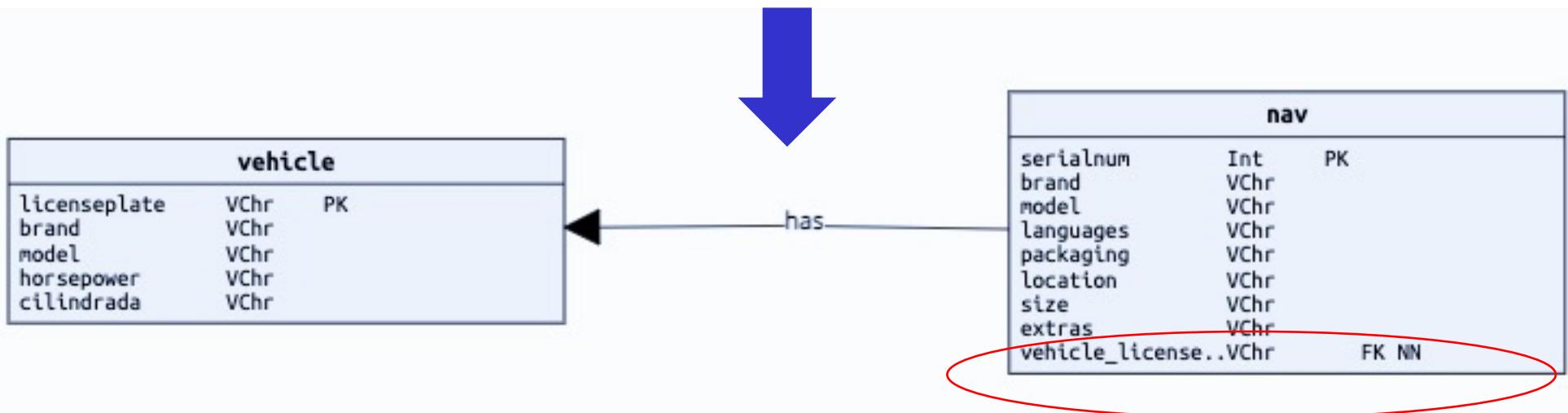
Regra 1

Relacionamento binário de grau 1:1 e participação obrigatória de ambas entidades

- É apenas necessária uma tabela.
- A chave primária dessa tabela pode ser a chave primária de qualquer das entidades.

vehicle_nav		
licenseplate	VChr	PK
brand	VChr	
model	VChr	
horsepower	VChr	
cilindrada	VChr	
nav_serialnum	Int	NN UN
nav_brand	VChr	
nav_model	VChr	
nav_languages	VChr	
nav_packaging	VChr	
nav_location	VChr	
nav_size	VChr	
nav_extras	VChr	

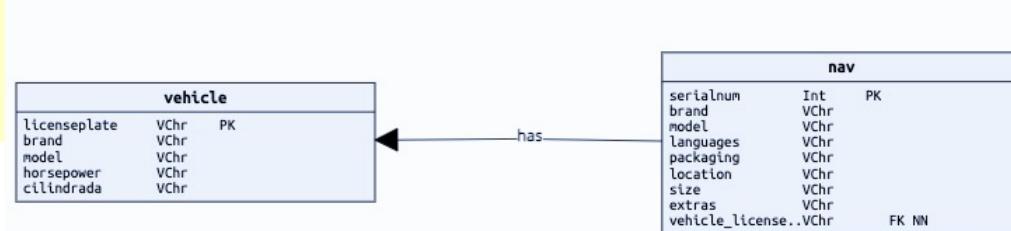
1:1 mand:opt (SIMPLIFIED OR NORMAL)



Regra 1.2

Relacionamento binário de grau 1:1 e participação obrigatória de apenas uma das entidades

- São necessárias duas tabelas, uma para cada entidade.
- 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 na tabela correspondente à entidade cuja participação é obrigatória (é chave estrangeira).





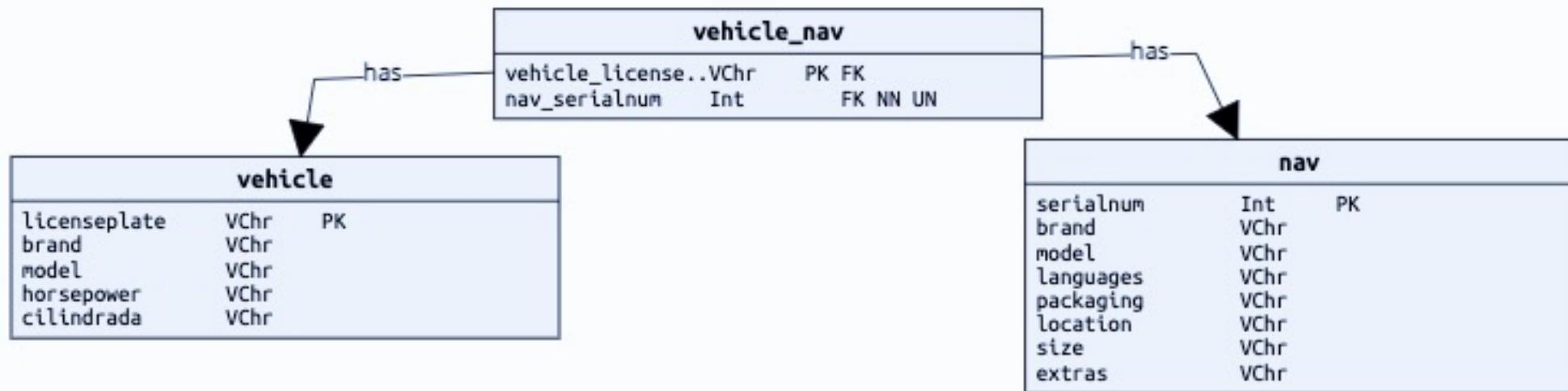
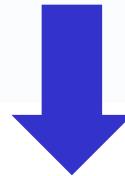
Normal

PostgreSQL 9.6.1

1:1 opt-opt (NORMAL)

Vehicle		
licensePlate	VChr	PK
brand	VChr	
model	VChr	
horsepower	VChr	
cilindrada	VChr	

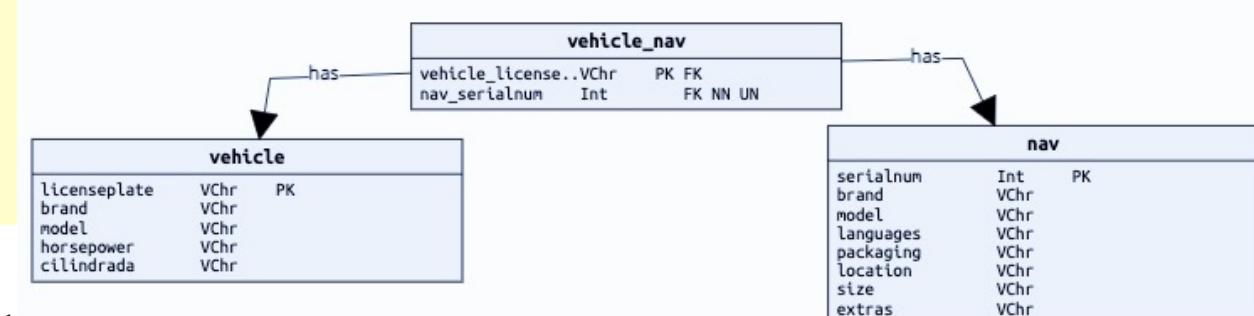
Nav		
serialNum	Int	PK
Brand	VChr	
Model	VChr	
Languages	VChr	
packaging	VChr	
location	VChr	
size	VChr	
extras	VChr	



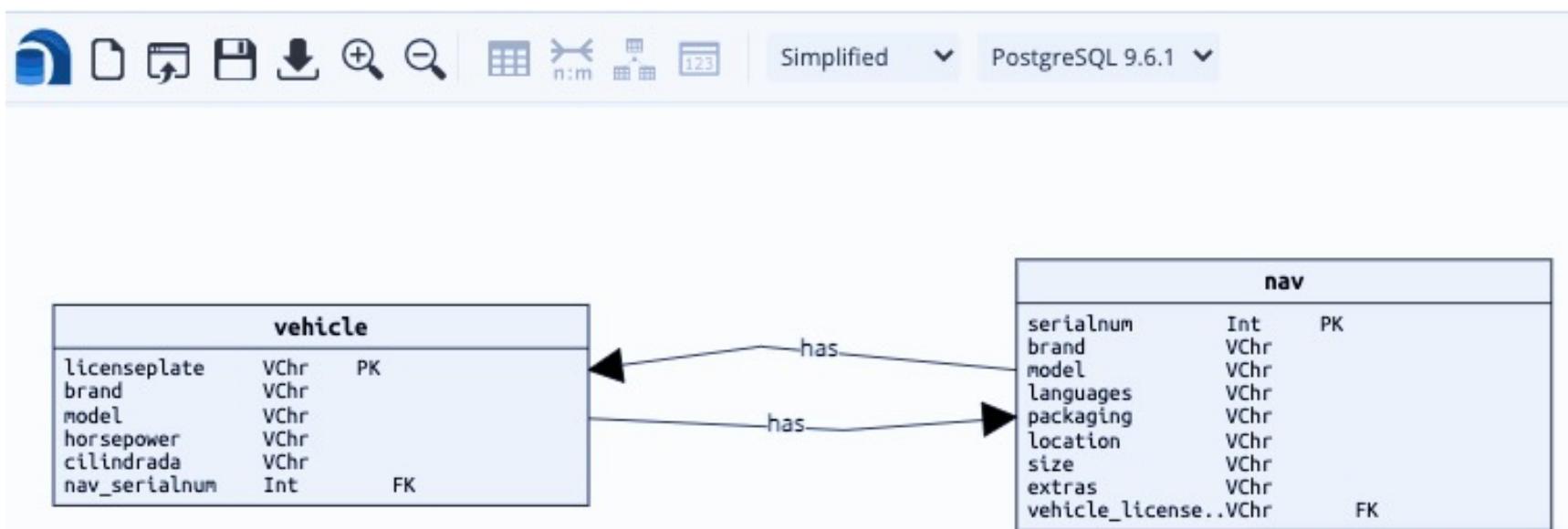
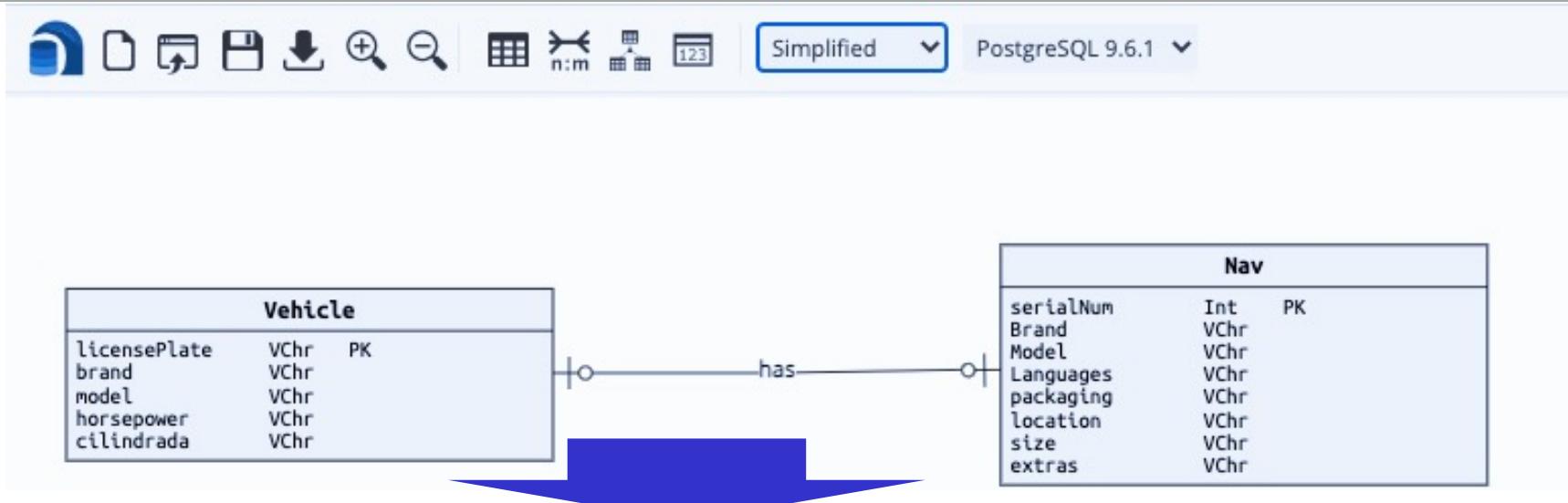
Regra 1.3

Relacionamento binário de grau 1:1 e participação não obrigatória de ambas entidades

- 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 correspondente ao relacionamento terá entre os seus atributos as chaves primárias das duas entidades (como chaves estrangeiras).
- Qualquer dos atributos da tabela do relacionamento pode ser a chave primária da tabela.



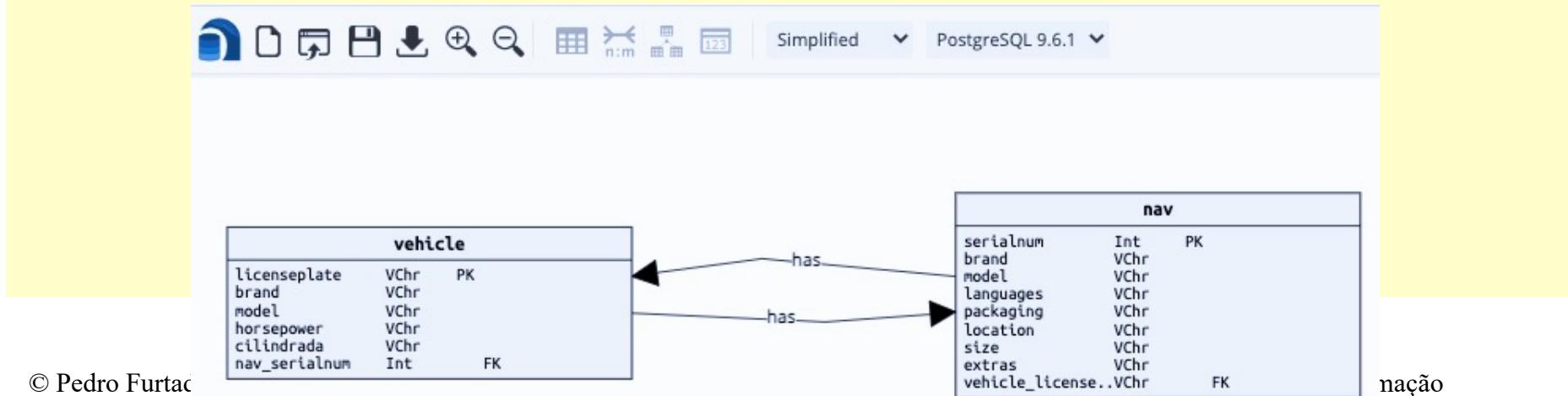
1:1 opt-opt (SIMPLIFIED)



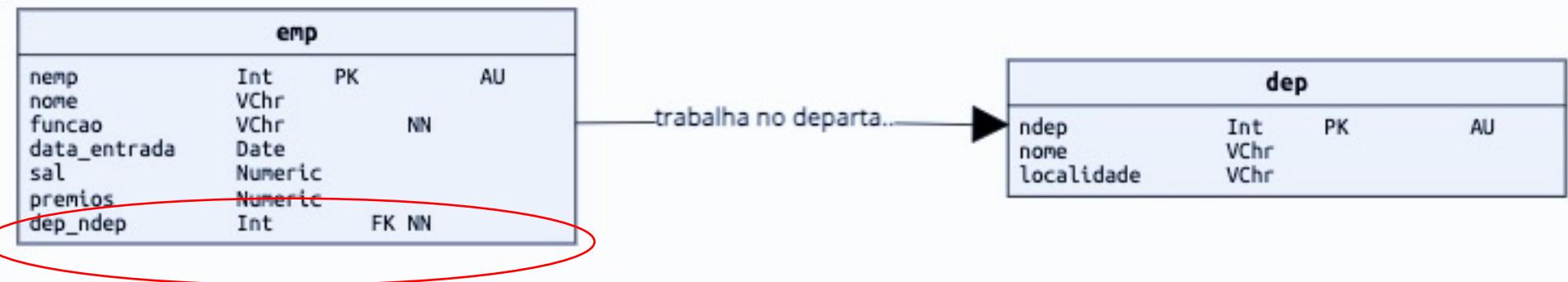
Regra 1.3 simplified

Relacionamento binário de grau 1:1 e participação não obrigatória de ambas entidades

- São necessárias duas tabelas, uma para cada entidade.
- A chave primária de cada entidade serve de chave primária na tabela correspondente.
- Cada uma das duas tabelas resultantes terá uma chave estrangeira que serão as chaves primárias da outra tabela



1:N mand from side N



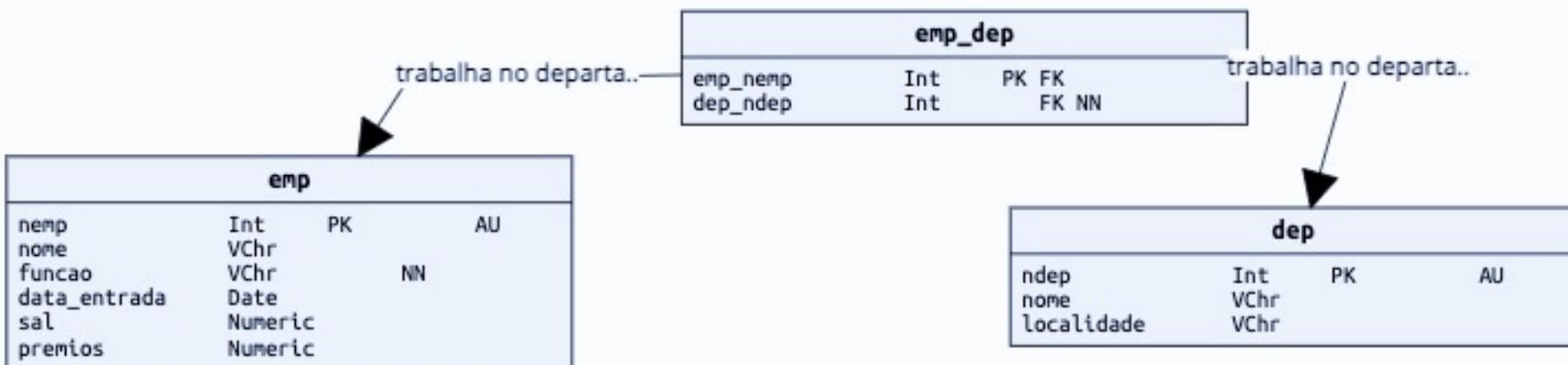
Regra 2

Relacionamento binário de grau 1:N e participação obrigatória do lado N

- São necessárias duas tabelas, uma para cada entidade.
- 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 (chave estrangeira) na tabela correspondente à entidade do lado N.



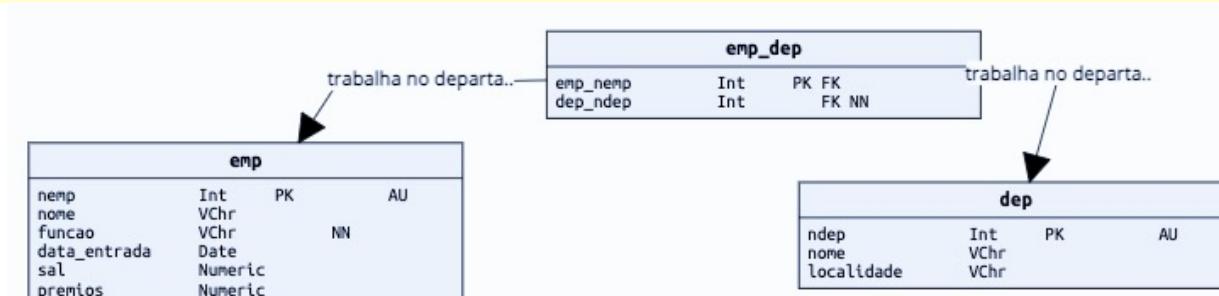
1:N opt from side N



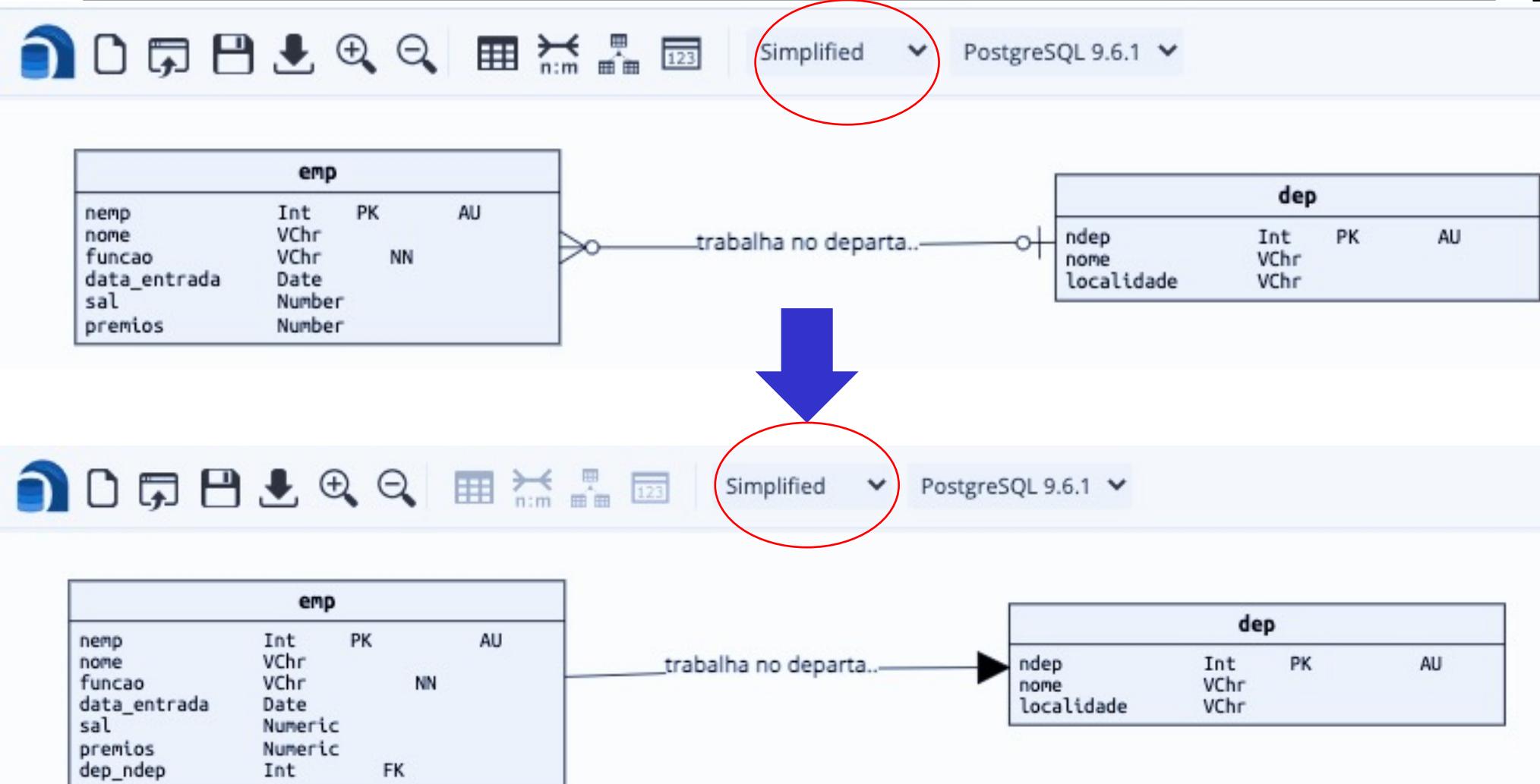
Regra 2.1

Relacionamento binário de grau 1:N e participação não obrigatória do lado N

- São 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 cada entidade).
- A tabela relativa ao relacionamento terá de ter entre os seus atributos as chaves primárias de cada uma das entidades (como chaves estrangeiras). A chave primária da tabela correspondente ao relacionamento deve ser o atributo que é chave na tabela do lado N.



1:N opt from side N, SIMPLIFIED

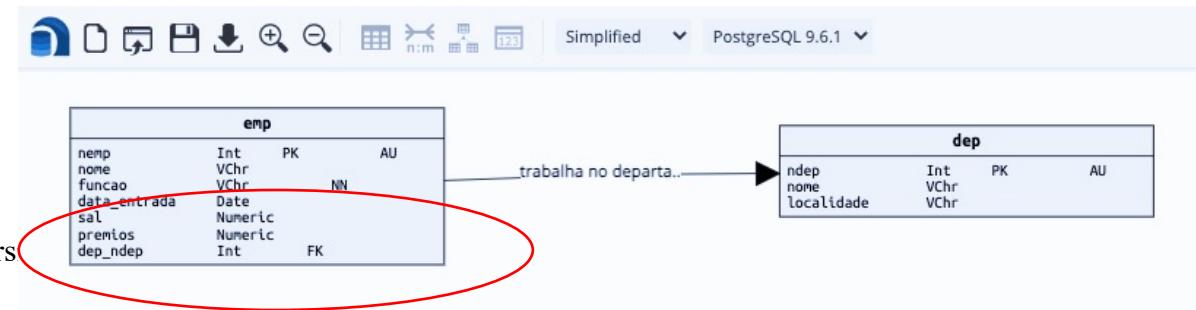


Regra 2.2 (simplified)

Relacionamento binário de grau 1:N e participação obrigatória do lado N

- São necessárias duas tabelas, uma para cada entidade.
- 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 (chave estrangeira) na tabela correspondente à entidade do lado N.

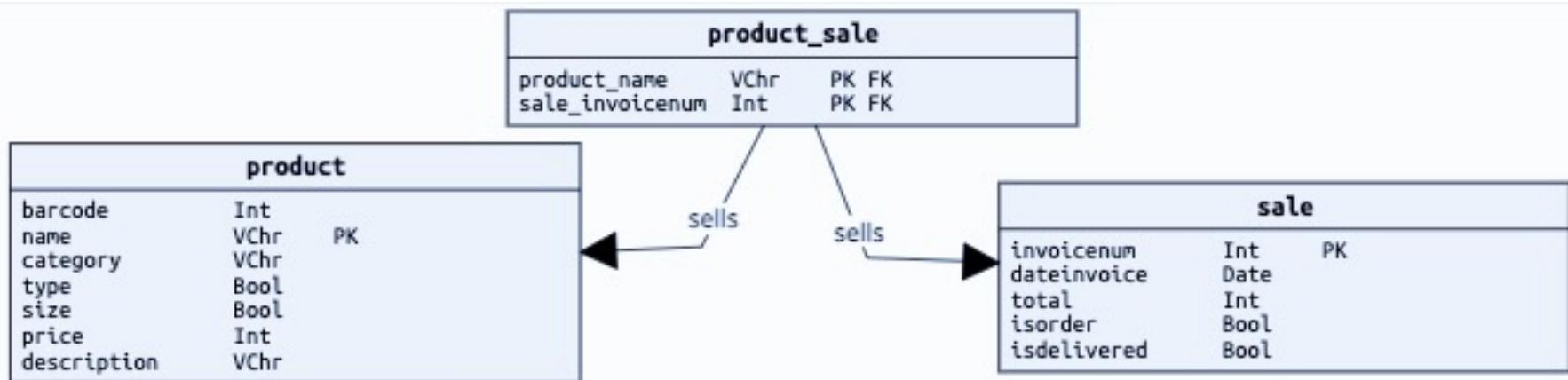
UC



N:M



UC

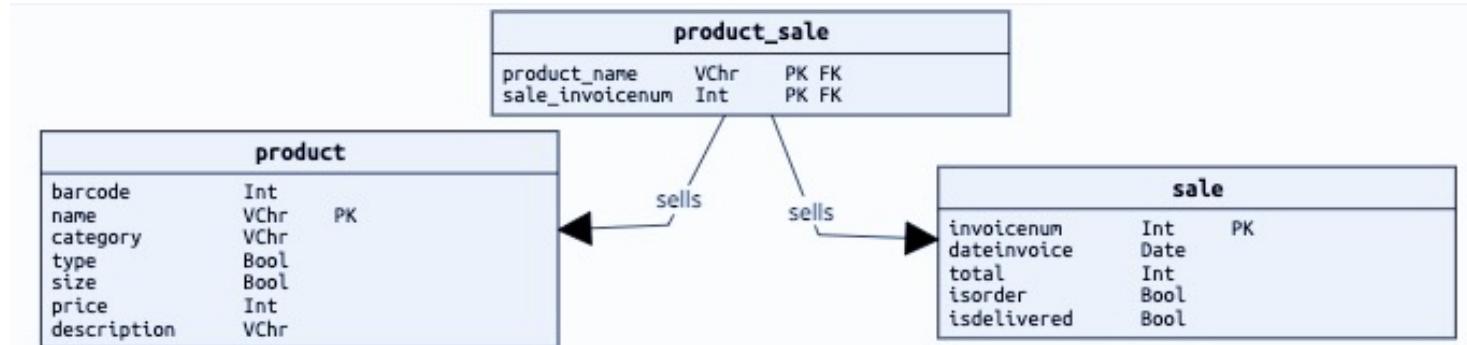


bdai

Regra 3

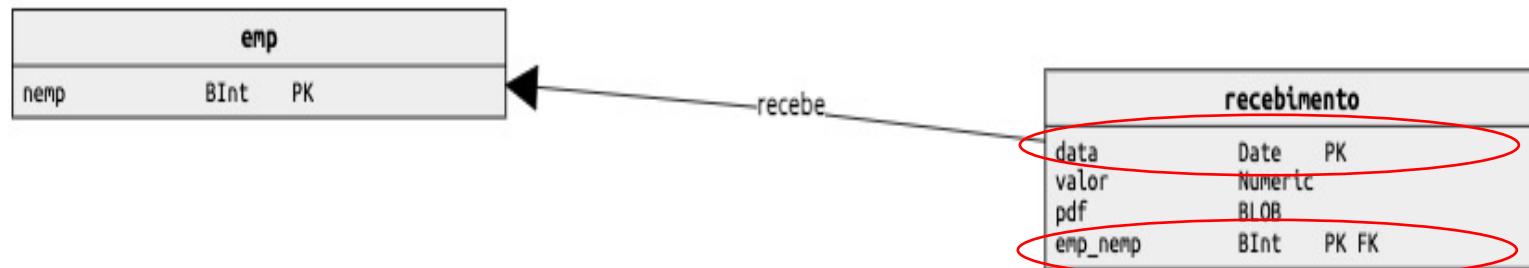
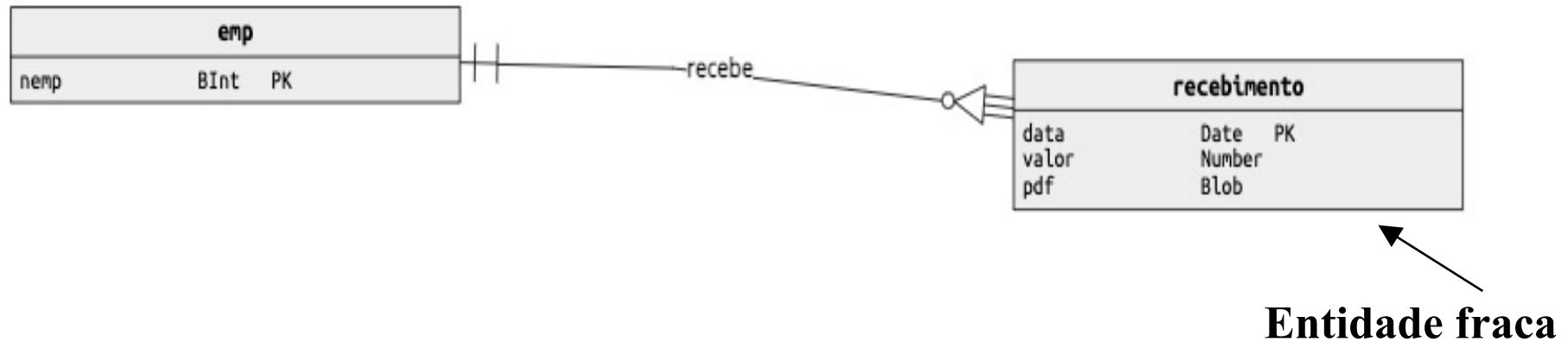
Relacionamento binário de grau M:N

- 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 as chaves primárias de cada uma das entidades (que são chaves estrangeiras, estabelecendo a ligação com cada uma das outras duas tabelas). A chave primária da tabela relativa ao relacionamento é composta pelos dois atributos (chaves primárias de cada uma das entidades)



Entidades fracas

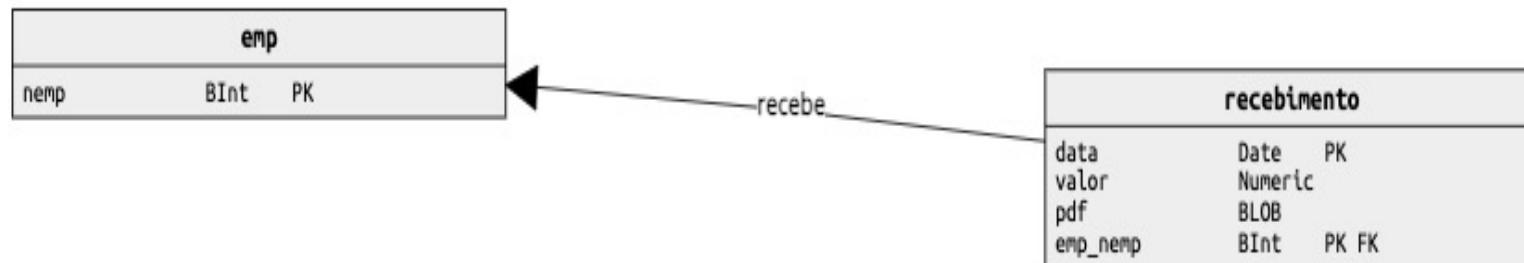
As entidades fracas não têm, naturalmente, atributos que possam constituir chaves candidatas e a identificação de instâncias específicas destas entidades depende sempre de outra entidade, designada por entidade identificadora.



Tabelas para entidades fracas: regra 4

Entidades Fracas

- Para cada entidade fraca EF é criada uma tabela contendo todos os atributos de EF.
- Esta tabela deve incluir também, como chave estrangeira, o atributo que é chave primária da entidade identificadora.
- A chave primária da tabela da EF é a combinação de da chave parcial da EF e do atributo que é chave primária da entidade identificadora.

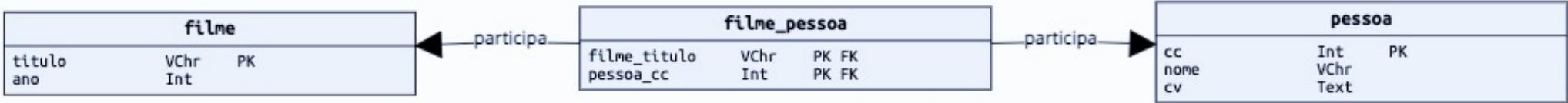


Como adicionar atributo ou
relacionamento a relacionamento M:N?

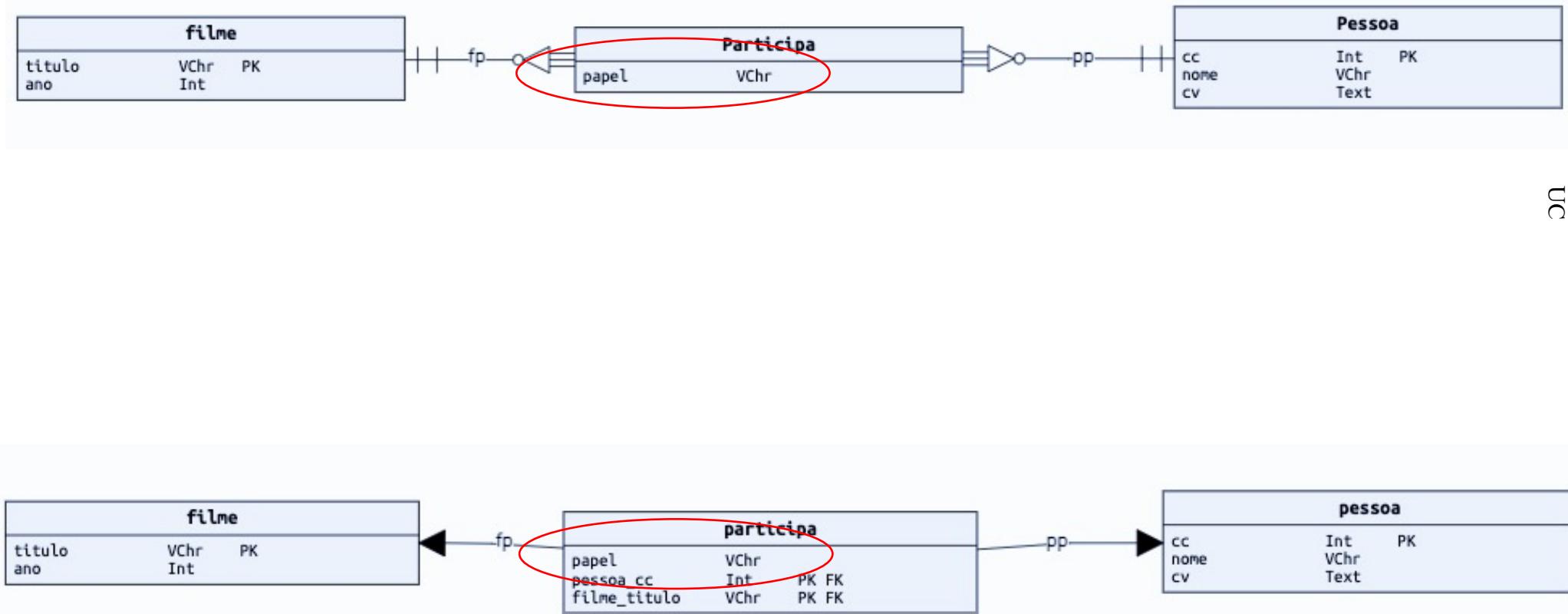
m:n... Mas como adicionar papel?



UC



bdai



Relacionamentos podem ser transformados em entidades para attrs ou rels



- Um médico prescreve a um doente
- O médico faz varias prescrições
- Pode fazer mais do que uma numa data
- Um paciente pode ter várias prescrições
- Pode ter prescrições do mesmo médico em varias datas
- Prescreve um ou mais medicamentos
- **ONDE FICAM OS MEDICAMENTOS????**

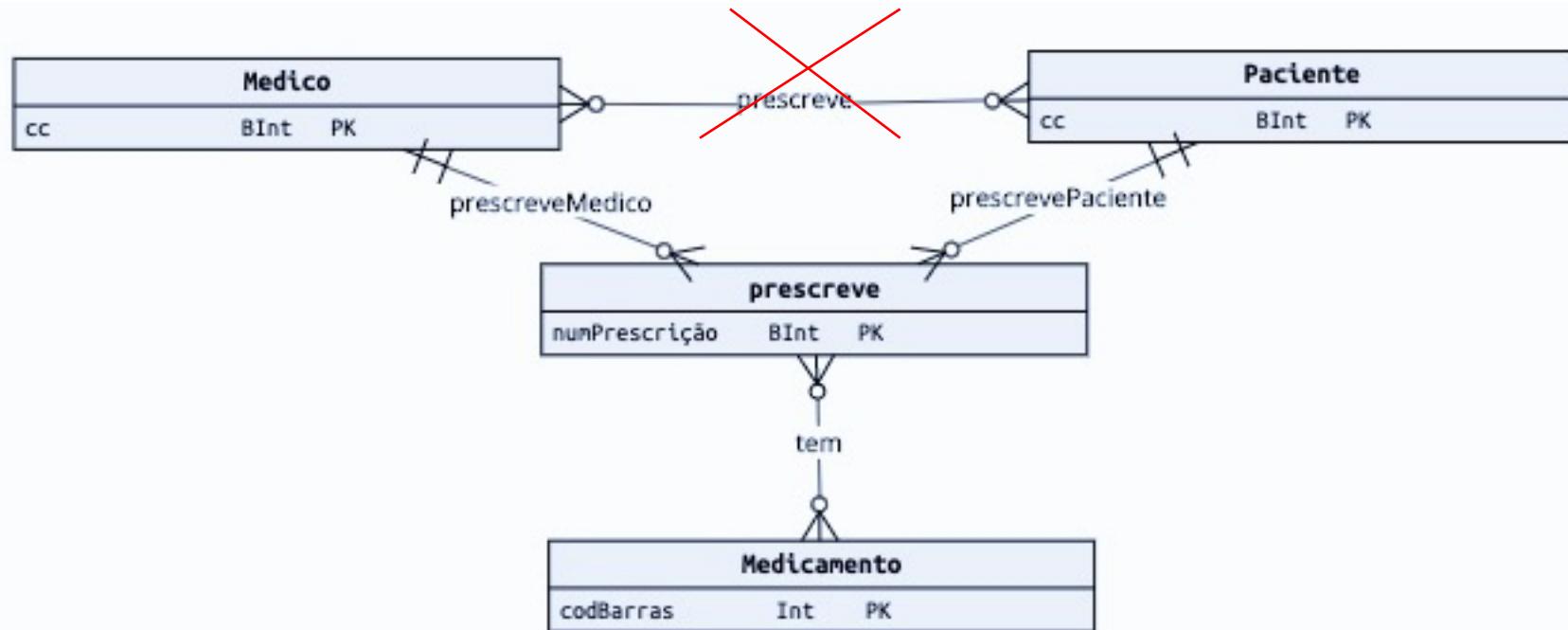
Relacionamentos podem ser transformados em entidades para attrs ou rels

© Pedro Furtado

- Um médico prescreve a um doente



UC

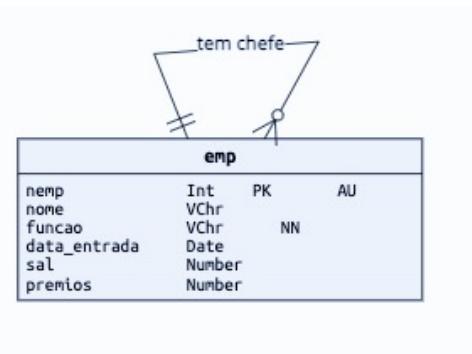


bdaí

Relacionamentos recursivos

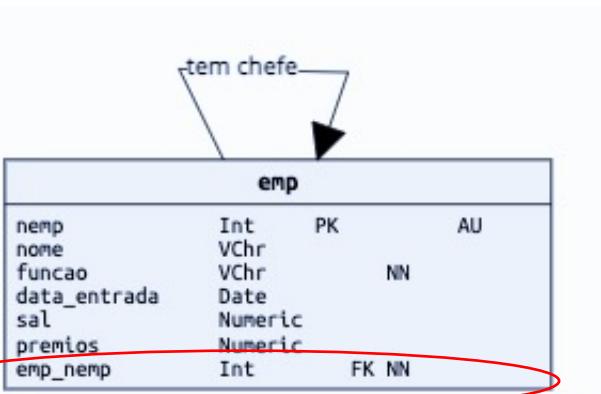
É como se fossem duas entidades diferentes, aplicam-se as mesmas regras

Diagrama ER



Um determinado empregado pode chefiar outros empregados.

Diagrama físico



O esquema físico corresponde à mesma tabela à qual é acrescentada uma chave estrangeira

Especialização e herança

Especialização:

Processo de definição de **sub-entidades** de uma entidade. Essa entidade, a partir da qual se definem sub-entidades, designa-se por **super-entidade**.

UC

Há um relacionamento entre uma super-entidade e as suas sub-entidades

As sub-entidades herdam os atributos e os relacionamentos da super-entidade.

bdai

- Herança genérico
- Representação da herança no onda

Geração de tabelas para super-entidades e sub-entidades

© Pedro Furtado

UC

bdaI

- Não há uma regra fixa: há um conjunto de alternativas;
- Cabe ao projectista escolher, para cada caso, a alternativa que produz o conjunto de tabelas mais adequado.

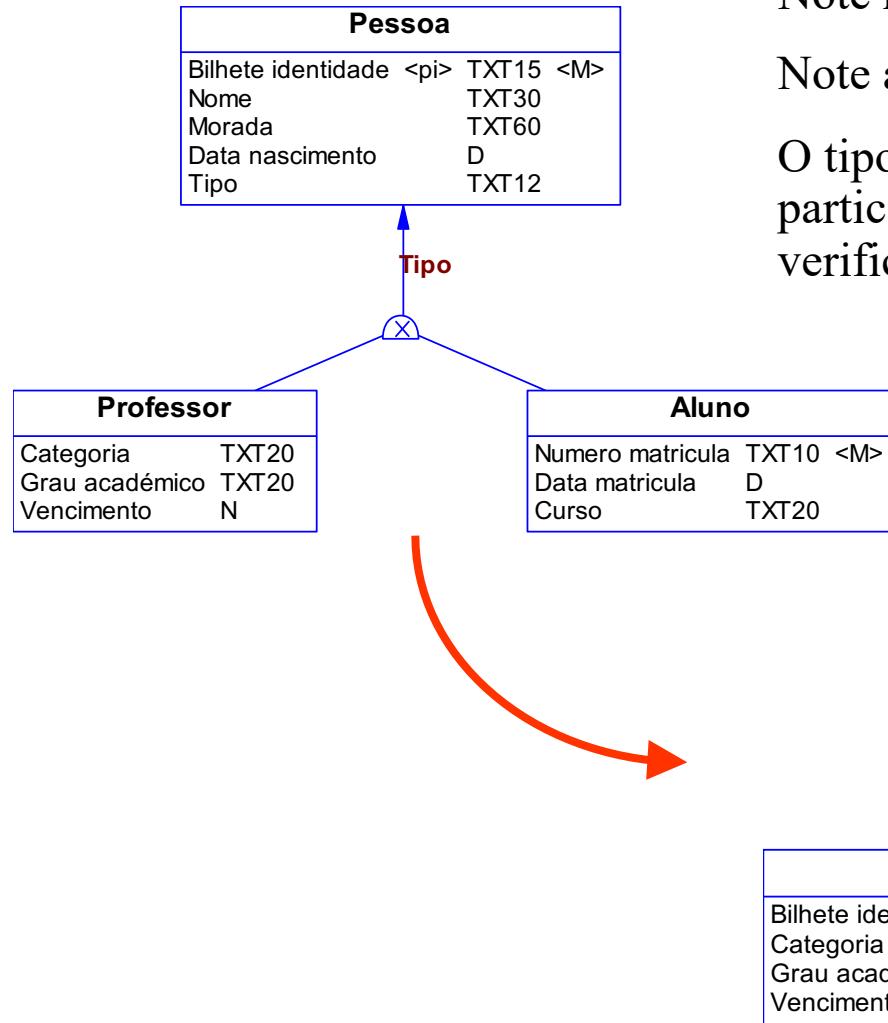
Geração de tabelas para super-entidades e sub-entidades: alternativa A

Alternativa A

- Uma tabela para a super-entidade com os atributos específicos da super-entidade.
- Uma tabela para cada sub-entidade com os atributos específicos de cada sub-entidade.
- A chave primária da super-entidade aparece nas tabelas das sub-entidades, onde é simultaneamente chave estrangeira e chave primária

Usa-se geralmente quando o número de atributos específicos de cada sub-entidade é grande, tendo também a super-entidade bastantes atributos, embora a decisão também se prenda com relacionamentos de super/sub entidades. Pode usar-se qualquer que seja o tipo de relacionamento (mutuamente exclusivo ou não) e de participação obrigatória ou não.

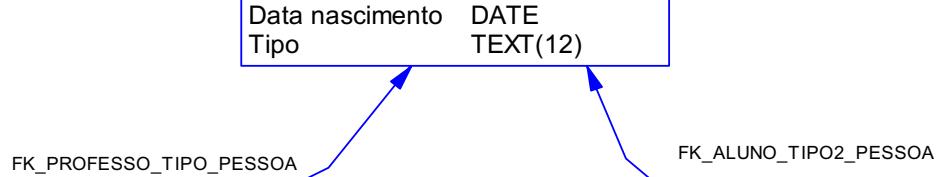
Exemplo da alternativa A



Note nas chaves primárias no ER e físico

Note as chaves estrangeiras no físico

O tipo de relacionamento (mutuamente exclusivo ou não) e a participação (obrigatória ou não) só são usados nas verificações de integridade.

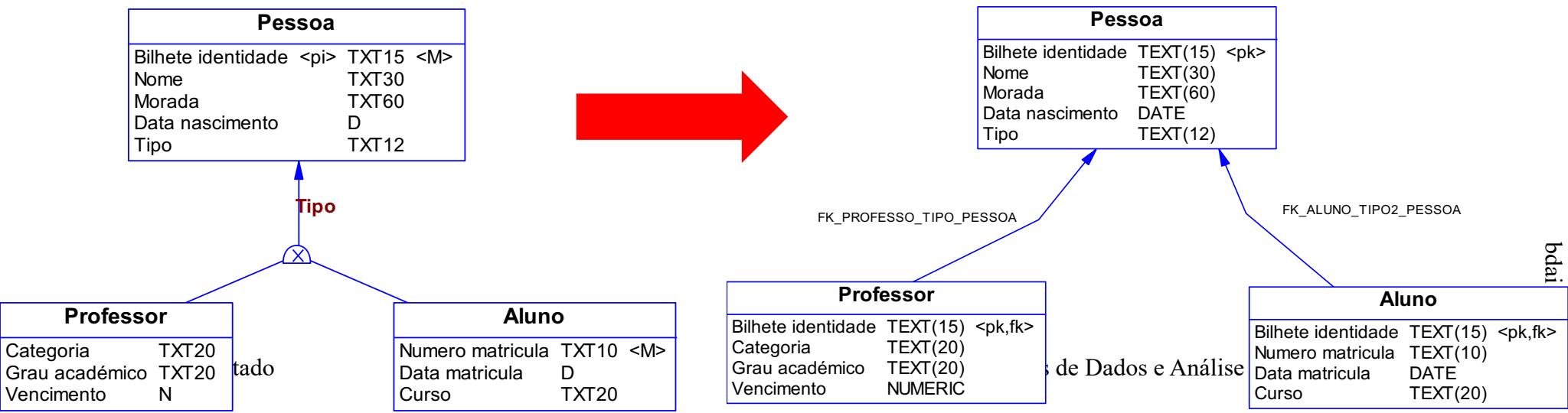


Professor	
Bilhete identidade	TEXT(15) <pk,fk>
Categoria	TEXT(20)
Grau académico	TEXT(20)
Vencimento	NUMERIC

Aluno	
Bilhete identidade	TEXT(15) <pk,fk>
Numero matricula	TEXT(10)
Data matricula	DATE
Curso	TEXT(20)

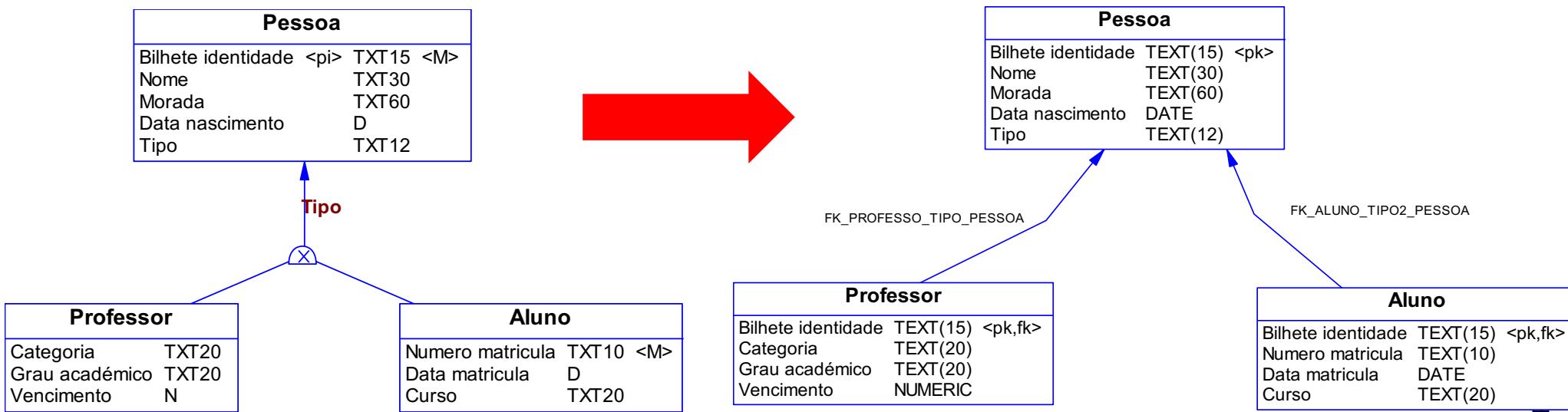
Algumas regras para inserções e apagamentos

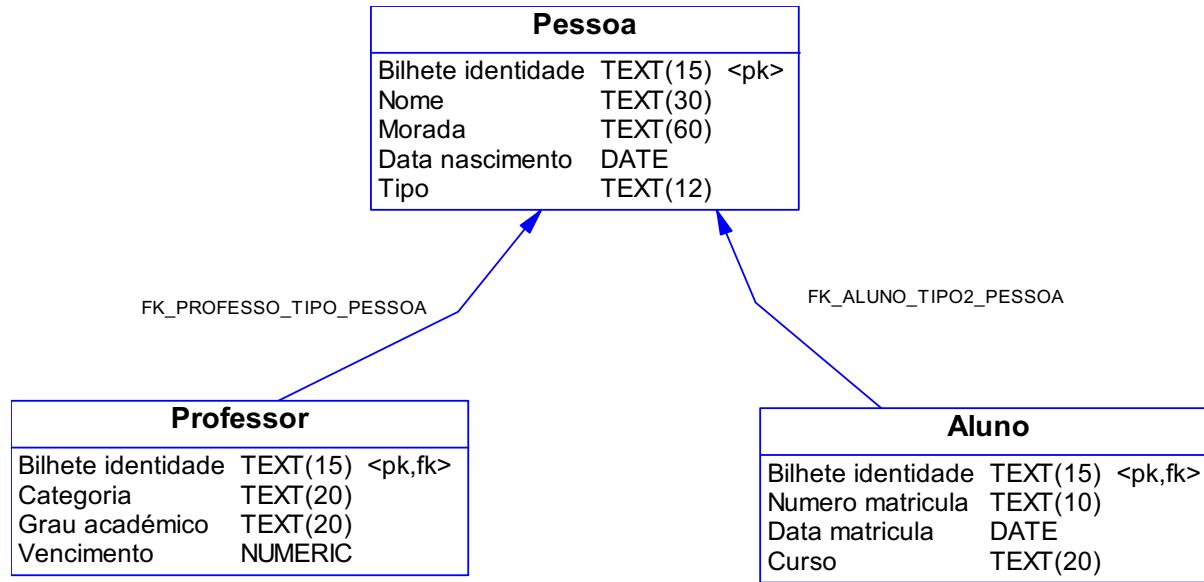
- Uma pessoa específica passa a ser representada na tabela pessoa, mas também numa das tabelas subentidade, caso a pessoa tenha de ser um dos casos
 - Logo, inserções de novas pessoas passam a inserir em mais do que uma tabela
 - Apagar uma ocorrência da super-entidade obriga a apagá-la automaticamente de todas as sub-entidades a que a ocorrência pertence.
- O atributo tipo é muito importante, deve inclui-lo sempre e também os valores possíveis,
 - como restrição (check(tipo=='Professor' or tipo=='aluno'))
 - ou pelo menos como texto



Algumas regras para inserções e apagamentos

- Pode definir-se que a pessoa possa ser de outro tipo que não inclui sub-entidade (não terá atributos específicos).
- Pode haver exclusividade ou não. Essa e outras restrições têm de ser escritas explicitamente e verificadas no código de aplicações





INSERÇÃO DE PROFESSOR:

Insert into Pessoa values(10,'Pedro Manuel','Rua da Alegria','2012-12-12','Professor');
 Insert into Professor values(10,'Prof. Auxiliar','PhD',1000);

ALTERAÇÃO DE NOME E VENCIMENTO DE PROFESSOR:

Update Pessoa set Nome='Pedro Maria Manuel' where "Bilhete identidade"=10;
 Update Professor set Vencimento = 1500 where "Bilhete identidade"=10;

APAGAMENTO DE PROFESSOR:

Delete from Professor where "Bilhete identidade"=10;
 Delete from Pessoa where "Bilhete identidade"=10;

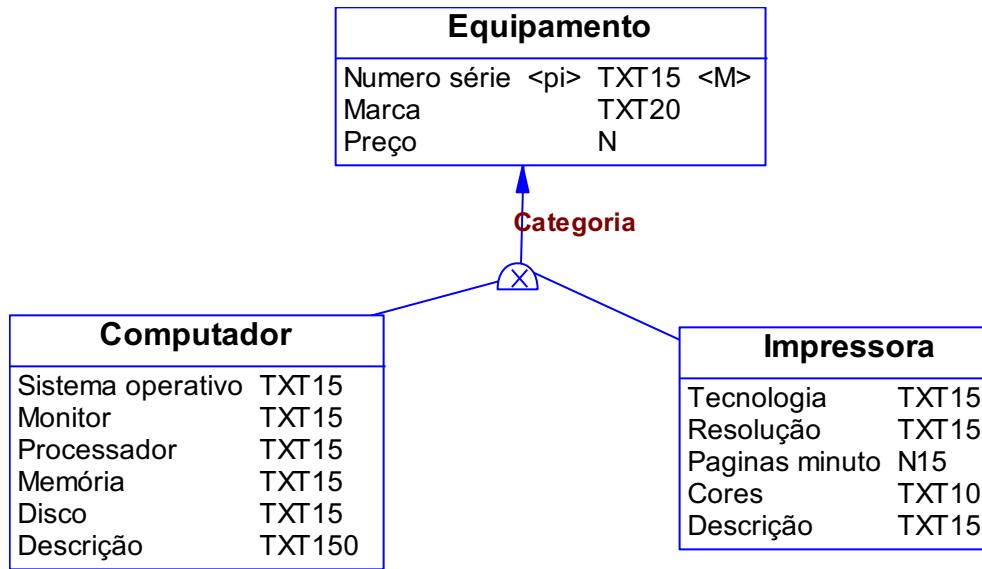
Geração de tabelas para super-entidades e sub-entidades: alternativa B

Alternativa B

- Uma tabela para cada sub-entidade (e não há tabela para a super-entidade).
- Cada tabela contém todos os atributos da super-entidade e os atributos específicos da sua sub-entidade.
- A chave primária de cada tabela é a chave primária da super-entidade .

Só se deve usar em relacionamentos mutuamente exclusivos com participação obrigatória. Uma possível heurística é que esta regra deve usar-se (em vez da alternativa A) quando o número de atributos da super-entidade é pequeno, embora a decisão também se prenda com relacionamentos de super/sub entidades.

Exemplo da alternativa B



O conjunto de todas as ocorrências (registos) da super-entidade Equipamento pode obter-se pela união das duas sub-entidades que se constituem em tabelas.

Computador		
Número série	TEXT(15)	<pk>
Sistema operativo	TEXT(15)	
Monitor	TEXT(15)	
Processador	TEXT(15)	
Memória	TEXT(15)	
Disco	TEXT(15)	
Descrição	TEXT(150)	
Marca	TEXT(20)	
Preço	NUMERIC	

Impressora		
Número série	TEXT(15)	<pk>
Tecnologia	TEXT(15)	
Resolução	TEXT(15)	
Páginas minuto	N15	
Cores	TEXT(10)	
Descrição	TEXT(150)	
Marca	TEXT(20)	
Preço	NUMERIC	

Geração de tabelas para super-entidades e sub-entidades: **alternativa C**

Alternativa C

- Uma única tabela para a super-entidade e para todas as sub-entidades.
- A chave primária da tabela é a chave primária da super-entidade .

Nesta alternativa a tabela contém campos nulos (não está normalizada).

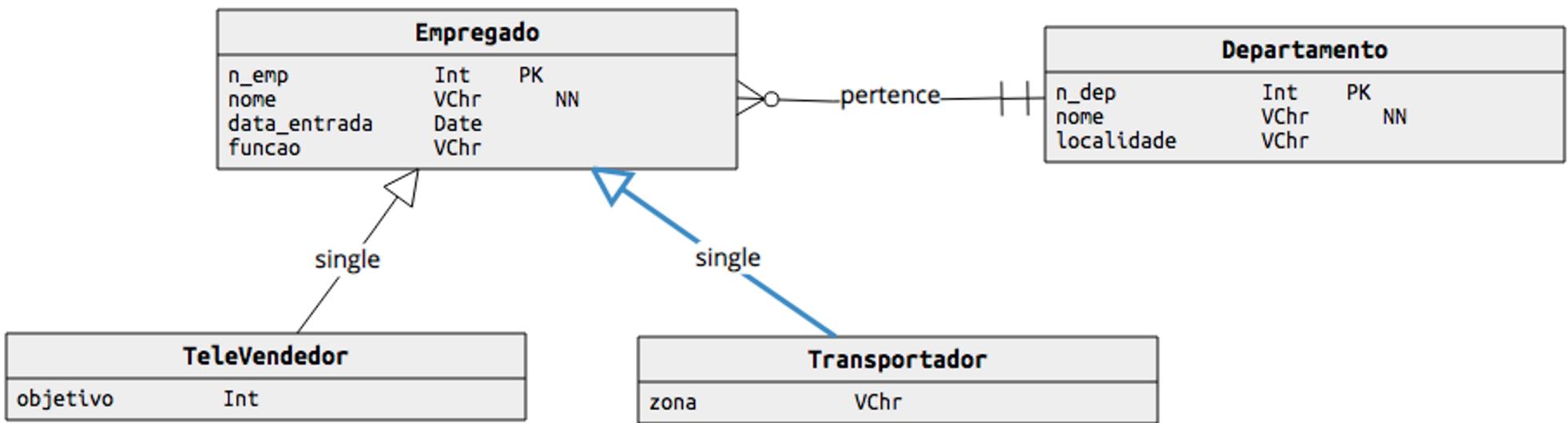
Praticamente só se usa quando cada subclasse tem poucos atributos específicos (um ou dois) e não participa outros relacionamentos com outras entidades

Herança no onda

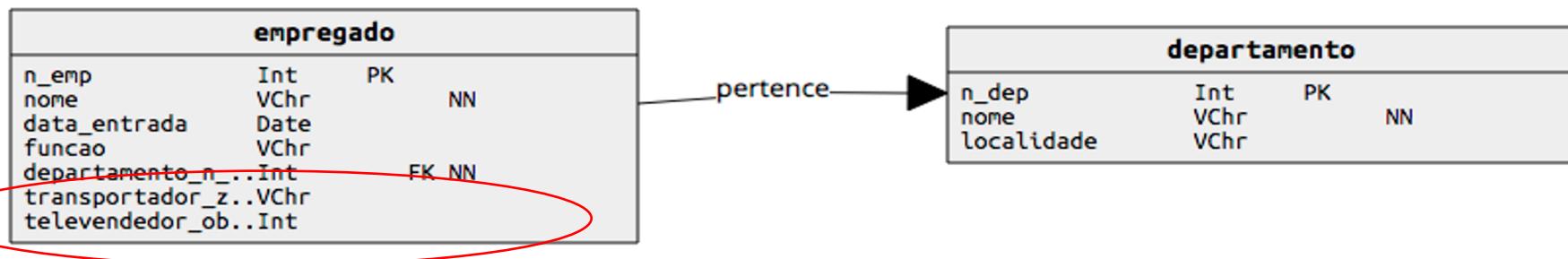
single, concrete e complete

Herança no onda

- Não esquecer que as três possibilidades são válidas (single, concrete, complete)
- Frequentemente a alternativa complete e single são as mais adequadas, mas não exclui a concrete

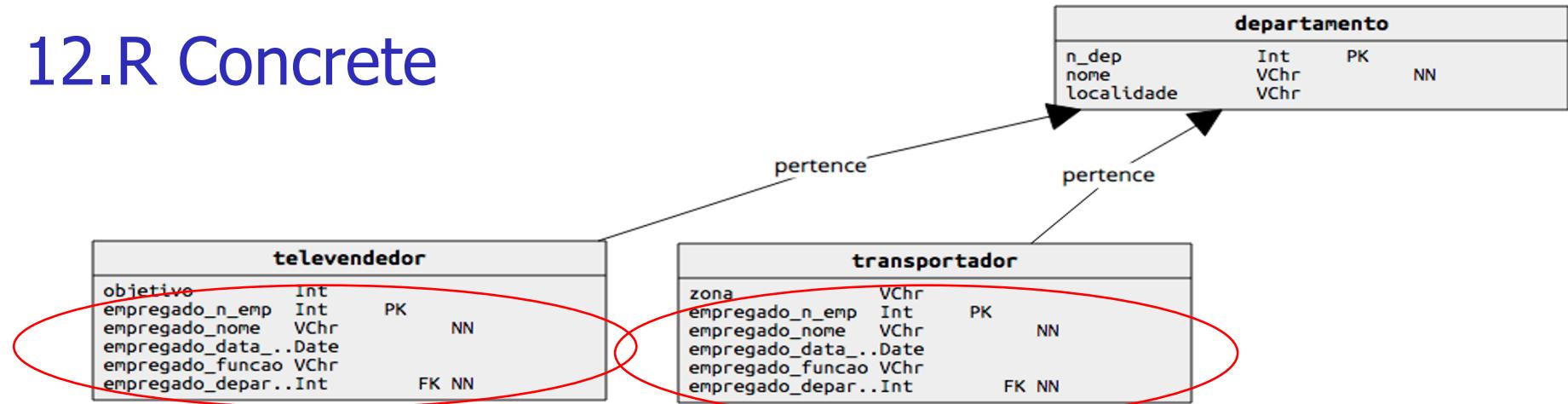


12.R Single



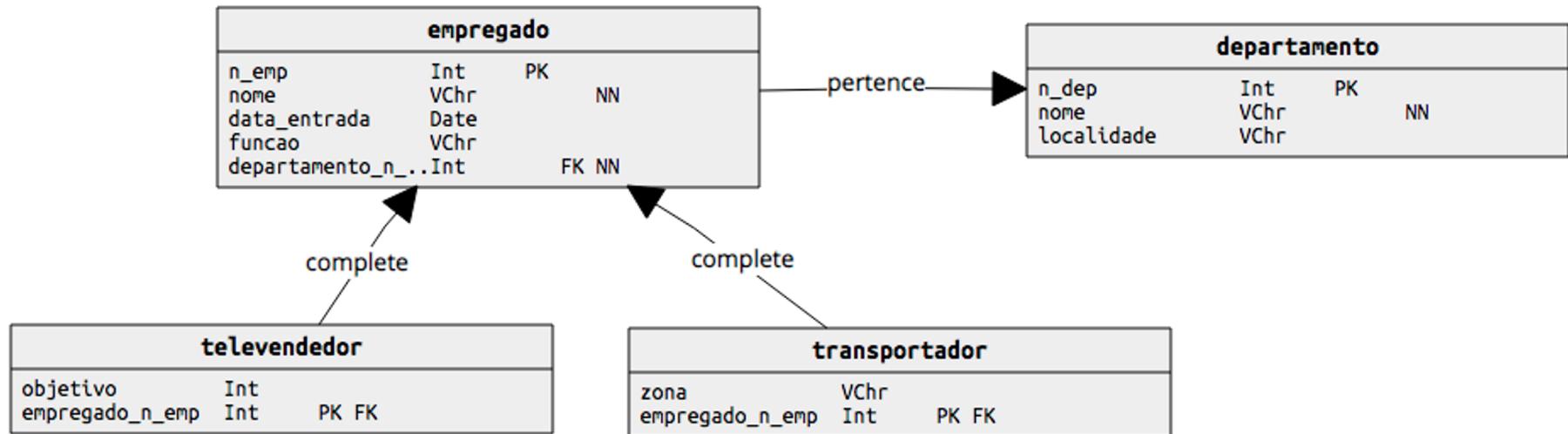
- Resumindo: quaisquer sub-entidades são representadas no mesmo pai
- Todos os attrs de sub-entidade que não a correcta ficarão a null
- Relacionamentos de sub-entidades também passam automaticamente a relacionar-se com a entidade/tabela pai, simplesmente porque esta representa todas as sub-entidades
- (+) simples (1 tabela)
- (-) nulls e menos explicitação, logo restrições implícitas

12.R Concrete



- Resumindo: não tenho pai, só filhos
- Todos os attrs de pai repetidos em cada filho
- Relacionamentos de pai passam para cada um dos filhos
- (+) muito claro e explícito quem é o quê
- **(--) muitas pesquisas passam a ser duplicadas, porque agora tenho de pesquisar nos dois filhos separadamente**
- **(--) pode aumentar mto complexidade, ja que relacionamentos com o pai sao multiplicados (passam) para cada filho**

12.R Complete

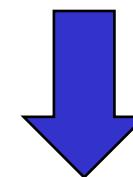
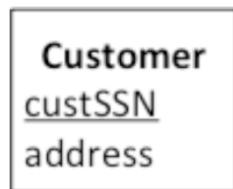
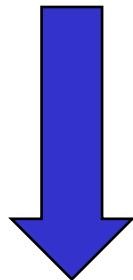
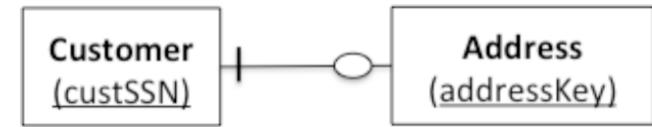
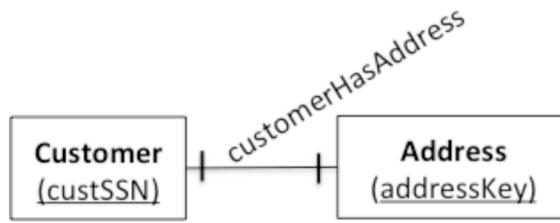


- Resumindo: tenho pai e filhos
- (+) clareza(1 tabela por cada filho, uma do pai) , restrições explicitas
- (+) clareza, simplicidade e menos tabelas e confusão tb relativa a relacionamentos:rels de todos com o pai
- (-) necessidade de inserir/apagar em pai e filho simultaneamente, porque uma instancia agora é representada por ambos (pai e filho)

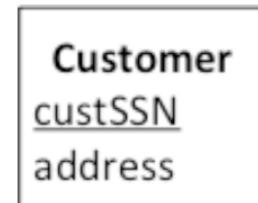
Extra Project-tasks

Algumas partes ...

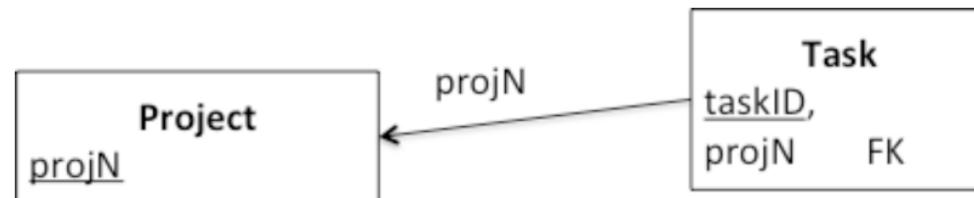
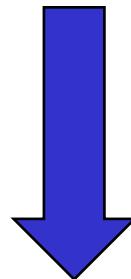
1:1



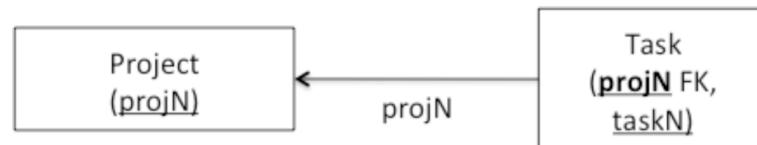
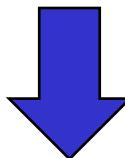
ou



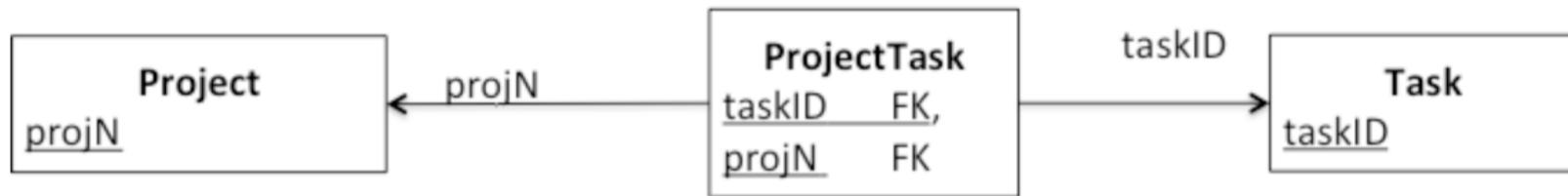
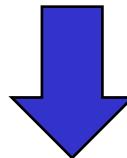
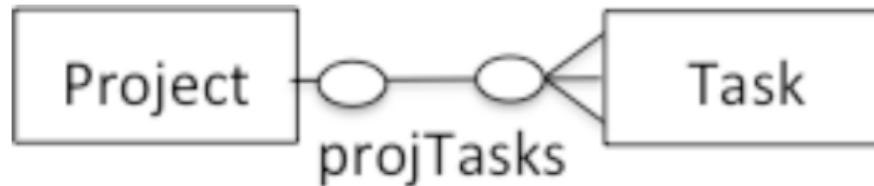
1:N



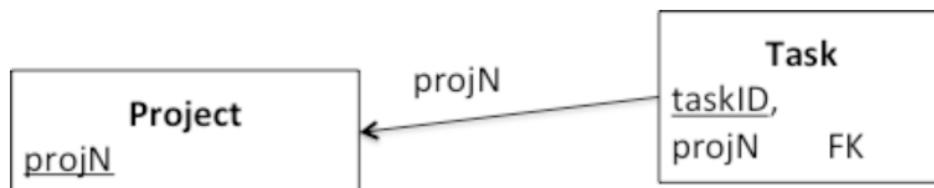
Project Tasks Weak



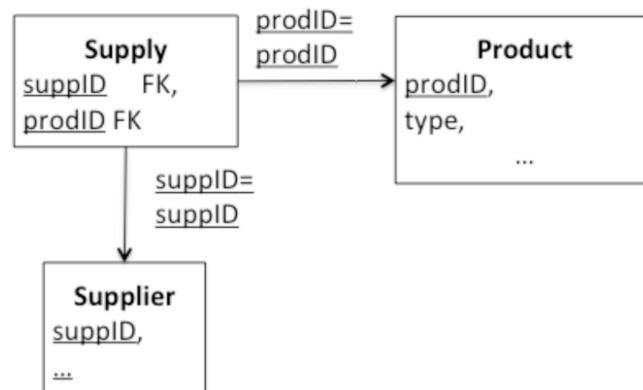
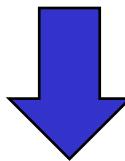
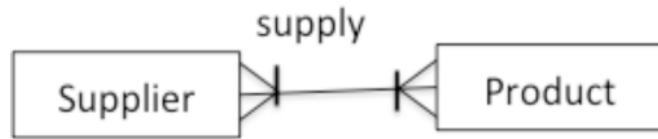
1:N (b)



ou

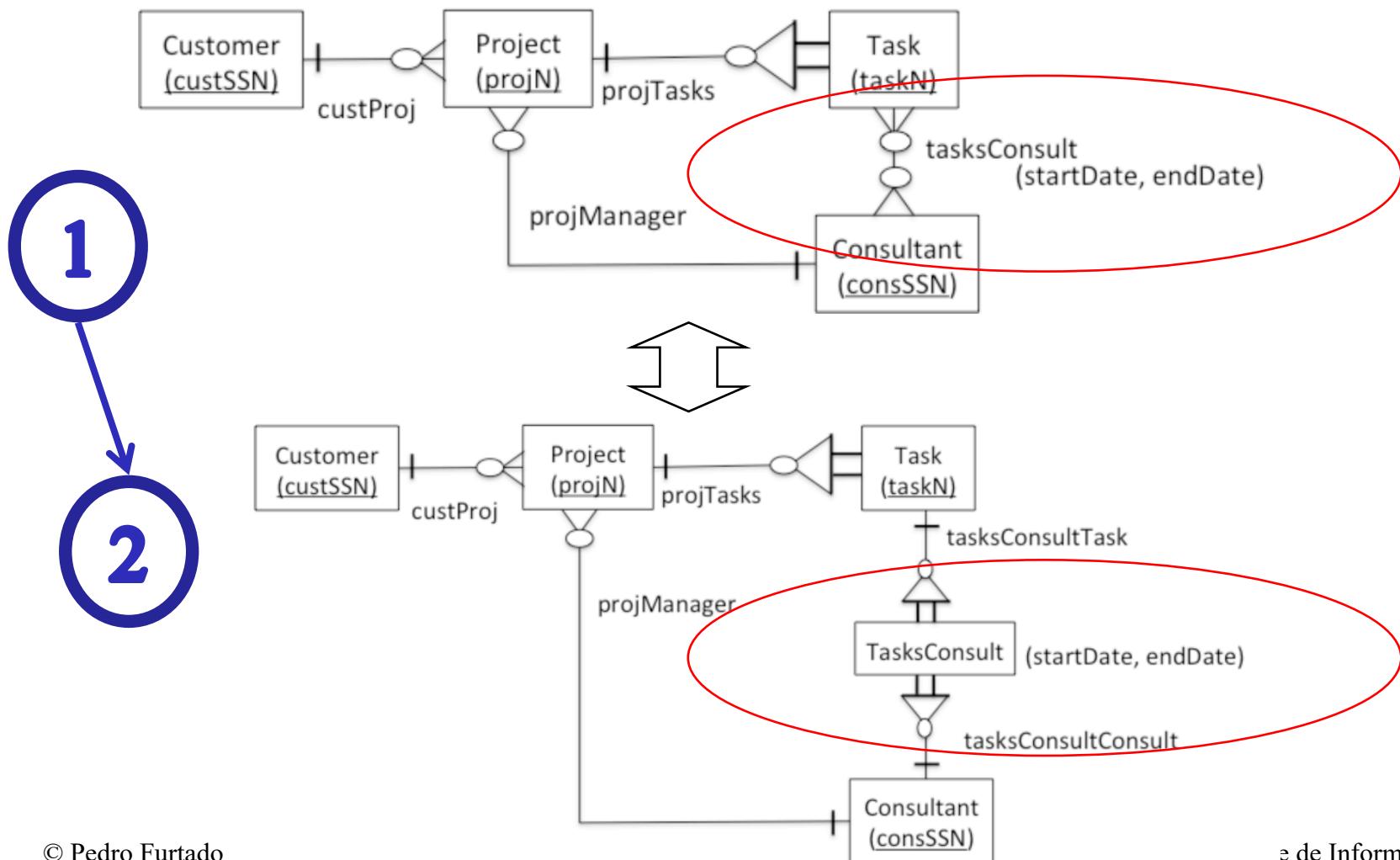


N:M



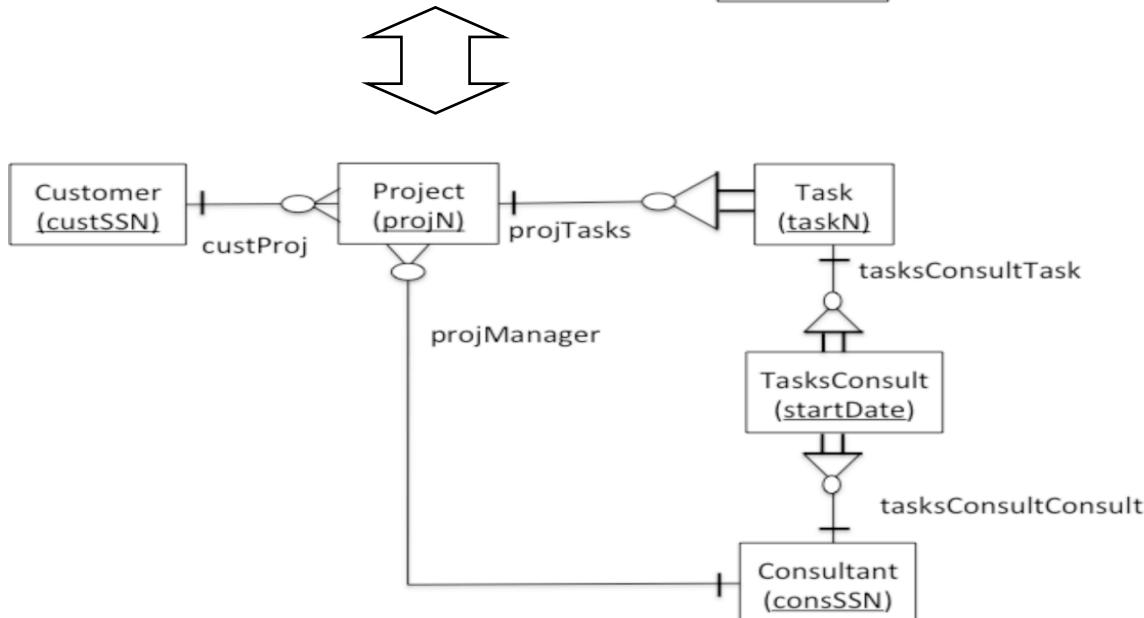
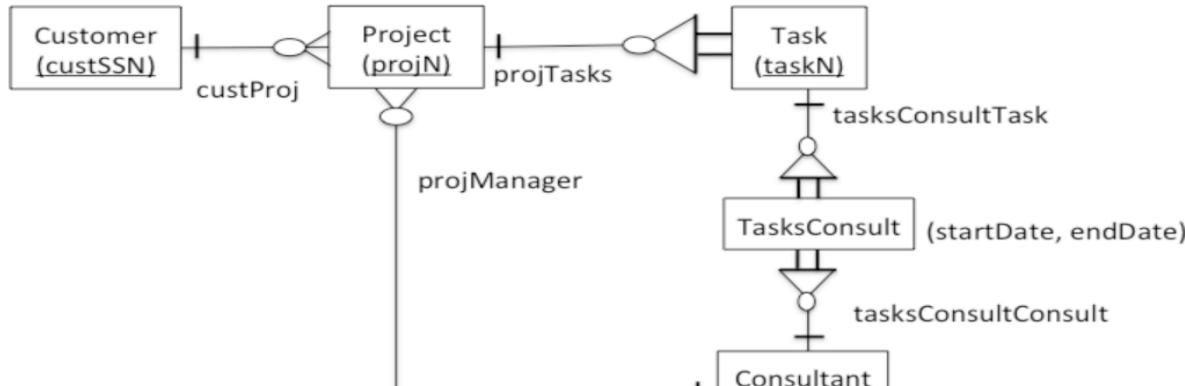
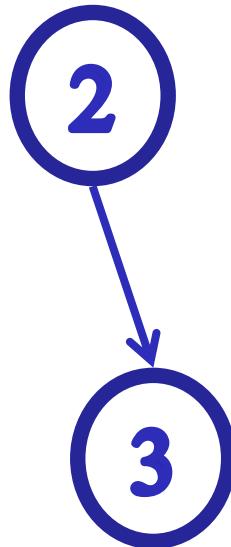
M:N com atributos

Datas de inicio e fim de participação de consultor em tarefa

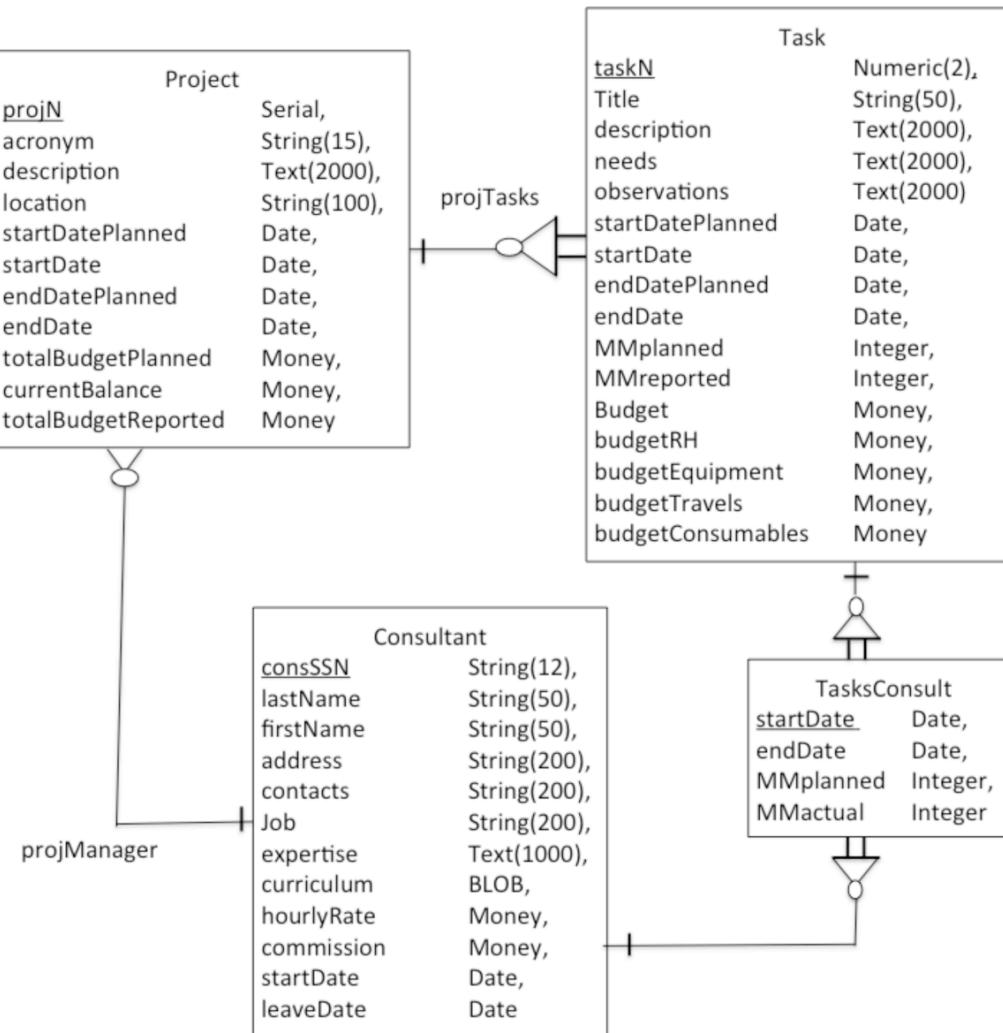


M:N com atributos

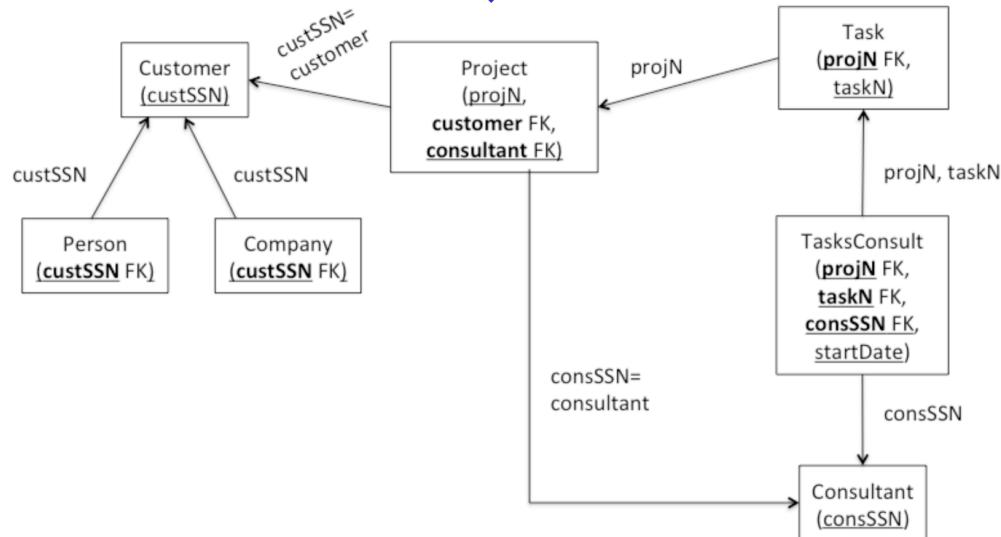
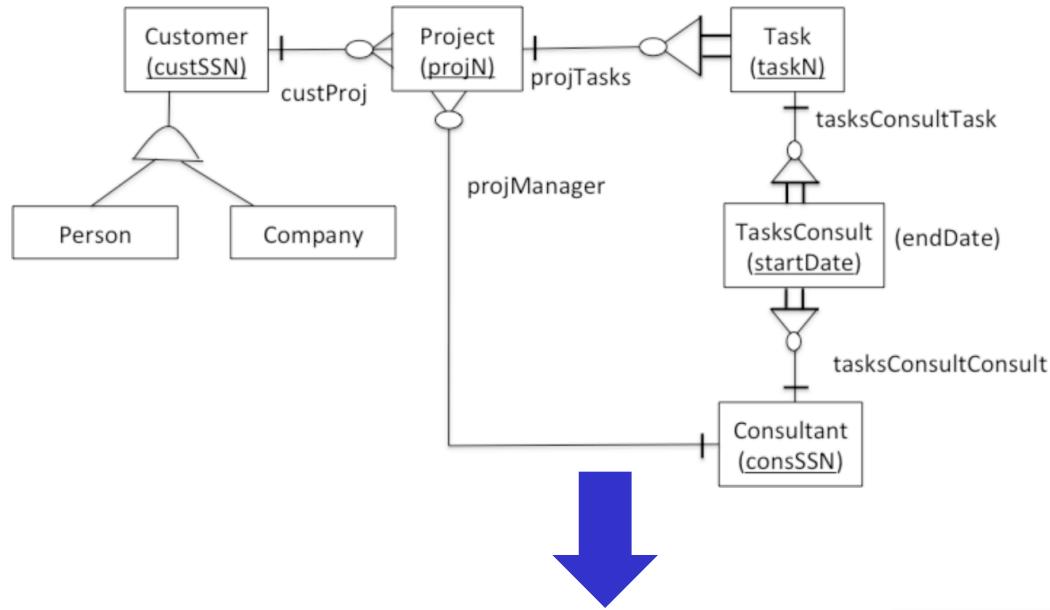
Identificador de TaskConsultant



Atributos – project, task, consultants

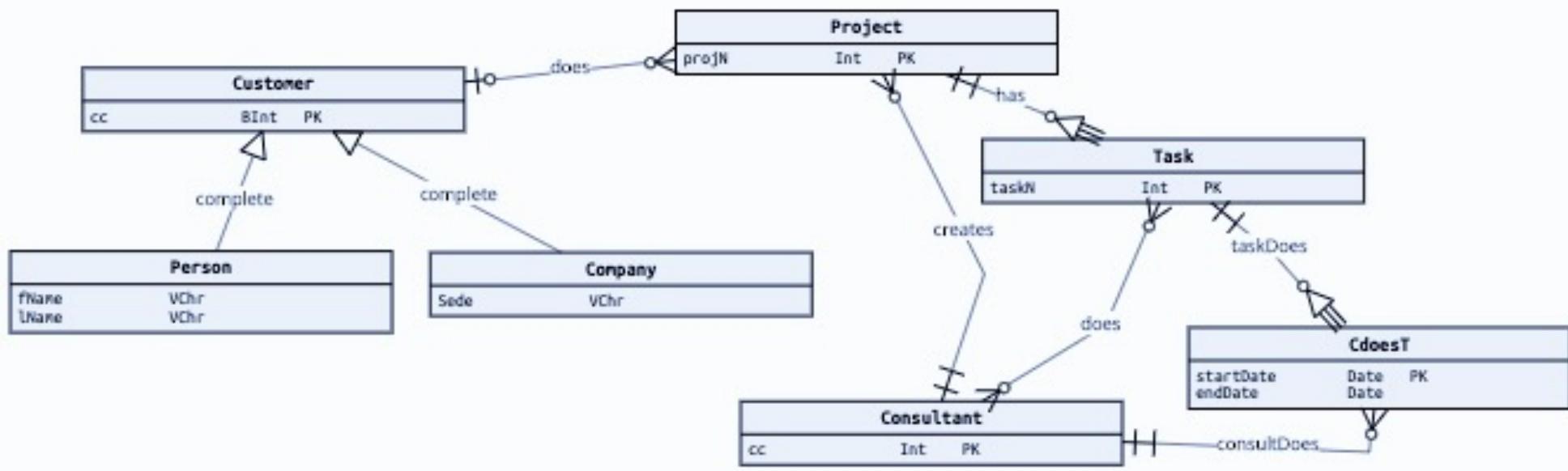


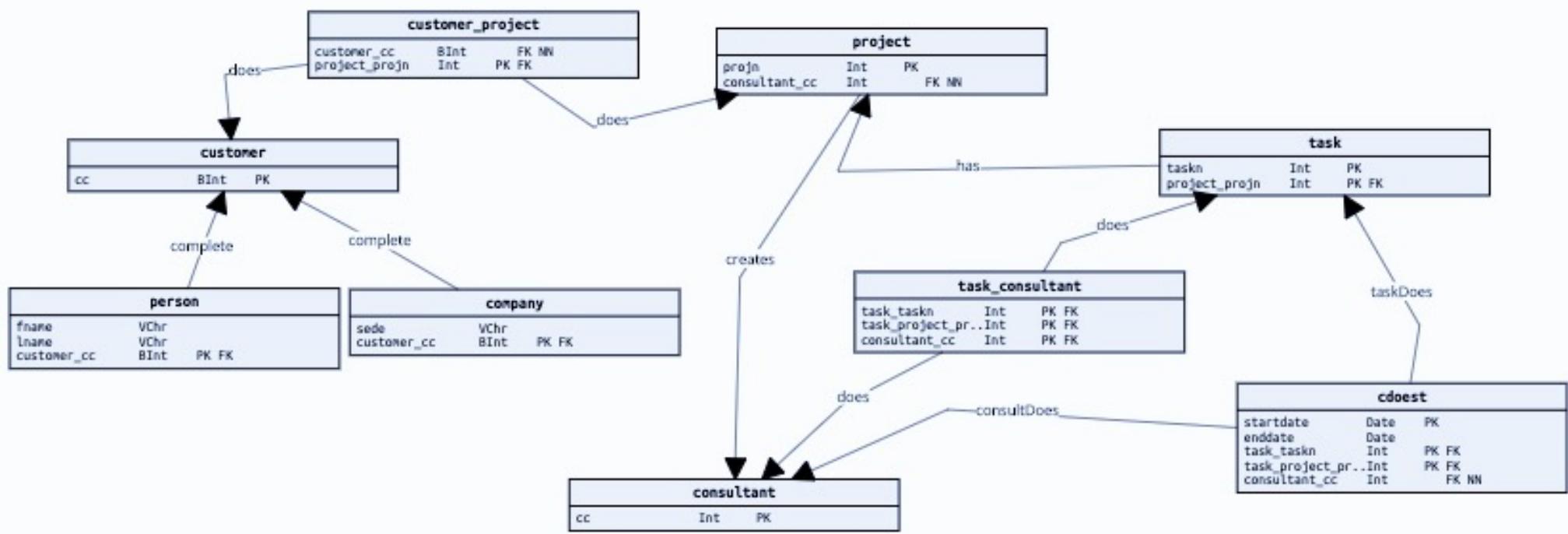
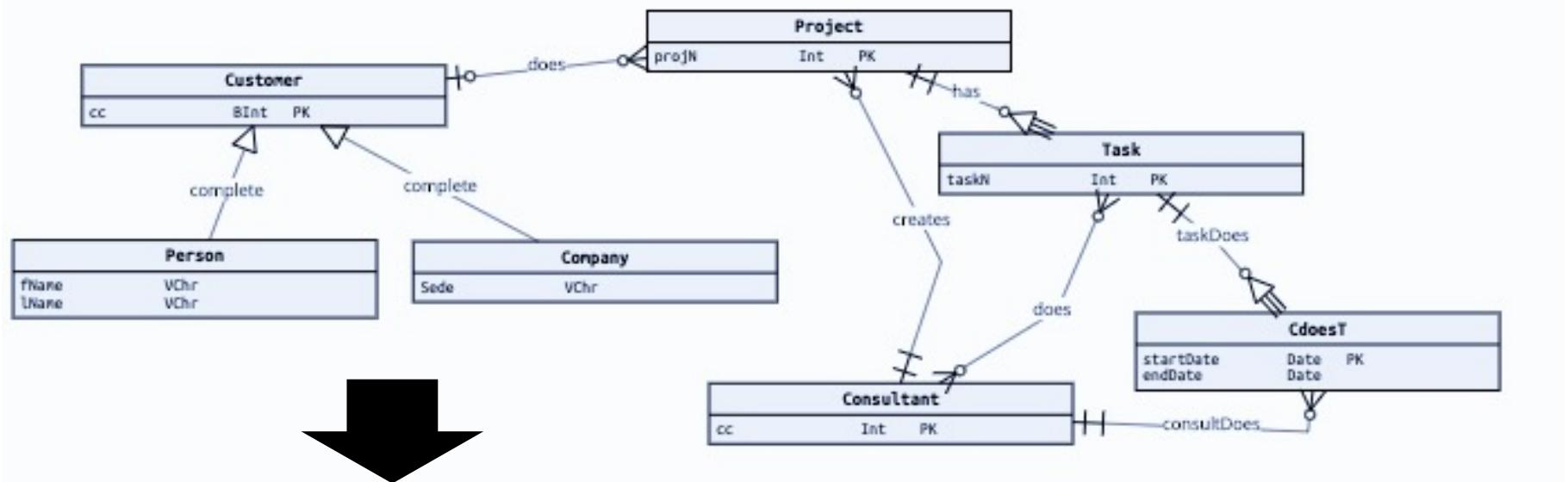
ER para Relacional



Extras

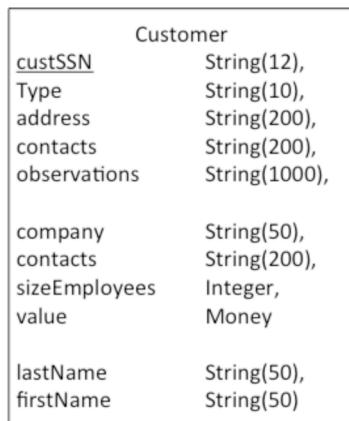
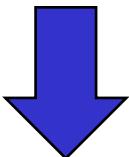
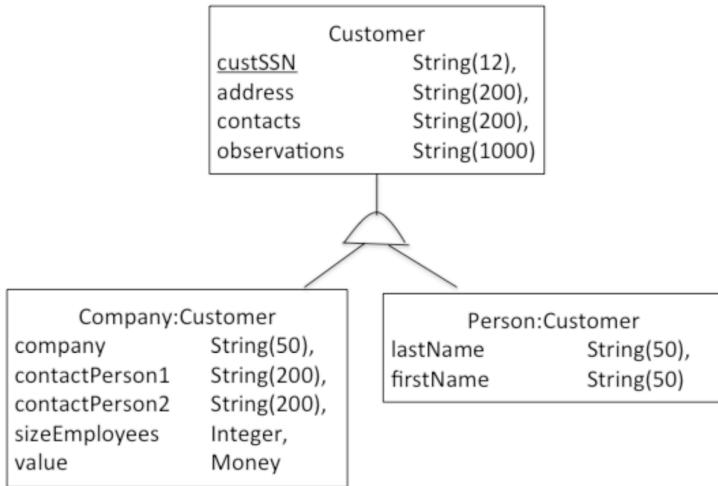
Faz-se bem o ER e físico no onda





Exemplo 2 – Project customers

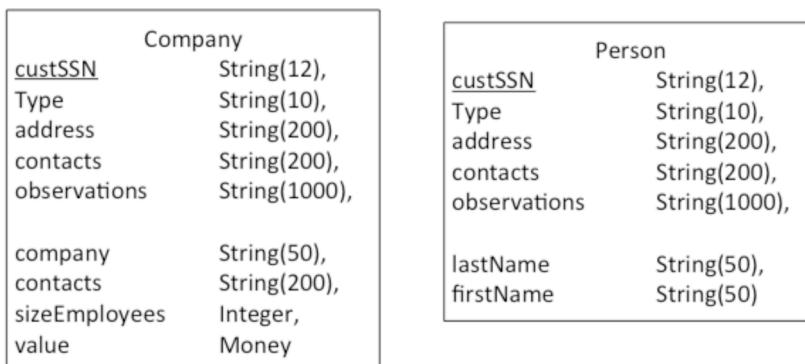
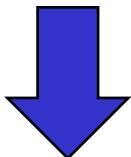
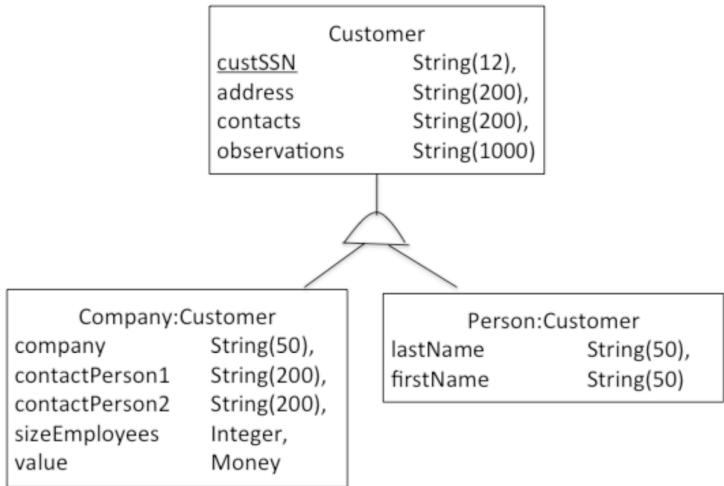
Inheritance (1) - SINGLE



```

Customer(
  //custSSN, type,
  address, contacts, observations
  203456765, 'Person', 'Rua da Alegria, 216, Lisboa, Portugal', '21-
  223543, 96-3454332', null,
  //company, contactPerson1, contactPerson2, sizeEmployees, value
  null, null, null, null, null,
  //lastName, firstName
  'Jorge', 'Silva'
)
  
```

Inheritance (2) CONCRETE



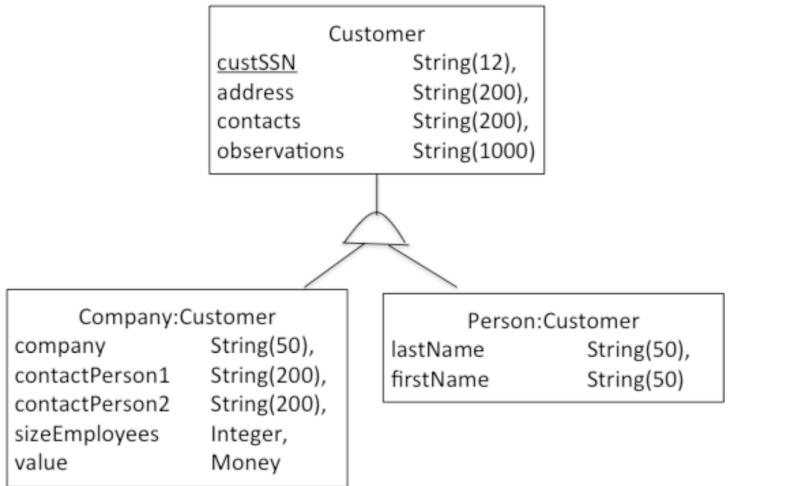
```

Person(
//custSSN, type,address, contacts, observations
203456765, 'Person', 'Rua da Alegria, 216, Lisboa,
Portugal', '21-223543, 96-3454332', null,
  
```

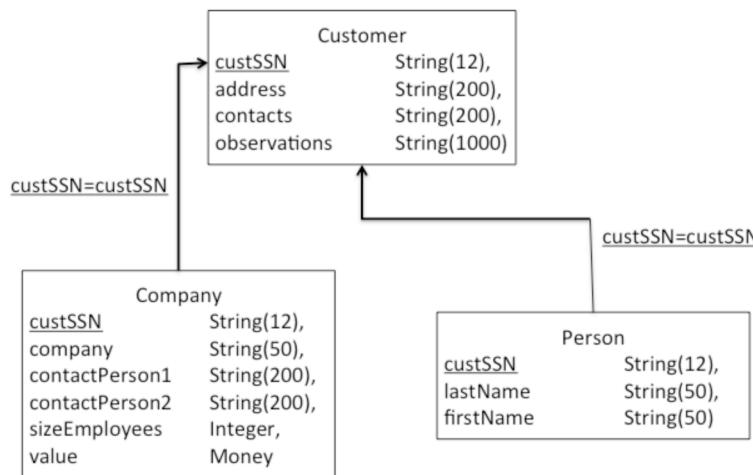
```

//lastName, firstName
'Jorge', 'Silva'
)
  
```

Inheritance (3) - COMPLETE



Customer(
 //custSSN, type,address, contacts, observations
 203456765, 'Person', 'Rua da Alegria, 216, Lisboa, Portugal', '21-223543, 96-3454332', null)
)

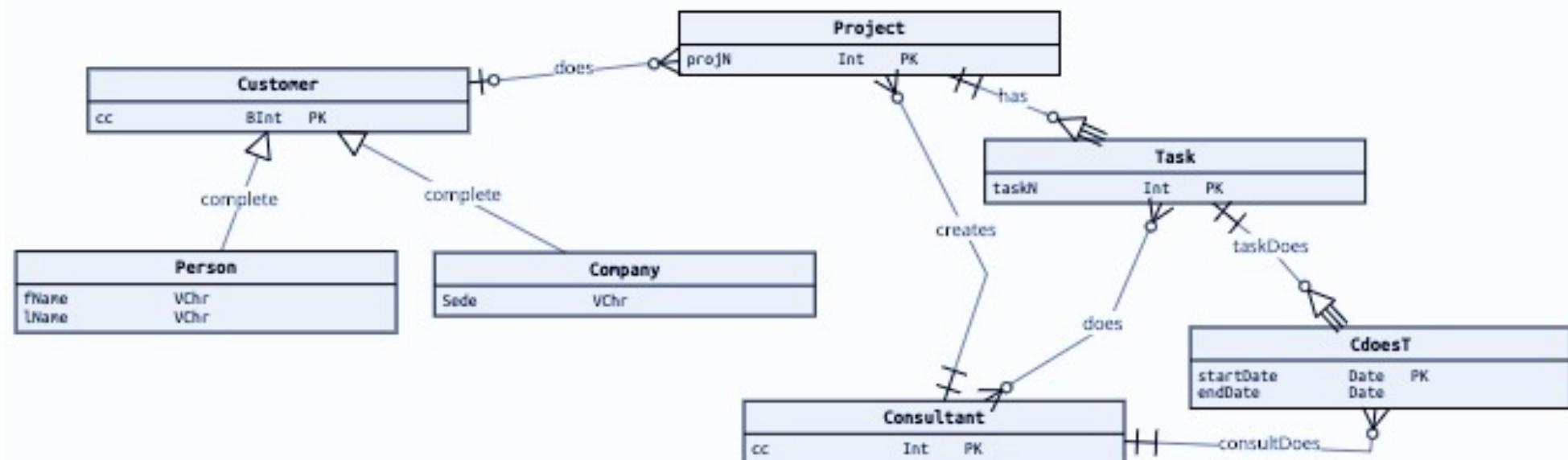


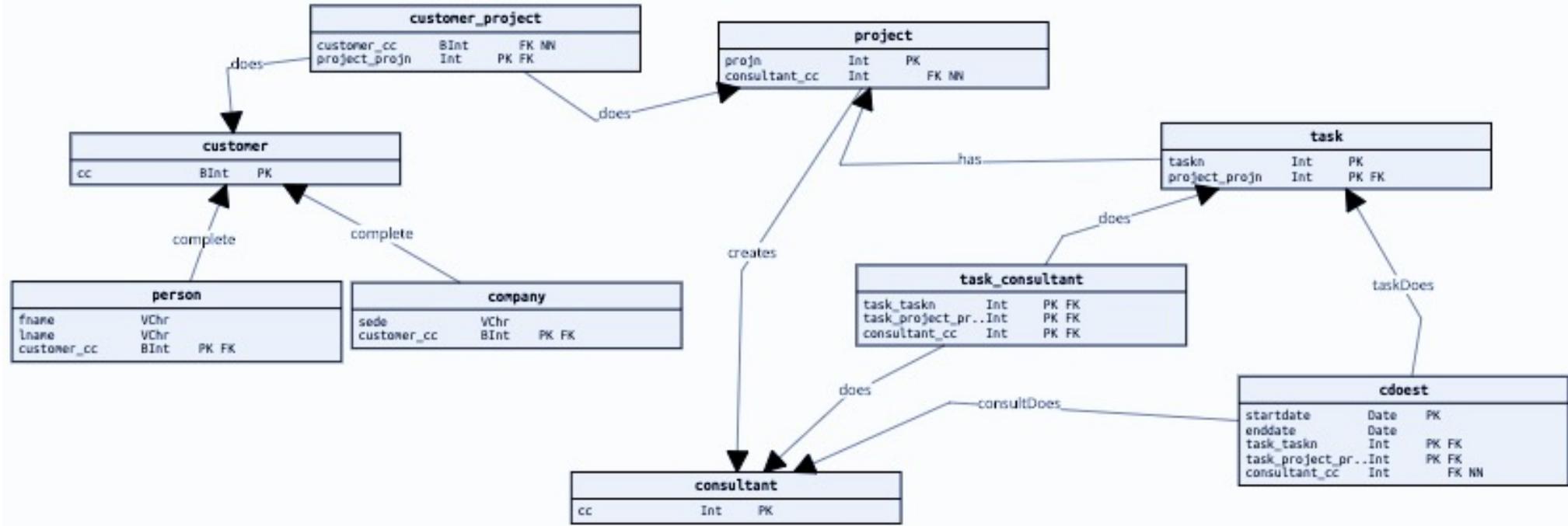
Customer(
 //custSSN, type,address, contacts, observations
 203456765, 'Person', 'Rua da Alegria, 216, Lisboa, Portugal', '21-223543, 96-3454332', null)
)

Person(
 //custSSN,lastName, firstName
 203456765, 'Jorge', 'Silva'
)

ER e físico completos

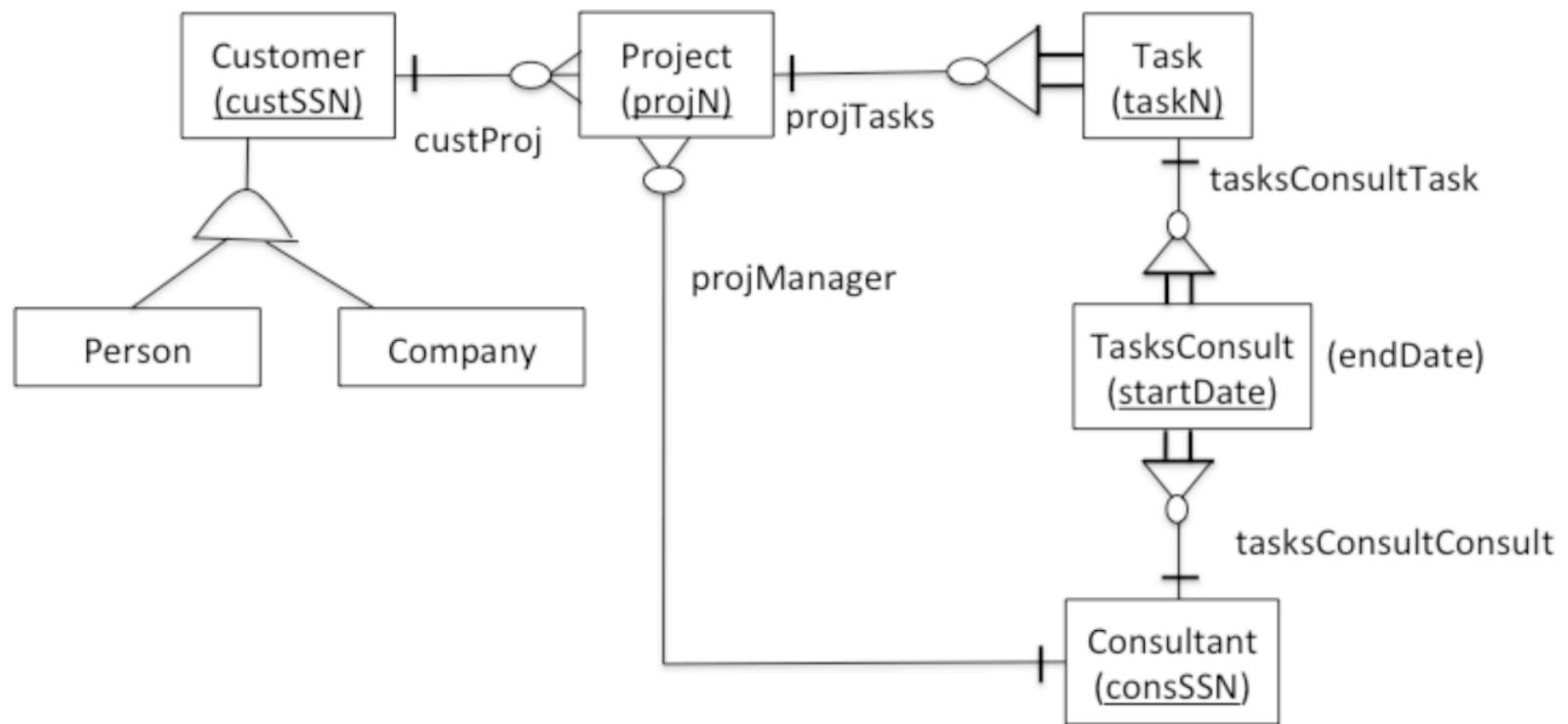
Project-tasks





Extras

Fazer R no onda ...



UC

bdai

```

CREATE TABLE Customer(
    custSSN PRIMARY KEY
    type
    address
    contacts
    observations
);

CREATE TABLE Company (
    custSSN PRIMARY KEY
    company
    contactPerson1
    contactPerson2
    sizeEmployees
    value
    FOREIGN KEY custSSN references Customer(custSSN)
);

CREATE TABLE Person(
    custSSN PRIMARY KEY
    lastName
    firstName
    FOREIGN KEY custSSN references Customer(custSSN)
);

Consultant(consSSN)
);

```

```

CREATE TABLE Project(
    projN PRIMARY KEY
    customer
    REFERENCES Customer(custSSN),
    projectManager
    REFERENCES Consultant(consSSN),
    Acronym
    Description
    Location
    startDatePlanned
    endDate
    CHECK(endDate>startDate),
    totalBudgetplanned Numeric(10,2),
    currentBalance
    totalBudgetreported
);

```

```

CREATE TABLE Task(
    projN
    taskN
    title
    description
    needs
    observations
    startDatePlanned
    startDate
    endDatePlanned
    endDate
    CHECK(endDate>startDate),
    MMplanned
    MMreported
    budget
    budgetRH
    budgetEquipment
    budgetConsumables
);

```

```

CREATE TABLE Consultant(
    consSSN PRIMARY KEY
    lastName NOT NULL
    firstName NOT NULL
    address
    contacts
    job
    expertise
    curriculum
    hourlyRate
    commission
    startDate
    leaveDate
);

```

```

CREATE TABLE TaskConsultant(
    projN
    taskN
    consSSN
    startDate
    numberOfDays
    description
    FOREIGN KEY projN REFERENCES Task(projN),
    FOREIGN KEY taskN REFERENCES Task(taskN),
    FOREIGN KEY consSSN REFERENCES
);

```

Extras ... Os exemplos
(para aulas mais à frente)

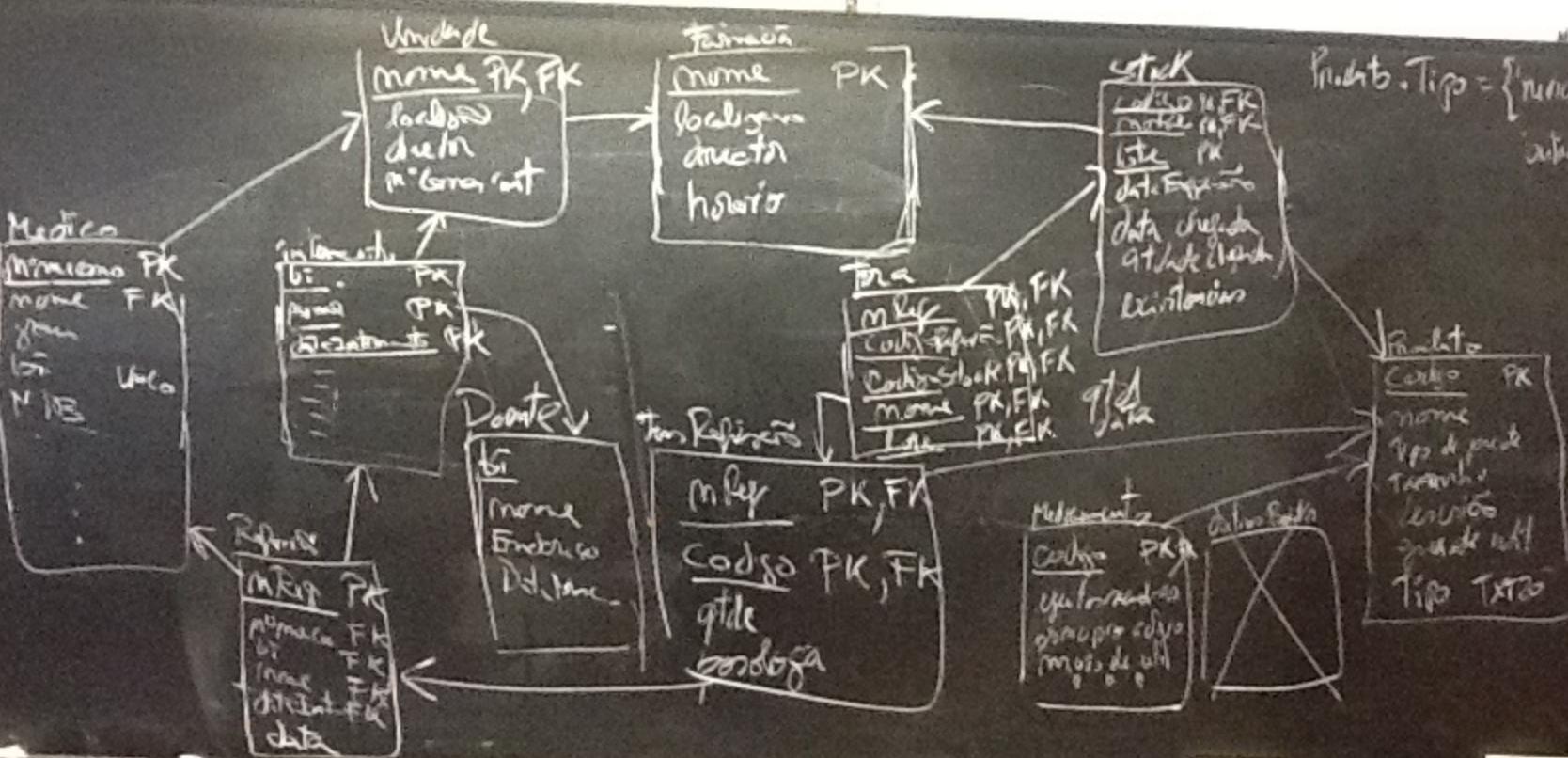
Exemplo do Hospital – ER-R

Um grande hospital pretende informatizar a **gestão da administração de medicamentos** aos seus doentes, para o que necessita de uma base de dados cuja concepção deve ter em conta os seguintes pontos:

- Cada unidade do hospital (cardiologia, otorrino, medicina interna, etc.) tem a sua farmácia própria. Os medicamentos existentes na farmácia de cada unidade são genericamente determinados pelas necessidades habituais dos doentes internados nessa unidade. Contudo, a farmácia de cada unidade tem também alguns medicamentos de uso mais geral e até medicamentos que também são típicos das farmácias de outras unidades. O hospital tem ainda uma farmácia central cujo stock pretende cobrir os medicamentos mais vulgares de todas as especialidades.
- Os medicamentos usados no hospital são todos genéricos e são descritos por um vasto conjunto de atributos (código, composição química, apresentação [comprimidos, injeções, etc.], modo de utilização, indicações, etc.). Para cada medicamento específico existente em stock é ainda necessário registar a data limite da validade. As farmácias existentes no hospital também têm outros produtos, tais como próteses, algodão, seringas descartáveis, etc. Estes produtos devem também fazer parte das existências (na base de dados) de cada farmácia;

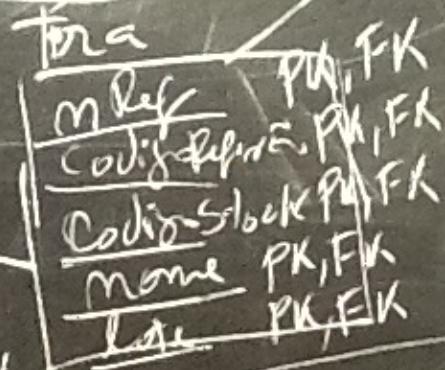
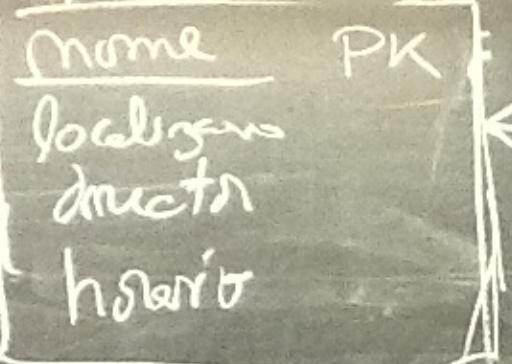
- A gestão do stock de medicamentos é global. Isto é, é feita com base na totalidade dos medicamentos existentes no hospital (farmácias das unidades + farmácia central). Para tal, a base de dados deve permitir saber que medicamentos/produtos existem em todo o hospital e em que farmácia é que eles estão;
- Sempre que um médico prescreve medicamentos a um dado doente (internado no hospital) é feita uma requisição. Tipicamente, cada requisição inclui diversos medicamentos. Sempre que possível, a requisição deve ser satisfeita na farmácia da unidade em que o doente está internado. Os medicamentos (de uma requisição) que não existam na farmácia da unidade em que foi emitida a requisição são levantados de qualquer outra farmácia do hospital em que eles existam.

Apresente um diagrama ER que represente corretamente a estrutura da base de dados pretendida. Pode introduzir novos elementos que completem a descrição do problema acima efectuada, se tal lhe parecer necessário e/ou conveniente. Todos os pressupostos em que se basear, que não tenham sido referidos no texto, têm de ser justificados. Mostre também o esquema físico (tabelas e vistas, se achar necessário ter vistas) e explique os aspectos mais relevantes da preservação da integridade da base de dados.

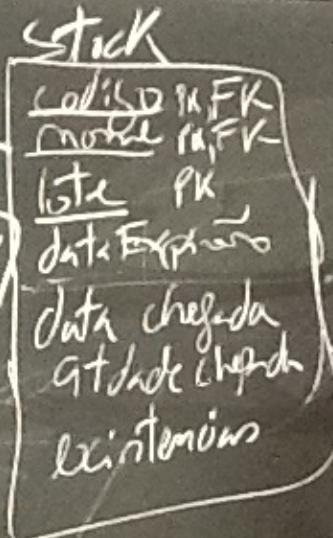
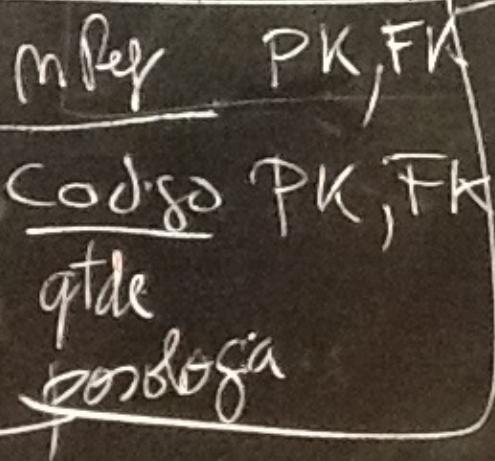


Primitivos.Tipos = { 'numerico',
'autoPreenchido' }

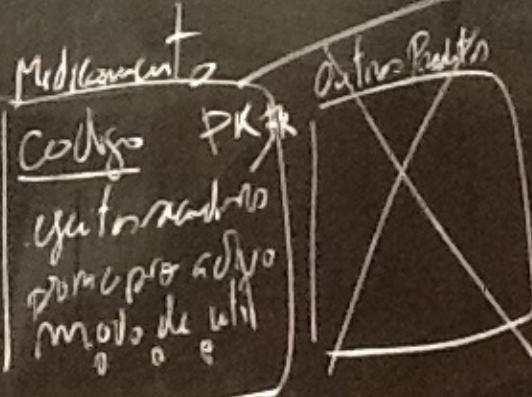
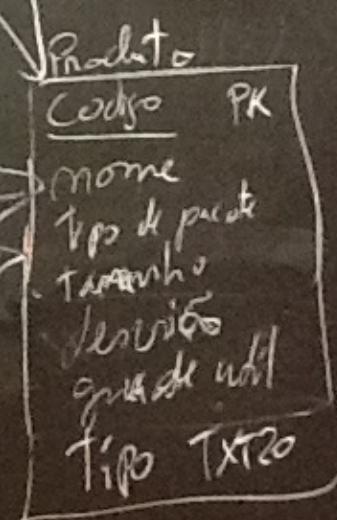
Farmácia



Termos Refinados

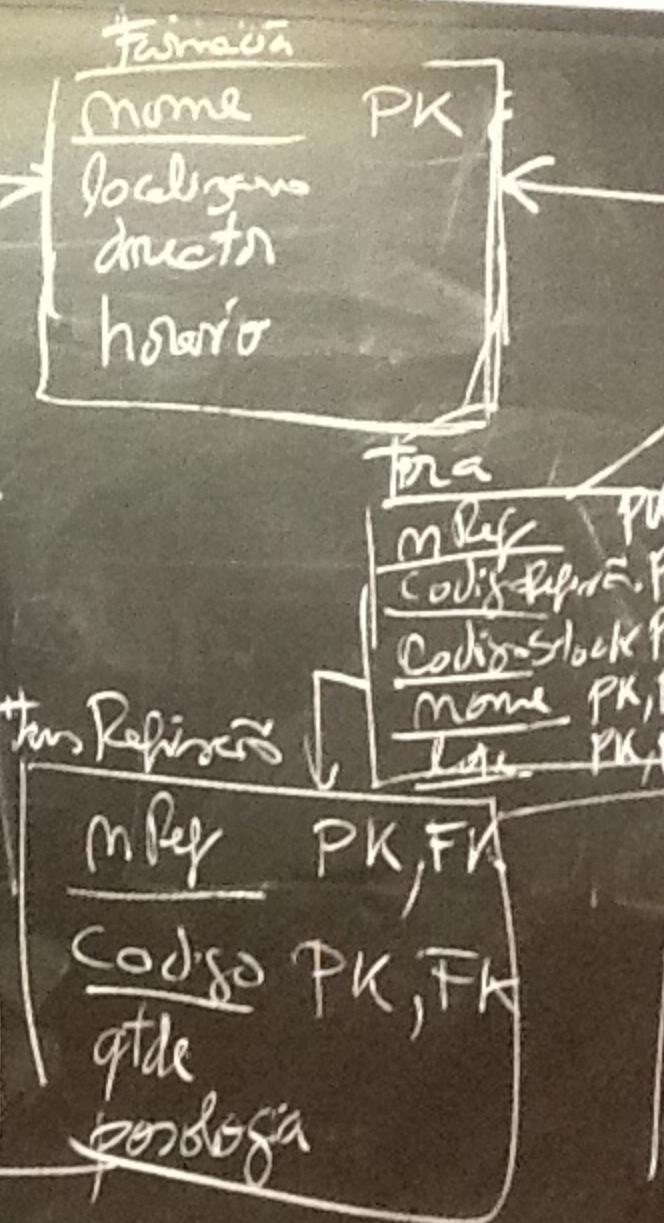
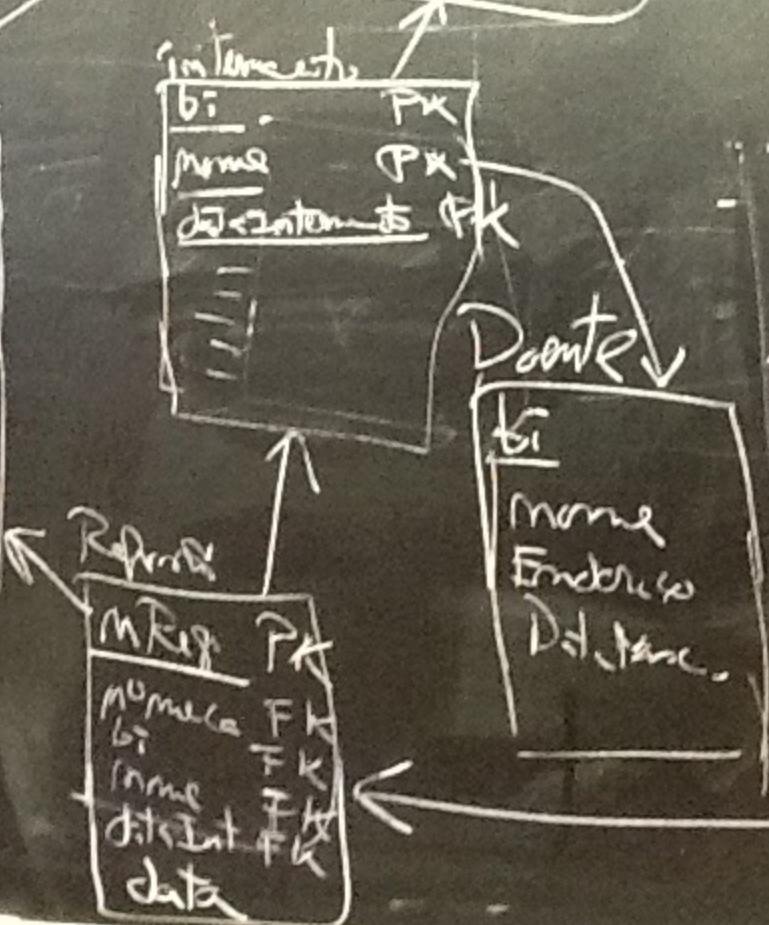


Produto.Tipo = {'Medicamento',
'OutroProduto'}



Medico

matrícula	PK
nome	FK
CRM	
CRM	
Nº	



Exemplo da eleições ER-R

Exemplo das Eleições

- A Comissão Nacional de Eleições pretende ter uma base de dados para gerir as próximas eleições em que os eleitores já vão votar por um processo electrónico.
- O cartão de eleitor passará a ser um cartão magnético (do tipo do Multibanco), que o identificará univocamente, podendo os eleitores votar nos terminais Multibanco (usando o cartão de eleitor) ou pela Internet.
- Para votar o utilizador terá apenas de ir ao Multibanco mais próximo no dia definido para as eleições, introduzir o cartão na máquina e digitar o seu código secreto recebido da Comissão Nacional de Eleições. Depois basta escolher o partido ou coligação em que deseja votar e sair.
- Na Internet o eleitor deverá aceder à página da Comissão Nacional de Eleições, digitar manualmente o número do seu cartão de eleitor e o código secreto, votando depois de forma semelhante ao que pode fazer nos terminais Multibanco.
- Este sistema tem muitas vantagens, tais como eliminar a operação de contagem de votos (que demora bastante tempo e dá azo a erros e a fraudes) ou permitir a um eleitor votar em qualquer ponto do País, independentemente da freguesia onde está recenseado.
- Deixa também de ser necessário ter o antiquado sistema de mesas de voto. O projecto do esquema de dados desta aplicação deve obedecer aos seguintes requisitos:

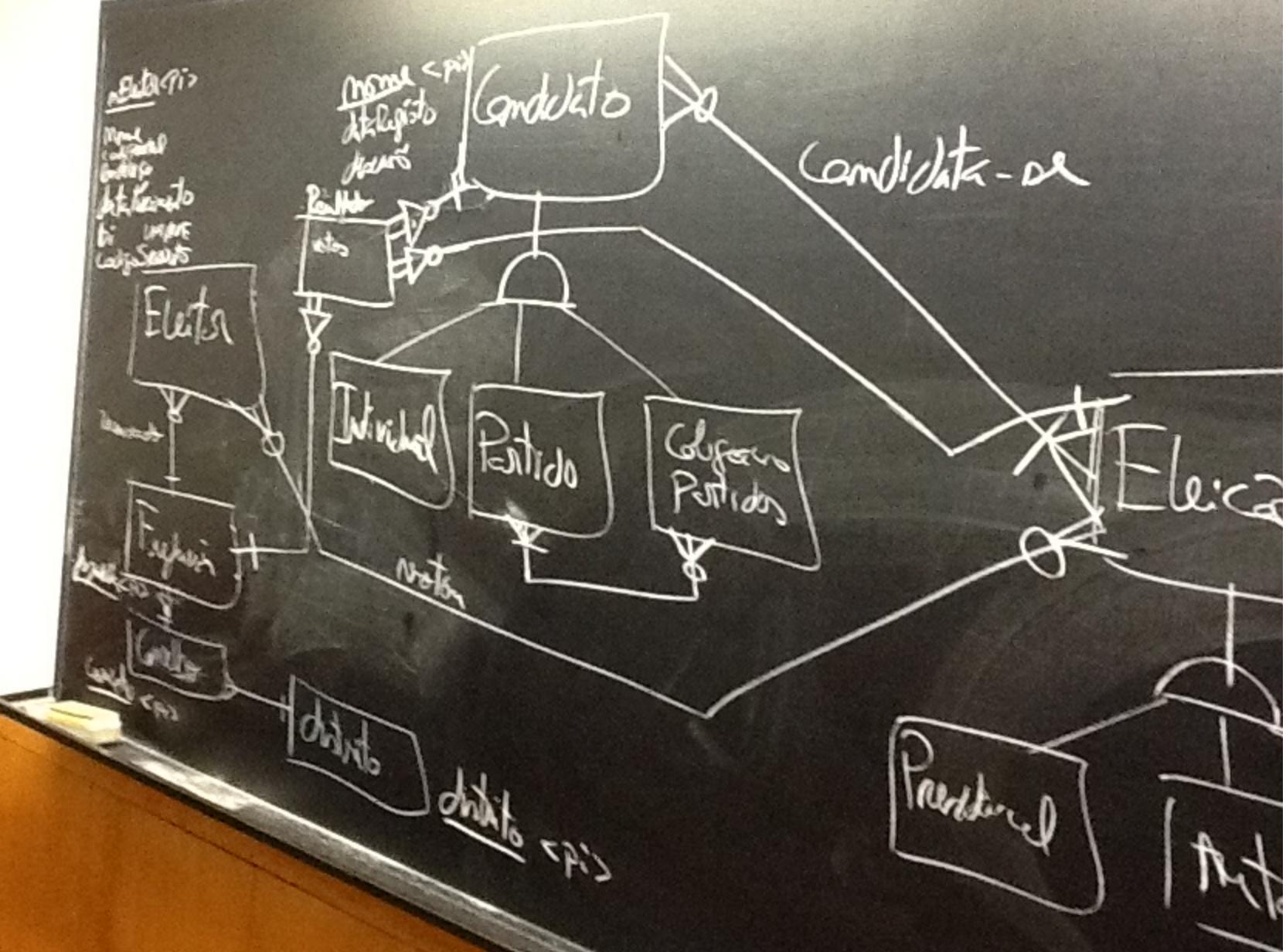
Exemplo das Eleições

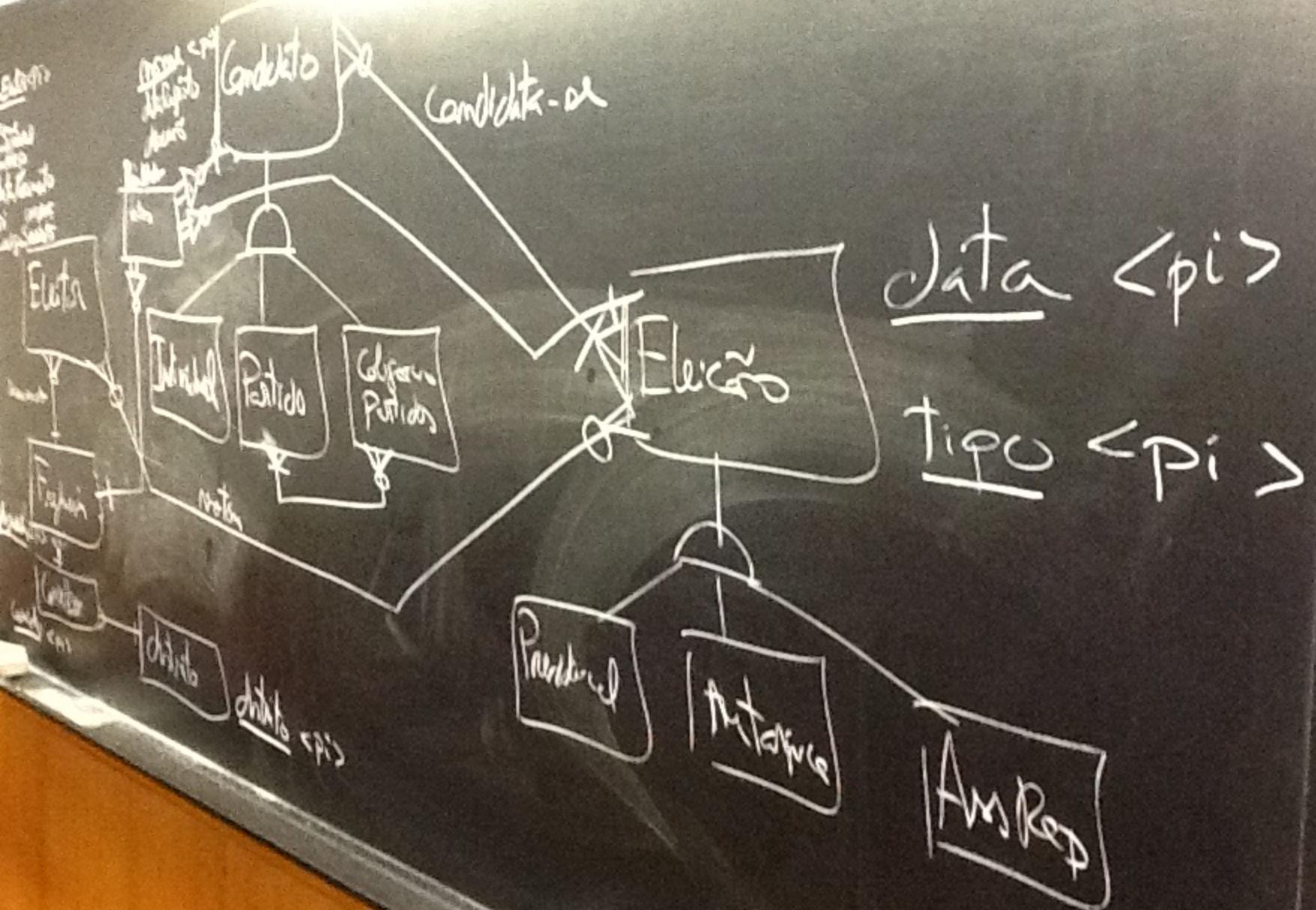
- O novo sistema destina-se às eleições para a assembleia da república, eleições autárquicas e presidenciais, devendo poder armazenar os resultados de diversas eleições, de forma a tornar fáceis as comparações entre votações em eleições sucessivas.
- A base de dados regista informação sobre todos os eleitores. Para além da informação habitual (nome, endereço postal, data de nascimento, BI, etc.) é necessário registar também o número do cartão do eleitor e o seu código secreto (este atributo será cifrado na tabela mas isso faz parte da matéria de Bases de Dados II). Não se esqueça de que o número de eleitores é muito grande, pelo que é importante normalizar as tabelas que registam informação sobre eleitores para minimizar o espaço ocupado pela base de dados.
- Cada eleitor está recenseado numa freguesia, pelo que apesar de poder votar em qualquer lugar, o seu voto ficará associado à freguesia onde o votante está recenseado. Isto não é muito importante para as eleições para a assembleia da república, que tem carácter nacional, mas é muito relevante para as eleições autárquicas. Claro está que a base de dados necessita de ter informação relativa à divisão administrativa do país, ou seja, precisa de conhecer todas as freguesias, concelhos e distritos portugueses.
- A uma dada eleição concorrem partidos ou coligações de partidos. Estes partidos e/ou coligações podem variar ao longo do tempo, ou seja de eleição para eleição. No caso particular das eleições autárquicas, e para simplificar, vamos admitir em todas as freguesias concorrer os mesmos partidos e/ou coligações (como se sabe, não é assim na realidade, visto cada freguesia ter boletins de voto próprios para as autárquicas). Nas eleições presidenciais as candidaturas são, obviamente, em nome individual.
- O sistema tem de garantir o anonimato dos votos, ou seja, na base de dados não poderá ficar armazenada qualquer informação que permita relacionar um eleitor com o seu voto. No entanto, é necessário evitar que o mesmo eleitor vote mais do que uma vez na mesma eleição.

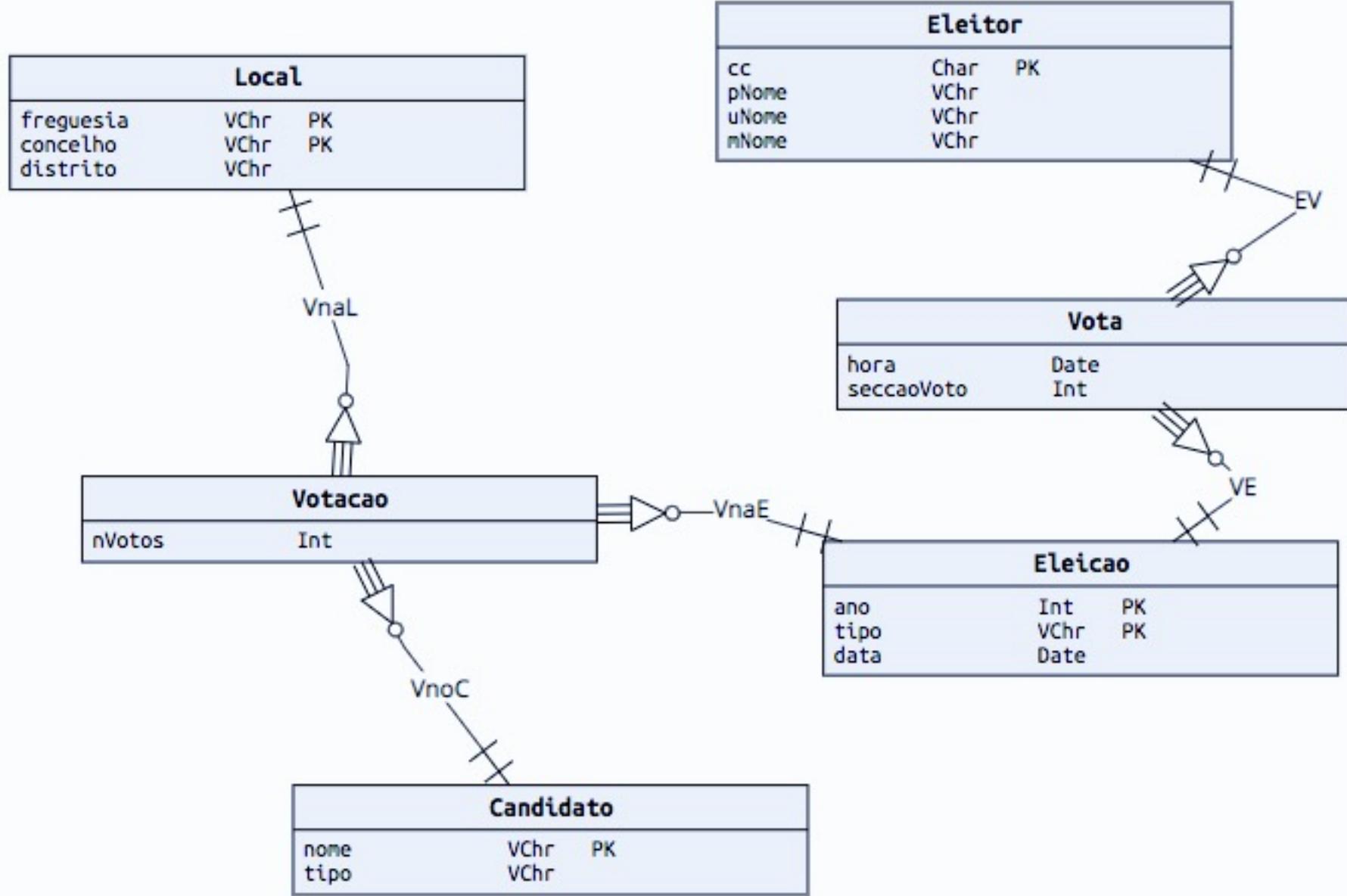
Exemplo das Eleições

- Apresente um diagrama ER que represente correctamente a estrutura da base de dados pretendida. Pode introduzir novos elementos que completem a descrição do problema acima efectuada, se tal lhe parecer necessário e/ou conveniente. Todos os pressupostos em que se basear, que não tenham sido referidos no texto, têm de ser justificados. Mostre também o esquema físico (tabelas e vistas, se achar necessário ter vistas) e explique os aspectos mais relevantes da preservação da integridade da base de dados.

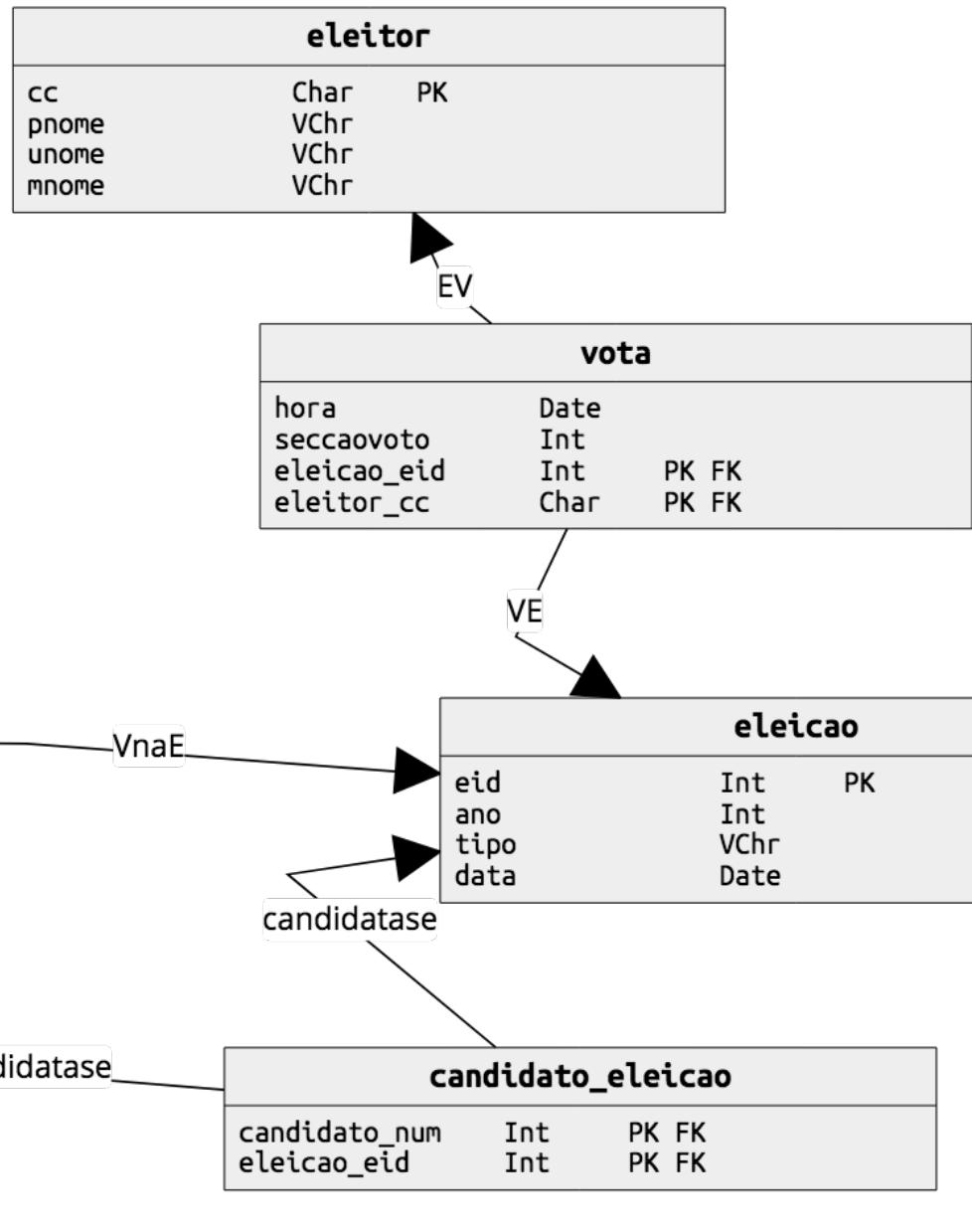
Power Person







local		
freguesia	VChr	PK
concelho	VChr	PK
distrito	VChr	



Relacionamento binário de grau 1:N – caso 4

Caso 1 (1:N) - participação obrigatória da entidade do lado N
(disciplina tem de ter sempre um professor)

Diagrama ER

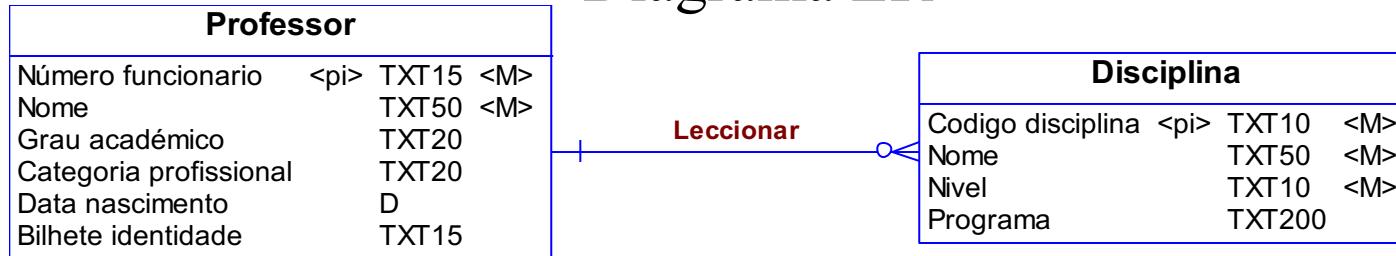
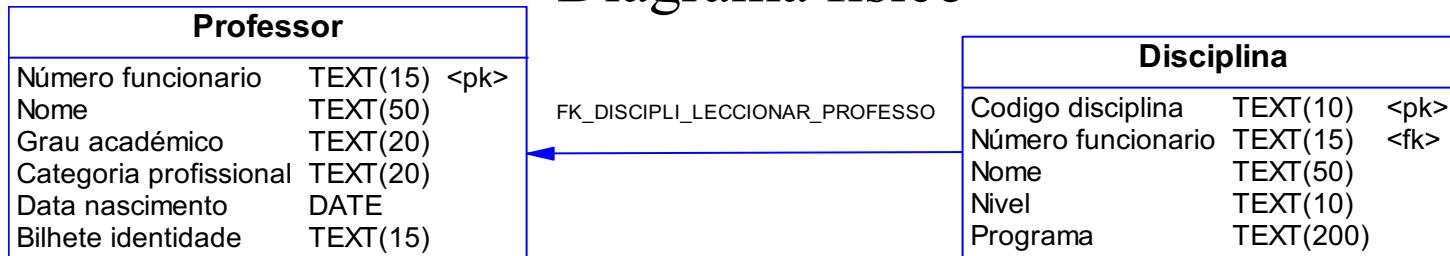


Diagrama físico

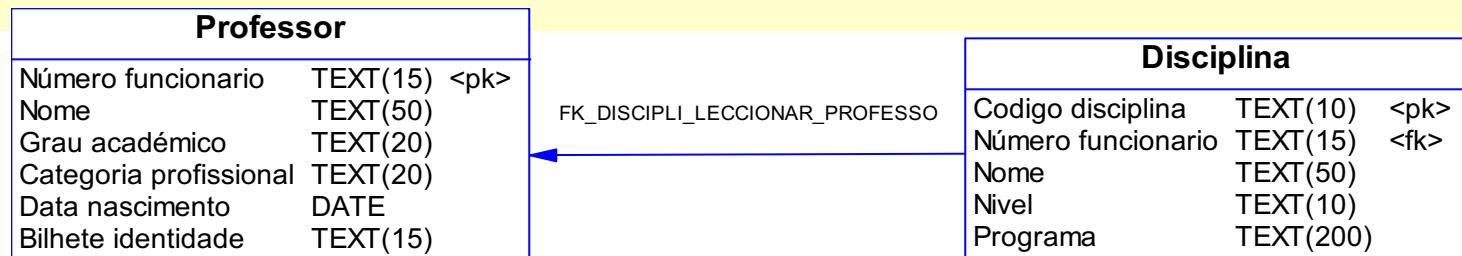


São necessárias duas tabelas. A tabela Disciplina tem uma chave estrangeira que a liga à tabela Professor

Regra 4

Relacionamento binário de grau 1:N e participação obrigatória do lado N

- São necessárias duas tabelas, uma para cada entidade.
- 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 (chave estrangeira) na tabela correspondente à entidade do lado N.



Relacionamento binário de grau 1:N – caso 5

Caso 2 (1:N) - participação não obrigatória do lado N

Diagrama ER

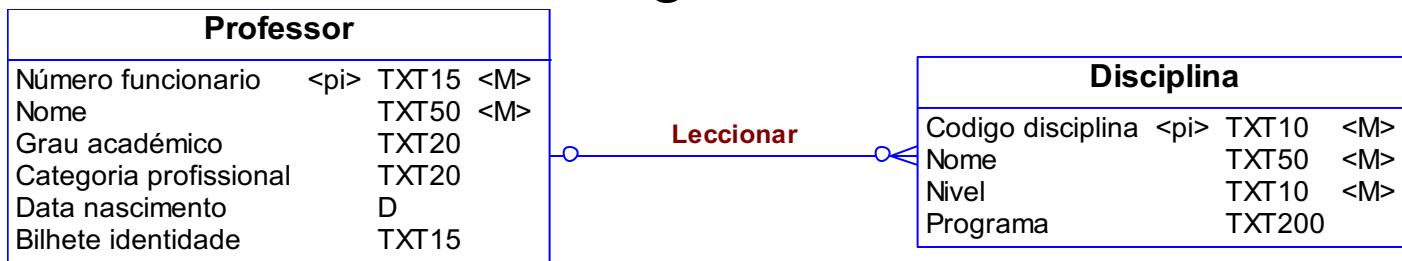
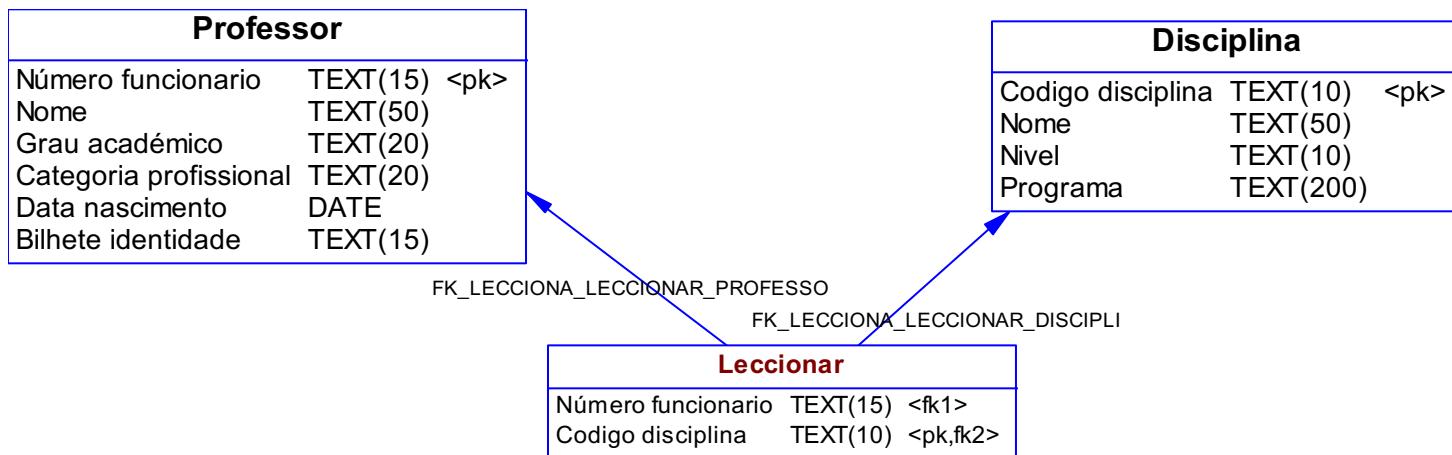


Diagrama físico (**versão mais “formal”= FK not null**)



Regra 5

Relacionamento binário de grau 1:N e participação não obrigatória do lado N

- São 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 cada entidade).
- A tabela relativa ao relacionamento terá de ter entre os seus atributos as chaves primárias de cada uma das entidades (como chaves estrangeiras). A chave primária da tabela correspondente ao relacionamento deve ser o atributo que é chave na tabela do lado N.

Simplificação do caso 5

A geração de tabelas para o caso 5 (relacionamento de grau 1:N com participação não obrigatória do lado 1) pode simplificar-se, dando origem a apenas duas tabelas (como no caso 4)
Neste caso a FK poderá ser null

Diagrama ER

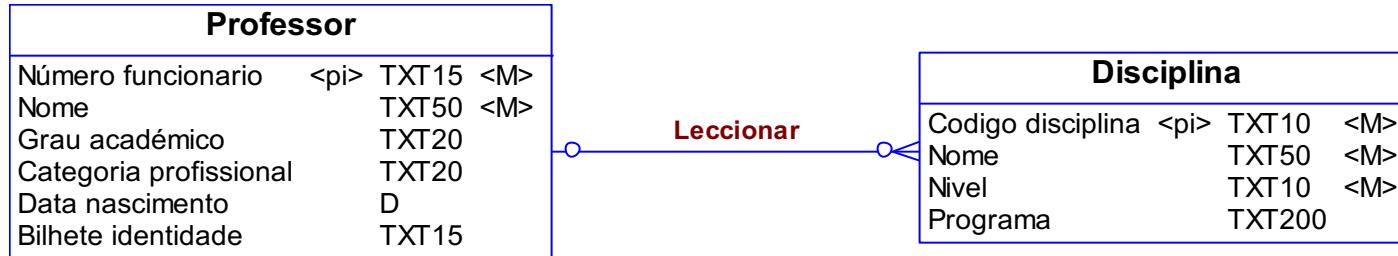
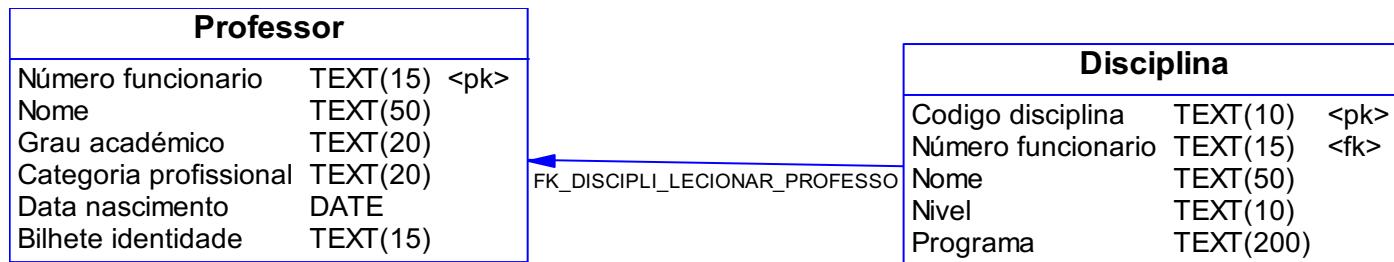


Diagrama físico



Relacionamento binário de grau M:N – caso 6

Diagrama ER

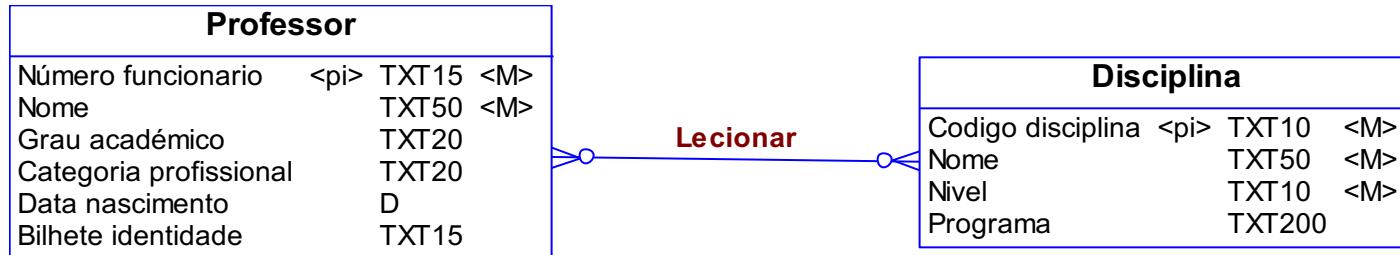
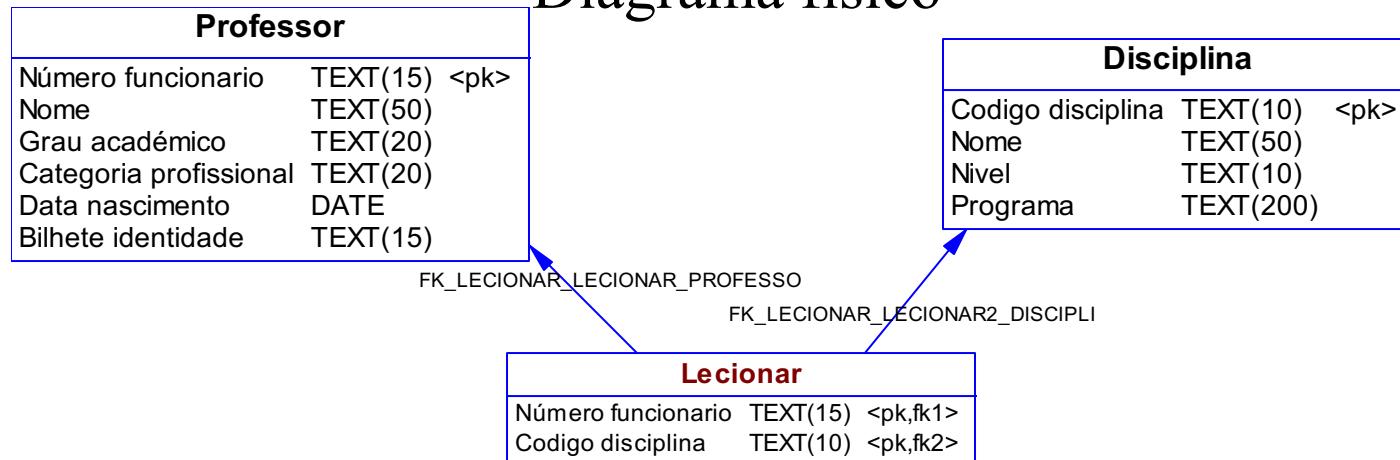


Diagrama físico



Quando o grau de relacionamento binário é do M:N, independentemente do tipo de participação ser obrigatório ou não, são sempre necessárias três tabelas.

Regra 6

Relacionamento binário de grau M:N

- 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 as chaves primárias de cada uma das entidades (que são chaves estrangeiras, estabelecendo a ligação com cada uma das outras duas tabelas). A chave primária da tabela relativa ao relacionamento é composta pelos dois atributos (chaves primárias de cada uma das entidades)

