

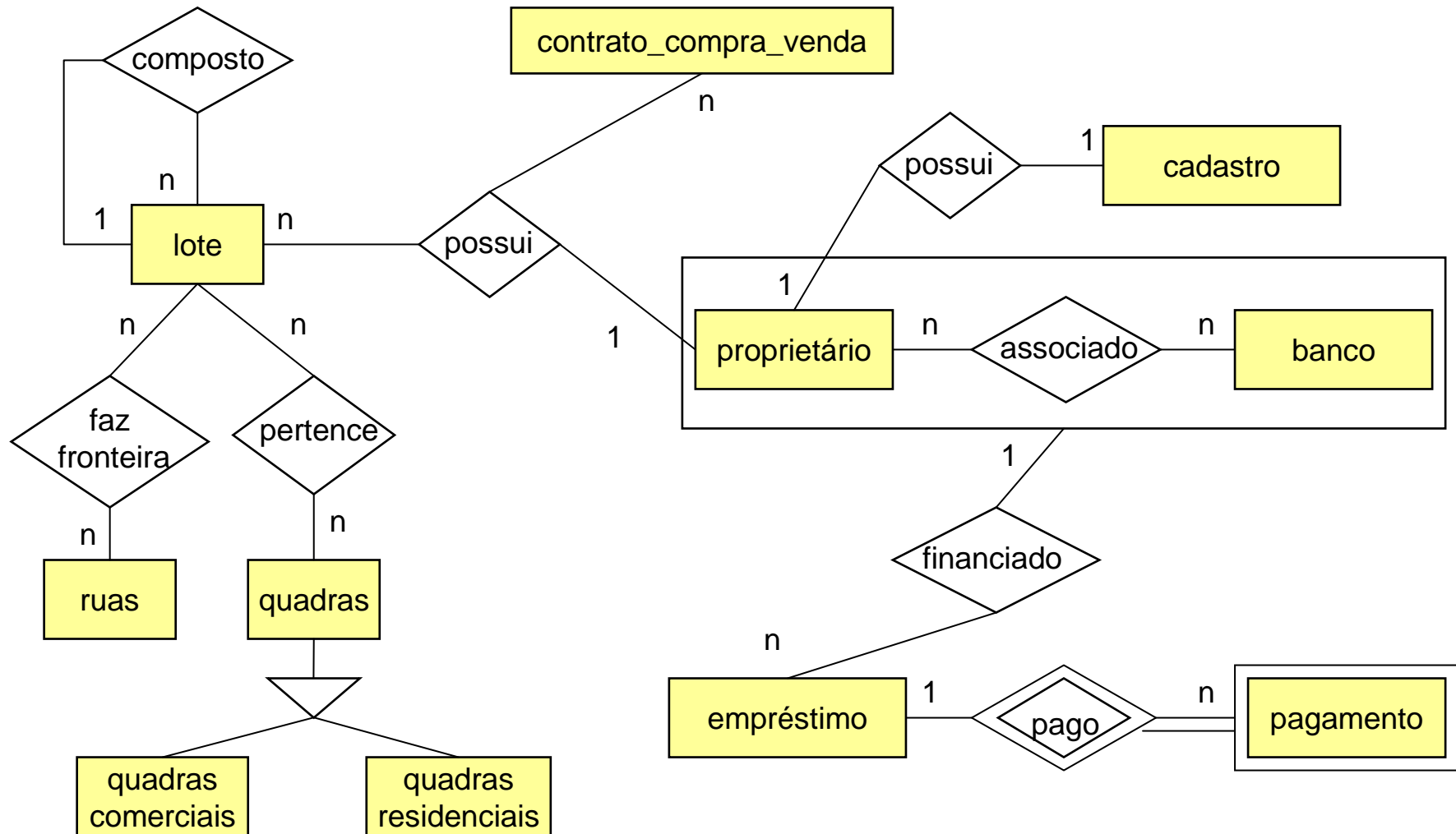


MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



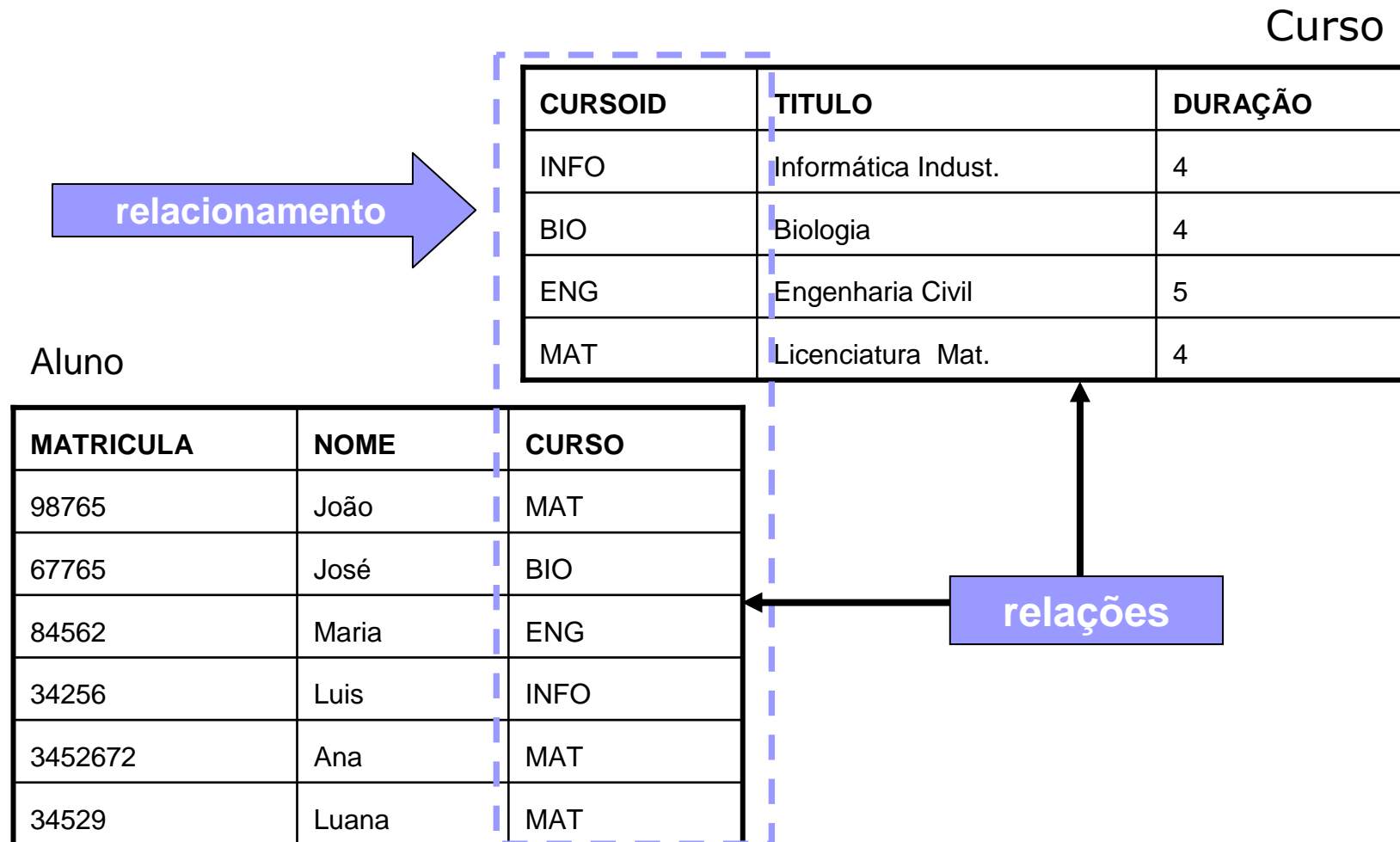


# Modelo Entidade-Relacionamento (E-R)





# Modelo Relacional



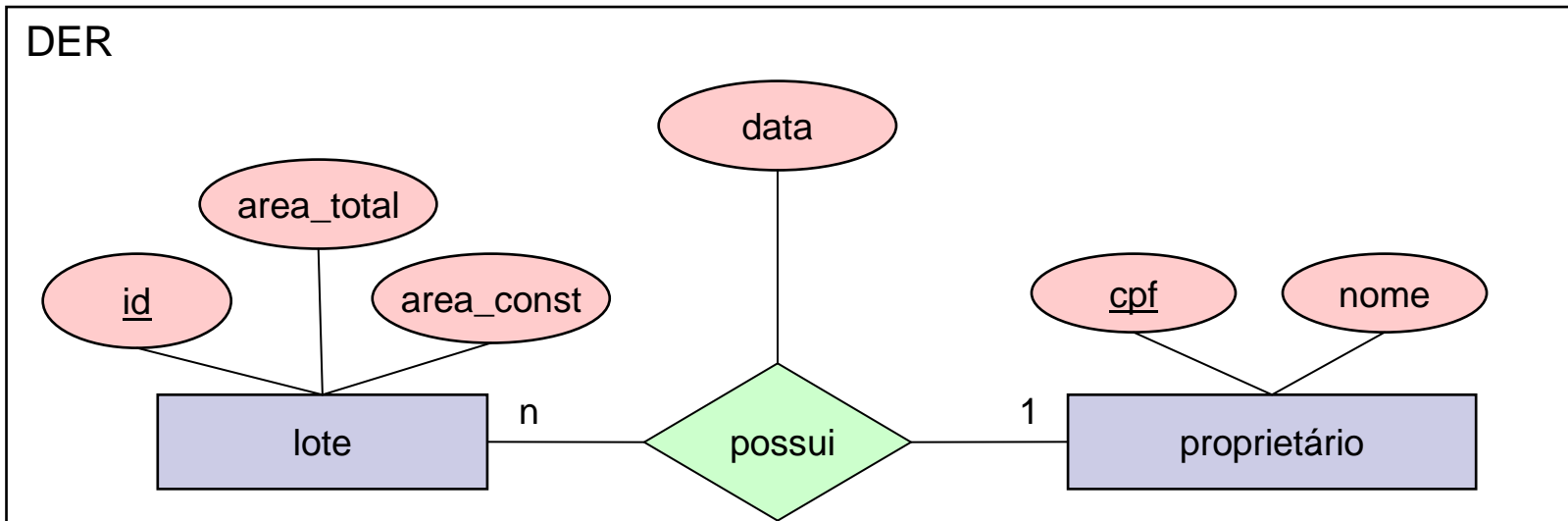


# Conversão E-R → Modelo Relacional

- Entidades com atributos chaves bem definidos geram uma relação.
- Relacionamentos podem gerar uma relação adicionando-se os atributos chaves das entidades relacionadas e os atributos do relacionamento.
- Entidades com atributos chaves não bem definidos geram uma relação adicionando-se a chave da relação que dependem.



# Conversão E-R → Modelo Relacional - Exemplo



## Relações

Lote (id, area\_total, area\_const)

Lote\_proprietario (id\_lote, cpf, data)

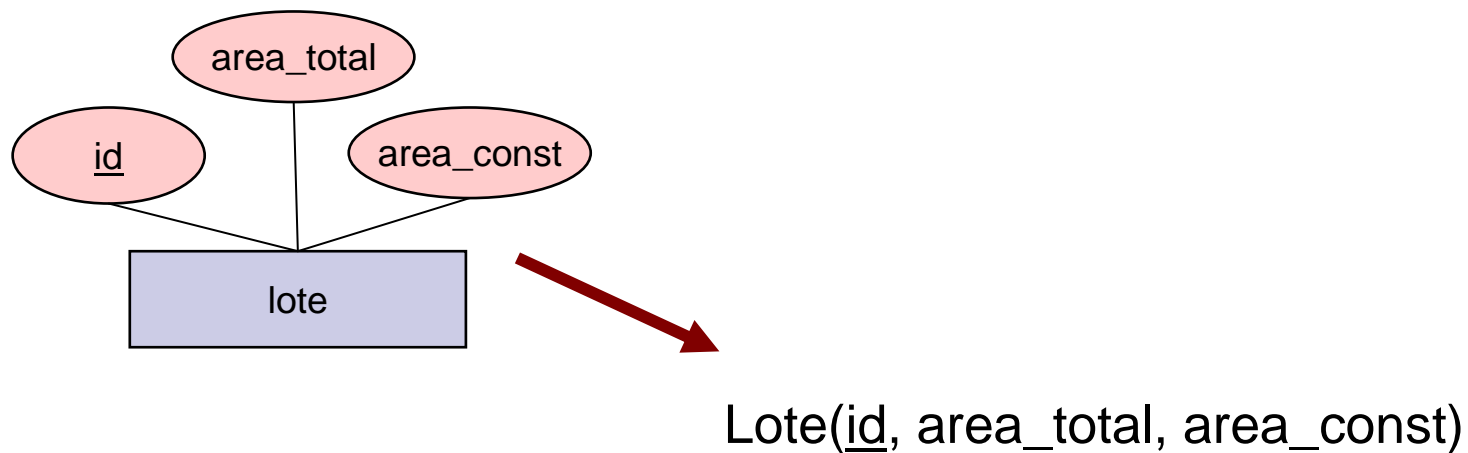
Proprietario (cpf, nome)



# Conversão E-R → Modelo Relacional

- Cada entidade é traduzida para uma tabela.
- Cada atributo (simples) da entidade define uma coluna da tabela.
- A coluna correspondente ao atributo identificador é chave primária

Ex:





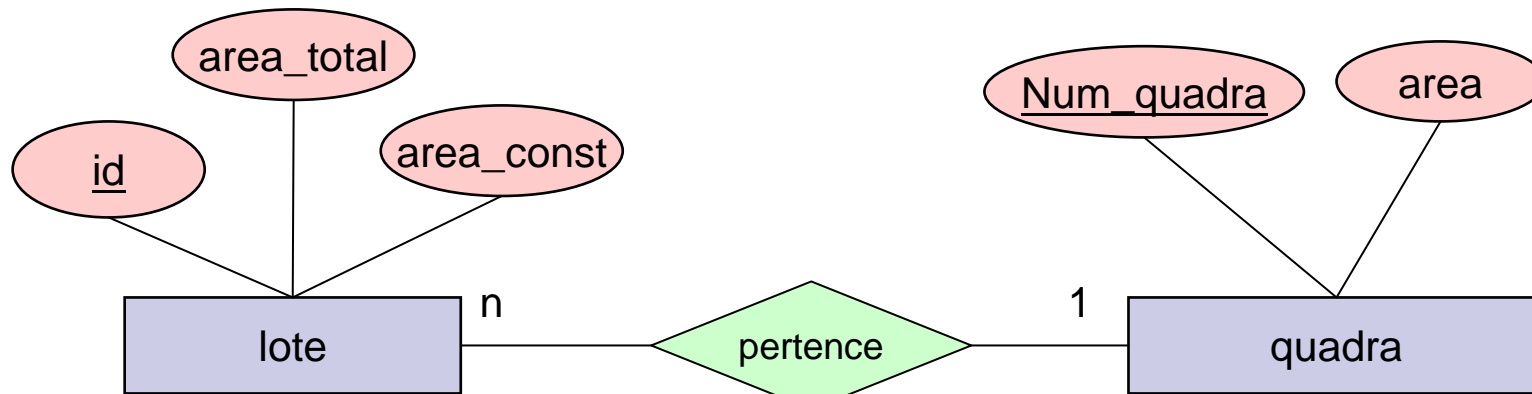
# Conversão E-R → Modelo Relacional

- Relacionamento
  - A tradução do relacionamento depende da cardinalidade das entidades que participam do relacionamento.
  
  - Formas básicas de tradução:
    - Tabela própria
    - Colunas adicionais dentro da tabela de entidade



# Conversão E-R → Modelo Relacional

- Relacionamento 1:N ou N:1



Lote(id, area\_total, area\_const, num\_quadra)

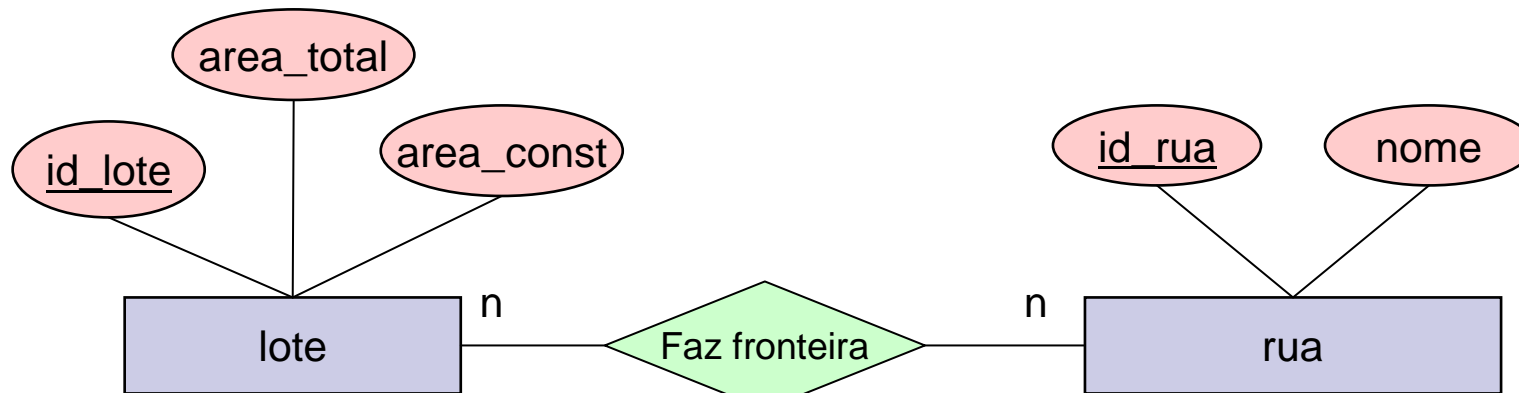
Quadra(num\_quasea, area)





# Conversão E-R → Modelo Relacional

- Relacionamento N:N



Lote(id\_lote, area\_total, area\_const)

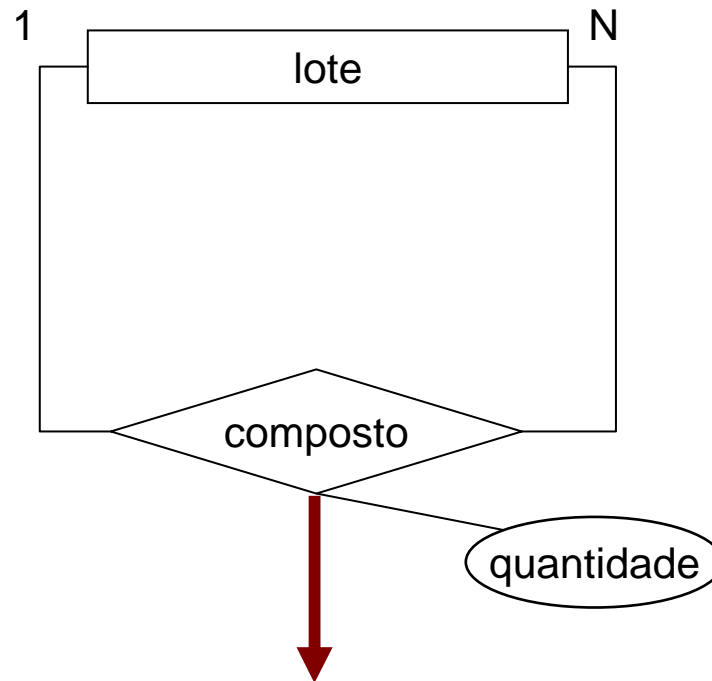
Fronteira(id\_lote, id\_rua, num\_inicial, num\_final)

Rua(id\_rua, nome)



# Conversão E-R → Modelo Relacional

- Auto relacionamento



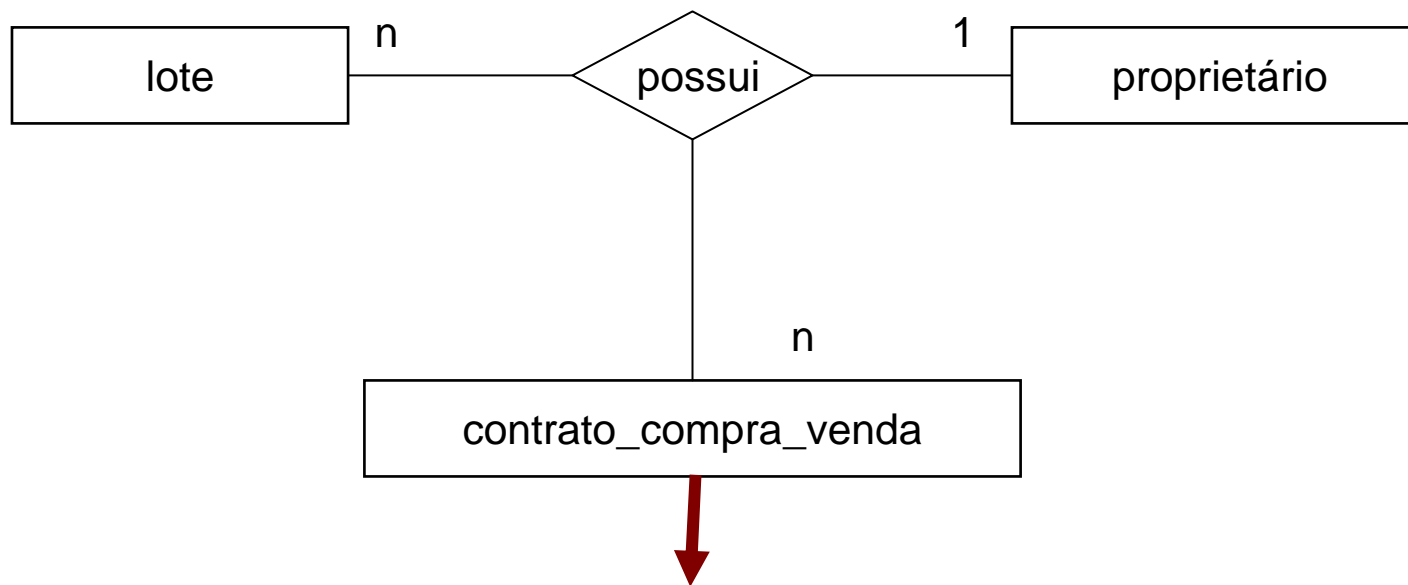
Lote(id\_lote, area\_total, area\_const)

Composição(id\_lote, id\_lote\_comp, quantidade)



# Conversão E-R → Modelo Relacional

- Relacionamento múltiplo



Lote(id\_lote, area\_total, area\_const)

Proprietario(cpf, nome)

Contrato(id\_contrato, documento)

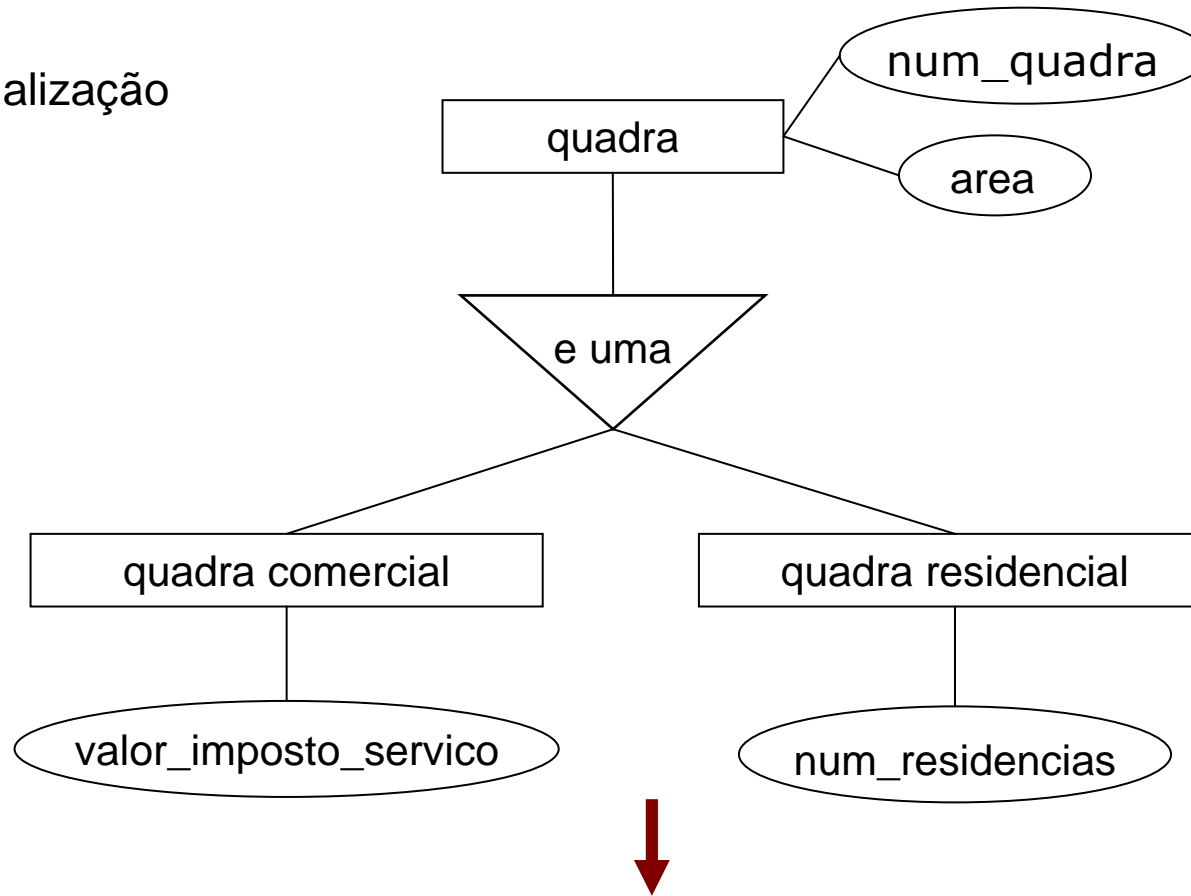
Lote\_Prop\_Contr(id\_lote, cpf, id\_contrato, data)



# Conversão E-R → Modelo Relacional

- Especialização

Solução 1



Quadra comercial (num\_quadra, area, imposto\_servico)

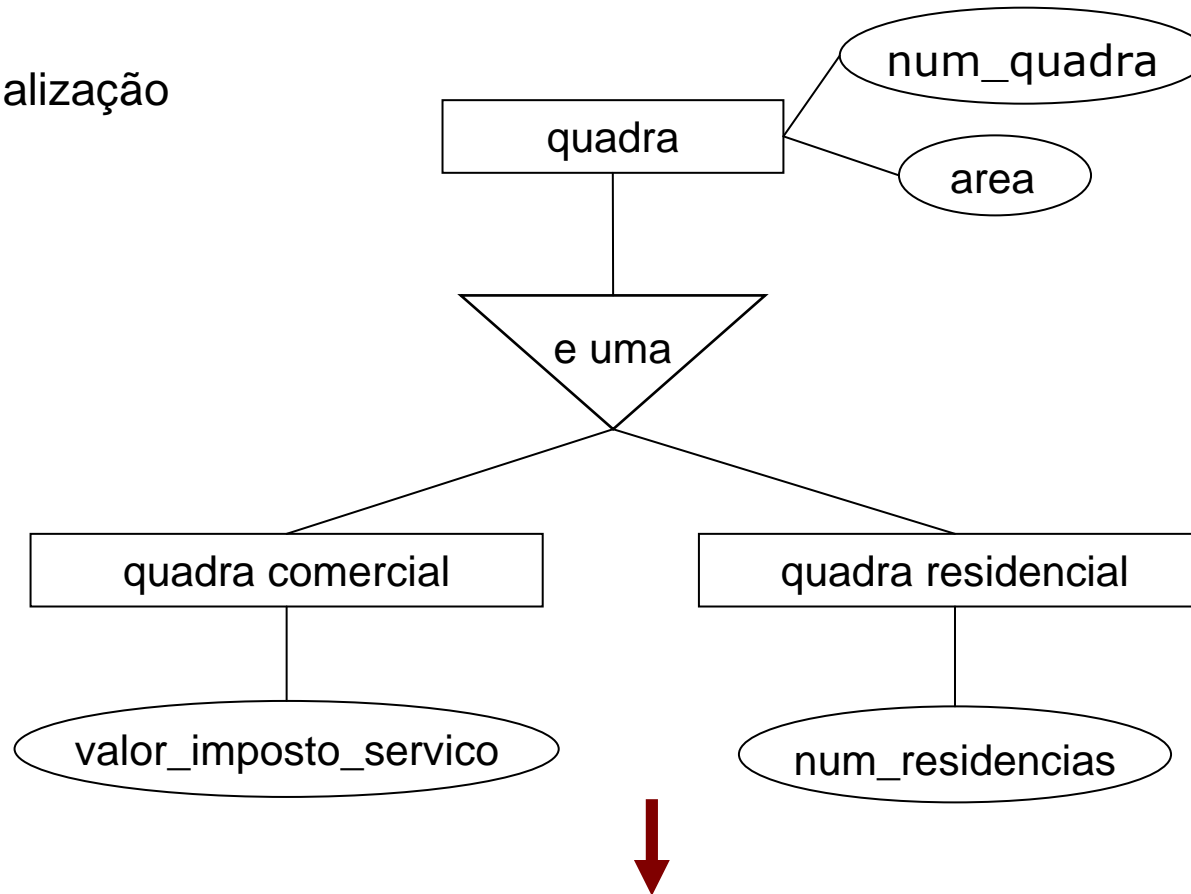
Quadra\_residencial (num\_quadra, area, num\_residencias)



# Conversão E-R → Modelo Relacional

- Especialização

Solução 2



Quadra(num\_quadra, area)

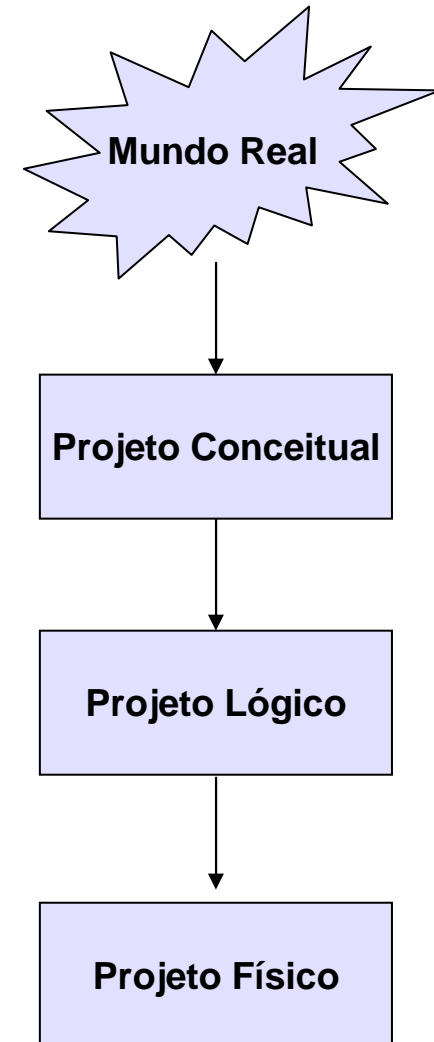
Quadra comercial (num\_quadra, imposto\_servico)

Quadra\_residencial (num\_quadra, num\_residencias)



# Fases de projeto de um Banco de Dados

- Projeto Conceitual
  - Abstração do mundo real
  - Gera um esquema conceitual de BD independente do SGBD
- Projeto Lógico
  - O esquema conceitual é mapeado para o modelo de implementação de dados do SGBD
- Projeto Físico
  - Especificação das necessidades de recursos do SGBD como estruturas de dados, métodos de acesso e segurança





# Projeto Lógico de BD

- Normalização
  - Processo pelo qual um esquema de tabelas (relações) insatisfatório é quebrado de forma que seus atributos formem relações menores que sejam mais adequadas:
    - Sem redundância de informações
    - Maior facilidade de manutenção
  - Baseado em varias regras de normalização:
    - 1ª forma normal
    - 2ª forma normal
    - 3ª forma normal



# Normalização

- Exemplo:

solicitacao\_compra

num_solic	data_solic	cod_func	nome_func	contato	cod_prod	desc_prod	quant_prod
001	12/06/09	func01	Joao Americo	39436523 39456444	2345	papel	3
001	12/06/09	func01	Joao Americo	39436523 39456444	2398	tinta	2
001	12/06/09	func01	Joao Americo	39436523 39456444	4300	impressora	1
002	6/04/03	func02	Luis Souza	39436518	2345	papel	1
002	6/04/03	func02	Luis Souza	39436518	1200	CD	1





# Problemas da Tabela Não-Normalizada

- Redundância dos dados
  - Possui vários grupos repetidos
- Anomalias de inserção
  - Inserir um novo funcionário
  - Inserir um novo produto
- Anomalias de atualização
  - Alterar o nome de um funcionário
- Anomalias de remoção
  - Remover um produto

num_solic	data_solic	cod_func	nome_func	contato	cod_prod	desc_prod	quant_prod
001	12/06/09	func01	Joao Americo	39436523 39456444	2345	papel	3
001	12/06/09	func01	Joao Americo	39436523 39456444	2398	tinta	2
001	12/06/09	func01	Joao Americo	39436523 39456444	4300	impressora	1
002	6/04/03	func02	Luis Souza	39436518	2345	papel	1
002	6/04/03	func02	Luis Souza	39436518	1200	CD	1



# Problemas da Tabela Não-Normalizada

- Redundância dos dados
  - Possui vários grupos repetidos
- Anomalias de inserção
  - Inserir um novo funcionário
  - Inserir um novo produto
- Anomalias de atualização
  - Alterar o nome de um funcionário
- Anomalias de remoção
  - Remover um produto

num_solic	data_solic	cod_func	nome_func	contato	cod_prod	desc_prod	quant_prod
001	12/06/09	func01	Joao Americo	39436523 39456444	2345	papel	3
001	12/06/09	func01	Joao Americo	39436523 39456444	2398	tinta	2
001	12/06/09	func01	Joao Americo	39436523 39456444	4300	impressora	1
002	6/04/03	func02	Luis Souza	39436518	2345	papel	1
002	6/04/03	func02	Luis Souza	39436518	1200	CD	1



# Problemas da Tabela Não-Normalizada

- Redundância dos dados
  - Possui vários grupos repetidos
- Anomalias de inserção
  - Inserir um novo funcionário
  - Inserir um novo produto
- Anomalias de atualização
  - Alterar o nome de um funcionário
- Anomalias de remoção
  - Remover um produto

↳ Gera linhas incompletas!

num_solic	data_solic	cod_func	nome_func	contato	cod_prod	desc_prod	quant_prod
001	12/06/09	func01	Joao Americo	39436523 39456444	2345	papel	3
001	12/06/09	func01	Joao Americo	39436523 39456444	2398	tinta	2
001	12/06/09	func01	Joao Americo	39436523 39456444	4300	impressora	1
002	6/04/03	func02	Luis Souza	39436518	2345	papel	1
002	6/04/03	func02	Luis Souza	39436518	1200	CD	1



# Problemas da Tabela Não-Normalizada

- Redundância dos dados
  - Possui vários grupos repetidos
- Anomalias de inserção
  - Inserir um novo funcionário
  - Inserir um novo produto
- Anomalias de atualização
  - Alterar o nome de um funcionário
- Anomalias de remoção
  - Remover um produto

Tem que alterar várias linhas!

num_solic	data_solic	cod_func	nome_func	contato	cod_prod	desc_prod	quant_prod
001	12/06/09	func01	Joao Americo	39436523 39456444	2345	papel	3
001	12/06/09	func01	Joao Americo	39436523 39456444	2398	tinta	2
001	12/06/09	func01	Joao Americo	39436523 39456444	4300	impressora	1
002	6/04/03	func02	Luis Souza	39436518	2345	papel	1
002	6/04/03	func02	Luis Souza	39436518	1200	CD	1



# Problemas da Tabela Não-Normalizada

- Redundância dos dados
  - Possui vários grupos repetidos
- Anomalias de inserção
  - Inserir um novo funcionário
  - Inserir um novo produto
- Anomalias de atualização
  - Alterar o nome de um funcionário
- Anomalias de remoção
  - Remover um produto

Remove informação sobre os funcionários

num_solic	data_solic	cod_func	nome_func	contato	cod_prod	desc_prod	quant_prod
001	12/06/09	func01	Joao Americo	39436523 39456444	2345	papel	3
001	12/06/09	func01	Joao Americo	39436523 39456444	2398	tinta	2
001	12/06/09	func01	Joao Americo	39436523 39456444	4300	impressora	1
002	6/04/03	func02	Luis Souza	39436518	2345	papel	1
002	6/04/03	func02	Luis Souza	39436518	1200	CD	1



# Normalização

- 1ª Forma normal:
  - Uma relação esta na 1FN se, e somente se, todos os domínios contiverem apenas valores atômicos.
  - Uma relação está na 1FN quando seus atributos não contém grupos de repetição
  - Uma maneira de trazer uma tabela para a 1FN é separar as entidades claramente identificadas em tabelas separadas



# Normalização

## 1ª Forma normal:

solicitacao\_compra

num_solic	data_solic	cod_func	nome_func	cod_prod	desc_prod	quant_prod
001	12/06/03	func01	Joao Silva	2345	papel	3
001	12/06/03	func01	Joao Silva	2398	tinta	2
001	12/06/03	func01	Joao Silva	4300	impressora	1

↓ 1FN

solicitacao\_compra

num_solic	data_solic	cod_func	nome_func
001	12/06/03	func01	Joao Silva

solicitacao\_produtos

num_solic	cod_prod	desc_prod	quant_prod
001	2345	papel	3
001	2398	tinta	2
001	4300	impressora	1



# Normalização

- Dependência funcional

- Dada uma relação R, o atributo Y de R é funcionalmente dependente do atributo X de R

$$(R.X \rightarrow R.Y)$$

se, e somente se, sempre que duas tuplas de R têm o mesmo valor para X elas tem também o mesmo valor para Y.

- Ex.:
  - cod\_func → nome\_func
  - cod\_prod → desc\_prod
  - num\_solic, cod\_prod → quant\_prod





# Normalização

- 2ª Forma normal:
  - Uma relação está na segunda forma normal se, e apenas se, estiver na 1FN, e cada atributo não-chave for totalmente dependente funcional da chave primária.
  - Ocorre quando a chave primária é composta por mais de um campo.
    - verificar se todos os campos que não fazem parte da chave dependem de todos os campos que compõem a chave. Se algum campo depender somente de parte da chave composta, então este campo deve pertencer a outra tabela.



# Normalização

- 2ª Forma normal:

solicitacao\_produtos

num_solic	cod_prod	desc_prod	quant_prod
001	2345	papel	3
001	2398	tinta	2
001	4300	impressora	1

↓ 2FN

produtos

cod_prod	desc_prod
2345	papel
2398	tinta
4300	impressora

solicitacao\_produtos

num_solic	cod_prod	quant_prod
001	2345	3
001	2398	2
001	4300	1



# Normalização

- 2ª Forma normal - resultado:

solicitacao\_compra

num_solic	data_solic	cod_func	nome_func
001	12/06/03	func01	Joao Silva

produtos

cod_prod	desc_prod
2345	papel
2398	tinta
4300	impressora

solicitacao\_produtos

num_solic	cod_prod	quant_prod
001	2345	3
001	2398	2
001	4300	1



# Normalização

- 3ª Forma normal:
  - Um relação está na terceira forma normal se e apenas se, estiver na 2FN, e não tiver dependências transitivas
  - *Dependência transitiva*: ocorre quando um atributo não-chave, além de depender da chave primária da tabela, depende funcionalmente de outro atributo ou combinação de atributos não-chave.
  - Em uma tabela na 3FN não existem atributos não-chave que tenham dependência de outros atributos não chave.



# Normalização

- 3ª Forma normal:

solicitacao\_compra

num_solic	data_solic	cod_func	nome_func
001	12/06/03	func01	Joao Silva

↓ 3FN

funcionarios

cod_func	nome_func
func01	Joao Silva

solicitacao\_compra

num_solic	data_solic	cod_func
001	12/06/03	func01



# Normalização

- 3ª Forma normal - resultado:

funcionarios

cod_func	nome_func
func01	Joao Silva

solicitacao\_compra

num_solic	data_solic	cod_func
001	12/06/03	func01

produtos

cod_prod	desc_prod
2345	papel
2398	tinta
4300	impressora

solicitacao\_produtos

num_solic	cod_prod	quant_prod
001	2345	3
001	2398	2
001	4300	1



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

**SQL**



# SQL

- O que é a SQL?
  - *Structured Query Language*
  - Permite o acesso e a manipulação de uma base de dados relacional
  - É um padrão ANSI (American National Standards Institute)
- O que é possível fazer com a SQL?
  - Executar consultas, recuperar dados, inserir, atualizar e remover registros, criar novos bancos, criar novas tabelas, criar *stored procedures* e *views*, definir permissões sobre tabelas, *procedures* e *views*.
- SQL é padrão mas...
  - Existem diferentes versões de SQL. Mas espera-se que a maioria dos comandos sejam suportados de maneira similar





# SQL - Structured Query Language

- Linguagem de consulta usada pela maioria de SGBD-R e SGBD-OR
- Baseada na álgebra e cálculo relacional
- É dividida em:
  - Linguagem de manipulação de dados (SQL DML)
  - Linguagem de definição de dados (SQL DDL)
  - Definição de visões (SQL DDL)
  - Especificação de autorização (SQL DDL)
  - Especificação de integridade (SQL DDL)
  - Controle de transação (SQL DDL)



# SQL - Structured Query Language

- Alguns comandos em SQL

<b>Comandos</b>	<b>Usado para</b>	<b>Tipo</b>
<i>select</i>	Consultar dados	DML
<i>insert, update, delete</i>	Incluir, alterar e remover dados	DML
<i>commit, rollback</i>	Controlar transações	DDL
<i>create, alter, drop</i>	Definir, alterar e remover esquemas (tabelas)	DDL



# SQL - Structured Query Language

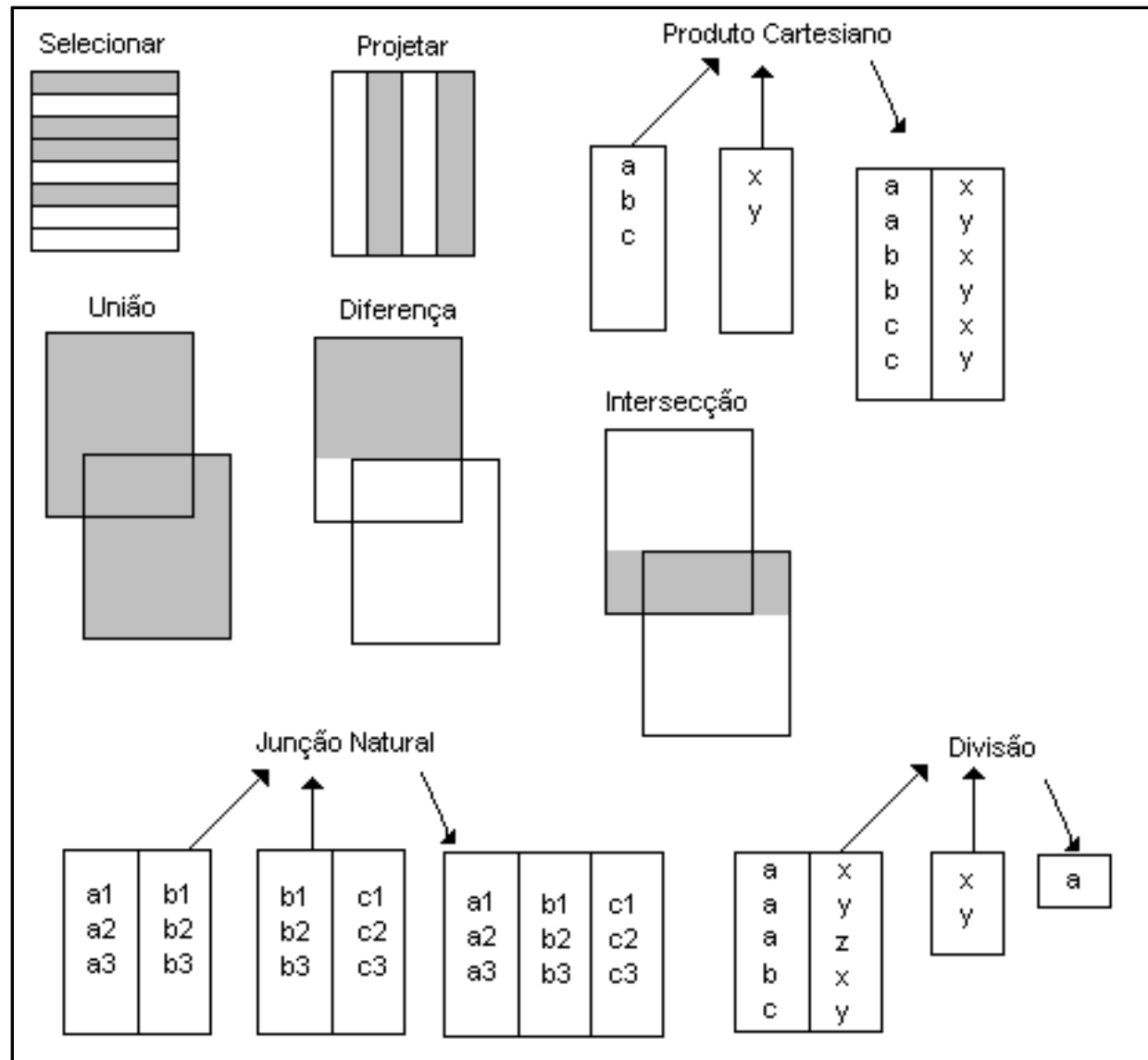
```
CREATE TABLE cliente
(nome                CHAR(20) NOT NULL,
 endereço           CHAR(30),
 cidade             CHAR(30),
 PRIMARY KEY        (nome))
```

```
ALTER TABLE cliente ADD RG CHAR(10)
```

```
SELECT nome, endereco
FROM cliente
WHERE cidade = 'São José dos Campos'
```



# Álgebra Relacional - Resumo





# SQL - Básico

- Seleção

```
SELECT *  
FROM solicitacao_compra  
WHERE cod_func = 'func01'
```

- Projeção

```
SELECT cod_func  
FROM solicitacao_compra
```

- Produto Cartesiano

```
SELECT funcionarios.*, solicitacao_compra.*  
FROM funcionarios INNER JOIN solicitacao_compra  
ON funcionarios.cod_func = solicitacao_compra.cod_func
```