

Bases de Dados e Análise de Informação

Pedro Furtado

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

Physical model = tables

- Given a **real-world need**, create the database = **data model**.
- Schema = set of tables with attributes, constraints, primary keys, foreign keys that represents the data of the application -> **Physical model** or the **relational model**
- **Conceptual or entity-relationship model = set of concepts and how they relate**

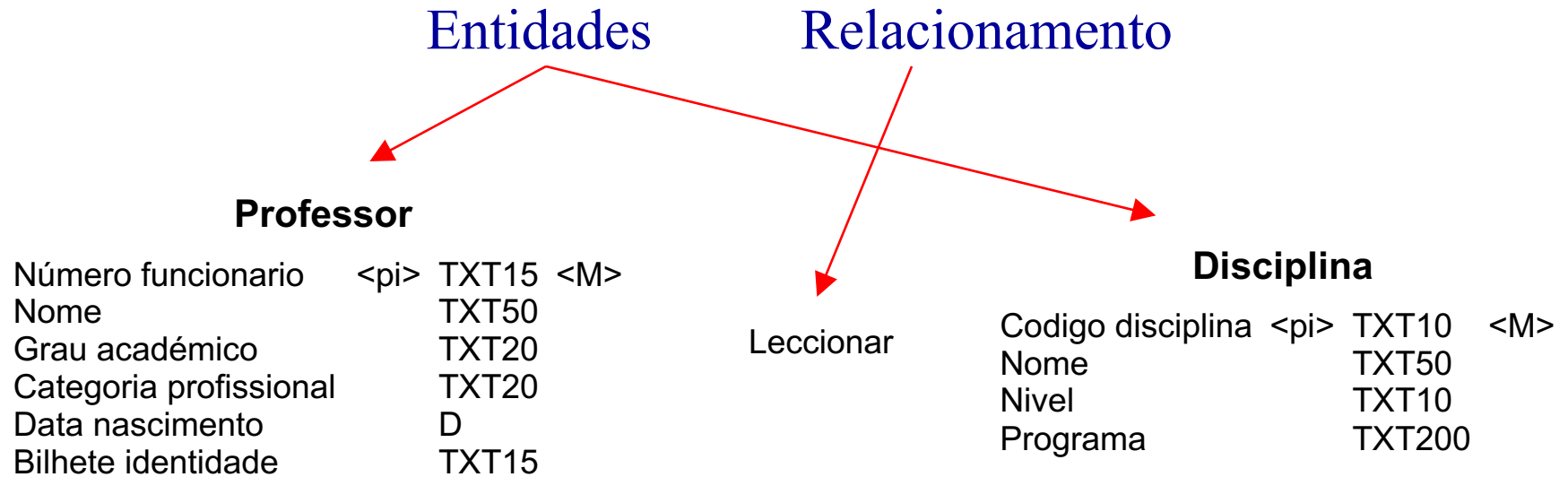
Tables or entities?

- Why entities and relationships = **entity-relationship or conceptual model** ?
 - at the **start** of the modelling we only have **vague ideas and concepts** of what is needed,
 - it is better to **concentrate on concepts** instead of more “physical” details such as foreign keys or all the details needed.
 - entities, which are containers holding data;**
 - relationships, which characterize how entities relate to each other.**
- If an entity is employee and the other entity is department, the relationship may be “**works_in**”.
- The physical model will be created automatically from the conceptual model using a fixed set of rules that translate entities and relationships into tables, attributes and foreign keys.

Entidades e relacionamentos

- **Entidade**
coisa relativa ao problema a tratar e sobre a qual há interesse em guardar/manipular informação; uma entidade deve ter ocorrências e deve ser possível distinguir uma ocorrência de outra ocorrência.
- **Relacionamento**
ligação entre entidades.
- **Atributo**
característica de uma entidade (em certos casos também de um relacionamento)

Diagramas ER



Diferentes maneiras de mostrar as entidades - 1

Professor	Leccionar	Disciplina	Só mostra o nome das entidades
------------------	-----------	-------------------	--------------------------------

Professor Número funcionario	Leccionar	Disciplina Codigo disciplina	Mostra o nome e a chave primária
--	-----------	--	----------------------------------

Professor Número funcionario <pi> <M> Nome <M> Grau académico Categoria profissional Data nascimento Bilhete identidade	Leccionar	Disciplina Codigo disciplina <pi> <M> Nome <M> Nivel <M> Programa	Mostra o nome e todos os atributos, indicando os que são obrigatórios (M) e chave primária (pi)
--	-----------	--	---

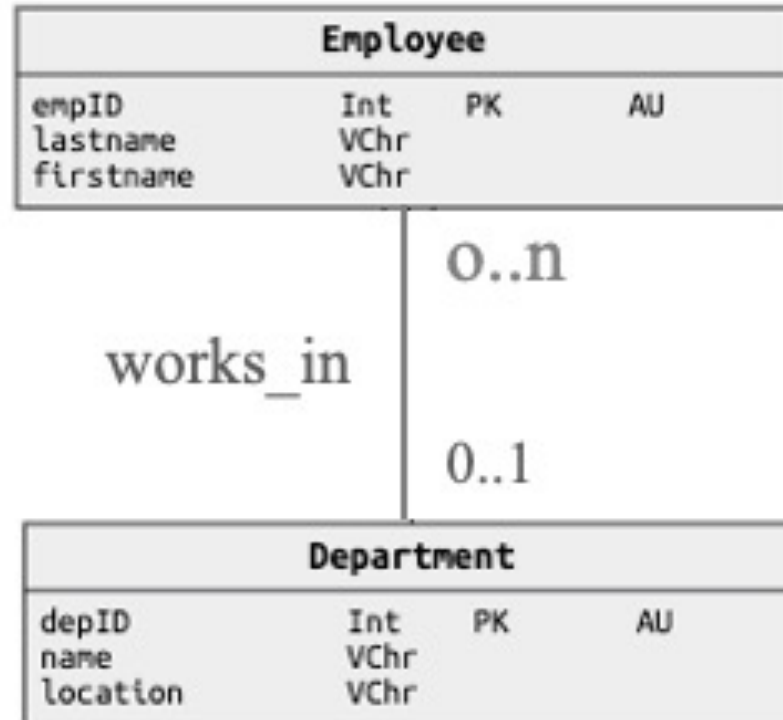
Professor Número funcionario <pi> TXT15 <M> Nome TXT50 <M> Grau académico TXT20 Categoria profissional TXT20 Data nascimento D Bilhete identidade TXT15	Leccionar	Disciplina Codigo disciplina <pi> TXT10 <M> Nome TXT50 <M> Nivel TXT10 <M> Programa TXT200	Mostra também tipos dos dados
--	-----------	---	-------------------------------

Concept of entity

Employee			
empID	Int	PK	AU
lastname	VChr		
firstname	VChr		

Attributes have (conceptual) data types and other constraints defined on them.

Concept of relationship

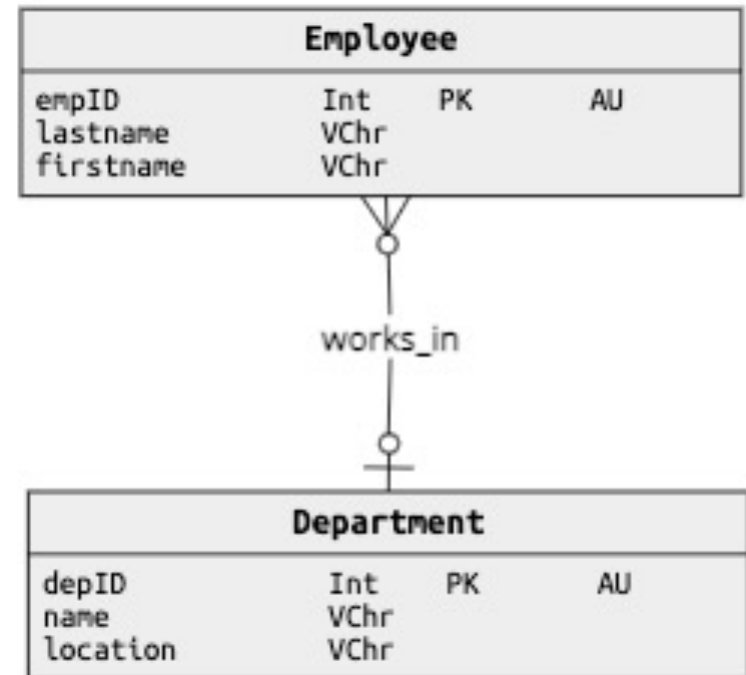
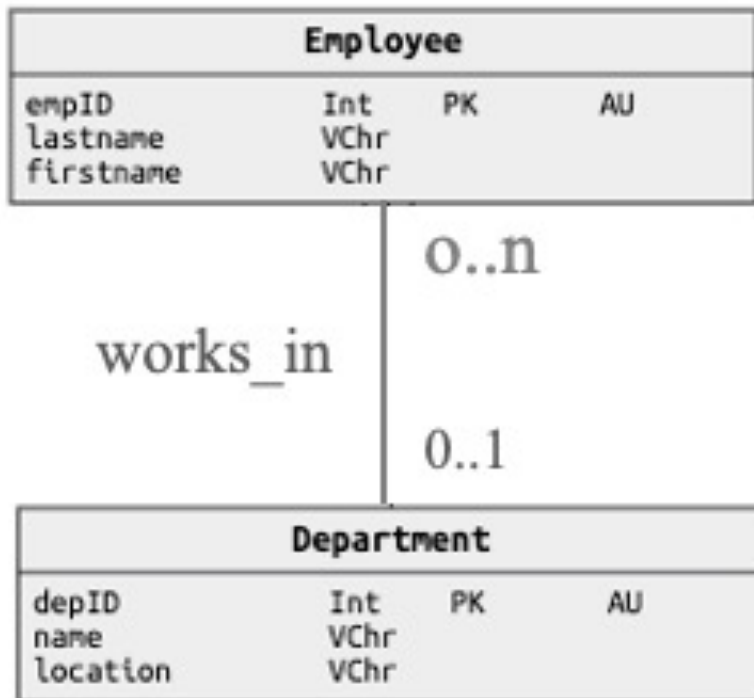


Crow notation

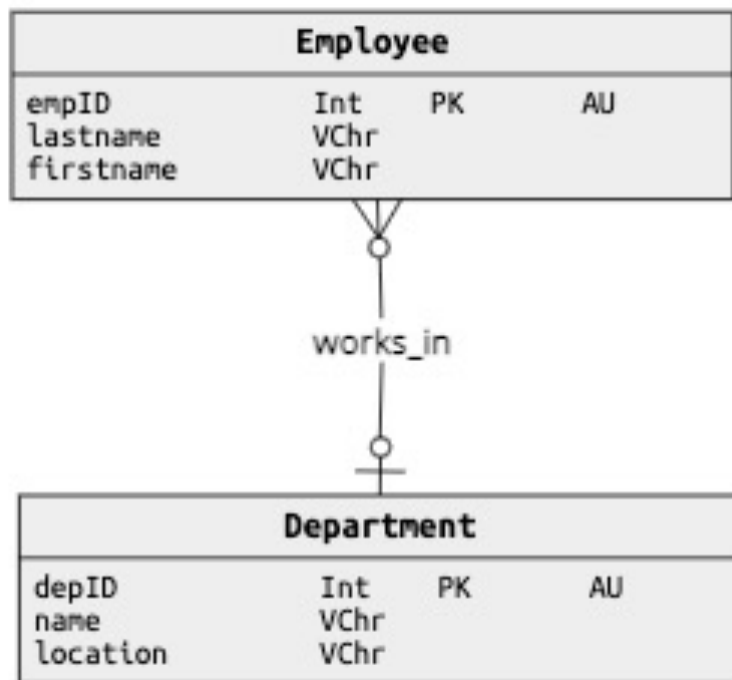
○ = 0 (zero symbol)

— = 1 (one symbol)

⤴ = n (multiple symbol)



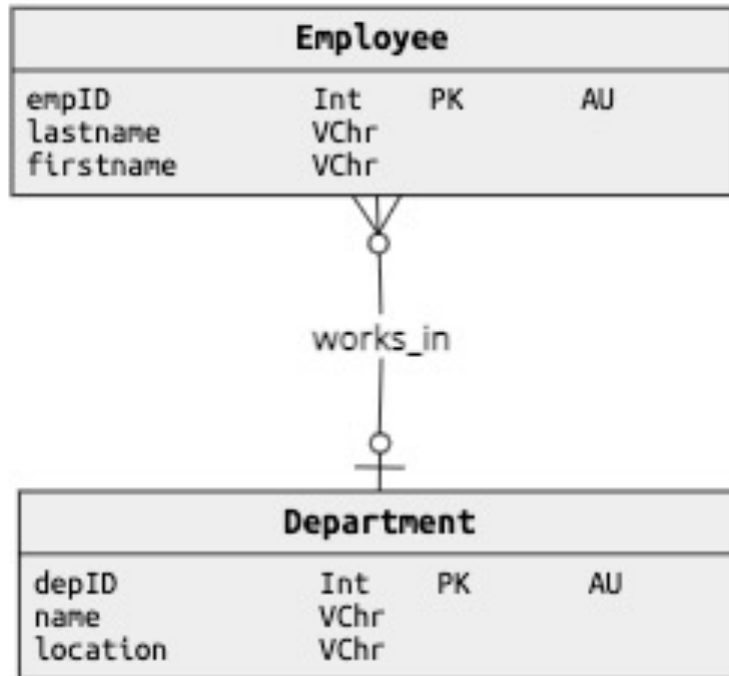
Crow notation ...



An employee has 0 or 1 departments

A department has 0 or more employees

Crow notation



○ = 0 (zero symbol)

— = 1 (one symbol)

↗ = n (multiple symbol)

Participation may take two values: “optional” or “mandatory”. If there is a zero symbol in one side, the participation of the entity in the relationship is optional, otherwise it is mandatory.

Relacionamentos

Os relacionamentos são caracterizados por:

– Participação

Indica se o relacionamento é **obrigatório** ou **não obrigatório**. Por exemplo:

Professor **Leccionar** Disciplina

Neste caso uma disciplina relaciona-se de forma obrigatória com um professor (tem de ter sempre professor) mas o professor pode não ter disciplina (não dá aulas).

– Cardinalidade

Indica qual o **grau** do relacionamento. Por exemplo:

Professor **Leccionar** Disciplina

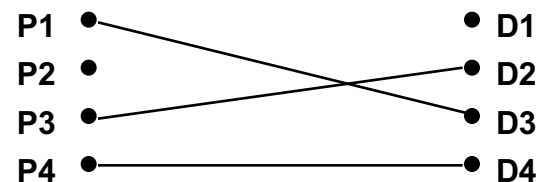
Neste exemplo um professor pode dar mais de uma disciplina mas uma disciplina só é dada por um professor.

Todas as combinações entre participação e cardinalidade são possíveis num relacionamento.

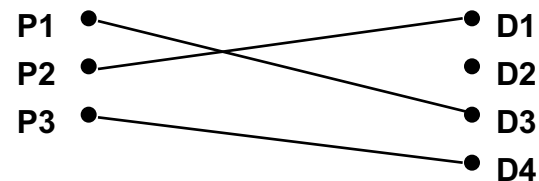
grau 1:1

Participação obrigatória de professor no relacionamento (tem de dar uma disciplina)

Professor Leccionar Disciplina



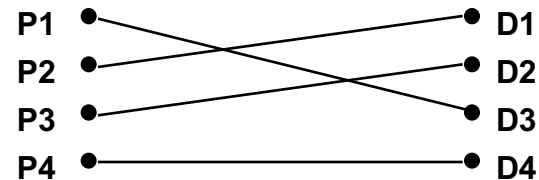
Professor Leccionar Disciplina



Professor Leccionar Disciplina

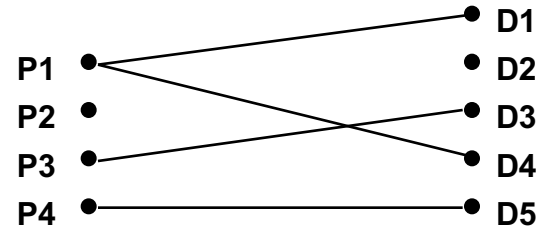


Professor Leccionar Disciplina

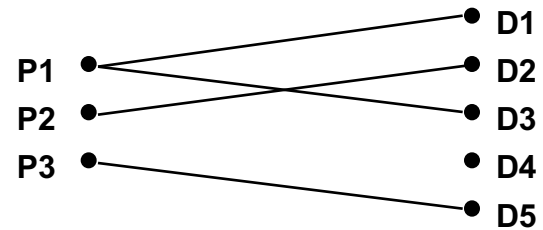


grau 1:N

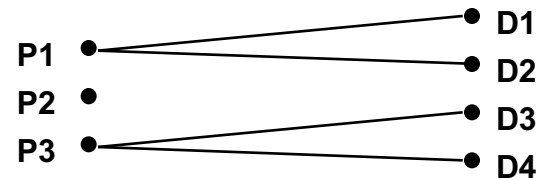
Professor Leccionar Disciplina



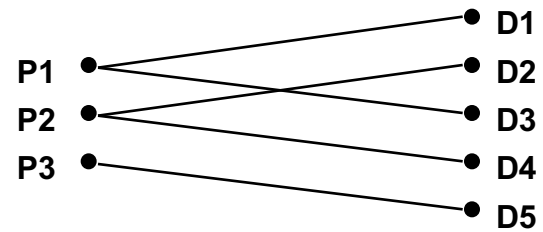
Professor Leccionar Disciplina



Professor Leccionar Disciplina

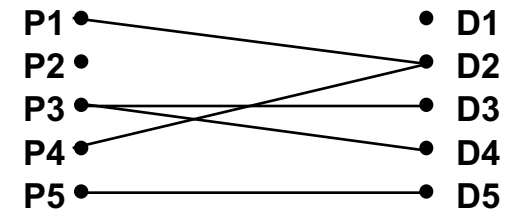


Professor Leccionar Disciplina

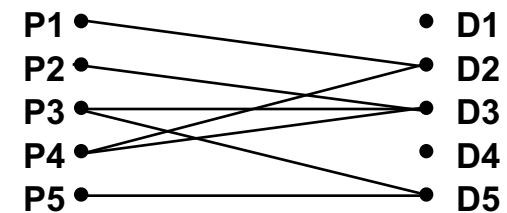


grau N:M

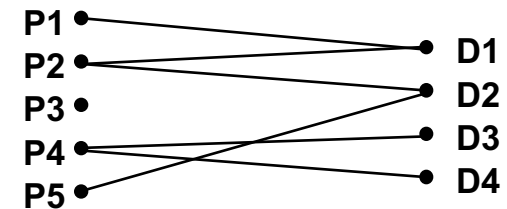
Professor Leccionar Disciplina



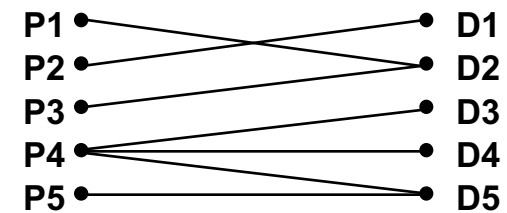
Professor Leccionar Disciplina



Professor Leccionar Disciplina



Professor Leccionar Disciplina



Onda





Onda - entidades

Properties

Entity Name Filme

Fields

Field Name	Field Type	Actions
titulo	Varchar	↑↓ X
ano	Date	↑↓ X

Check Constraints

Properties

Entity Name Filme

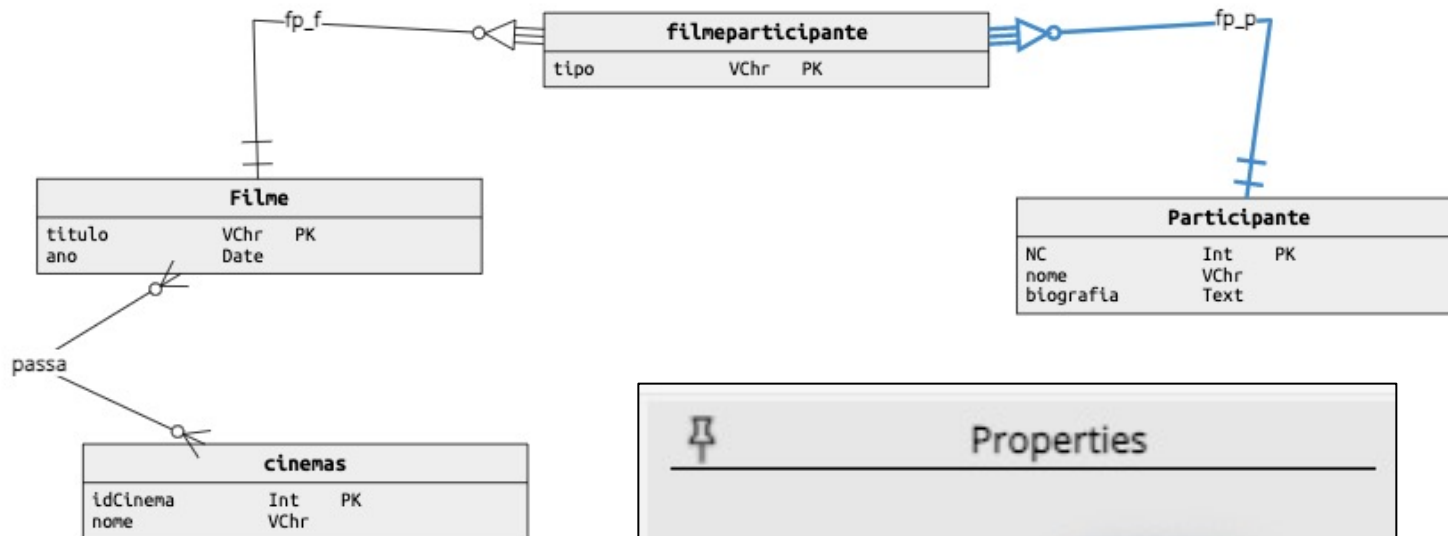
Fields

Field Name	Field Type	Default Value	Length	Primary Key	Not NULL	Unique	Actions
titulo	Varchar		100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	↑↓ X
ano	Date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	↑↓ X

Check Constraints



Onda - relacionamentos



Properties

Relation name: fp_p

filme participante Cardinality: 0..1
1..1
✓ 0..n
1..n

filme participante is weak: ☒

Participante Cardinality: 1..1

Participante is weak: ☐

Properties

Relation name: fp_p

filme participante Cardinality: 0..1
1..1
✓ 0..n
1..n

filme participante is weak: ☒

Participante Cardinality: 1..1

Participante is weak: ☐

A department has employees

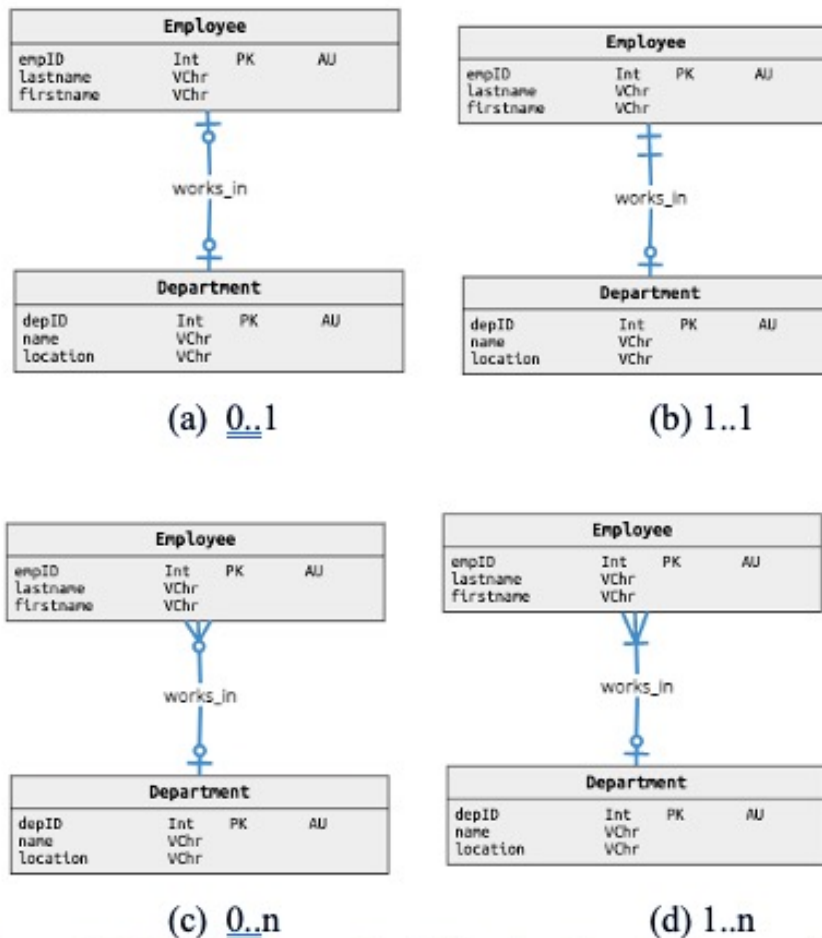
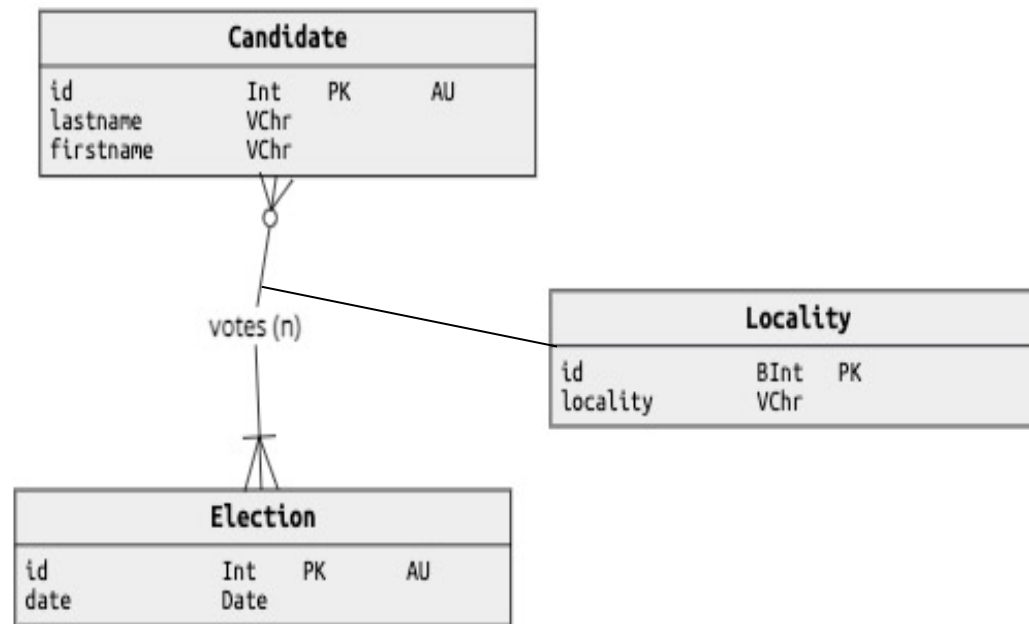


Figure 98. Possible cardinalities for Employee works_in

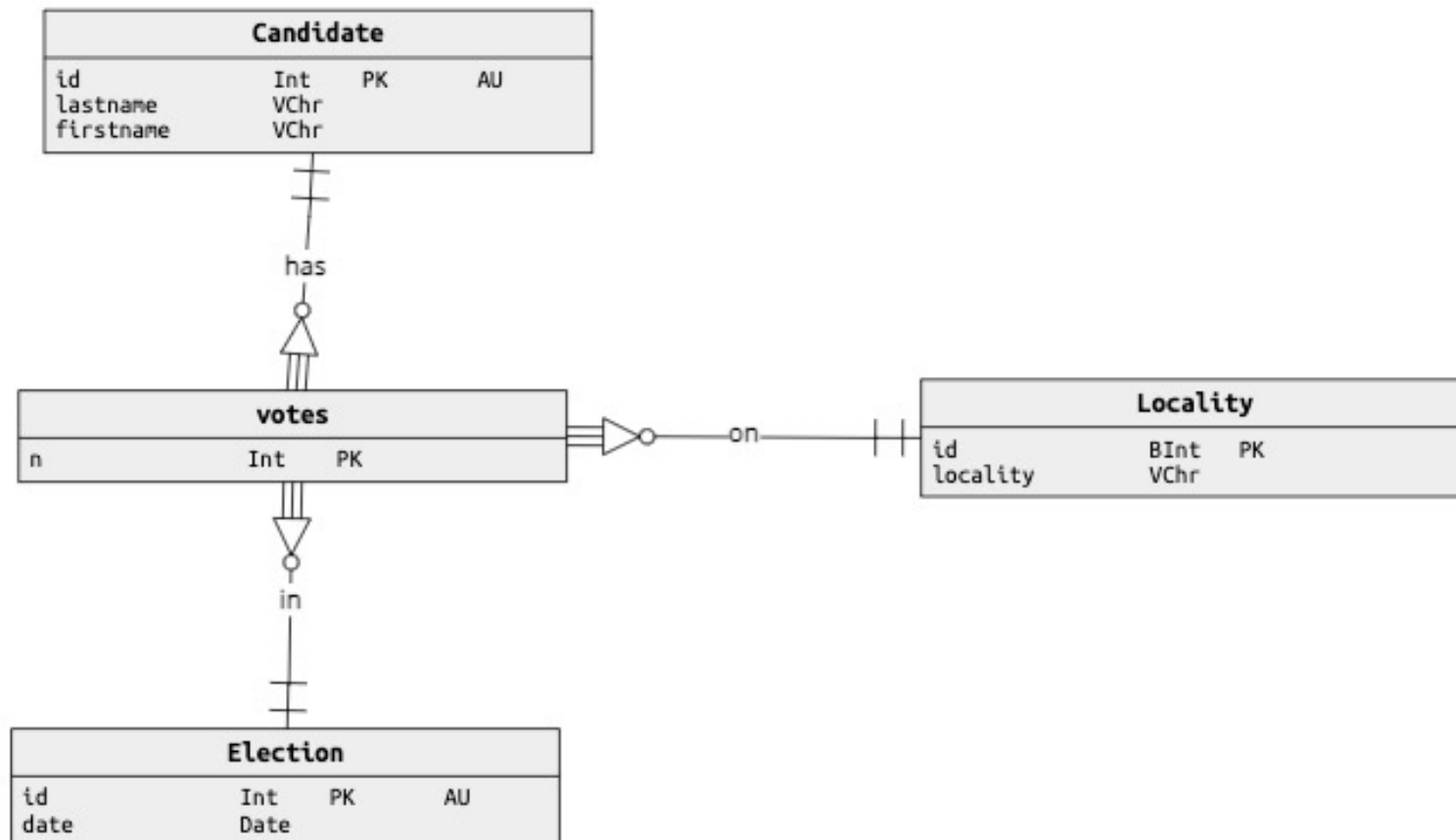
Nature of relationships

- These are binary relationships...
... Could there be n-ary relationships ???

How should I represent ternary relationships?



... through relationship entities and binary relationships

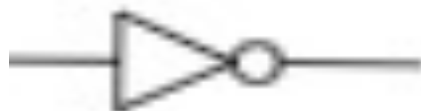
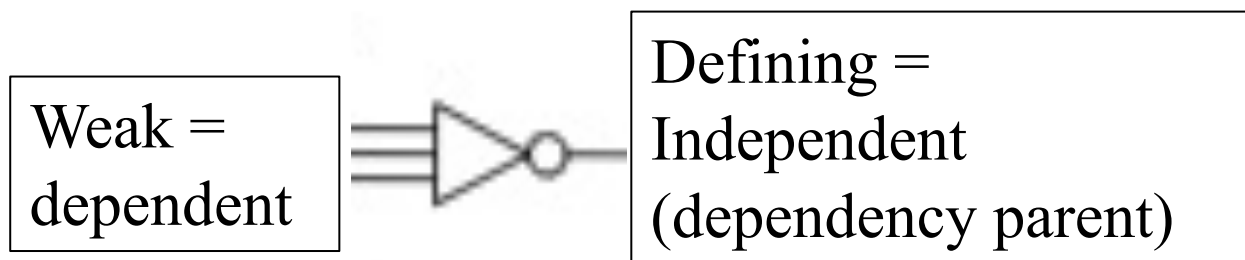


Weak/dependent entities, weak/dependent relationships

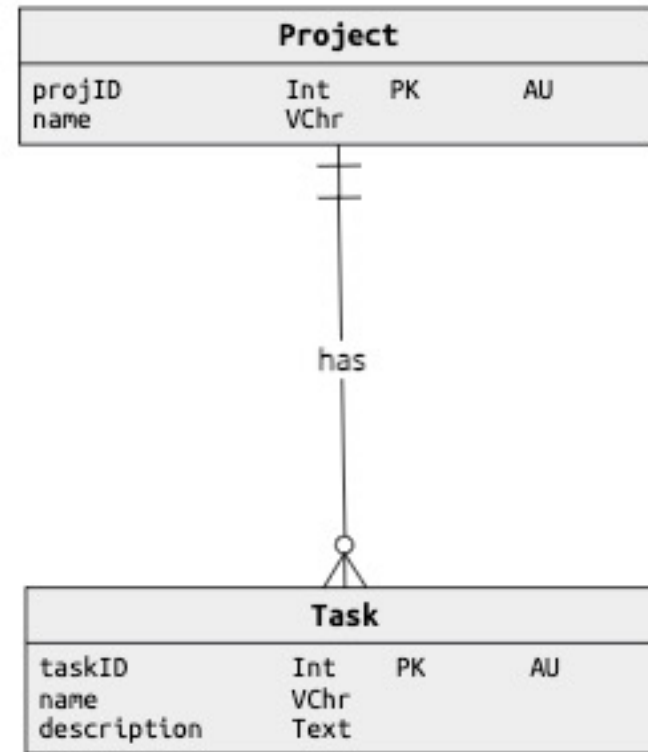
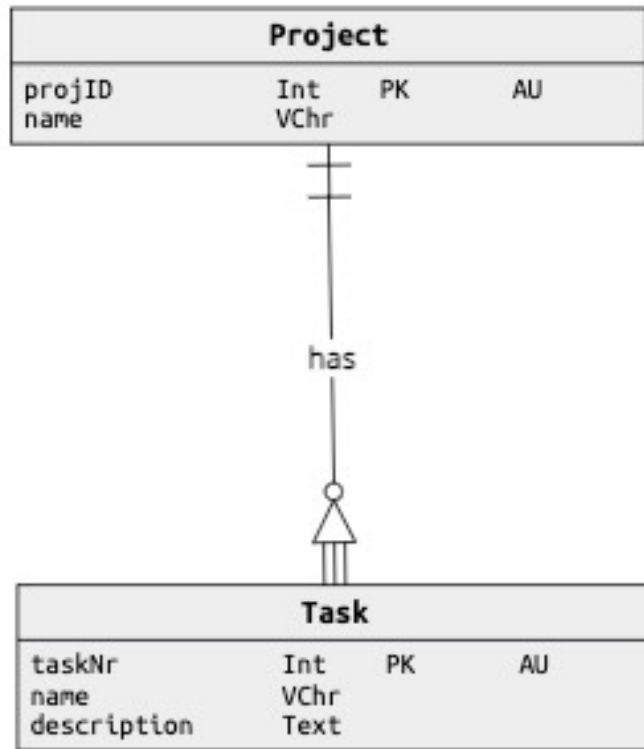
- means that the entity participating in the relationship in the side of the base of the triangle in the symbol is **weak** =

... = its identification depends on the entity on the other vertex of the triangle.

... = it exists only as a dependent entity

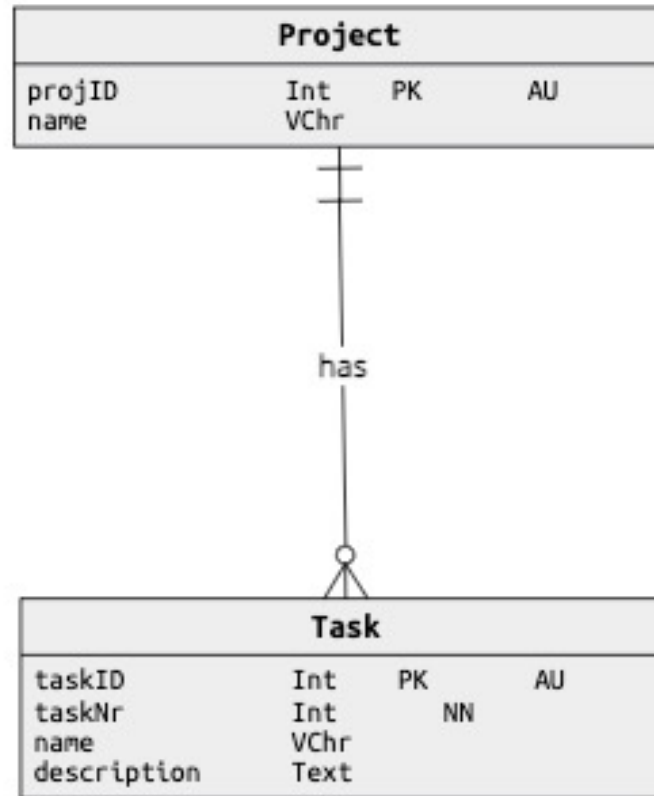


Task can be defined as weak / dependent on Project ...



Note what the primary keys are

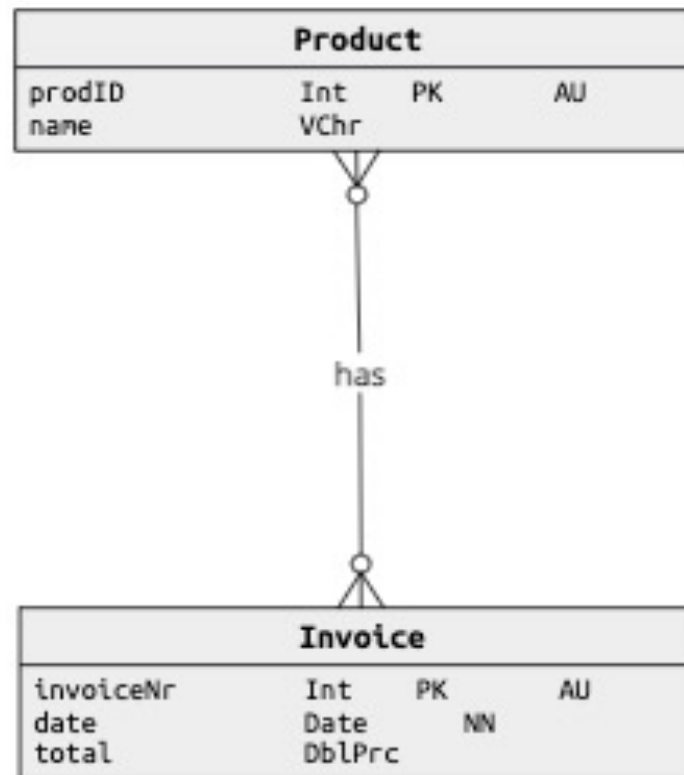
Of course there is a third choice...(both)



Note what the primary keys are

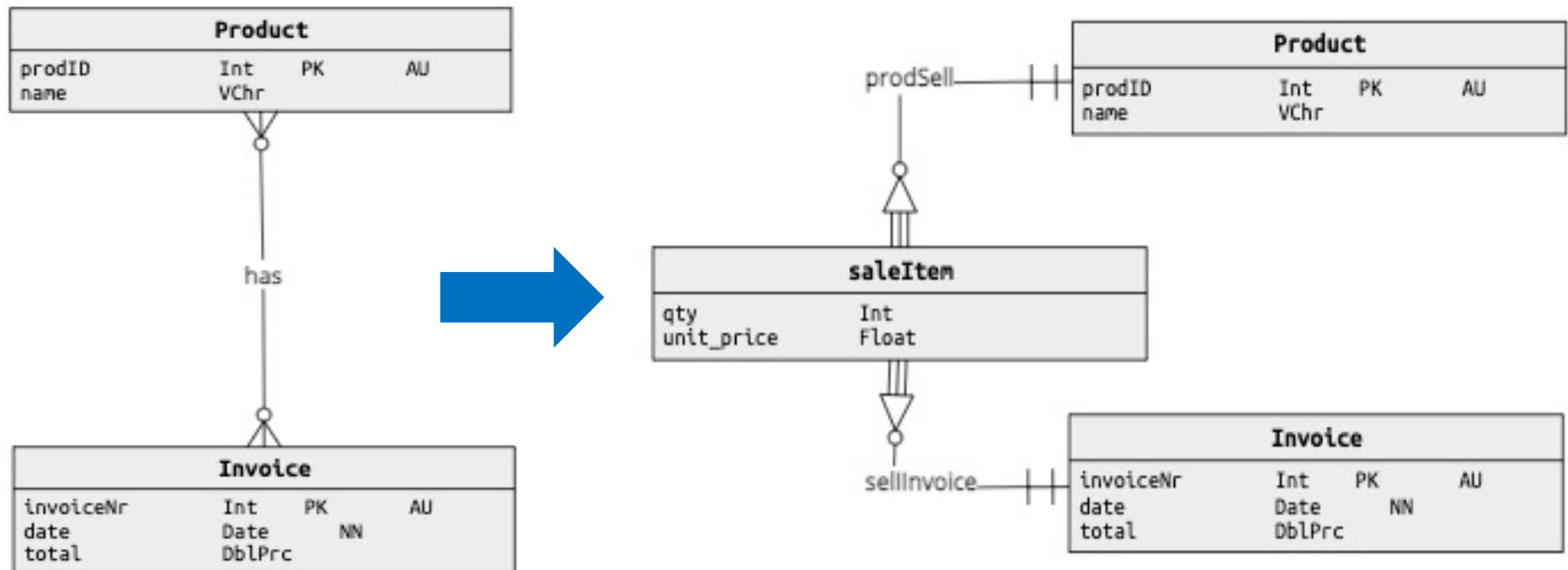
Many:many => weak rel entity

- A two entity many-many relationship ...



Many:many => weak rel entity

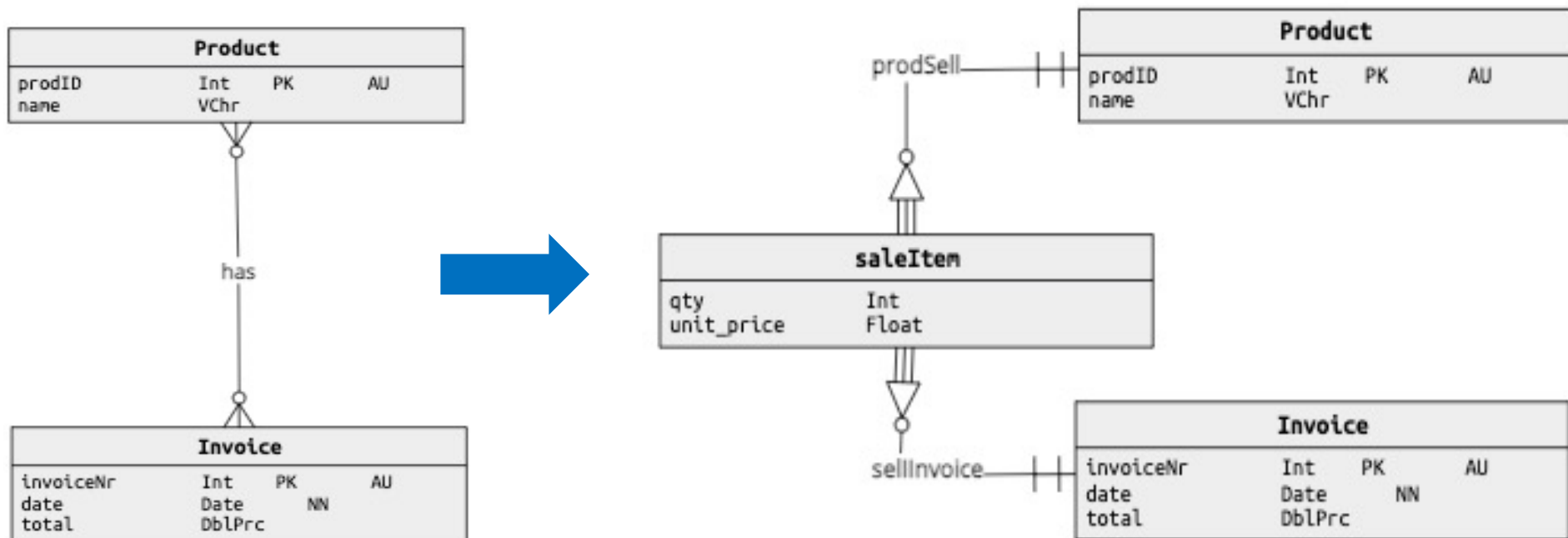
- **A two entity many-many relationship ...** is equivalent to a **three entity** and **two relationships** model



Note the Primary keys in each case ...

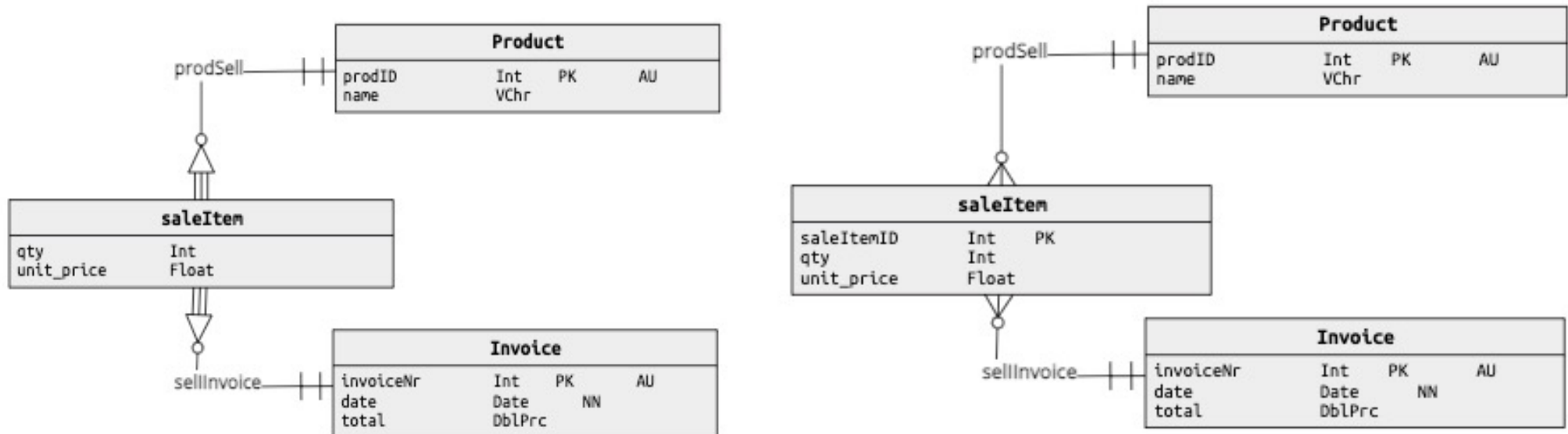
Many:many => weak rel entity

- A two entity many-many relationship is equivalent to a **three entity** and **two relationships** model
- The third is a **relationship entity** (saleItem in the example) with two weak relationships linking to each of the original entities



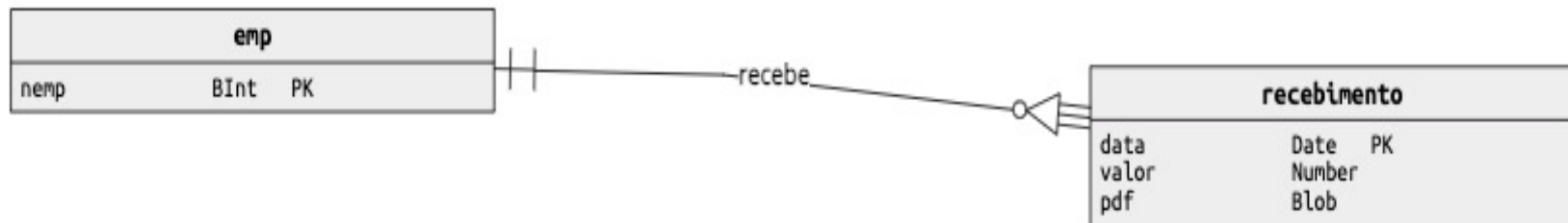
... and this is **only partly** equivalent to two 1:many relationships...

- What changes in terms of primary keys??? Why?

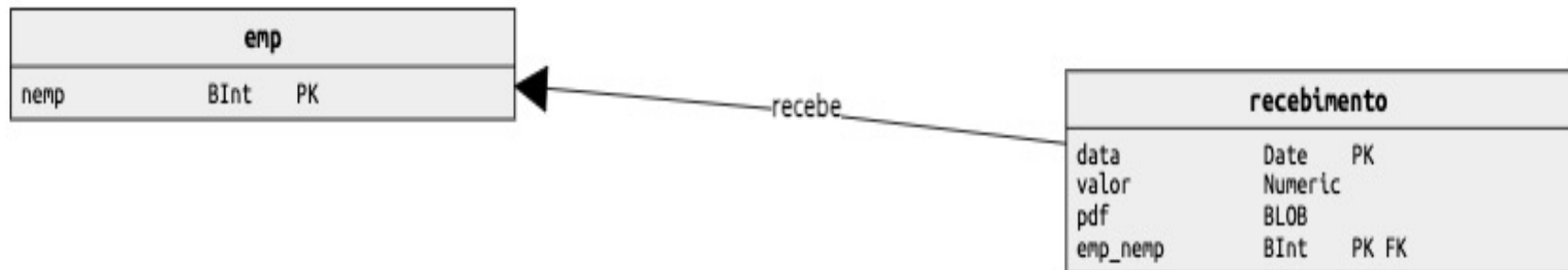


Another example ...

- Entidade relacionamento ou conceptual

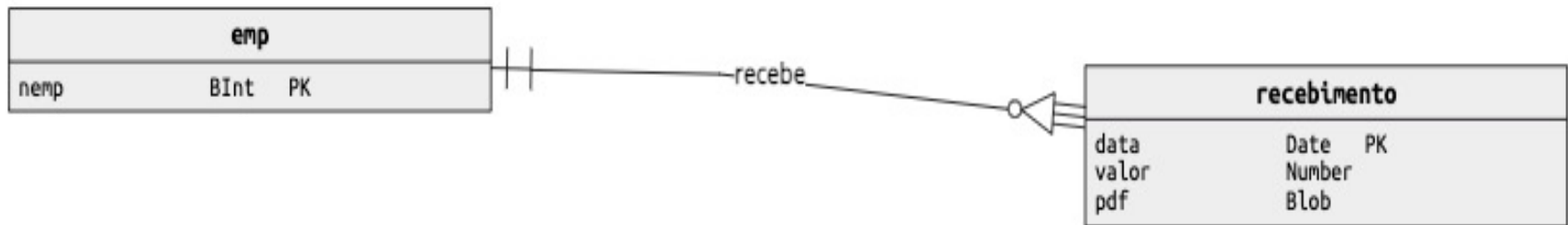


- Tabelas ou diagrama físico



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.



Entidade identificadora

Relacionamento identificador
(*dependent no Power Designer*)

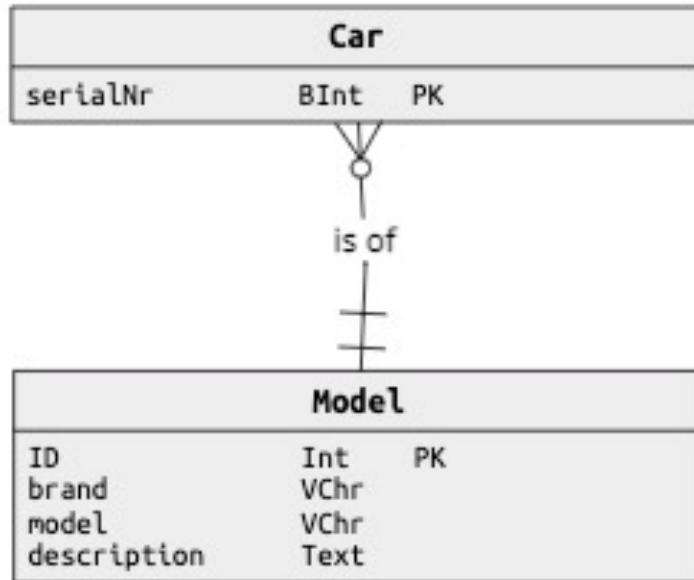
Entidade fraca

Chave parcial

Atributo ou conjunto de atributos que permitem identificar univocamente uma ocorrência da entidade fraca para uma dada ocorrência da entidade identificadora (e.g., Nome é chave parcial de Dependente)

Confusing the meaning of entities: what is a product?

A specific car can be a product...
(one license plate)...



Product		
barcode	BInt	PK
name	VChr	
category	VChr	
unit price	Float	
qty in stock	Int	

...but also often a "product" is a type of product with many instances

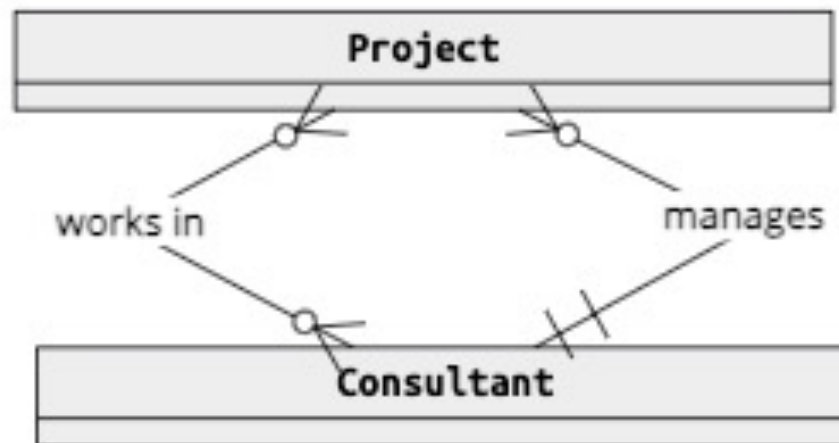
C: make sure you know what your "product means", and ...

e.g. please do not represent each specific screw individually in a database

Multiple relationships between two entities are the normal

- These two relationships are needed ...

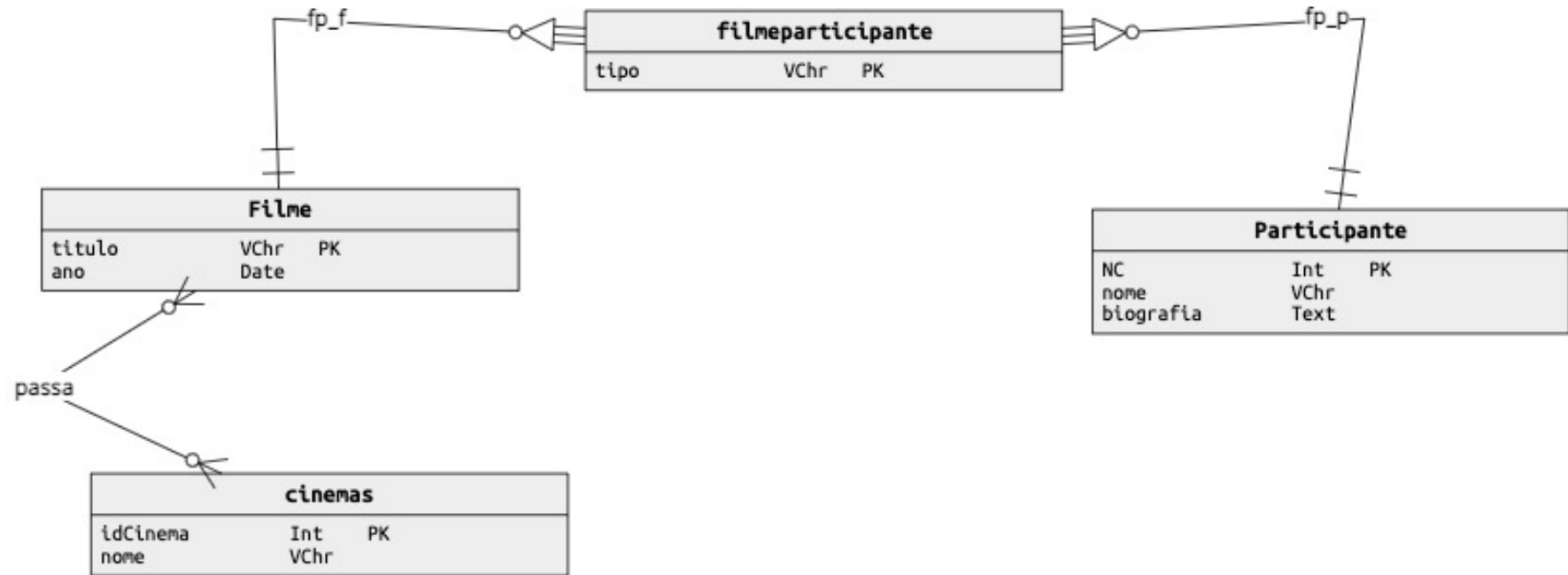
... Sometimes its even a lot more ...



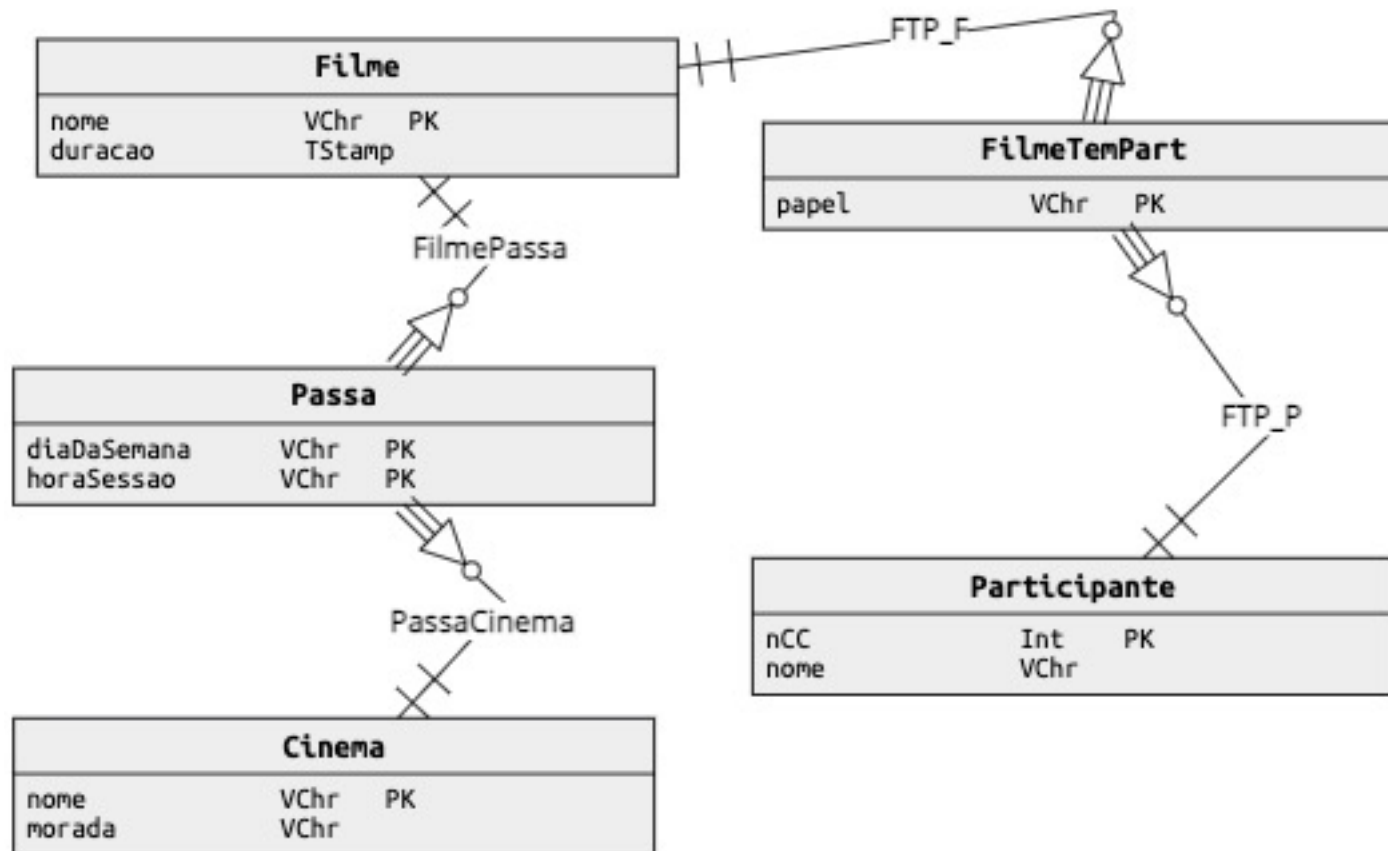
ER no ONDA: Filmes

- Um filme passa em cinemas
- O filme tem participantes que tem um dado papel (actor, realizador, director de fotografia, etc)

ER no Onda



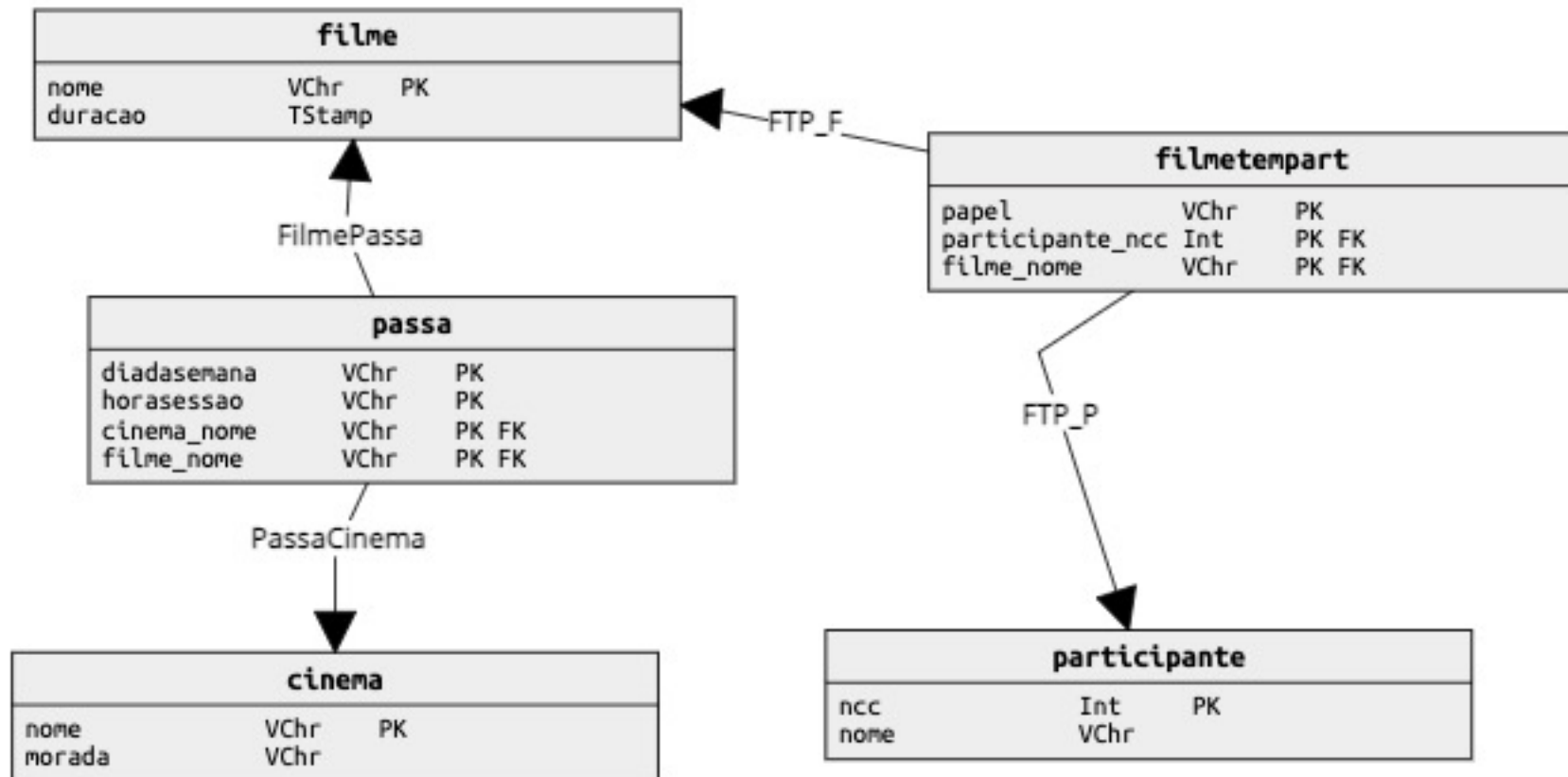
Conceptual diagram...



Physical diagram

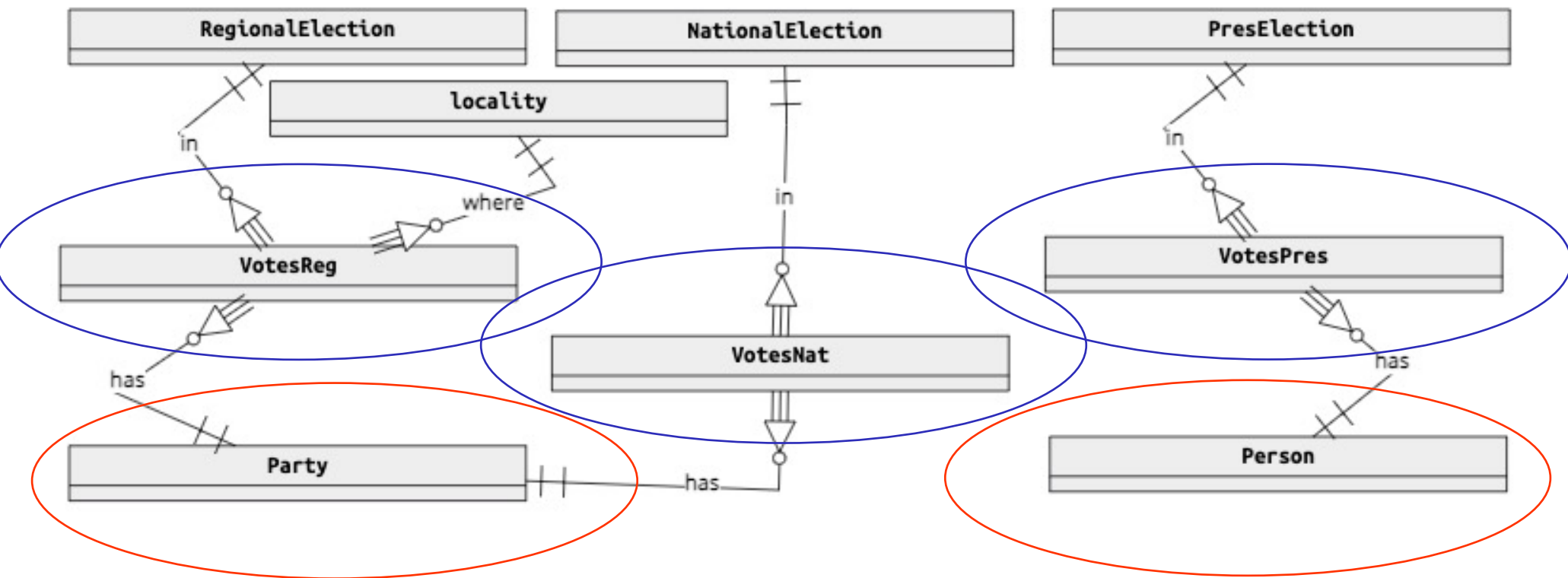
Obtained automatically from conceptual diagram...

... Based on fixed set of rules...



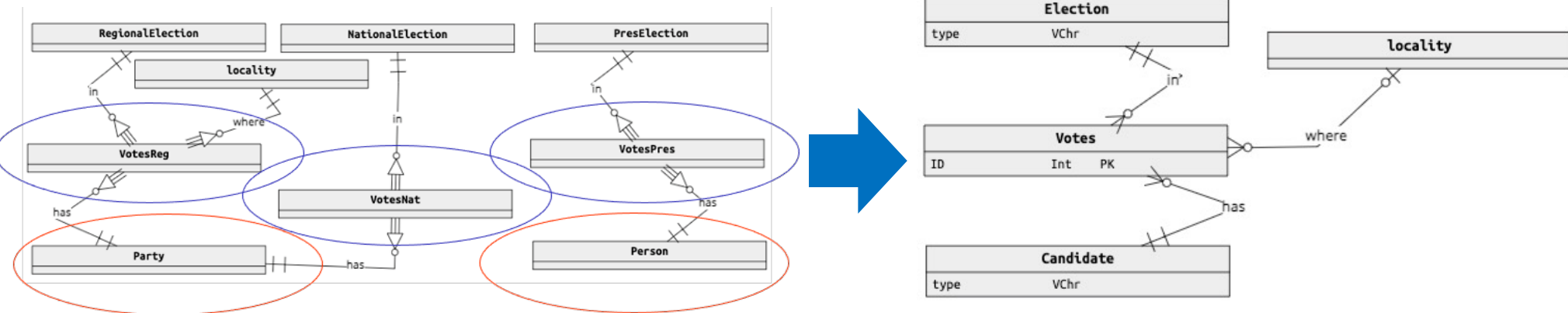
Reusing and Inheritance: WHY ITS SO IMPORTANT

So many “votes”... WHY? Complex...



Reusing and Inheritance: how ...

- Party and person are CANDIDATES...
- No matter what election it is... It is still an ELECTION...
- ... And everybody VOTES in an ELECTION



Especialização e herança

Especialização:

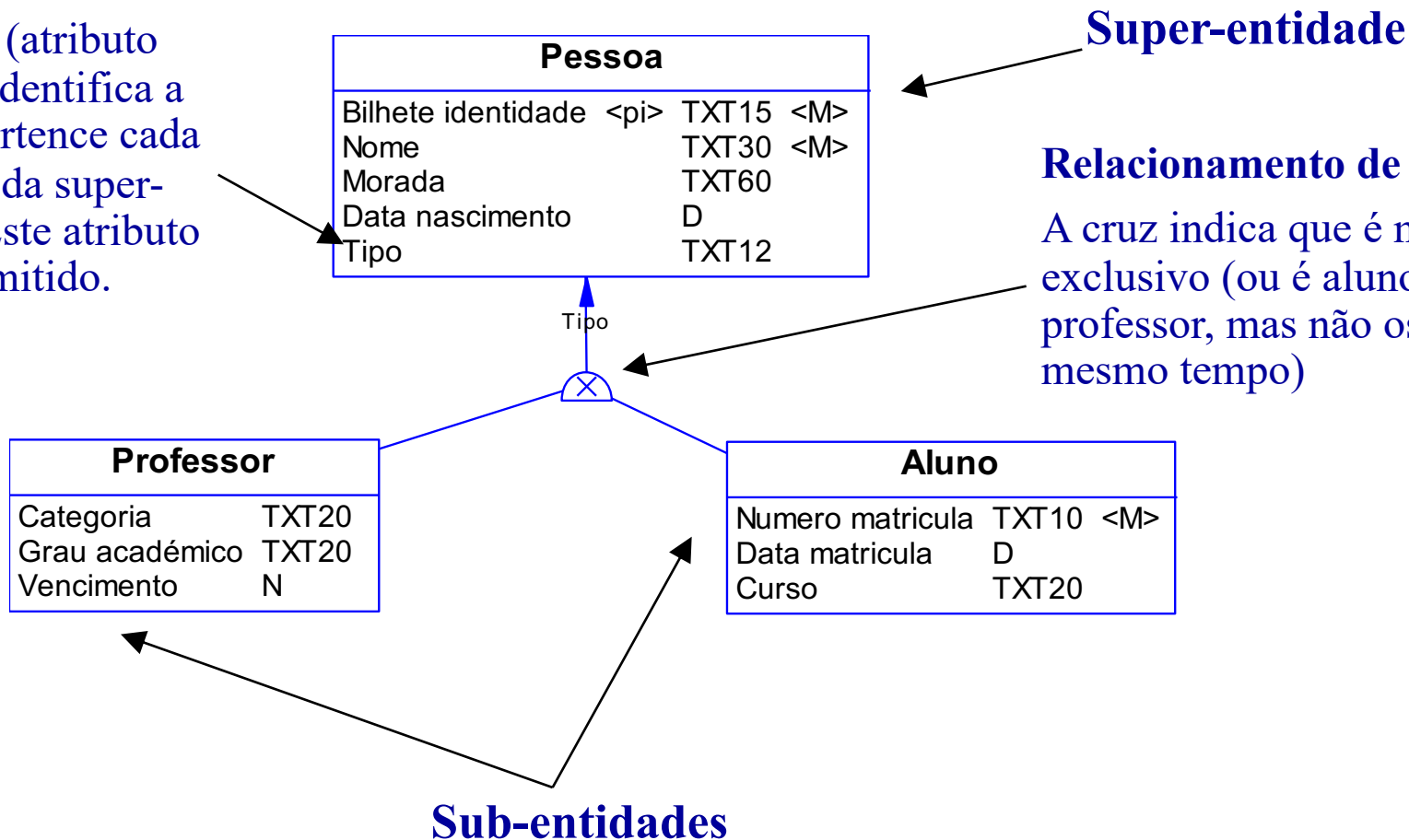
Processo de definição de **sub-entidades** de uma entidade. Essa entidade, a partir da qual se definem sub-entidades, designa-se por **super-entidade**.

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

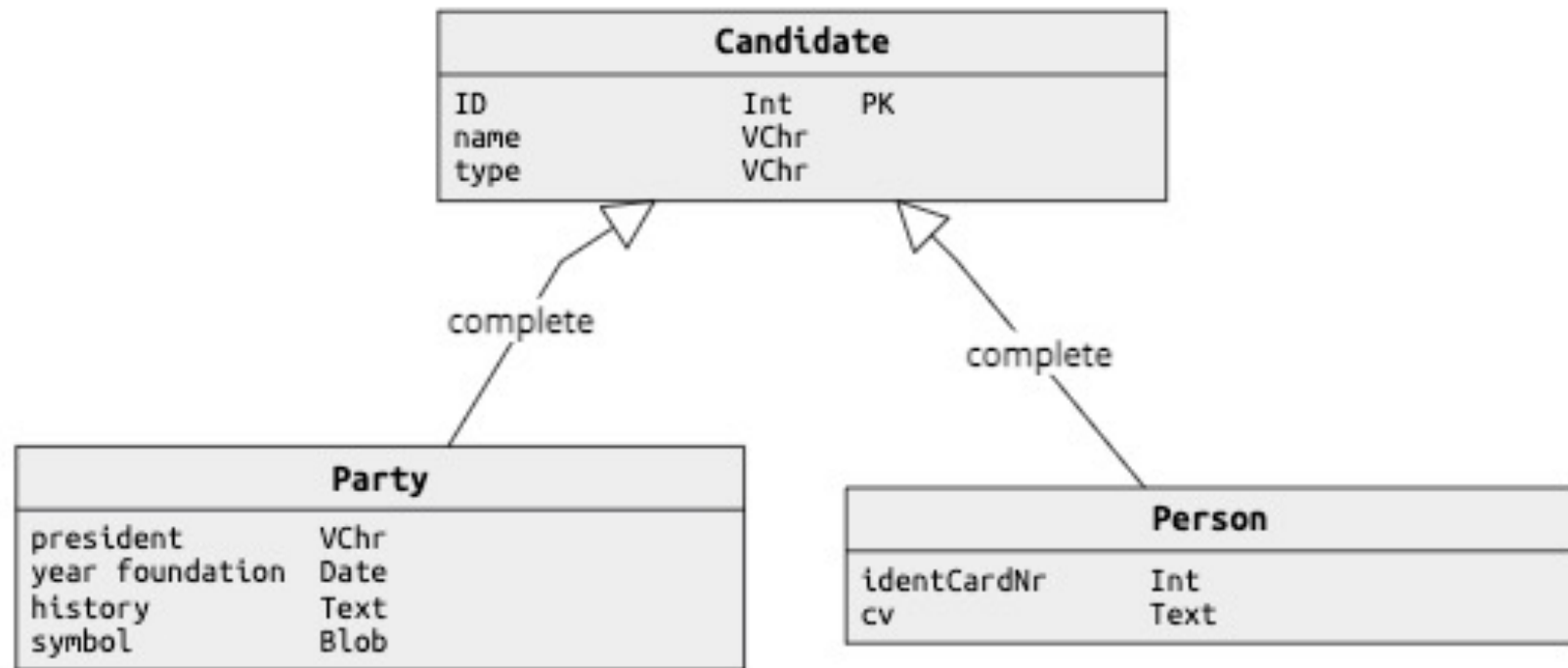
As sub-entidades herdam os atributos e os relacionamentos da super-entidade.

Exemplo de especialização

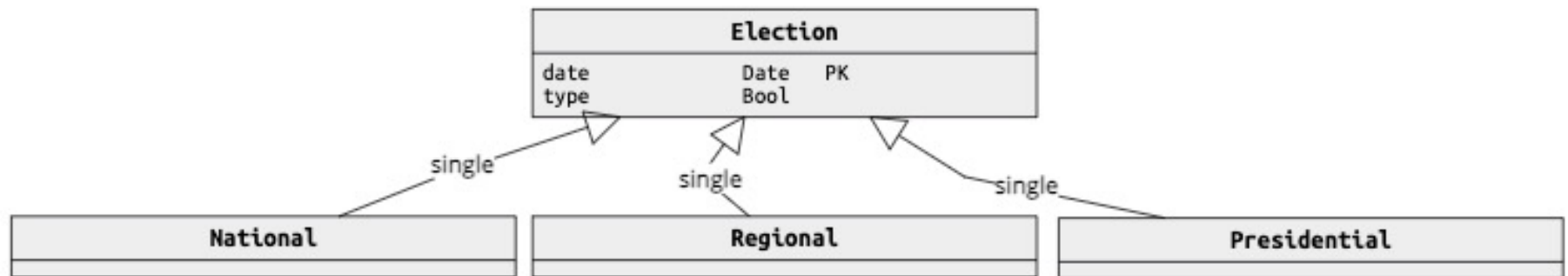
Predicado (atributo Tipo) que identifica a que tipo pertence cada ocorrência da super-entidade. Este atributo pode ser omitido.



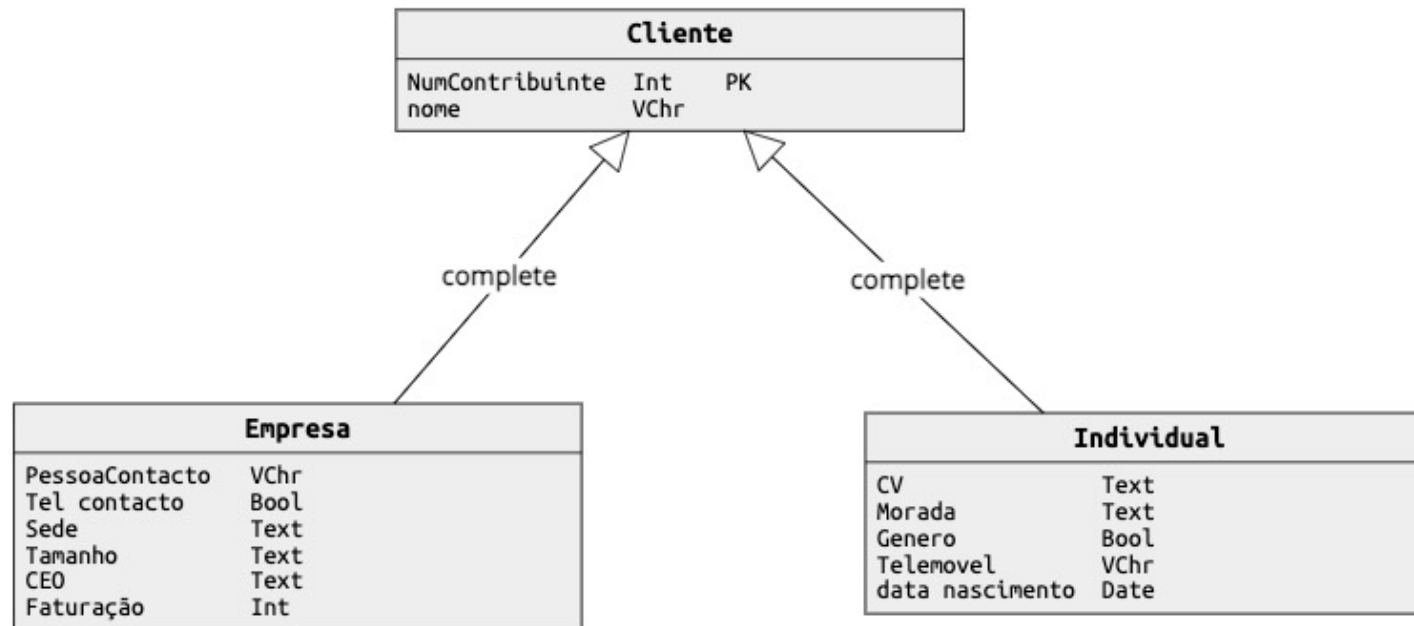
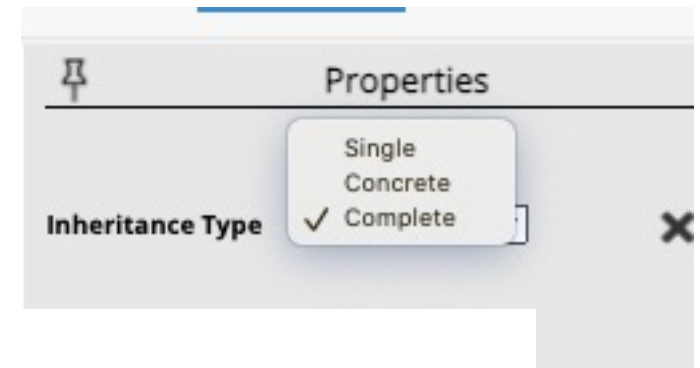
Election candidates



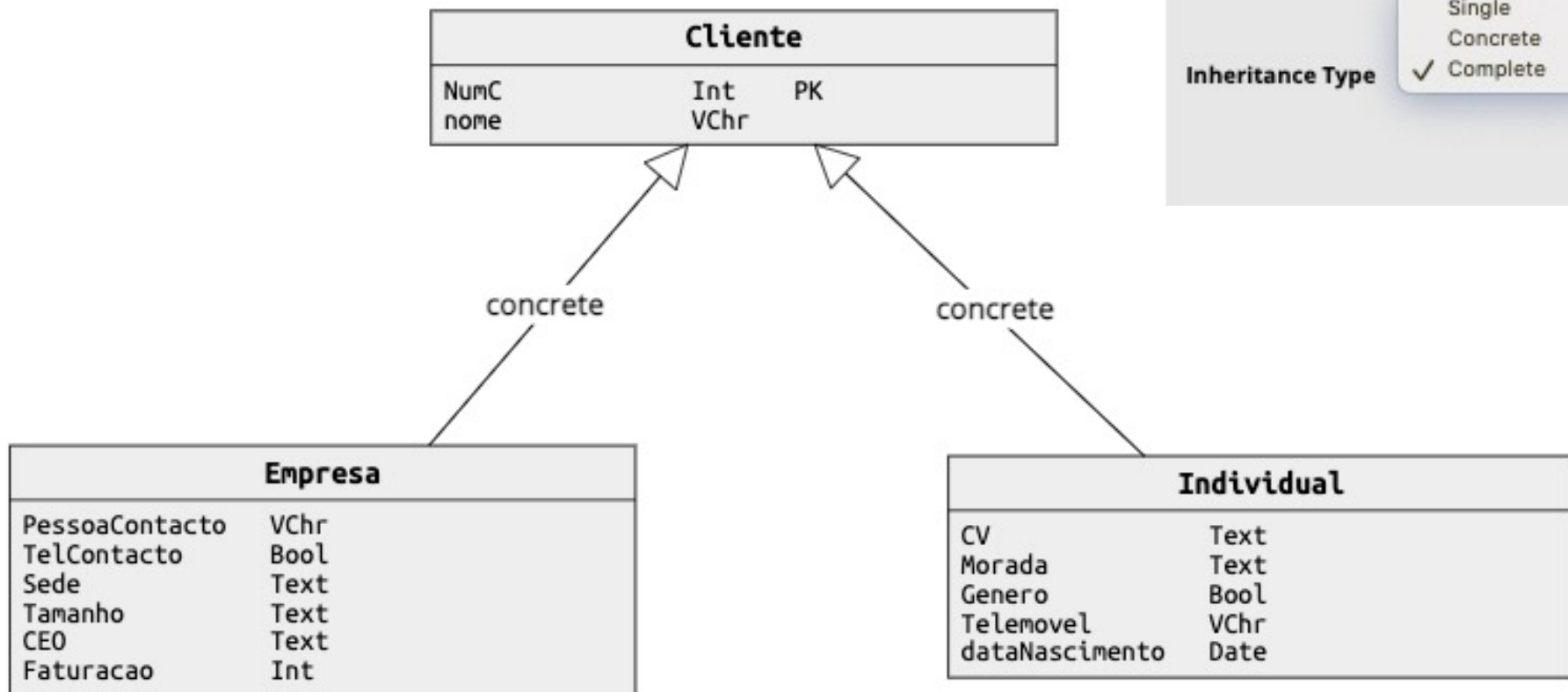
Election



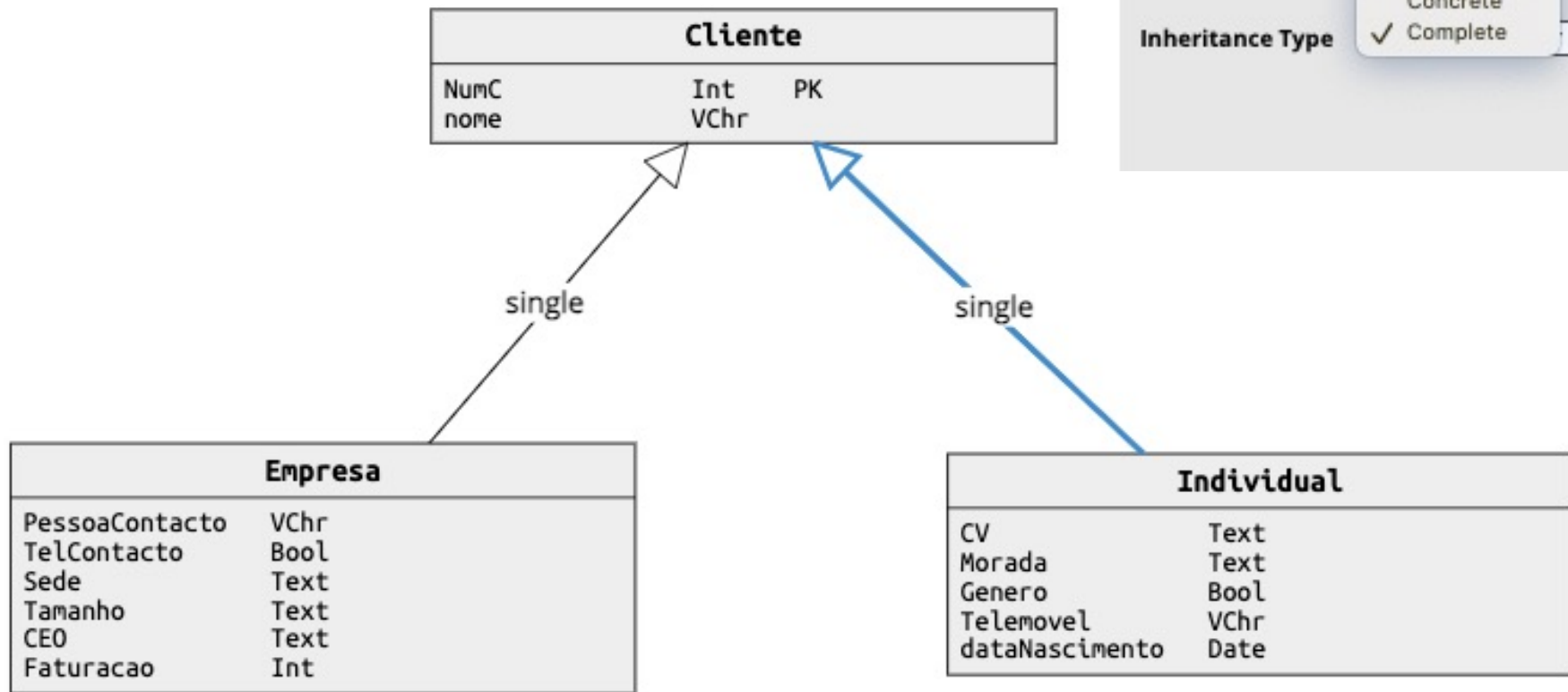
Complete = all tables



Concrete = 2 tables



A single table



-
- (a) the “complete” type means that there will be one table for the parent and one table for each of the children, and each table retains the attributes of the corresponding entity. Each children will inherit the parent primary identifier as its primary key, and it will also be a foreign key to the parent;
- (b) type “single” means that there will be no children, there will only be a single table (the parent table). In that case the parent will have all attributes from all children besides its own attributes when it is transformed into a table. If there are any relationships of specific children with other entities, then they become relationships of the parent with those entities;
- (c) type “concrete” means that there will be no parent, only children tables. Each children will in this case inherit all attributes and relationships from the parent and keep their own as well.
-

An example ER design ...

A database of projects...

Get someone to try to do it

He interviews me ...

He draws on the blackboard ...

NEXT GUY, random nr generator...

Solution (only after the interactive procedure)

Lets see the database of projects

Exemplo: a database of projects

- Uma empresa faz projectos para clientes
- A empresa tem consultores
- Um consultor pode ser tb “Project Manager”, pessoa que negocia e cria os projectos
- Os projectos têm tarefas, e os consultores podem participar nas tarefas de projectos
- Para cada tarefa e cada consultor que trabalhou nessa tarefa, também gostaríamos de saber quando ele começou e terminou o seu trabalho na tarefa, e o esforço (medido como homens-mês), que irá definir o seu pagamento relativo a essa tarefa.

Exemplo: a database of projects

- Quando um projeto é planeado, as suas tarefas devem ser inseridas no sistema. Uma tarefa é identificada por um número de ordem (começando em 1), e um título. Por exemplo, a tarefa 1 do projeto poderia ser a configuração de algumas infra-estruturas para o resto do projecto, chamando-se "Setup Infra-estrutura".
- Além do identificador e o título, as tarefas incluem um texto de descrição, e um texto que descreve as necessidades da tarefa.
- As tarefas devem ter um registo de datas planeadas e datas reais de início e fim, bem como orçamento previsto e real.
- As tarefas também registam as necessidades de recursos humanos em termos de unidades homem-mês (mm). Deve haver dois campos MM: planeadas e reais.

Versao 0 e cardinalidade



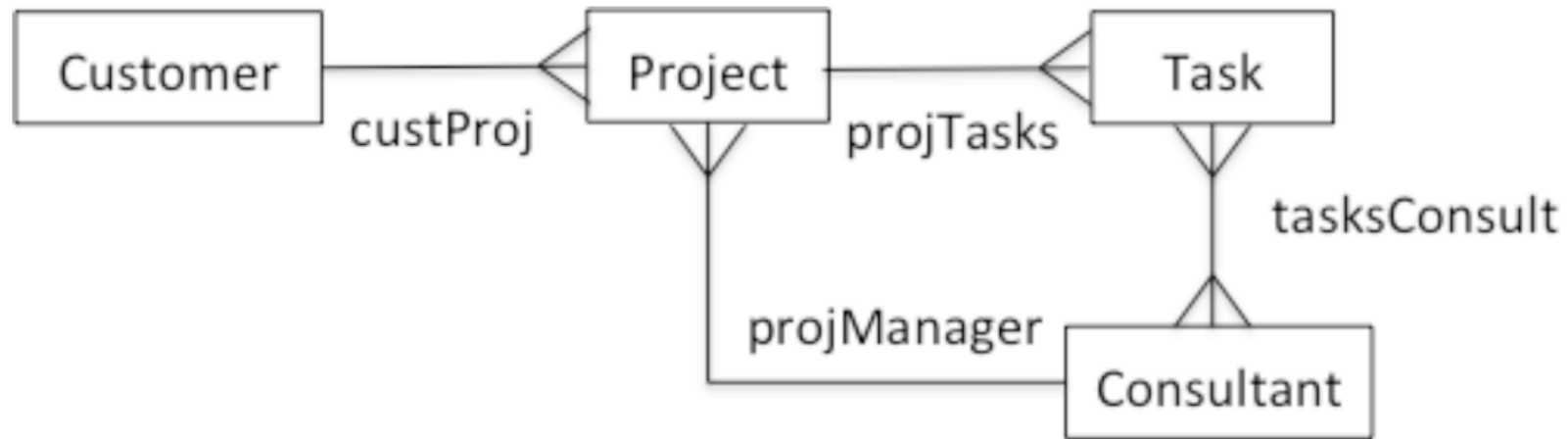
- A entidade Project tem muitas tarefas (Project 1:n Task)
- Note que, em alguns casos reais uma tarefa poderia ser associada a mais do que um projecto. Não é o caso neste exemplo...

Versao 0 e cardinalidade



- A entidade Task pode ter participação de muitos Consultores, e um Consultor pode participar em muitas tarefas do mesmo ou de diferentes projectos (Task n:n Consultant);

Versao 0 e cardinalidade



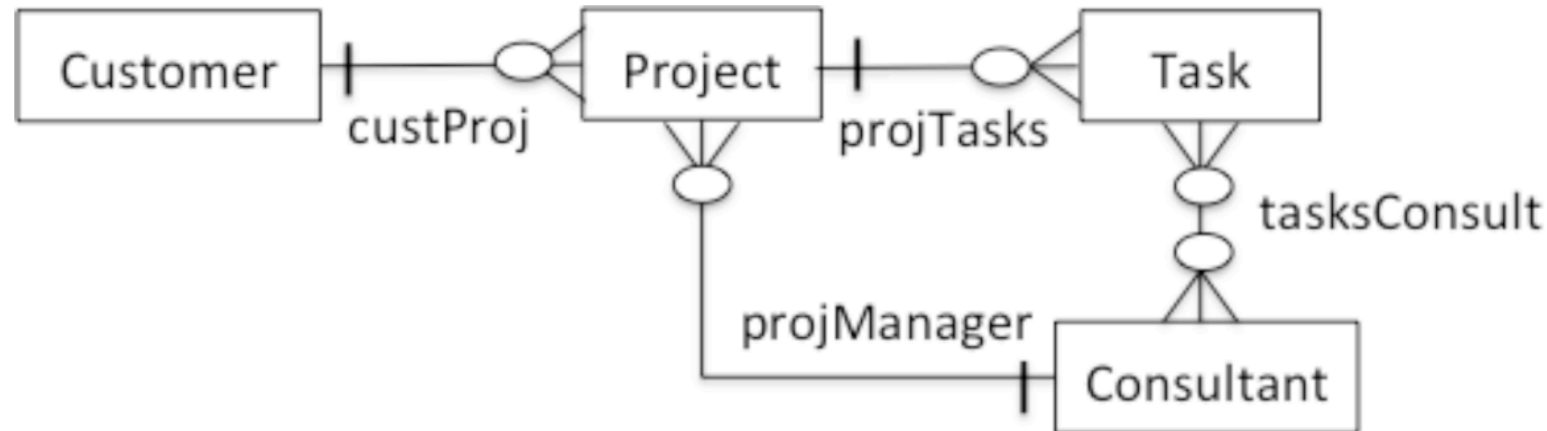
- A entidade Projeto tem um Cliente (Customer 1 : n Project), uma vez que consideramos que um projeto tem um único cliente) .
- Observe que , em alguns casos do mundo real , seria possível que um projeto seja adjudicado a vários clientes, portanto, é útil fazer perguntas . **Quando nada é dito , devemos assumir a semântica mais comum e , em seguida, verificar com os clientes do sistema de informação;**
- Um Consultor específico é o Project Manager (Consultor 1 : n Project), uma vez que consideramos que um projeto tem um único gestor de projecto) .

Participação de relacionamentos



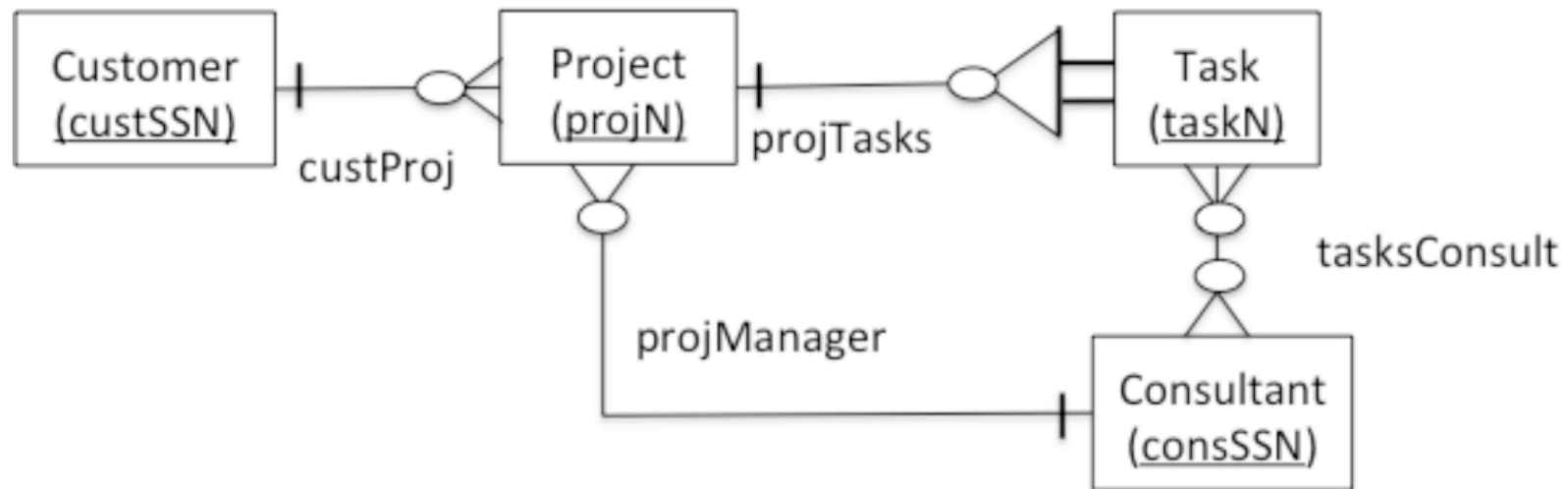
- Um projeto pode ou não ter tarefas associadas com ele. Quando um projeto é registado inicialmente , podemos inserir o seu título , descrição, cliente e muitas outras informações , sem descrever as suas tarefas imediatamente . Mais tarde, vamos inserir as tarefas, planear o projeto e o seu orçamento . Isso explica a participação opcional da entidade projeto no relacionamento com Task.
- Uma tarefa registada na base de dados deve estar associada a um projecto específico. Isso explica a participação obrigatória de tarefas.

Participação de relacionamentos



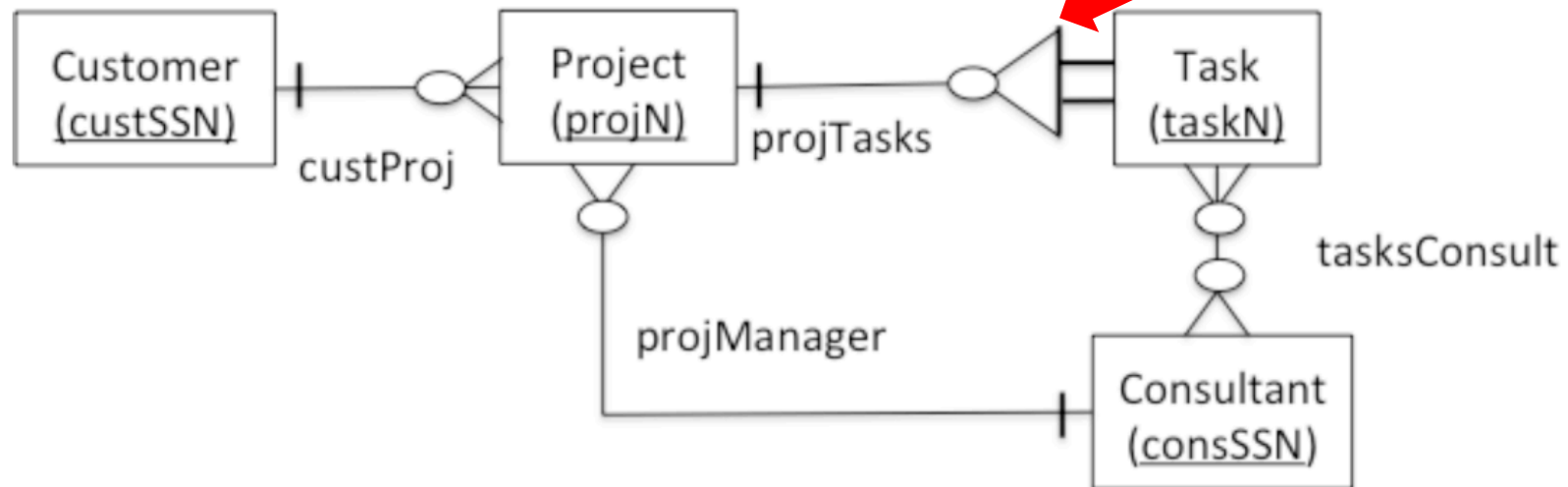
- Os clientes podem ser adicionados ao sistema de informação sem serem associados com qualquer projecto , embora devam ser mais tarde associados com alguns projecto (s) .
- Os projetos devem ser sempre associados com algum cliente , que também é responsável pelo pagamento do projeto.
- Uma tarefa pode não ter nenhum consultor, e um consultor pode ser registado sem a necessidade de associar a ele / ela para qualquer tarefa.
- Um consultor pode também não estar associado a qualquer projeto quando é inserido pela primeira vez, mas um projeto tem sempre um gestor de projeto, uma vez que é o responsável também pela criação do projeto.

Identificadores



- customer social security number (custSSN), identifica clientes e consultores (consSSN);
- project number (projN) para projectos: ARTIFICIAL KEY (incrementing numbers)
- task number (taskN) para tasks: ARTIFICIAL and ORDINAL (incrementing numbers)

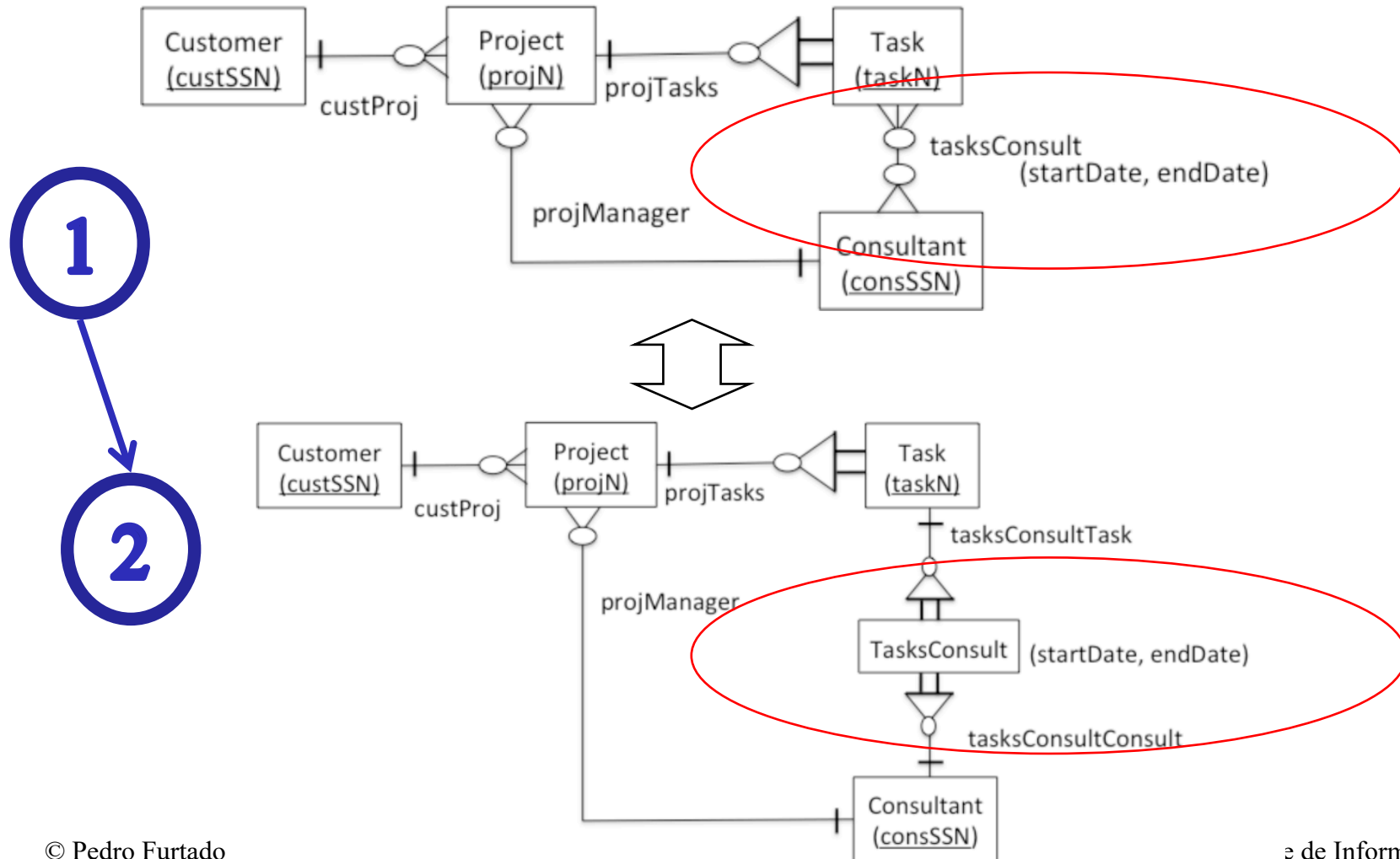
Identificadores - dependência



- task number (taskN) não é suficiente para identificar uma tarefa
 - Task 1 existe em P1 e em P2.
- **Relacionamento dependente** – a entidade dependente (Task) é parcialmente identificada pela entidade da qual depende. “**weak entity**” (entidade fraca).
- O identificador de Task será (taskN, projN), **mas projN é implícito no diagrama, não se põe.**

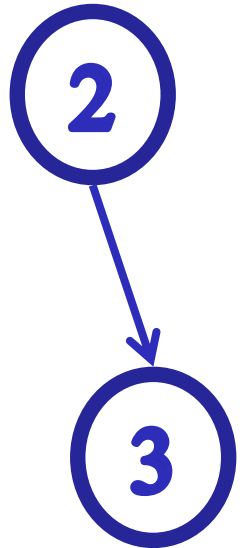
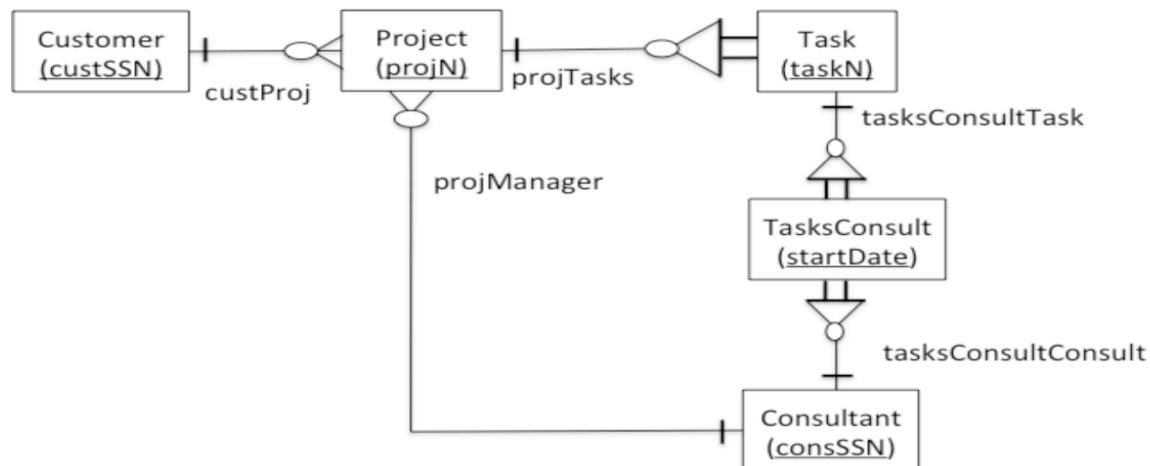
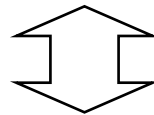
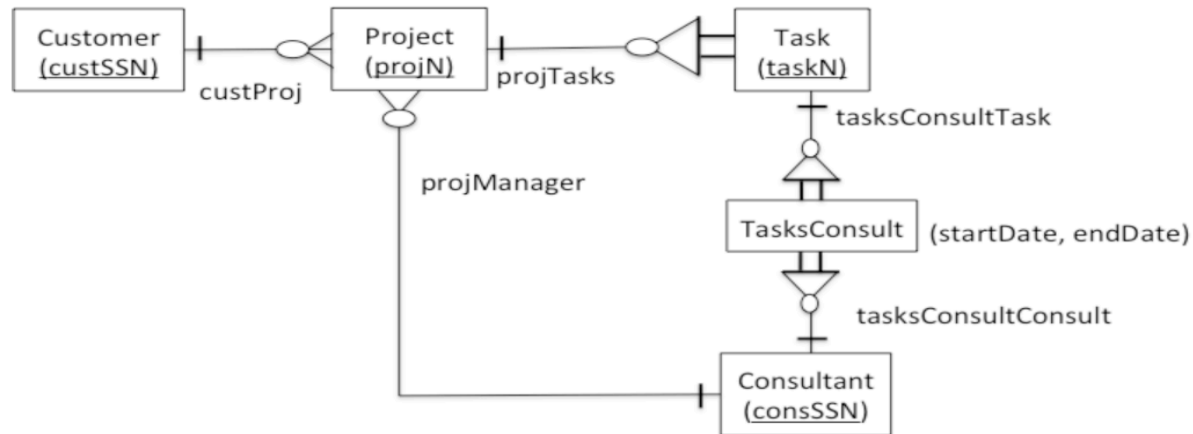
M:N com atributos

Datas de inicio e fim de participação de consultor em tarefa

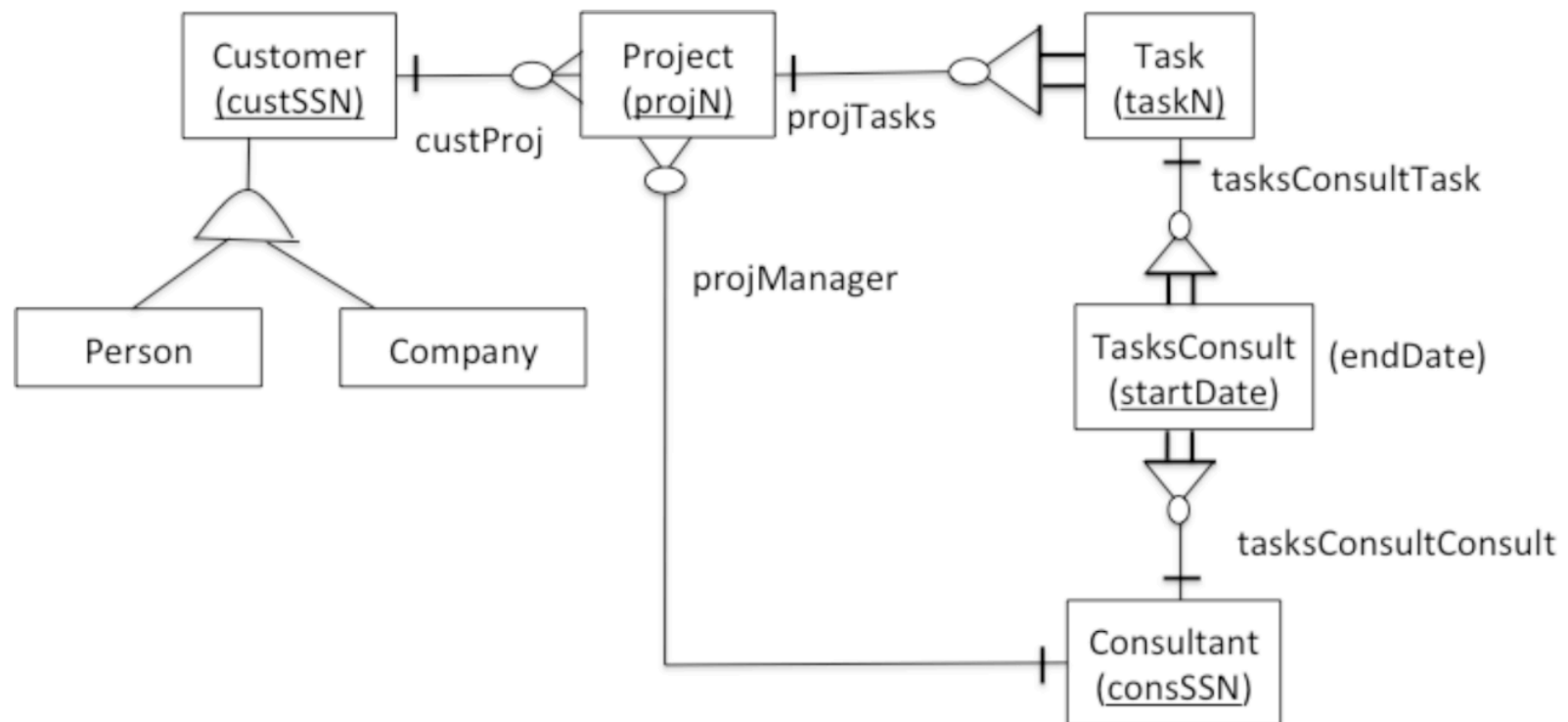


M:N com atributos

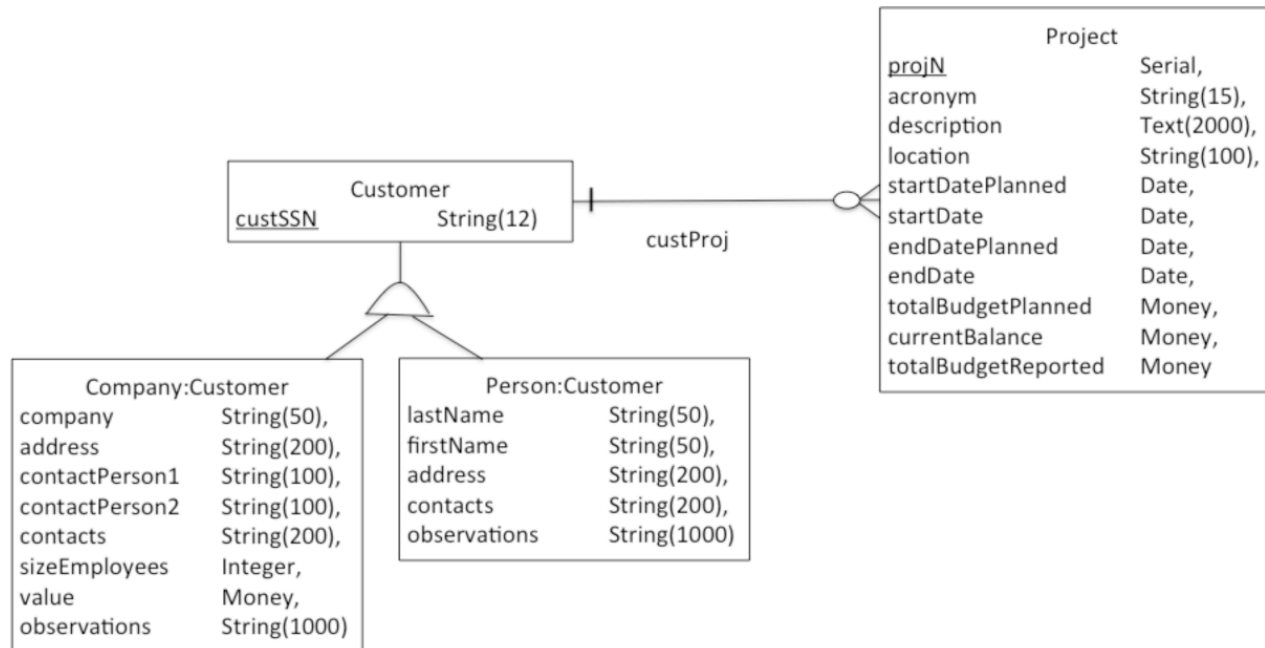
Identificador de TaskConsultant



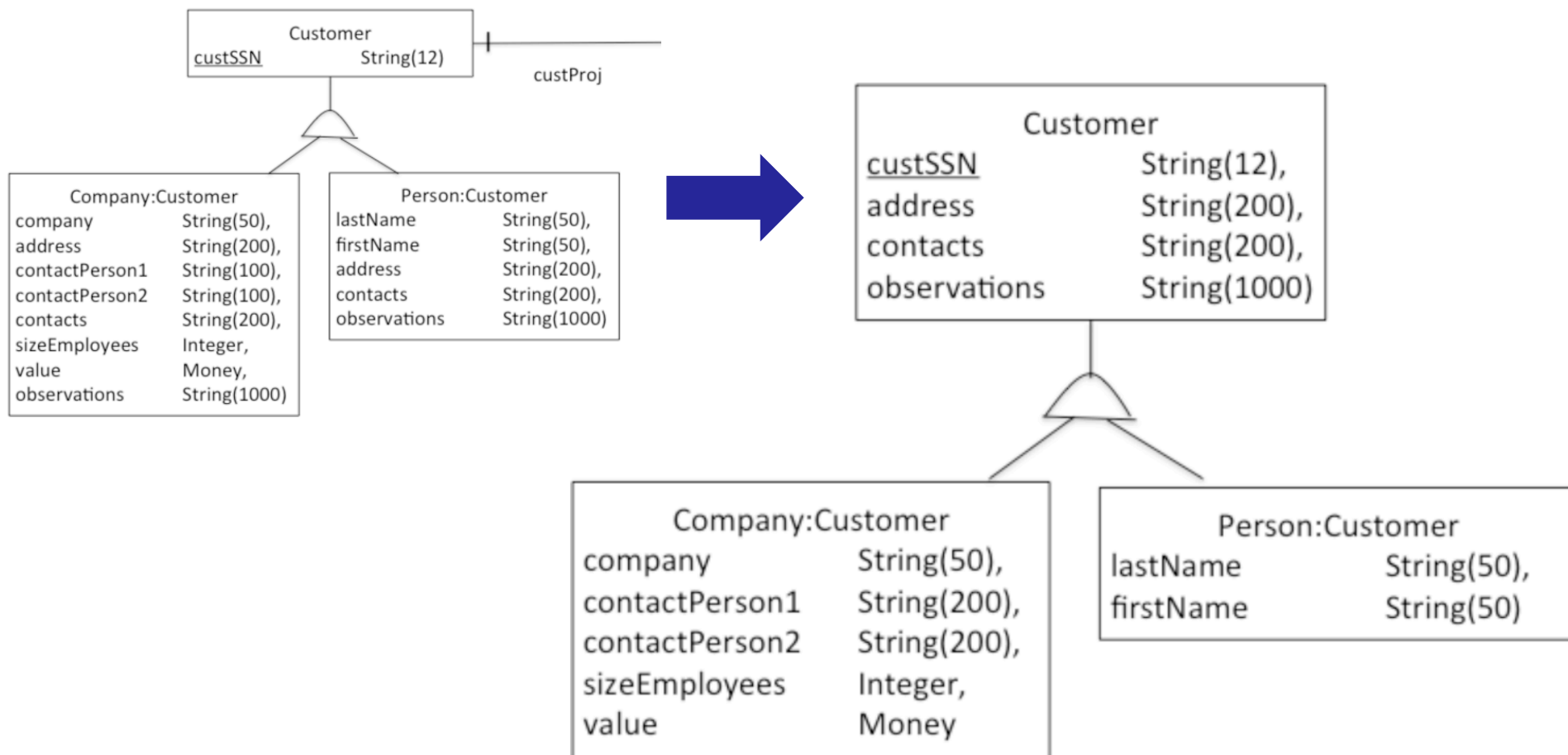
Herança Cliente



Attributos – customer, project



Atributos – Customer



Atributos – project, task, consultants

