



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Introdução a Sistemas de Banco de Dados

Karine Reis Ferreira

karine@dpi.inpe.br

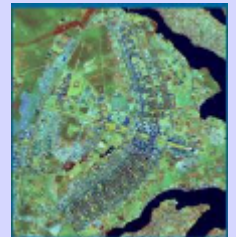
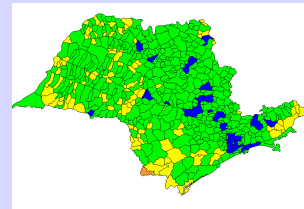
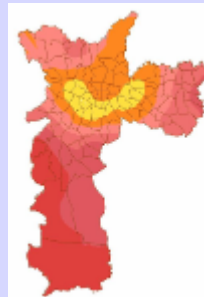
Sistemas de Banco de Dados

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

Sistemas de Banco de Dados

Bancos de Dados:

é uma coleção de dados relacionados de um determinado domínio.



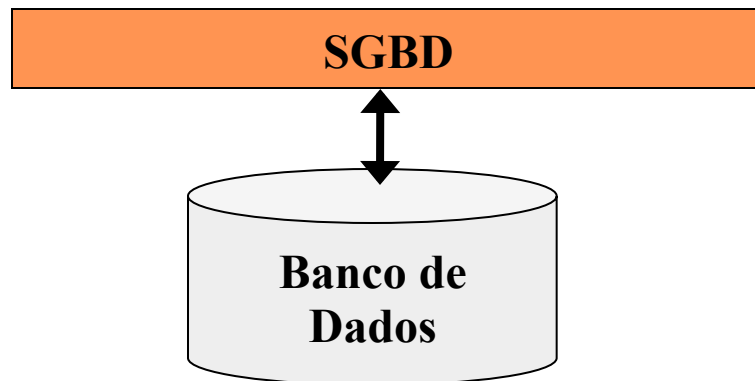
Sistemas de Banco de Dados

SGBD – Sistemas Gerenciadores de Bancos de Dados ou
DBMS – Database Management Systems

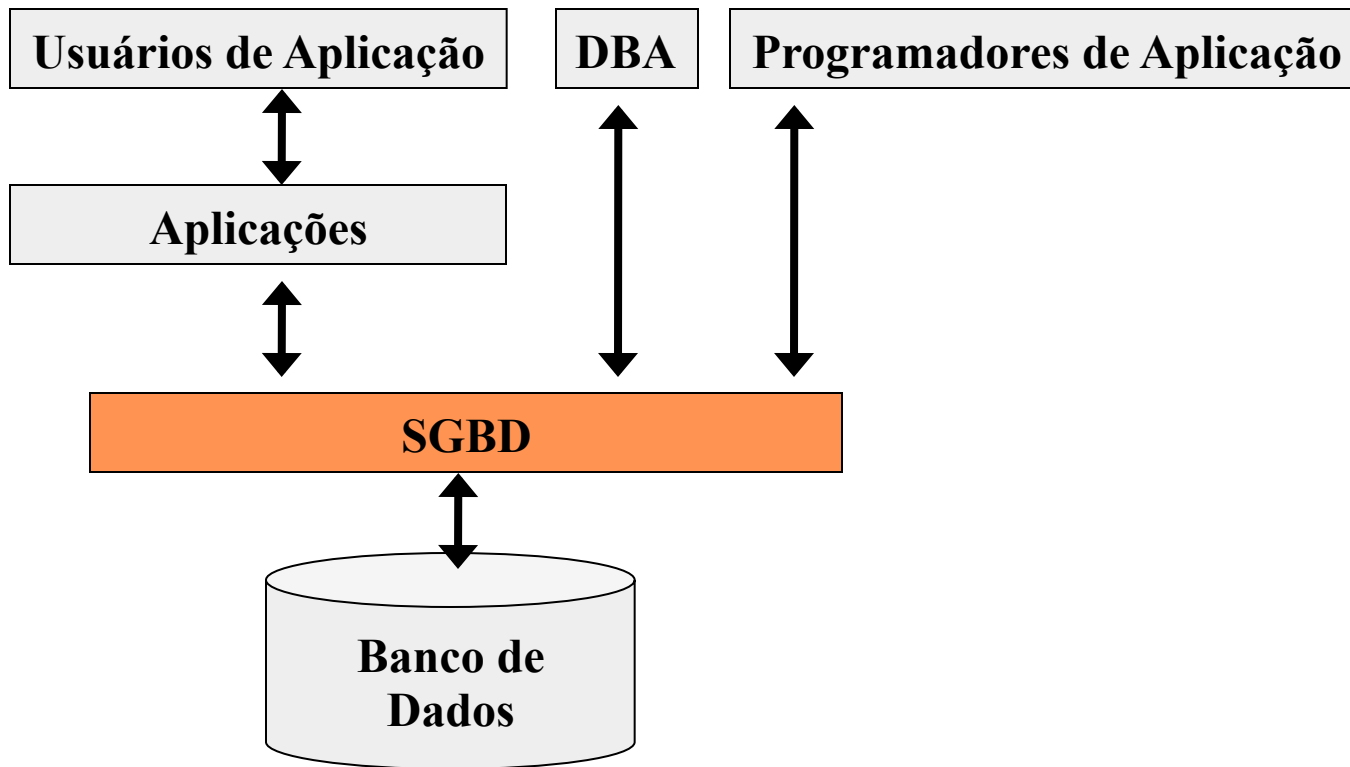
Sistema Gerenciador de Bancos de Dados (SGBD ou DBMS):

é uma coleção de programas para criar, gerenciar e manipular um banco de dados.

Exemplos: MySQL, Oracle, PostgreSQL, ...

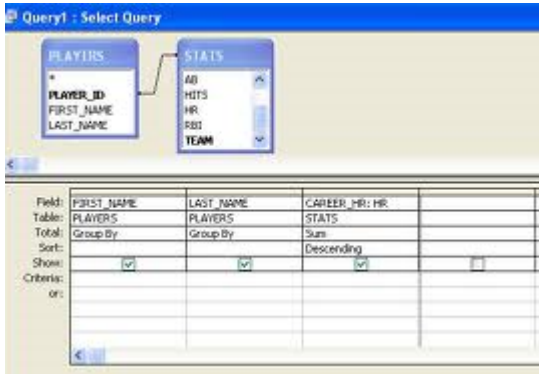


Sistemas de Banco de Dados



Sistemas de Banco de Dados

GUI

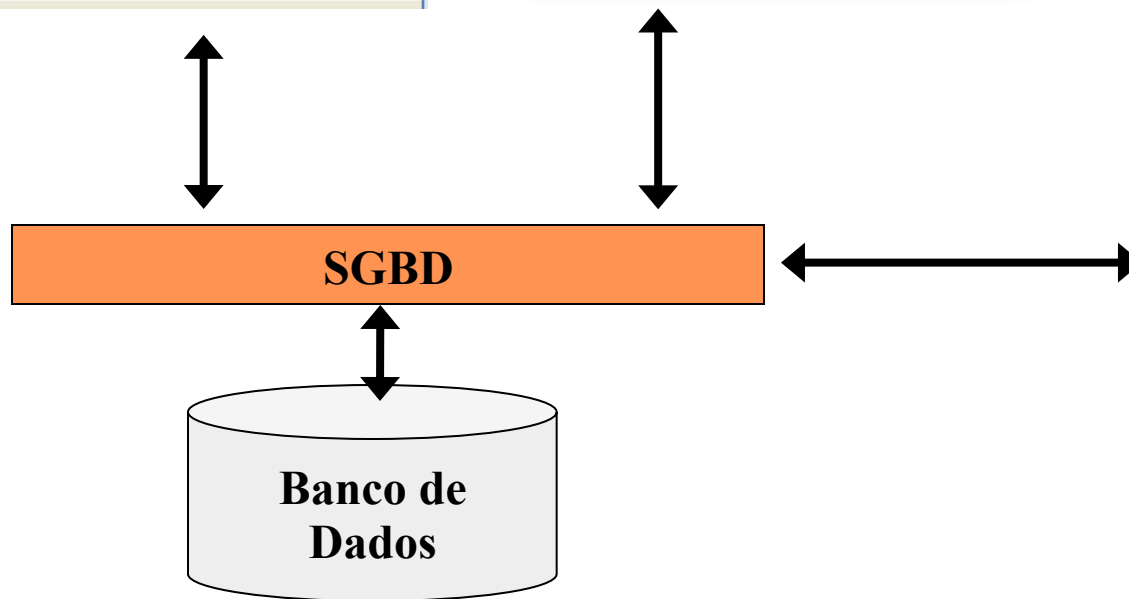


Prompt

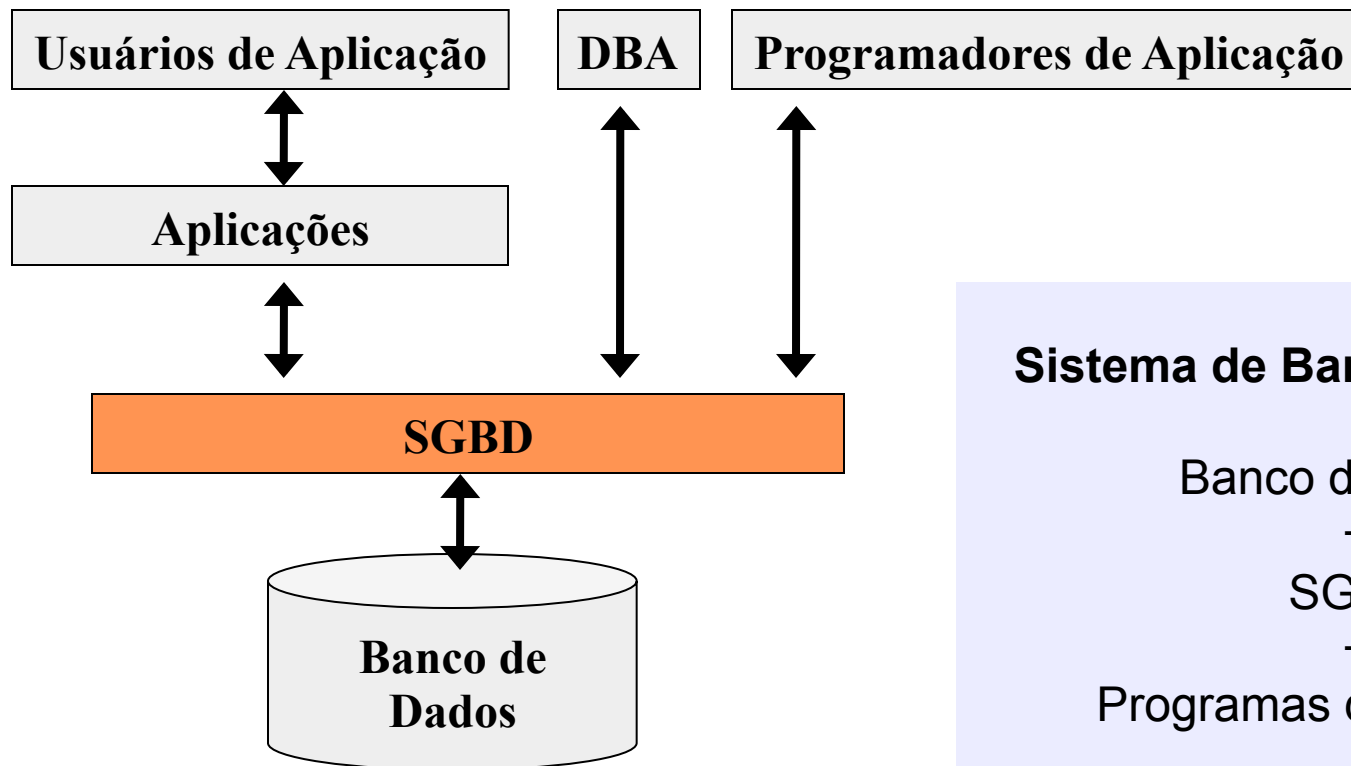


API

```
int main()
{
  ...
}
```



Sistemas de Banco de Dados



Sistema de Bancos de Dados:

Banco de Dados

+

SGBD

+

Programas de Aplicação

SGBD

- Armazenar, consultar e atualizar o banco de dados de maneira eficiente
 - Linguagem de definição de dados (SQL-DDL)
 - Linguagem de consulta e manipulação de dados (SQL-DML)
- Manter a consistência e integridade dos dados
 - Evitar redundância
- Controlar acessos concorrentes (multiusuários)
- Manter a segurança dos dados
 - Recuperar falhas e fazer cópias de reserva (backup)
 - Restringir e controlar os acessos dos usuários
 - Restrições de segurança

Alguns SGBDs bem conhecidos

IBM

DB2.

Universal
Database

ORACLE®

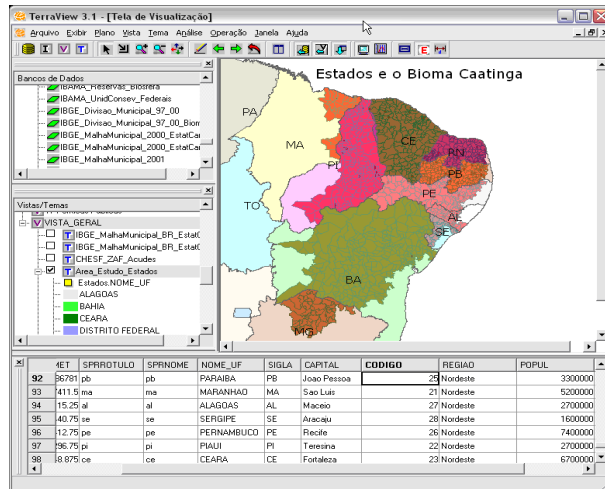


PostgreSQL



Exemplo de aplicação

TerraView



SGBD



Sistemas de Informações Geográficas (SIG): Sistema computacional capaz de armazenar, processar e manipular dados geográficos (Worboys and Duckham, 2004)

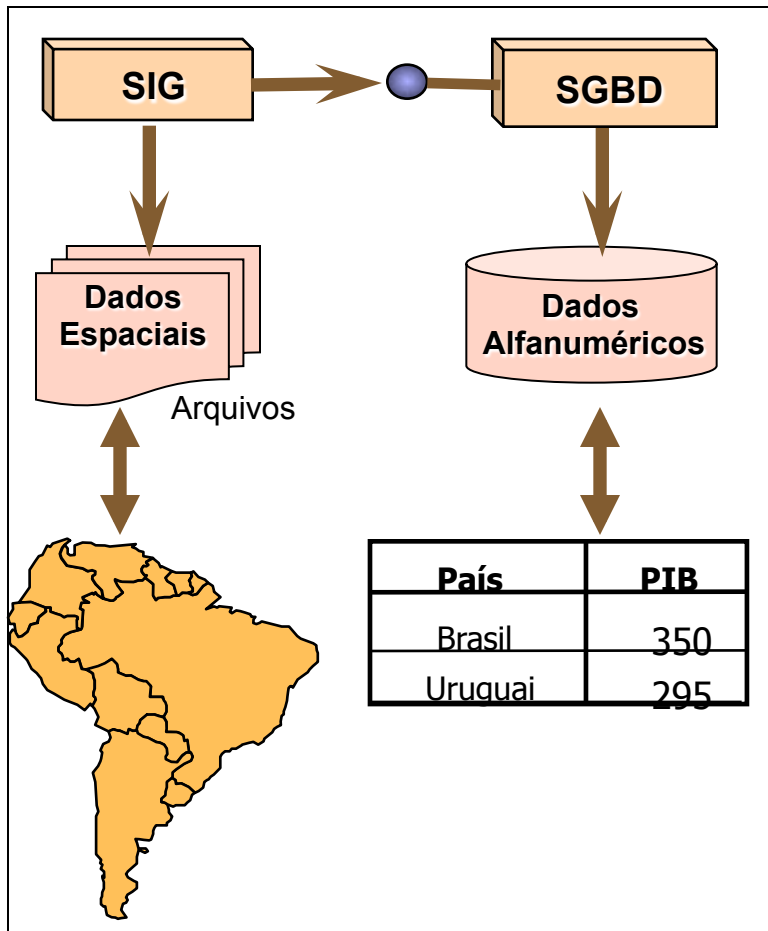
Sistemas de Informações Geográficas (SIG)

- *Geographic Information System (GIS)*
- Funcionalidades de um SIG (Rigaux et al, 2002):
 - Entrada e validação de dados espaciais;
 - Armazenamento e gerenciamento;
 - Saída e apresentação visual;
 - Transformação de dados espaciais;
 - Interação com o usuário;
 - Combinação de dados espaciais - novas representações do espaço
 - Ferramentas para análise espacial.

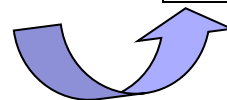
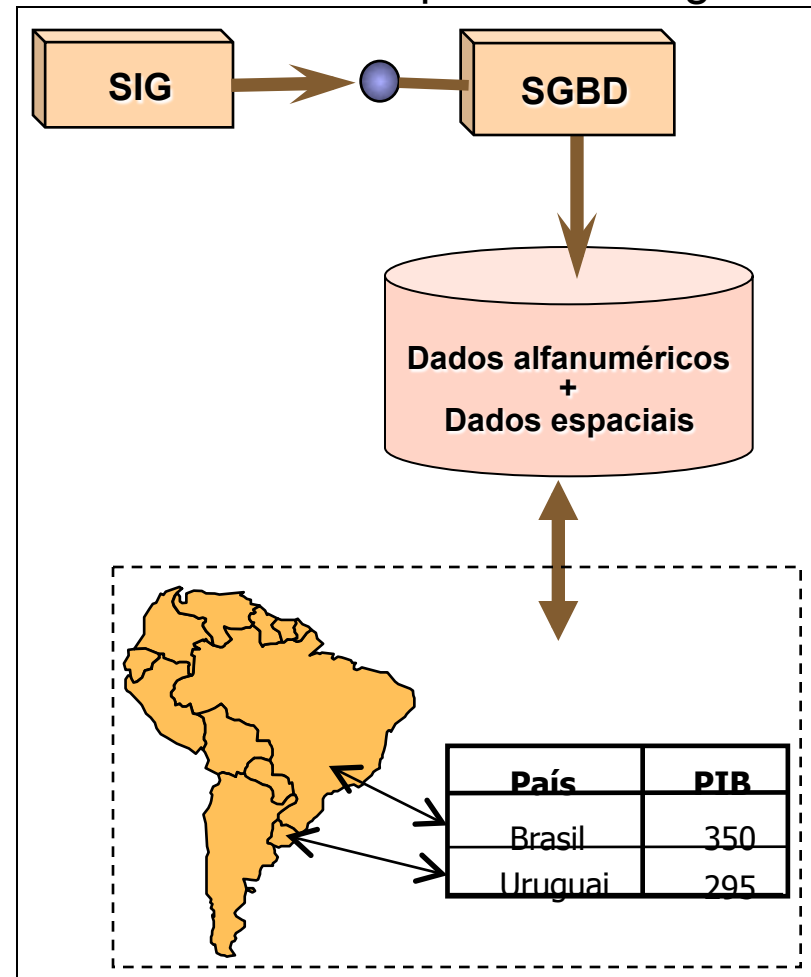
Evolução dos SIGs

Armazenamento de dados geográficos

Arquitetura Dual



Arquitetura Integrada





MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Modelo Conceitual de Entidade-Relacionamento (ER)

Projeto de Bancos de Dados

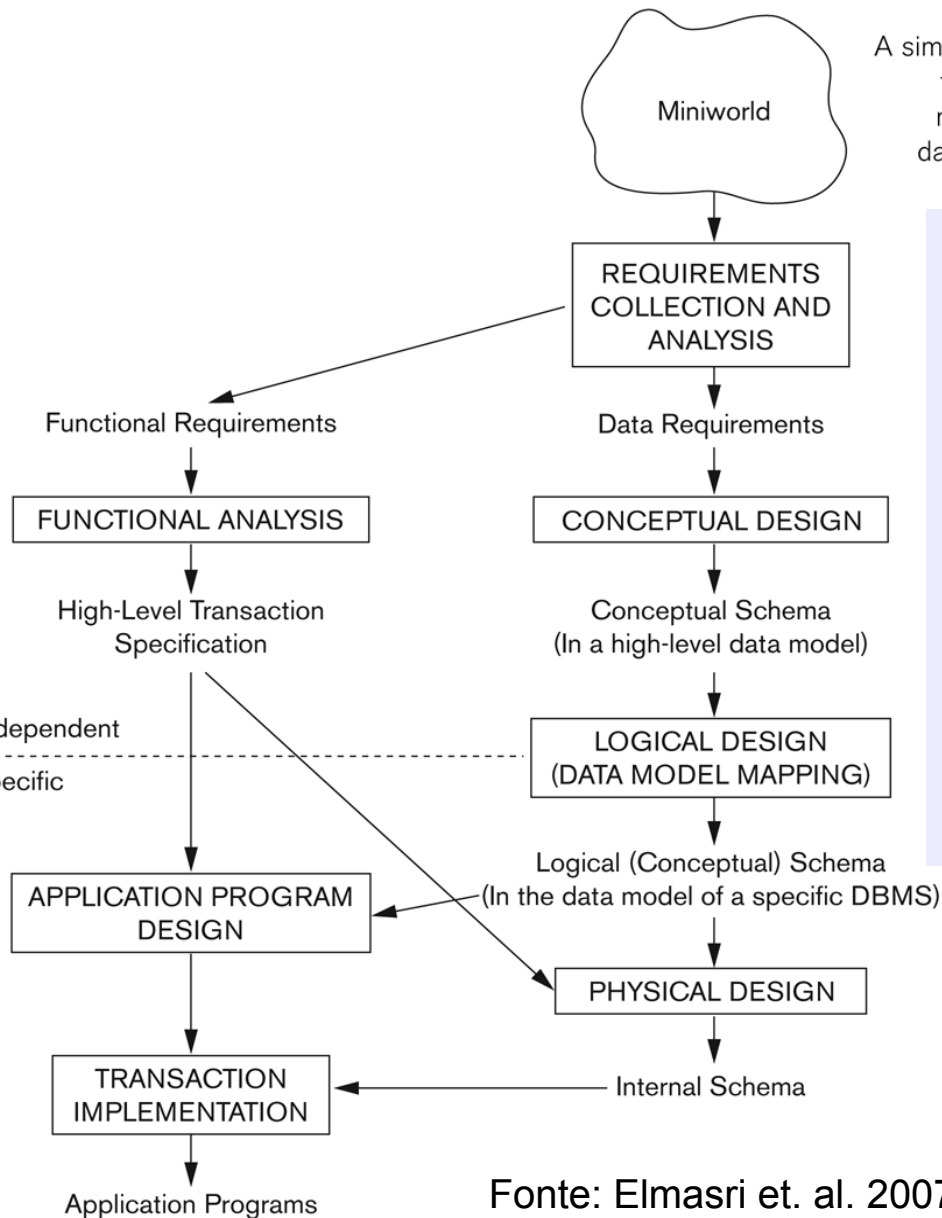


Figure 3.1
A simplified diagram to illustrate the main phases of database design.

Projeto Conceitual

- Abstração do mundo real
- Descrição detalhada das entidades, relacionamentos e restrições.
- Esquema conceitual de BD independente do SGBD
- Modelo Entidade-Relacionamento (ER)

Projeto de Bancos de Dados

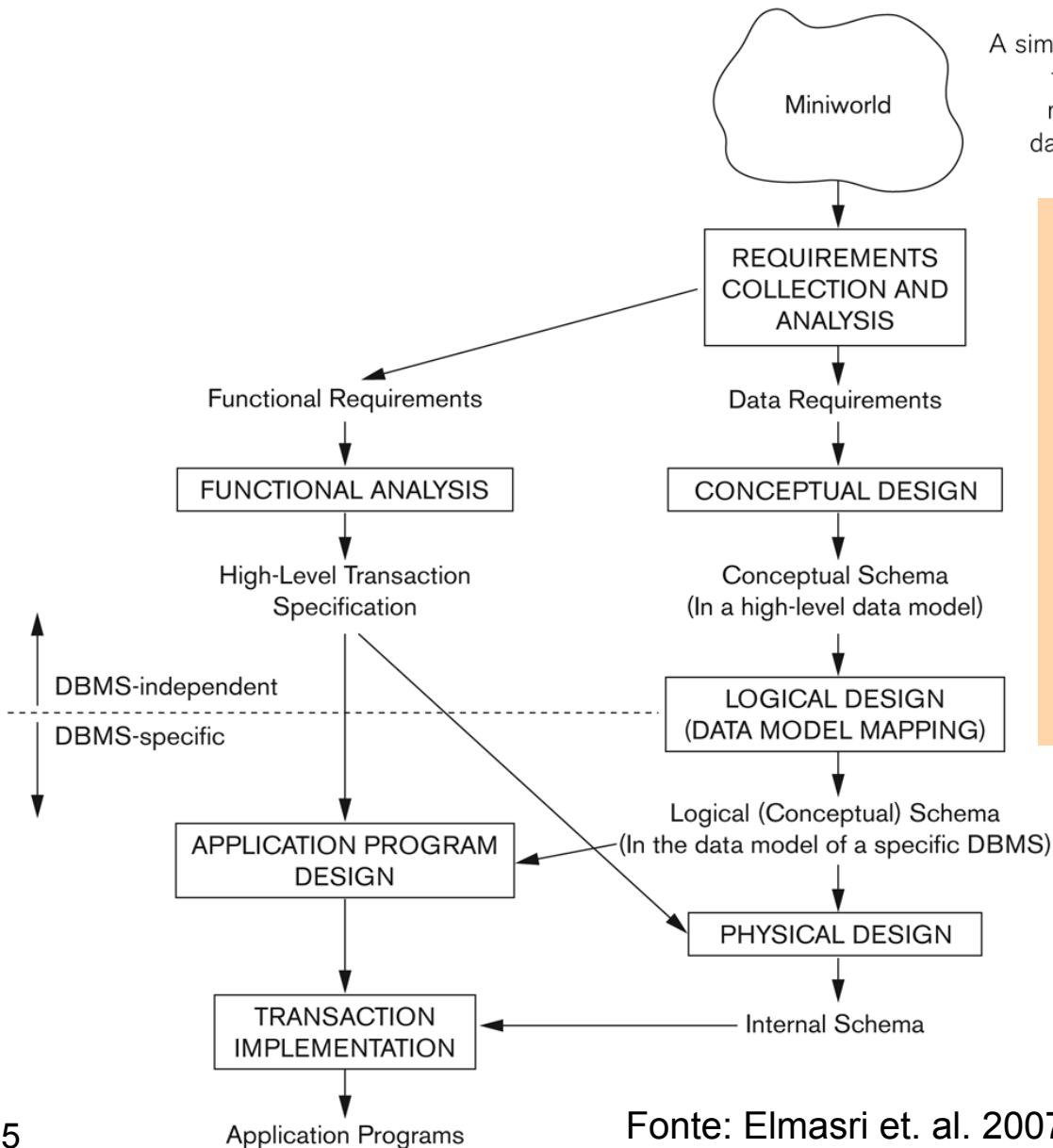


Figure 3.1

A simplified diagram to illustrate the main phases of database design.

Projeto Lógico

- Mapeamento do modelo conceitual para o modelo do SGBD (relacional)
- Dependente de SGBD
- Modelo Relacional

Projeto de Bancos de Dados

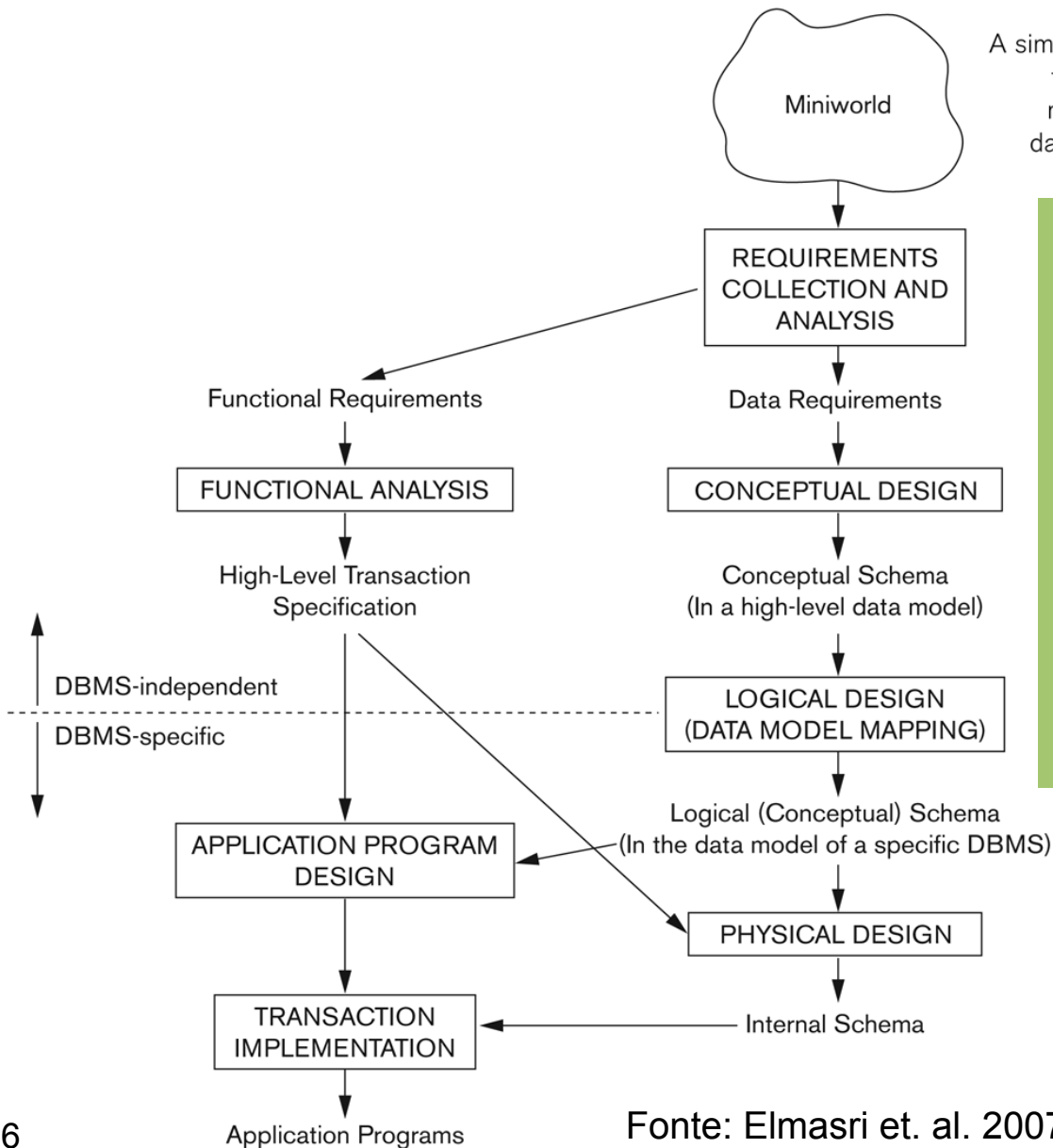


Figure 3.1

A simplified diagram to illustrate the main phases of database design.

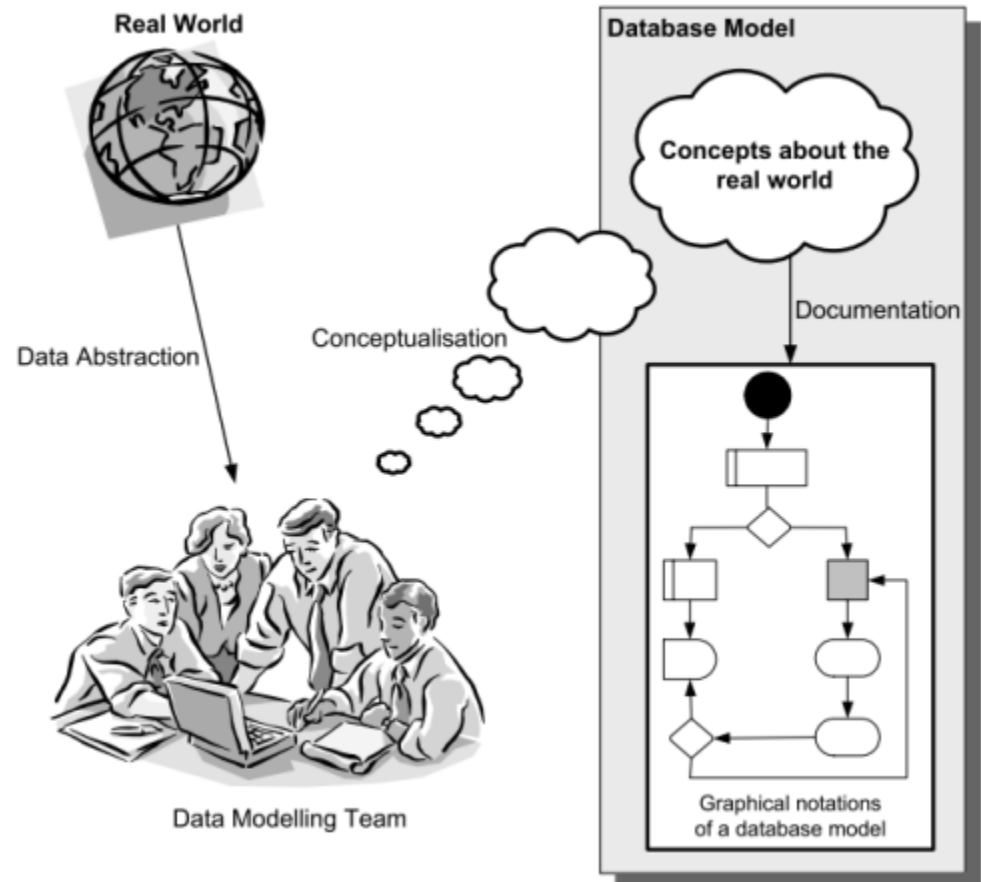
Projeto Físico

- Definição dos métodos de acesso
- Particionamento físico de tabelas
- Controle do Buffer de memória, tamanho de página no disco...

Modelo de dados

Conjunto de conceitos usados para representar os dados, seus relacionamentos e as regras que garantem a sua consistência.

No processo de modelagem é necessário construir uma abstração dos objetos e fenômenos do mundo real.



Modelo Entidade-Relacionamento (E-R)

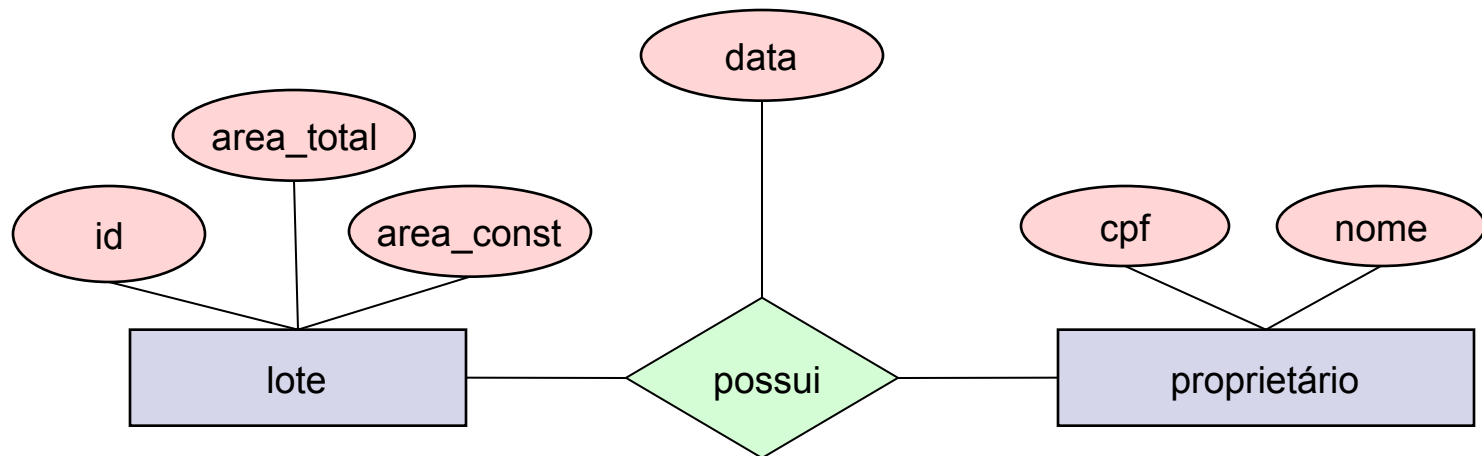
- Introduzido em 1976 por Peter Chen, é a abordagem mais adotada para modelagem conceitual de dado. Tem por objetivo facilitar o projeto do banco de dados representando sua estrutura lógica
- Definição: modelo baseado na percepção do mundo real como um conjunto de objetos chamados entidades e pelo conjunto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

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

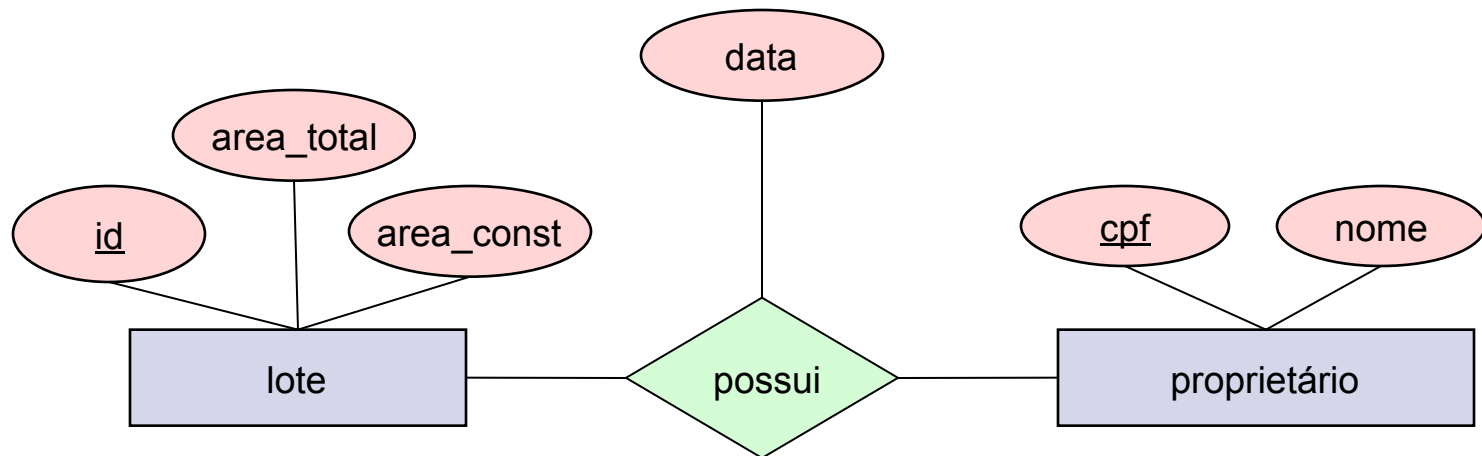
Modelo Entidade-Relacionamento (E-R)

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



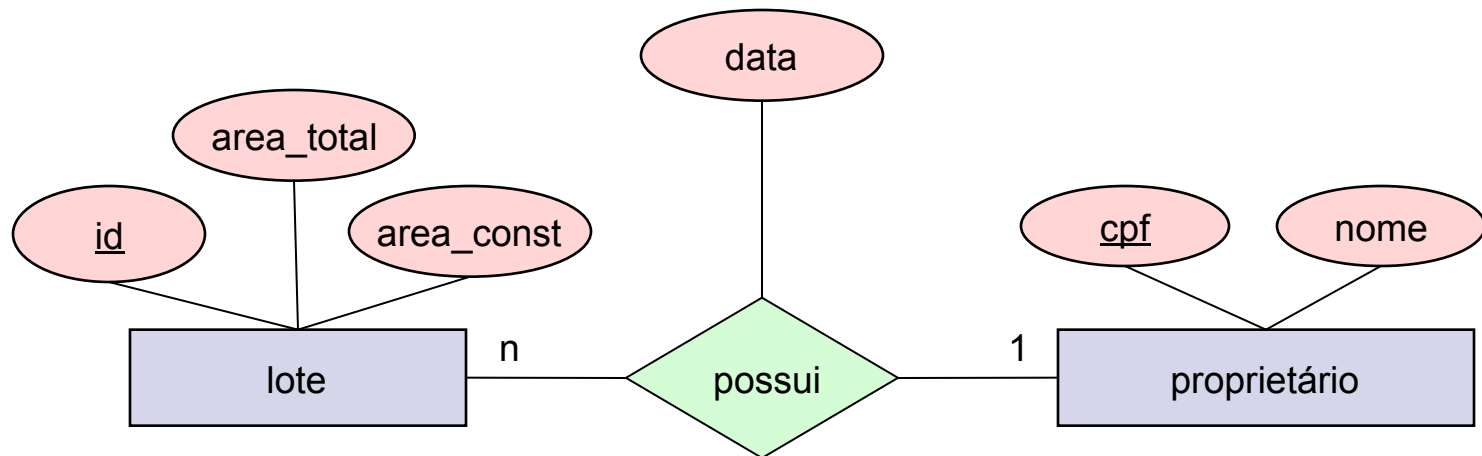
Modelo Entidade-Relacionamento (E-R)

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



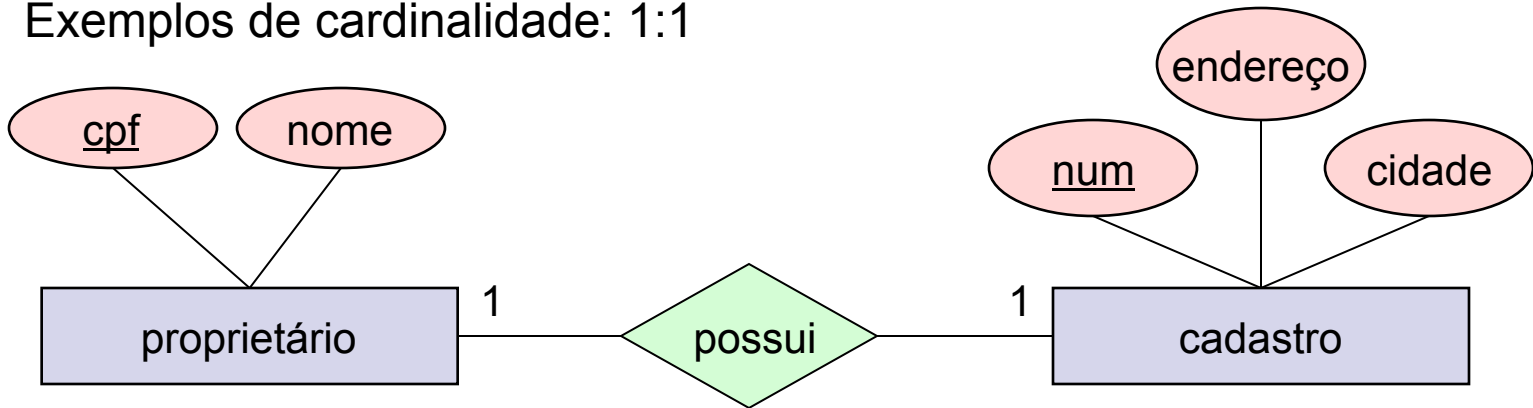
Modelo Entidade-Relacionamento (E-R)

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

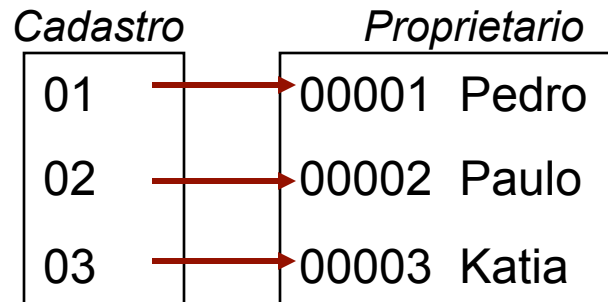


Modelo Entidade-Relacionamento (E-R)

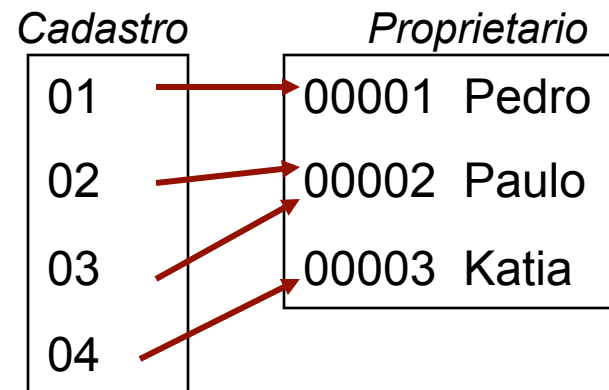
- Exemplos de cardinalidade: 1:1



Relação correta

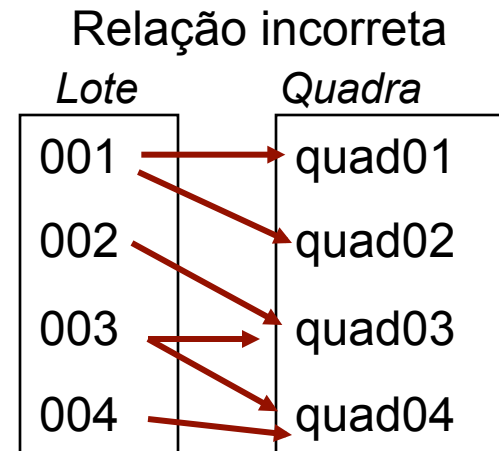
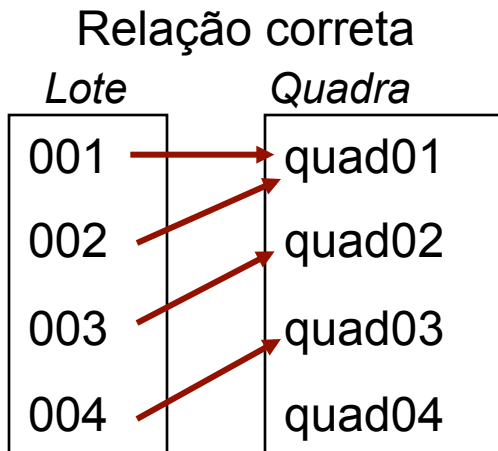
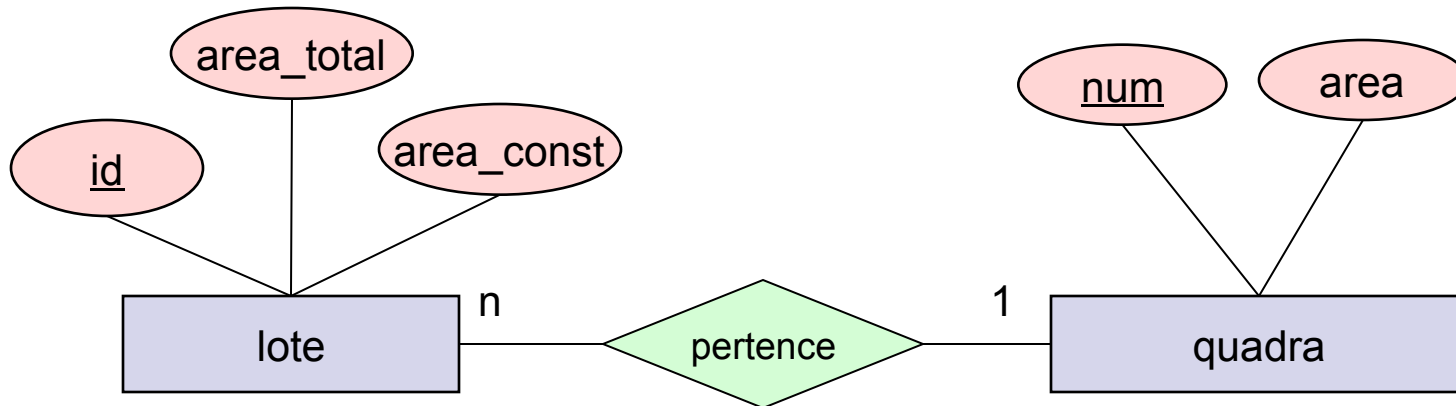


Relação Incorreta



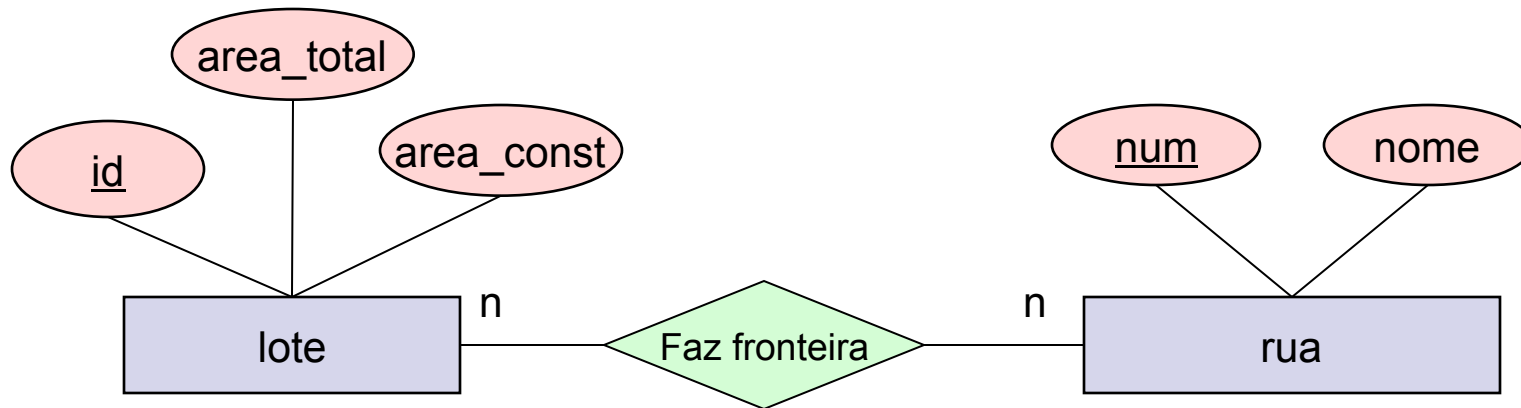
Modelo Entidade-Relacionamento (E-R)

- Exemplos de cardinalidade: 1:N

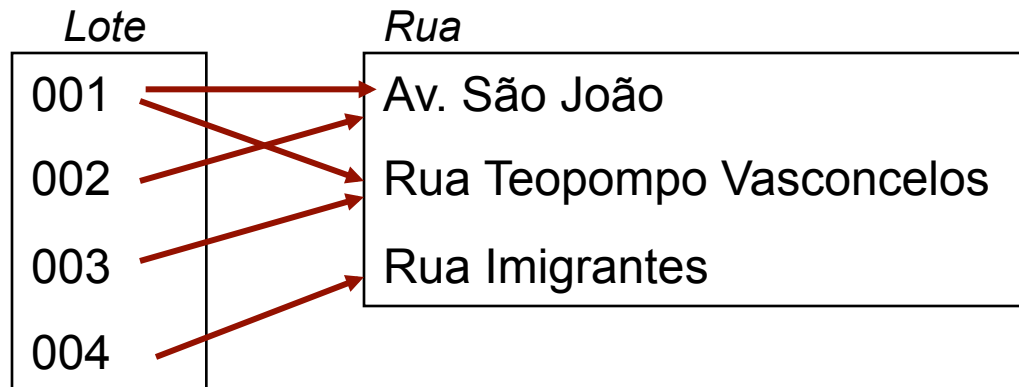


Modelo Entidade-Relacionamento (E-R)

- Exemplos de cardinalidade: N:N

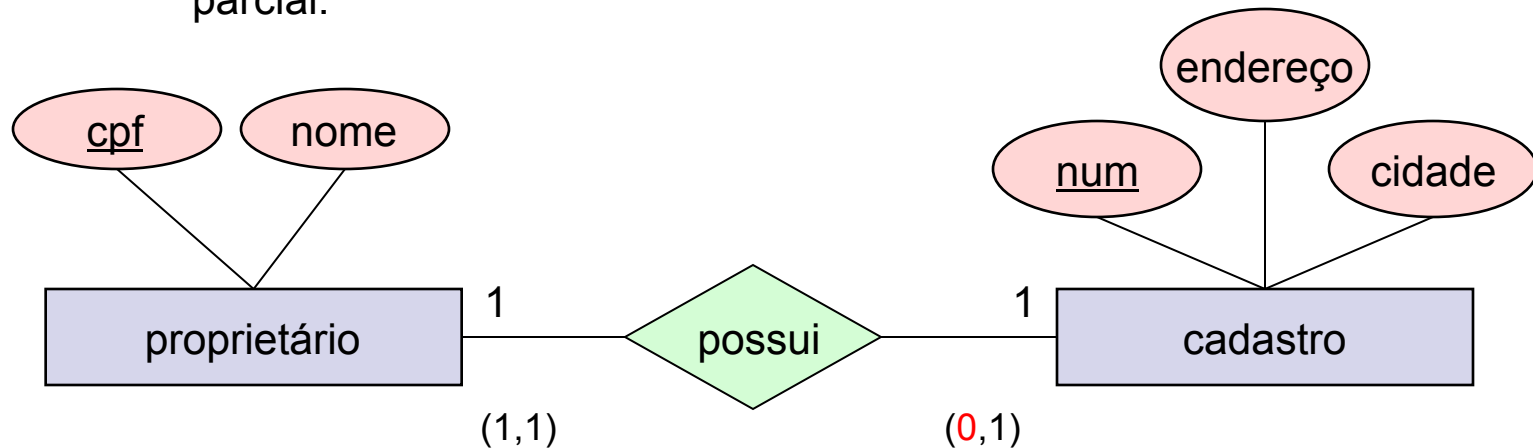


Relação correta



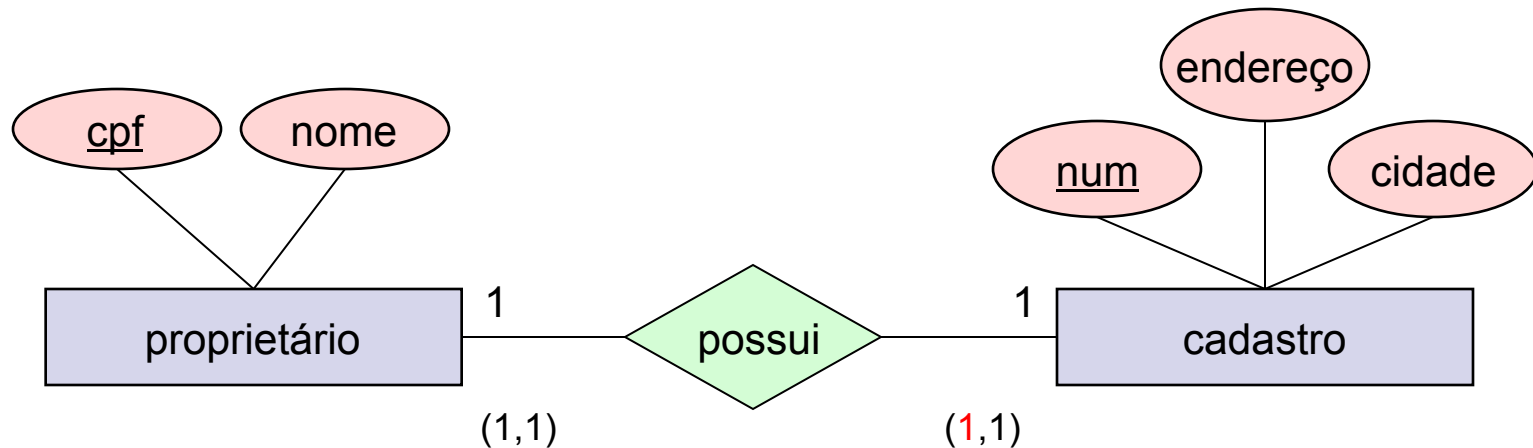
Modelo Entidade-Relacionamento (E-R)

- 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.



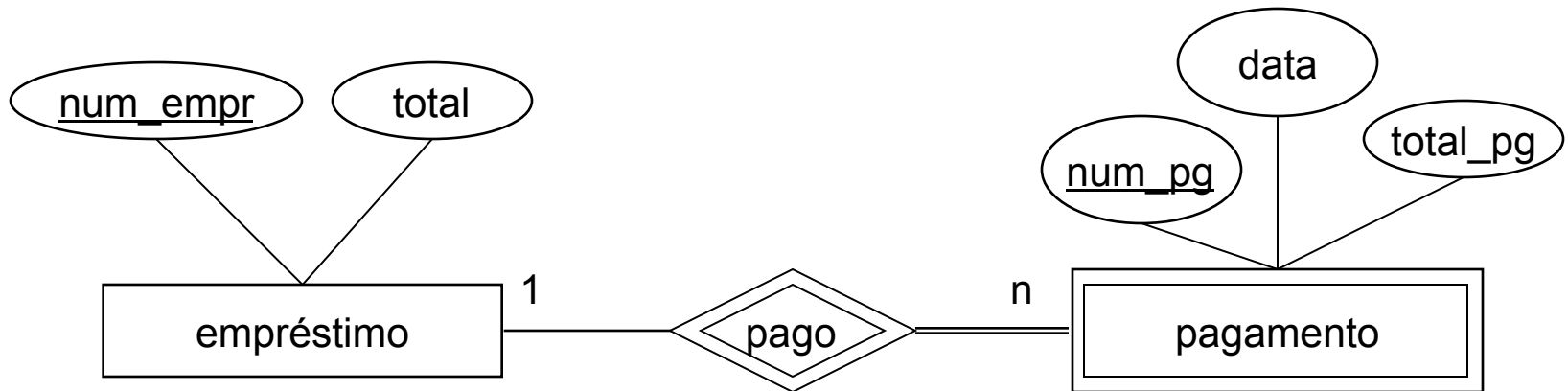
Modelo Entidade-Relacionamento (E-R)

- 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.



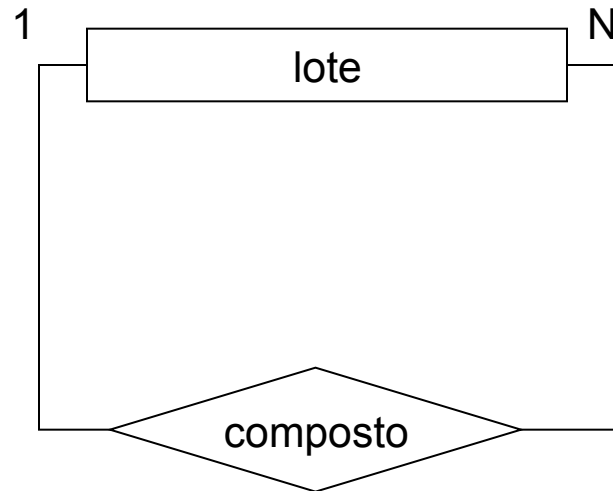
Modelo Entidade-Relacionamento (E-R)

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



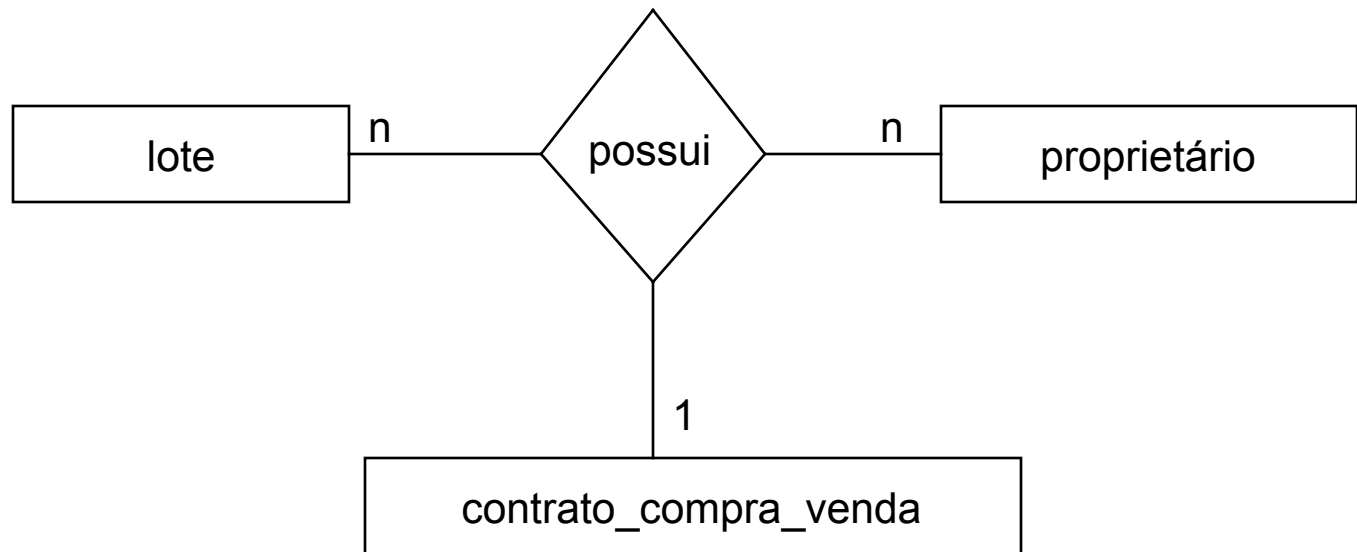
Modelo Entidade-Relacionamento (E-R)

- 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.

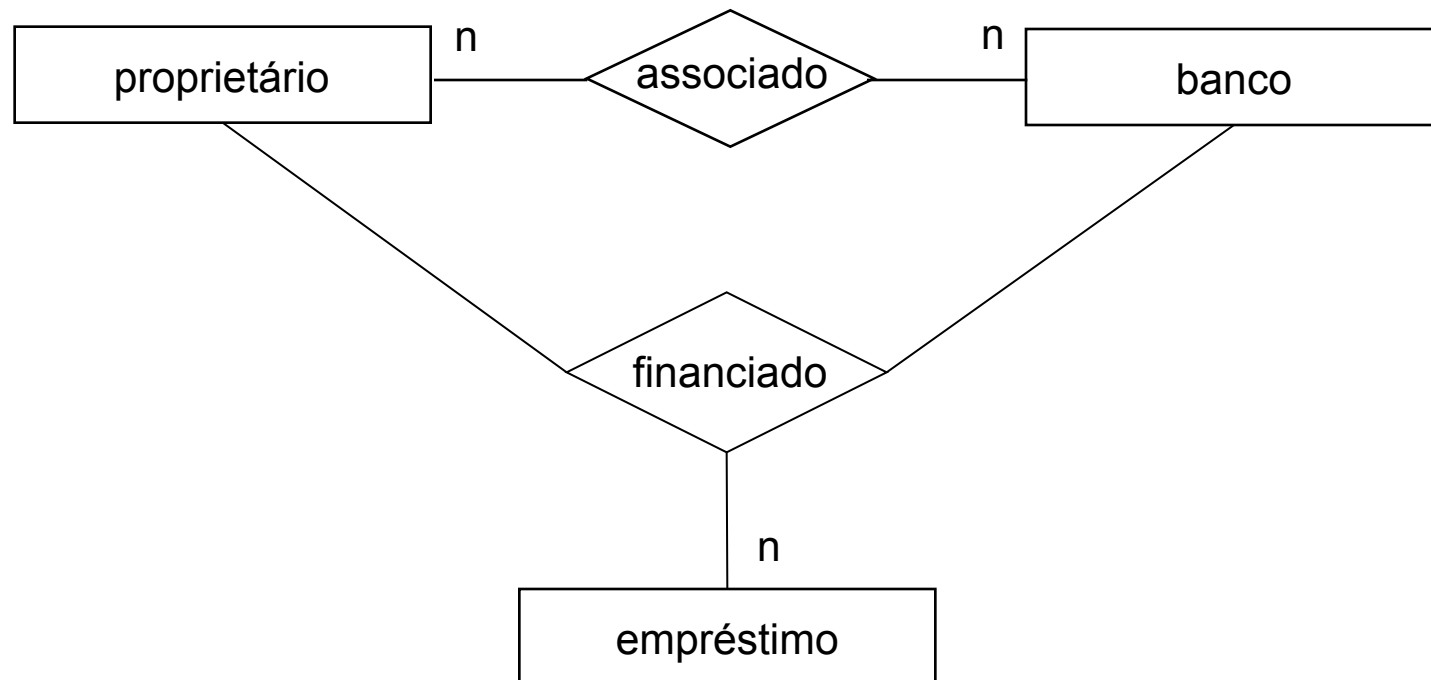


Modelo Entidade-Relacionamento (E-R)

- 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.

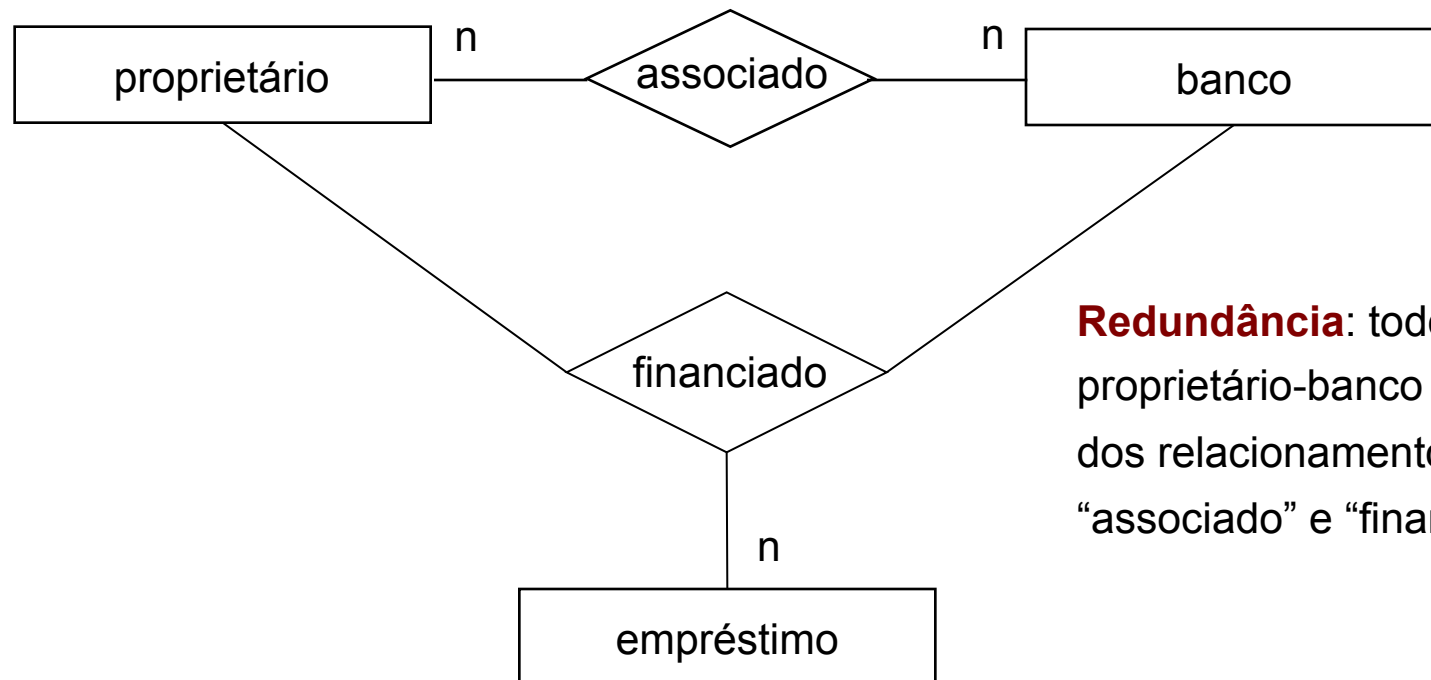
Modelo Entidade-Relacionamento (E-R)

- 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)

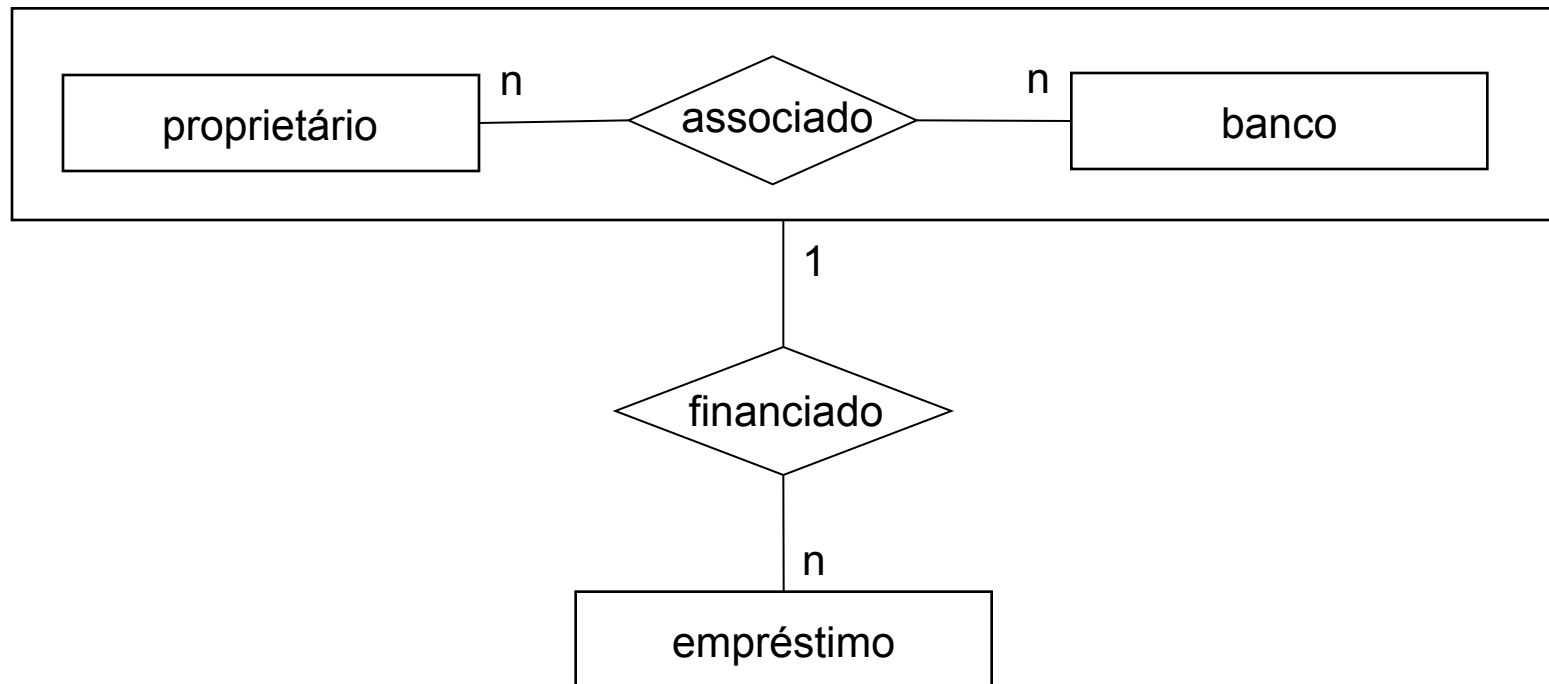
- Ex.: um proprietário pode estar associado a vários bancos. Pra cada banco associado, o proprietário pode ter um ou mais empréstimos.



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.

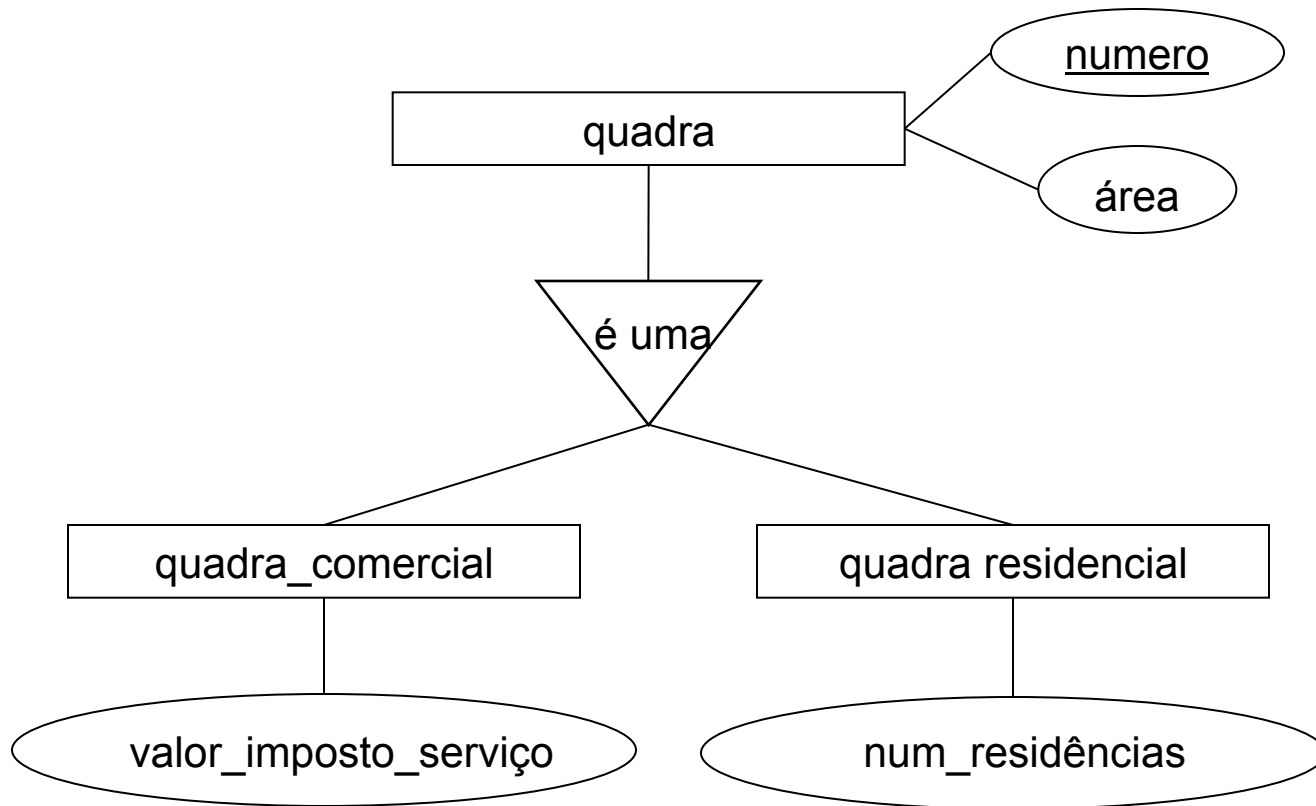


Modelo Entidade-Relacionamento (E-R)

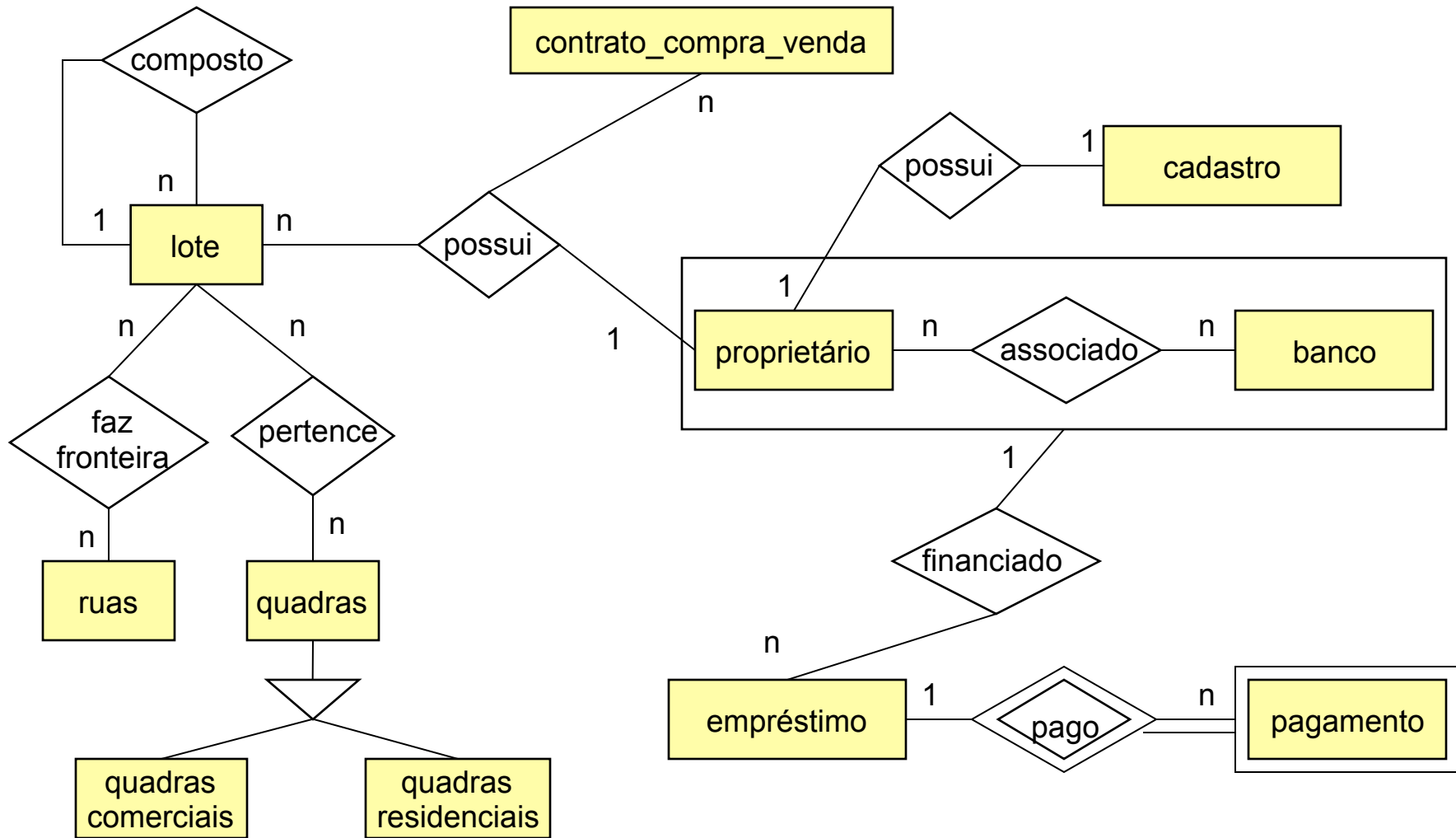
- Generalização
 - Enfatiza a semelhança entre diferentes tipos de entidades e abstrai suas diferenças.
- Especialização:
 - Subdivide entidades semelhantes em conjuntos de entidades mais específicas e enfatiza suas diferenças.

Modelo Entidade-Relacionamento (E-R)

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



Modelo Entidade-Relacionamento (E-R)



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



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Modelo Relacional, Álgebra Relacional e SQL

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*
- Diferentemente do modelo E-R, um modelo relacional implica uma implementação para um SGBD relacional em particular:
 - IBM, Informix, Microsoft, Oracle, Sybase, etc.

Modelo Relacional

- A maioria dos SGBDs seguem um modelo relacional ou objeto-relacional
- **Modelo de Dados Relacional:** Banco de dados é organizado em uma coleção de relações ou tabelas relacionadas entre si.

Aluno

MATRICULA	NOME	CURSOID
98765	João	MAT
67765	José	BIO
84562	Maria	ENG
34256	Luis	INFO
3452672	Ana	MAT
34529	Luana	MAT

Curso

CURSOID	TITULO	DURAÇÃO
INFO	Informática Indust.	4
BIO	Biologia	4
ENG	Engenharia Civil	5
MAT	Licenciatura Mat.	4

Modelo Objeto-Relacional

- É uma extensão do Modelo Relacional com conceitos da modelagem Orientada por Objeto.
- Combina os benefícios dos dois modelos.
- Fornecem suporte para:
 - Criar objetos complexos
 - Executar consultas avançadas sobre dados complexos
- A linguagem de consulta OR é uma extensão da linguagem SQL para suportar o modelo de objetos




Modelo Objeto-Relacional

Exemplo: Oracle

```
CREATE TYPE SDO_GEOMETRY AS OBJECT (  
  SDO_GTYPE          NUMBER,  
  SDO_SRID           NUMBER,  
  SDO_POINT          SDO_POINT_TYPE,  
  SDO_ELEM_INFO     SDO_ELEM_INFO_ARRAY,  
  SDO_ORDINATES     SDO_ORDINATE_ARRAY);
```

```
CREATE TABLE Estados (  
  SIGLA    VARCHAR(2)  
  POP      NUMBER(10,10)  
  GEOM     SDO_GEOMETRY)
```

Estados

SIGLA	POP	GEOM
MG	222222	
RJ	333333	
SP	444444	

Modelo Relacional: Relação

Conceito de *relação*:

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

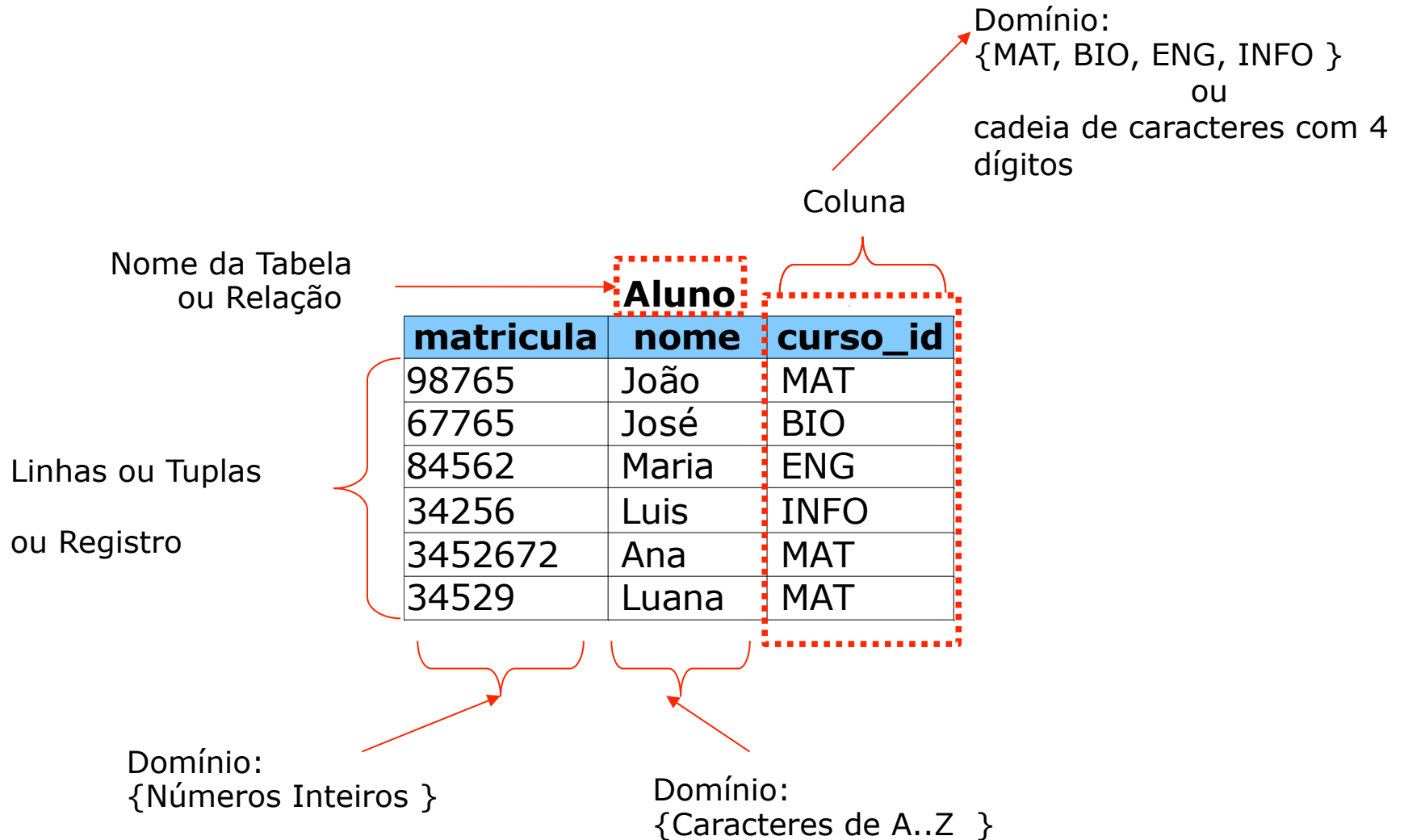
$$R = \{ (d_1, d_2, \dots, d_n) \mid d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n \}$$

- 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: Relação

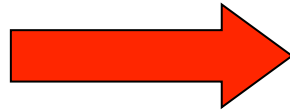
- No dia a dia, usamos o termo tabela ao invés de relação
- Cada tabela possui um nome
- Cada linha representa uma tupla da Relação:
 - Algumas pessoas usam o termo registro ao invés de linhas
- A ordem das linhas não é significativa:
 - Mas todas as linhas são distintas
- A ordem das colunas é significativa:
 - Corresponde à ordem D_1, D_2, \dots, D_n dos domínios sobre os quais a Relação é definida
- Cada coluna (também chamada de atributo ou campo) possui um nome

Modelo Relacional: Relação

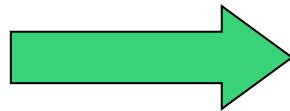


Instâncias x Esquemas

Esquema da
relação Aluno



Instâncias da
relação Aluno



Aluno		
matricula	nome	curso_id
98765	João	MAT
67765	José	BIO
84562	Maria	ENG
34256	Luis	INFO
3452672	Ana	MAT
34529	Luana	MAT

Modelo Relacional: Operações

- Os operadores existentes realizam operações de recuperação que geram novas tabelas a partir das existentes
- Permitem, por exemplo, extrair um subconjunto das linhas ou colunas
- Ex: informações aluno 34256

Aluno		
matricula	nome	curso_id
98765	João	MAT
67765	José	BIO
84562	Maria	ENG
34256	Luis	INFO
3452672	Ana	MAT
34529	Luana	MAT



Resultado:

Aluno		
matricula	nome	curso_id
34256	Luis	INFO

Modelo Relacional: Chaves

- Chave Primária (*Primary Key* ou *PK*):
 - Campo ou conjunto de campos cujos valores identificam unicamente cada linha de uma tabela

chave primária

Aluno

matricula	nome	curso_id
98765	João	MAT
67765	José	BIO
84562	Maria	ENG
34256	Luis	INFO
3452672	Ana	MAT
34529	Luana	MAT

chave primária
composta

TelefoneAluno

matricula	fone	tipo
98765	9876-0001	CEL
98765	551-4243	RES
67765	551-4243	RES
84562	8123-0179	CEL
34256	8888-8888	CEL
34529	9999-9999	CEL

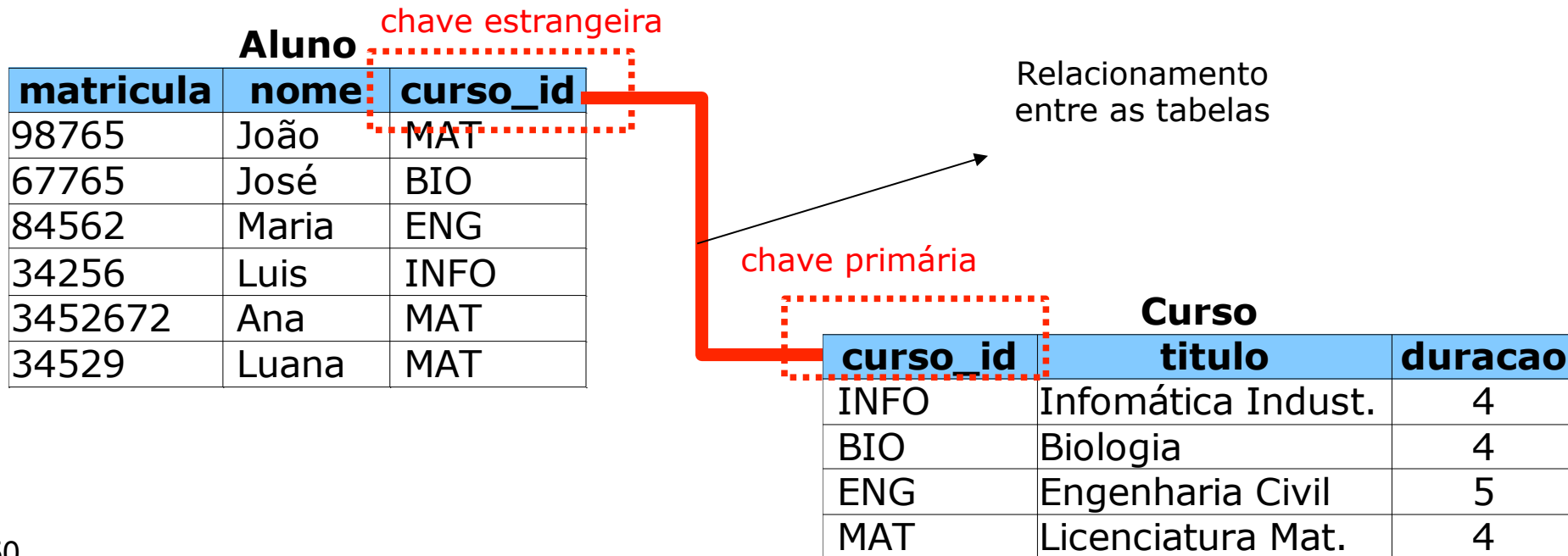
chave primária

Curso

curso_id	titulo	duracao
INFO	Infomática Indust.	4
BIO	Biologia	4
ENG	Engenharia Civil	5
MAT	Licenciatura Mat.	4

Modelo Relacional: Chaves

- Chave Estrangeira (*Foreign Key* ou *FK*):
 - Coluna ou combinação de colunas, cujos valores aparecem necessariamente na chave primária de uma outra tabela
 - Obs.: uma chave estrangeira não precisa ter o mesmo nome do que a chave primária correspondente na outra tabela



Restrições de Integridade

- Também conhecidas como *constraints*
- Especificam critérios para manter a consistência dos dados
- Tipos de restrições de integridade:
 - Restrições de domínio: especifica qual o tipo de cada atributo
 - Restrições de chave: chave primária e candidata (valores únicos)
 - Restrições de nulidade: especifica se o valor de um atributo pode ou não ser nulo
 - Restrição de integridade valor (semântica ou física):
 - Exemplo: o preço de um produto não pode ser zero (preço ≥ 0)
 - Restrições de integridade referencial ou chave estrangeira:
 - Especifica uma restrição de integridade referencial entre duas relações R_1 e R_2

Transação

Conjunto de procedimentos executados por um SGBD e que para o usuário é visto como uma única ação.

Ex.: transferência de R\$ 500,00 do cliente A para o cliente B.

tempo	Cliente A	Cliente B
T1	Verificar saldo	
T2	Debitar R\$500	
T3		Depositar R\$500

Erro no sistema



Transação

A integridade da transação depende de 4 propriedades (ACID):

- (1) **Atomicidade:** Uma transação nunca é executada pela metade.
- (2) **Corretude ou Consistência:** A execução de uma transação isolada (sem execução concorrente de outra transação) preserva a consistência do banco.
- (3) **Isolamento:** Transações concorrentes não podem ser intercaladas de forma a gerar inconsistência.
- (4) **Durabilidade:** Se a transação completar-se com sucesso, todas as mudanças feitas no banco de dados persistem, até mesmo se houver falhas no sistema (Mecanismo para recuperar falhas - Rollback).

Trigger

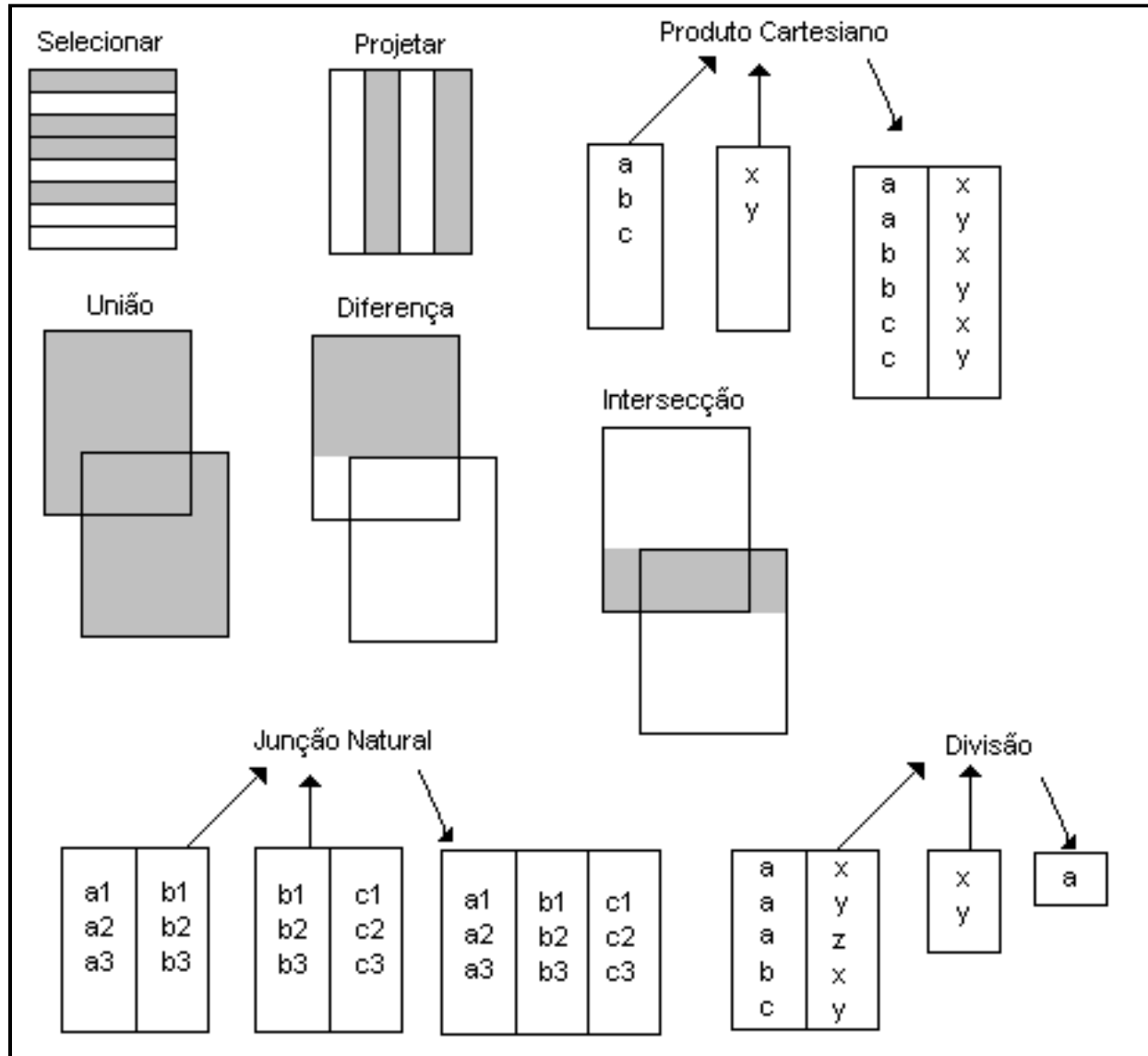
Trigger (gatilho): um gatilho é um comando executado automaticamente pelo sistema como um efeito colateral de uma modificação no banco de dados. Exemplo: sempre que alguém alterar a nota de um aluno, recalcule sua média.



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

Álgebra Relacional - Resumo



Operadores da Álgebra Relacional

- Seleção:
 - seleciona tuplas que satisfazem um certo predicado ou condição

Clientes

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

a) selecionar tuplas cujo nome = João

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

<i>Nome</i>	<i>Registro</i>
João	1

Operadores da Álgebra Relacional

b) selecionar as tuplas de Clientes cujo registro > 1

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

Nome	Registro
Maria	2
José	3

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

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

Nome	Registro
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

Nome	Registro
João	1
Maria	2
José	3



Nome
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

Nome	Conta
João	1
Maria	2
José	3

Emprestimo

Nome	Empréstimo
Paulo	100
Maria	200
Carlos	300

Resultado da união

=

Nome
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

Nome	Conta
João	1
Maria	2
José	3

Emprestimo

Nome	Empréstimo
Paulo	100
Maria	200
Carlos	300

Resultado da diferença

Nome
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)

<i>Nome_{cc}</i>	<i>Conta</i>	<i>Nome_{emp}</i>	<i>Empréstimo</i>
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

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

ContaCorrente

Nome	Conta
João	1
Maria	2
José	3

Emprestimo

Nome	Empréstimo
Paulo	100
Maria	200
Carlos	300

=

Resultado da intersecção

Nome
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:

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

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

SQL - Structured Query Language

- Linguagem padrão (ISO) para sistemas de bancos de dados
- É uma linguagem declarativa de alto nível que permite:
 - Consultar dados
 - Definir e alterar dados
 - Definir visões
 - Especificar autorização e regras de segurança
 - Definir restrições de integridade
 - Criar índices
 - Controlar transações
 - ...

SQL - Structured Query Language

- Linguagem de consulta usada pelos SGBD-R e SGBD-OR
- Baseada na álgebra 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

SQL-DDL

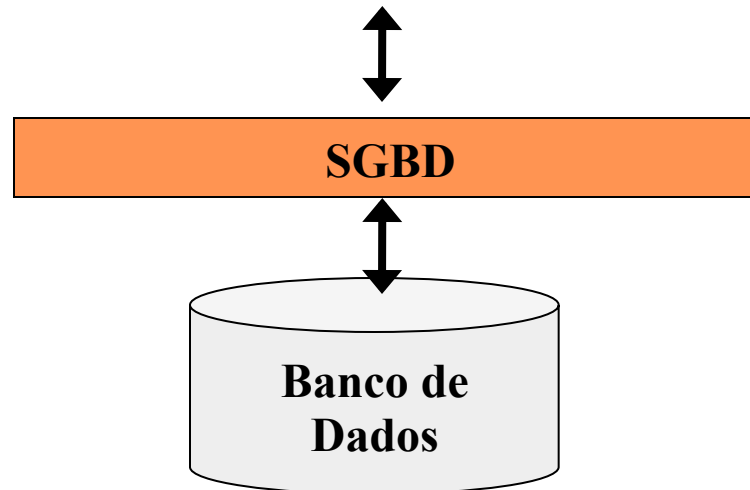
```
CREATE DATABASE Teste
```

```
CREATE TABLE Estados (  
  NOME      VARCHAR(100)  
  SIGLA     VARCHAR(2)  
  POP       NUMBER(10,10))
```

SQL-DML

```
INSERT INTO Estados  
VALUES ("Minas  
Gerais", "MG",  
9999)
```

```
SELECT *  
FROM Estados  
WHERE SIGLA = "MG"
```



SQL - Structured Query Language

- Alguns comandos em SQL

Comandos	Usado para	Tipo
<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'
```

SQL - Structured Query Language

DDL– Data Definition Language

CREATE DATABASE – cria um novo banco de dados

ALTER DATABASE – modifica um banco de dados

CREATE TABLE – cria uma nova tabela

ALTER TABLE – altera uma tabela

DROP TABLE – remove uma tabela

CREATE INDEX – cria um índice

DROP INDEX – remove um índice

SQL - Structured Query Language

DML – Data Manipulation Language

SELECT – extrai dados de um banco de dados

UPDATE – altera os dados de um banco de dados

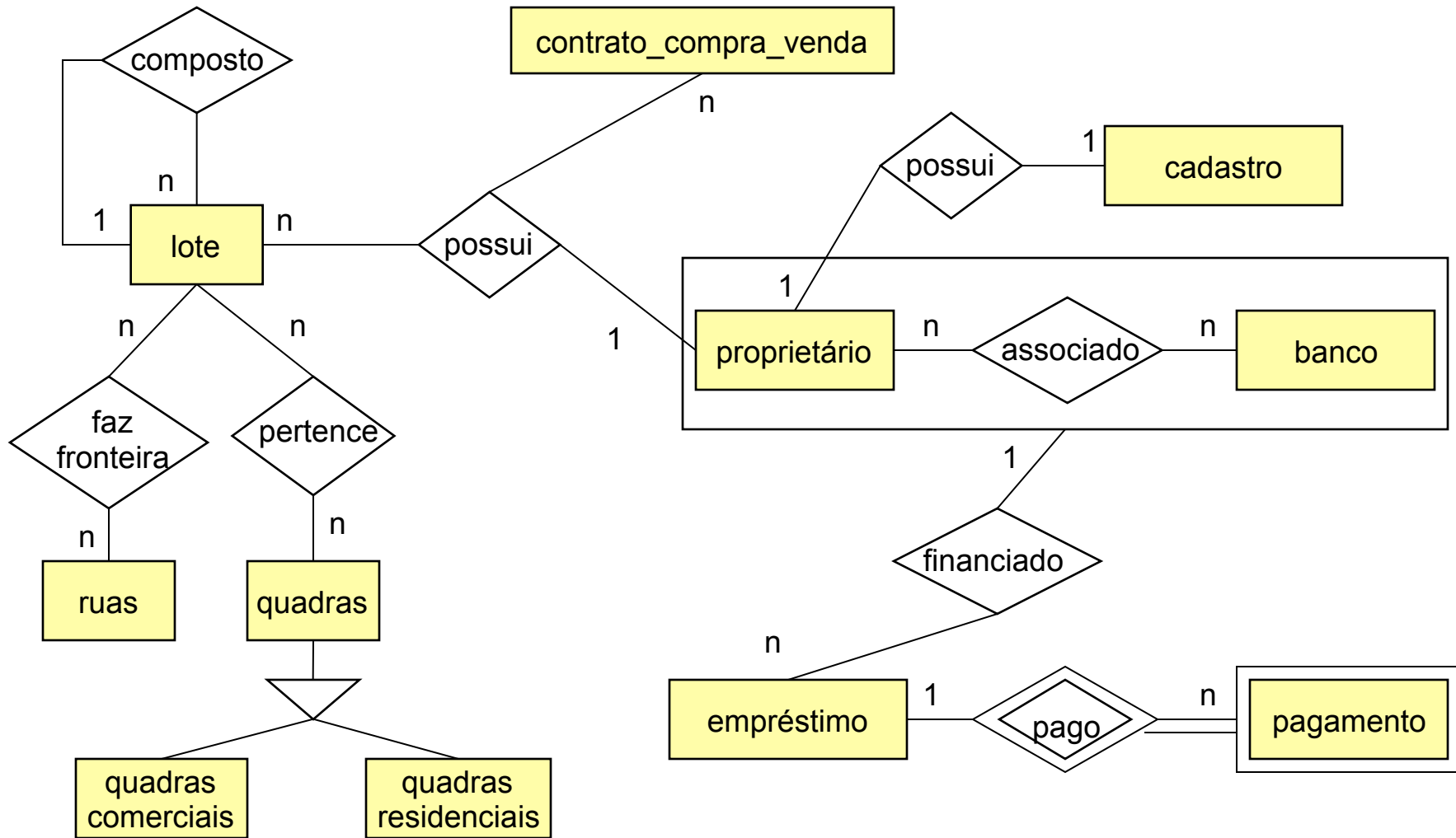
DELETE – apaga dados de um banco de dados

INSERT INTO – insere dados no banco de dados

SQL - Structured Query Language

Próxima aula - laboratório: SQL na prática!!!!

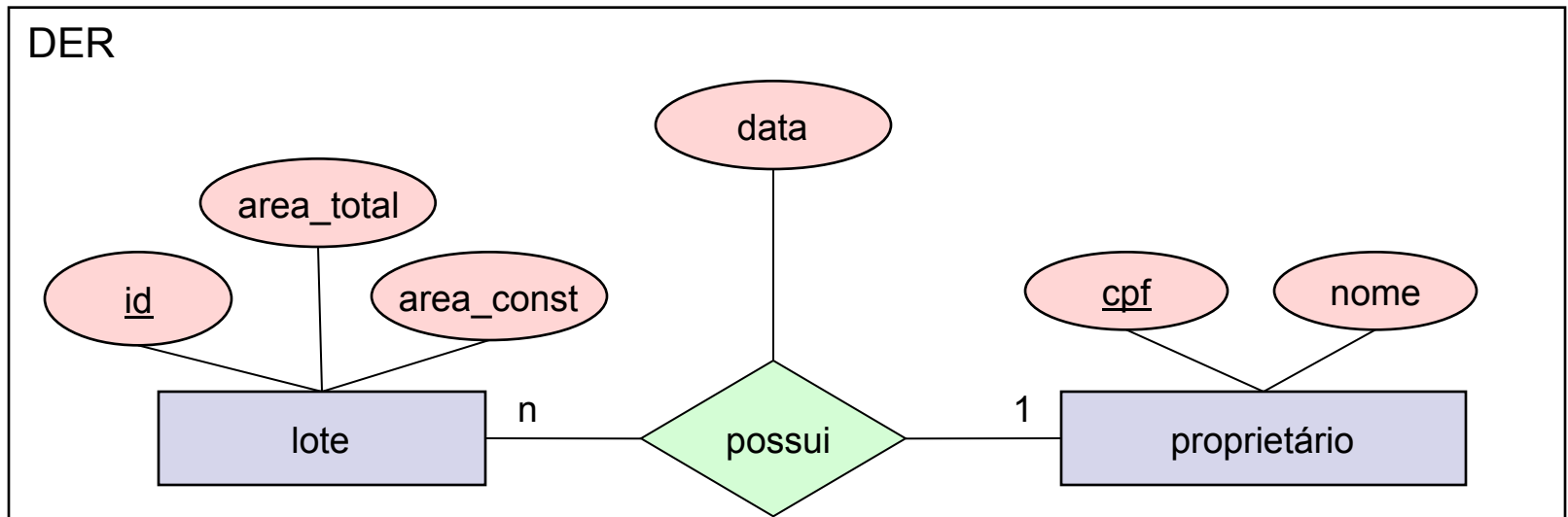
Modelo Entidade-Relacionamento (E-R)



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)

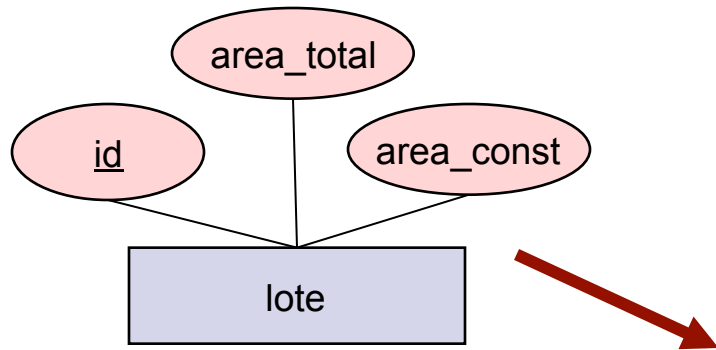
Lote_proprietario (id_lote, cpf, data)

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:



Lote(id, area_total, area_const)

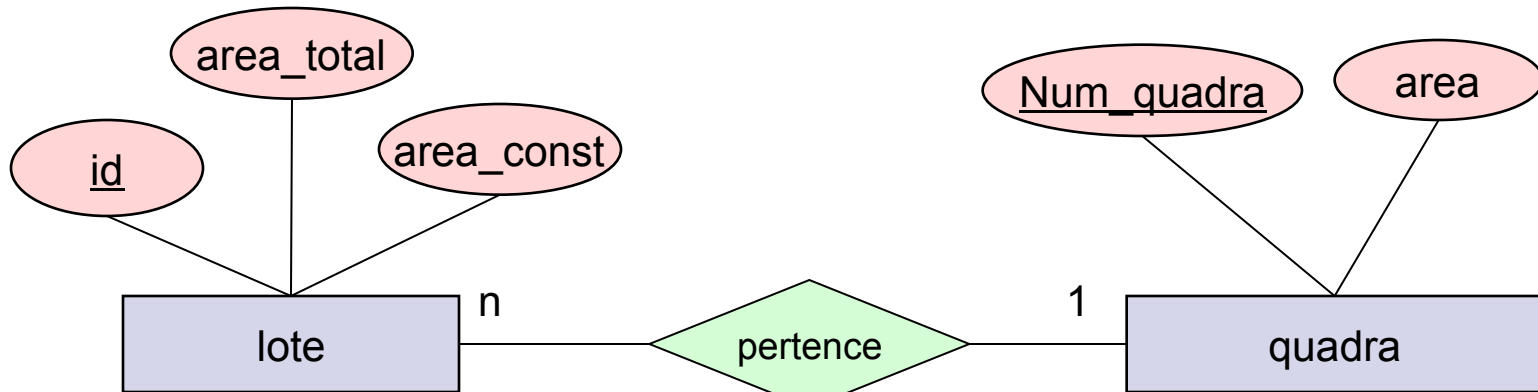
Conversão E-R → Modelo Relacional

- 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

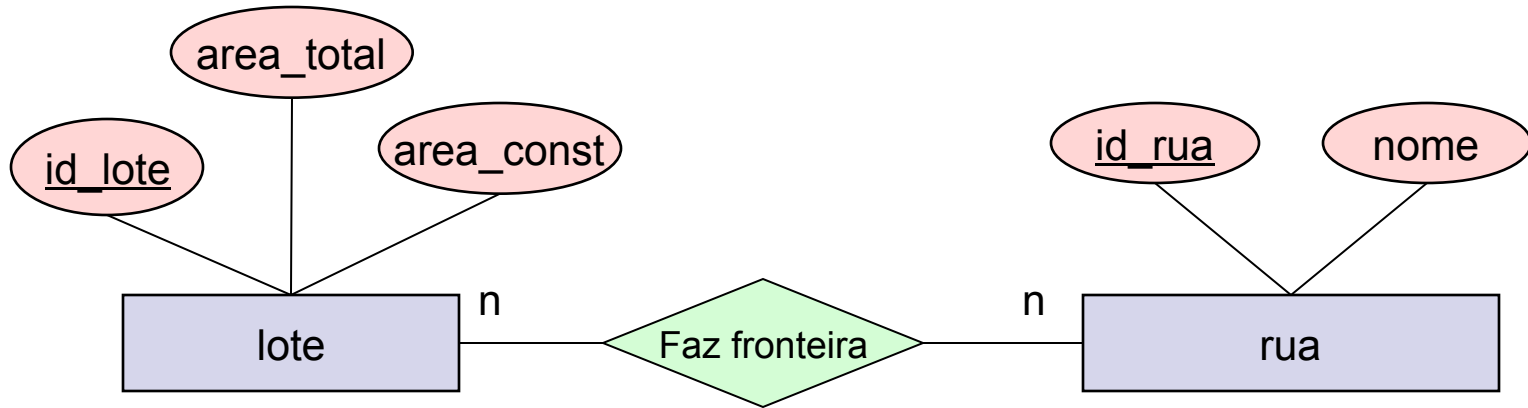


Lote(id, area_total, area_const, num_quadra)

Quadra(num_quadra, area)

Conversão E-R → Modelo Relacional

- Relacionamento N:N



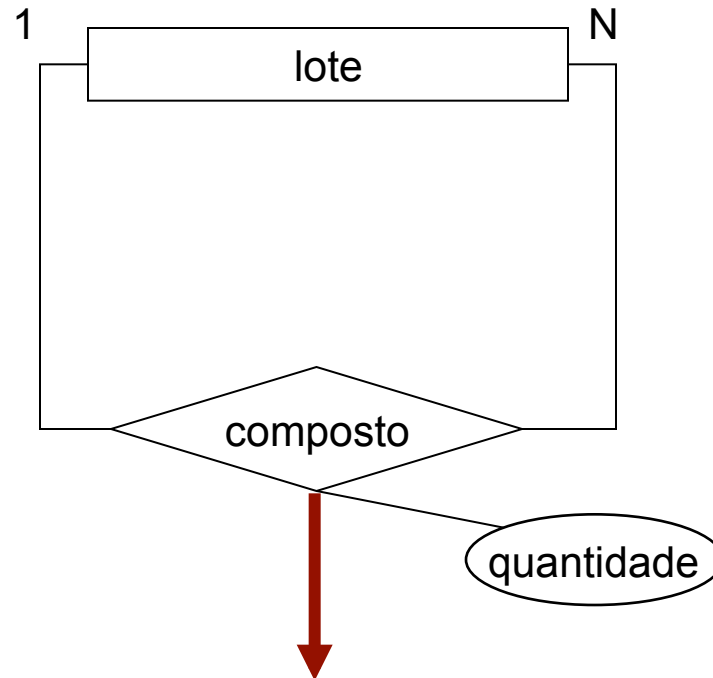
Lote(id_lote, area_total, area_const)

Fronteira(id_lote, id_rua, num_inicial, num_final)

Rua(id_rua, nome)

Conversão E-R → Modelo Relacional

- Auto relacionamento

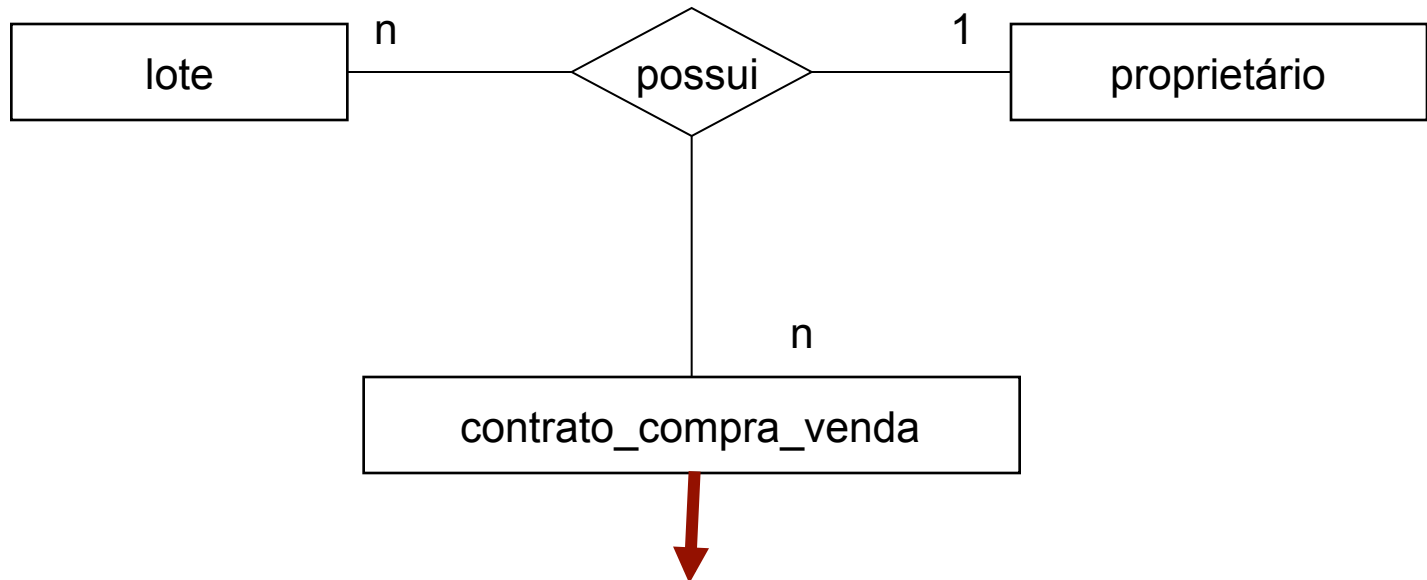


Lote(id_lote, area_total, area_const)

Composição(id_lote, id_lote_comp, quantidade)

Conversão E-R → Modelo Relacional

- Relacionamento múltiplo



Lote(id_lote, area_total, area_const)

Proprietario(cpf, nome)

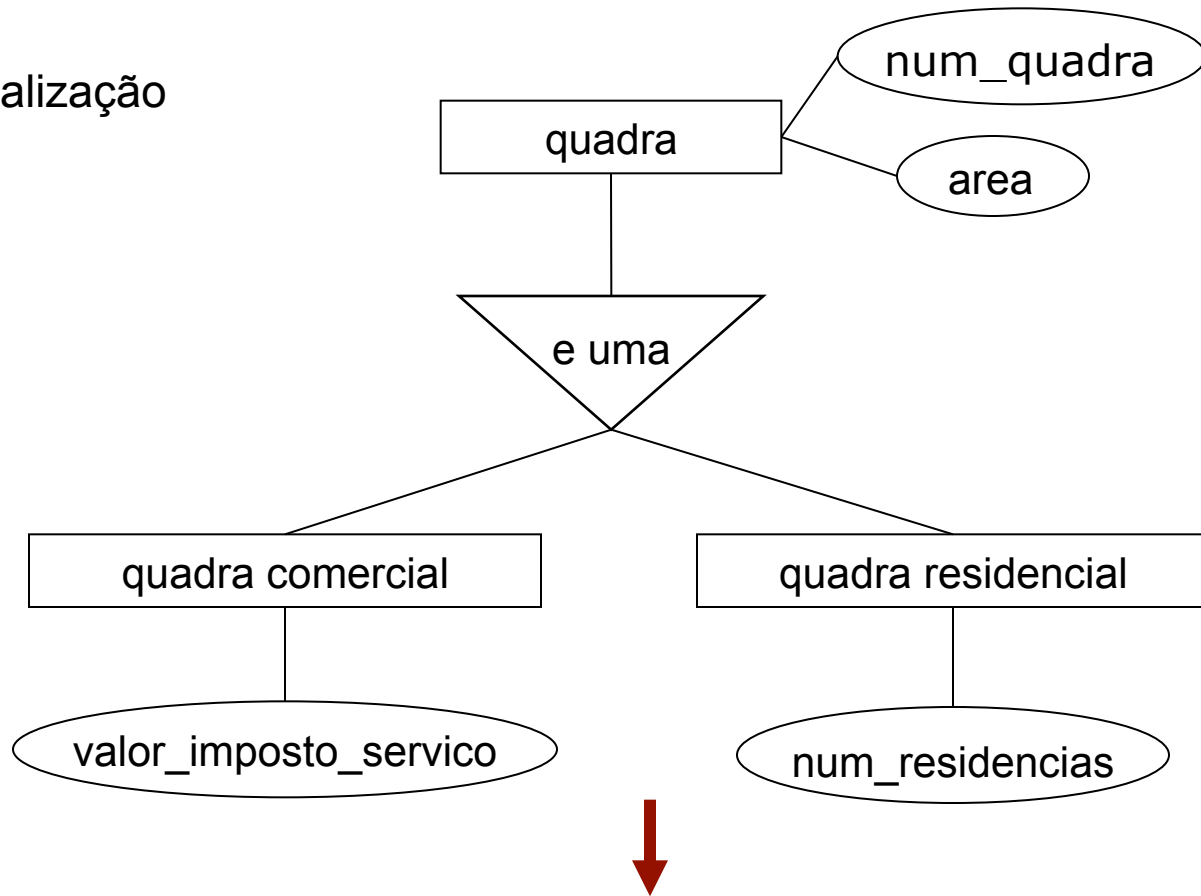
Contrato(id_contrato, documento)

Lote_Prop_Contr(id_lote, cpf, id_contrato, data)

Conversão E-R → Modelo Relacional

- Especialização

Solução 1



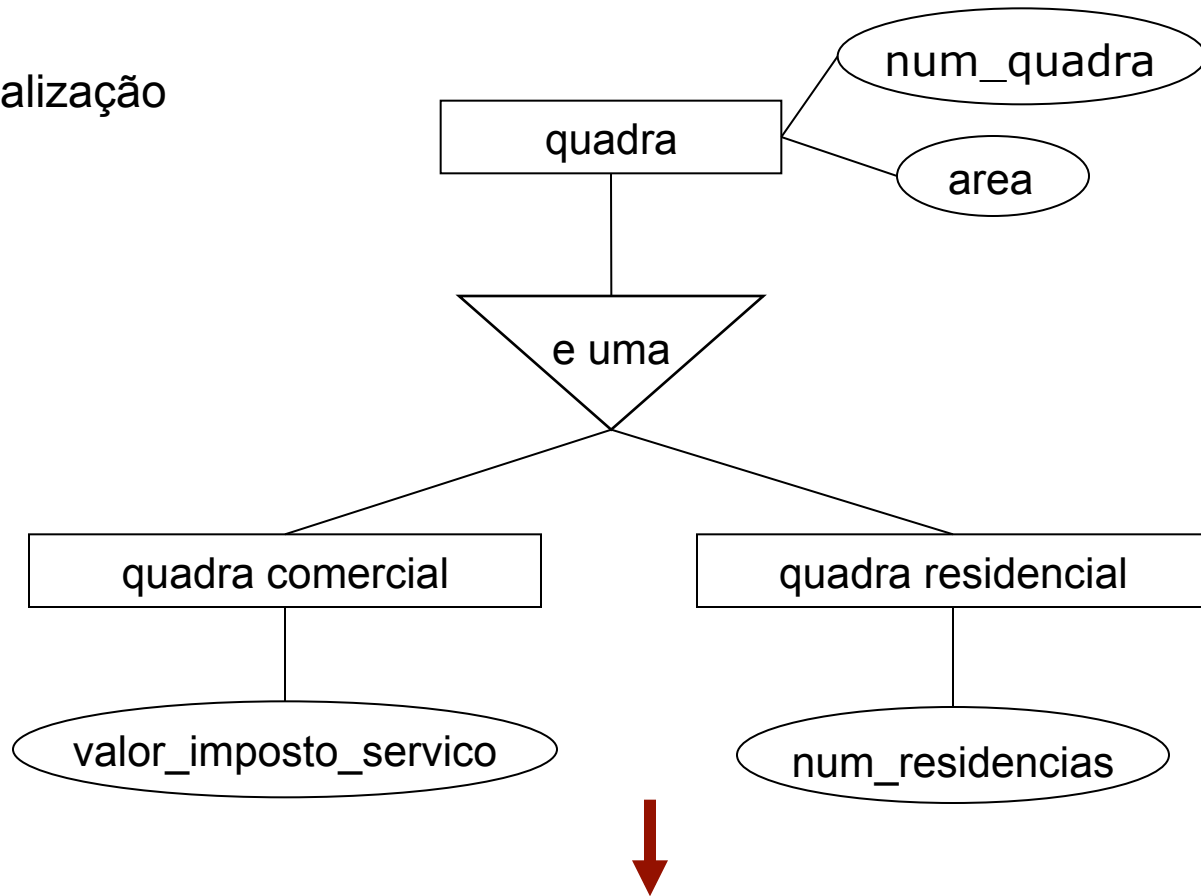
Quadra_comercial (num_quadra, area, imposto_servico)

Quadra_residencial (num_quadra, area, num_residencias)

Conversão E-R → Modelo Relacional

- Especialização

Solução 2



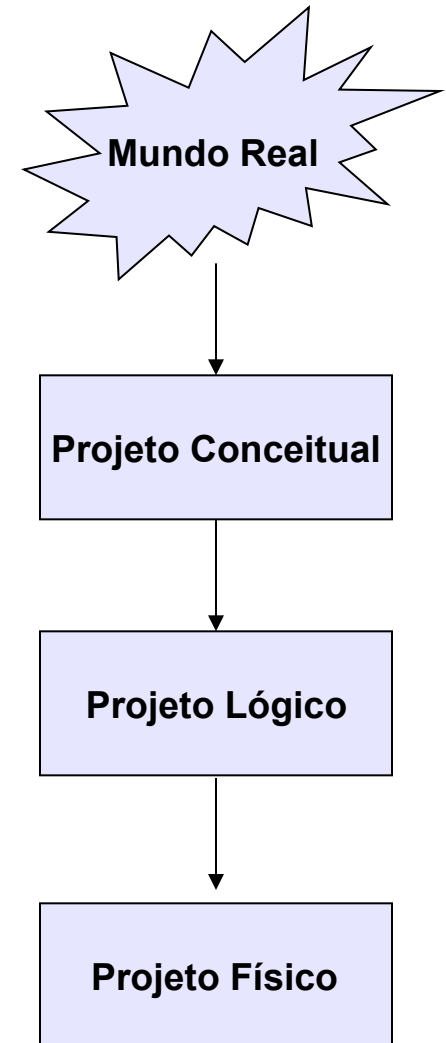
Quadra(num_quadra, area)

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
- Regra de ouro: entre com o mínimo de data necessário, evite duplicar informação com menor risco para a integridade dos dados

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 \rightarrow nome_func
 - cod_prod \rightarrow desc_prod
 - num_solic, cod_prod \rightarrow 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

Exemplos de SGBDs relacionais e objeto-relacionais



- Auto-contido: necessita do mínimo de suporte de outras bibliotecas ou sistemas operacionais
- Não é um servidor: é responsável por ler e escrever os arquivos da base, sem comunicação entre processos
- Não requer configuração
- Transacional
 - Transação: uma instrução simples e lógica sobre o banco de dados. Ex: transferência de fundos de uma conta para outra.
 - **Atomic**: tudo ou nada
 - **Consistent**: de um estado consistente para outro
 - **Isolated**: uma transação não altera os dados manipulados por outra
 - **Durable**: transações que são executadas com sucesso, não são perdidas



- SQLiteStudio is a SQLite database manager with the following features:
- Single executable file - no need to install or uninstall. Binary distribution is just the single, ready to use file.
- Intuitive interface,
- All SQLite3 and SQLite2 features wrapped within simple GUI,
- Cross-platform - runs on Windows 9x/2k/XP/2003/Vista/7, Linux, MacOS X, Solaris, FreeBSD and should work on other Unixes (not tested yet).
- Localizations, currently translated to: English, Polish, Spanish, German, Russian, Japanese, Italian, Dutch, Chinese,



- Exporting to various formats (SQL statements, CSV, HTML, XML),
- Numerous small additions, like formatting code, history of queries executed in editor windows, on-the-fly syntax checking, and more,
- UTF-8 support,
- skinnable (interface can look native for Windows 9x/XP, KDE, GTK, Mac OS X, or draw widgets to fit for other environments, WindowMaker, etc),
- Configurable colors, fonts and shortcuts.
- Open source and free - Released under GPLv2 licence.

PostgreSQL



<http://www.postgresql.org>



PostgreSQL

- Site oficial: <http://www.postgresql.org>
- Sistema Gerenciador de Bancos de Dados Objeto-Relacional
- Um dos mais avançados SGBD open source: licença BSD
- Multi-plataforma: AIX, Solaris, Linux, OpenBSD, Windows e outros
- Características:
 - Funções:
 - PL/PgSQL, PL/Perl, PL/Python, PL/Tcl, C, C++, ...
 - Triggers (gatilhos)
 - Índices: B+-tree, hash, GiST
 - Transação e Concorrência
 - Chaves estrangeiras (foreign keys)
 - Tipos de dados e funções definidos pelo usuário



Utilização do PostgreSQL

- Através de interfaces de programação:
 - C, PHP, Java (JDBC), ODBC, Python, Perl
- Através de terminais interativos:
 - ferramentas que permitem estabelecer comunicação com o servidor de bancos de dados e enviar comandos SQL para que o servidor faça o processamento.
 - Os dois terminais mais conhecidos são:
 - psql (terminal – sem interface gráfica)
 - **PgAdmin III** (ferramenta gráfica)

PgAdmin III



The screenshot displays the PgAdmin III application window. The title bar reads "pgAdmin III". The menu bar includes "Arquivo", "Editar", "Visualizar", "Ferramentas", and "Ajuda". The toolbar contains various icons for connection, refresh, navigation, and help.

The "Navegador de objetos" (Object Navigator) pane on the left shows a tree structure:

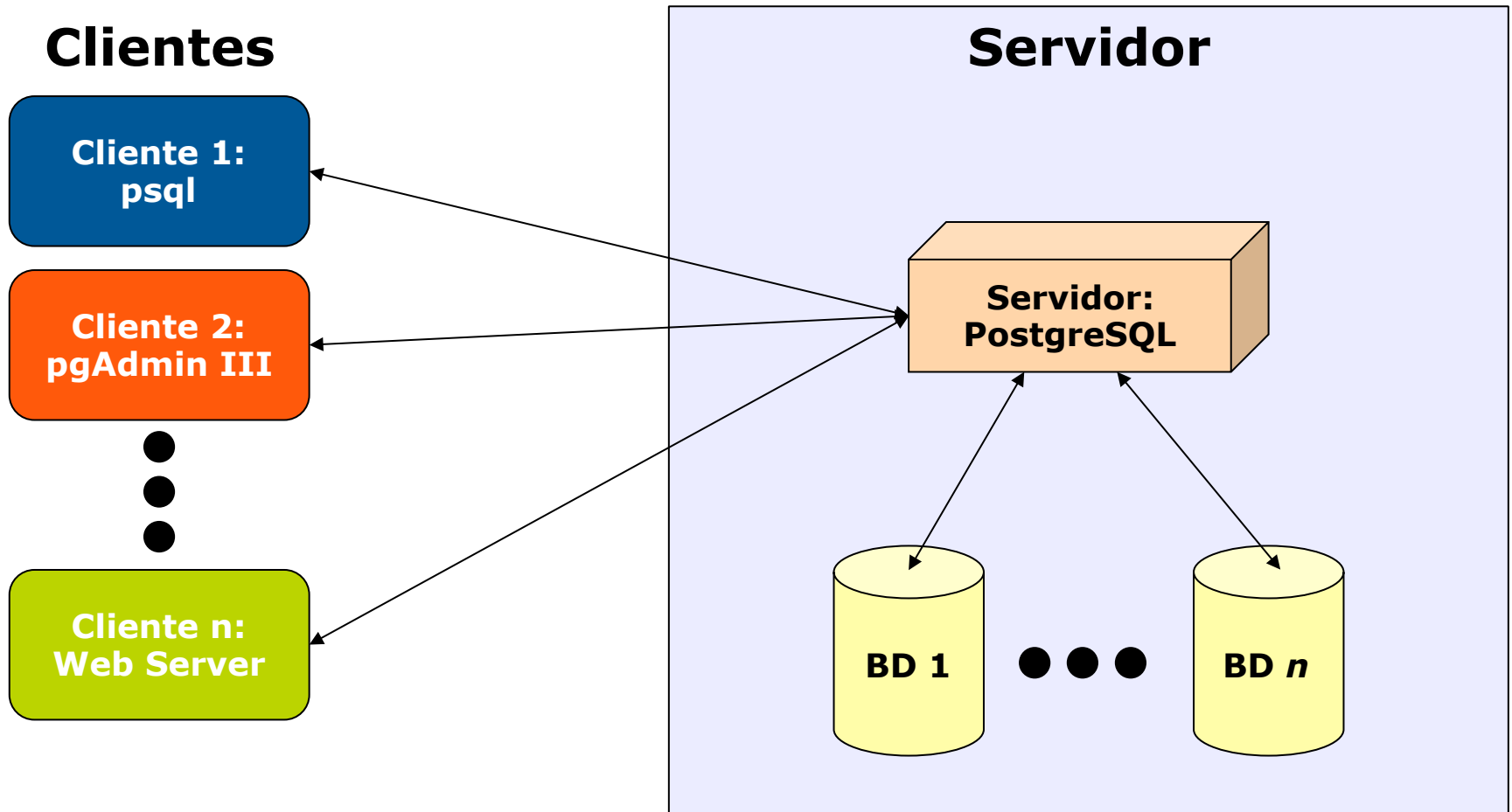
- Servidores (1)
 - PostgreSQL Database Server 8.3 (localhost:5432)
 - Bancos de Dados (3)
 - dbgeo
 - postgres
 - template_postgis
 - Tablespaces (2)
 - Roles do Grupo (0)
 - Login Roles (1)

The "Propriedades" (Properties) pane on the right shows the following table:

Propriedade	Valor
Descrição	PostgreSQL Database Server 8.3
Nome da Máquina	localhost
Porta	5432
Serviço	pgsql-8.3
Manutenção do ban...	postgres
Nome de usuário	postgres
Armazenar senha?	Sim
Restaurar ambiente?	Não

The "Painel SQL" (SQL Panel) at the bottom is currently empty. The status bar at the bottom indicates "Estabelecendo conexão...Concluído" and "0.25 seg."

Cliente/Servidor



- Gerencia os arquivos dos bancos de dados
- Aceita conexões com as aplicações clientes
- Realiza operações nos bd em nome dos clientes