

# Modelo ER

## ■ Tópicos

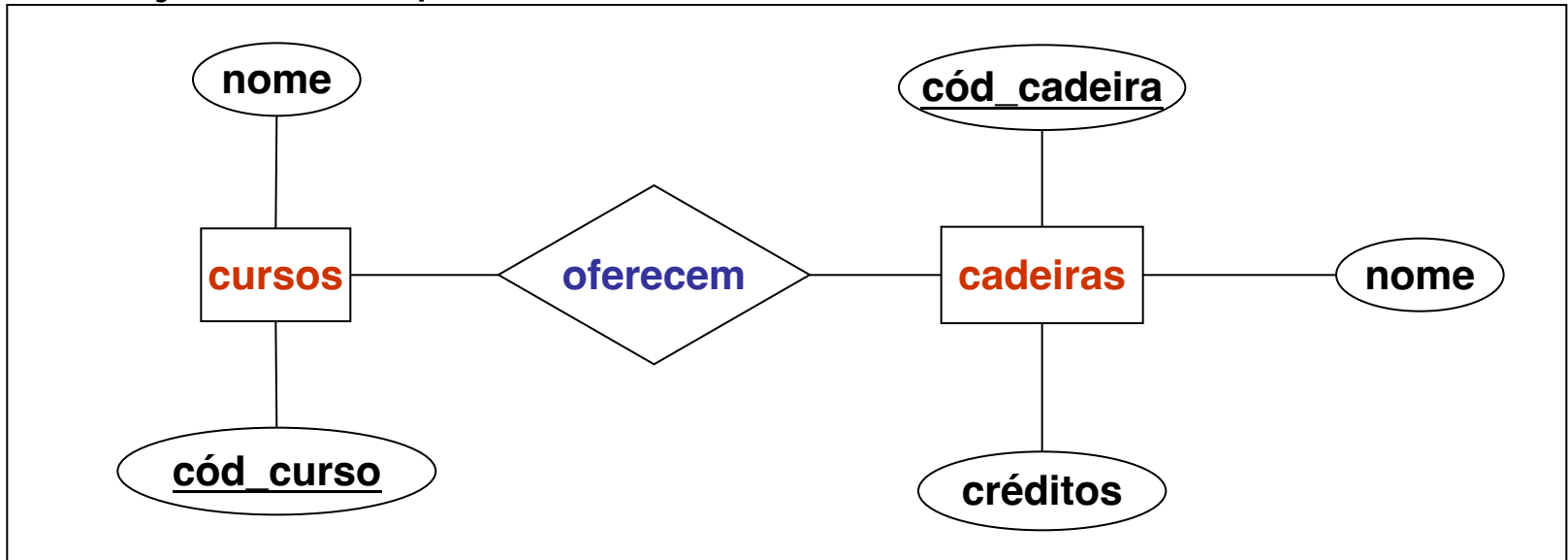
- ✧ O processo de desenho
- ✧ Conjuntos de entidades
- ✧ Conjuntos de relações
- ✧ Restrições de Mapeamento
- ✧ Chaves
- ✧ Diagrama ER
- ✧ Opções de desenho
- ✧ Extensões ao modelo ER
- ✧ Exemplo

## ■ Bibliografia

- ✧ Capítulo 7 do livro recomendado (6ª edição)
- ✧ Capítulo 6 do livro recomendado (7ª edição)
- ✧ Capítulo 2 do livro de Ramakrishnan et al.

# Diagramas ER (DER)

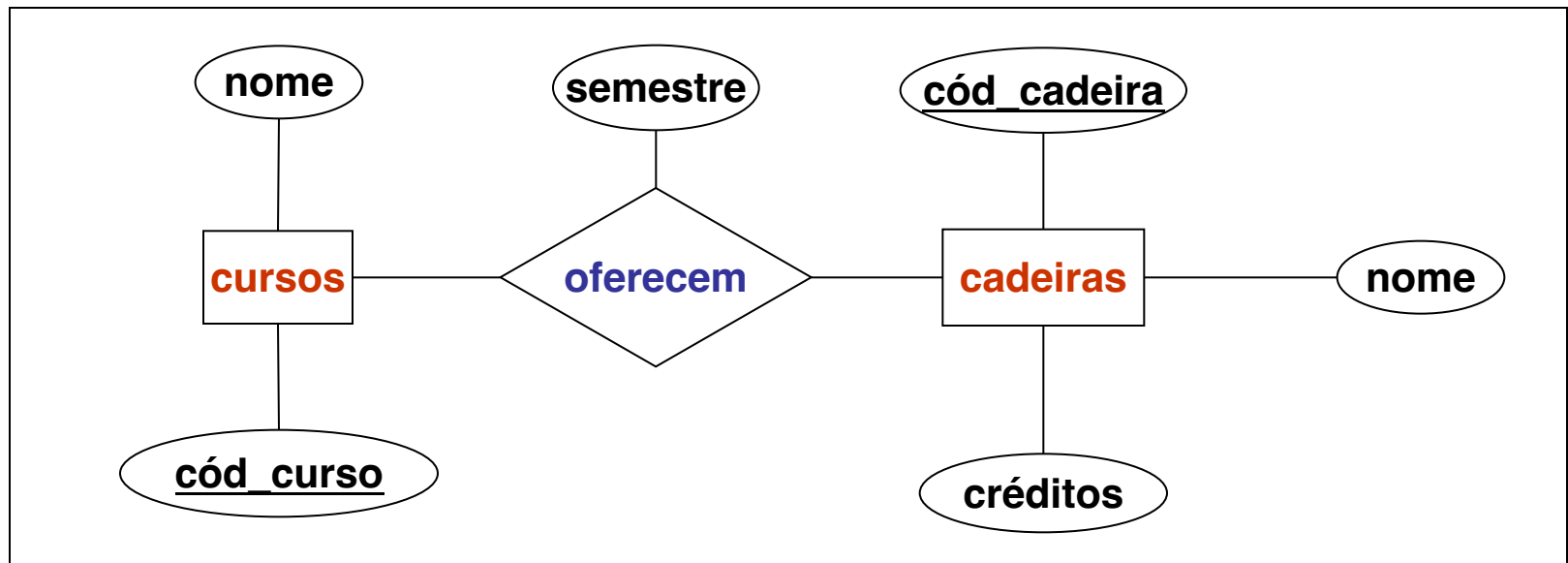
- Permitem representar graficamente as entidades, atributos, relações, restrições de mapeamento



- **Retângulos** representam conjuntos de entidades.
- **Losangos** representam conjuntos de relações.
- **Elipses** representam atributos
- **Linhas** ligam atributos aos conjuntos de entidades e conjuntos de entidades a conjuntos de associações.
- **Sublinhado** representa atributos constituintes da chave primária

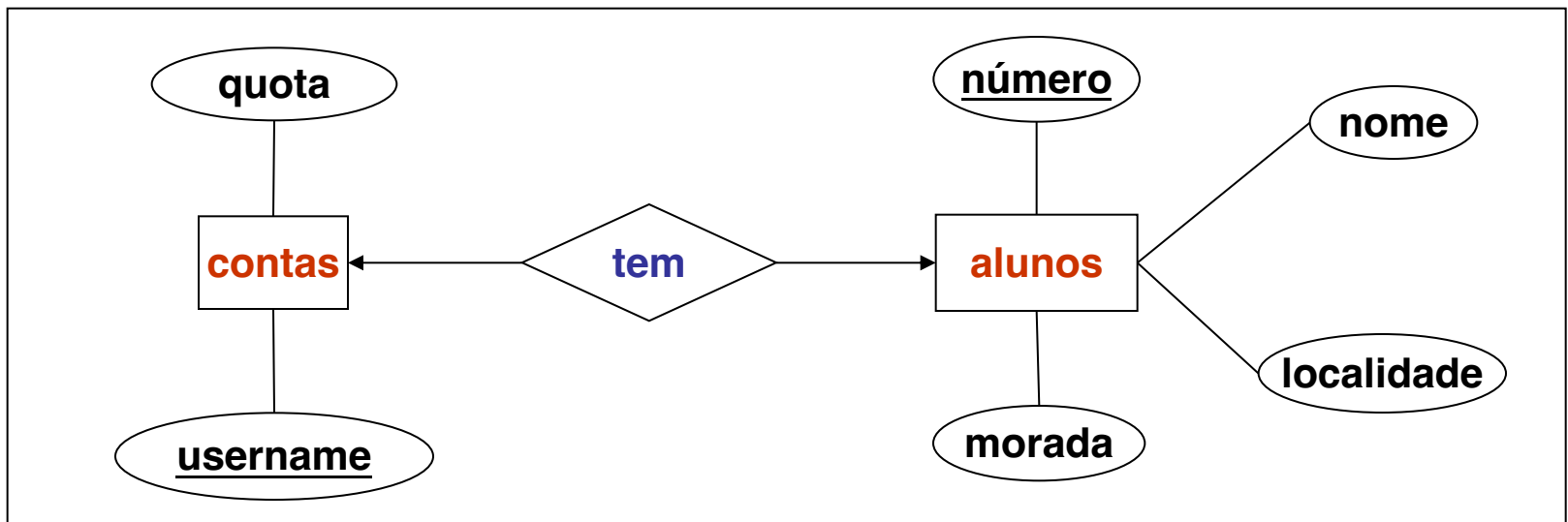
# Conjs. de Relação com Atributos

- Podemos ter atributos em conjuntos de relações. Por exemplo, para manter informação sobre o semestre em que uma dada cadeira é oferecida a um dado curso.



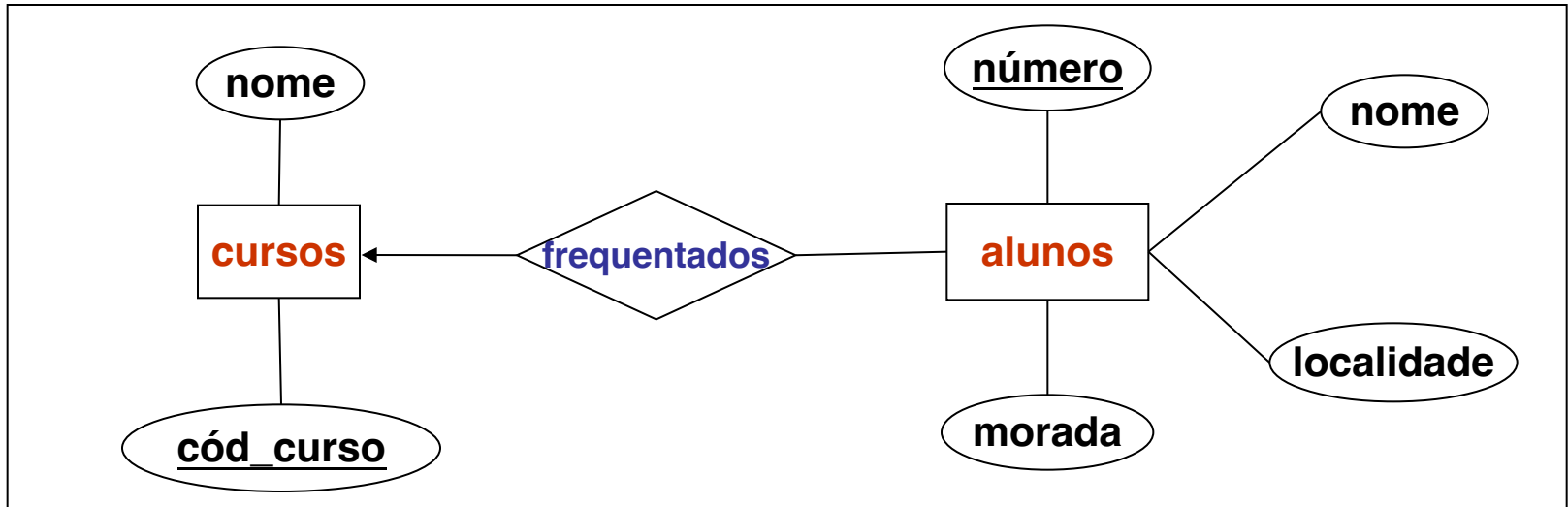
# Restrições de Mapeamento

- As restrições de mapeamento são expressas desenhando uma seta ( $\rightarrow$ ), significando “um,” ou uma linha ( $-$ ), significando “muitos,” entre o conj. de relações e o conj. de entidades.
- E.g.: relação um para um:
  - ★ Um aluno está associado no máximo a uma conta, através da relação *tem*.
  - ★ Uma conta está associada no máximo com um aluno.



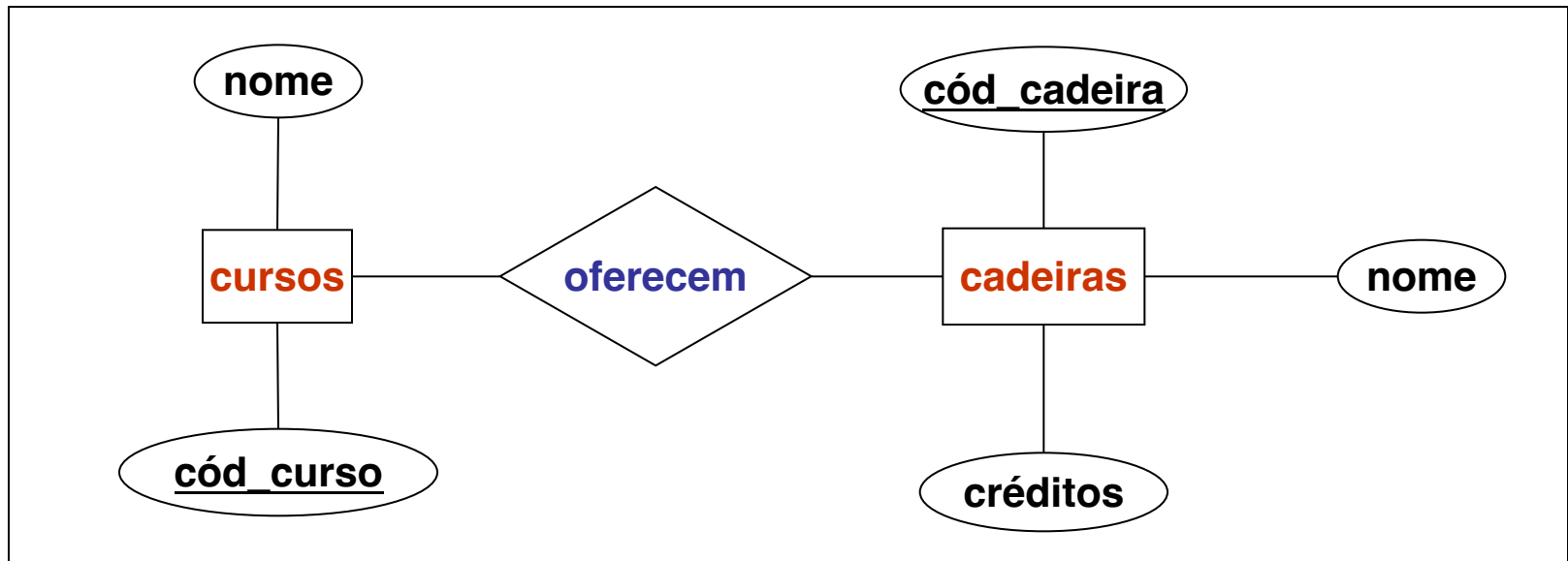
# Associações muitos para um

- Na relação muitos para um, um aluno está associada no máximo com um curso através da relação *frequentados*, enquanto que um curso pode estar associado com vários alunos (podendo ser 0) através de *frequentados*



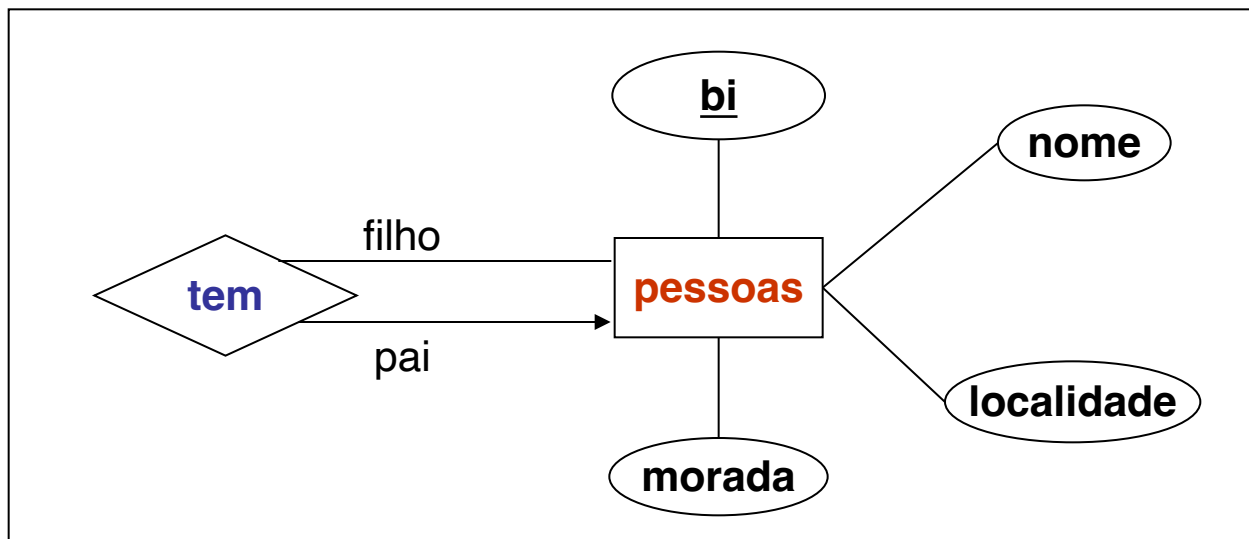
# Associações muitos para muitos

- Uma cadeira está associado com várias cursos (possivelmente 0) através da relação *oferecem*
- Um curso está associado com várias cadeiras (possivelmente 0) através da relação *oferecem*



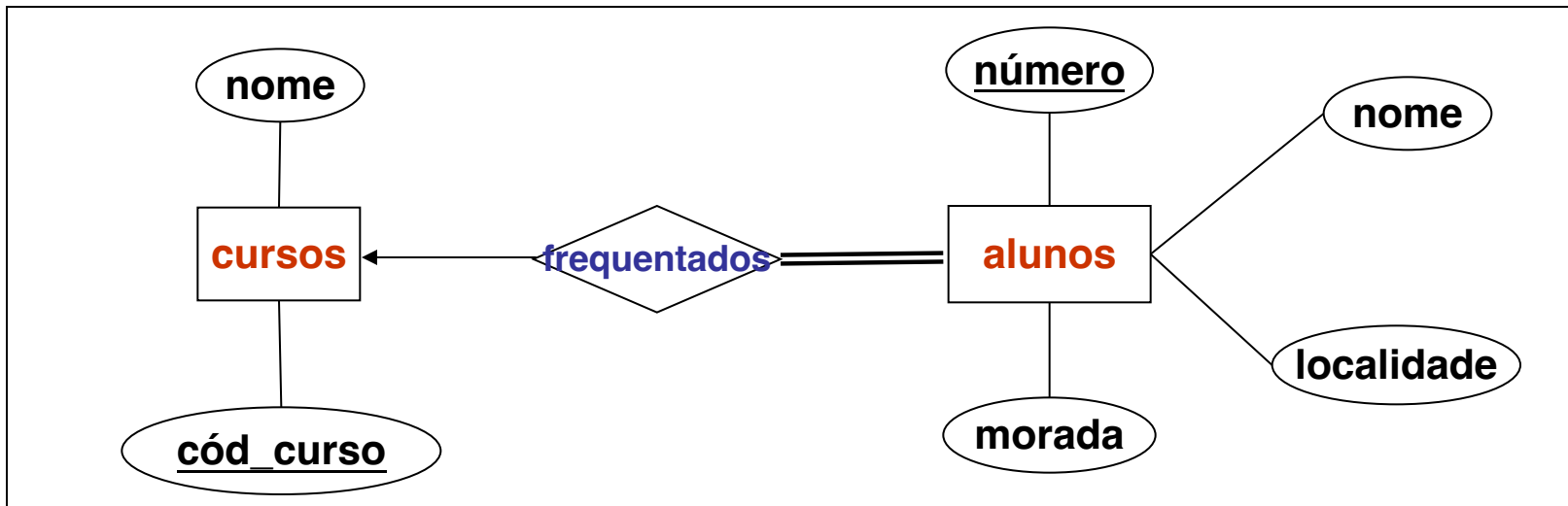
# Papéis

- Os conjuntos de entidades participantes numa relação não são obrigatoriamente distintos:
- As etiquetas “pai” e “filho” são designadas **papéis**; especificam como as entidades *pessoa* se relacionam por intermédio do conjunto de relações *tem*.
- Os papéis são indicadas nos DERs anotando as linhas que ligam os losangos aos retângulos.
- Os papéis são opcionais, sendo utilizados para clarificar a semântica da relação.



# Participação de um Conj. de Entidades num Conj. de Relação

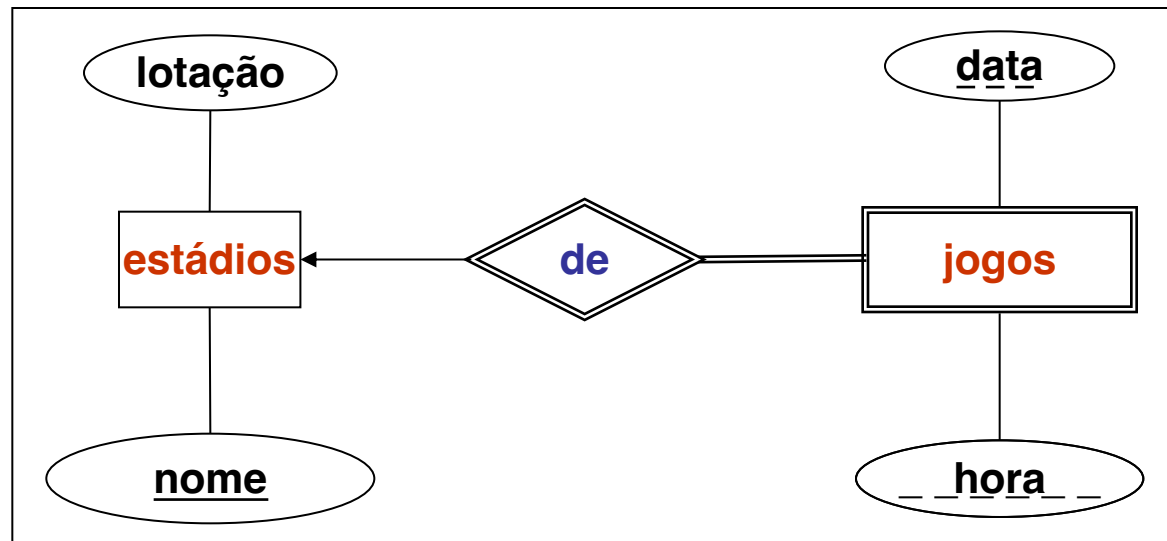
- **Participação total** (indicado por uma linha dupla): toda a entidade do conjunto de entidades participa em pelo menos uma relação do conjunto de relações.
  - E.g. a participação de *alunos* em *frequenta* é total
    - todo o aluno tem de ter um curso associado
- **Participação parcial**: algumas entidades podem não participar em qualquer relação do conjunto de relações.
  - E.g. a participação de *cursos* em *frequenta* é parcial pois podem existir cursos sem aluno (e.g. quando o curso é criado)





# Conjunto de Entidades Fraco

- Um conjunto de entidades fraco é representado por um retângulo duplo.
- O discriminante do conjunto de entidades fraco é sublinhado a tracejado.
- As relações entre o conjunto fraco e os dominantes são representadas por losangos duplos.



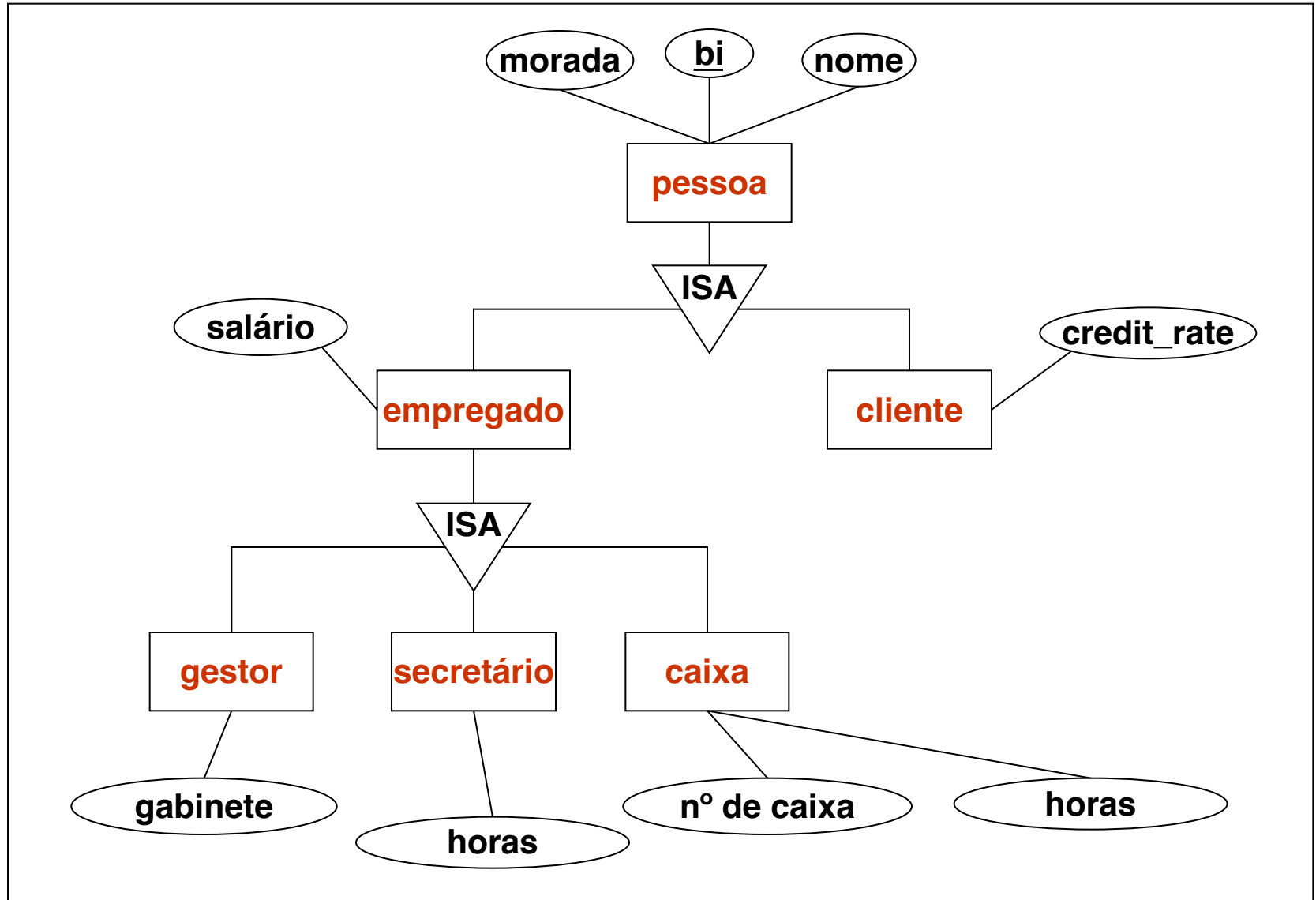
# Especialização/Generalização

- Há entidades que são “parecidas” mas não exactamente dum mesmo conjunto.
  - ✱ E.g. quer os empregados quer os clientes têm um nome, morada, telefone, etc. Mas os empregados têm salário (e os clientes não) e os clientes tem rating de crédito (e os empregados, enquanto tal, não).
- **Método de desenho descendente**: designamos subgrupos dentro de um conjunto de entidades que são distintas de outras entidades nesse conjunto (Especialização).
  - ✱ E.g. designar subgrupo empregados e clientes dentro do conjunto mais geral de pessoas.
- **Método de desenho ascendente** (bottom-up) (outra maneira de ver) – combinar num conjunto de entidades de maior nível (generalização) um certo número de conjuntos de entidades que partilham as mesmas características
- Os subgrupos (especializações) tornam-se conjuntos de entidades de “menor” nível que têm atributos ou participam em relações que não se aplicam ao conjunto de entidades de maior nível.

# DER de Especialização/Generalização

- Desenhado por um *triângulo* anotado com ISA: um cliente é (“is a”) uma *pessoa*.
- **Herança de atributos** – um conjunto de entidades de menor nível herda todos os atributos e participa em todas as relações do conjunto de entidades de maior nível ao qual está ligado.
  - ✦ Os atributos dos conjuntos de entidades mais gerais **não** são explicitamente representados nos conjuntos de entidades mais específicos.
    - ❖ Nomeadamente, os atributos que constituem a chave primária apenas são indicados no conjunto de entidades mais geral.

# Exemplo de Especialização

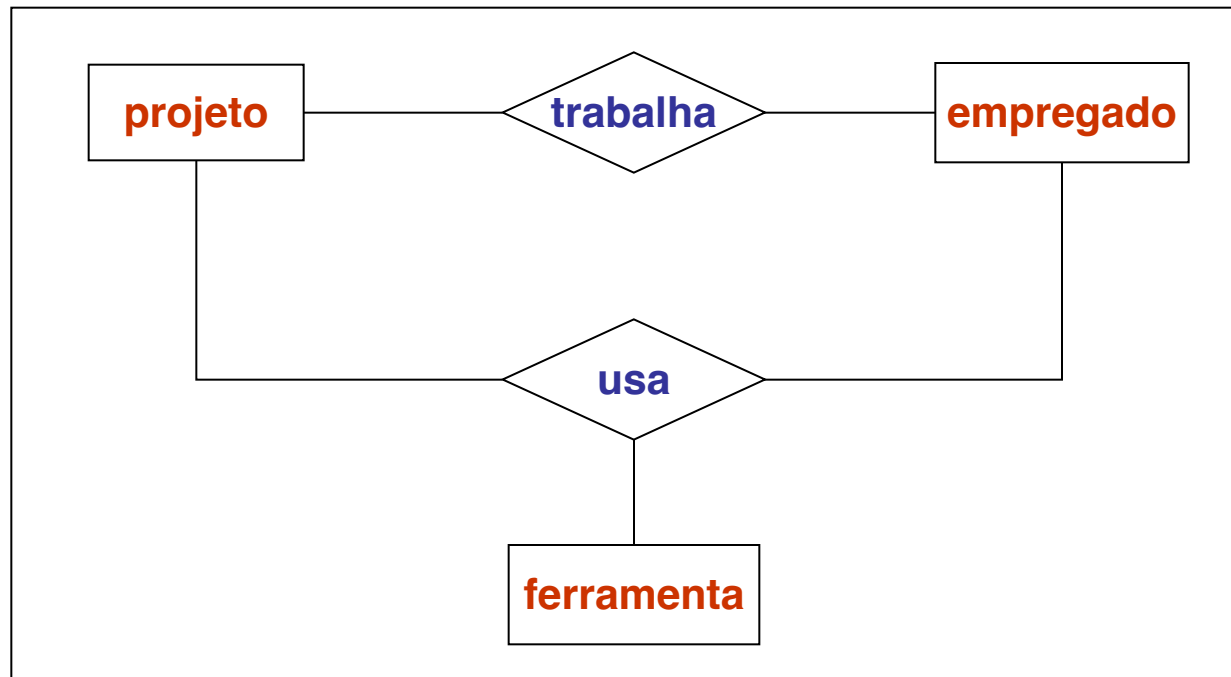


# Restrições de Desenho para Especialização/Generalização

- **Restrição de pertença** – especifica se uma entidade no conjunto de maior nível *pode ou não pertencer* a mais que um conjunto do nível inferior.
  - ✱ **disjuntas**: só pode pertencer a um do nível inferior (anotado com a palavra *disjoint* ao lado do triângulo)
  - ✱ **sobrepostas**: pode pertencer a mais que um.
- **Restrição de completude** – especifica se uma entidade no conjunto de maior nível *tem ou não que pertencer* a pelo menos um dos conjuntos do nível inferior.
  - ✱ **total**: tem de pertencer pelo menos a um (indicado por uma linha dupla)
  - ✱ **parcial**: pode não pertencer a nenhum

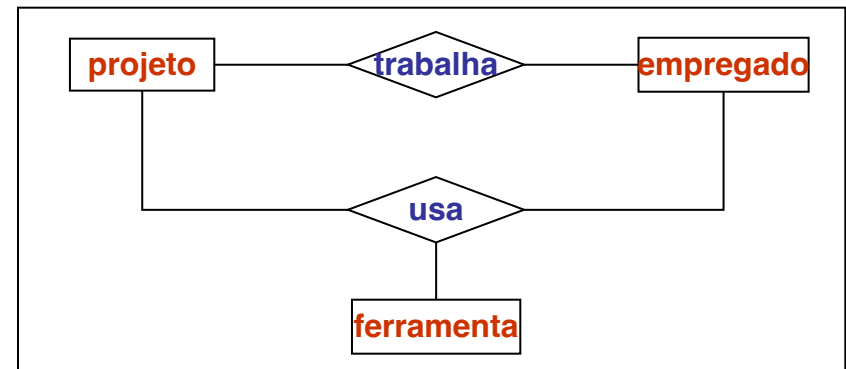
# Agregação

- Consideremos um cenário onde existem **projetos**, **empregados** e **ferramentas**:
  - ★ *Um empregado pode trabalhar em vários projetos (e num projeto pode haver vários empregados).*
  - ★ *Queremos guardar informação sobre as ferramentas usadas por um empregado num projeto.*



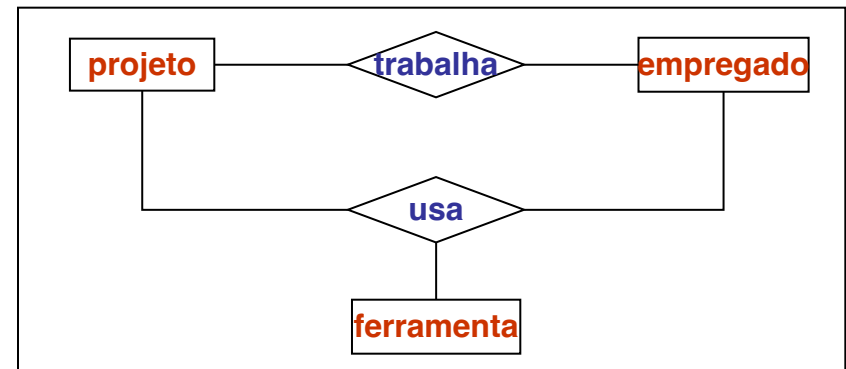
# Agregação

- O conjunto de relações **trabalha** e **usa** contém uma intersecção de informação.
  - ✦ Cada relação **usa** corresponde a uma relação **trabalha**.
  - ✦ No entanto, algumas relações **trabalha** podem não corresponder a nenhuma relação **usa**.
    - ❖ Por isso, não podemos eliminar a relação **trabalha**.
- As ferramentas devem apenas ser usadas por empregados em projetos nos quais esses empregados trabalhem.
  - ❖ A relação **usa** permite relacionar ferramentas com projetos e empregados que não estão relacionados através da relação **trabalha**



# Agregação

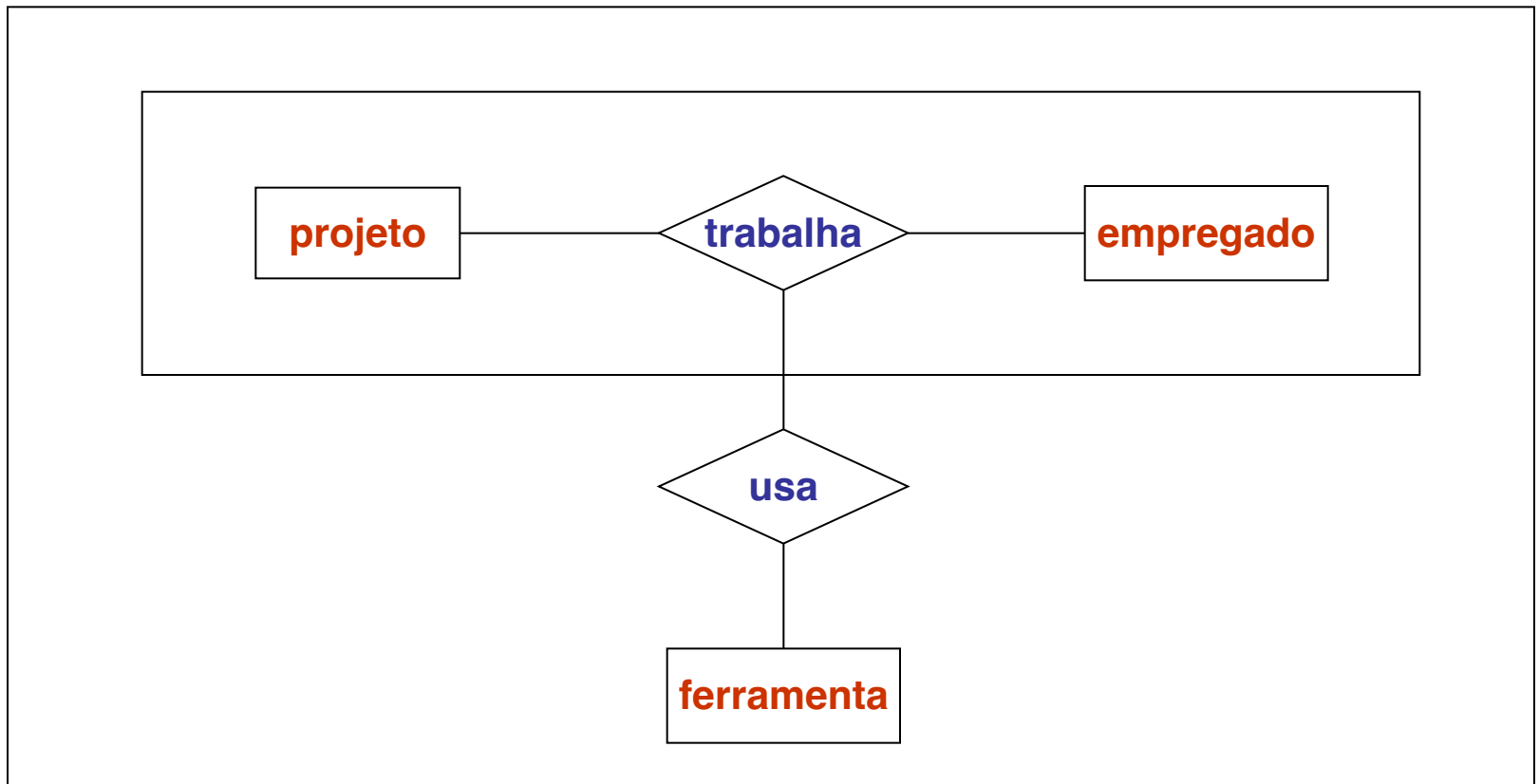
- Elimina-se esta redundância e restringe-se a participação na relação **usa** através da introdução do conceito de **agregação**.
  - ✦ Tratando relações como entidades abstratas.
    - ❖ Neste caso, tratando a relação **trabalha** como sendo uma entidade abstrata
  - ✦ Permitindo relações em que participam estas entidades abstratas.
    - ❖ Neste caso, permitindo que esta entidade abstrata **trabalha** participe na relação **usa**.



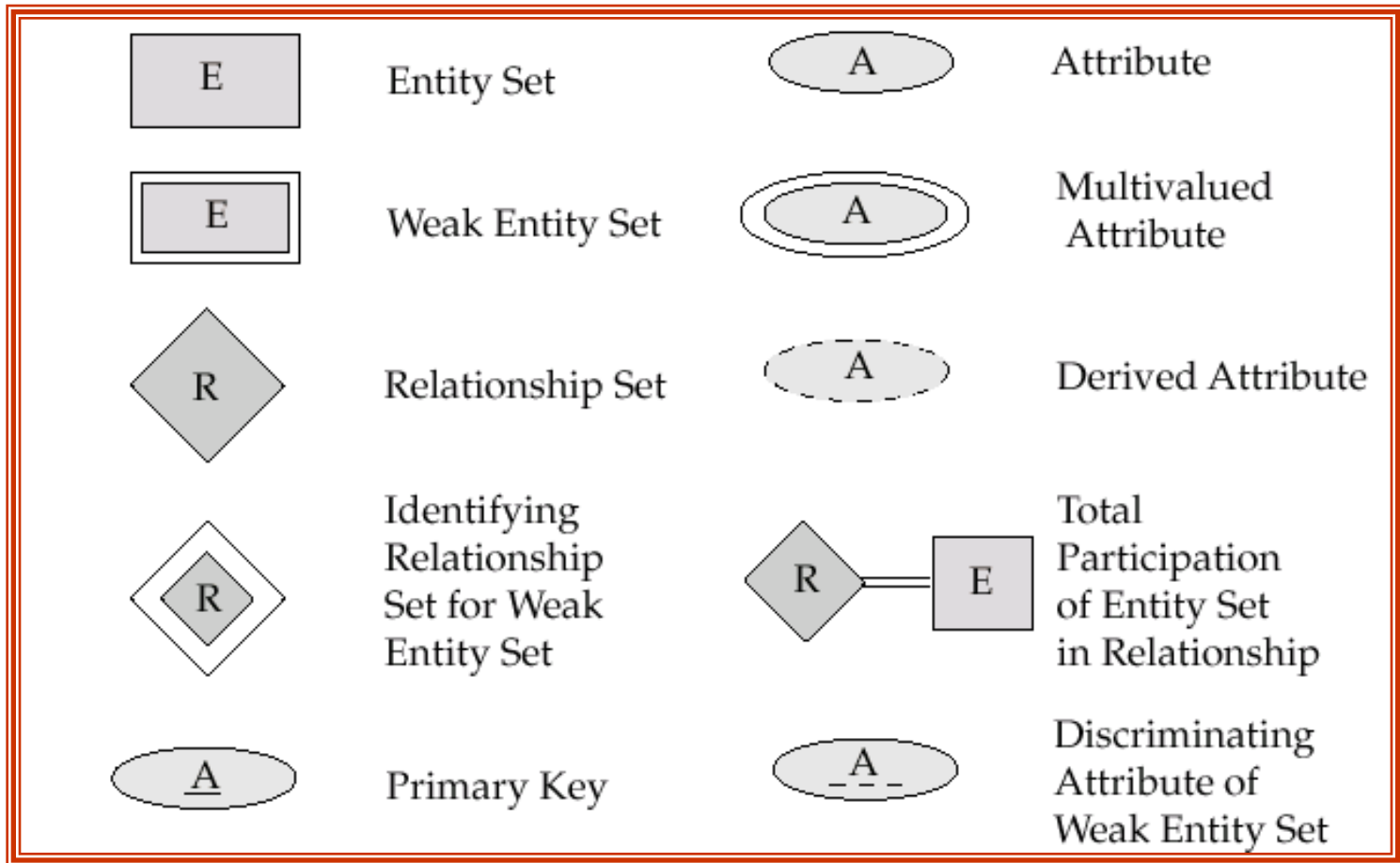


# DER com Agregação

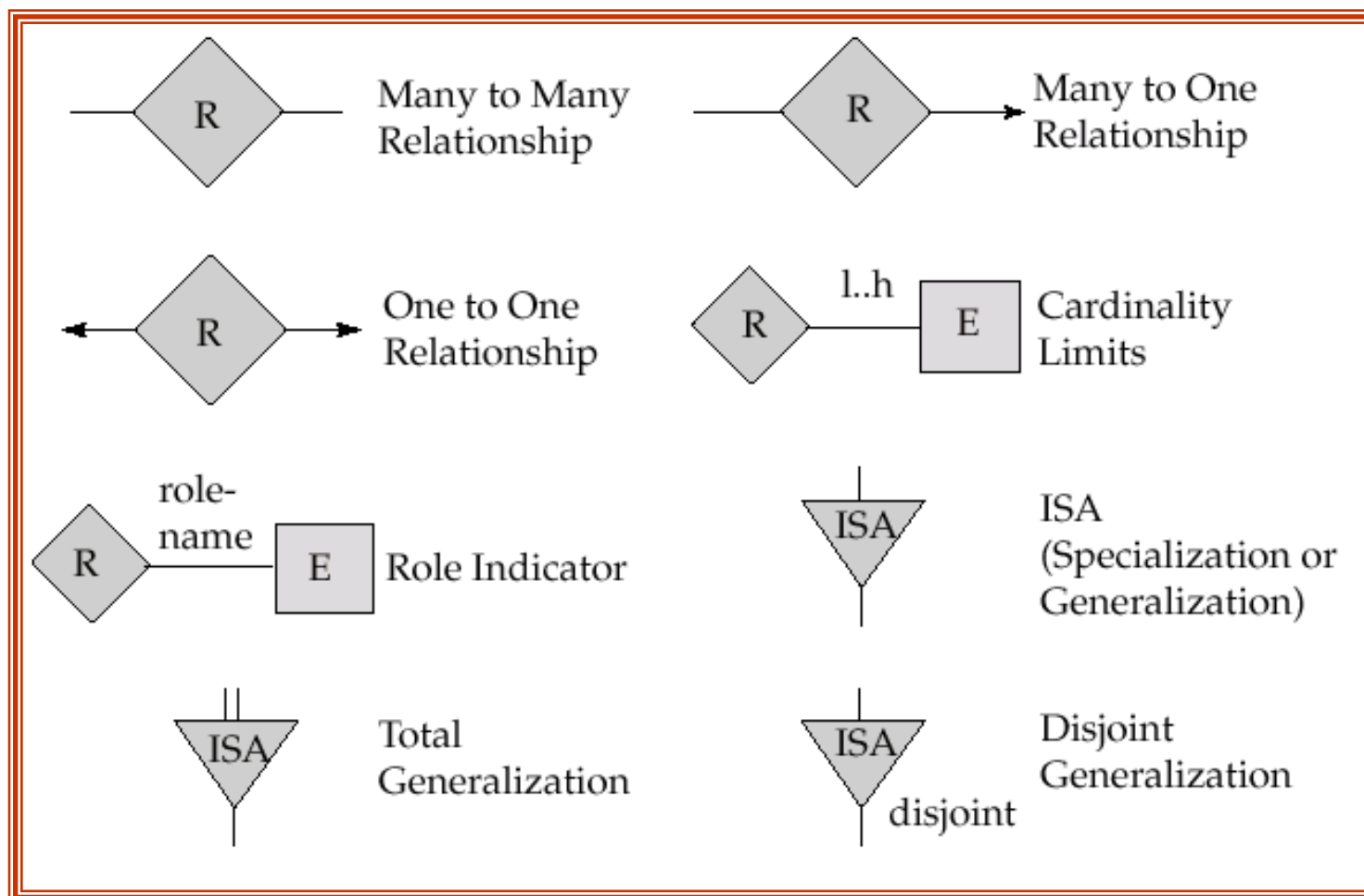
- Sem introdução de redundância, o diagrama representa:
  - ✦ Empregados trabalham em projetos;
  - ✦ Cada combinação projeto/empregado pode ter associada ferramentas.



# Sumário dos Símbolos Utilizados na Notação ER



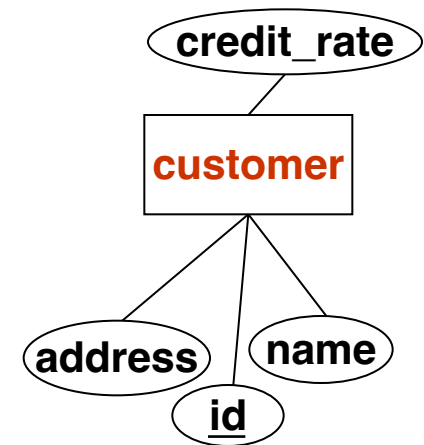
# Sumário dos Símbolos (Cont.)



# DER de um Banco

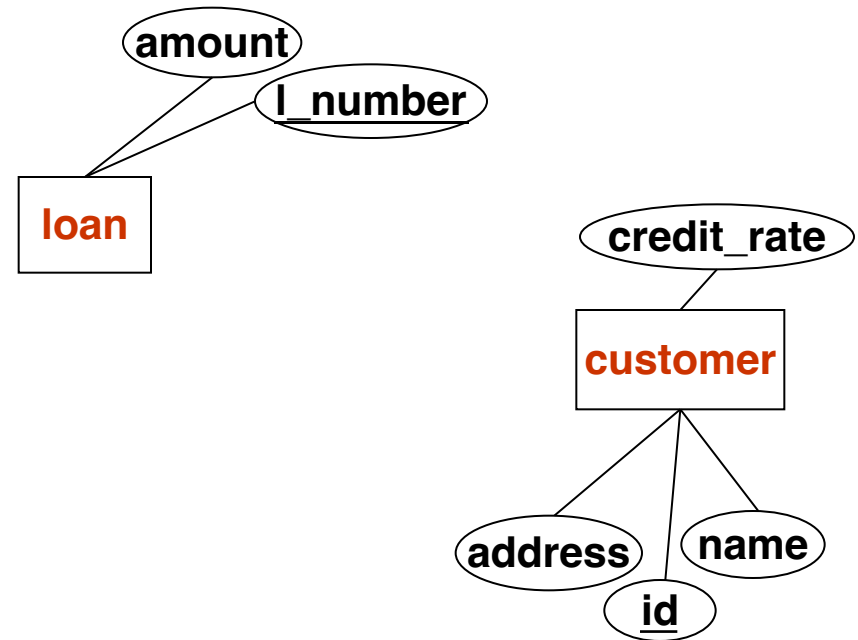
- O objectivo é desenhar uma base de dados para um banco com vários balcões, cada um representado por um nome, cidade e ativos.
- Pretende-se guardar informação sobre clientes (bi, nome, morada e credit\_rate), as suas contas (nº e saldo) e os seus empréstimos (nº e valor).
- Cada conta está associada a um balcão e tem um, ou mais, titulares. O mesmo se passa com os empréstimos.
- Pretende-se guardar a data de acesso de cada titular a cada conta.

# DER de um Banco



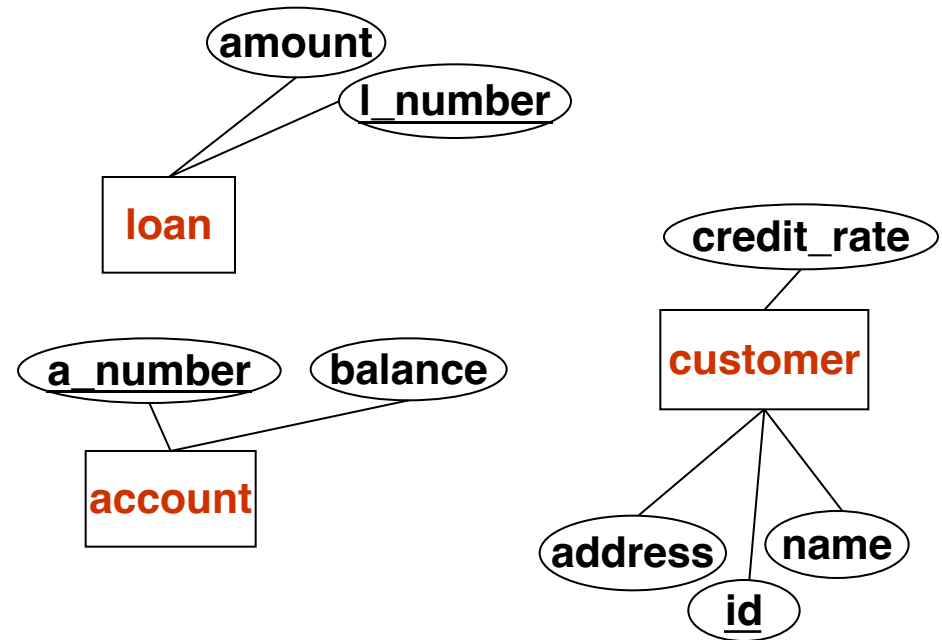
- O objectivo é desenhar uma base de dados para um banco com vários balcões, cada um representado por um nome, cidade e ativos.
- Pretende-se guardar informação sobre clientes (bi, nome, morada e credit\_rate), as suas contas (nº e saldo) e os seus empréstimos (nº e valor).
- Cada conta está associada a um balcão e tem um, ou mais, titulares. O mesmo se passa com os empréstimos.
- Pretende-se guardar a data de acesso de cada titular a cada conta.

# DER de um Banco



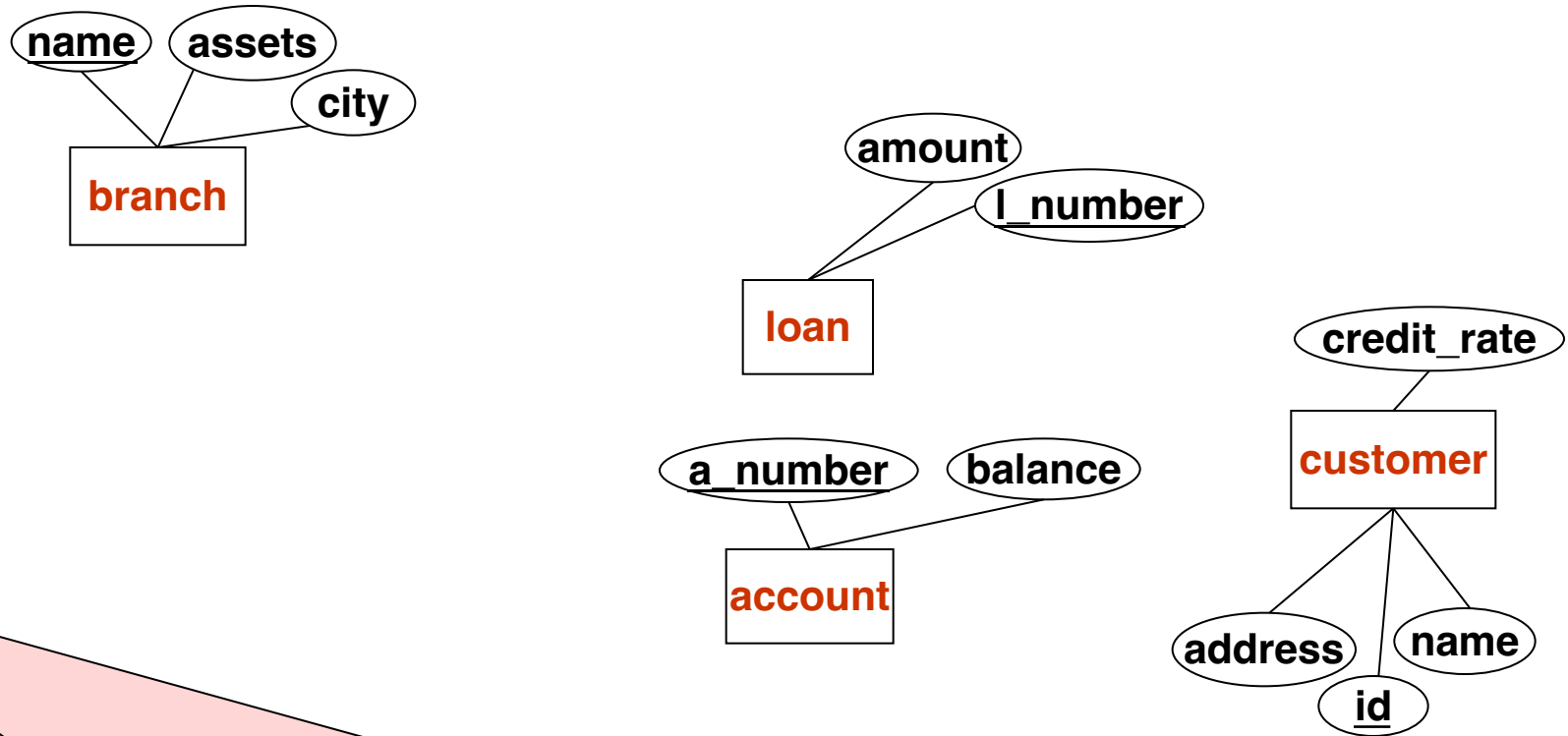
- O objectivo é desenhar uma base de dados para um banco com vários balcões, cada um representado por um nome, cidade e ativos.
- Pretende-se guardar informação sobre clientes (bi, nome, morada e credit\_rate), as suas contas (nº e saldo) e os seus empréstimos (nº e valor).
- Cada conta está associada a um balcão e tem um, ou mais, titulares. O mesmo se passa com os empréstimos.
- Pretende-se guardar a data de acesso de cada titular a cada conta.

# DER de um Banco



- O objectivo é desenhar uma base de dados para um banco com vários balcões, cada um representado por um nome, cidade e ativos.
- Pretende-se guardar informação sobre clientes (bi, nome, morada e **credit\_rate**), as suas contas (nº e saldo) e os seus empréstimos (nº e valor).
- Cada conta está associada a um balcão e tem um, ou mais, titulares. O mesmo se passa com os empréstimos.
- Pretende-se guardar a data de acesso de cada titular a cada conta.

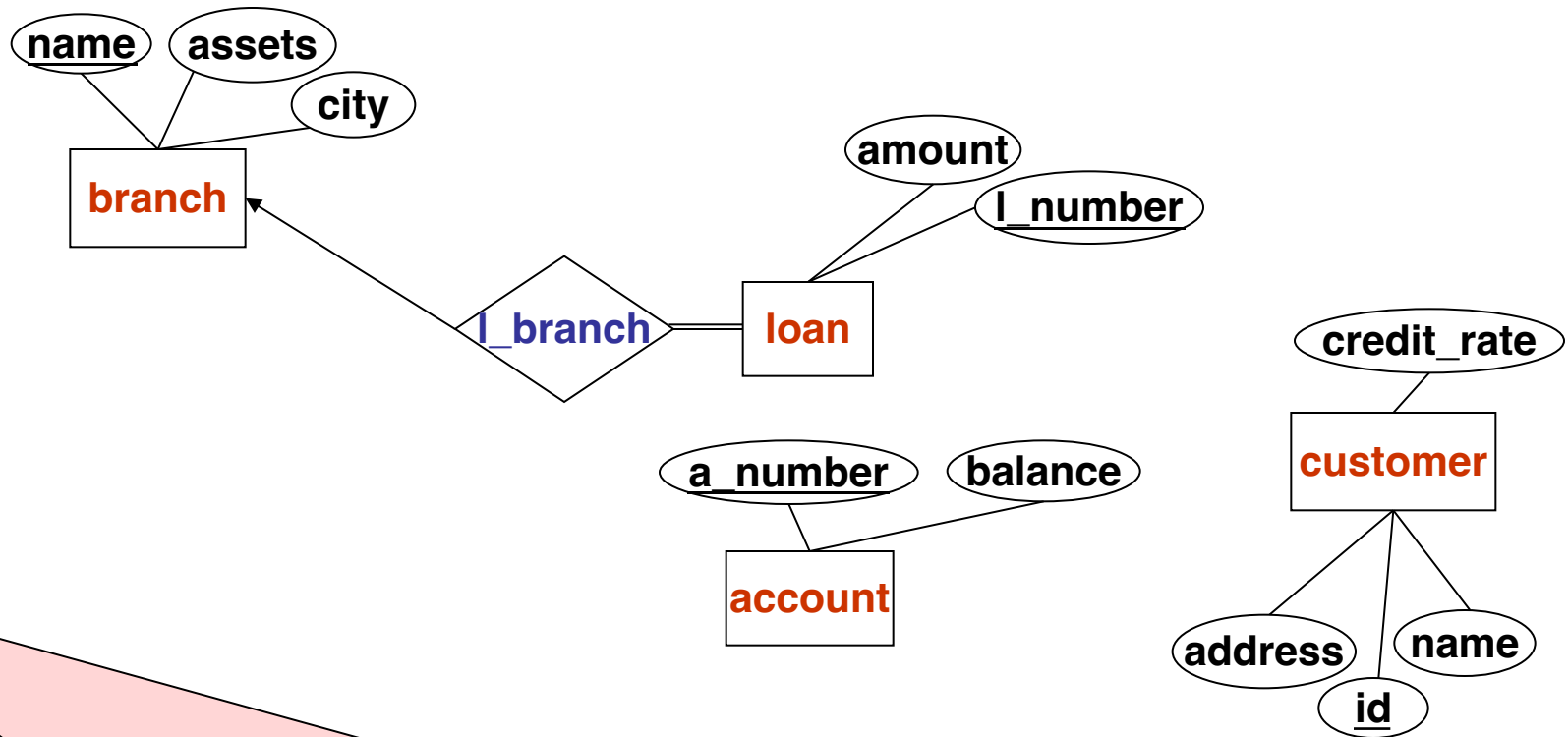
# DER de um Banco



- O objectivo é desenhar uma base de dados para um banco com vários balcões, cada um representado por um nome, cidade e ativos.
- Pretende-se guardar informação sobre clientes (bi, nome, morada e credit\_rate), as suas contas (nº e saldo) e os seus empréstimos (nº e valor).
- Cada conta está associada a um balcão e tem um, ou mais, titulares. O mesmo se passa com os empréstimos.
- Pretende-se guardar a data de acesso de cada titular a cada conta.

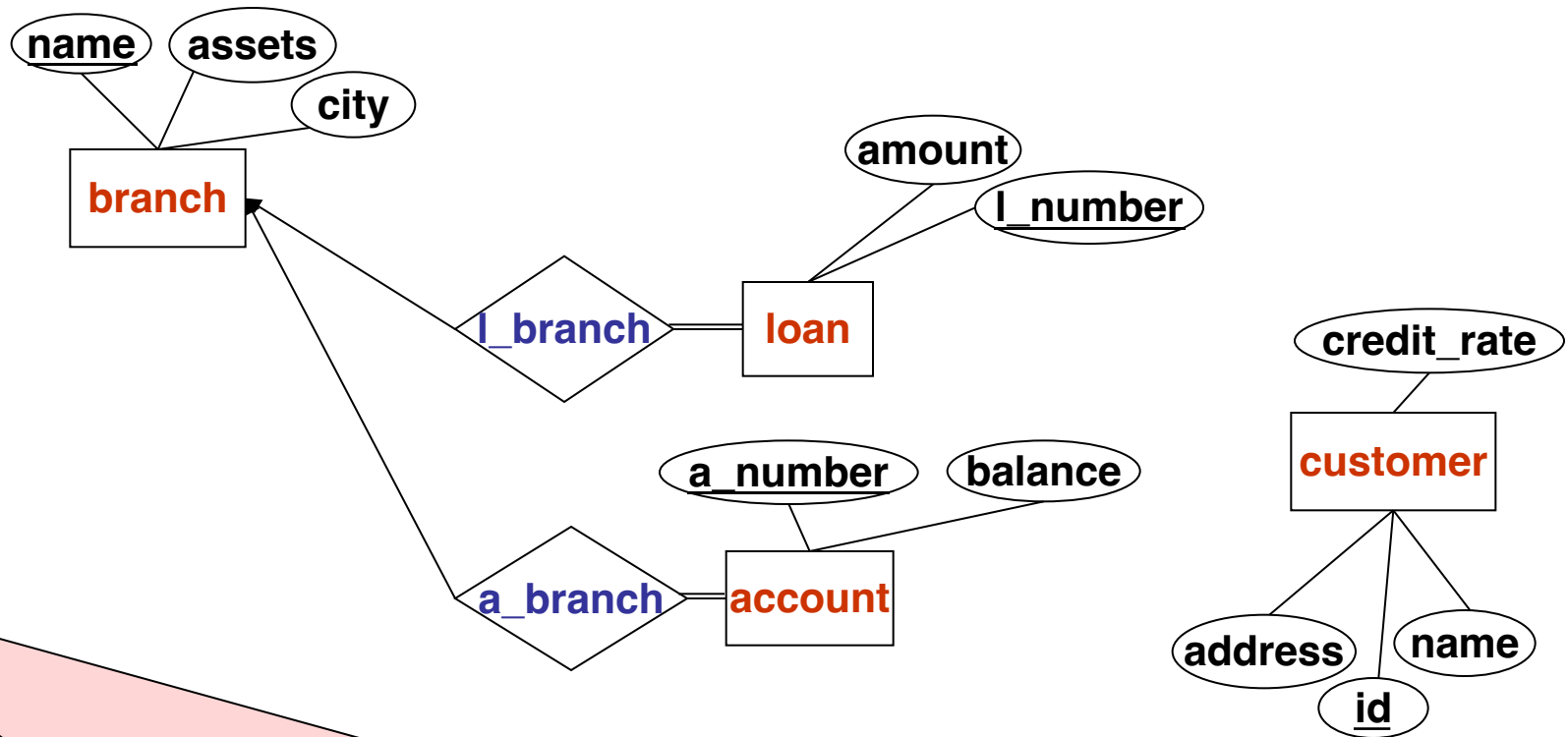


# DER de um Banco



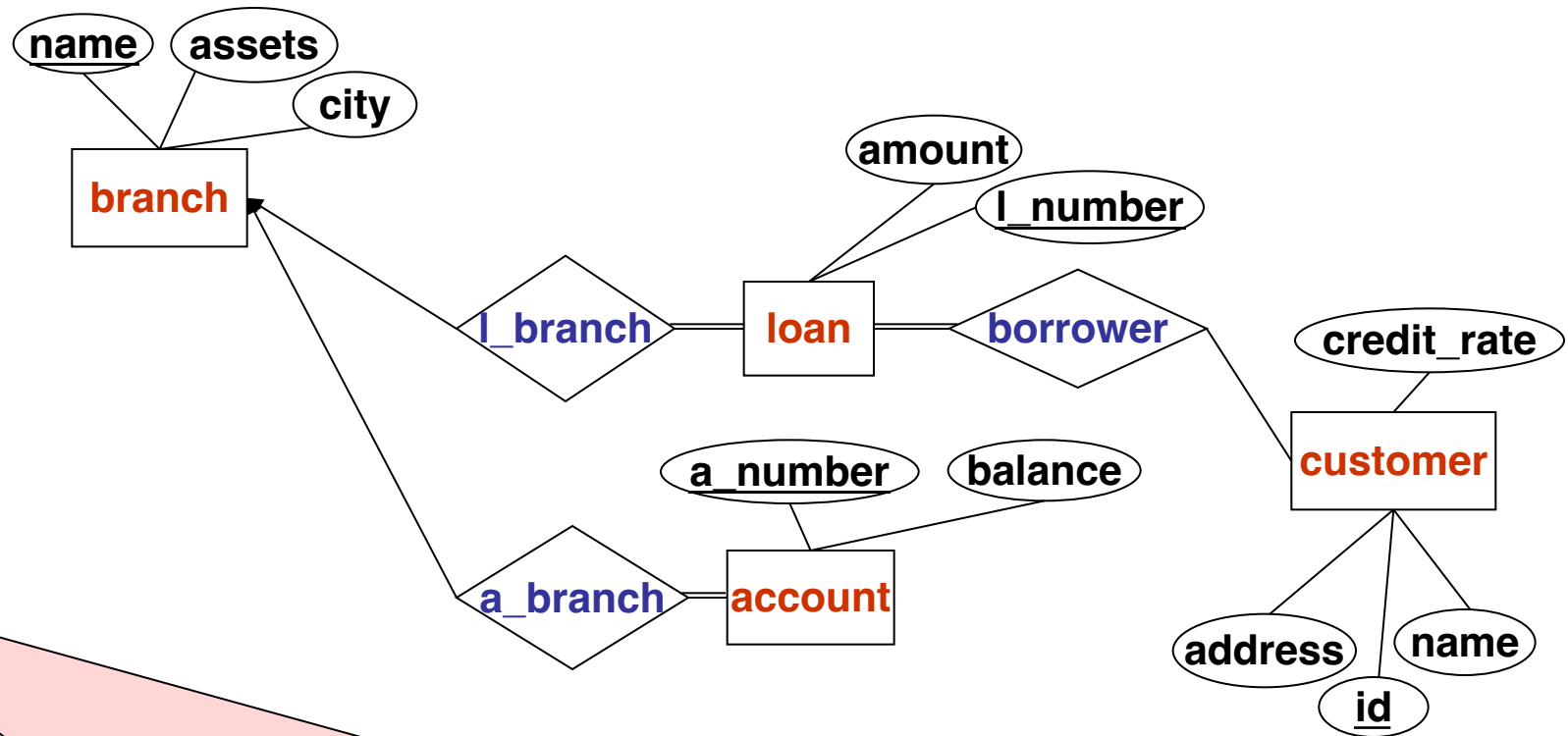
- O objectivo é desenhar uma base de dados para um banco com vários balcões, cada um representado por um nome, cidade e ativos.
- Pretende-se guardar informação sobre clientes (bi, nome, morada e credit\_rate), as suas contas (nº e saldo) e os seus empréstimos (nº e valor).
- Cada conta está associada a um balcão e tem um, ou mais, titulares. O mesmo se passa com os empréstimos.
- Pretende-se guardar a data de acesso de cada titular a cada conta.

# DER de um Banco



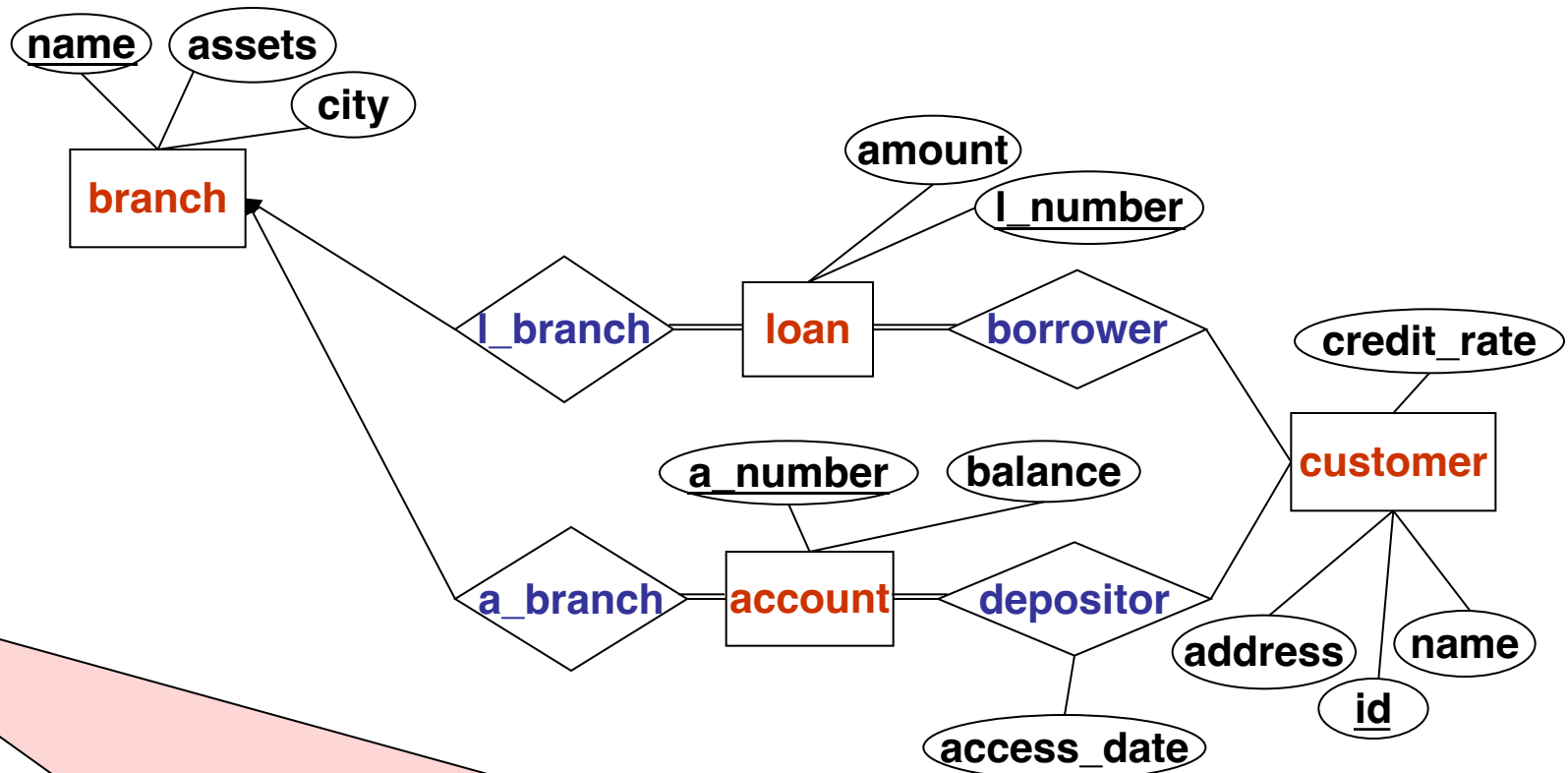
- O objectivo é desenhar uma base de dados para um banco com vários balcões, cada um representado por um nome, cidade e ativos.
- Pretende-se guardar informação sobre clientes (bi, nome, morada e credit\_rate), as suas contas (nº e saldo) e os seus empréstimos (nº e valor).
- Cada conta está associada a um balcão e tem um, ou mais, titulares. O mesmo se passa com os empréstimos.
- Pretende-se guardar a data de acesso de cada titular a cada conta.

# DER de um Banco



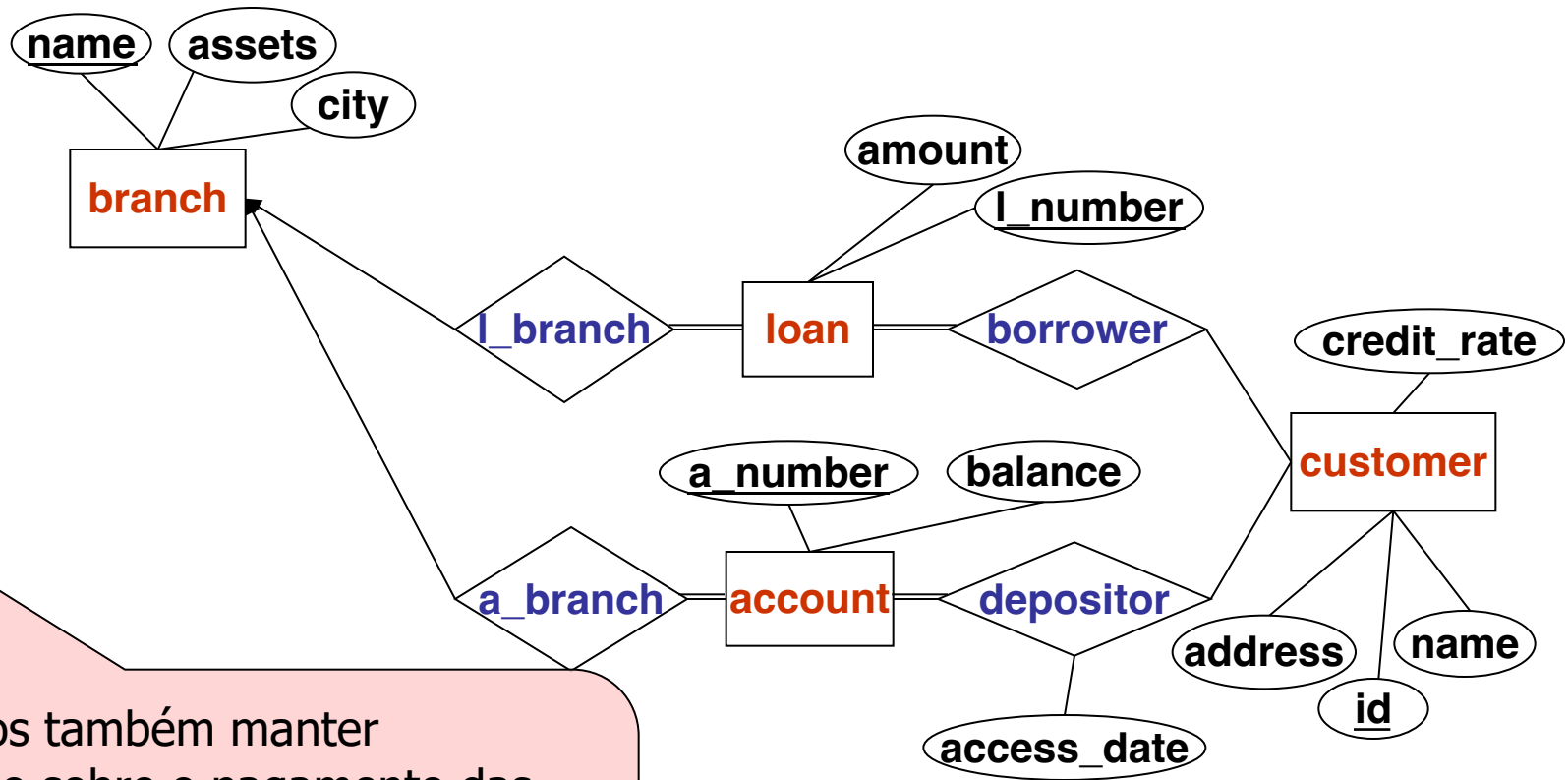
- O objectivo é desenhar uma base de dados para um banco com vários balcões, cada um representado por um nome, cidade e ativos.
- Pretende-se guardar informação sobre clientes (bi, nome, morada e credit\_rate), as suas contas (nº e saldo) e os seus empréstimos (nº e valor).
- Cada conta está associada a um balcão e tem um, ou mais, titulares. O mesmo se passa com os empréstimos.
- Pretende-se guardar a data de acesso de cada titular a cada conta.

# DER de um Banco



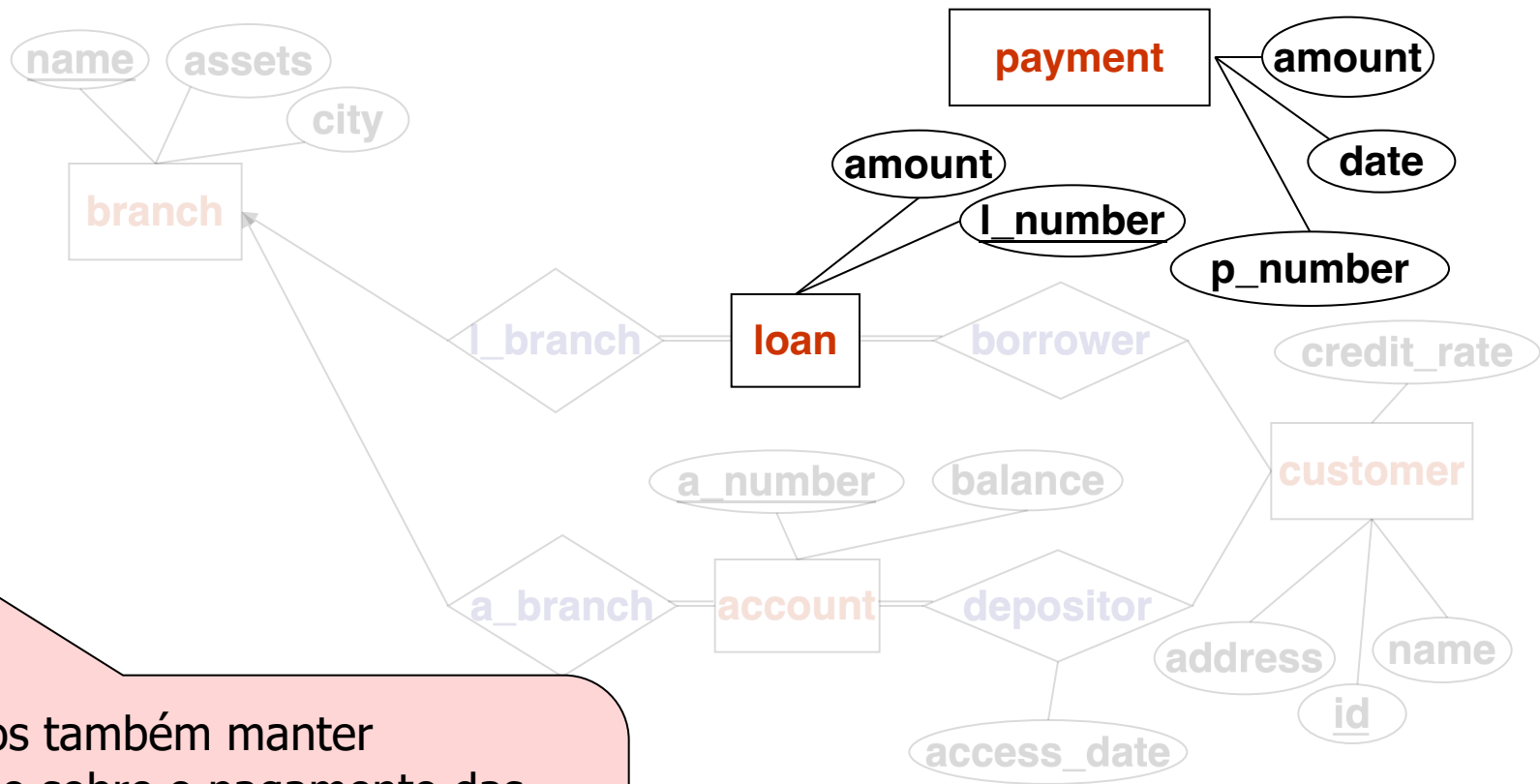
- O objectivo é desenhar uma base de dados para um banco com vários balcões, cada um representado por um nome, cidade e ativos.
- Pretende-se guardar informação sobre clientes (bi, nome, morada e credit\_rate), as suas contas (nº e saldo) e os seus empréstimos (nº e valor).
- Cada conta está associada a um balcão e tem um, ou mais, titulares. O mesmo se passa com os empréstimos.
- Pretende-se guardar a data de acesso de cada titular a cada conta.

# DER de um Banco



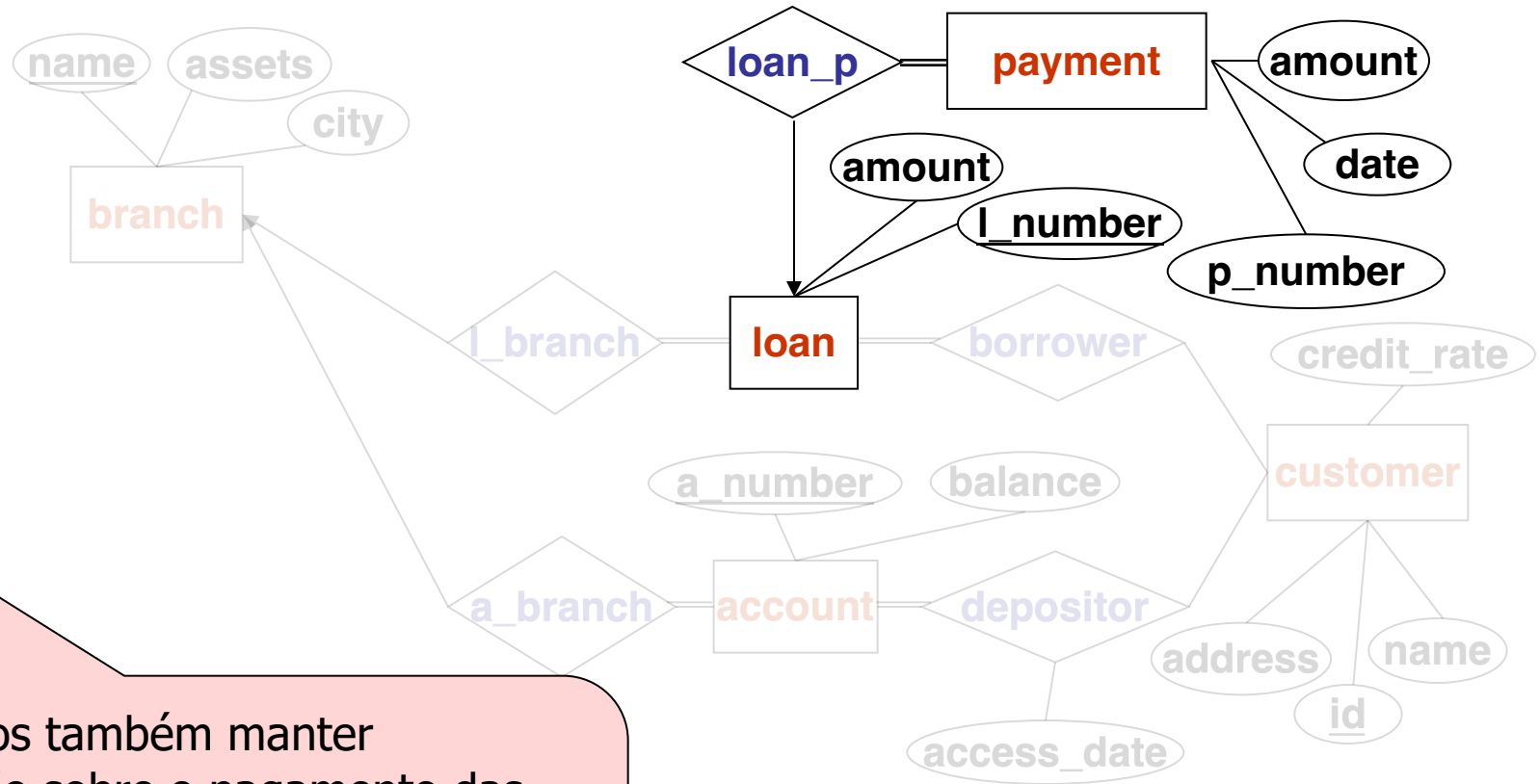
- Queremos também manter informação sobre o pagamento das prestações dos empréstimos. De cada prestação paga sabemos o seu número, a data de pagamento e o valor pago.

# DER de um Banco



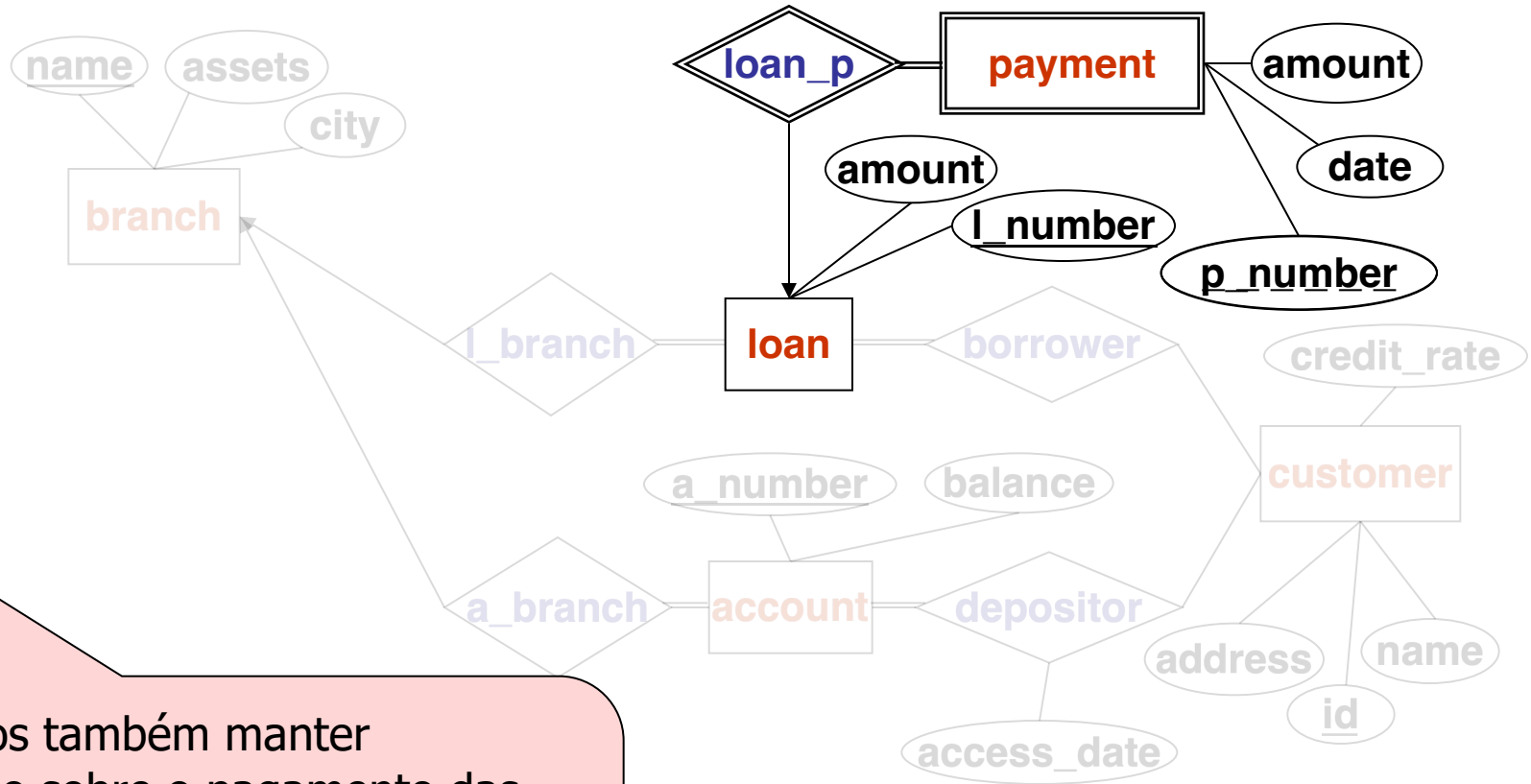
- Queremos também manter informação sobre o pagamento das prestações dos empréstimos. De cada prestação paga sabemos o seu número, a data de pagamento e o valor pago.

# DER de um Banco



- Queremos também manter informação sobre o pagamento das prestações dos empréstimos. De cada prestação paga sabemos o seu número, a data de pagamento e o valor pago.

# DER de um Banco

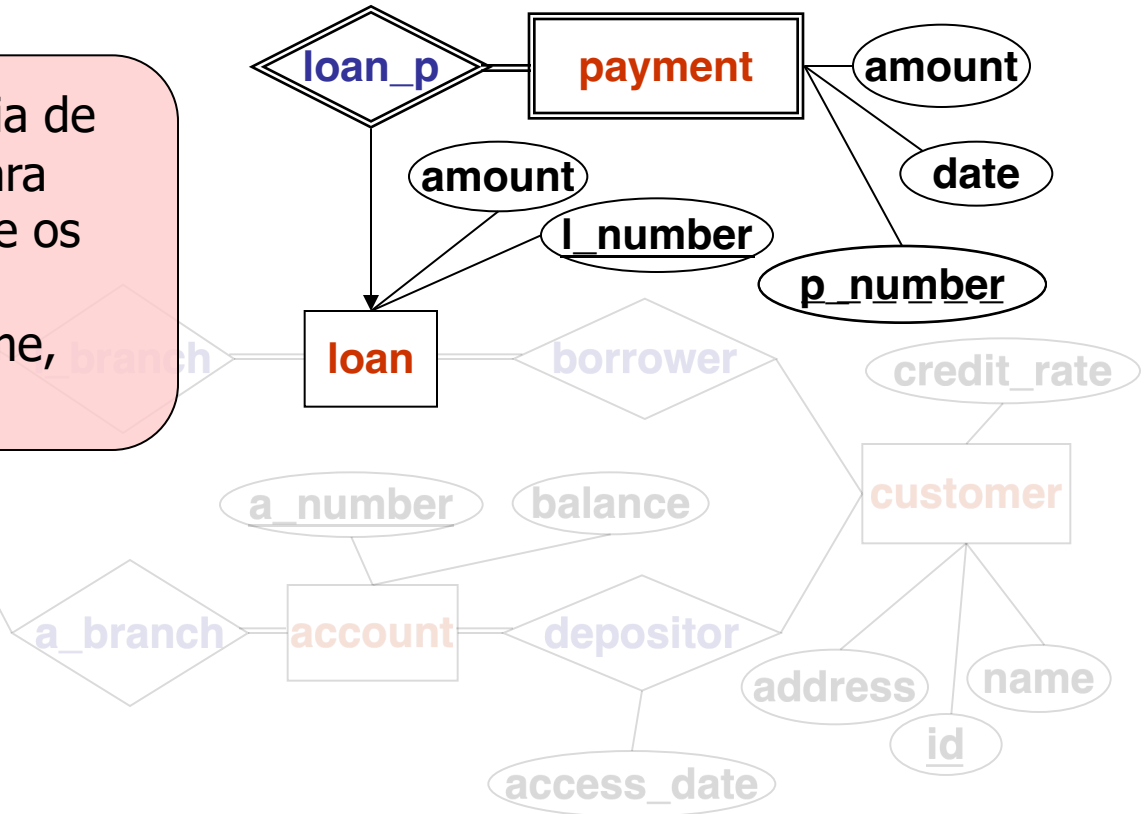


- Queremos também manter informação sobre o pagamento das prestações dos empréstimos. De cada prestação paga sabemos o seu número, a data de pagamento e o valor pago.



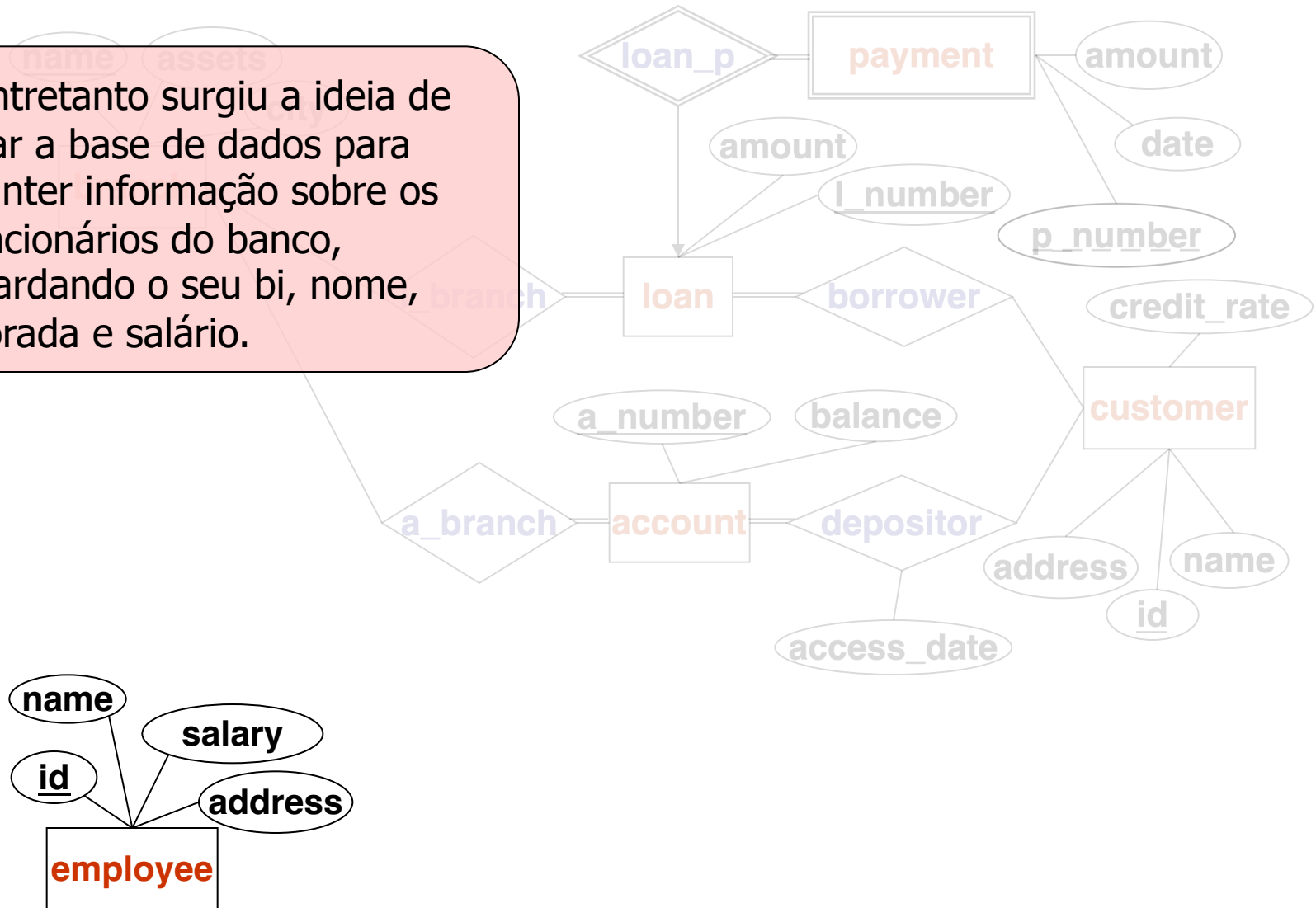
# DER de um Banco

- Entretanto surgiu a ideia de usar a base de dados para manter informação sobre os funcionários do banco, guardando o seu bi, nome, morada e salário.



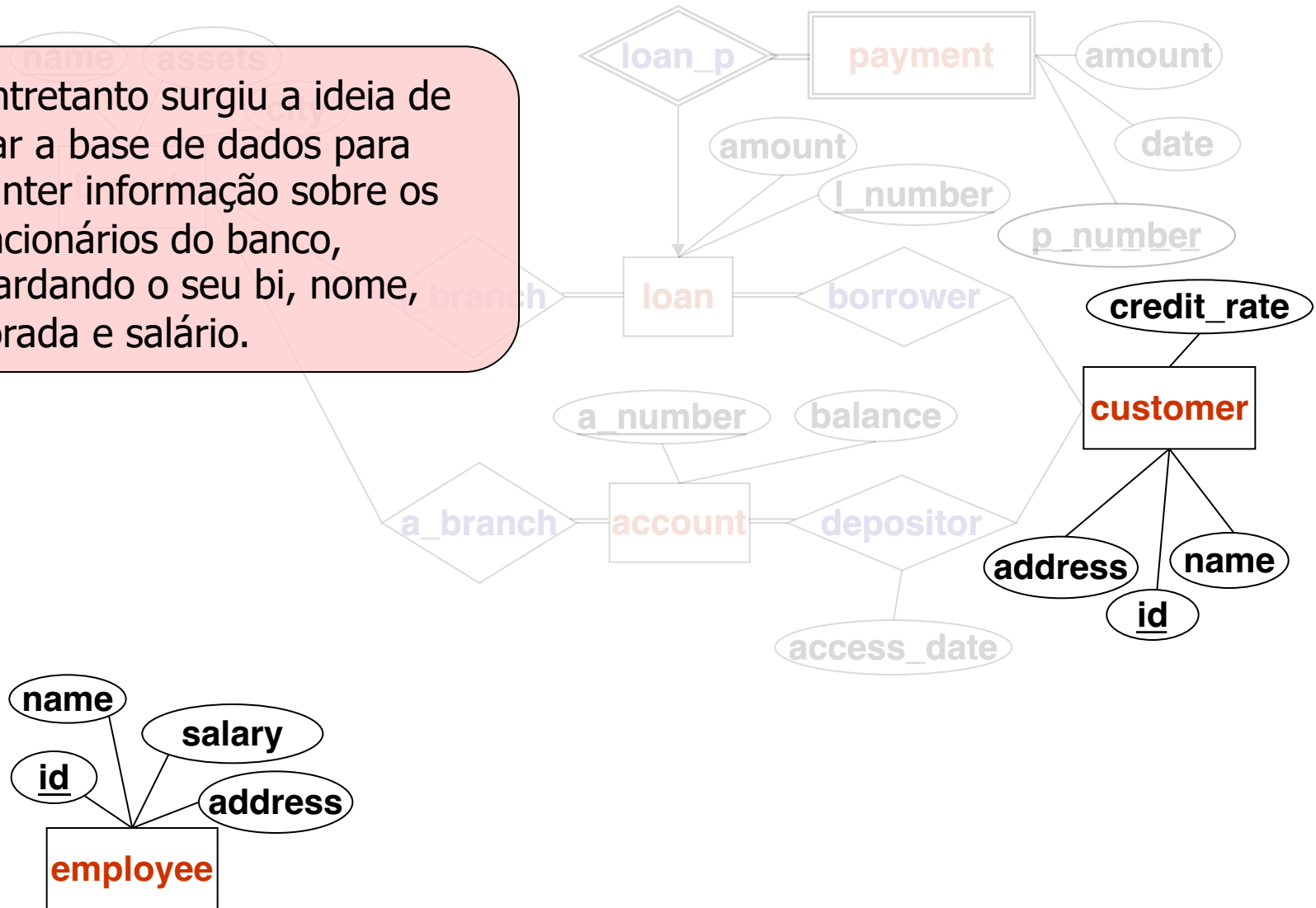
# DER de um Banco

- Entretanto surgiu a ideia de usar a base de dados para manter informação sobre os funcionários do banco, guardando o seu bi, nome, morada e salário.



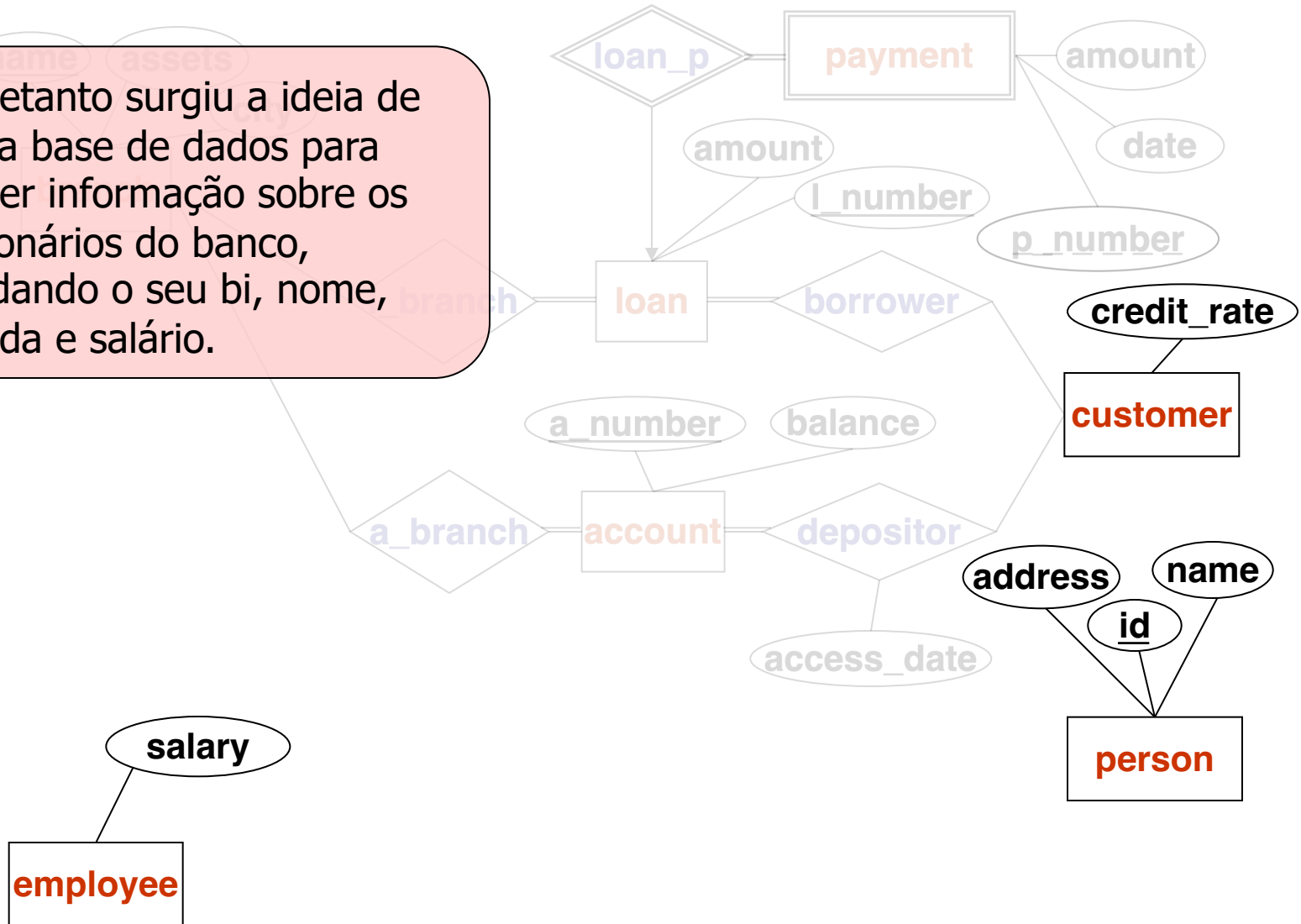
# DER de um Banco

- Entretanto surgiu a ideia de usar a base de dados para manter informação sobre os funcionários do banco, guardando o seu bi, nome, morada e salário.



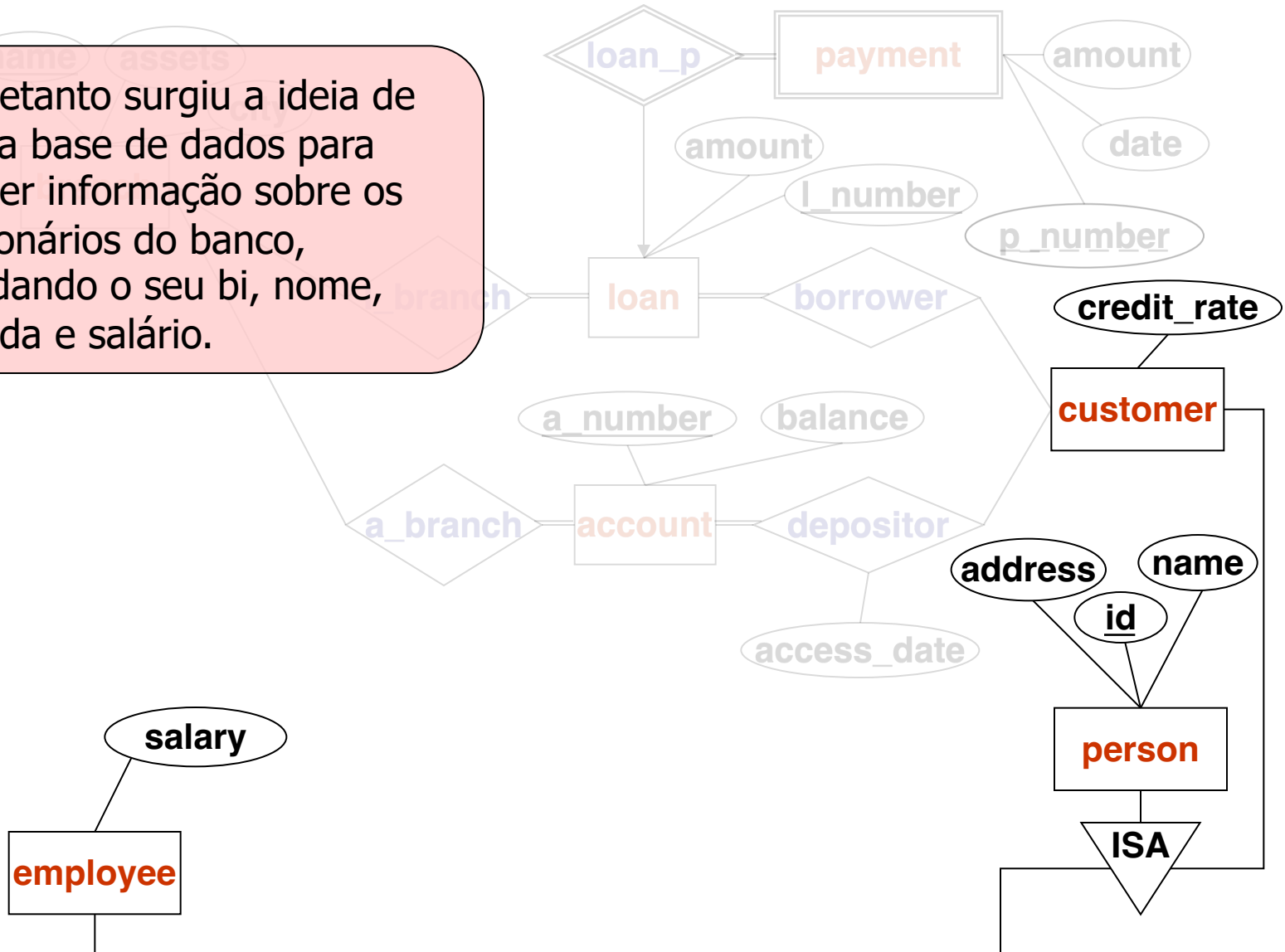
# DER de um Banco

- Entretanto surgiu a ideia de usar a base de dados para manter informação sobre os funcionários do banco, guardando o seu bi, nome, morada e salário.

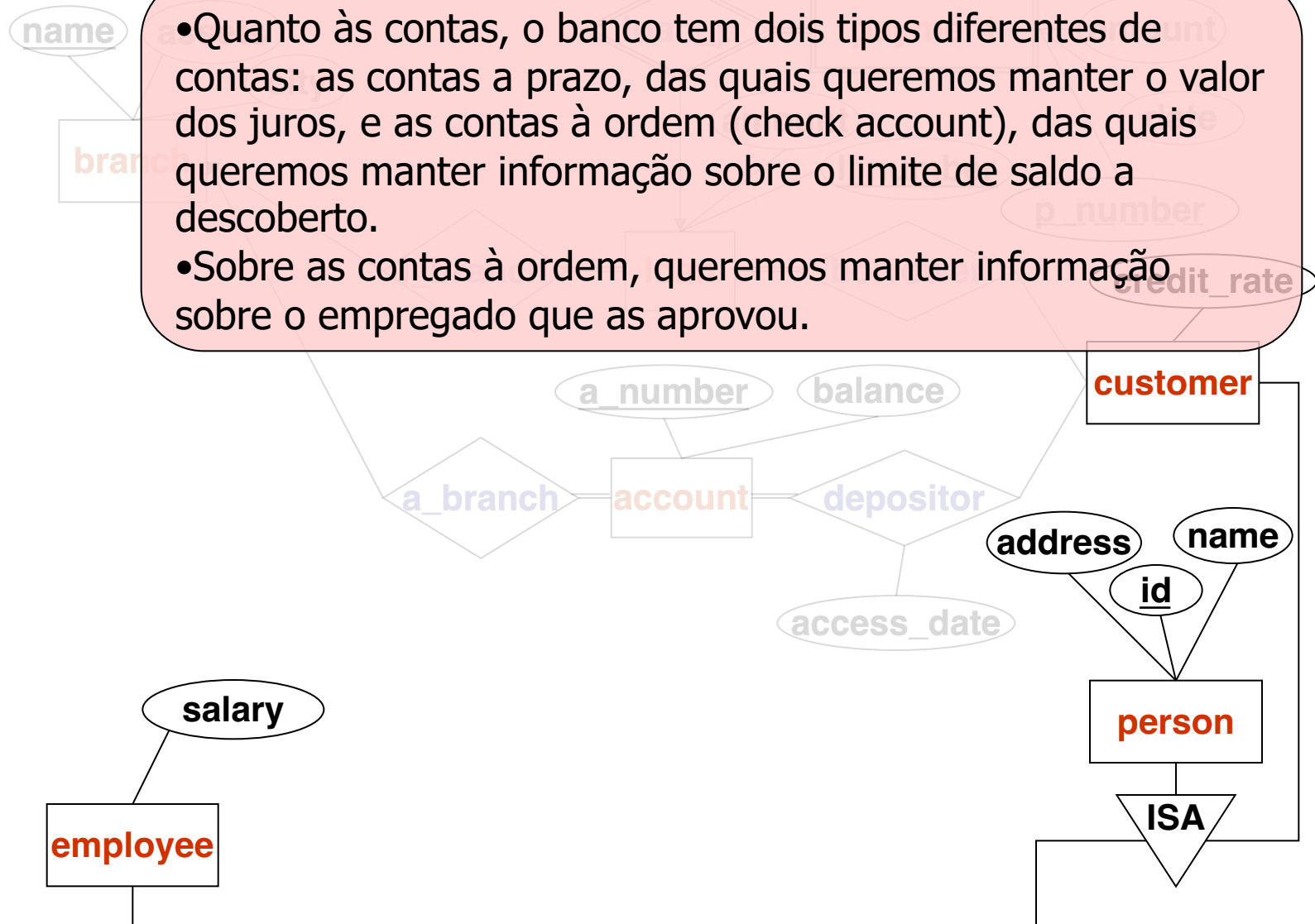


# DER de um Banco

- Entretanto surgiu a ideia de usar a base de dados para manter informação sobre os funcionários do banco, guardando o seu bi, nome, morada e salário.

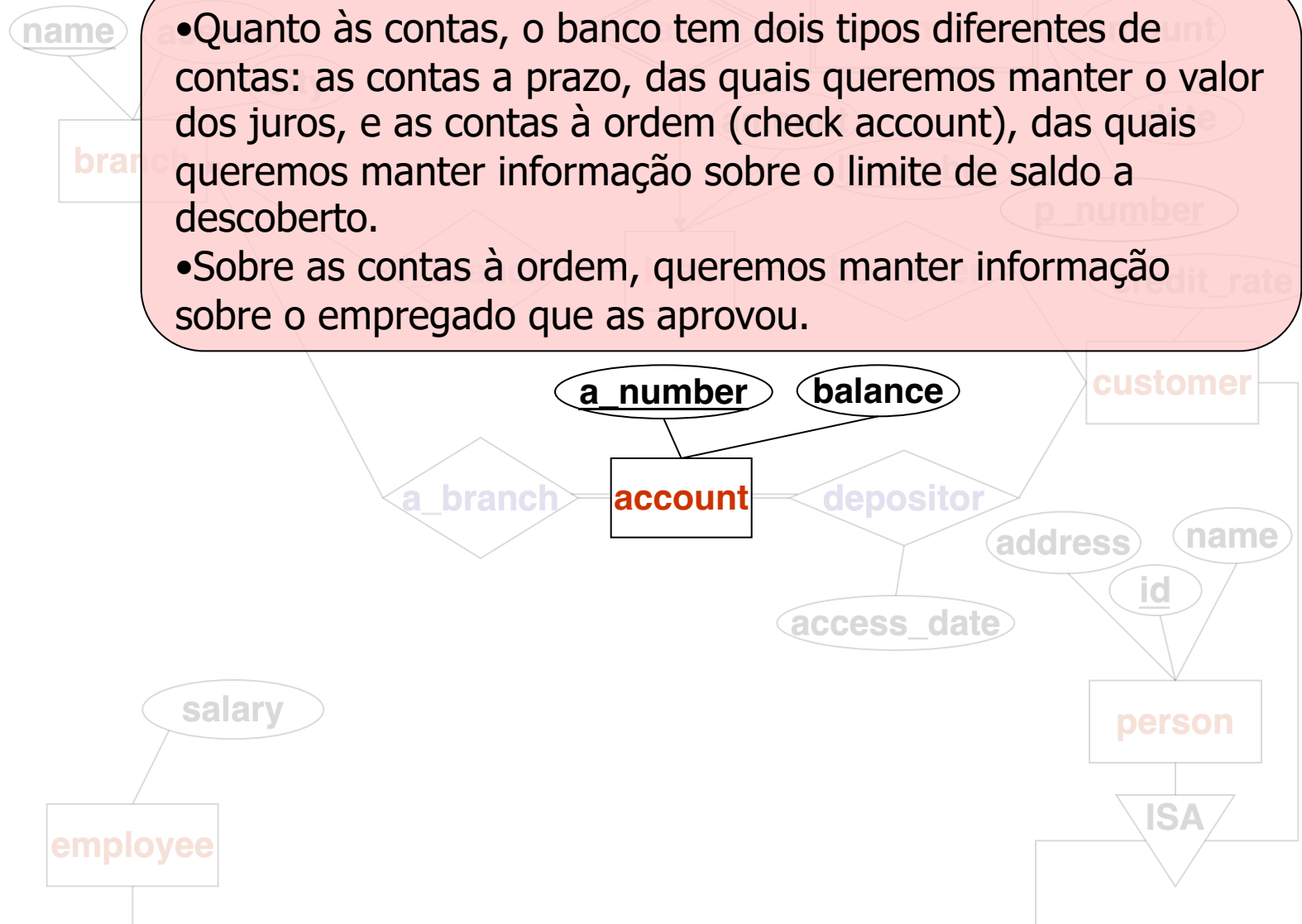


# DER de um Banco



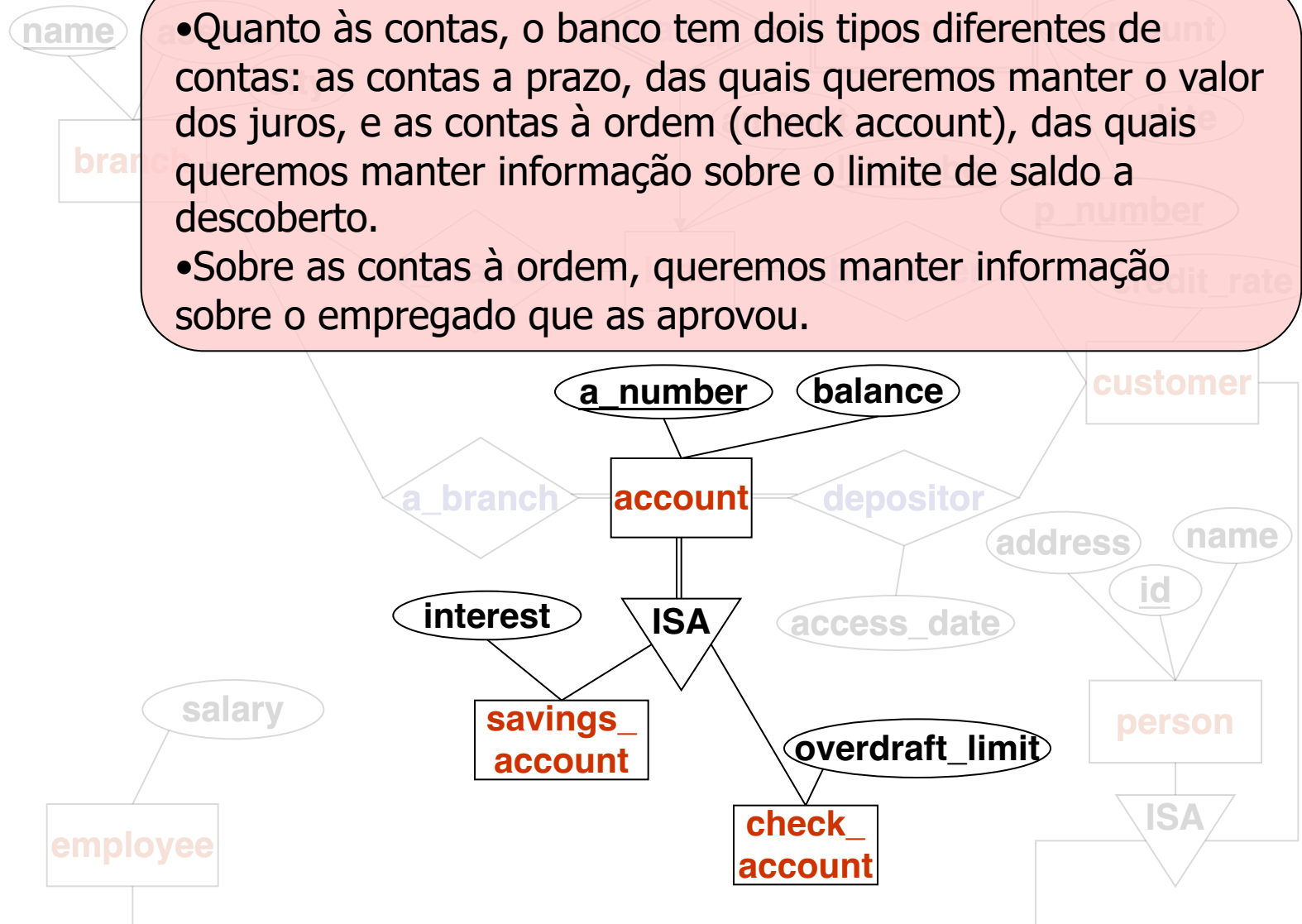
# DER de um Banco

- Quanto às contas, o banco tem dois tipos diferentes de contas: as contas a prazo, das quais queremos manter o valor dos juros, e as contas à ordem (check account), das quais queremos manter informação sobre o limite de saldo a descoberto.
- Sobre as contas à ordem, queremos manter informação sobre o empregado que as aprovou.



# DER de um Banco

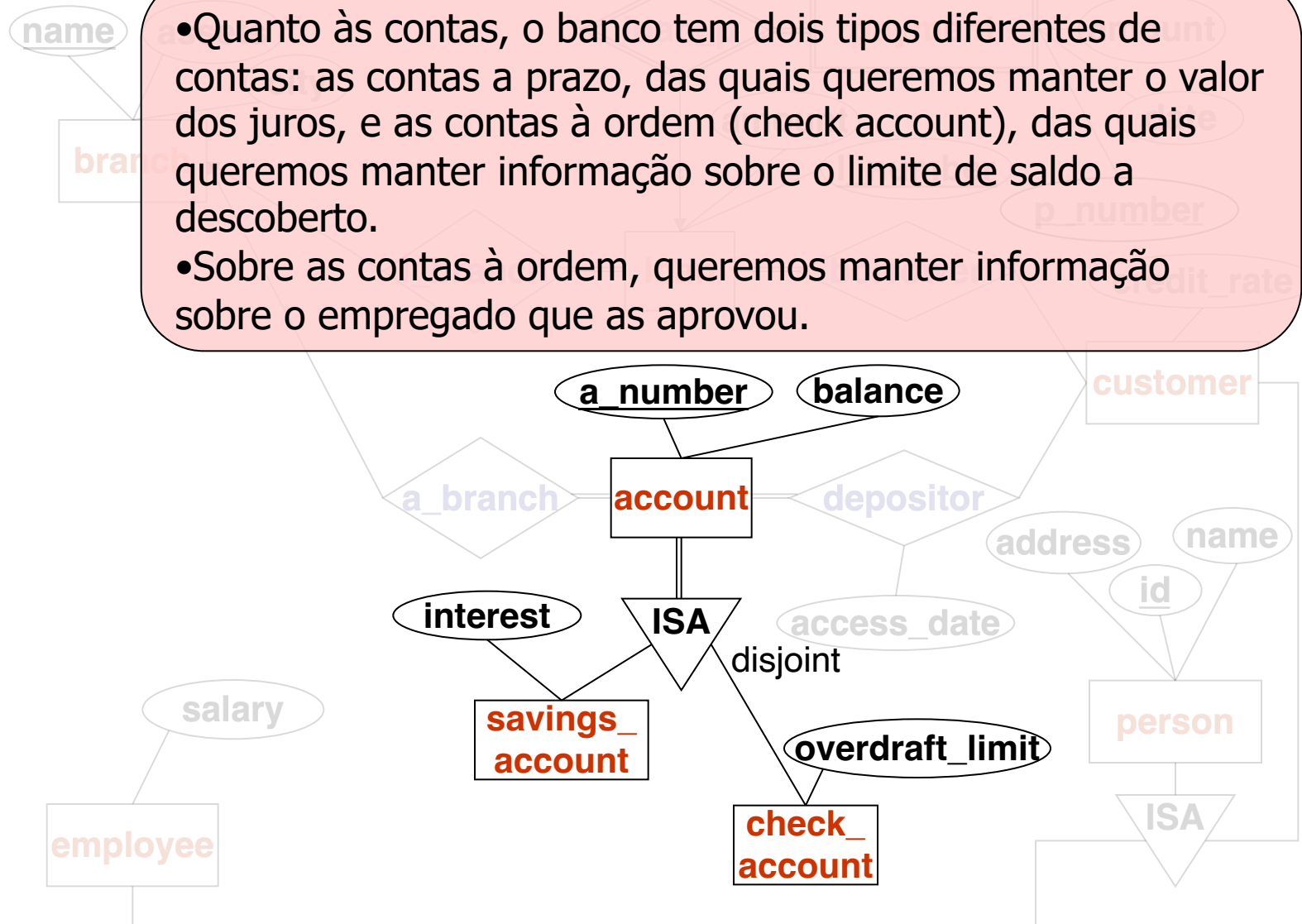
- Quanto às contas, o banco tem dois tipos diferentes de contas: as contas a prazo, das quais queremos manter o valor dos juros, e as contas à ordem (check account), das quais queremos manter informação sobre o limite de saldo a descoberto.
- Sobre as contas à ordem, queremos manter informação sobre o empregado que as aprovou.





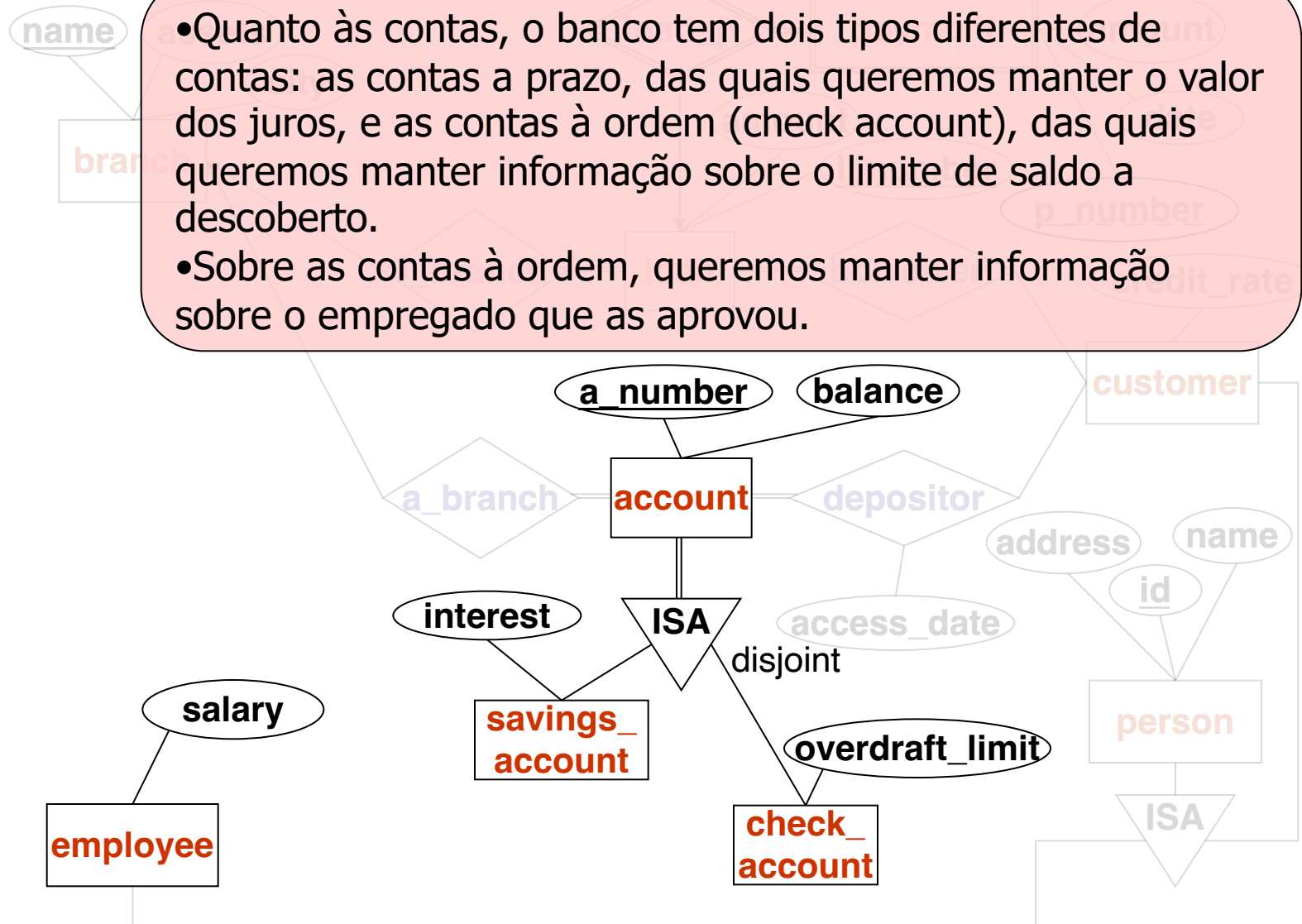
# DER de um Banco

- Quanto às contas, o banco tem dois tipos diferentes de contas: as contas a prazo, das quais queremos manter o valor dos juros, e as contas à ordem (check account), das quais queremos manter informação sobre o limite de saldo a descoberto.
- Sobre as contas à ordem, queremos manter informação sobre o empregado que as aprovou.



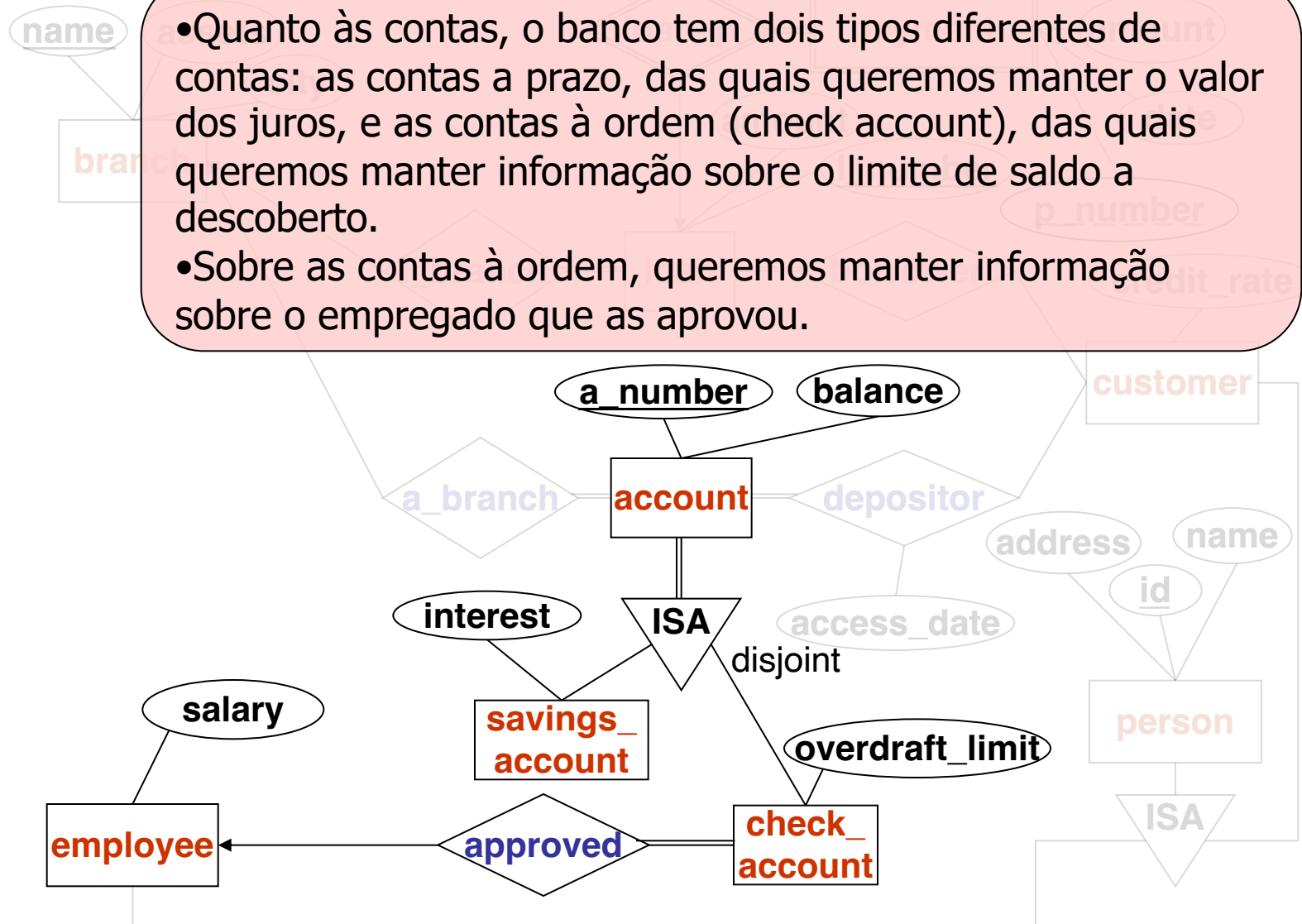
# DER de um Banco

- Quanto às contas, o banco tem dois tipos diferentes de contas: as contas a prazo, das quais queremos manter o valor dos juros, e as contas à ordem (check account), das quais queremos manter informação sobre o limite de saldo a descoberto.
- Sobre as contas à ordem, queremos manter informação sobre o empregado que as aprovou.

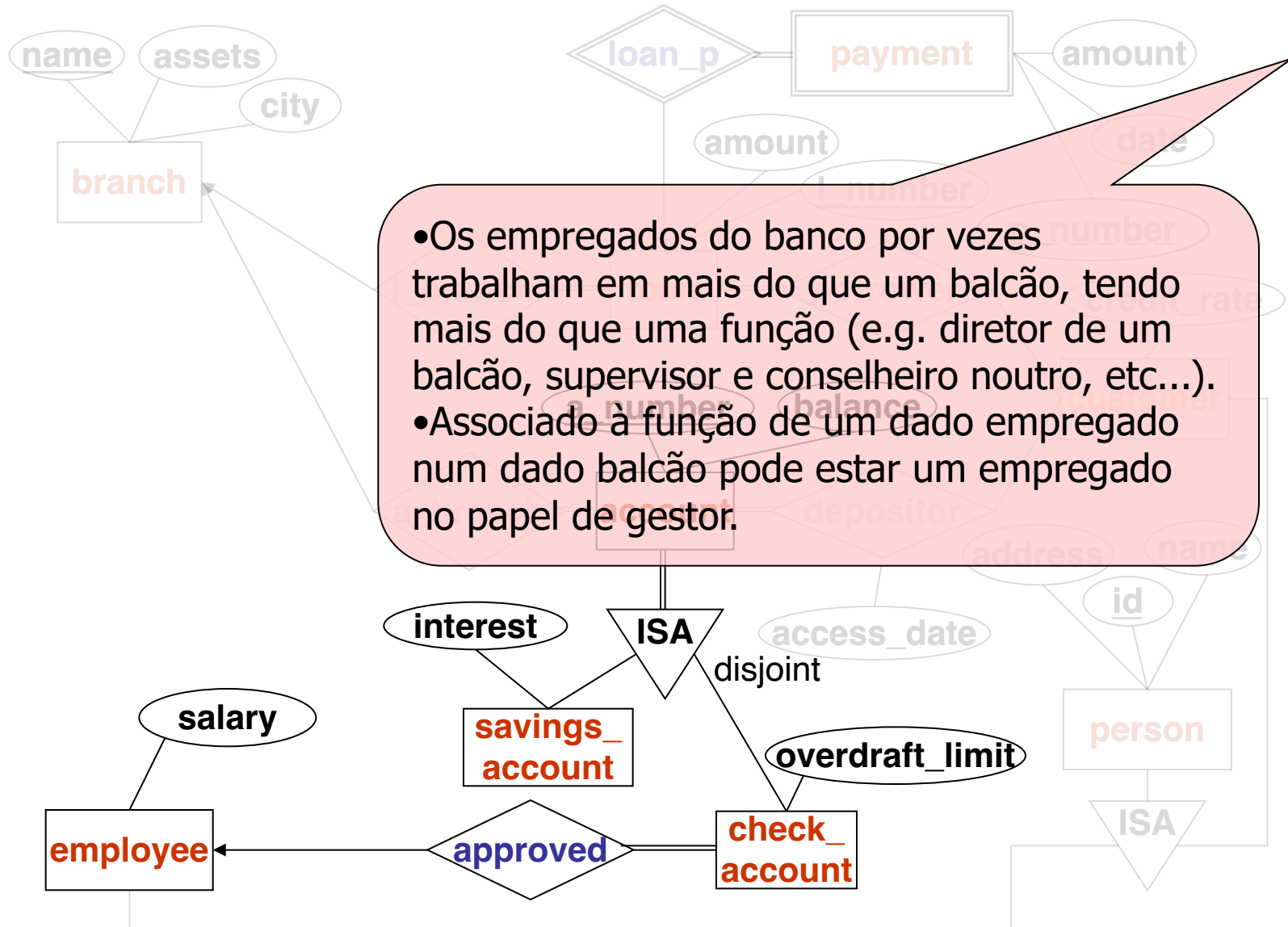


# DER de um Banco

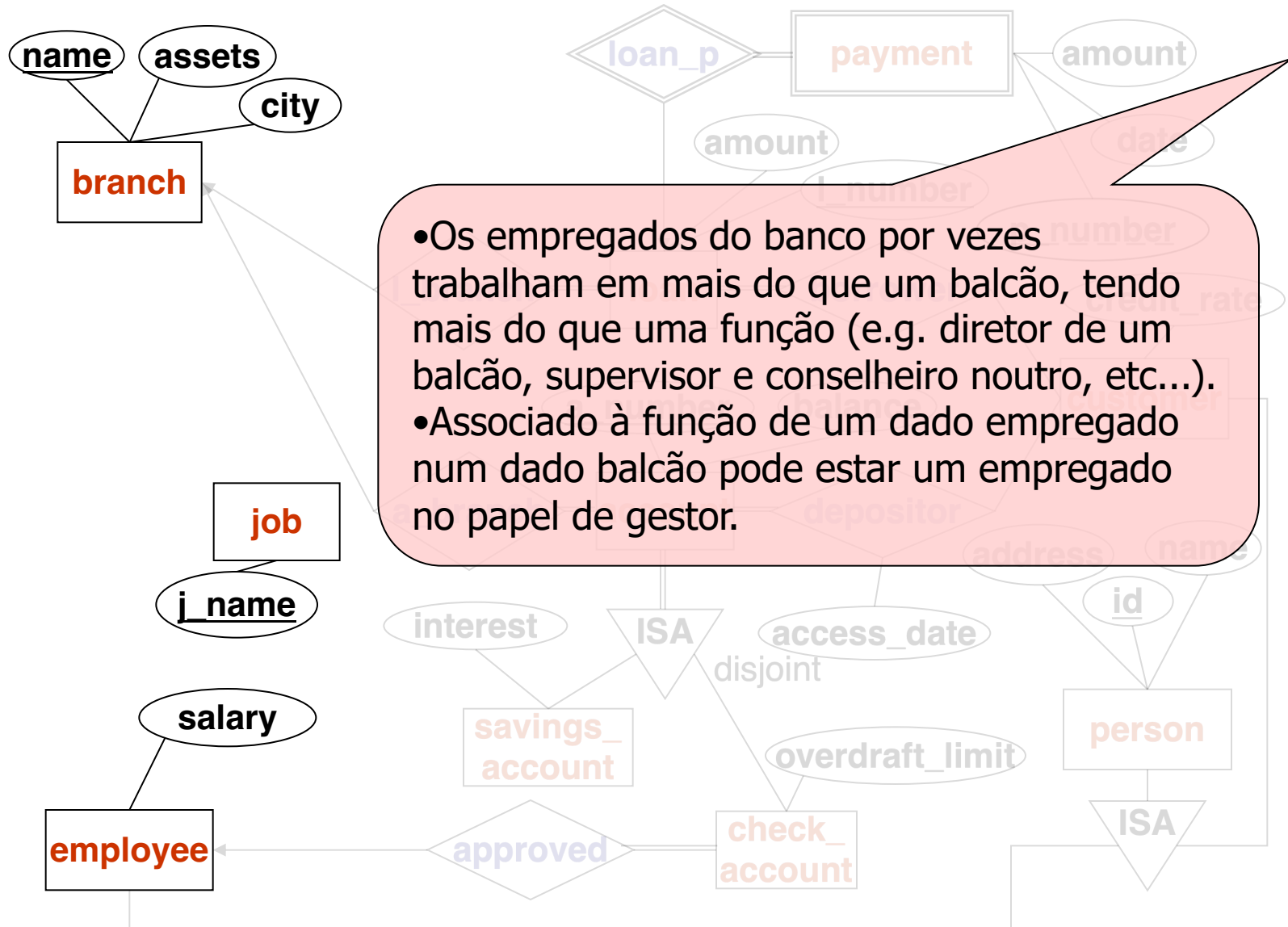
- Quanto às contas, o banco tem dois tipos diferentes de contas: as contas a prazo, das quais queremos manter o valor dos juros, e as contas à ordem (check account), das quais queremos manter informação sobre o limite de saldo a descoberto.
- Sobre as contas à ordem, queremos manter informação sobre o empregado que as aprovou.



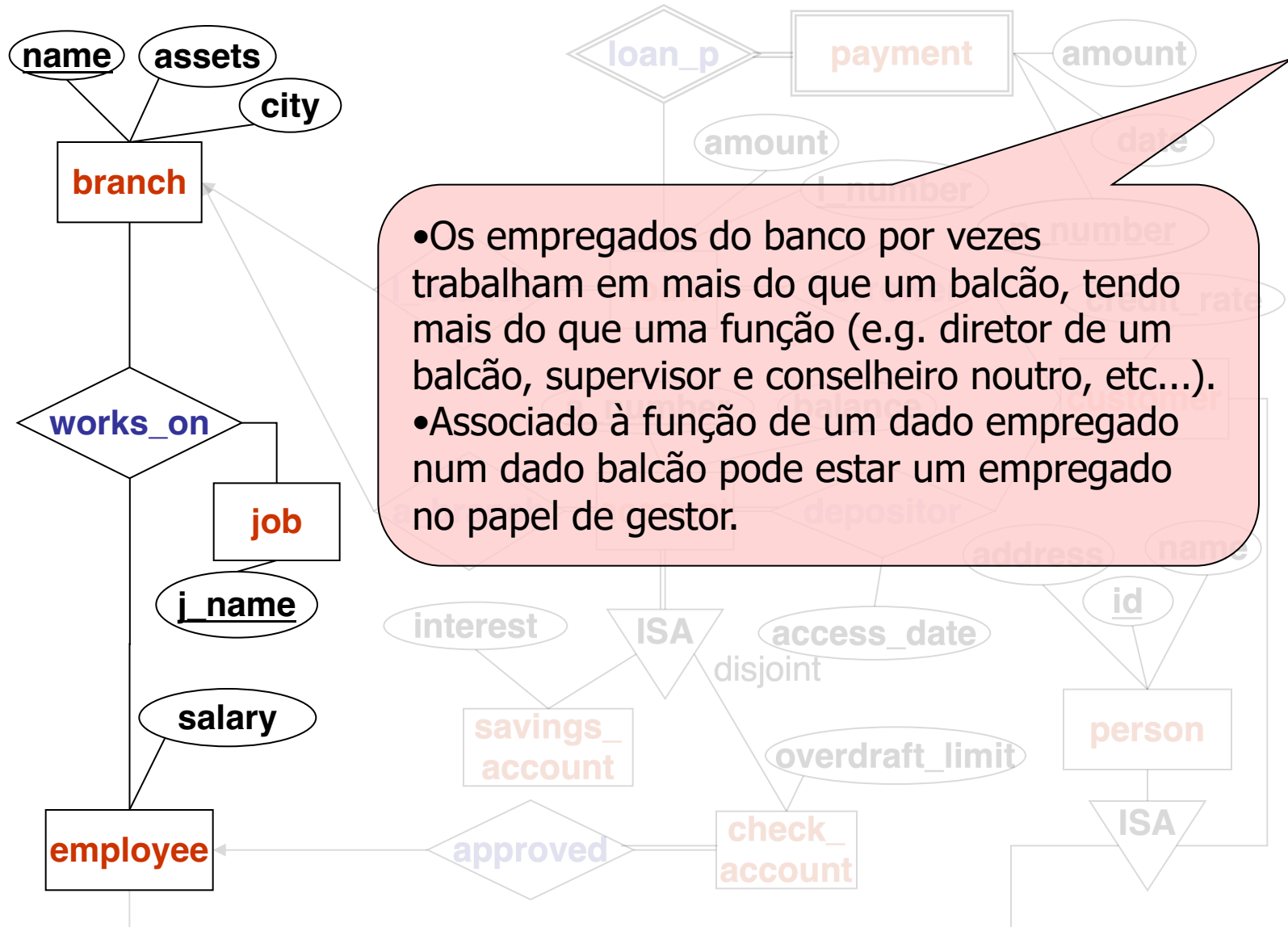
# DER de um Banco



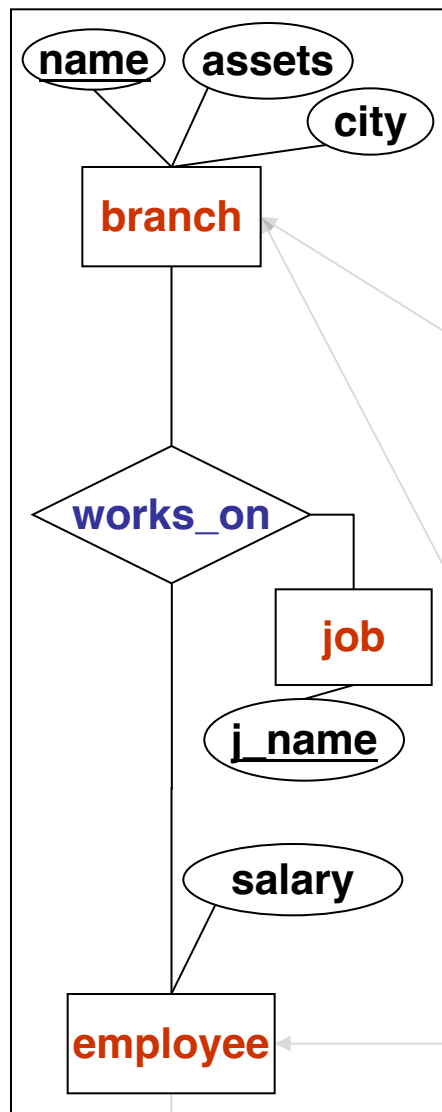
# DER de um Banco



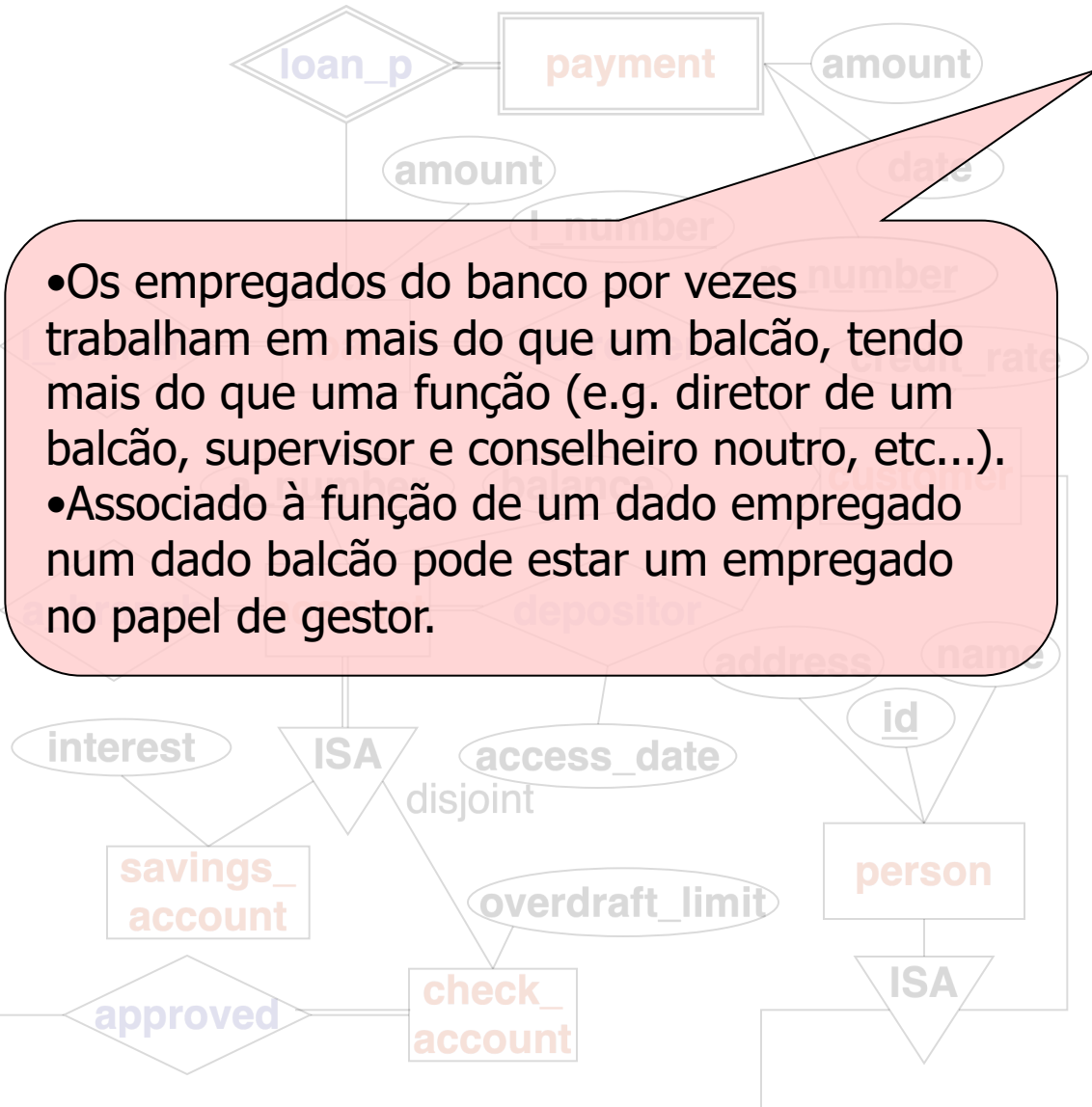
# DER de um Banco



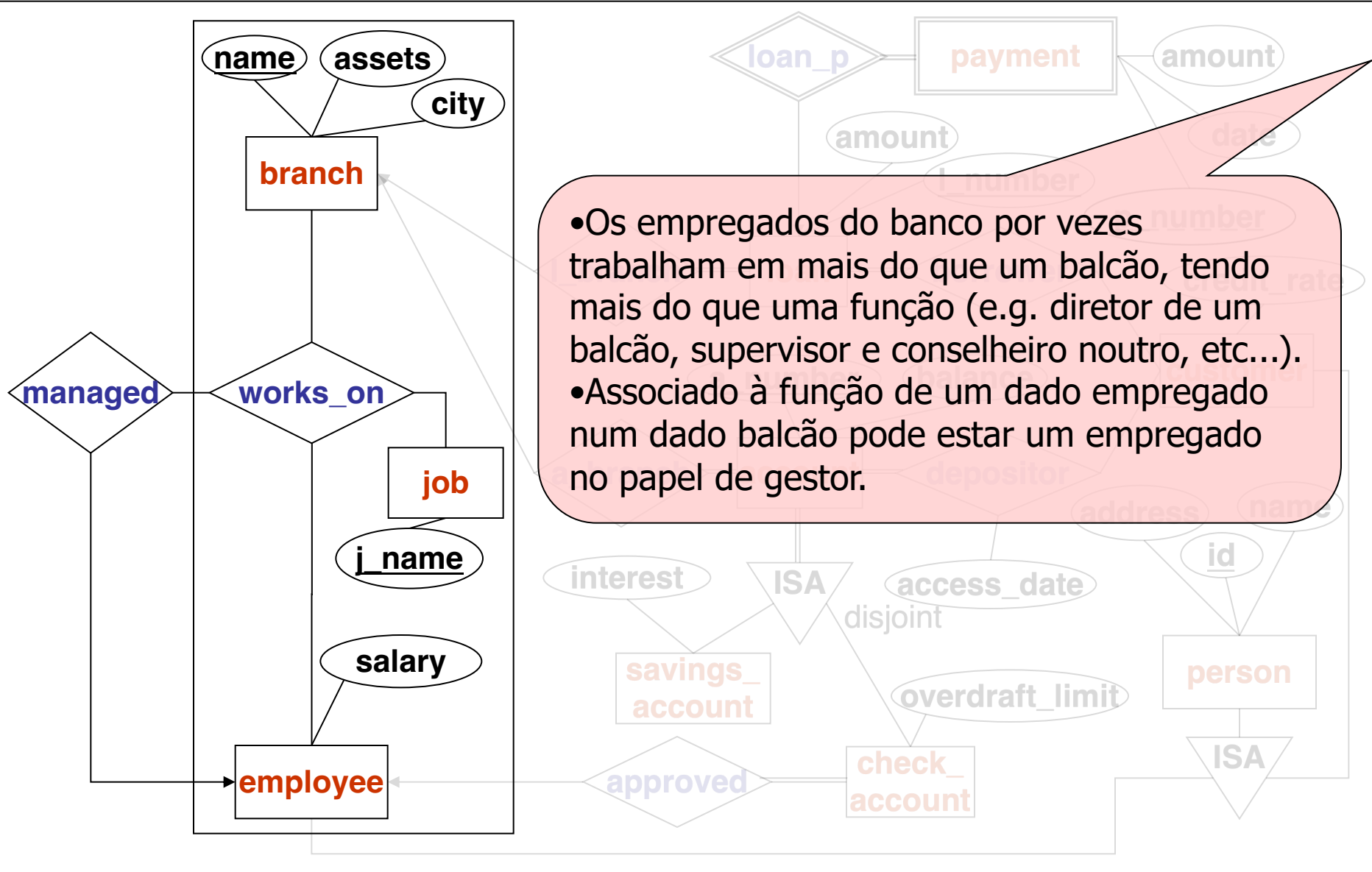
# DER de um Banco



- Os empregados do banco por vezes trabalham em mais do que um balcão, tendo mais do que uma função (e.g. diretor de um balcão, supervisor e conselheiro noutro, etc...).
- Associado à função de um dado empregado num dado balcão pode estar um empregado no papel de gestor.

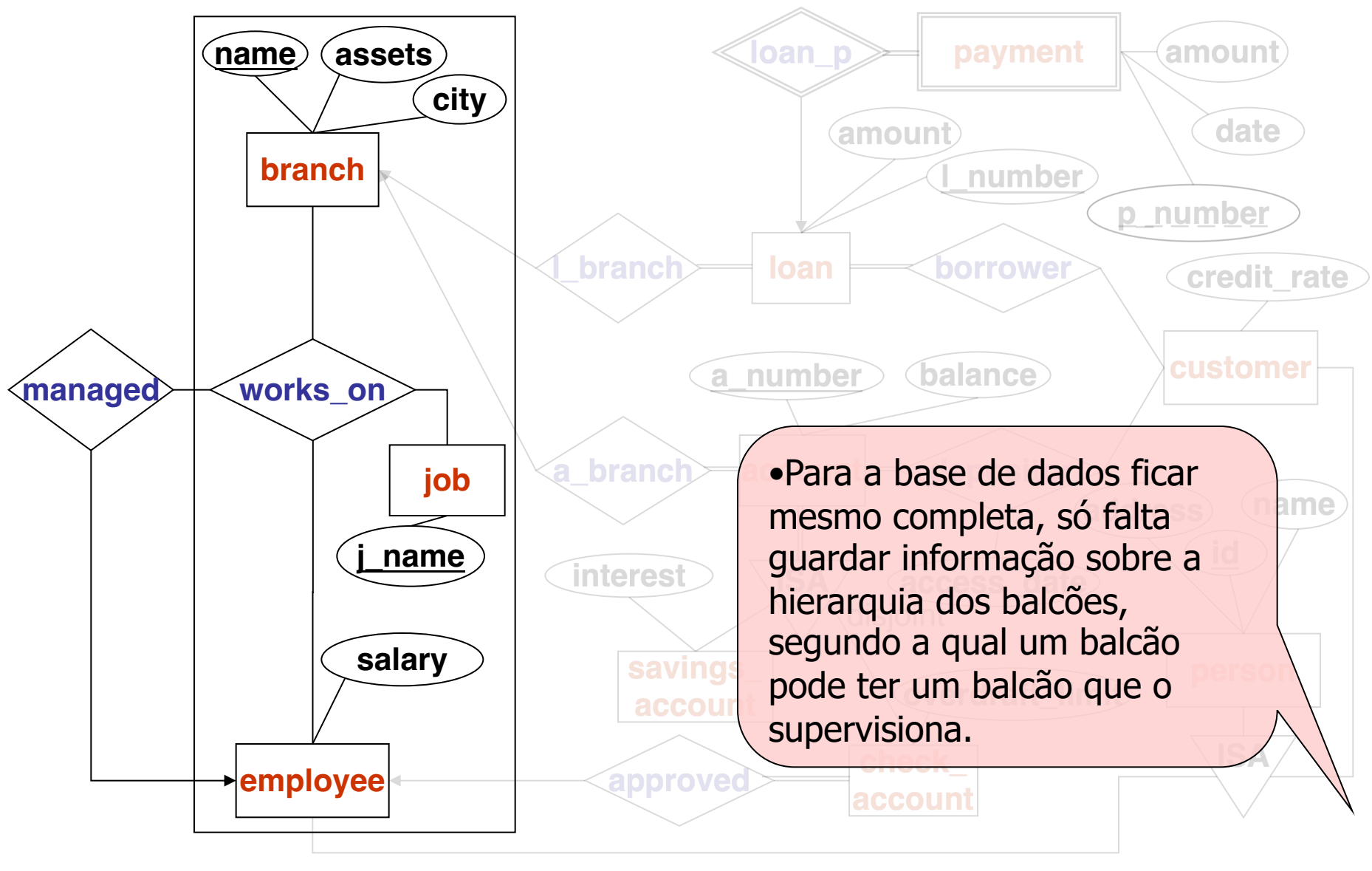


# DER de um Banco

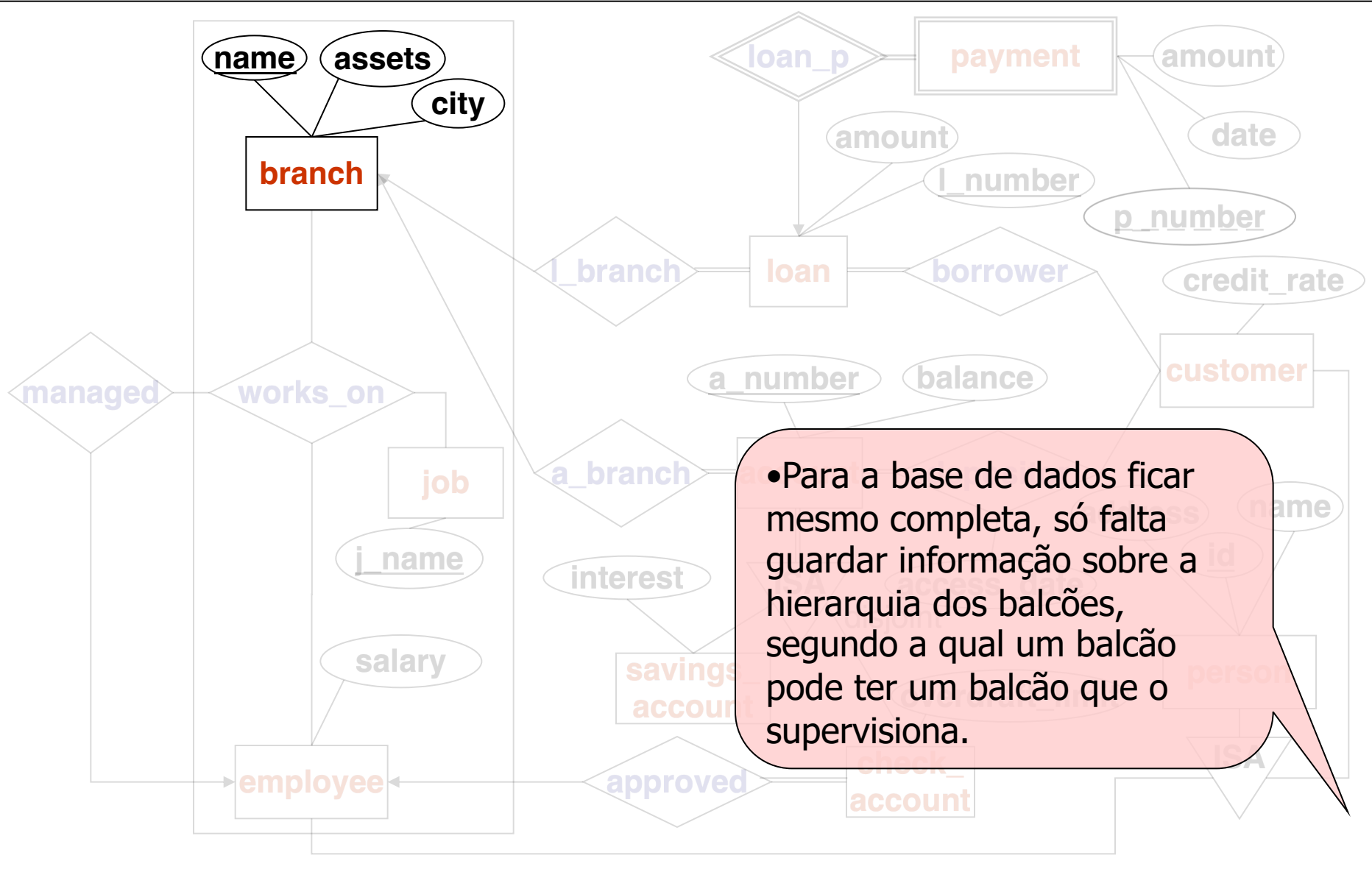




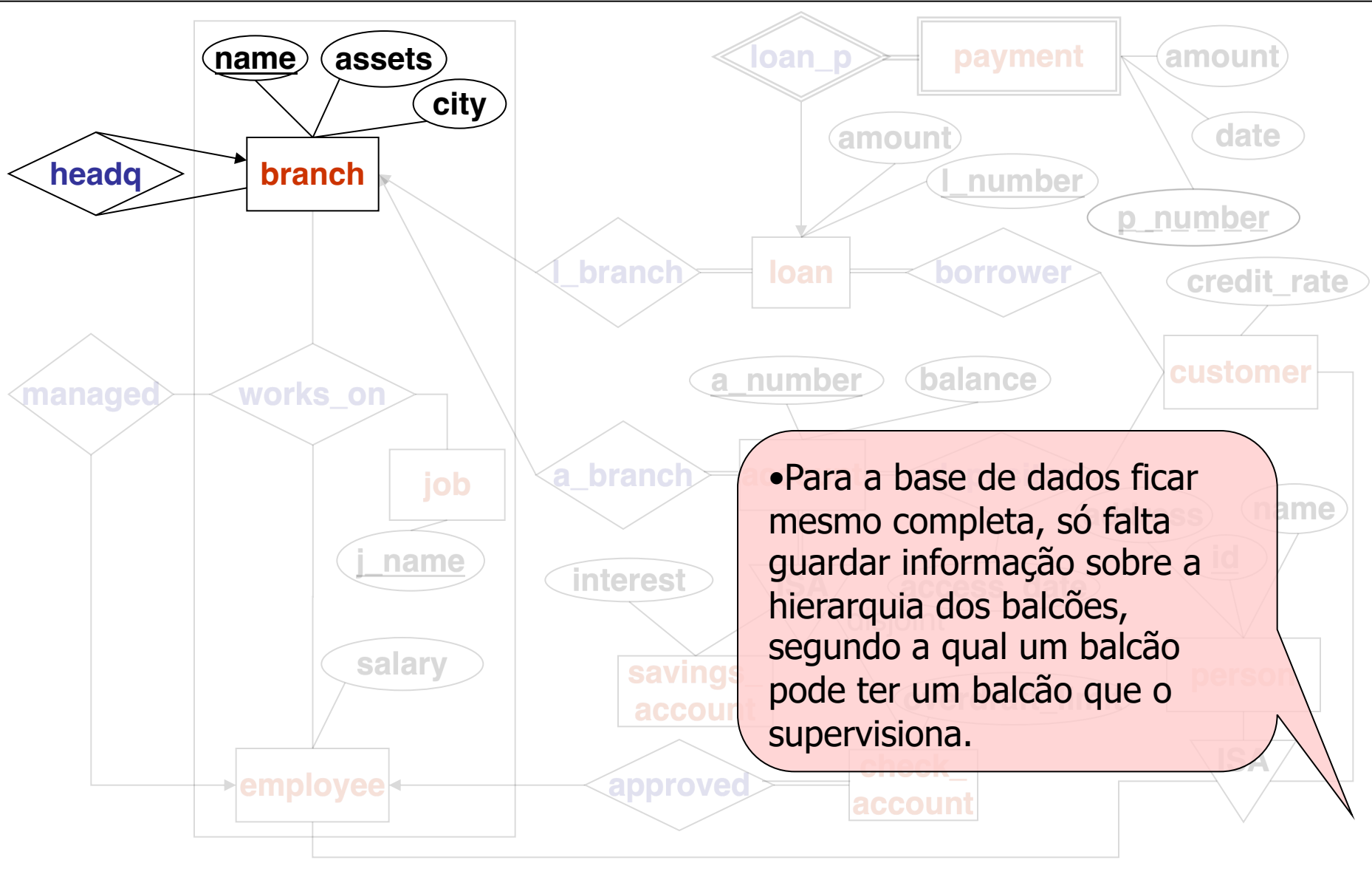
# DER de um Banco



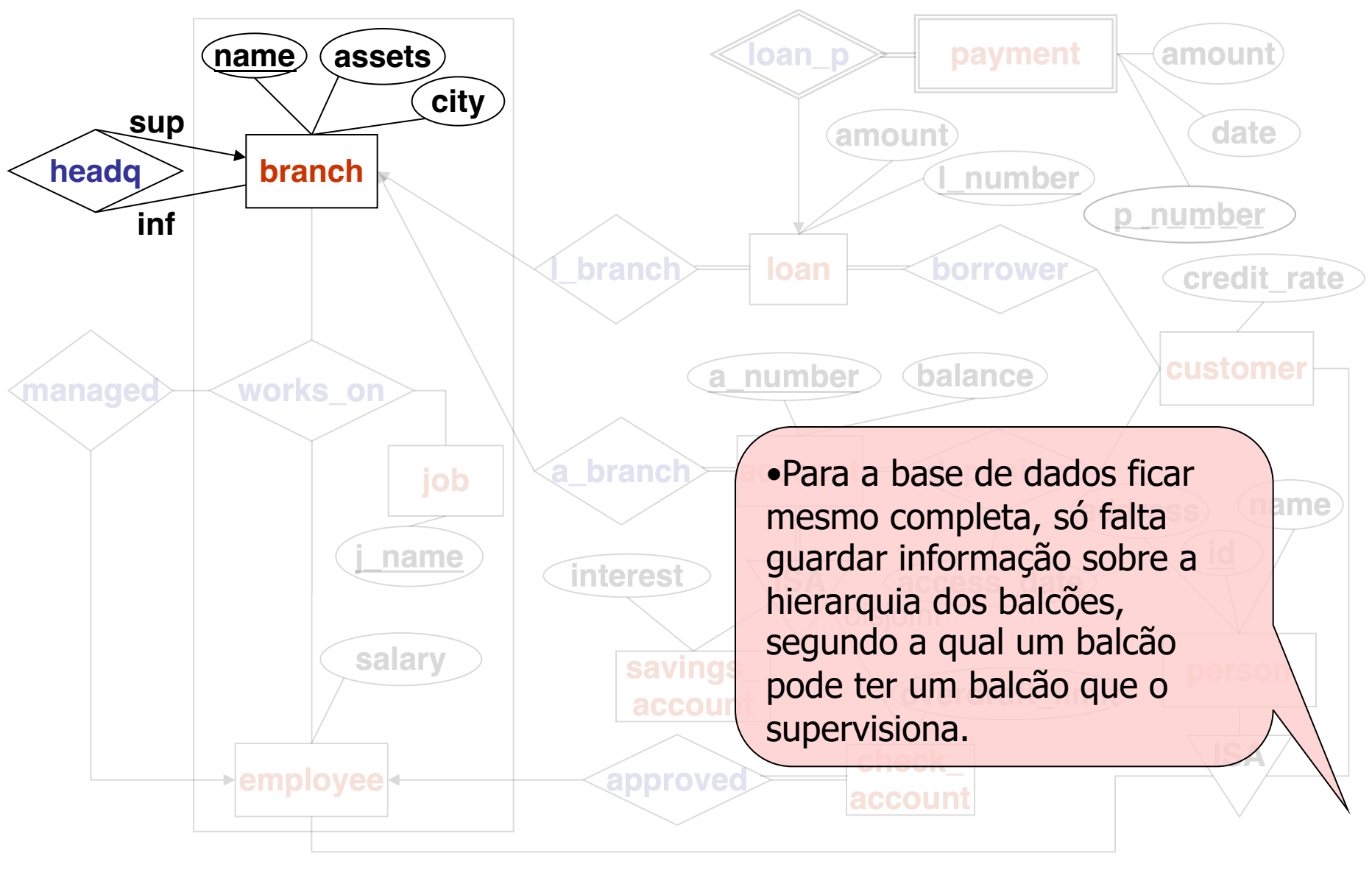
# DER de um Banco



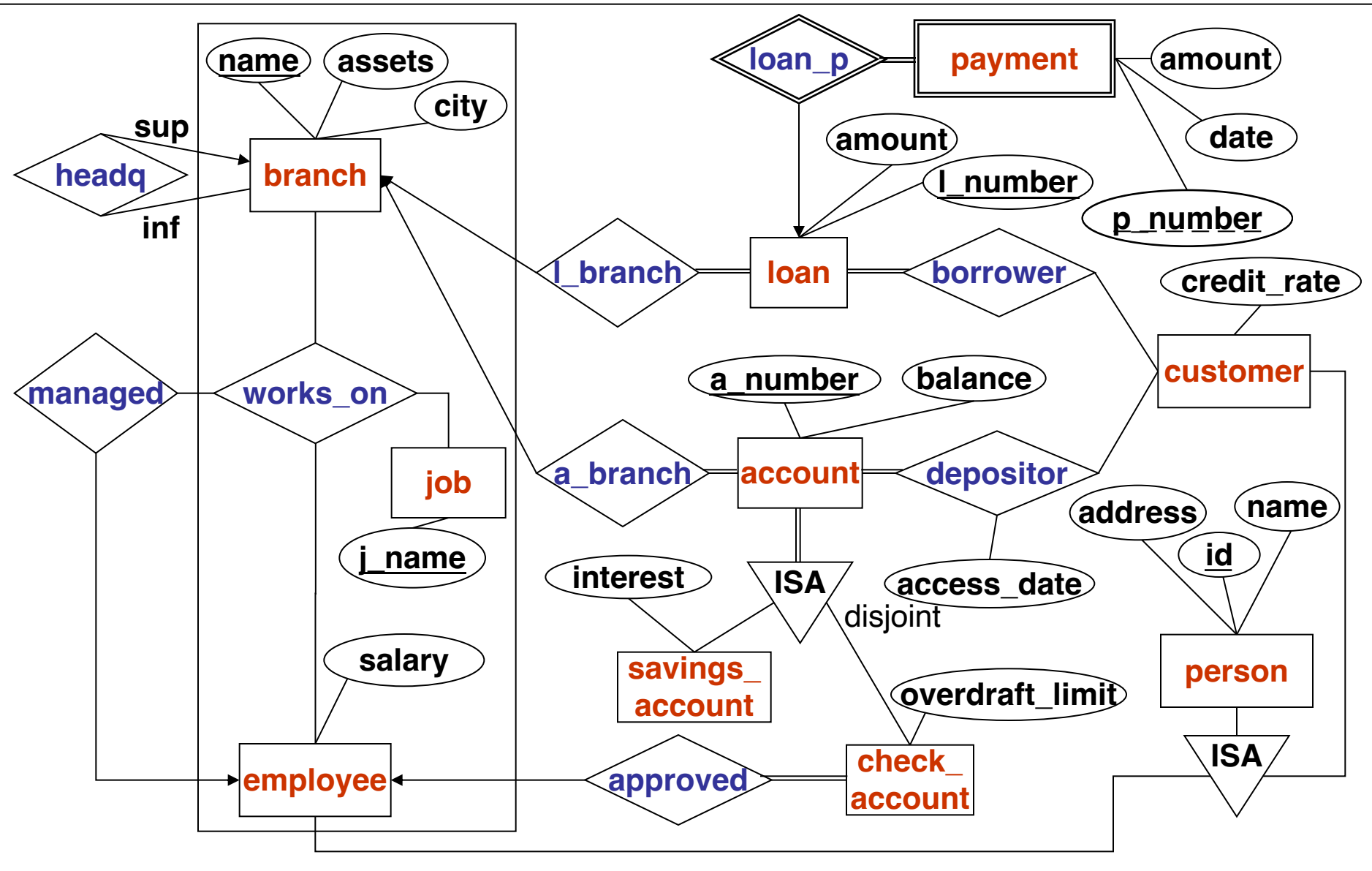
# DER de um Banco



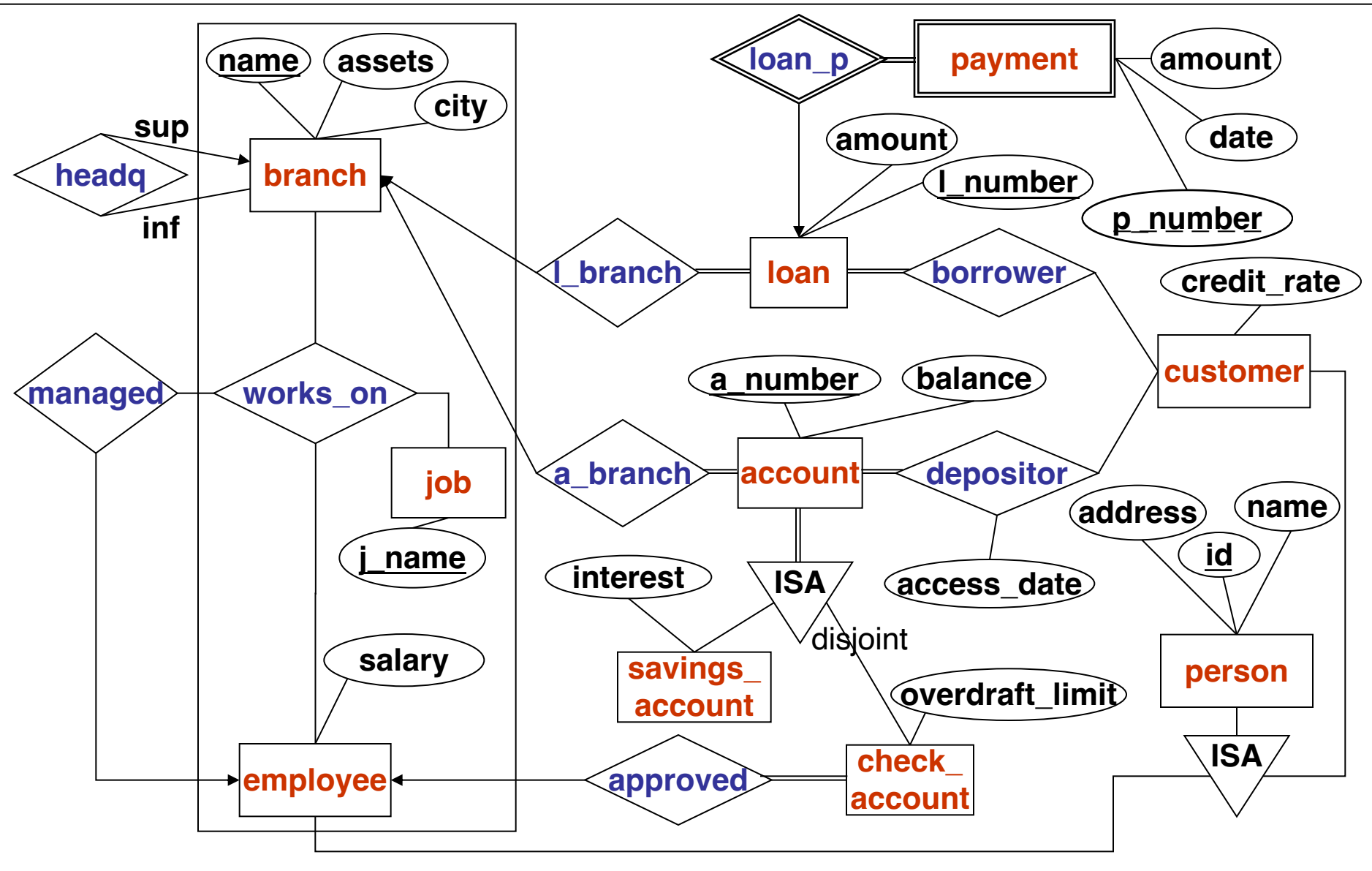
# DER de um Banco



# DER de um Banco

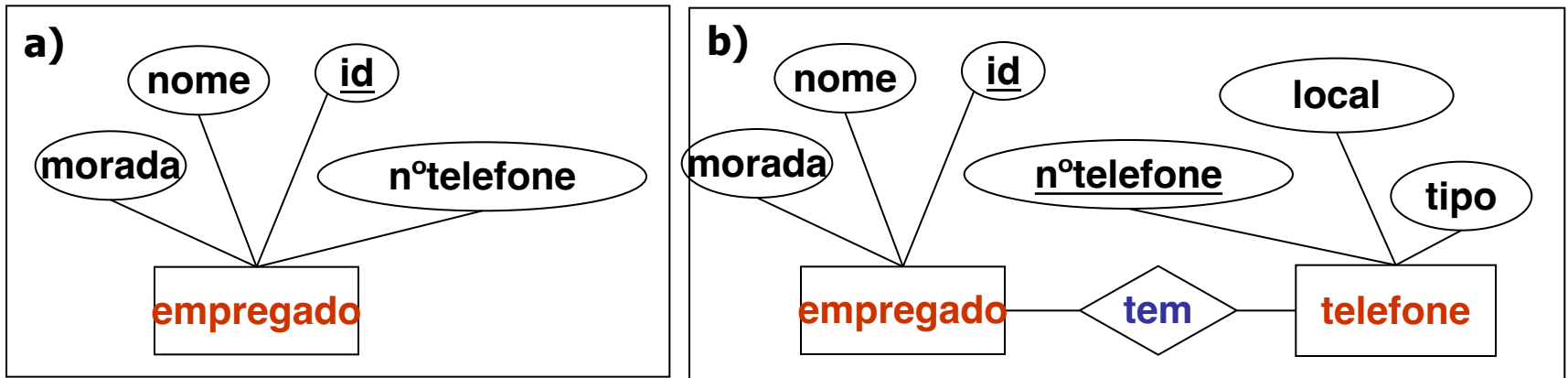


# DER de um Banco



# Opções de desenho

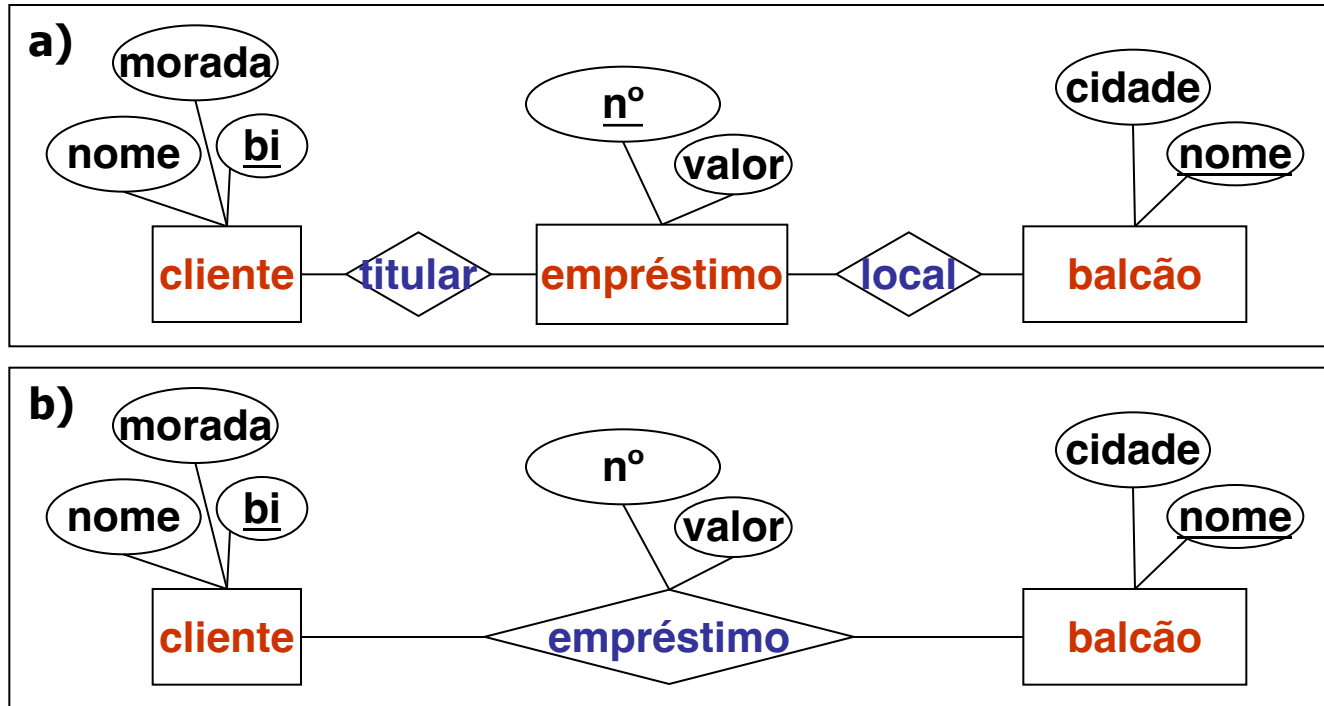
## ■ Atributos vs Conjuntos de Entidades



- Em a), um *empregado* só pode ter um número de telefone, enquanto em b) pode ter vários.
- O modelo b) é mais apropriado quando se pretende manter outras propriedades associadas ao telefone.
- O modelo a) é mais simples enquanto o modelo b) é mais geral.
- A escolha depende do que está a ser modelado.

# Opções de desenho

## ■ Conjuntos de Entidades vs Conjuntos de Relações

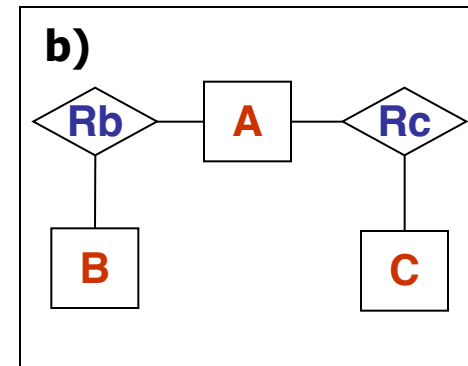
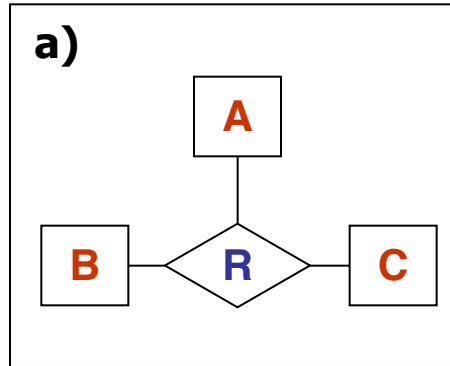


- Se cada empréstimo está associado com exatamente um cliente e está localizado em exatamente um balcão, então o modelo b) poderá servir.
- O modelo b) só permite um empréstimo por cada cliente em cada balcão.
- O modelo b) não é adequado para modelar empréstimos com mais do que um titular. Nesse caso, teríamos de ter uma relação para cada titular, onde replicaríamos a informação relativa ao nº e valor do empréstimo (redundante).



# Opções de desenho

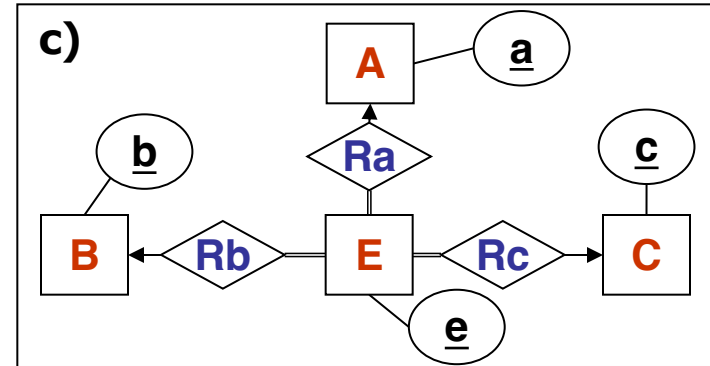
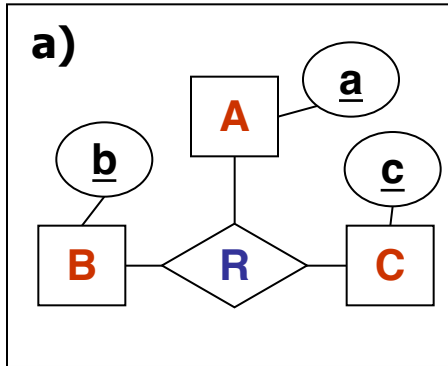
## ■ Conjuntos de relações binários vs. n-ários



- Há relações aparentemente n-árias que são melhor representadas por várias relações binárias.
- Por exemplo, uma relação **R** em a) relacionando uma criança (**A**) com o seu pai (**B**) e a sua mãe (**C**) não é tão conveniente como a representação em b). Em b) é possível incluir informação sobre um dos progenitores sem conhecer o outro.
- Nem sempre é possível representar o modelo a) através do modelo b). Por exemplo, se quisermos representar que um empregado (**A**) tem determinados cargos (**B**) em determinados balcões (**C**) não o podemos fazer com b).

# Opções de desenho

## ■ Conjuntos de relações binários vs. n-ários



- É sempre possível substituir uma relação n-ária por n relações binárias, adicionando uma nova entidade.
- No caso de  $n=3$  representado em a) podemos substituí-lo por c) criando o conjunto de entidades E (com os atributos de R e um novo atributo  $e_i$  que será a sua chave) e, para cada  $(a_i, b_i, c_i) \in R$ , temos  $(e_i, a_i) \in Ra$ ,  $(e_i, b_i) \in Rb$ ,  $(e_i, c_i) \in Rc$ .

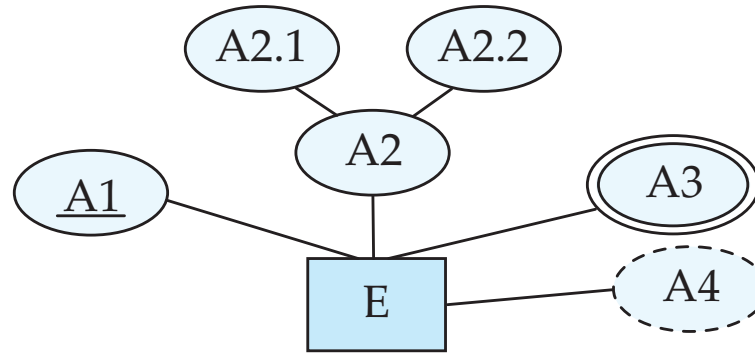
### ■ Desvantagens:

- ✱ Obriga à criação de um novo atributo, **artificial**, que, juntamente com as novas relações vão implicar maior complexidade na implementação.
- ✱ Uma relação n-ária ilustra melhor a participação dos vários conjuntos de entidades na relação.
- ✱ Existem restrições sobre as relações n-árias que não podem ser traduzidas por restrições sobre as relações binárias (e.g. cada par de elementos de B e C só poderem estar associados no máximo com um elemento de A).

# Outras notações ER

## ■ Chen, IDEF1X, ...

entity set E with  
simple attribute A1,  
composite attribute A2,  
multivalued attribute A3,  
derived attribute A4,  
and primary key A1



weak entity set



generalization



total  
generalization

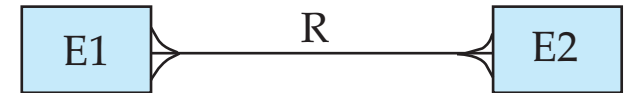
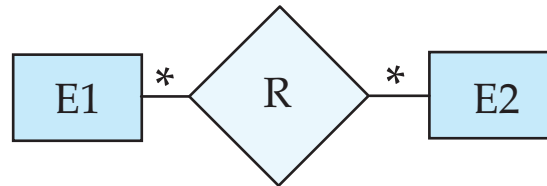


# Outras notações ER

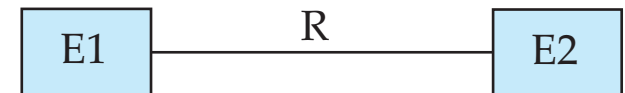
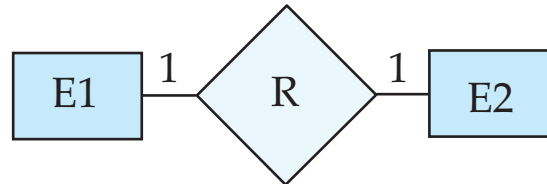
## Chen

## IDEF1X (pés de galinha)

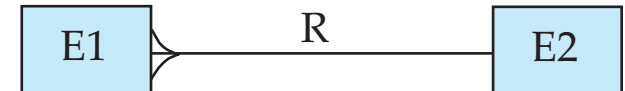
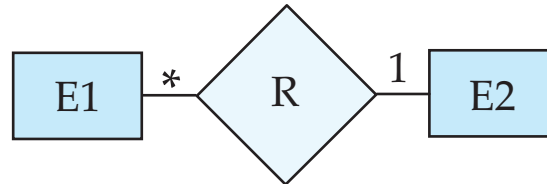
many-to-many  
relationship



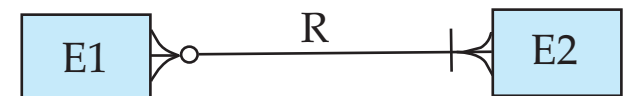
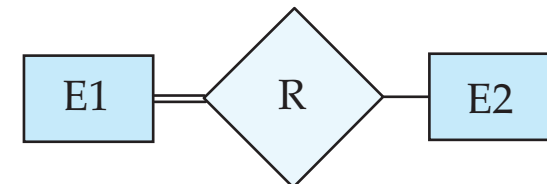
one-to-one  
relationship



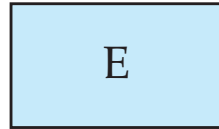
many-to-one  
relationship



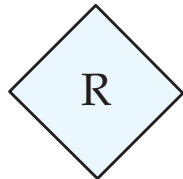
participation  
in R: total (E1)  
and partial (E2)



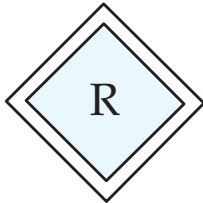
# Notação ER utilizada na última edição do livro



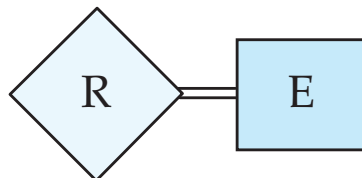
entity set



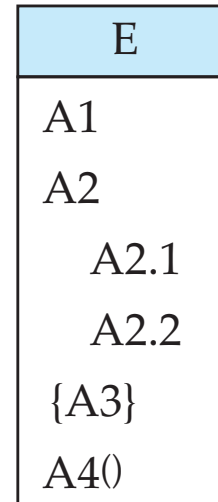
relationship set



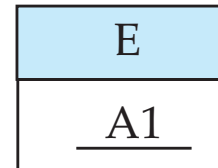
identifying  
relationship set  
for weak entity set



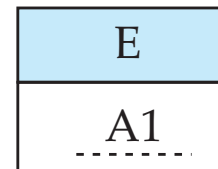
total participation  
of entity set in  
relationship



attributes:  
simple (A1),  
composite (A2) and  
multivalued (A3)  
derived (A4)

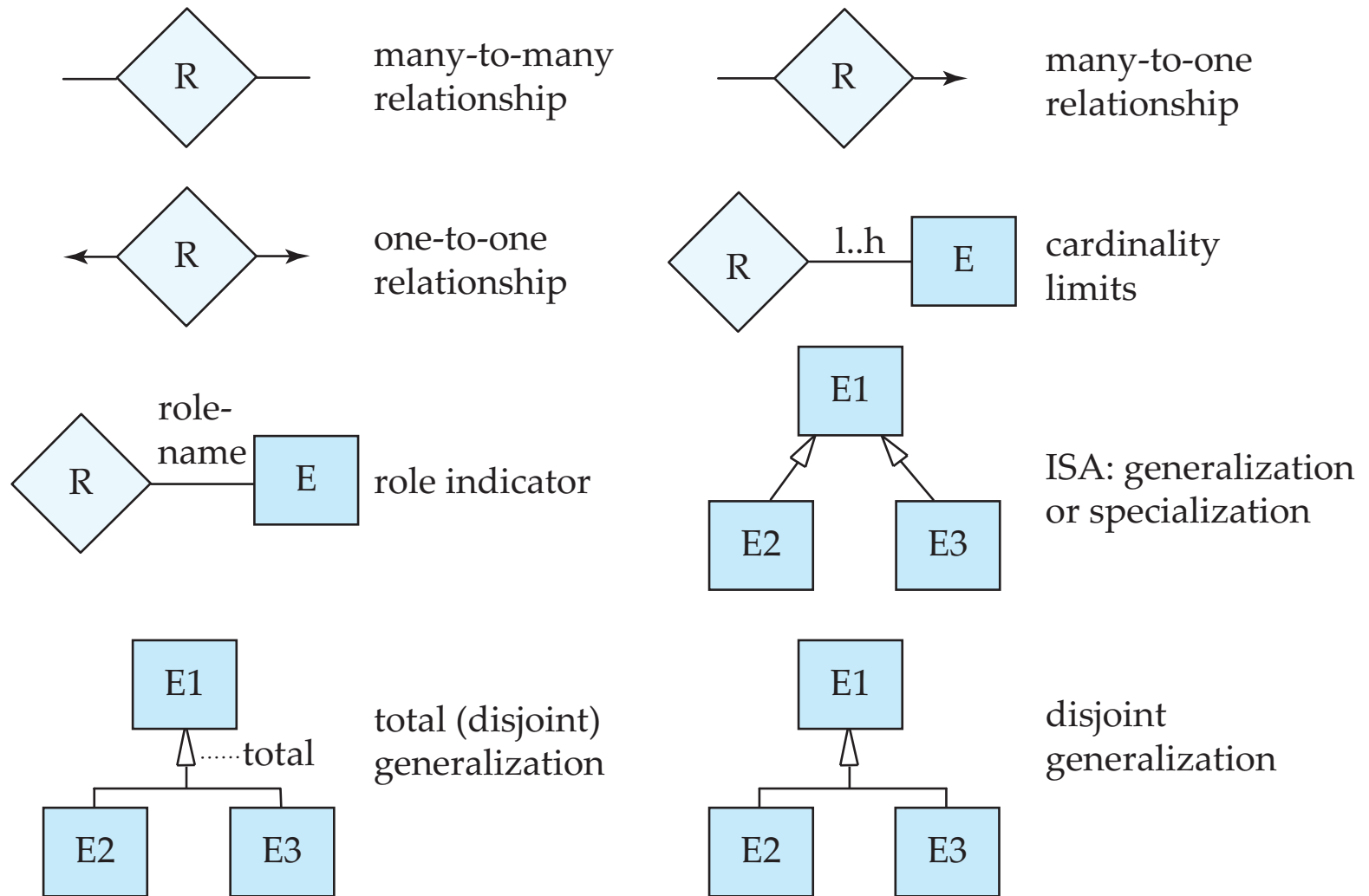


primary key



discriminating  
attribute of  
weak entity set

# Notação ER utilizada na última edição do livro



# Exemplo: concursos (Teste 1 22/23)

Uma agência governamental pretende criar uma base de dados sobre as candidaturas aos vários concursos diferentes que promove para financiamento da Ciência. Cada um dos concursos é identificado por um nome, tem uma tipologia e é realizado em edições anuais com um limite máximo de budget que pode variar conforme o ano.

As candidaturas são submetidas a uma edição de um concurso, e pretendemos guardar da candidatura o id, a área científica e o budget. As candidaturas são individuais (por exemplo a bolsas) ou de projeto, e no segundo caso queremos guardar o nome do projeto.

Os investigadores são identificados por um researchID e queremos guardar as suas afiliações a centros de investigação, dos quais queremos manter o seu nome e um id.

Para as candidaturas individuais queremos guardar quem é o investigador proponente e a sua afiliação única na altura da candidatura, e para os projetos quem são os investigadores participantes e quem é o investigador responsável, em ambos os casos junto a respetiva afiliação única na altura da candidatura.