

ACH2025

Laboratório de Bases de Dados

Aula 2

Revisão de Conceitos

Professora:

➤ Fátima L. S. Nunes



Conceitos Básicos

- **Campo** – representação informatizada de um dado real / menor unidade de informação com valor significativo para o usuário
- **Dado** – conteúdo do campo
- **Registro** – conjunto de campos
- **Arquivo** – conjunto de registros
- **Banco de Dados** – armazenamento físico dos arquivos
- **Sistema Gerenciador de Banco de Dados (SGBD)** – *software* responsável pelo armazenamento e recuperação dos dados do BD.



Problemas dos Sistemas de Arquivos Tradicionais

- **Redundância**
- **Inconsistência**
- **Dificuldade no acesso a dados**
- **Isolamento dos dados**
- **Múltiplos usuários**
- **Segurança**
- **Integridade**
- **Atomicidade**

Problemas dos Sistemas de Arquivos Tradicionais

- **Redundância**

- Arquivos e aplicações criados e mantidos por diferentes programadores → arquivos com formatos diferentes e programas escritos em diversas linguagens de programação.
- Informação **repetida** em diversos lugares (arquivos).

- **Inconsistência**

- Decorrência da redundância.
- Várias cópias dos dados poderão divergir ao longo do tempo.



Problemas dos Sistemas de Arquivos Tradicionais

- **Dificuldade no acesso aos dados**
 - Necessidade de construir programas para obter qualquer informação solicitada.

Exemplo

Uma empresa precisa dos nomes de todos os clientes que fazem aniversário no mês de fevereiro, mas esta solicitação não foi prevista no projeto do sistema → existe somente uma aplicação para gerar a relação de todos os clientes da empresa.

Alternativas:

- 1) separar manualmente da lista de todos os clientes aqueles que necessita*
- 2) requisitar um programador para escrever o programa necessário.*

Ambas alternativas são insatisfatórias.

...Mais tarde a empresa precisa saber os clientes que têm saldo negativo.



Problemas dos Sistemas de Arquivos Tradicionais

- **Isolamento dos dados**
 - Dados dispersos em vários arquivos e arquivos em diferentes formatos → difícil escrever novas aplicações para recuperação apropriada dos dados.

Problemas dos Sistemas de Arquivos Tradicionais

- **Múltiplos usuários**

- Atualizações concorrentes podem resultar em inconsistências.

Exemplo

Conta corrente com saldo = R\$500,00

Dois clientes debitam da conta A simultaneamente (\$50 e \$100, respectivamente)

Na execução dos programas, ambos lêem o saldo antigo, retiram, cada um seu valor correspondente, sendo o resultado armazenado.

Dependendo de qual deles registre seu resultado primeiro, o saldo da conta A será \$450 ou \$400, ao invés do valor correto de \$350.

Problemas dos Sistemas de Arquivos Tradicionais

- **Segurança**
 - Definir autorizações de acesso a diferentes usuários.
- **Integridade**
 - Valores dos dados armazenados devem satisfazer a certas restrições para manutenção da **consistência**.

O valor da nota final de um aluno deve estar entre 0 e 10. Os programadores determinam o cumprimento desta restrição através da adição de código apropriado aos vários programas aplicativos. Entretanto, quando aparecem novas restrições, é difícil alterar todos os programas para incrementá-las. O problema é ampliado quando as restrições atingem diversos itens de dados em diferentes arquivos.

Exemplo



Problemas dos Sistemas de Arquivos Tradicionais

- **Atomicidade**

- Algumas operações devem ser feitas de forma única, atômica, a fim de assegurar a integridade e consistência dos dados.

um programa para transferir R\$50,00 da conta A para uma conta B. Se ocorrer falha no sistema durante sua execução, é possível que os 50 reais sejam debitados da conta A sem serem creditados na conta B, criando um estado inconsistente no banco de dados.

Exemplo

É essencial para a consistência do banco de dados que ambos, débito e crédito ocorram, ou nenhum deles seja efetuado. Isto é, a transferência de fundos deve ser uma operação *atômica* – deve ocorrer por completo, ou não ocorrer.

Regras para que um sistema de manipulação de dados seja um SGBD

- **Auto-contenção** – conter dados, suas descrições, relacionamentos e formas de acesso.
- **Independência dos Dados** – aplicações imunes a mudanças na estrutura de armazenamento e à estratégia de acesso a dados.
- **Abstração dos Dados** – usuário não precisa saber detalhes sobre armazenamento real.
- **Visões** – formas diferentes de ver os dados de acordo com necessidade dos usuários.
- **Transações** – gerenciar integridade sem precisar de aplicativos.
- **Controle automático de acesso** – vários usuários, travamento eficiente.



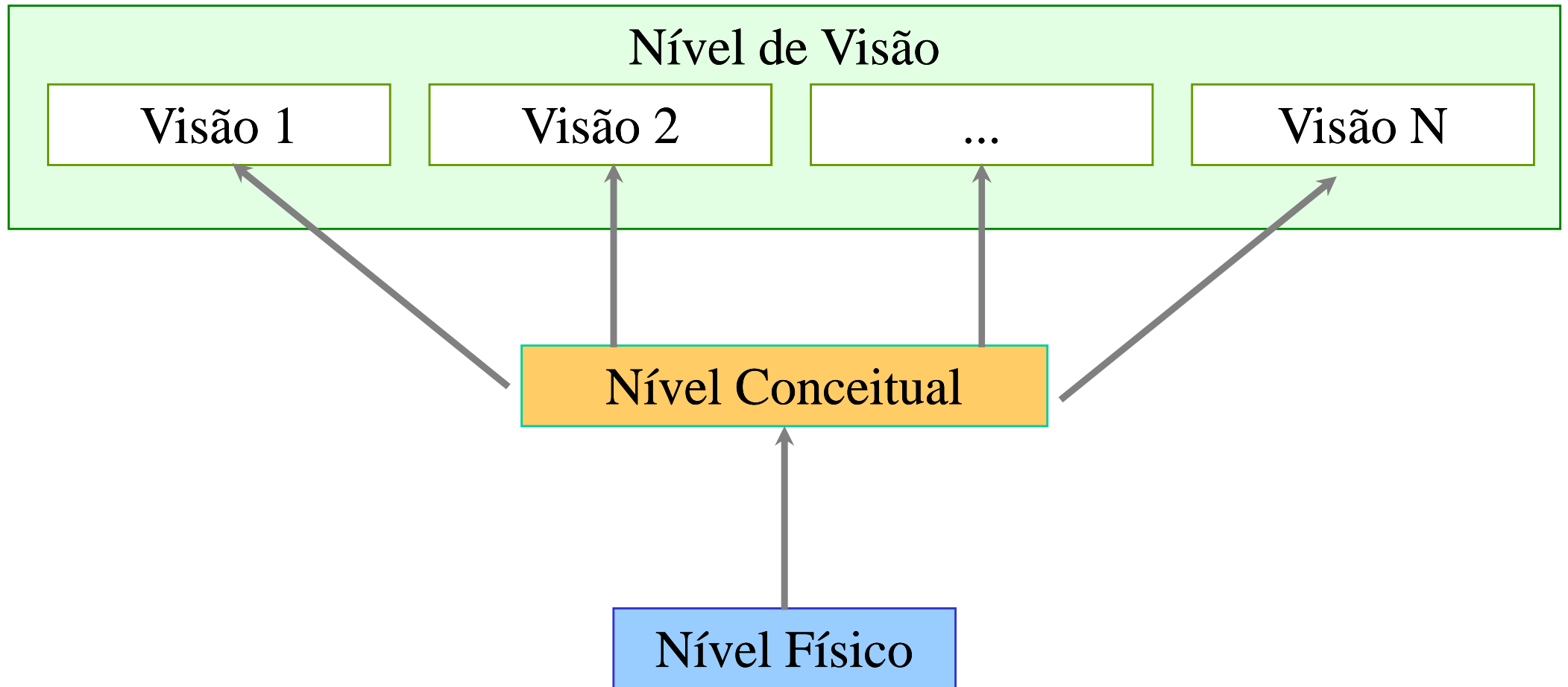
Características de um SGBD

- **Controle de Redundâncias** – informações armazenadas em um único lugar.
- **Compartilhamento dos Dados** – garantir concorrência ao acesso dos dados, sem erro.
- **Controle de Acesso** – seleção de permissões por usuário.
- **Interfaceamento** – facilidade para recuperação de informação.
- **Esquematização** – mecanismos que possibilitem a compreensão do relacionamento entre as tabelas e sua manutenção.
- **Controle de Integridade** – aplicações e acessos não podem comprometer integridade dos dados.
- **Backups** – facilidade para recuperar falhas de *hardware* e *software*.

Abstração de Dados

- **SGBD:** estruturas complexas para a representação dos dados no banco de dados.
- Complexidade precisa ser escondida dos usuários.
- **Níveis de abstração:** grande vantagem dos SGBD
 - **Nível físico:** mais baixo – descreve **como** os dados são armazenados
 - **Nível conceitual ou lógico:** **quais** dados são armazenados e quais relacionamentos entre eles.
 - **Nível visão:** expõe apenas parte do banco de dados.

Abstração de Dados



Instâncias e Esquemas

- **instância** → conjunto de informações contidas em determinado BD em um dado momento.
- **esquema** → projeto geral do BD → mudados com pouca frequência.
- Analogia com linguagem de programação:

```
var ClienteNovo: cliente
```

Área de memória que contém um registro do tipo cliente.
Valor de ClienteNovo em um dado momento é análogo à **instância**.
Definição do tipo é análogo ao **esquema**.

```
Type cliente = record  
  cliente-nome: string;  
  cliente-CPF: string;  
  cliente-endereco: string;  
  cliente-cidade: string  
end;
```



Independência de Dados

Capacidade de modificar a definição dos esquemas em determinado nível, sem afetar o esquema do nível superior

- Independência de dados física: modifica o esquema físico sem que, com isso, qualquer programa aplicativo precise ser reescrito (ocasionais para aumento de desempenho).
- Independência de dados lógica: modifica o esquema lógico sem que, com isso, qualquer programa aplicativo precise ser reescrito (sempre que uma estrutura lógica do BD é alterada)
 - ➔ mais difícil de ser atingida.

Linguagens de Banco de Dados

- *SGBDs:*
 - *Uma linguagem para os esquemas (DDL)*
 - *Uma linguagem para consultas e atualizações (DML)*
- Data Definition Language (DDL):
 - Especificação do esquema de dados
 - resultado da compilação de instruções DDL → conjunto de tabelas que constituem o dicionário de dados ou diretório de dados
 - **dicionário de dados**: arquivo de *metadados* (dados a respeito de dados)
→ no SGBD o diretório é consultado antes que o dado real seja modificado
 - Estrutura de memória e método de acesso usados pelo BD: DDL especial denominada (linguagem de definição e armazenamento de dados – *data storage and definition language*)

Linguagem de Manipulação de Dados (DML)

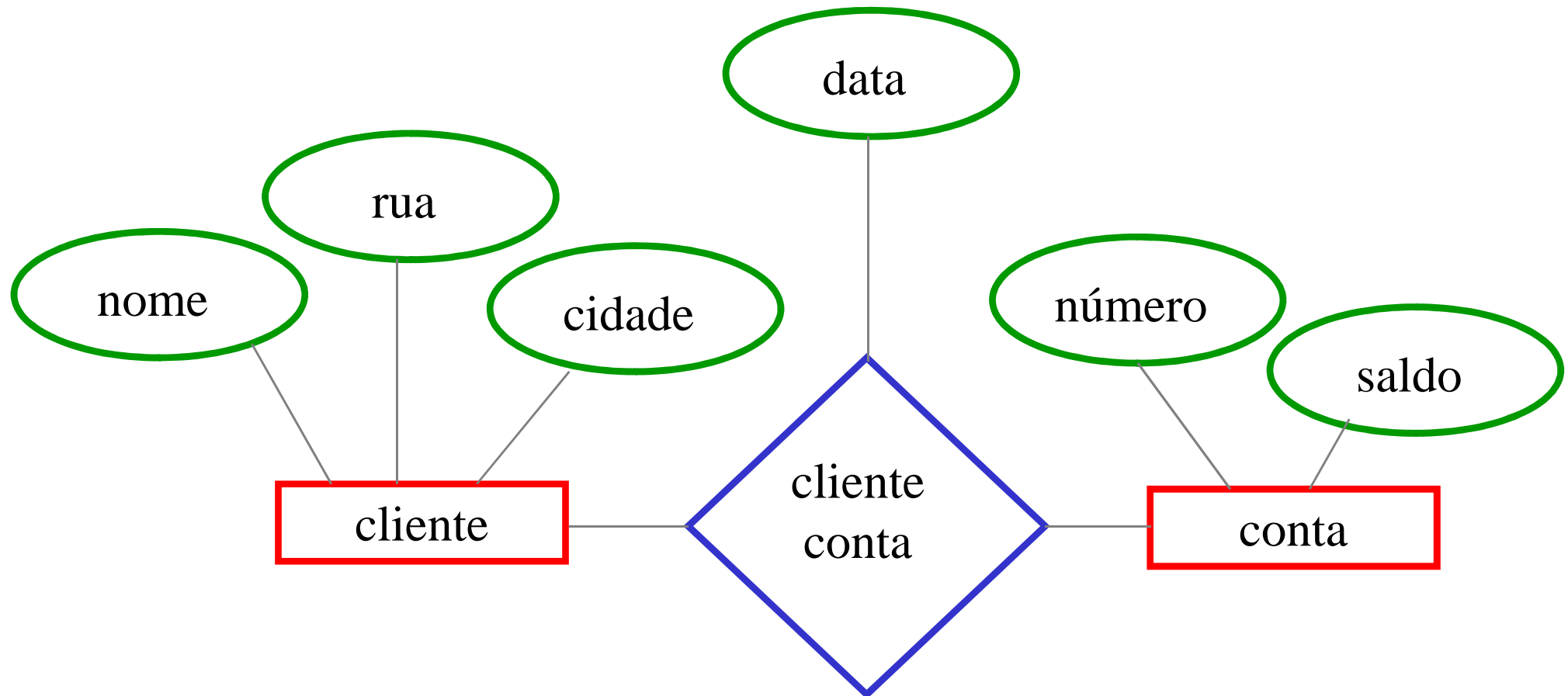
- *Data Manipulation Language* (DML)
 - manipulação de dados:
 - recuperação das informações armazenadas no BD
 - inserção de novas informações no BD
 - remoção de informações do BD
 - modificação das informações do BD
 - Objetivo ➔ proporcionar interação humana eficiente com o sistema.
 - DML viabiliza o acesso (manipulação) dos dados organizados por um modelo de dados apropriado.

Modelo Entidade-Relacionamento

- **Projeto de um BD:**
 - (1) modelo conceitual
 - (2) projeto lógico
- **Modelo Entidade-Relacionamento:**
 - modelo conceitual de dados
 - criado em 1976 por Peter Chen
 - representado graficamente pelo Diagrama Entidade-Relacionamento (DER)

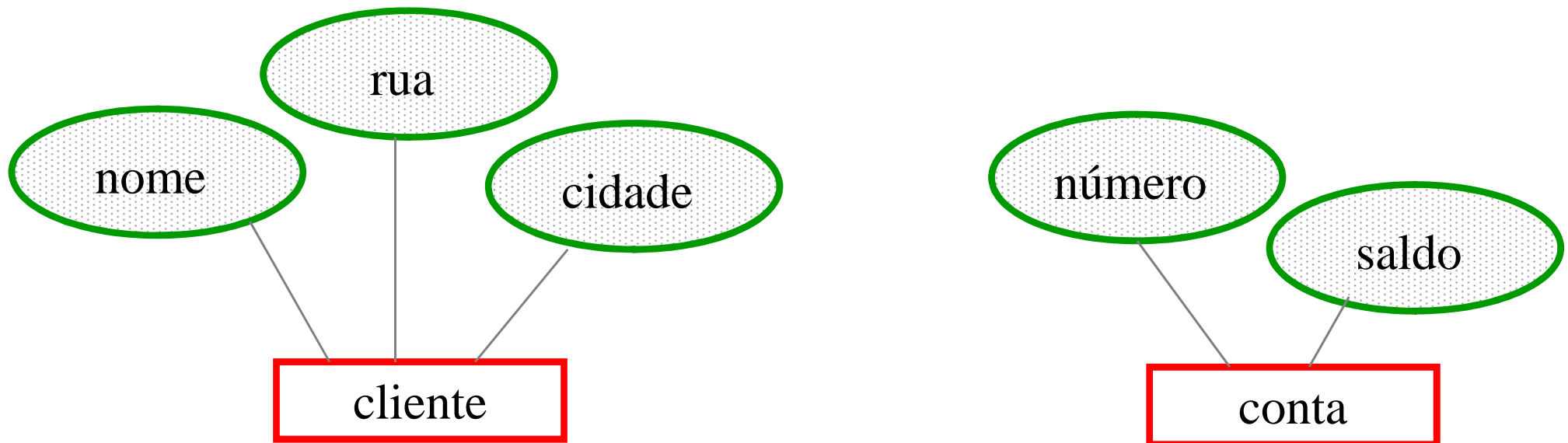
Modelo Entidade-Relacionamento

- Diagrama Entidade-Relacionamento (DER)



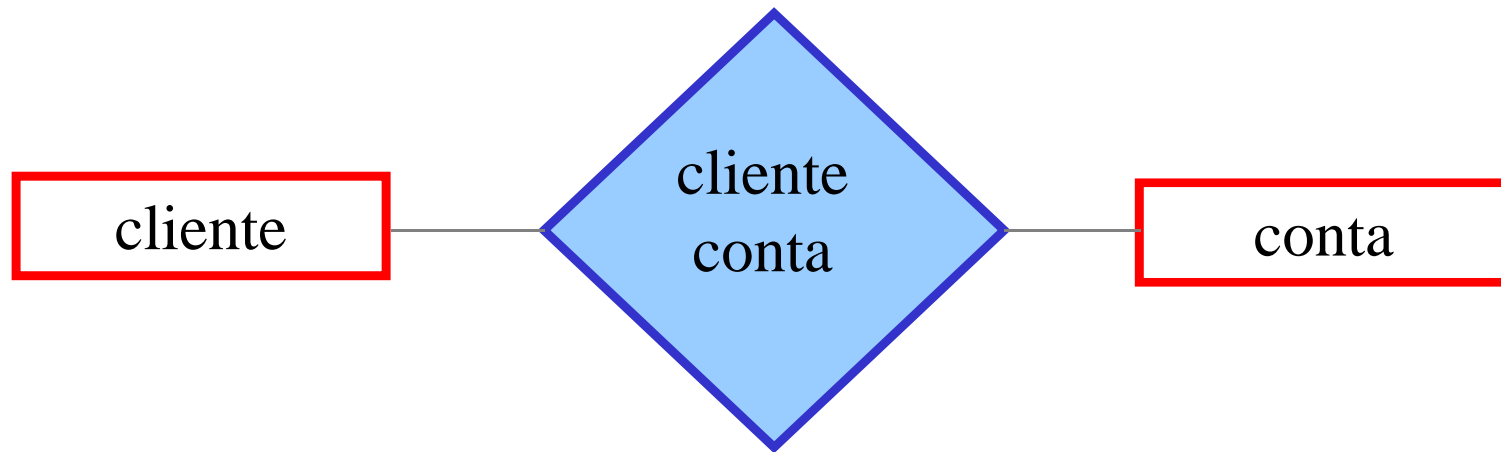
Entidades e atributos - Representação

- Exemplo:

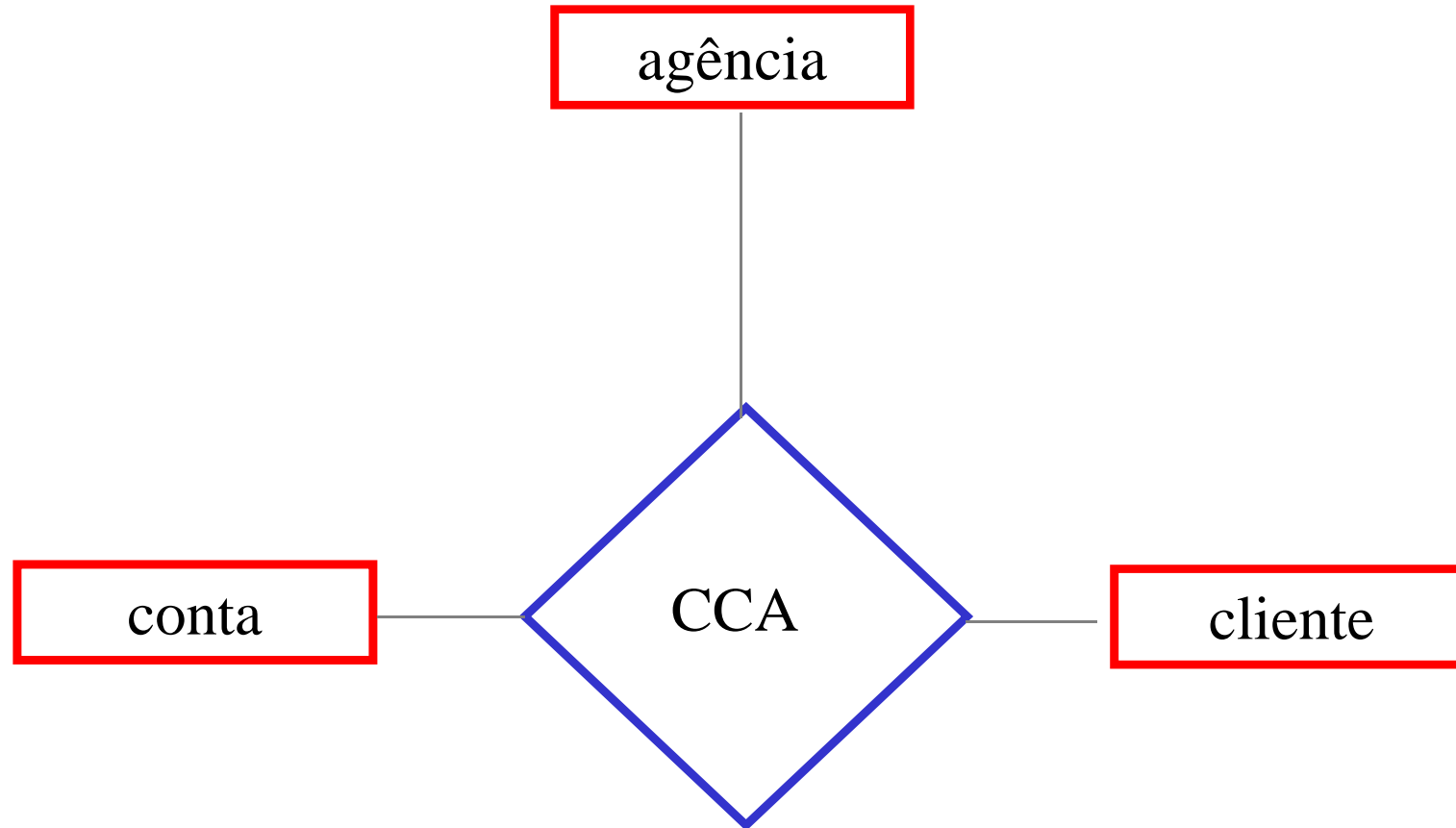


Representação de Relacionamentos

- Exemplo:

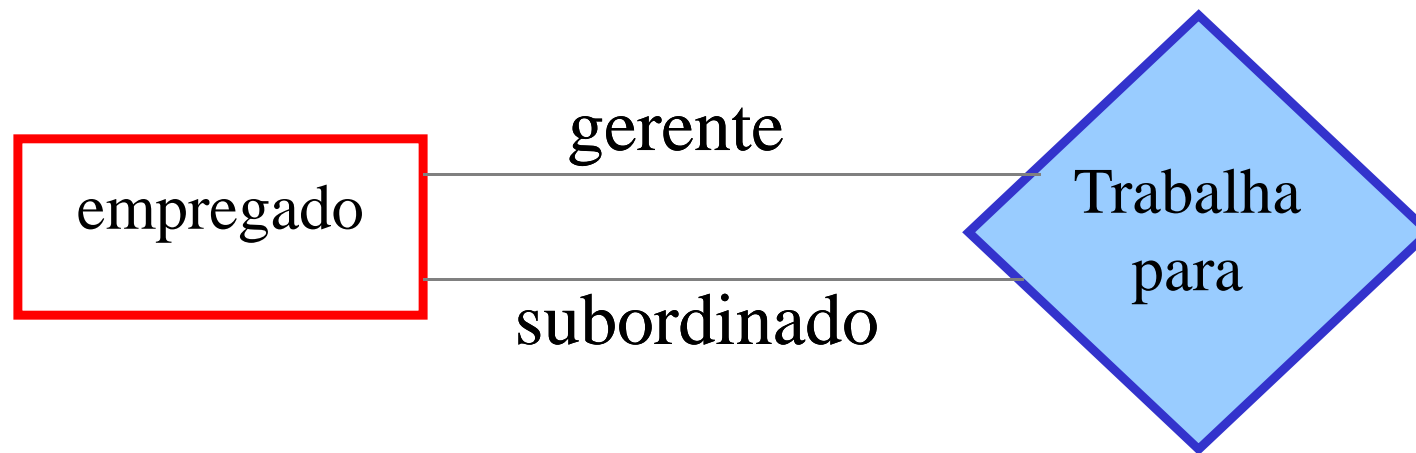


Representação de Relacionamento Ternário



Relacionamentos e Conjuntos de Relacionamentos

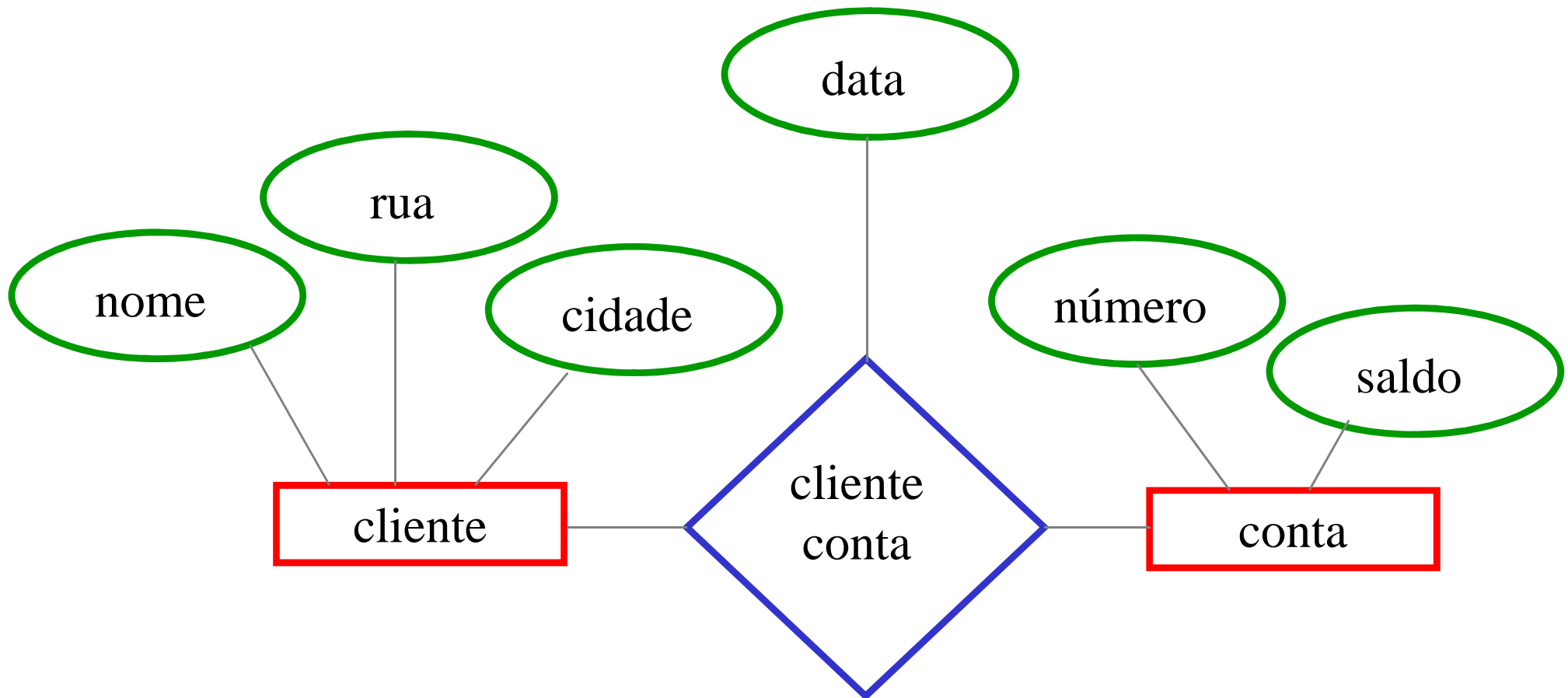
- Exemplo:



Relacionamentos e Conjuntos de Relacionamentos

- Uma instância de relacionamento em um esquema E-R representa a existência de uma **associação** entre a entidade e o mundo real no qual se insere o empreendimento que está sendo modelado.
- **Atributos descritivos** → atributos próprios do conjunto de relacionamentos.
 - **Exemplo:** *data* poderia ser um atributo do conjunto de relacionamentos *ClienteConta*, especificando quando aquela conta foi movimentada pela última vez.

Representação do Modelo E-R

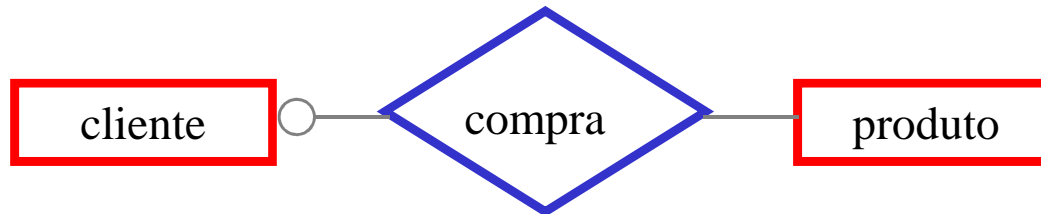
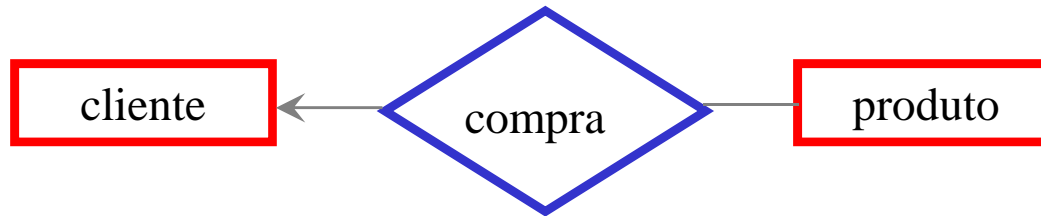
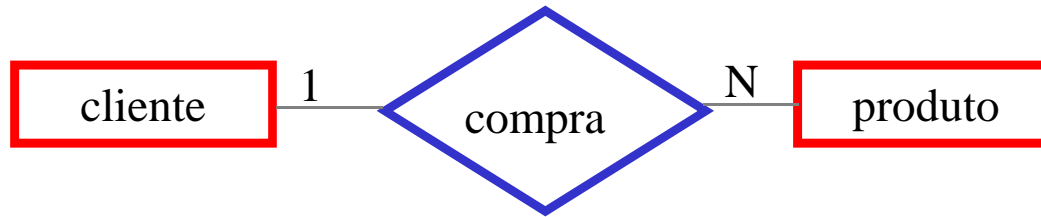


Cardinalidade de mapeamentos

- Há dois tipos de cardinalidade: máxima e mínima
- Cardinalidade máxima:
 - expressa o **número máximo de entidades** ao qual outra entidade pode estar associada via um relacionamento.
 - úteis principalmente em relacionamentos binários
- Cardinalidade mínima:
 - expressa o **número mínimo de entidades** ao qual outra entidade pode estar associada via um relacionamento.

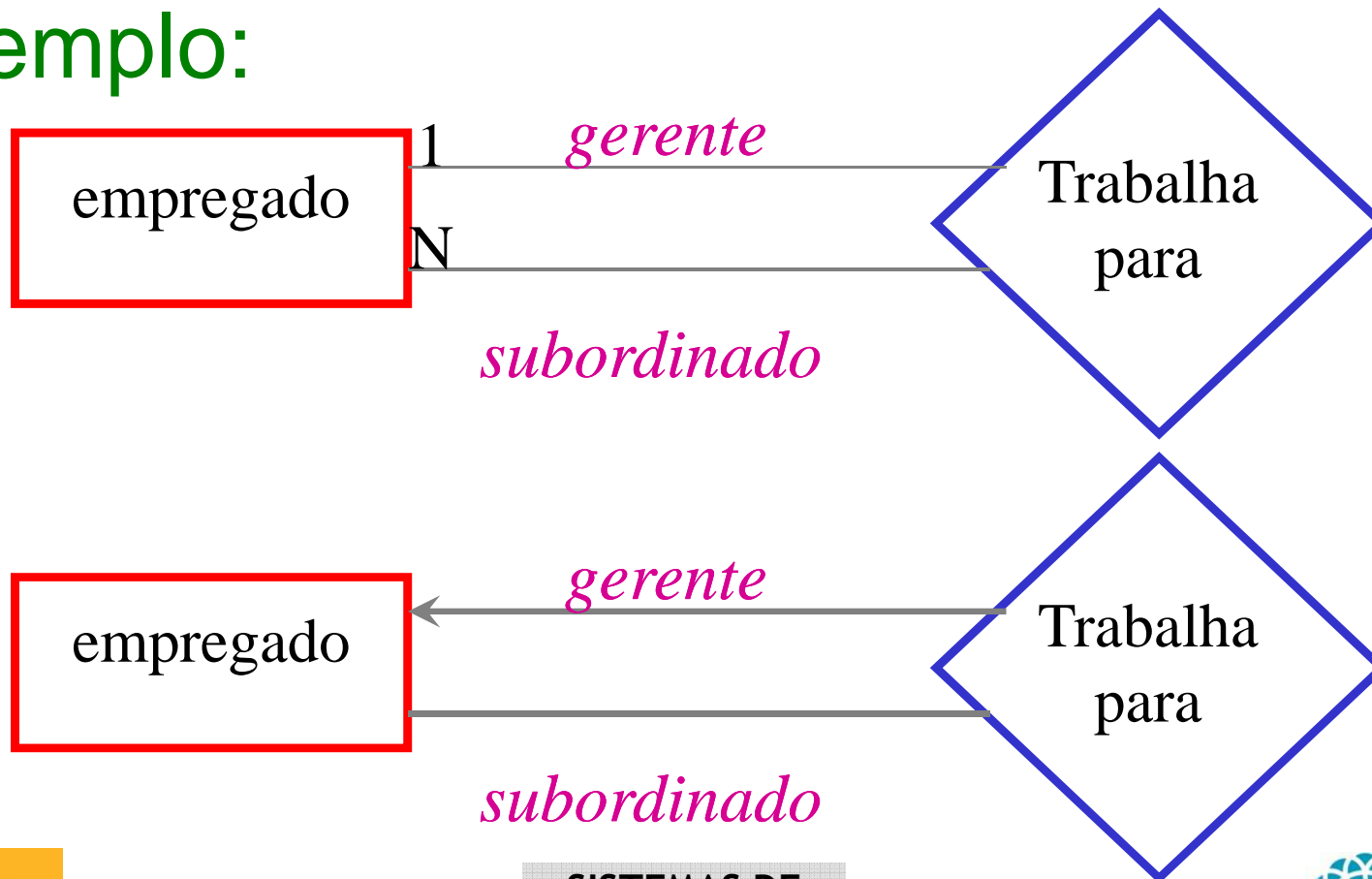
Cardinalidade – Outras notações

- Exemplo: cardinalidade um-para-muitos



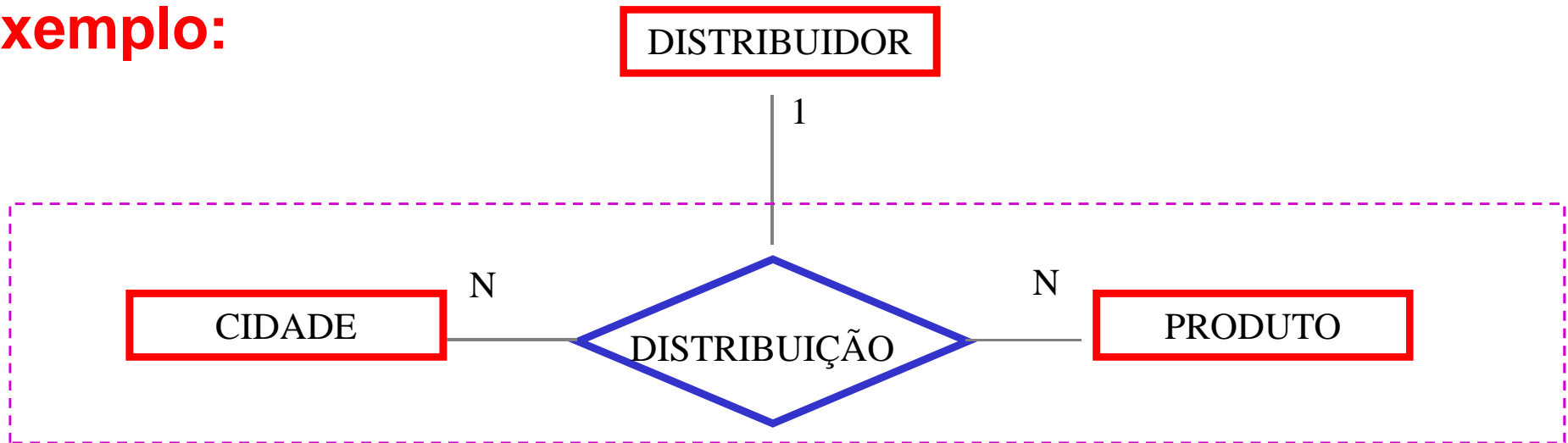
Representação de autorrelacionamento

- **Papel** deve ser especificado através de rótulos nas linhas que ligam os retângulos aos losangos.
- **Exemplo:**



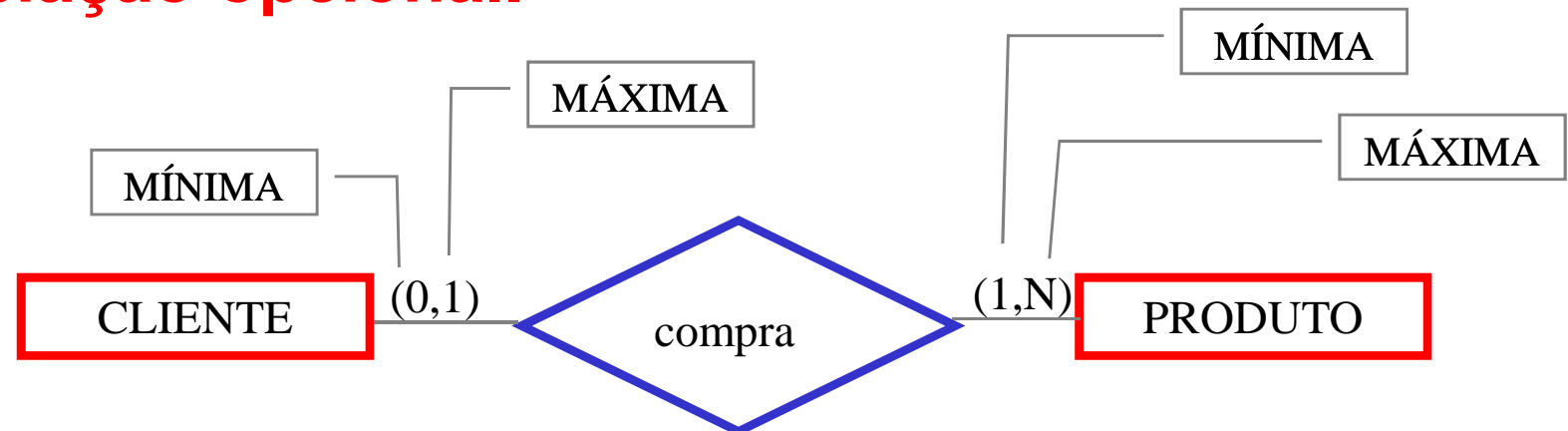
Cardinalidade de Relacionamento Ternário

- Em um relacionamento **R** entre três entidades **A**, **B** e **C**, a cardinalidade de **A** e **B** dentro de **R** indica quantas ocorrências de **C** podem estar associadas a um par de ocorrências de **A** e **B**.
- Exemplo:**



Cardinalidade Mínima

- Número mínimo de entidades que são associadas a uma ocorrência de outro conjunto de entidades através de um relacionamento.
- Consideramos apenas duas cardinalidades mínimas: **zero** e **1**
- A cardinalidade mínima **1** também recebe a denominação de **associação obrigatória**.
- A cardinalidade mínima **0** também recebe a denominação de **associação opcional**.



Chaves Primárias

- Conceitualmente, **entidades** e **relacionamentos** são distintos, mas como diferenciá-los no Banco de Dados?
- Para conjunto de entidades:
 - Distinção → *superchave*
 - *Superchave* → conjunto de um ou mais atributos que permitem identificar uma entidade das demais.
 - **Exemplo:**
 - podem ser superchaves de cliente:
 - ✍ **CPF**
 - ✍ **Nome-cliente + CPF**
 - ✍ **Nome-cliente + data_nascimento + endereço**
(considerando que não há pessoas como mesmo nome, nascidas na mesma data, morando no mesmo endereço)
 - não pode ser superchave de cliente:
 - ✍ **Nome-cliente**

Chaves Primárias

- Se ***K*** é uma **superchave**, qualquer conjunto de atributos que contenha ***K*** é uma superchave.
- **superchave** pode conter atributos desnecessários
- O que nos interessa → **superchaves** menores possíveis → aquelas em que nenhum subconjunto é superchave → **chaves candidatas**.

- **Exemplo**

- podem ser chaves candidatas:

- ✎ **CPF**

- ✎ **RG + estado**

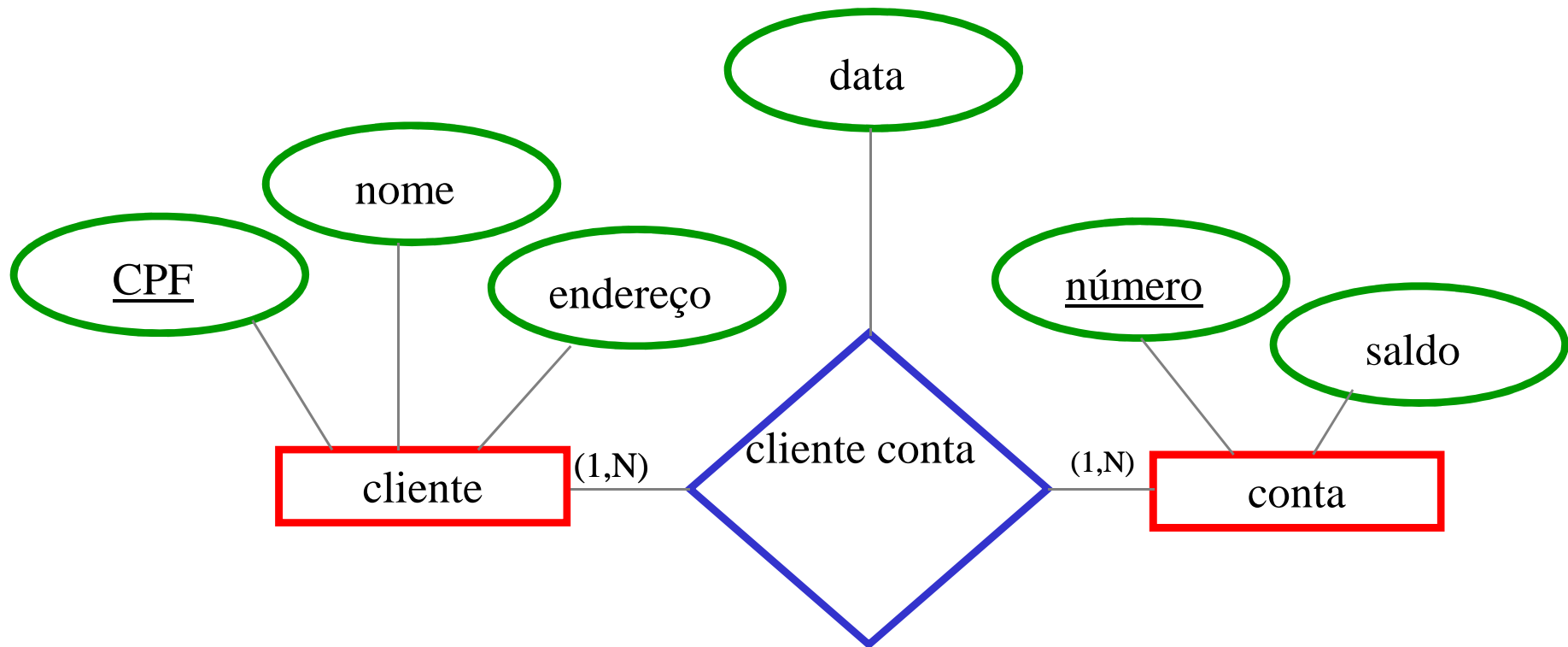
- não pode ser chave candidata:

- ✎ **Nome-cliente + CPF**

Chaves Primárias

- *Chave Primária* → chave candidata escolhida pelo projetista do BD como mecanismo principal para **identificação de uma entidade como única**.
 - **Exemplo:** chave primária de *cliente*:
→ CPF
- *Chave (primária, candidata, superchave)* → propriedade do conjunto de entidade e não de uma entidade individualmente.
 - Duas entidades individuais em um mesmo conjunto não podem ter, simultaneamente, valores iguais em seus atributos-chaves.

Chaves Primárias - representação no DER



Dependência de existência

- A existência de uma entidade **A** depende da existência da entidade **B**.
 - **A** é dependente da existência de **B**.
 - Se **B** for excluído, o mesmo deve acontecer com **A**.
 - **A** é chamada entidade subordinada
 - **B** é chamada entidade dominante

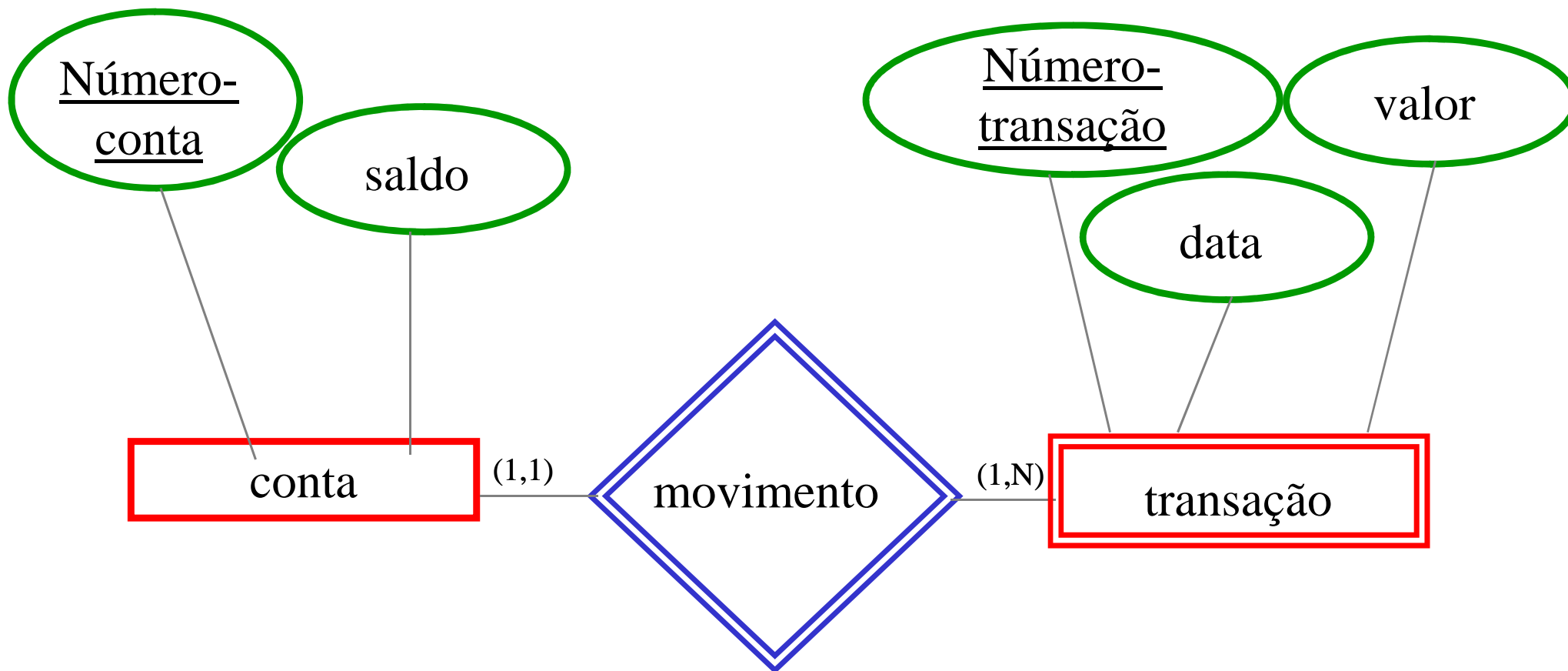
Dependência de existência

- A participação de um conjunto de entidades E no conjunto de relacionamento R é dita total se todas as entidades em E participam em pelo menos um relacionamento R .
 - Se somente algumas entidades em E participam no relacionamento R , a participação do conjunto de entidades E no relacionamento R é dita parcial.
- 👉 ***A participação total está estreitamente relacionada à existência de dependência***

Chaves Primárias – Entidade Forte e Fraca

- Um conjunto de entidades **pode não possuir atributos suficientes** para formação de uma chave primária.
 - **Exemplo:**
 - conjunto de entidades *transação*, com os atributos ***número-transação***, ***data*** e ***valor***.
 - movimentações em contas diferentes podem ter o mesmo número de transação.

Representação de Entidade Fraca



Chaves Primárias – Entidade Forte e Fraca

- **Entidade fraca** → não possui *chave primária própria*
- **Entidade forte** → possui *chave primária própria*.
- **Entidade fraca** e **Entidade forte** ↔ conceito de dependência.
- **Entidade forte** → entidade **dominante**.
- **Entidade fraca** → entidade **subordinada**.
- **Discriminador** de um conjunto de entidades fracas → conjunto de atributos que permite distinguir uma particular entidade em um subconjunto de entidades, mas não permite distingüi-la de todas as demais entidades.
 - Também conhecido como **identificador** ou **chave parcial**
 - **Exemplo:**
 - conjunto de entidades **transação: número-transação** → identificam unicamente uma transação para cada conta.

Chaves Primárias – Entidade Forte e Fraca

- Chave primária de um conjunto de entidades fracas → *chave primária da entidade forte da qual é existencialmente dependente + discriminador.*
 - Exemplo:
 - Chave primária de **transação**:
→ (número-conta, número-transação)

Diagrama Entidade-Relacionamento Estendido

- Grande parte dos bancos de dados podem ser modelados com os conceitos básicos do DER.
- No entanto, algumas extensões permitem refinamentos que podem ser muito úteis.
- Modelo Entidade-Relacionamento Estendido:
 - Especialização
 - Generalização
 - Conjuntos de entidades de nível superior e inferior
 - Herança de atributos
 - Agregação

Diagrama Entidade-Relacionamento Estendido

- **Especialização**
 - Resultado da separação de entidades de nível superior, formando um conjunto de entidades de nível inferior.
 - Processo *top-down*
 - Por quê? Atributos não se aplicam a todas as entidades.

Diagrama Entidade-Relacionamento Estendido

- Exemplo:

- Conjunto de entidades **conta**, com atributos **número-conta** e **saldo**
- Cada conta pode ser classificada como:
 - **conta-corrente**: com atributos adicionais **data-abertura** e **valor-limite**
 - **conta-poupança**: com atributos adicionais **taxa-juros** e **data-última-movimentação**
- Então:
 - **conta**: conjunto de entidades de nível superior: **superclasse**
 - **conta-corrente** e **conta-poupança**: conjuntos de entidades de nível inferior: **subclasses**

Diagrama Entidade-Relacionamento Estendido

Especialização

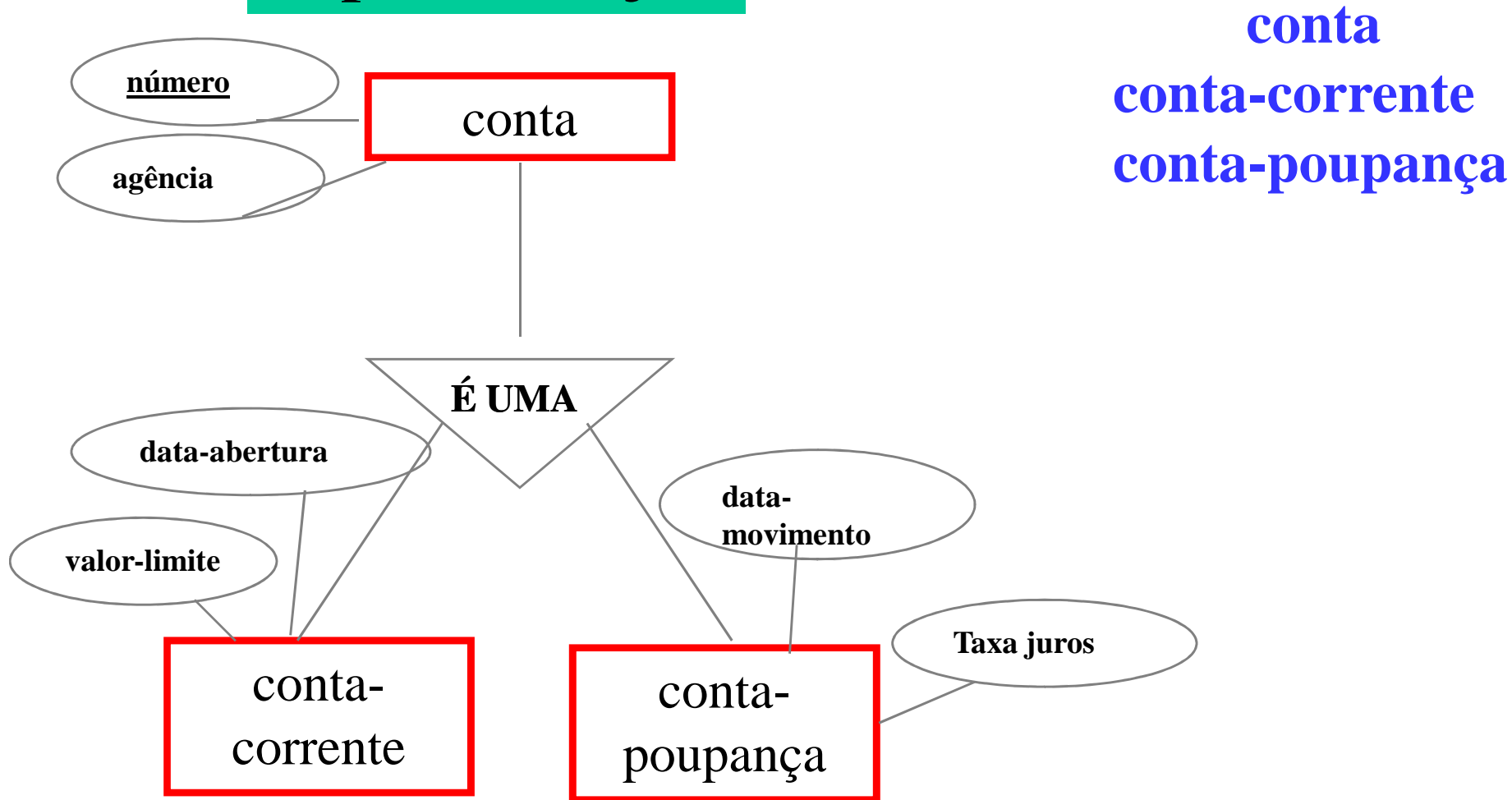
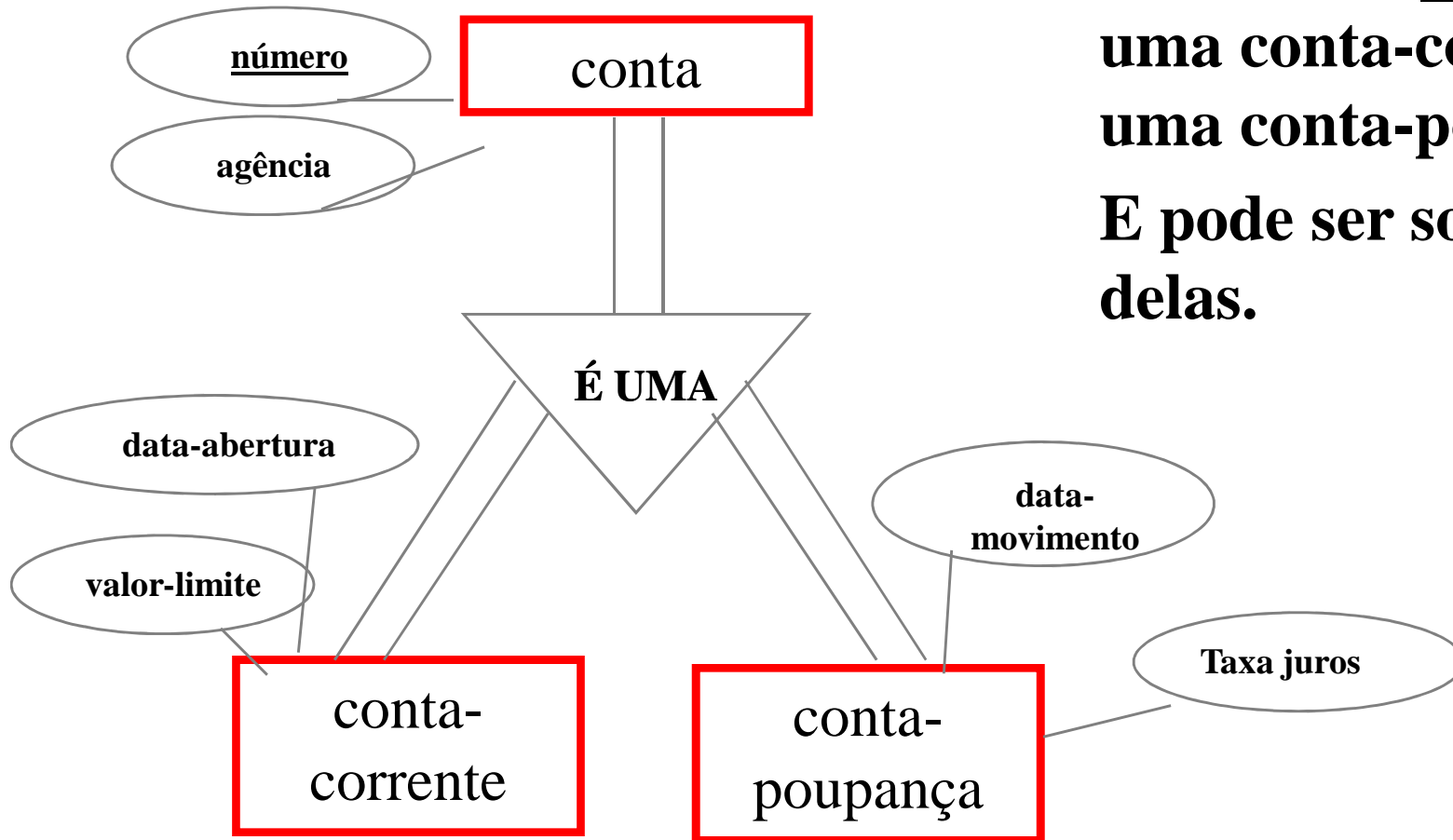


Diagrama Entidade-Relacionamento Estendido

- **Generalização**
 - Processo inverso da Especialização – *bottom up*.
 - Resultado da união de dois ou mais conjuntos de entidades de nível inferior produzindo um conjunto de entidades de nível superior.
 - Por quê? Atributos são comuns a dois ou mais conjuntos de entidades.

Diagrama Entidade-Relacionamento Estendido

Generalização



Uma conta deve ser uma conta-corrente ou uma conta-poupança. E pode ser somente uma delas.

Diagrama Entidade-Relacionamento Estendido

Generalização e Especialização podem conviver no mesmo conjunto de entidades

Exemplo: cada *empregado* é ou um *secretário* ou um *encarregado-empréstimo*. Cada *secretário* ou *encarregado-empréstimo* tem um *gerente*, que por sua vez, é um *empregado*.

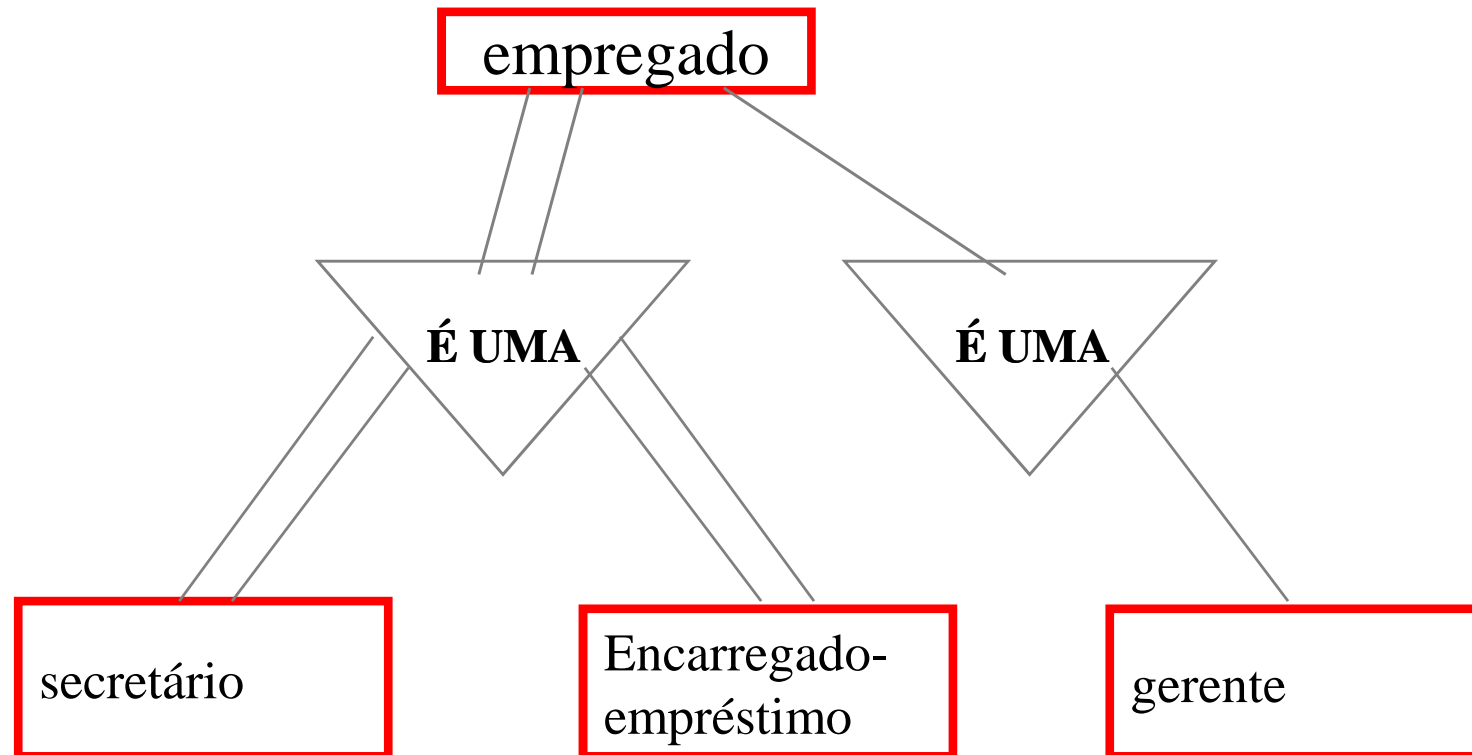


Diagrama Entidade-Relacionamento Estendido

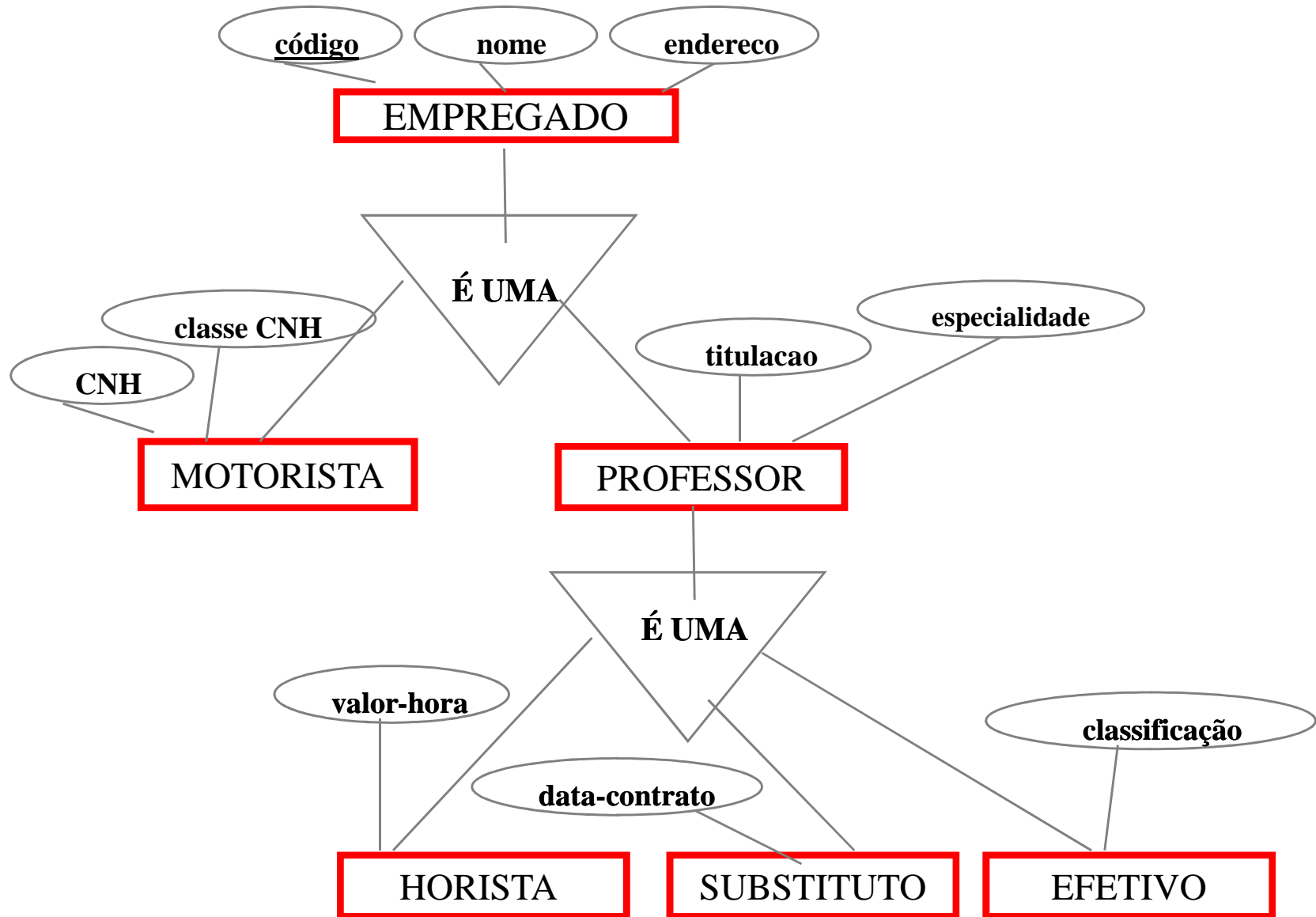


Diagrama Entidade-Relacionamento Estendido

- **Agregação**

- Limitação do Modelo E-R → não consegue expressar relacionamentos entre relacionamentos.
- Solução: mecanismo de agregação → transformação de um conjunto de relacionamentos em conjunto de entidades.

Diagrama Entidade-Relacionamento Estendido

- **Exemplo:** *empregado* que trabalha em um *projeto* e usa diversas *máquinas* para realizar seu trabalho.

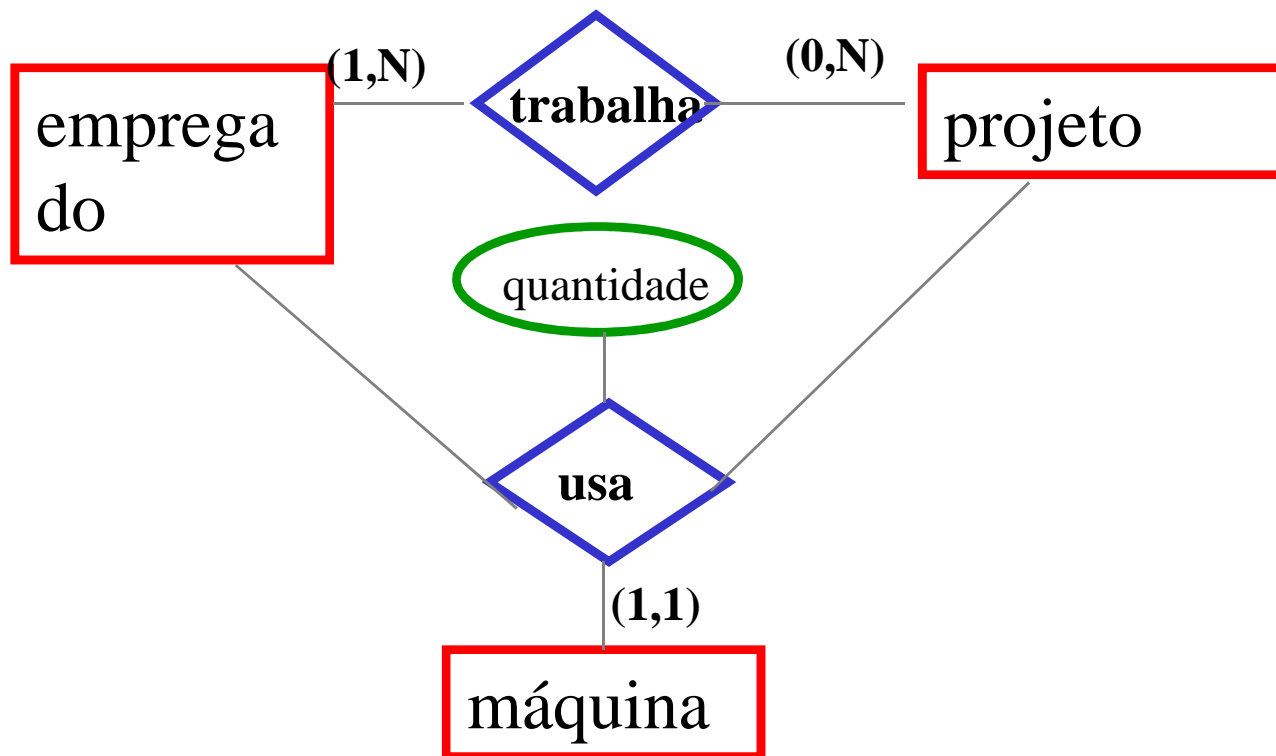
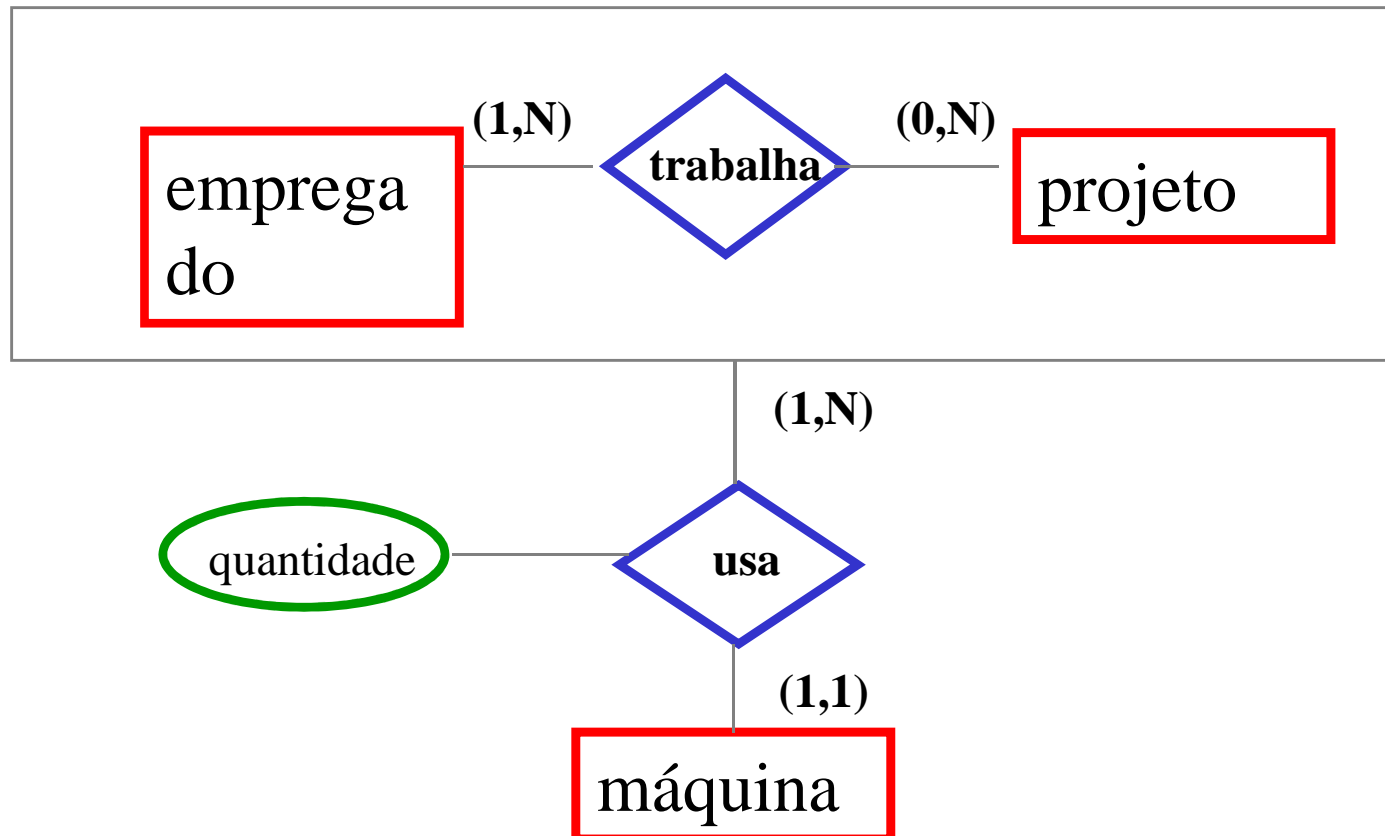


Diagrama Entidade-Relacionamento Estendido

- **Agregação** → abstração através da qual relacionamentos são tratados como entidades de nível mais alto.

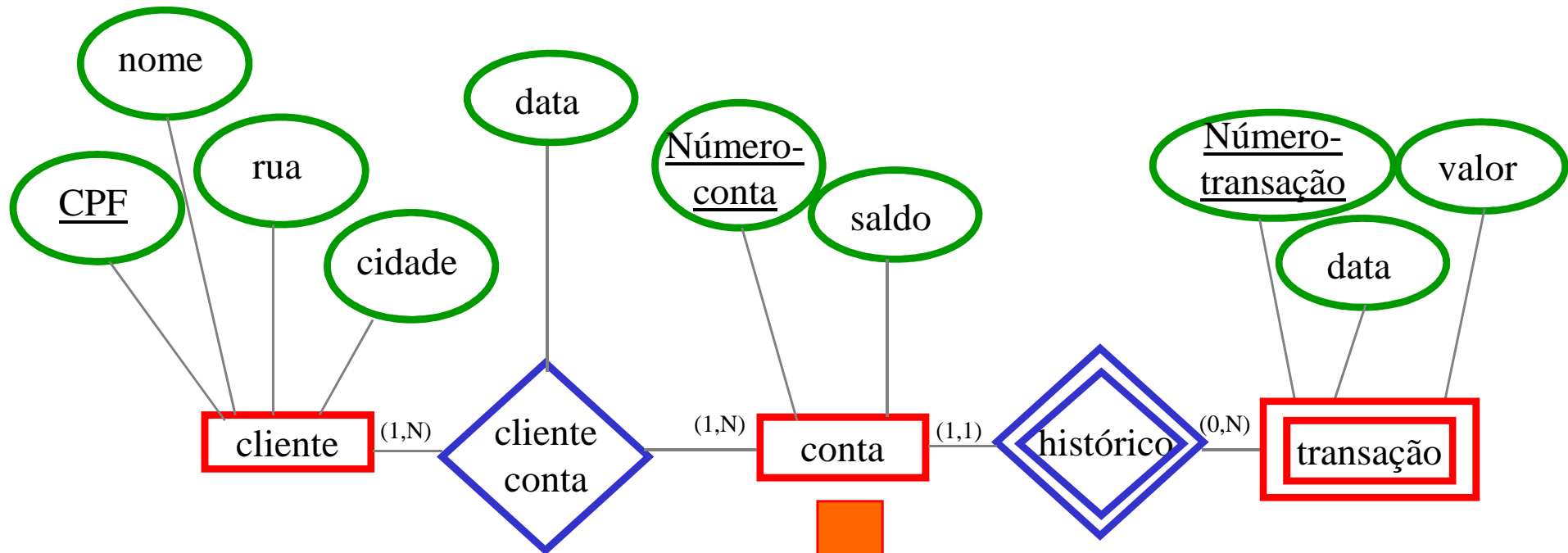


Reduzindo Diagramas E-R a Tabelas

- Conjunto de entidades → tabela com número de colunas = número de atributos.
- Entidades fortes:
 - Se E é um conjunto de entidades fortes com atributos a_1, a_2, \dots, a_n .
 - E será representada por uma tabela chamada E com n colunas distintas, cada uma correspondendo a um atributo de E .
 - Cada linha da tabela corresponde a uma entidade do conjunto de entidades E .

Reduzindo Diagramas E-R a Tabelas

Exemplo:



CONTA

<u>número-conta</u>	saldo
900	678,00
556	458,98
647	561,89
801	654,12

Reduzindo Diagramas E-R a Tabelas

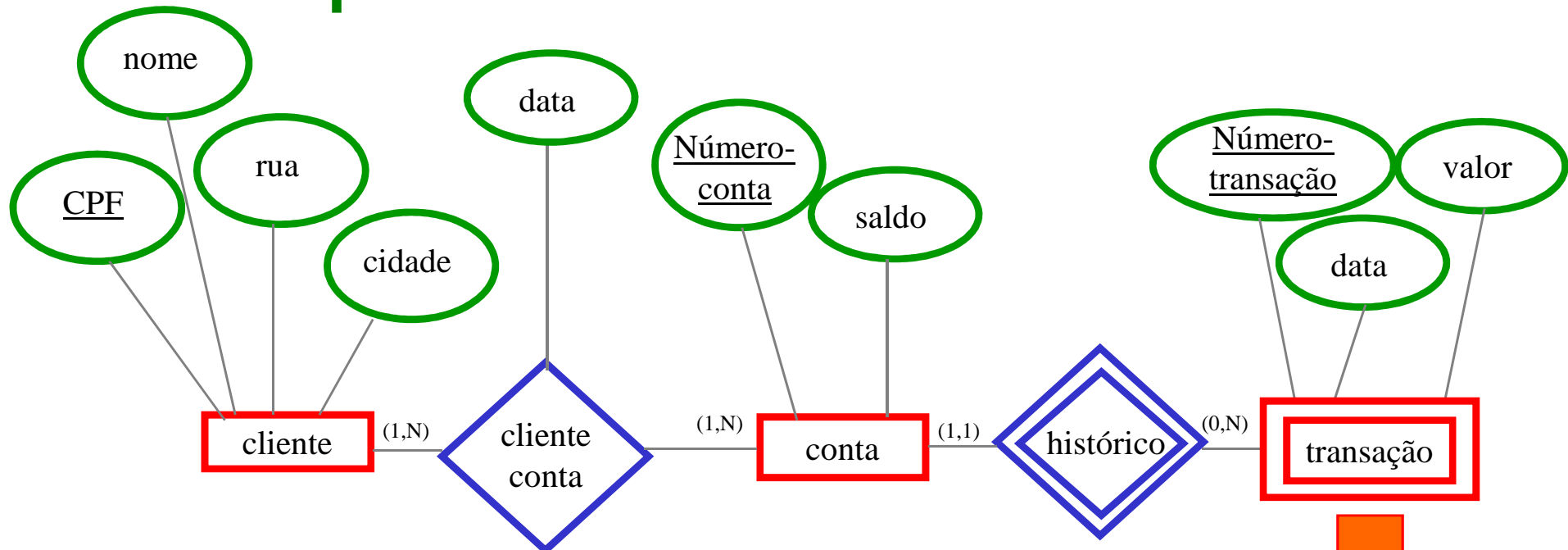
- Entidades fracas:

- Se **A** é um conjunto de entidades **fracas** com atributos **a_1, a_2, \dots, a_r**
- Se **B** é um conjunto de entidades **fortes** do qual A é dependente.
- Se a chave primária **B** é constituída de atributos **b_1, b_2, \dots, b_s** .
- **A** será representada por uma tabela chamada **A** com uma coluna para cada tributo do conjunto

$$\{a_1, a_2, \dots, a_r\} \cup \{b_1, b_2, \dots, b_s\}$$

Reduzindo Diagramas E-R a Tabelas

Exemplo:



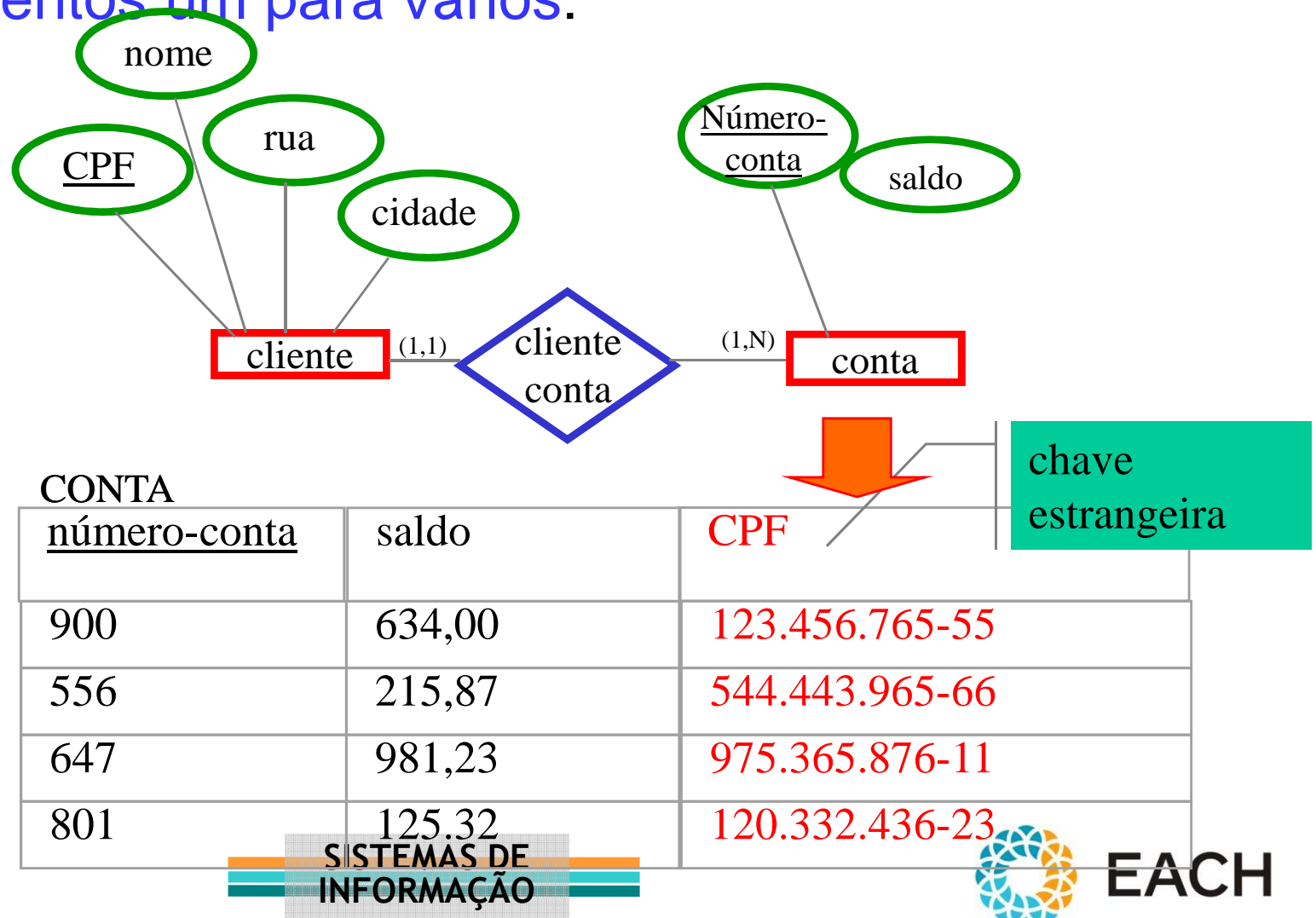
chave estrangeira /
chave primária

TRANSAÇÃO

<u>número-conta</u>	<u>número-transação</u>	data	valor
900	1	01/02/1999	78,00
900	2	08/02/2000	45,08
647	1	30/03/2002	16,89
647	2	05/11/2005	4,12

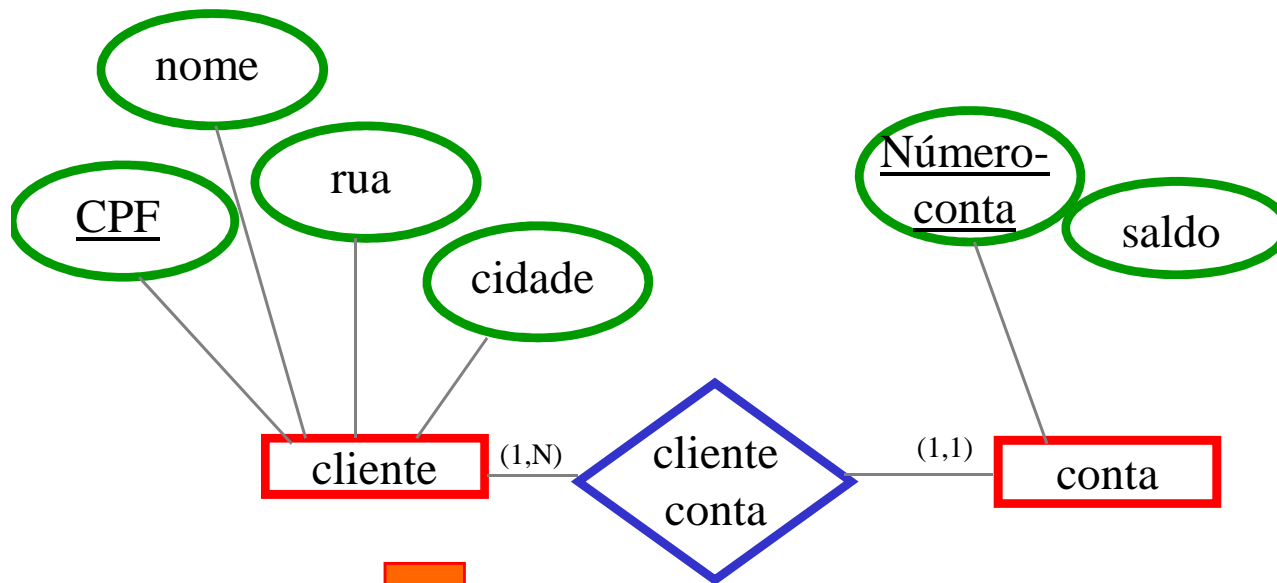
Reduzindo Diagramas E-R a Tabelas

- Conjunto de relacionamentos → depende da cardinalidade máxima.
- Relacionamentos um para vários:



Reduzindo Diagramas E-R a Tabelas

- Relacionamentos vários para um:



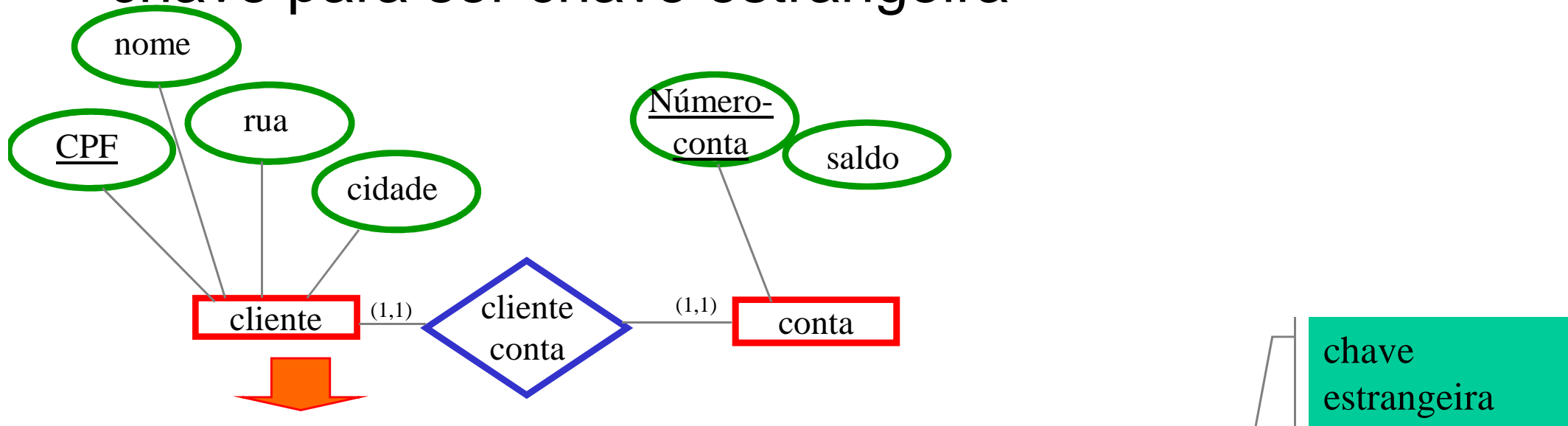
chave
estrangeira

CLIENTE

<u>CPF</u>	nome	rua	Cidade	número-conta
123.456.765-55	João	Av. Rio Branco 67	Vera Cruz	900
544.443.965-66	José	R. Maria José 89	Ourinhos	556
975.365.876-11	Antonio	Av. das Rosas 987	Marília	647
120.332.436-23	Maria	Av. Tiradentes 43	Marília	801

Reduzindo Diagramas E-R a Tabelas

- Relacionamentos um para um: escolhe-se a melhor chave para ser chave estrangeira



CLIENTE

<u>CPF</u>	nome	rua	Cidade	número-conta
123.456.765-55	João	Av. Rio Branco 67	Vera Cruz	900
544.443.965-66	José	R. Maria José 89	Ourinhos	556
975.365.876-11	Antonio	Av. das Rosas 987	Marília	647
120.332.436-23	Maria	Av. Tiradentes 43	Marília	801

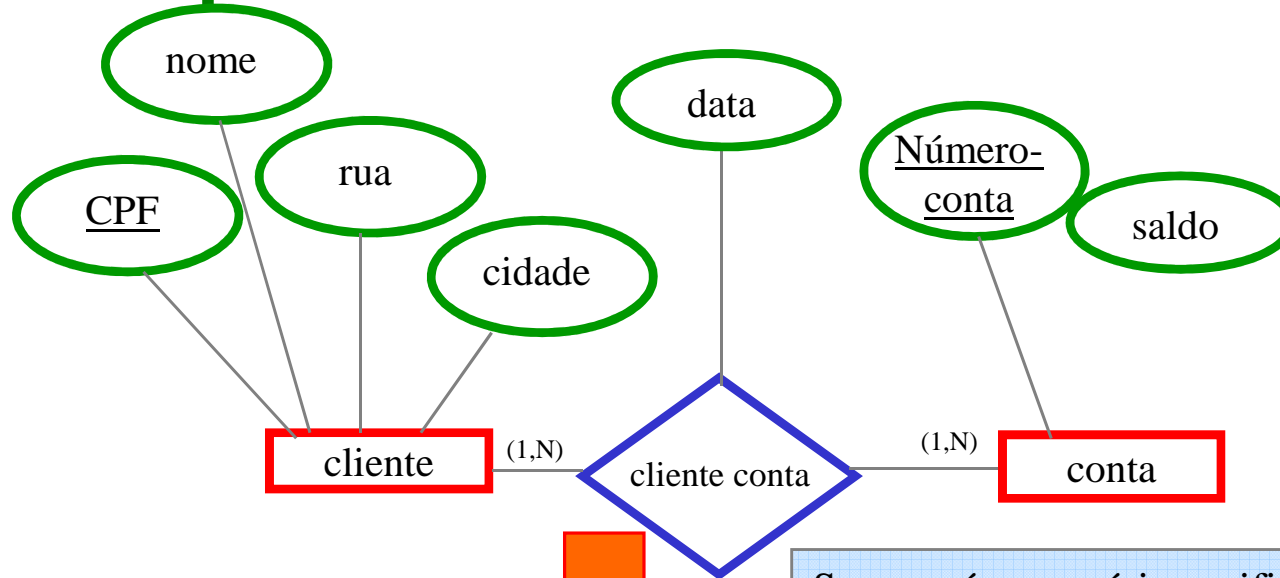
Reduzindo Diagramas E-R a Tabelas

- Relacionamentos vários para vários:
 - Se R é um conjunto de relacionamentos envolvendo os conjuntos de entidades E_1, E_2, \dots, E_n .
 - Se (E_i) é o conjunto de atributos que forma a chave primária para o conjunto de entidades E_i .
 - Se R possui atributos descritivos chamados $\{r_1, r_2, \dots, r_m\}$, R será representado por uma tabela com o seguinte conjunto de atributos:

$$U_{i=1}^n$$

Reduzindo Diagramas E-R a Tabelas

Exemplo:



Sempre é necessário verificar se a chave primária resultante é suficiente para garantir a unicidade de chaves!

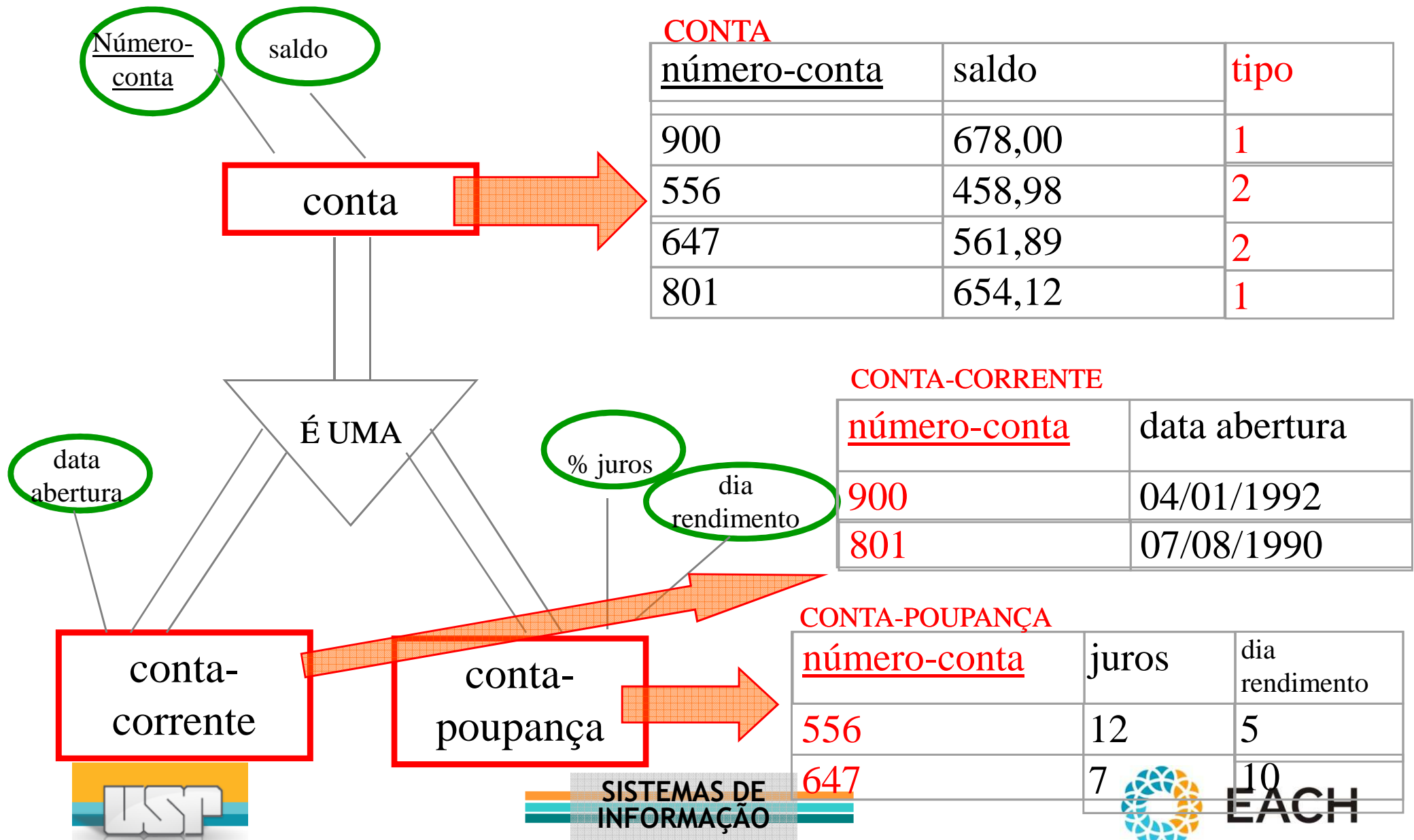
CLIENTECONTA		
<u>CPF</u>	<u>número-conta</u>	data
123.456.765-55	900	01/02/99
544.443.965-66	556	08/02/00
975.365.876-11	647	30/03/02
120.332.436-23	801	05/11/99

Reduzindo Diagramas E-R a Tabelas

- Generalização e Especialização:
 - Trata-se de forma semelhante a relacionamento um-para-um, considerando:
 - chave primária da entidade superior torna-se a chave primária de todas entidades inferiores
 - Acrescenta-se atributo na tabela resultante da entidade superior para distinguir entidade inferior.

Reduzindo Diagramas E-R a Tabelas

- Generalização e Especialização:

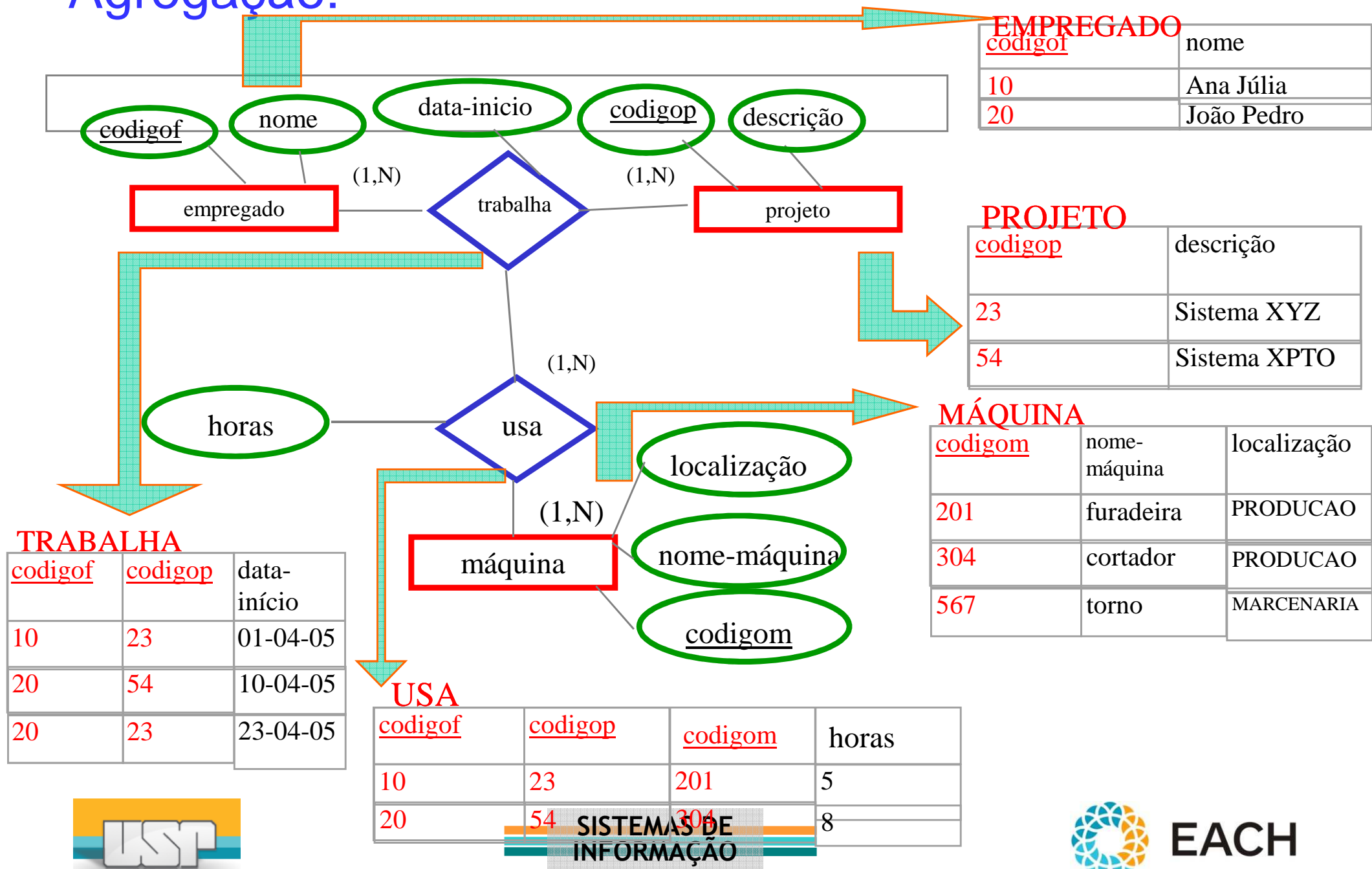


Reduzindo Diagramas E-R a Tabelas

- **Agregação:**
 - **A agregação é o resultado de um relacionamento com cardinalidade vários para vários:**
 - o relacionamento é transformado em uma tabela, contendo: chaves primárias dos conjuntos de entidades envolvidos e os atributos descritivos do relacionamento.

Reduzindo Diagramas E-R a Tabelas

- Agregação:



Modelo Relacional

- Conceito de *relação matemática* (semelhante a tabela de valores) como seu bloco de construção básica.
- Base teórica: teoria dos conjuntos e lógica de predicados de primeira ordem.
- Implementações comerciais:
 - Década de 80: SGBD Oracle e sistema SQL/DS, componente do SO MVS, da IBM.
 - SGBDRs mais conhecidos:
 - DB2 e Informix Dynamic Server (IBM)
 - Oracle e Rdb (Oracle)
 - SQL Server e Access (Microsoft)



Modelo Relacional

- **Banco de Dados: coleção de relações → cada relação tem um nome único.**
- **Informalmente, uma relação é semelhante a uma tabela de valores.**

Modelo Relacional - Conceitos

atributos

Número	Agência	Valor
25	Centro	236,99
678	Vila Verde	1598,70
433	Servidor	64,81
67	Matão	45,23
978	Nova Cidade	101,00

tuplas

Relação
CONTA

Domínio = conjunto de valores permitidos para o atributo → **atômicos (indivisíveis)**

Restrições do Modelo Relacional

- **Regras a respeito dos valores** que podem ser armazenados nas relações
 - **Garantem** que mudanças feitas no BD por usuários não resultem em inconsistência dos dados.
 - Valores **devem ser sempre satisfeitos** em quaisquer das relações ***R*** de um banco de dados ***BD***.
- **Três categorias:**
 - **Restrições inerentes baseadas em modelo.**
 - Exemplo: relação não pode ter tuplas repetidas.
 - **Restrições baseadas em esquemas:** que podem ser expressas diretamente nos esquemas do modelo de dados.
 - Exemplo: intervalo de um atributo, chave estrangeira.
 - **Restrições baseadas em aplicação:** devem ser expressas e impostas nos programas de aplicação.
 - Exemplo: tuplas que um usuário pode acessar em uma relação.

Restrições baseadas em esquema

1. Restrições de domínio

- Dentro de cada tupla , o valor de cada atributo A deve ser um valor atômico do domínio $dom(A)$.

2. Restrições de chave

- Duas tuplas distintas não podem ter valores idênticos para todos os atributos da chave primária e superchaves (unicidade de chave).

Restrições baseadas em esquema

3. Restrições de valores nulos

- Especifica que um atributo da relação não pode ser nulo. Por exemplo: NOME is not null.

4. Restrições de integridade de entidade

- Nenhum valor de chave primária pode ser nulo.

Restrições baseadas em esquema

5. Restrições de integridade referencial

- o conceito de integridade referencial depende do conceito de **chave estrangeira**
- **Chave estrangeira:**
 - Dois conjuntos de atributos **C** e **D** compatíveis → existe uma **ordem** entre os atributos de ambos os domínios tal que o **primeiro atributo de C tenha o mesmo domínio do primeiro atributo de D**, o mesmo valendo para os segundos atributos, e assim por diante.
 - **chave estrangeira** → conjunto de atributos $D \subseteq R1$ que não é chave em R1, é compatível com outro conjunto de atributos $C \subseteq Rk$ que é a chave primária da relação **Rk**.

Restrições baseadas em esquema

chave estrangeira → conjunto de atributos

$D \subseteq R1$ que não é chave na relação $R1$, mas é compatível com a chave primária de uma outra relação.

A **restrição de integridade referencial** determina que o valor dos atributos D numa tupla qualquer $t(D)$ da relação $R1$ onde D não é chave:

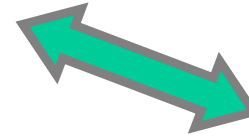
- ou é igual ao valor $t(C)$ na relação Rk onde C é chave
- ou é *nulo*.

Restrições baseadas em esquema

Exemplo 1:

Professor (número-professor, nome, data-admissão)

Disciplina (código, nome-disc, **prof-responsável**)



Professor

número-professor	nome	data-admissão
213	Antônio	02/02/1999
400	José	02/04/2000
67	Joana	05/01/1998
43	João	10/11/1997
25	Maria	14/11/1996

Disciplina

código	nome-disc	prof-responsável
CC876	Banco Dados	43
CC566	Linguagem I	NULL
AS654	Algoritmos	43
AS543	Compiladores	400

Restrições baseadas em esquema

6. Outros tipos de restrições

– Restrições de integridade semântica

- Exemplos:

- salário do empregado deve ser menor que o do chefe

- Número máximo de horas-extras é 35.

- Feitas através dos programas de aplicação ou da *linguagem de especificação de restrição* (gatilhos e asserções)

– Restrições de dependência funcional

- Estabelece relacionamento funcional entre dois conjuntos de atributos X e Y, sendo que X determina o valor de Y em todos os estados da relação.

- Exemplo: o RA 335432 determina sempre o nome da aluna Júlia Neme Delgado.

ACH2025

Laboratório de Bases de Dados

Aula 2

Revisão de Conceitos

Professora:

➤ Fátima L. S. Nunes

