



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

# SGBDs



# Sistemas de Arquivos Convencionais

- Características:
  - Vários arquivos, diferentes programas para recuperar informações
  
- Desvantagens:
  - Redundância e inconsistência de dados
  - Dificuldade de acessar informação (programa pode não estar disponível)
  - Anomalias de acesso concorrente (multiusuários)
  - Problemas de segurança e integridade
  - Isolamento de dados



# Sistemas de Banco de Dados

- Características:
  - Coleção de arquivos inter-relacionados + Conjunto de programas para armazenar, gerenciar e acessar esses arquivos.
  
- Vantagens
  - Consistência e integridade dos dados
  - Segurança
  - Controle de acesso concorrente (multi-usuário)
  - Backup e recuperação de falhas



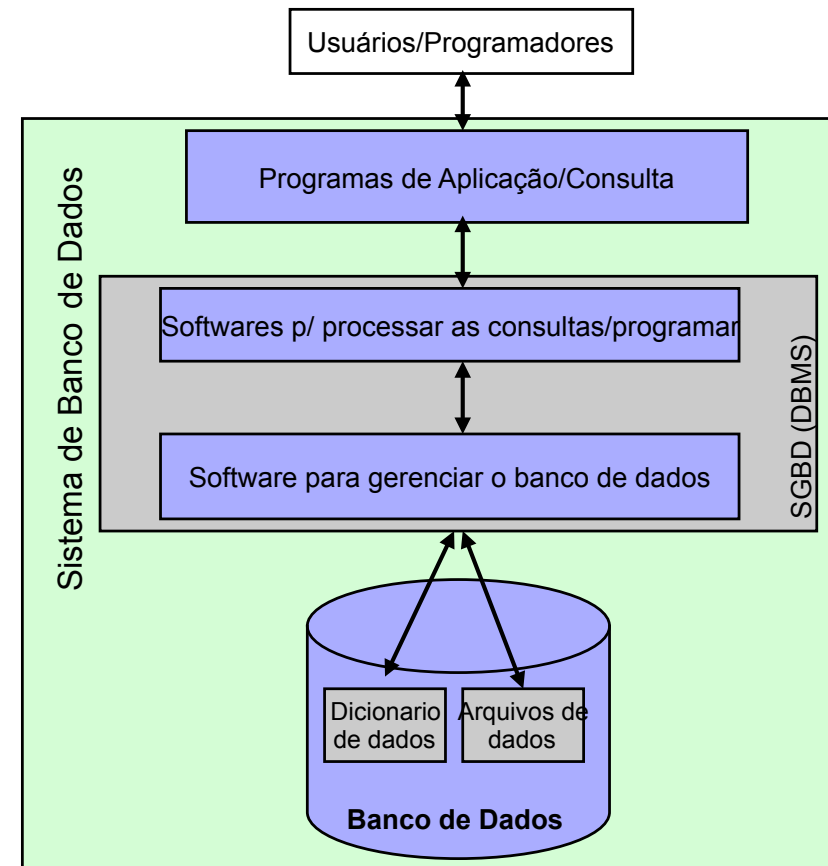
# Características de um SGBD

- Um **SGBD** facilita as seguintes tarefas:
  - **Definição** do BD: tipos de dados, estruturas e restrições que devem ser consideradas
  - **Construção** do BD: inserção propriamente dos dados no meio persistente;
  - **Manipulação** do BD
  - **Consulta** ao BD: recuperar uma informação específica
  - **Manutenção** ao BD: alterar uma informação armazenada
- São requisitos de um SGBD:
  - Facilidade de uso
  - Correção
  - Facilidade de manutenção
  - Confiabilidade
  - Segurança
  - Desempenho



# Um SGBD oferece

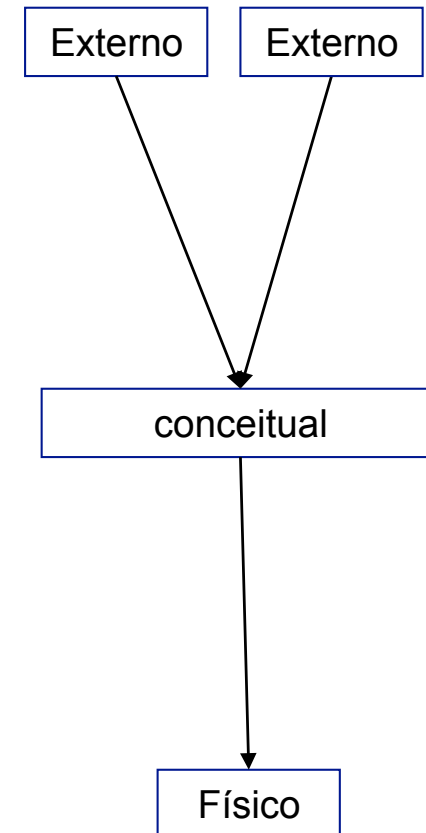
- Interfaces baseadas em modelo de dados de alto nível tanto para a definição da estrutura da base quanto para sua consulta
- Mecanismos que garantem restrições de integridade (ex. triggers, assertions)
- Atomicidade-consistência-integridade-durabilidade) (ex. controle de concorrência, subsistema de recuperação)
- Controle de acesso
- Métodos de acesso e armazenamento eficientes (ex. otimização de consultas)





# Especificação em níveis

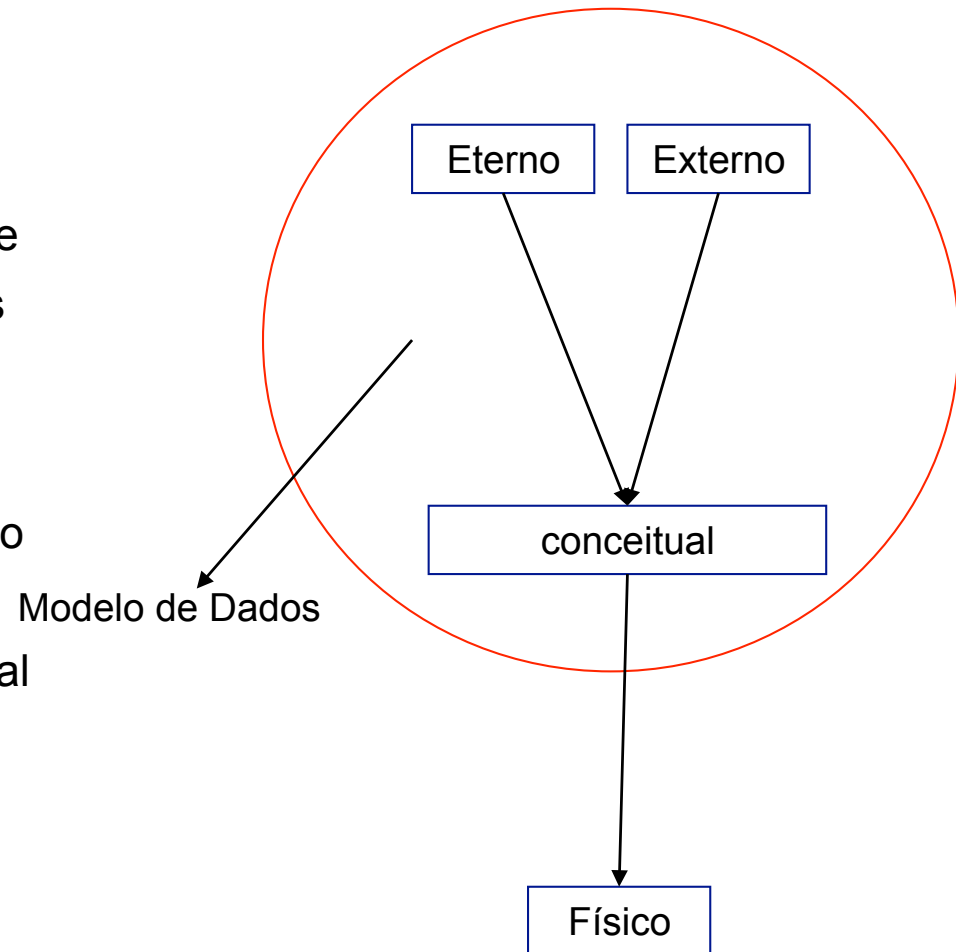
- nível externo:
  - especificação da organização conceitual do BD, vista por um grupo de usuários
- nível conceitual:
  - especificação da organização conceitual do BD, ou seja, o que o BD armazena
- nível físico ou interno:
  - especificação das estruturas de armazenamento do BD, ou seja, como o BD está armazenado





# Vantagens da especificação em níveis

- Facilidade de manutenção
- Independência física (dos dados)
  - permite modificar as estruturas de armazenamento sem impactar as aplicações
- Independência lógica (dos dados)
  - separação entre esquema externo e esquema conceitual permite modificar a organização conceitual com impacto mínimo nas aplicações (construídas sobre os esquemas externos)





# Modelos de dados

- Conjunto de conceitos usados para representar os dados, os relacionamentos entre esses dados e as restrições de consistência
- No processo de modelagem é necessário construir uma abstração dos objetos e fenômenos do mundo real.



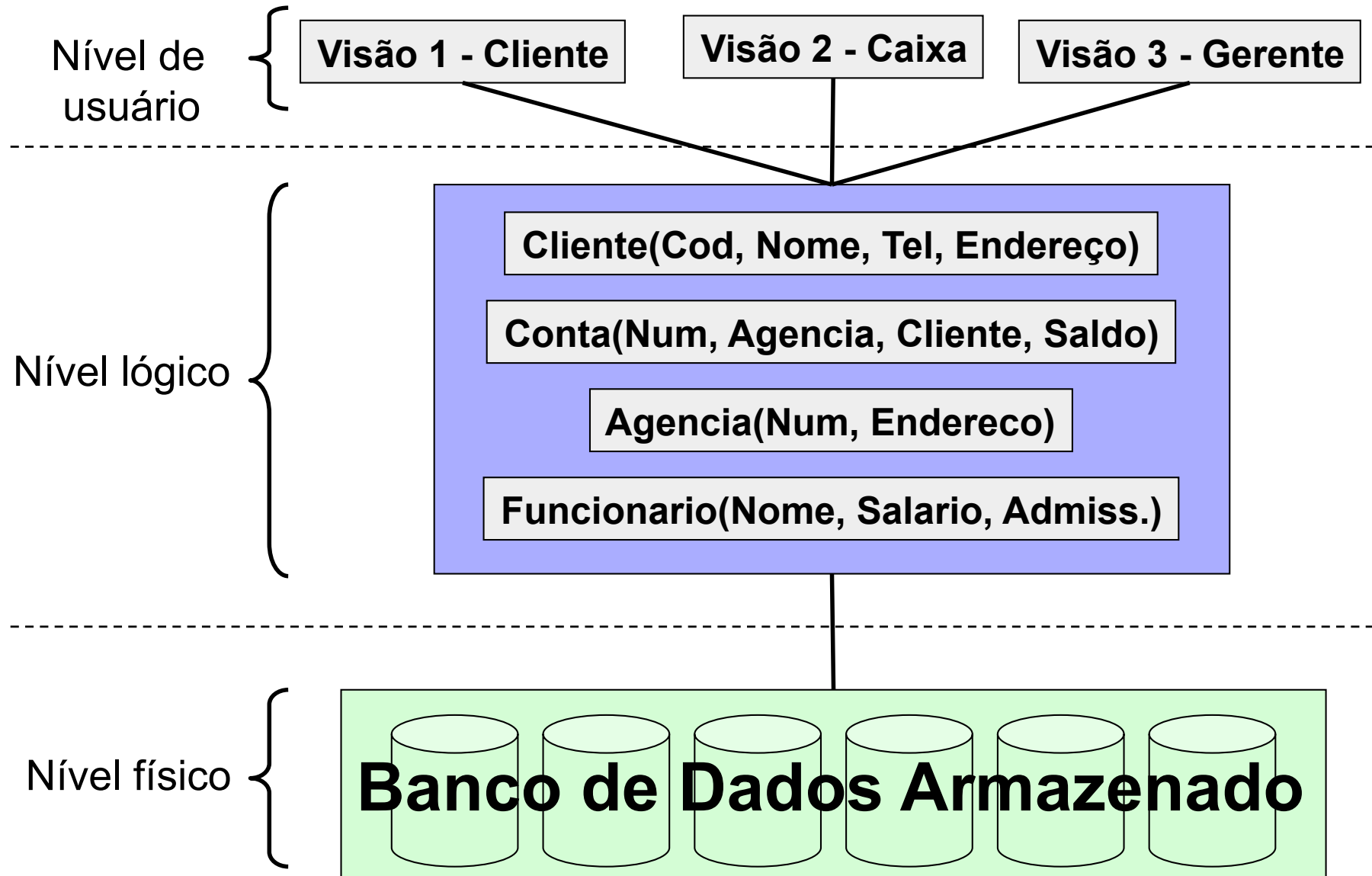


# Abstração de dados

- Nível de visões (de usuário)
  - Alto nível de abstração
  - Diferentes usuários podem ter diferentes visões do BD
- Nível lógico ou conceitual
  - Descreve *quais* dados estão armazenados e as relações entre eles
- Nível físico
  - Descreve *como* os dados estão armazenados
  - Baixo nível de abstração
  - Estruturas complexas e detalhadas



# Abstração de dados







# Projeto de um Banco de Dados

- Começa com um levantamento de requisitos
- Modelagem conceitual
- Modelagem lógica
- Refinamento e projeto físico
- Termina com o projeto físico do banco



# Modelo Entidade-Relacionamento (E-R)

- Introduzido em 1976 por Peter Chen, é a abordagem mais adotada para modelagem conceitual de dado. Objetivo é facilitar o projeto do banco de dados representando sua estrutura lógica
- Definição: modelo baseado na percepção do mundo real como um conjunto de objetos chamados entidades e pelo conjunto de relacionamentos entre esses objetos.
- Ao longo do tempo diferentes versões de foram sendo propostas para representar diferentes conceitos.



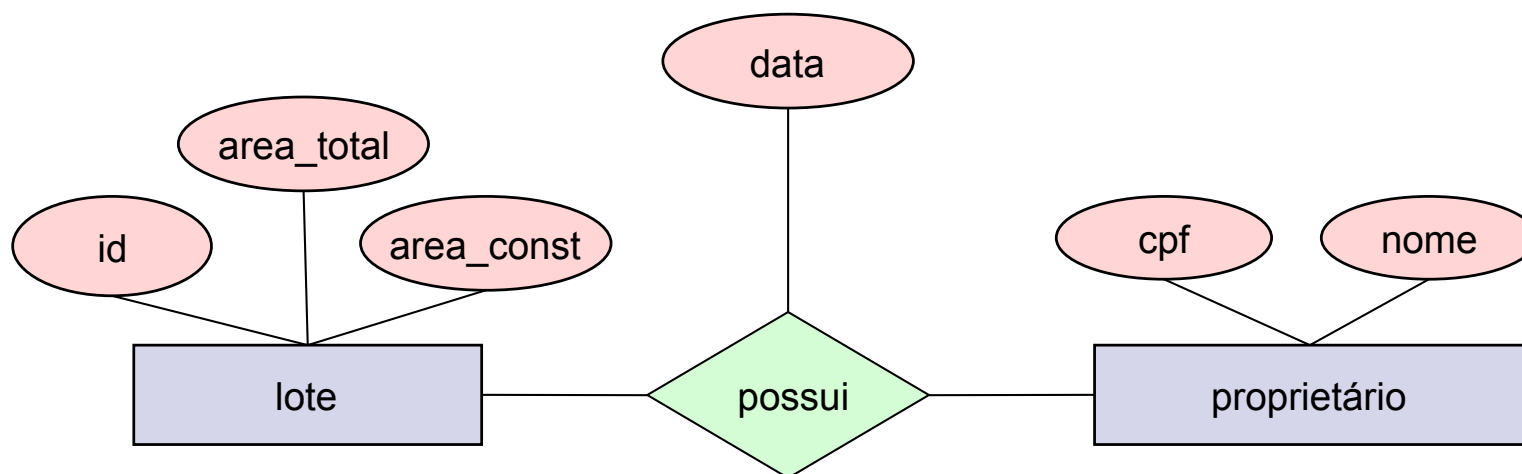
# Modelo Entidade-Relacionamento (E-R)

- Conceitos básicos:
  - Entidades
    - Objetos básicos do mundo real
    - Um conjunto de entidades agrupa entidades do mesmo tipo
  - Relacionamentos
    - Associação entre conjuntos de entidades
  - Atributos
    - Associados a entidades e a relacionamentos
    - Uma entidade é representada por um conjunto de atributos
    - Cada atributo possui um domínio



# Modelo Entidade-Relacionamento (E-R)

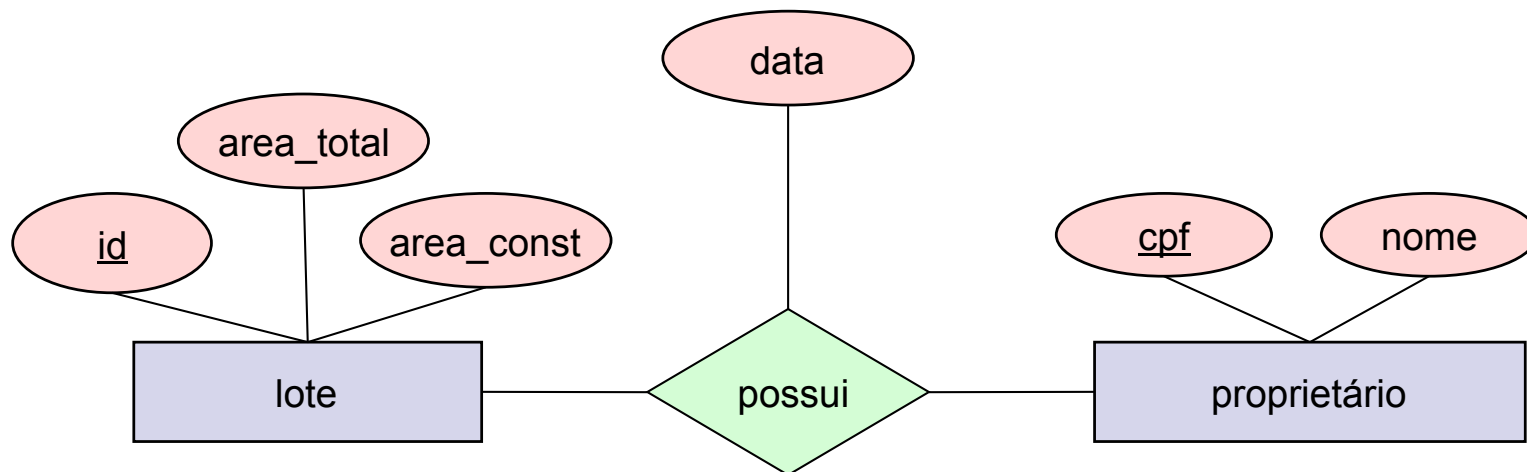
- Representação:
  - Retângulos: conjunto de entidades
  - Elipses: atributos
  - Losangos: relacionamentos





# Modelo Entidade-Relacionamento (E-R)

- Atributo identificador
  - Atributo que vai identificar unicamente cada instância da entidade (chave primária)
  - Devem ser sublinhados

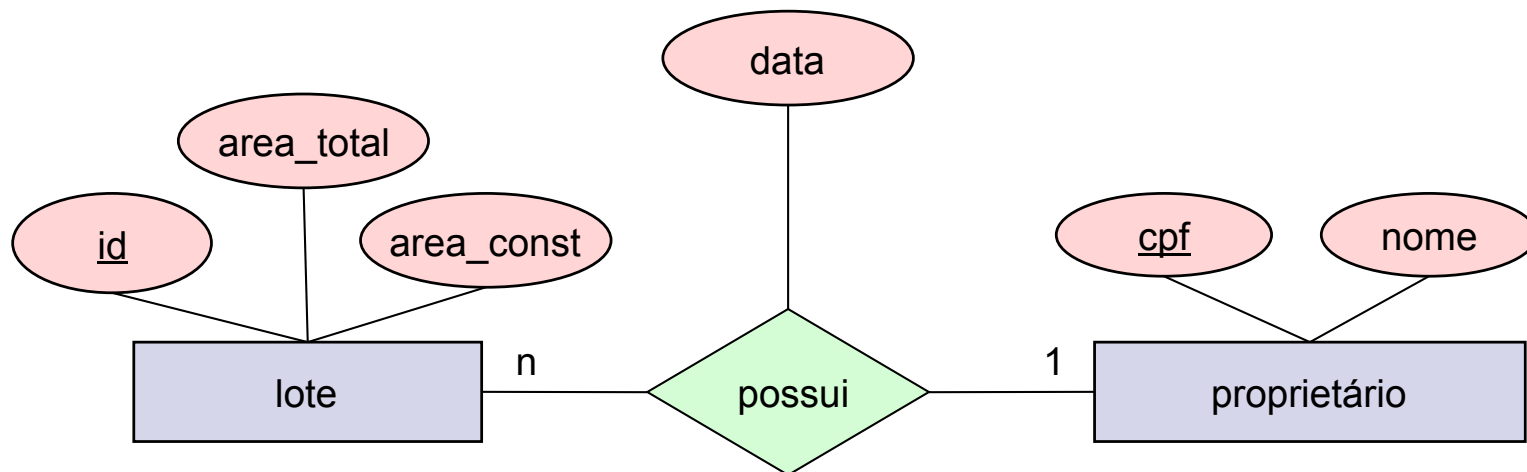






# Modelo Entidade-Relacionamento (E-R)

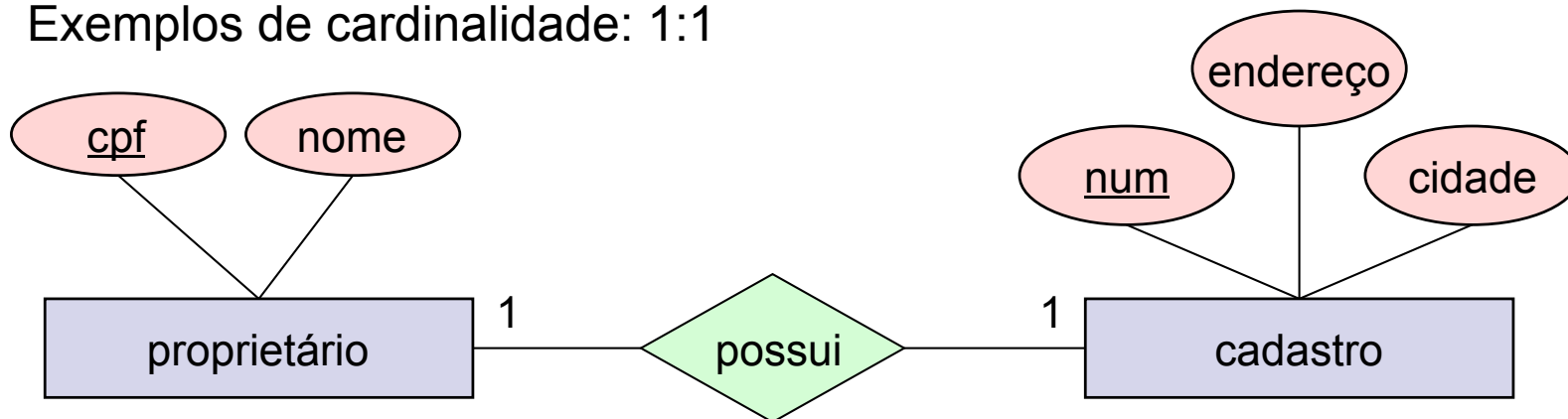
- Cardinalidade:
  - expressa o número de entidades que uma entidade pode estar associada com
  - 1:1, 1:n, n:1, n:n





# Modelo Entidade-Relacionamento (E-R)

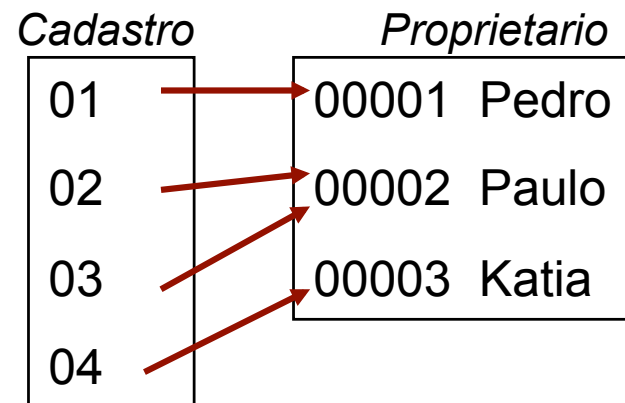
- Exemplos de cardinalidade: 1:1



Relação correta



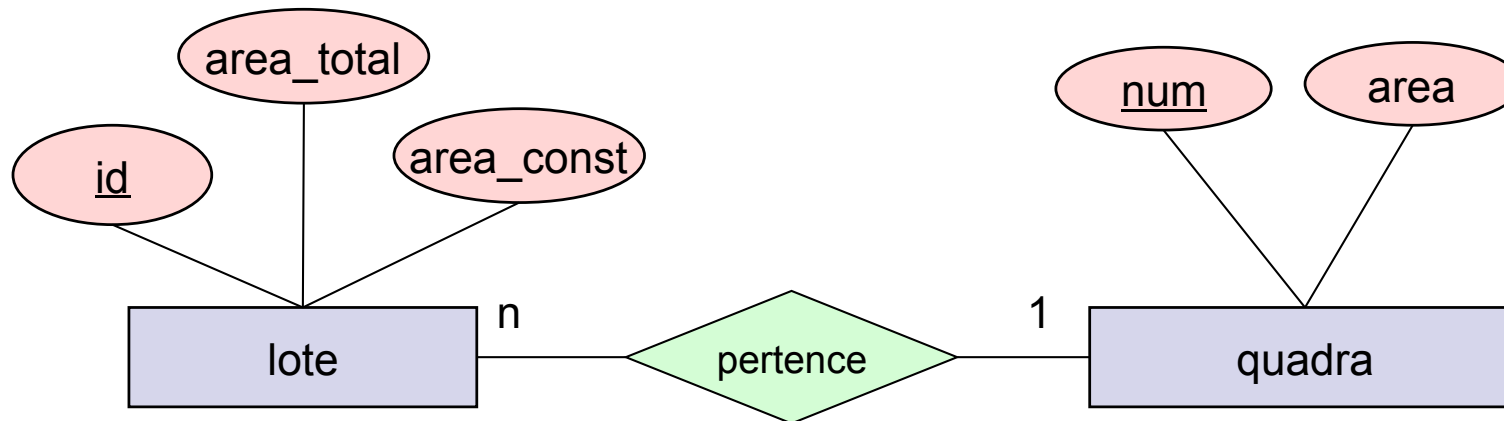
Relação Incorreta



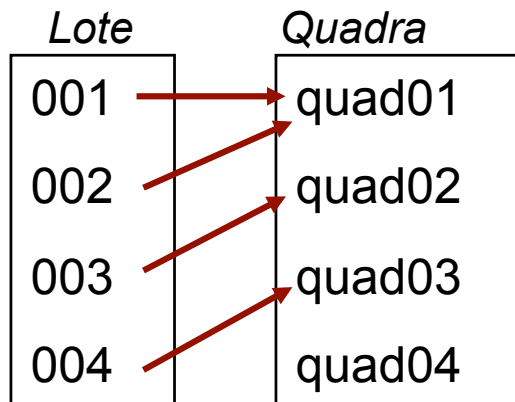


# Modelo Entidade-Relacionamento (E-R)

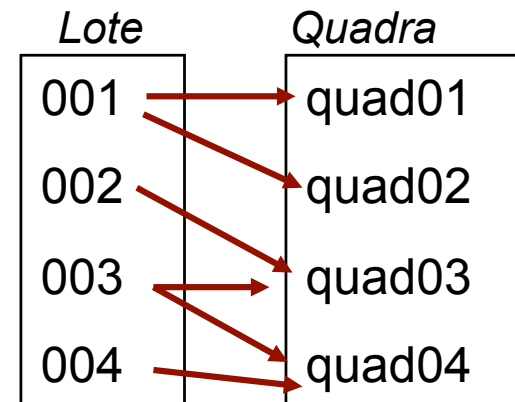
- Exemplos de cardinalidade: 1:N



Relação correta



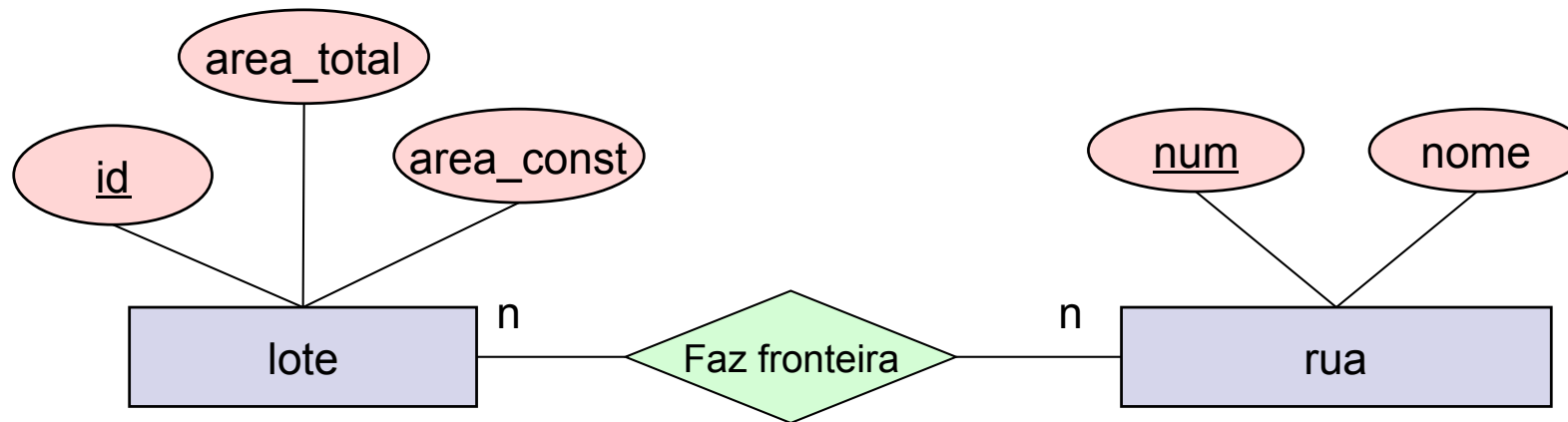
Relação incorreta



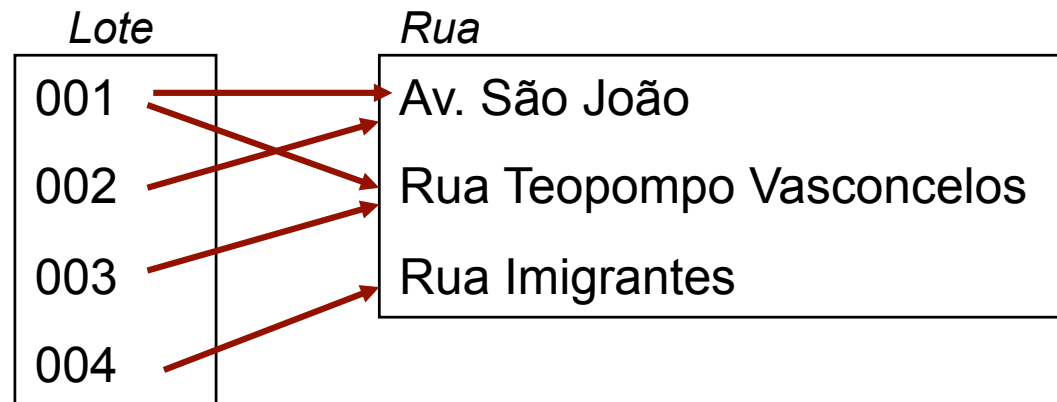


# Modelo Entidade-Relacionamento (E-R)

- Exemplos de cardinalidade: N:N



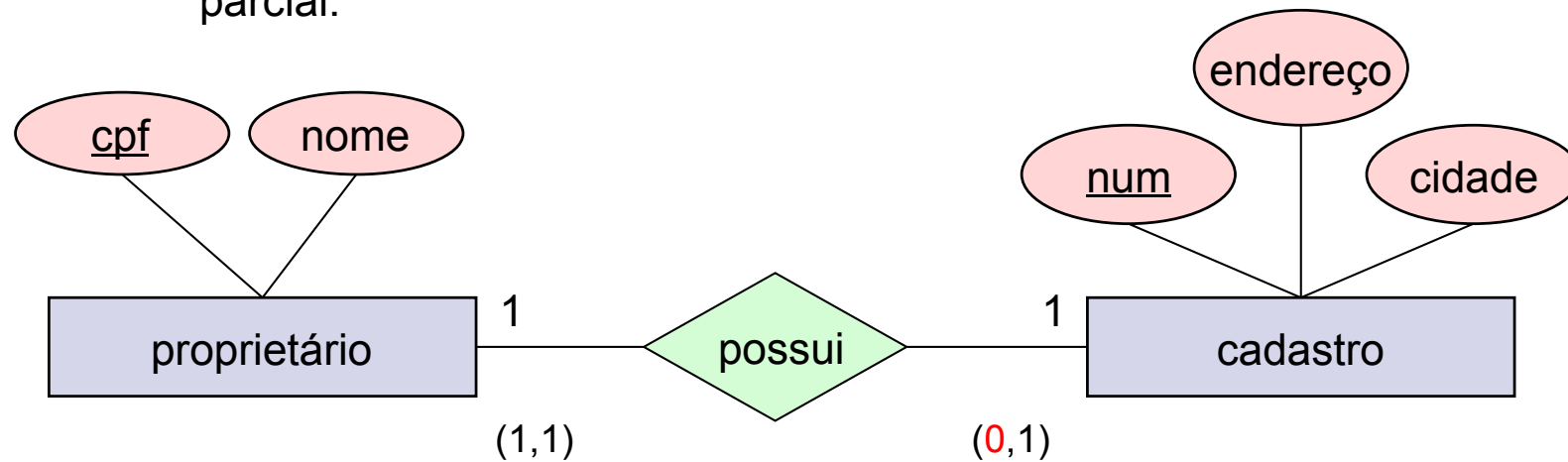
Relação correta





# Modelo Entidade-Relacionamento (E-R)

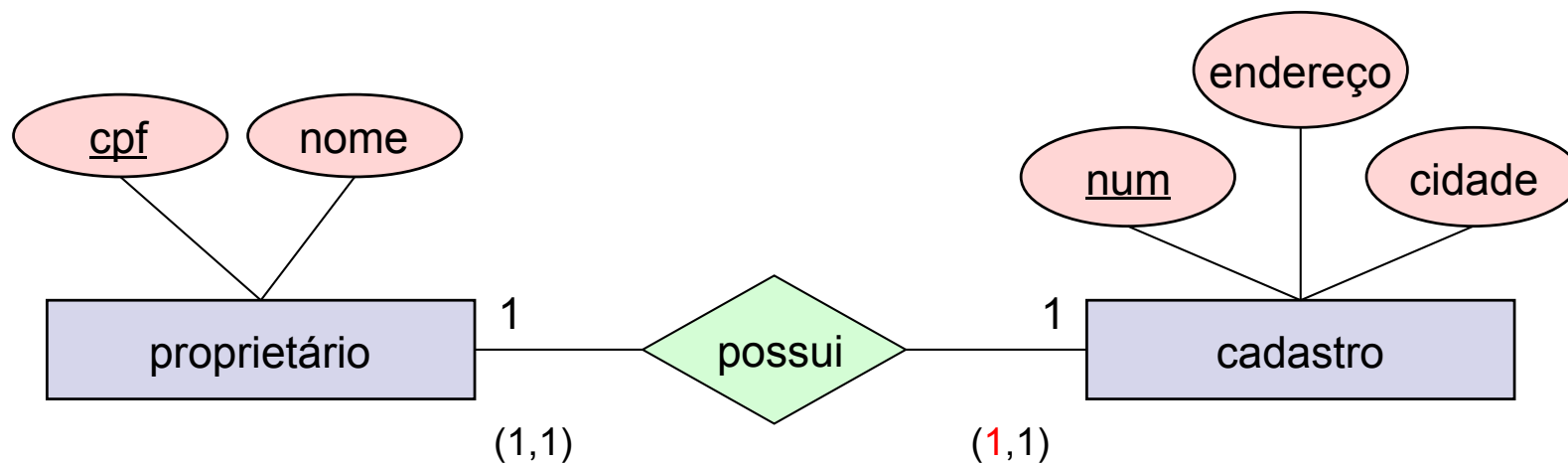
- Relacionamento parcial:
  - Quando não existe obrigatoriedade de todas as entidades de um conjunto participarem do relacionamento
    - Ex. Podem existir proprietários sem cadastro
    - A participação da entidade “proprietário” no relacionamento “possui” é parcial.





# Modelo Entidade-Relacionamento (E-R)

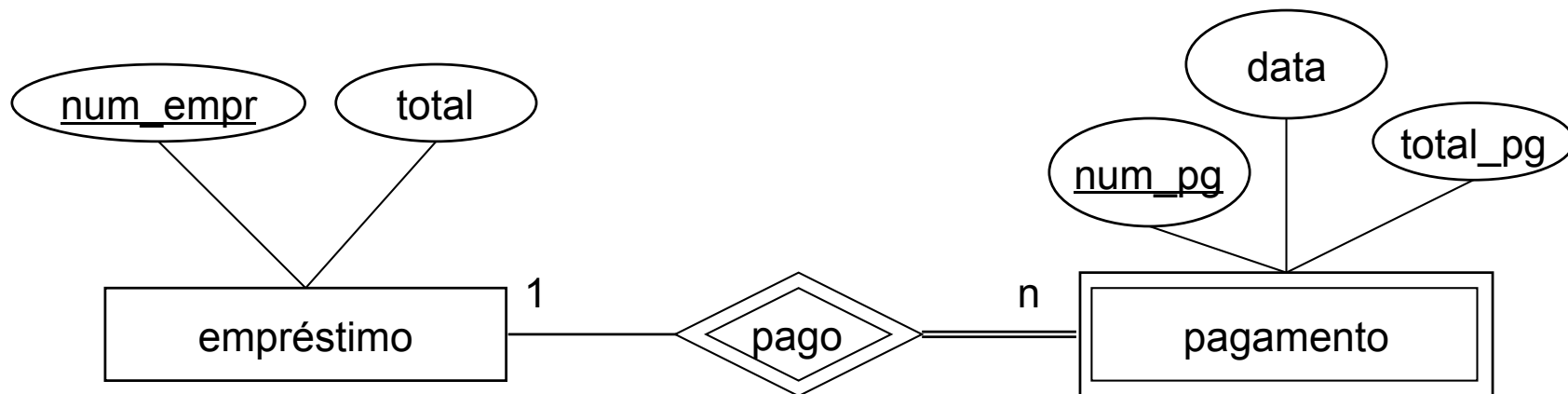
- Relacionamento total (dependência de existência):
  - Quando todas as entidades de um conjunto participam obrigatoriamente do relacionamento
    - Ex. Todo cadastro tem que estar associado a um proprietário.
    - A participação da entidade “cadastro” no relacionamento “possui” é total.





# Modelo Entidade-Relacionamento (E-R)

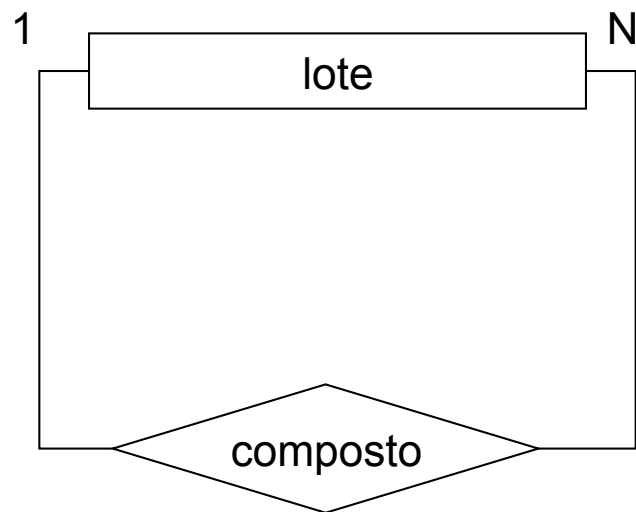
- Entidade fraca:
  - Quando a existência dessa entidade depende da existência de uma outra entidade (dependência de existência).





# Modelo Entidade-Relacionamento (E-R)

- Auto relacionamento:
  - Associa entidades de um conjunto a entidades desse mesmo conjunto
    - Ex. Um lote pode ser composto por outros lotes

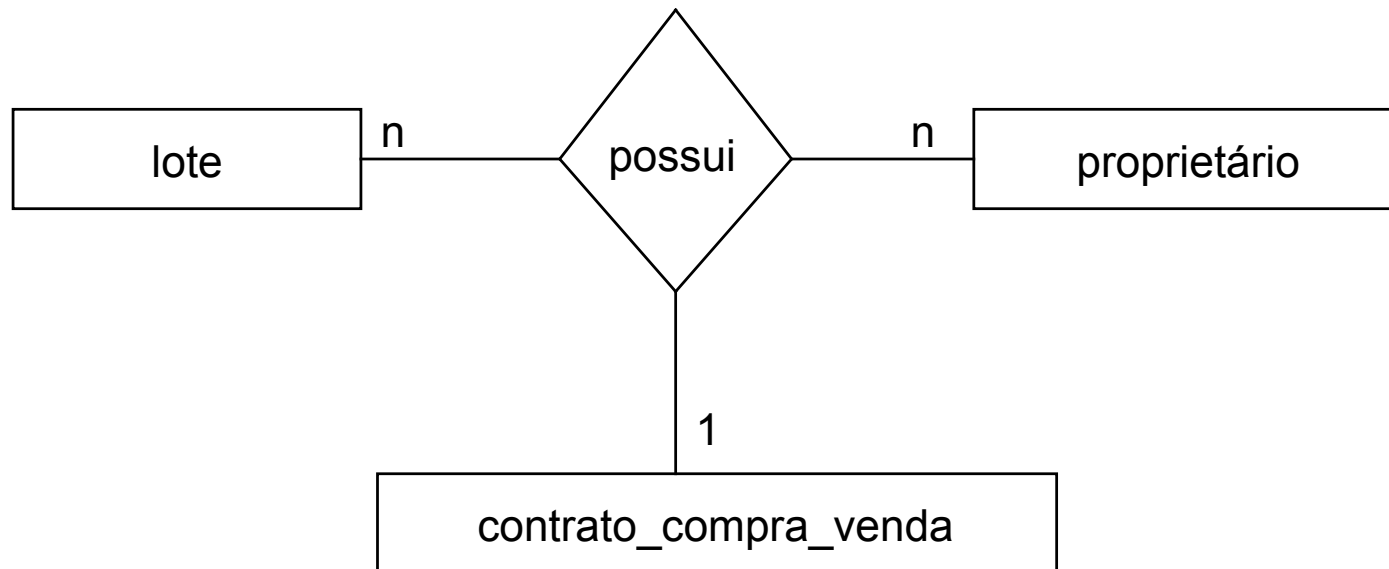






# Modelo Entidade-Relacionamento (E-R)

- Relacionamento múltiplo:
  - Associa mais de dois conjuntos de entidades
    - Ex. Um proprietário pode possuir vários lotes, cada um com um contrato de compra e venda diferente.





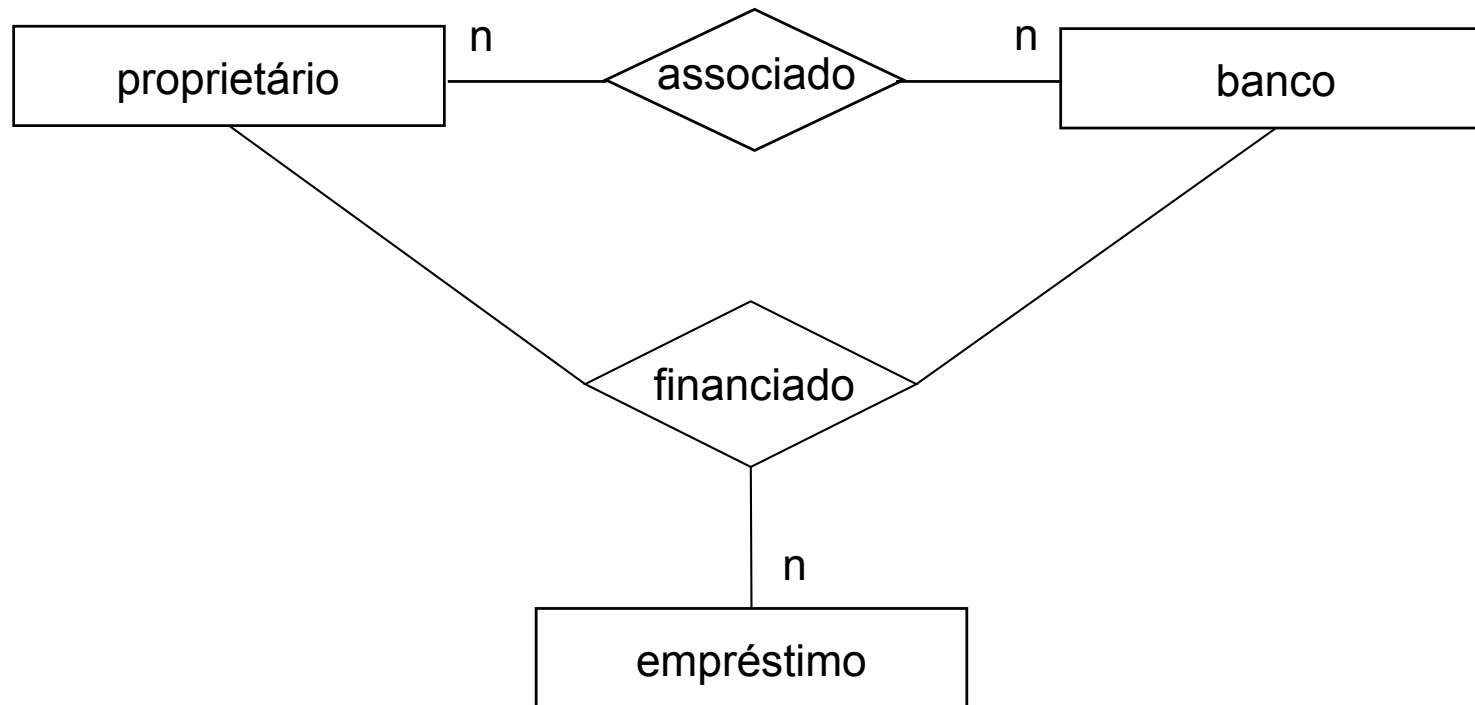
# Modelo Entidade-Relacionamento (E-R)

- Agregação:
  - Uma limitação do ER: expressar relacionamentos entre relacionamentos
  - Agregação é uma abstração na qual os relacionamentos são tratados como conjunto de entidades de nível superior, permitindo o relacionamento entre relacionamentos.



# Modelo Entidade-Relacionamento (E-R)

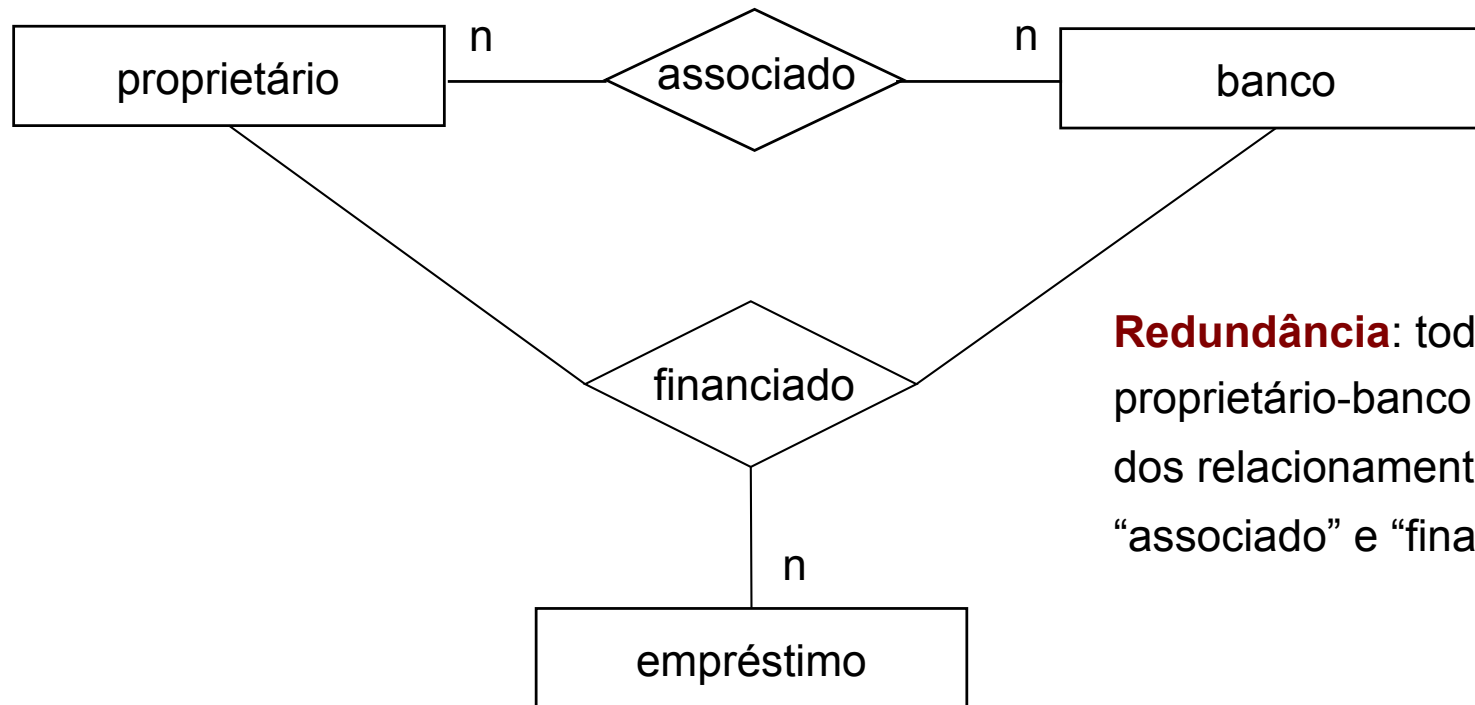
- Ex.: um proprietário pode estar associado a vários bancos. Pra cada banco associado, o proprietário pode ter um ou mais empréstimos.





# Modelo Entidade-Relacionamento (E-R)

- Ex.: um proprietário pode estar associado a vários bancos. Pra cada banco associado, o proprietário pode ter um ou mais empréstimos.

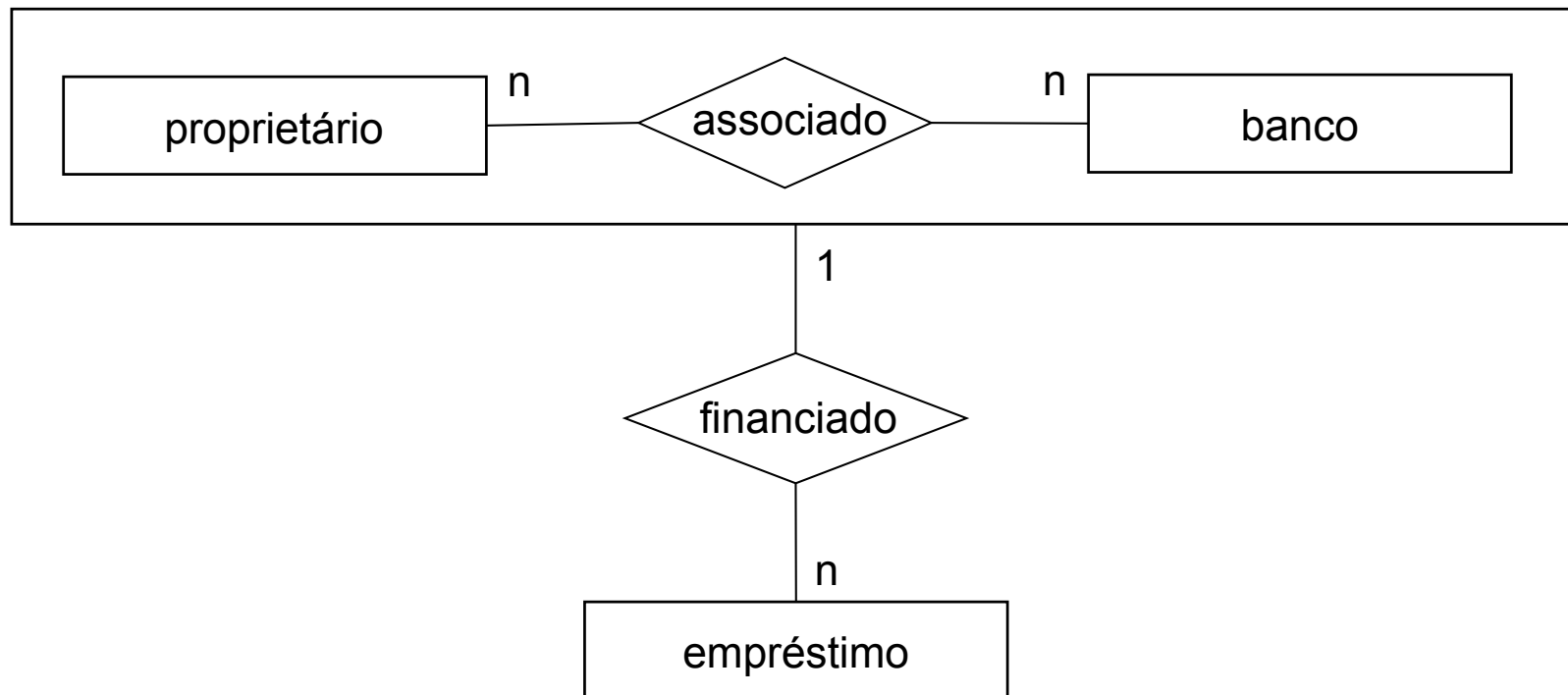


**Redundância:** todo par proprietário-banco faz parte dos relacionamentos “associado” e “financiado”



# Modelo Entidade-Relacionamento (E-R)

- Agregação:
  - Ex.: um proprietário pode estar associado a vários bancos. Pra cada banco associado, o proprietário pode ter um ou mais empréstimos.





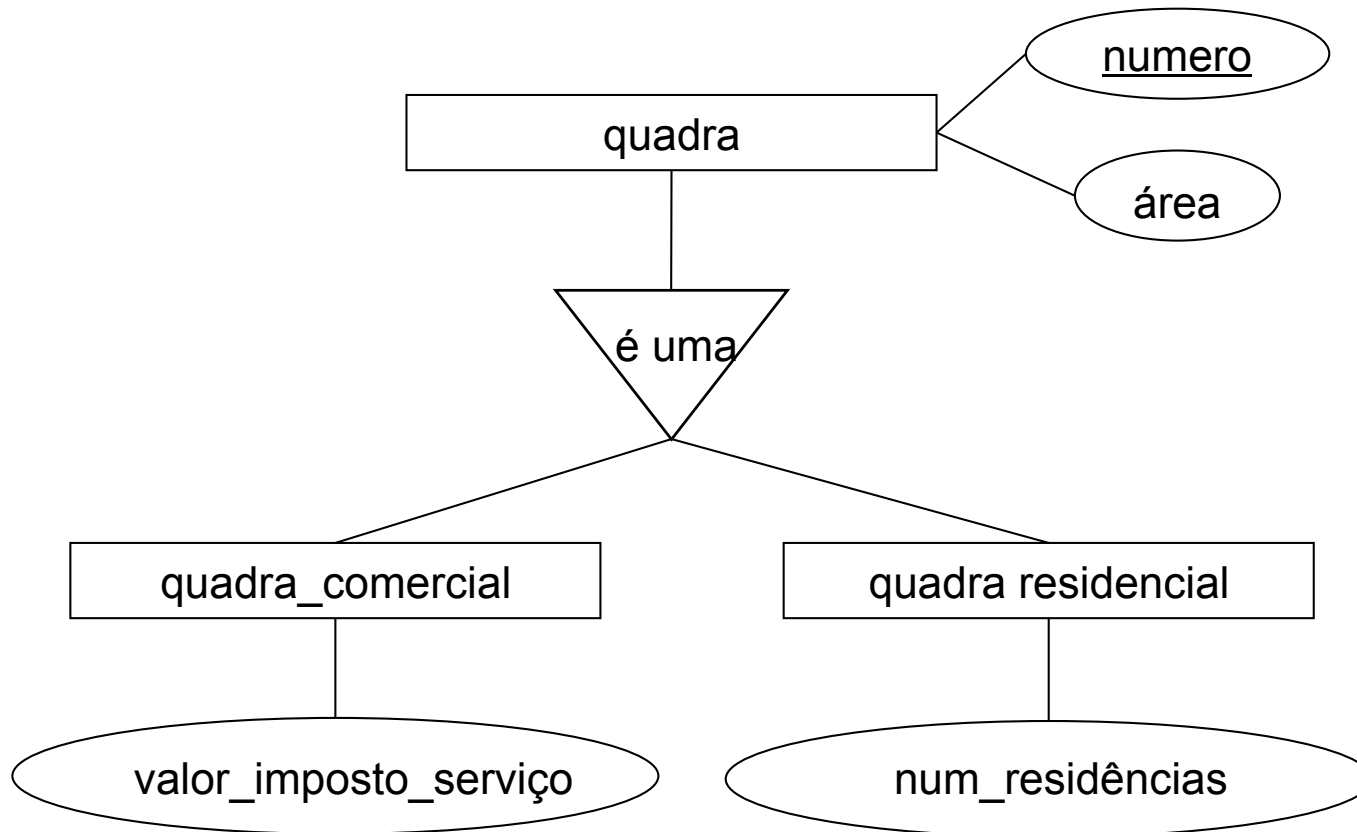
# Modelo Entidade-Relacionamento (E-R)

- Generalização
  - Enfatiza a semelhança entre diferentes tipos de entidades e abstrai suas diferenças.
  
- Especialização:
  - Subdivide entidades semelhantes em conjuntos de entidades mais específicas e enfatiza suas diferenças.



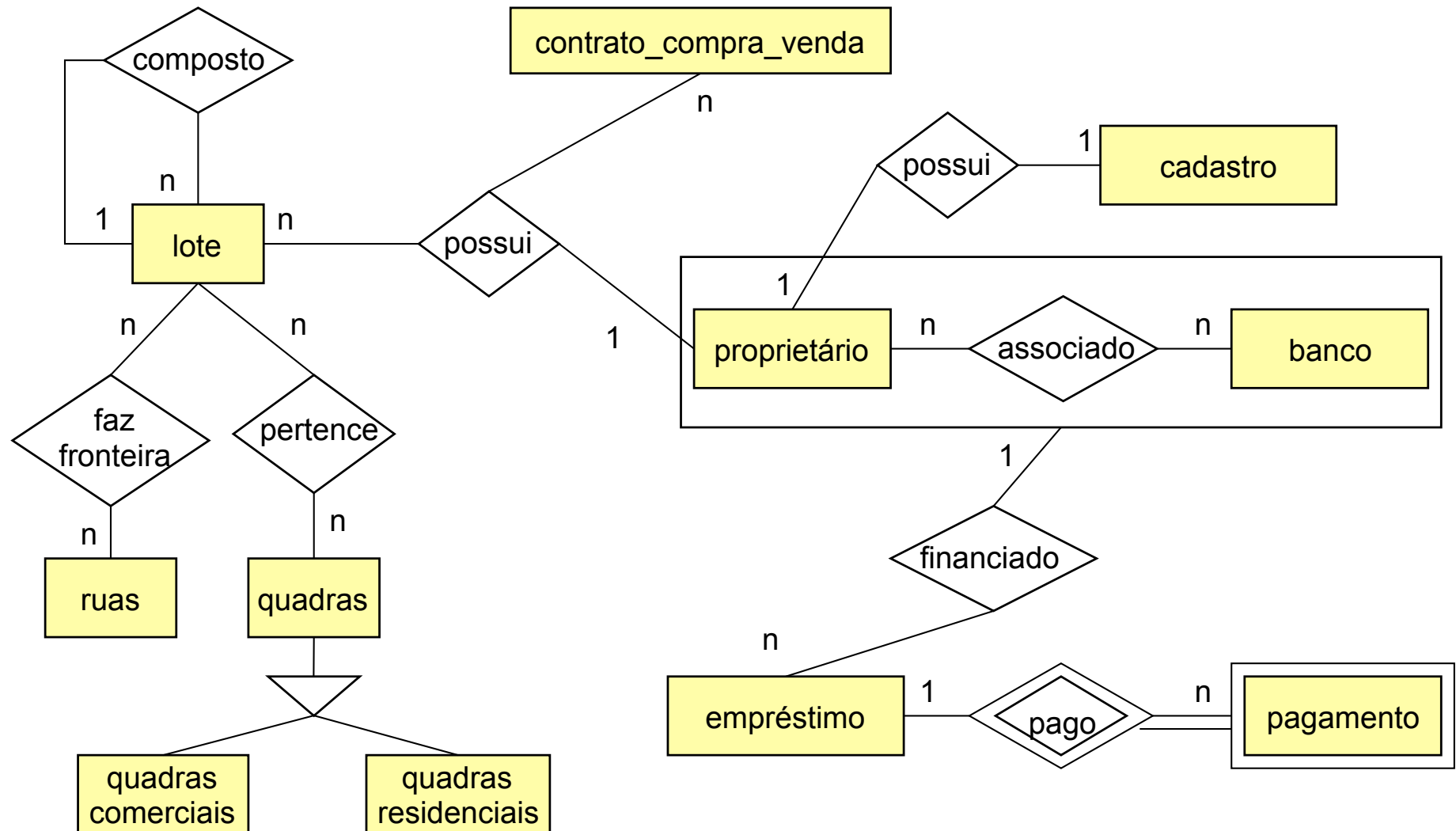
# Modelo Entidade-Relacionamento (E-R)

- Generalização e Especialização
  - Ex.: Uma quadra pode ser do tipo comercial ou residencial





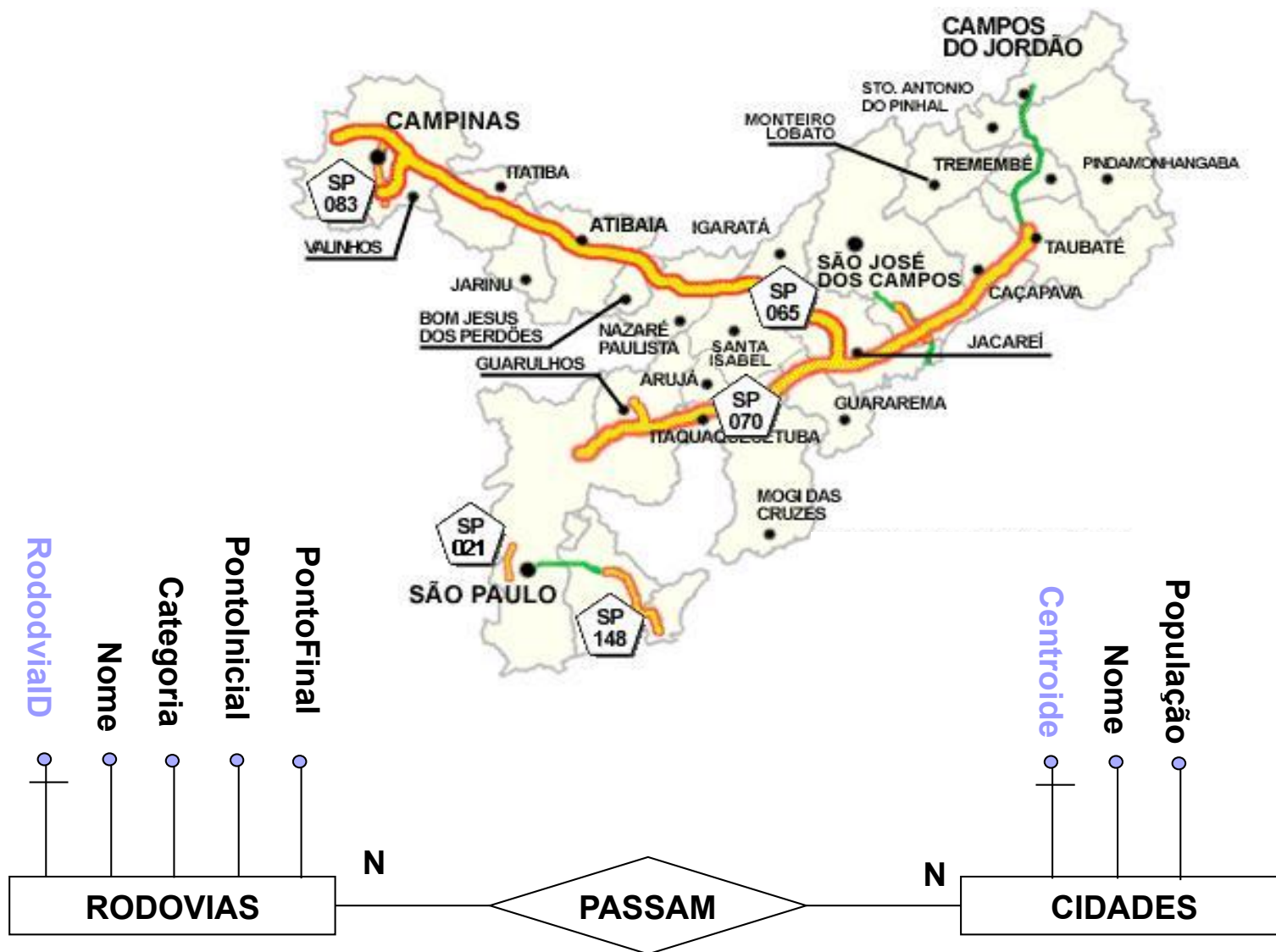
# Modelo Entidade-Relacionamento (E-R)







# Modelo Entidade-Relacionamento (E-R)





# O que é um bom modelo?

- Facilita a descrição de consultas corretas e compreensíveis
- Os elementos do modelo tem um significado intuitivo
- O modelo é o mais simples possível, mas não mais simples que isso
- É suficientemente abstrato para não sofrer alterações face a alterações menores no domínio do problema
- Se o domínio do problema se altera significativamente é fácil modificar o modelo de dados (suficientemente flexível)
- Mais tarde é necessário considerar o impacto do modelo de dados na eficiência das operações no banco de dados



# Decisões de projeto

- Um conceito deve ser modelado como entidade ou atributo?
- Um conceito deve ser modelado como uma entidade ou um relacionamento?
- Identificação dos relacionamentos: binário, ternário? Agregação?
- Restrições no modelo ER:
- Muita semântica pode (e deve) ser capturada mas algumas restrições não podem ser capturadas nos diagramas ER



# Exemplo: Entidade X Atributo

- Exemplo: o *endereço* de um *empregado* deve ser modelado como um atributo ou uma entidade?
- Depende do problema:
  - se um empregado pode ter mais que um endereço então ele deve ser uma entidade separada porque atributos só possuem um valor
  - se faremos consultas sobre partes do endereço (ex. cidade) então ele deve ser uma entidade separada porque atributos devem ser atômicos



# Modelo Relacional

- O modelo relacional foi inventado por E.F. (Ted) Codd como um modelo geral de dados (~1970)
- Baseia-se no princípio de que todos os dados são representados matematicamente como relações *n-árias*, onde uma *relação n-ária* é um subconjunto do produto cartesiano de  $n$  domínios
- Dados são manipulados através de um *cálculo* ou *álgebra relacional*
- Primeiro modelo de dados para Sistemas de Bancos de Dados comerciais
- É usado na maioria das aplicações para dados tradicionais
- Os SGBDs relacionais são baseados no modelo relacional
  - IBM, Informix, Microsoft, Oracle, Sybase, etc.




# Modelo Relacional

- Banco de Dados Relacional
  - Coleção de tabelas, compostas por colunas e linhas inter-relacionadas

## PROPRIETARIO

CPF	NOME	RUA	NUMERO	BAIRRO
08940256	JOÃO DA SILVA	SAO JOAO	180	CENTRO
03727298	HENRIQUE CARDOSO	IMIGRANTE	1700	VILA 12
97260089	JOSÉ DE SOUZA	SAO JOAO	35	CENTRO

## LOTE



NUMERO	PROPRIETARIO_CPF	AREA_TOTAL	AREA_CONST
00001	08940256	400.000	0
00003	03727298	150.000	75.00
00039	03727298	500.000	0



# Modelo Relacional

- Características:
  - Consiste em um conjunto de tabelas ou *relações* formadas por linhas e colunas
  
  - Cada coluna (campo):
    - Representa um atributo
    - Está associada a um domínio (conjunto de valores permitidos)
  
  - Cada linha (registro ou *tupla*):
    - Representa um relacionamento entre um conjunto de valores para cada atributo



# Modelo Relacional

- Conceito de *relação*
  - Define uma tabela do banco de dados
  - Domínio de um atributo: conjunto de possíveis valores

- Ex.:

$$D_1 = \{ x \in \mathfrak{R} \mid x \geq -5 \text{ e } x \leq 5 \}$$

$$D_2 = \{ y \in \mathfrak{R} \mid y \geq 0 \}$$





# Modelo Relacional

- Conceito de *relação*

- Dados os domínios  $D_1, D_2, \dots, D_n$  não necessariamente distintos, uma relação é definida como:

$$R = \{ (d_1, d_2, \dots, d_n) \mid d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n \}$$

- O conjunto  $(d_1, d_2, \dots, d_n)$  de valores ordenados define uma *tupla*
- Uma relação é o conjunto de *n-tuplas* ordenadas, onde  $n$  define o grau da relação



# Modelo Relacional

- Exemplo de *relação*

## PROPRIETARIO

CPF	NOME	RUA	NUMERO	BAIRRO
08940256	JOÃO DA SILVA	SAO JOAO	180	CENTRO
03727298	HENRIQUE CARDOSO	IMIGRANTE	1700	VILA 12
97260089	JOSÉ DE SOUZA	SAO JOAO	35	CENTRO

### Atributo

cpf

nome

rua

numero

bairro

### Domínio

inteiro longo positivo

conjunto de caracteres

conjunto de caracteres

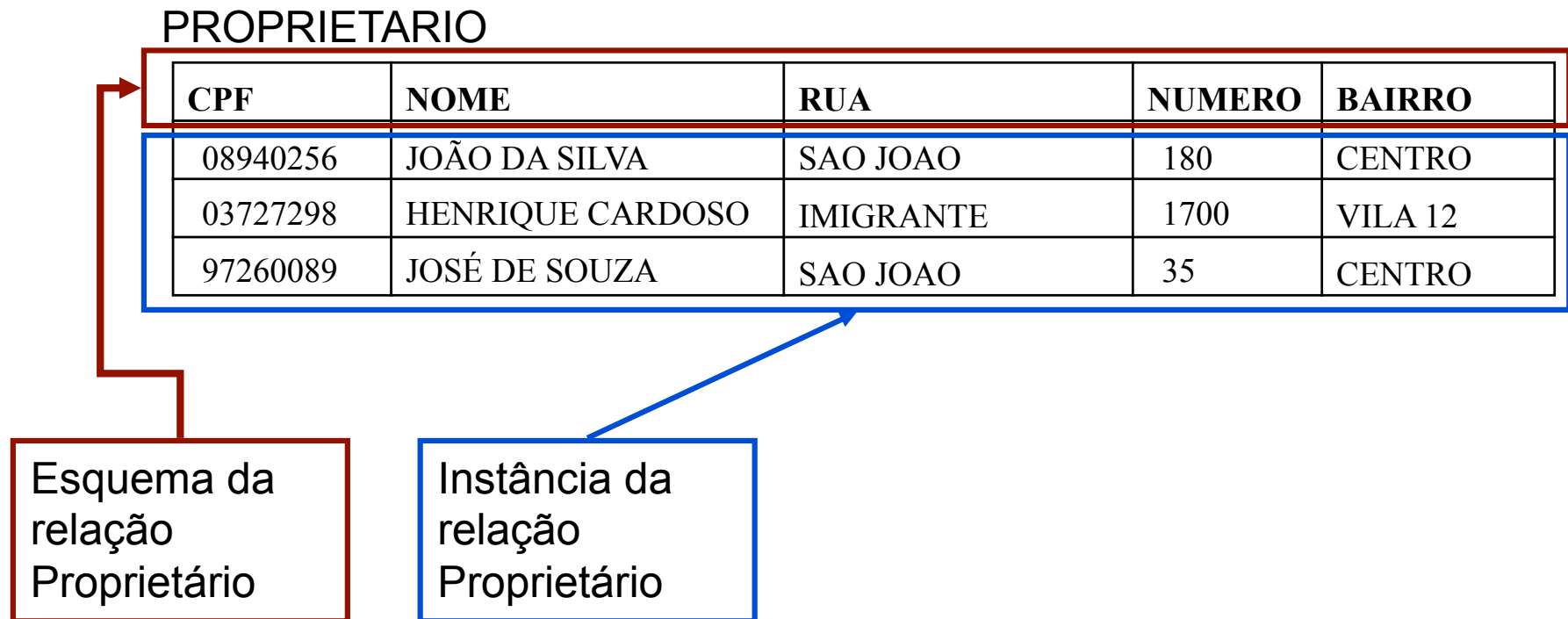
inteiro positivo

conjunto de caracteres



# Modelo Relacional

- Esquema x instância de *relação*





# Modelo Relacional

- Restrições de integridade
  - Uma das funcionalidades básicas que todo SGBD deve oferecer
  
  - É uma regra de consistência de dados que é garantida pelo SGBD
  
  - Alguns tipos de Restrições:
    - **Restrição de domínio**
    - **Restrição de chave**
    - **Integridade Referencial**



# Modelo Relacional

- Super-chave:
  - Conjunto de um ou mais atributos que permitem identificar cada registro da tabela como único.
  
- Chave candidata:
  - corresponde a super-chave mínima, ou seja, não existe sub-conjunto da chave candidata.
    - { cpf, nome } chave candidata?
    - { cpf } chave candidata?
  
- Chave primária:
  - chave candidata escolhida no projeto da tabela do banco para identificar unicamente cada registro ou *tupla*.



# Modelo Relacional

- Chave primária:
  - Coluna ou combinação de colunas cujos valores distinguem uma linha ou registro das demais dentro de uma tabela
  - Restrição de chave

chave primária

PROPRIETARIO

CPF	NOME	RUA	NUMERO	BAIRRO
08940256	JOÃO DA SILVA	SAO JOAO	180	CENTRO
03727298	HENRIQUE CARDOSO	IMIGRANTE	1700	VILA 12
97260089	JOSÉ DE SOUZA	SAO JOAO	35	CENTRO



# Modelo Relacional

- Chave primária:
  - Coluna ou combinação de colunas cujos valores distinguem uma linha ou registro das demais dentro de uma tabela
  - Restrição de chave

chave primária composta

				RUA
RUA	TRECHO	NUM_INICIAL	NUM_FINAL	BAIRRO
SAO JOAO	TRC01	0	180	CENTRO
SAO JOAO	TRC02	190	1700	CENTRO
IMIGRANTES	TRC01	0	500	VILA 1



# Modelo Relacional

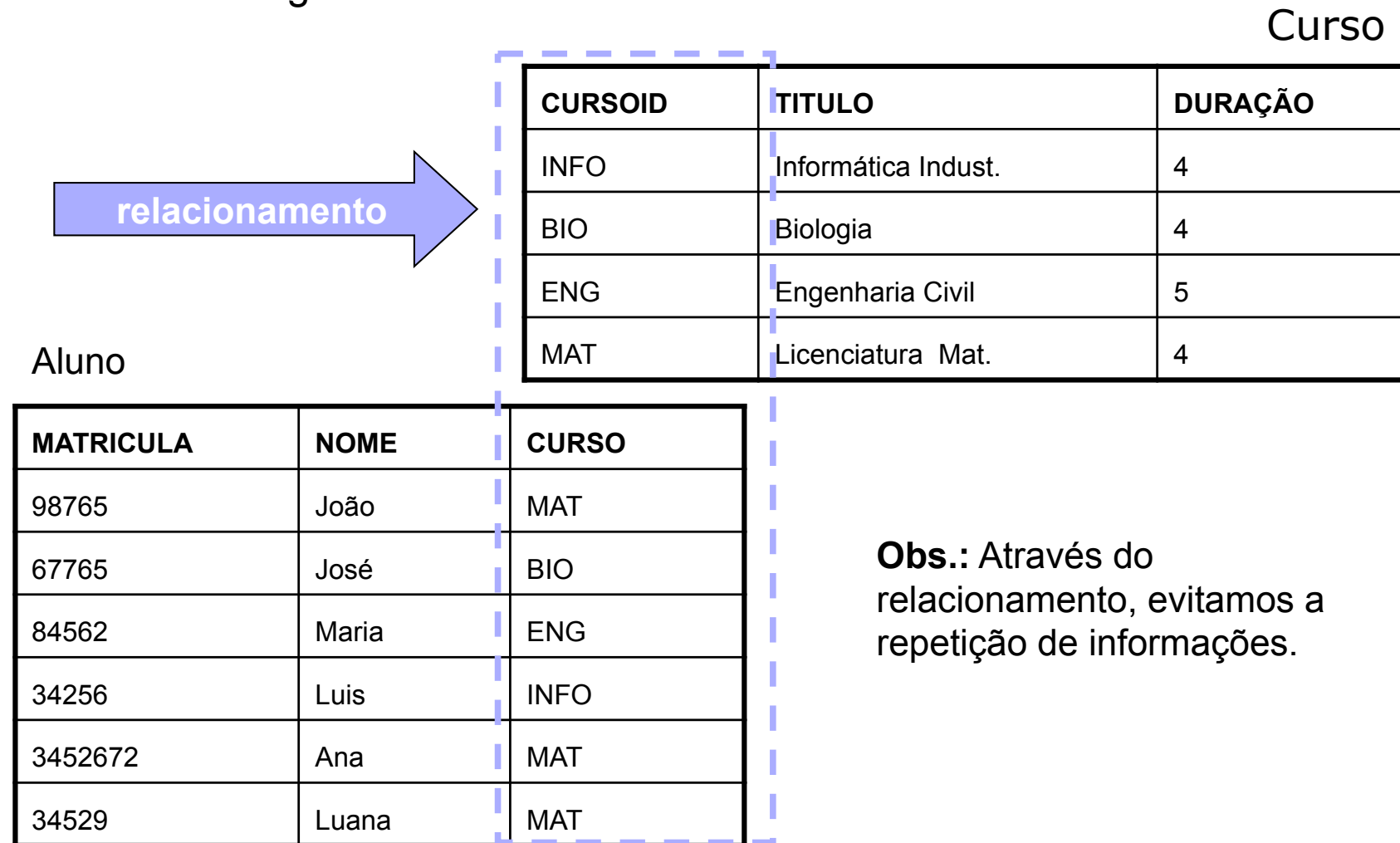
- Chave estrangeira
  - Implementa a restrição de integridade referencial.
  - Coluna ou combinação de colunas, cujos valores aparecem necessariamente na chave primária de uma outra tabela.
  - Mecanismo que permite a implementação de relacionamentos em um banco de dados relacional.





# Modelo Relacional

- Chave estrangeira





# Modelo Relacional

- Chave estrangeira
  - Uma chave estrangeira não precisa ser uma chave primária na sua relação.
  - Uma chave estrangeira não precisa ter o mesmo nome do que a chave primária correspondente na outra tabela (apenas o mesmo domínio).



# Modelo Relacional

- Chave estrangeira impõe restrições que devem ser garantidas ao serem executadas no BD:
  - Inclusão de uma linha na tabela que contém a chave estrangeira:
    - Garantir que o valor da chave estrangeira exista na chave primária da outra tabela.
  
  - Alteração do valor da chave estrangeira:
    - O novo valor deve aparecer na coluna da chave primária referenciada.



# Modelo Relacional

- Chave estrangeira impõe restrições que devem ser garantidas ao serem executadas no BD:
  - Exclusão de uma linha da tabela que contém a chave primária referenciada por uma chave estrangeira:
    - Não se exclui a linha caso exista um valor na tabela com a chave estrangeira.
    - Remove-se também a linha com o valor de chave estrangeira.
    - Valor da chave estrangeira é ajustado como NULL.



# Modelo Relacional

- Chave estrangeira impõe restrições que devem ser garantidas ao serem executadas no BD:
  - Alteração do valor da chave primária referenciada por alguma chave estrangeira:
    - Propagar a modificação
    - Não deixar que seja feita a modificação



# Álgebra Relacional

- Linguagem de consultas procedural
- Conjunto de operações que usam uma ou duas relações como entrada e geram uma relação de saída
  - *operação*  $(REL_1) \rightarrow REL_2$
  - *operação*  $(REL_1, REL_2) \rightarrow REL_3$
- Operações básicas:
  - Operações unárias:
    - seleção, projeção, renomeação
- Operações binárias:
  - produto cartesiano, união e diferença



# Operadores da Álgebra Relacional

- Seleção:

- seleciona tuplas que satisfazem um certo predicado ou condição

## Cientes

<i><b>Nome</b></i>	<i><b>Registro</b></i>
João	1
Maria	2
José	3

a) selecionar tuplas cujo nome = João

$\sigma_{\text{nome}=\text{"João"}}(\text{Clientes})$

<i><b>Nome</b></i>	<i><b>Registro</b></i>
João	1



# Operadores da Álgebra Relacional

b) selecionar as tuplas de Clientes cujo registro > 1

$\sigma_{\text{registro} > 1}$  (Clientes)

<b>Nome</b>	<b>Registro</b>
Maria	2
José	3

c) selecionar as tuplas de Clientes com registro > 1 e registro < 3

$\sigma_{\text{registro} > 1 \wedge \text{registro} < 3}$  (Clientes)

<b>Nome</b>	<b>Registro</b>
Maria	2





# Operadores da Álgebra Relacional

- Projeção:
  - gera novas relações excluindo alguns atributos
  - exemplo: projete o atributo nome sobre a relação Clientes

$\Pi_{\text{nome}}(\text{Clientes})$

Clientes

<b><i>Nome</i></b>	<b><i>Registro</i></b>
João	1
Maria	2
José	3



<b><i>Nome</i></b>
João
Maria
José



# Operadores da Álgebra Relacional

- União:
  - união de atributos do mesmo domínio que estão em relações diferentes
  - as relações devem possuir o mesmo número de atributos
  - exemplo: encontre todos os clientes da agência que possuem conta corrente ou empréstimo.
    - Relações existentes na agência:
      - ContaCorrente e Empréstimo



# Operadores da Álgebra Relacional

- União:  $\Pi_{\text{nome}}(\text{ContaCorrente}) \cup \Pi_{\text{nome}}(\text{Emprestimo})$

ContaCorrente

<b>Nome</b>	<b>Conta</b>
João	1
Maria	2
José	3

Emprestimo

<b>Nome</b>	<b>Empréstimo</b>
Paulo	100
Maria	200
Carlos	300

Resultado da união

=

<b>Nome</b>
João
Maria
José
Paulo
Carlos



# Operadores da Álgebra Relacional

- Diferença:
  - tuplas que se encontram em uma relação, mas não em outra
  - exemplo: encontre todos clientes sem empréstimo

$$\Pi_{\text{nome}}(\text{ContaCorrente}) - \Pi_{\text{nome}}(\text{Emprestimo})$$

ContaCorrente

<b>Nome</b>	<b>Conta</b>
João	1
Maria	2
José	3

-

Emprestimo

<b>Nome</b>	<b>Empréstimo</b>
Paulo	100
Maria	200
Carlos	300

=

Resultado da diferença

<b>Nome</b>
João
José



# Operadores da Álgebra Relacional

- Produto Cartesiano

- Faz todas as combinações entre as tuplas de duas relações
- Gera uma nova relação formada pela união dessas combinações
- Exemplo: produto cartesiano entre os clientes e os empréstimos de Maria

$\sigma_{\text{emprestimo.nome} = \text{“Maria”}}$  (ContaCorrente X Empréstimo)

<b><i>Nome<sub>cc</sub></i></b>	<b><i>Conta</i></b>	<b><i>Nome<sub>emp</sub></i></b>	<b><i>Empréstimo</i></b>
João	1	Maria	200
Maria	2	Maria	200
José	3	Maria	200



# Operadores da Álgebra Relacional

- Operadores derivados:

- Intersecção

- Seleciona tudo que está em ambas relações
    - Exemplo: todos os clientes que possuem empréstimo

$\Pi_{\text{nome}}$  (ContaCorrente)

$\Pi_{\text{nome}}$  (Emprestimo)

ContaCorrente

<b>Nome</b>	<b>Conta</b>
João	1
Maria	2
José	3

Emprestimo

<b>Nome</b>	<b>Empréstimo</b>
Paulo	100
Maria	200
Carlos	300

=

Resultado da intersecção

<b>Nome</b>
Maria



# Operadores da Álgebra Relacional

- Operadores derivados
  - Junção
    - Inclui um produto cartesiano, seguido de uma seleção (pode ter projeção ao final)
    - Exemplo: nomes dos clientes com conta corrente e número de empréstimo:

$$\Pi_{\text{contacorrente.nome, emprestimo.emprestimo}}$$
$$(\sigma_{\text{contacorrente.nome} = \text{emprestimo.nome}} (\text{ContaCorrente} \times \text{Emprestimo}))$$

$$\Pi_{\text{contacorrente.nome, emprestimo.emprestimo}}$$
$$(\text{ContaCorrente} \bowtie \text{Emprestimo})$$



# Álgebra Relacional - Resumo

