

Curso de Especialização:
Engenharia e Administração de Sistemas de Banco de Dados

Fundamentos de Sistemas de Banco de Dados

Conceitos Fundamentais

Profa. Dra. Gisele Busichia Baioco

gisele@ft.unicamp.br



UNICAMP



FACULDADE DE TECNOLOGIA



Conteúdo

- Banco de Dados – BD
- Sistema de Gerenciamento de Bancos de Dados – SGBD
- Sistema de Bancos de Dados – SBD
- Projeto de Bancos de Dados
- A Linguagem SQL – *Structured Query Language*



Banco de Dados – BD

■ Definição

- Um **Banco de Dados** ou **Base de Dados** é uma coleção de dados (fatos conhecidos) logicamente relacionados, com algum significado.

■ Propriedades

- Associações aleatórias de dados **não** podem ser chamadas de BD;
- Um BD é **projetado, construído e preenchido** com dados para um **propósito específico**.
 - tem um grupo de **usuários** e algumas **aplicações** para atender esses usuários;
- Um BD representa algum aspecto do mundo real, conhecido como **Mini-Mundo** ou **Universo de Discurso**.
 - mudanças no Mini-Mundo provocam mudanças no BD.

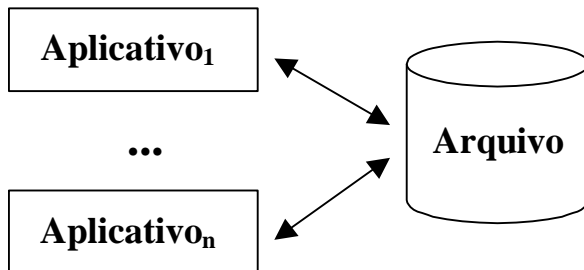
Sistema de Gerenciamento de Bancos de Dados – SGBD

código executável e dados compõem um único objeto.

■ Como armazenar dados em disco?

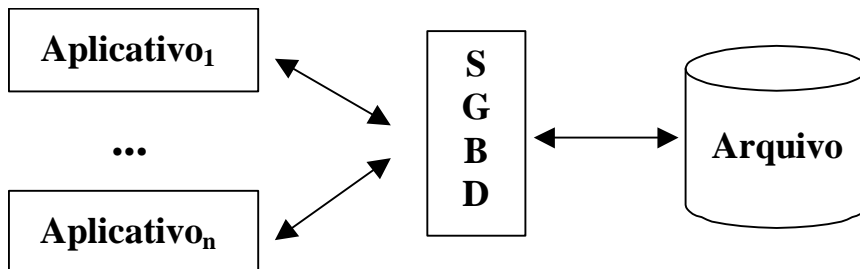
1. Programa aplicativo com os dados armazenados

2. Sistema de Arquivos



programas aplicativos ficam isolados dos dados;
procedimentos relacionados a gerência de dados (armazenamento e recuperação) ficam embutidos no código-fonte dos programas.

3. Sistema de Gerenciamento de Bancos de Dados



promove a independência dos programas aplicativos em relação aos dados armazenados.



Sistema de Gerenciamento de Bancos de Dados – SGBD

■ Definição

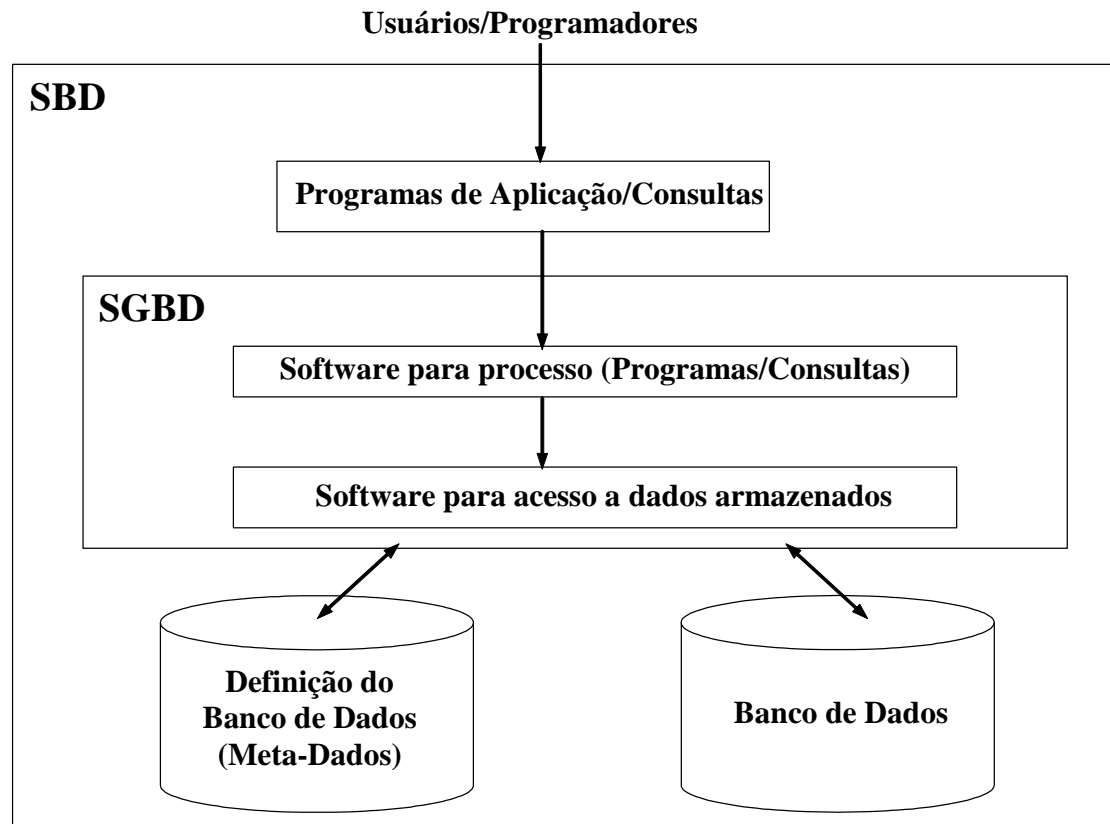
- Um **Sistema de Gerenciamento de Bancos de Dados** é um sistema de software de propósito geral que facilita o processo de **definir, construir e manipular** BDs de diversas aplicações.

■ Capacidades básicas

- Compartilhamento de Dados – controle de concorrência;
- Restrições de Acesso Multiusuário;
- Reforço de Restrições de Integridade – tipos de dados, relacionamentos;
- Fornecimento de *Backup* e *Recovery* no caso de falhas de hardware e software.

Sistema de Bancos de Dados – SBD

- **Sistema de Bancos de Dados:** composto pelos BDs e o SGBD.





Sistema de Bancos de Dados – SBD

- Um SBD é composto por três partes:
 - **Interface com o usuário:**
 - valida as solicitações dos usuários;
 - inicia operações em uma BD de acordo com as solicitações.
 - **Mecanismo de acesso:**
 - manutenção das estruturas de dados das BDs;
 - acesso aos dados;
 - manutenção da integridade dos dados.
 - **Armazenamento de dados:**
 - arquivos que contém os dados em si (BDs).



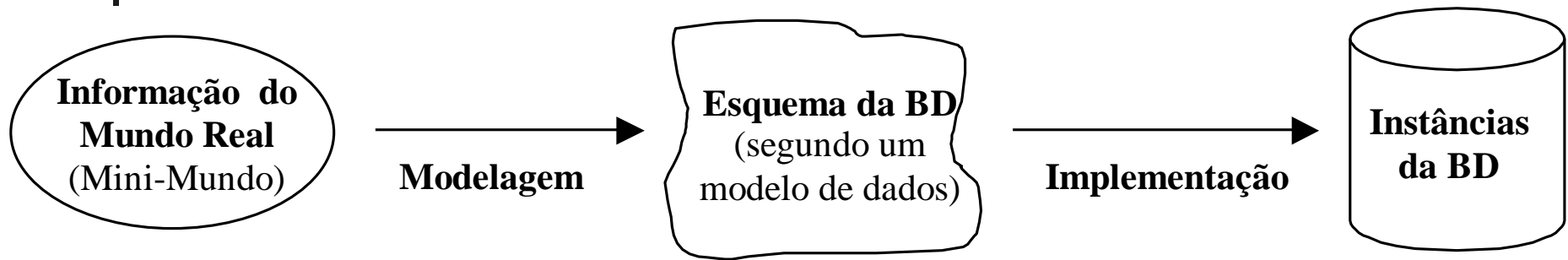
Sistema de Bancos de Dados – SBD

■ **Profissionais envolvidos:**

- Administrador do BD (DBA - *Database Administrator*):
 - autoriza o acesso ao BD, coordena e monitora seu uso;
 - soluciona problemas, tais como, quebra de segurança e baixo desempenho.
- Projetistas do BD:
 - identifica os dados a serem armazenados no BD;
 - escolhe estruturas apropriadas para representar e armazenar os dados.
- Usuários Finais:
 - acessam o BD para consultar, modificar dados e gerar relatórios.
- Analistas de Sistemas e Programadores de Aplicação:
 - os analistas especificam as transações para atender aos requisitos dos usuários finais;
 - os programadores implementam essas especificações produzindo os programas aplicativos.

Projeto de Bancos de Dados

Modelo de Dados, Esquemas e Instâncias

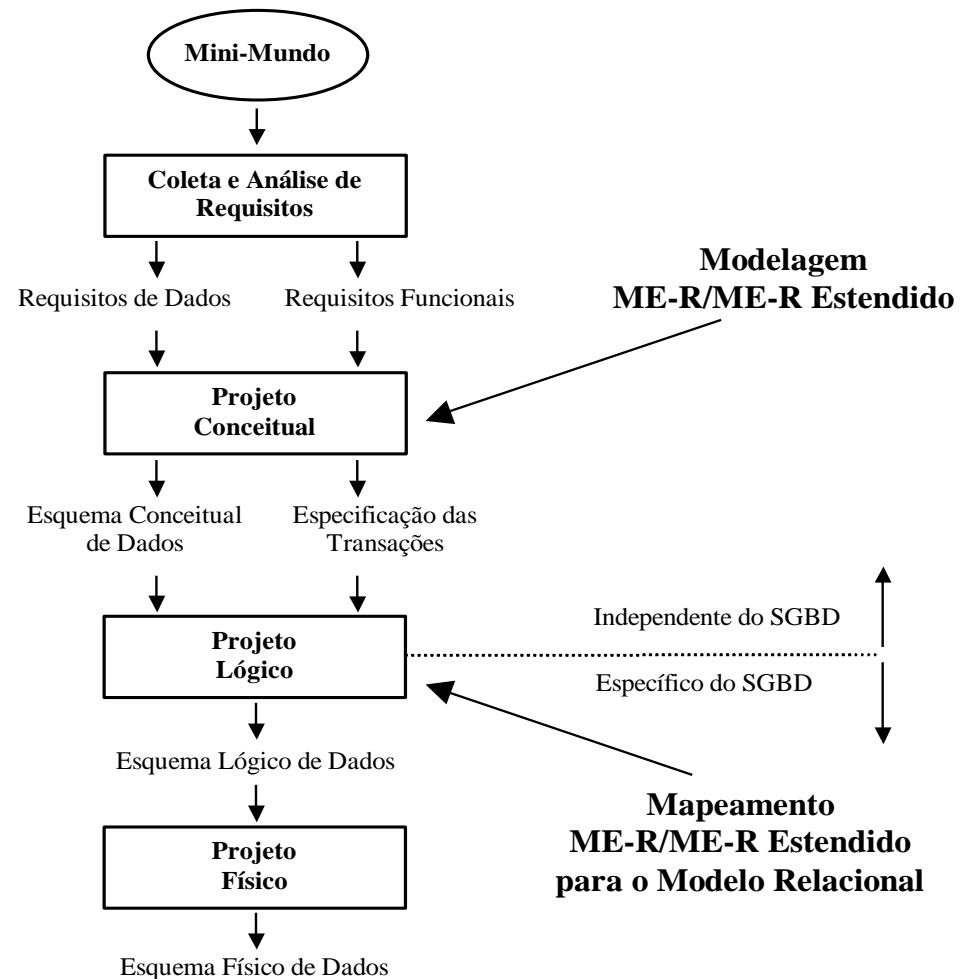


- **Modelo de Dados:** conjunto de conceitos e regras usados para descrever a estrutura de uma BD.
- **Modelagem de Dados:** processo de estruturar as informações do mundo real segundo um modelo de dados.
- **Esquema da BD:** é a descrição textual ou gráfica de uma BD de acordo com um determinado modelo de dados.
- **Instâncias (ou Ocorrências ou Estados) da BD:** são os dados de uma BD em um determinado momento do tempo.

Projeto de Bancos de Dados

Fases

- **Fases de Projeto de Bancos de Dados:**





Projeto de Bancos de Dados

Fase de Projeto Conceitual

- Abstrai a realidade independente do SGBD utilizado na implementação;
- Gera o **esquema conceitual de dados**;
- Modelos Conceituais:
 - Modelo Entidade-Relacionamento – ME-R;
 - ME-R Estendido;
 - Modelo de Classes da UML.



Projeto de Bancos de Dados

Projeto Conceitual – ME-R

- O ME-R possui 4 construtores semânticos:
 - Conjunto de Entidades;
 - Conjunto de Relacionamentos;
 - Atributos de Conjunto de Entidades;
 - Atributos de Conjunto de Relacionamentos.

Projeto de Bancos de Dados

Projeto Conceitual – ME-R

- Conjunto de Entidades (CE):
 - **Entidade** – representa um objeto sobre o qual deseja-se manter informações. Pode ser:
 - física: pessoa, carro, casa, funcionário, etc.
 - conceitual: departamento de uma empresa, curso de uma universidade, etc
 - **Conjunto de Entidades** – agrupa entidades que possuem propriedades semelhantes.
 - Representação:



CE

Projeto de Bancos de Dados

Projeto Conceitual – ME-R

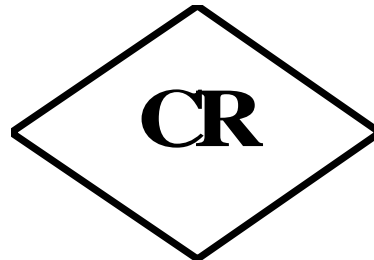
■ Conjunto de Relacionamentos (CR):

- **Relacionamento** – associação entre entidades.
Exemplos:

Funcionário.....Trabalhar.....Departamento

Aluno.....MatricularDisciplina

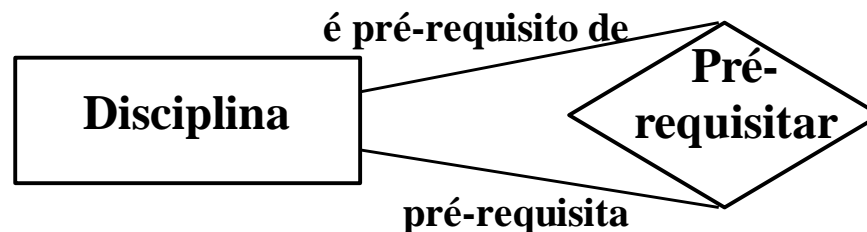
- **Conjunto de Relacionamentos** – coleção de relacionamentos que possuem propriedades semelhantes.
- Representação:



Projeto de Bancos de Dados

Projeto Conceitual – ME-R

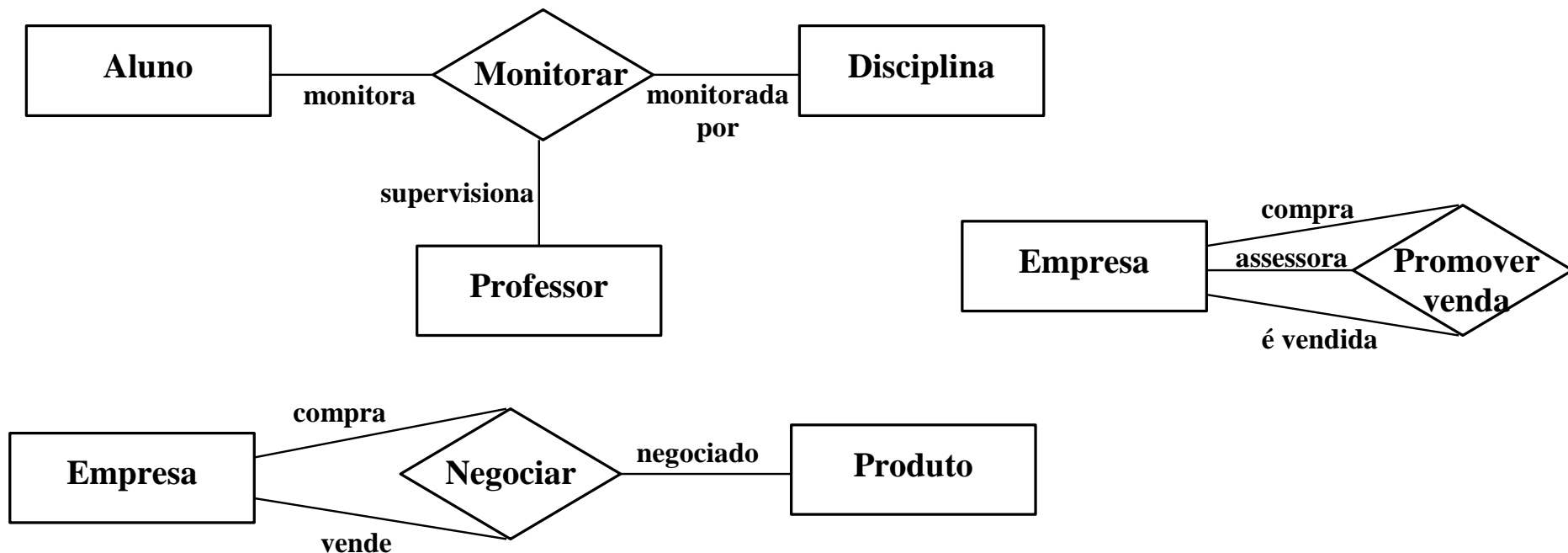
- Grau de relacionamentos: binários, ternários, etc
- Exemplos de relacionamento binários:



Projeto de Bancos de Dados

Projeto Conceitual – ME-R

- Grau de relacionamentos: binários, ternários, etc
- Exemplos de relacionamento ternários:

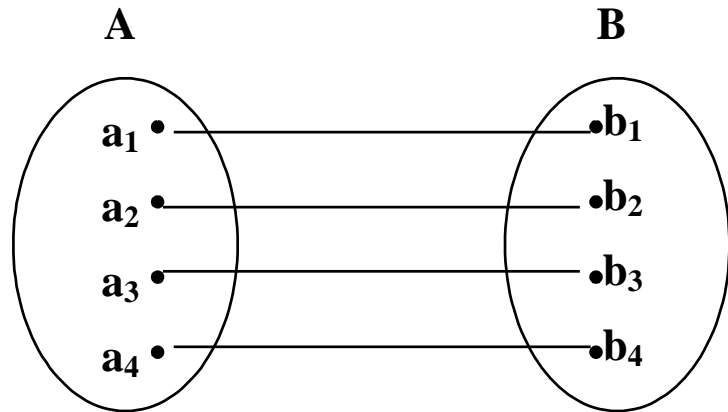


Projeto de Bancos de Dados

Projeto Conceitual – ME-R

- Cardinalidade de relacionamentos binários:

1:1 (um para um)



Exemplo:



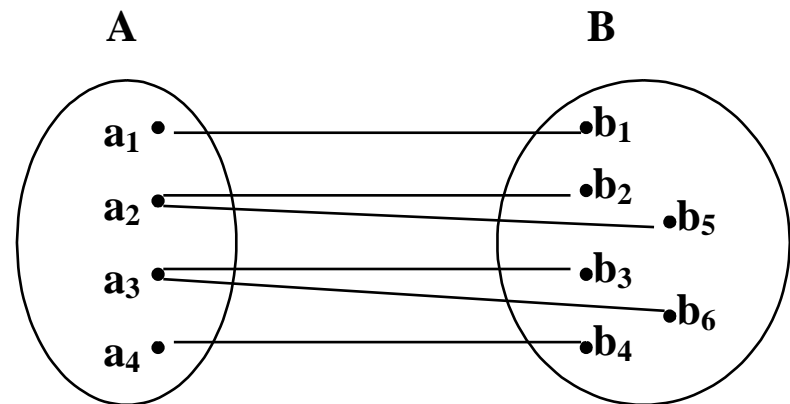
Projeto de Bancos de Dados

Projeto Conceitual – ME-R

- Cardinalidade de relacionamentos binários:

1:N (um para muitos)

N:1 (muitos para um)



Exemplo:

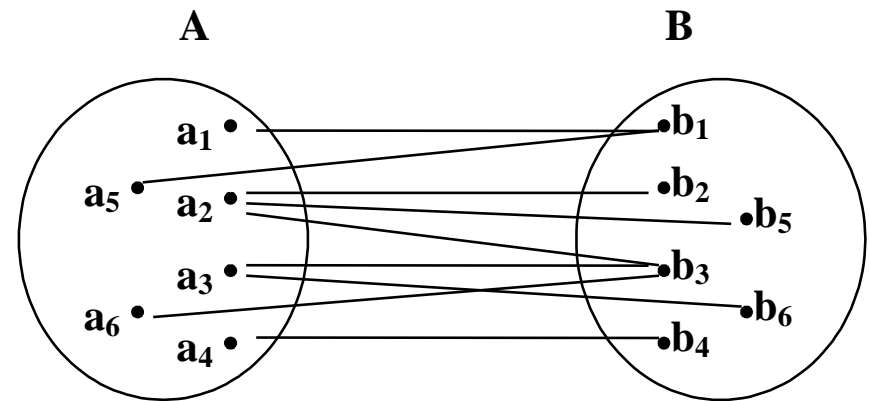


Projeto de Bancos de Dados

Projeto Conceitual – ME-R

- Cardinalidade de relacionamentos binários:

N:M (muitos para muitos)



Exemplo:



Projeto de Bancos de Dados

Projeto Conceitual – ME-R

- Cardinalidade de relacionamentos ternários:

Possibilidades:

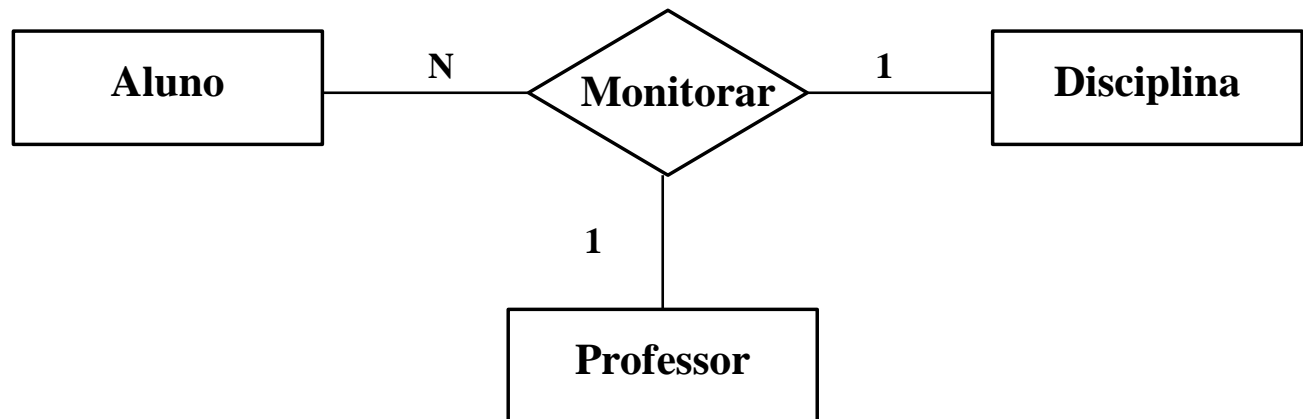
1:1:1

1:1:N

1:N:M

N:M:P

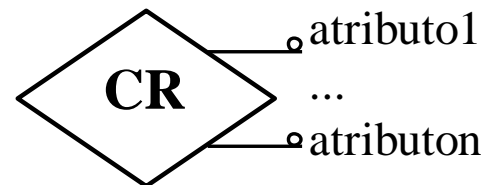
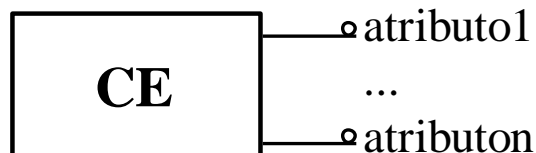
Exemplo:



Projeto de Bancos de Dados

Projeto Conceitual – ME-R

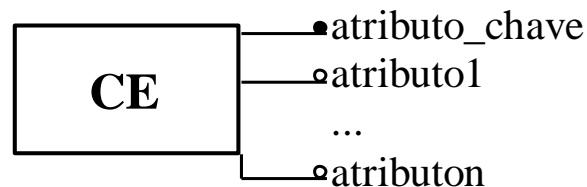
- Atributos de CEs e de CRs:
 - Descrevem os CEs e os CRs;
 - Cada entidade do CE e cada relacionamento do CR terão um **valor** para cada um de seus atributos;
 - Esses valores são os dados armazenados no BD;
 - Os atributos dos CEs são obrigatórios, os dos CRs são opcionais.



Projeto de Bancos de Dados

Projeto Conceitual – ME-R

- Atributos de CEs e de CRs:
 - **Chave de um CE:** conjunto de atributos cujos valores identifiquem univocamente cada entidade no CE;
 - Outras possíveis chaves não são indicadas no esquema de dados, pois não são contempladas pelo ME-R;
 - **Não existe a representação de chave em um CR.**

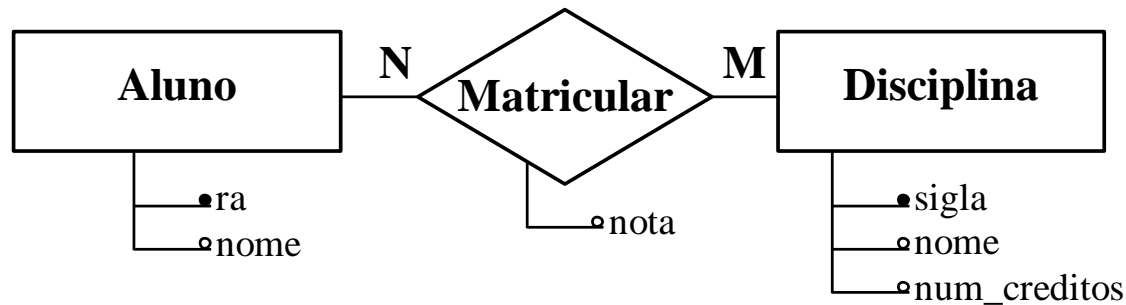


Projeto de Bancos de Dados

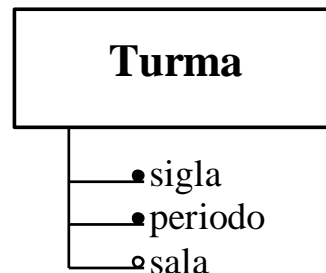
Projeto Conceitual – ME-R

- Atributos de CEs e de CRs:

- Exemplos:



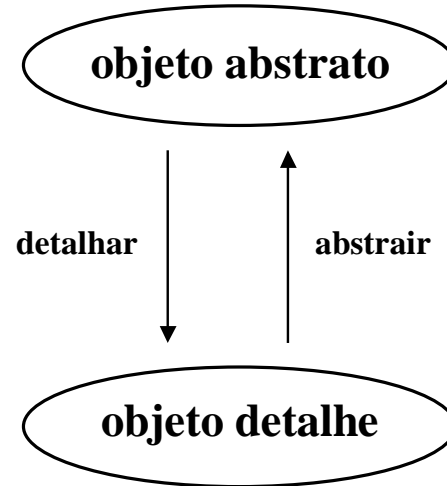
Chave composta:



Projeto de Bancos de Dados

Projeto Conceitual – ME-R Estendido

- Abstrações em modelos de dados:

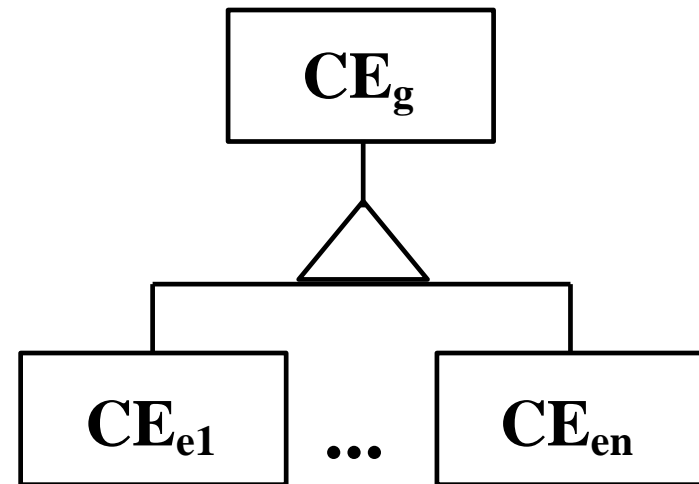
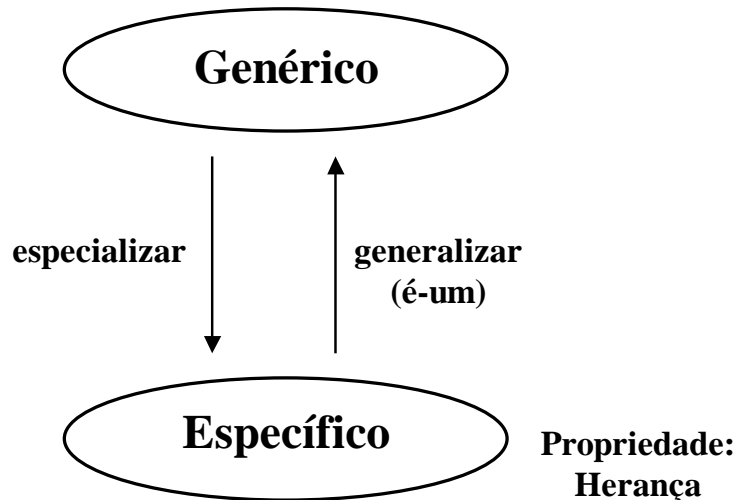


- Abstrações no ME-R Estendido:
 - Generalização;
 - Agregação.

Projeto de Bancos de Dados

Projeto Conceitual – ME-R Estendido

- Abstração de Generalização:
 - consiste em abstrair as características comuns contidas em vários CEs e generalizá-las em um único CE genérico (CE_g).

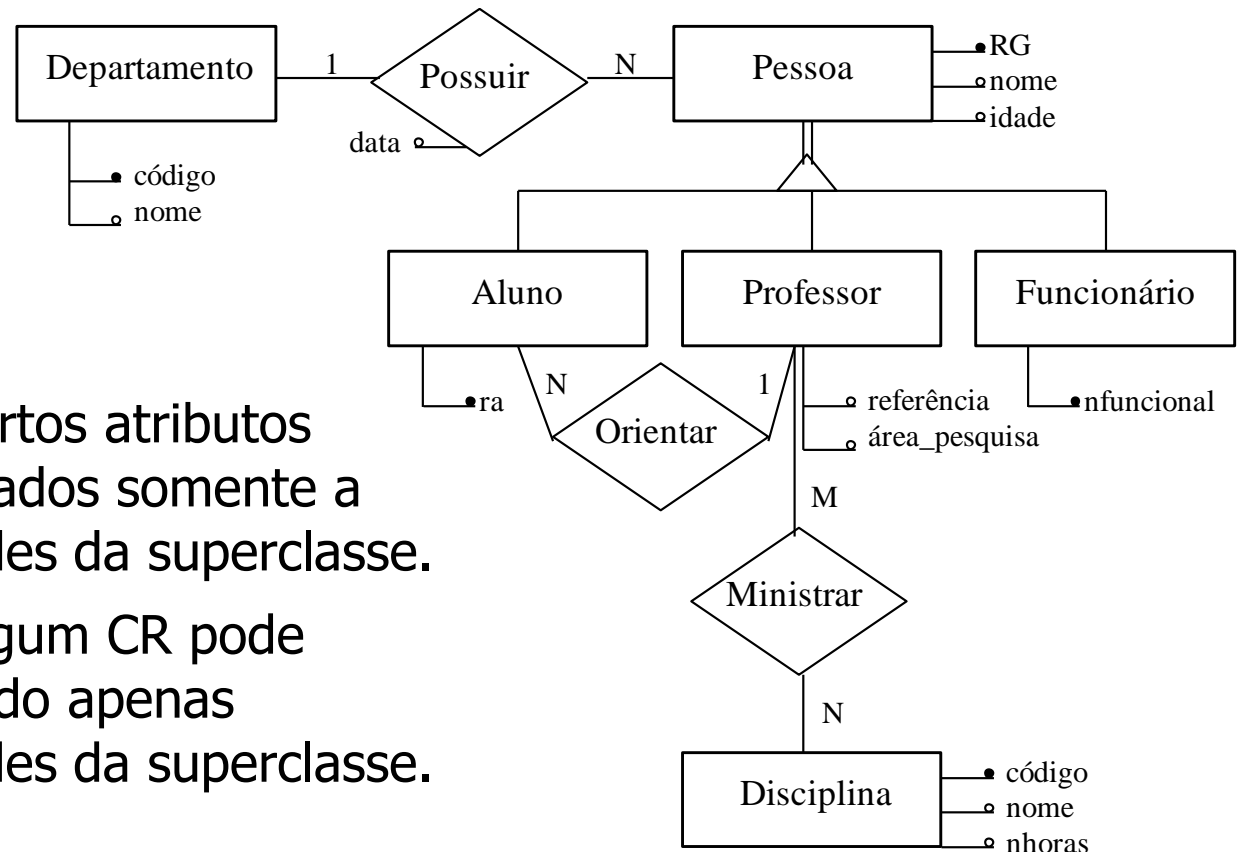


Projeto de Bancos de Dados

Projeto Conceitual – ME-R Estendido

Abstração de Generalização:

Exemplo:



Quando usar?

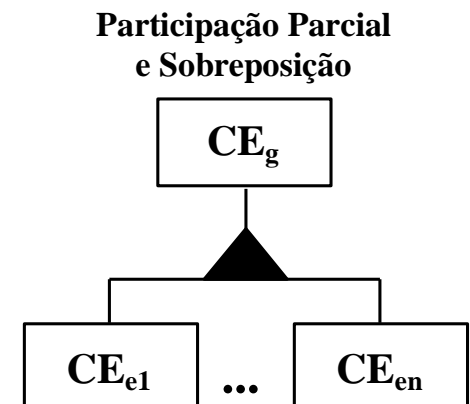
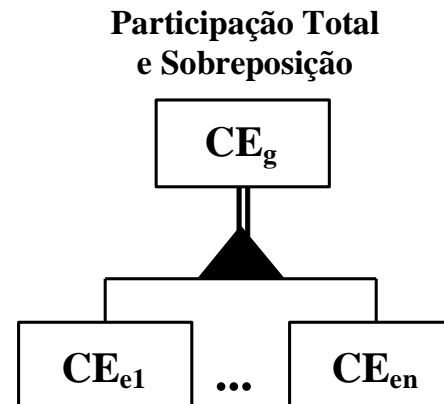
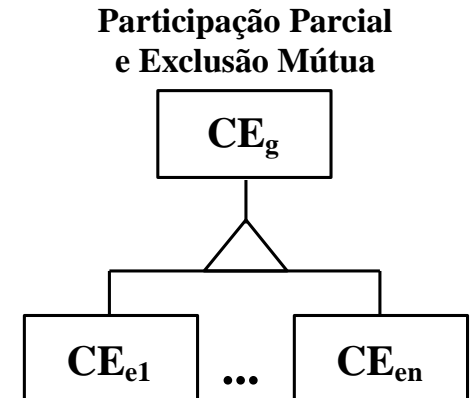
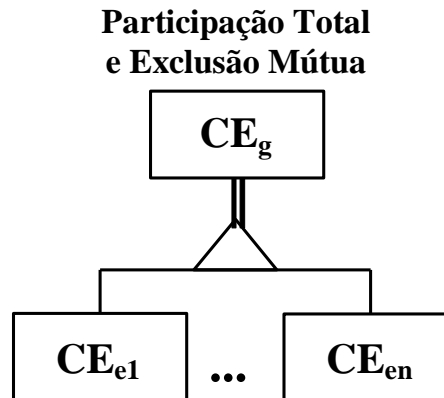
- 1o. Caso:** quando certos atributos podem ser aplicados somente a algumas entidades da superclasse.
- 2o. Caso:** quando algum CR pode estar relacionando apenas algumas entidades da superclasse.

Projeto de Bancos de Dados

Projeto Conceitual – ME-R Estendido

■ Abstração de Generalização:

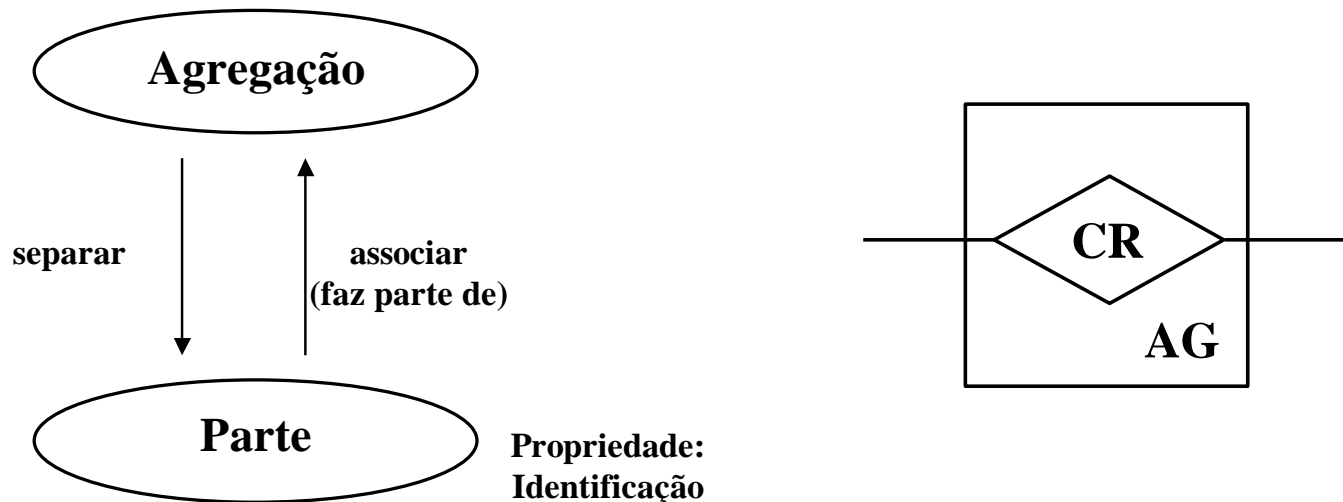
■ Representações:



Projeto de Bancos de Dados

Projeto Conceitual – ME-R Estendido

- **Abstração de Agregação:**
 - consiste em associar CEs cujas entidades estejam relacionadas através de um CR, gerando um objeto agregado AG, que pode ter atributos próprios.



Projeto de Bancos de Dados

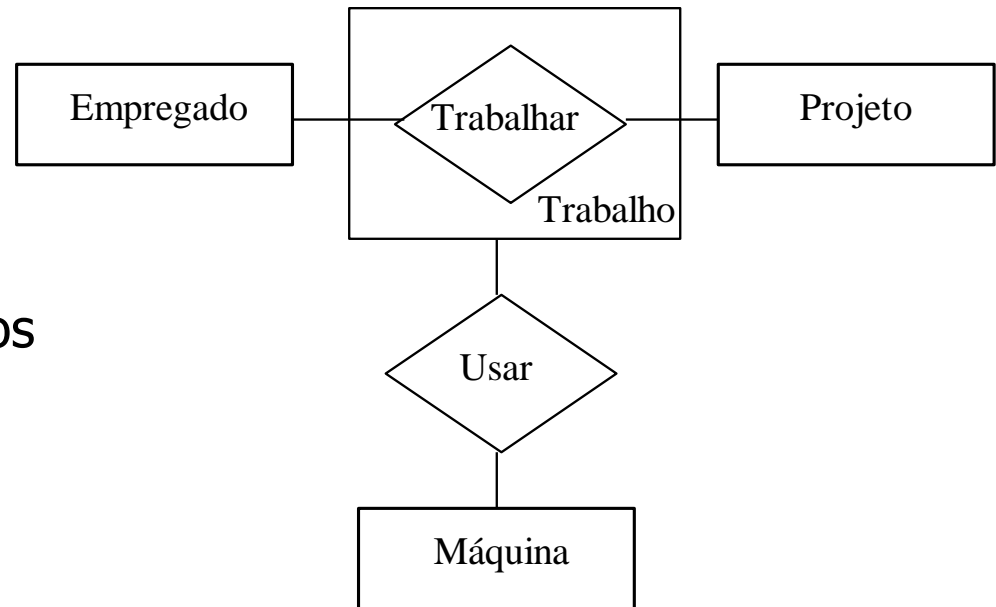
Projeto Conceitual – ME-R Estendido

Abstração de Agregação:

- Exemplos:

Quando usar?

1o. Caso: quando necessita-se representar relacionamentos entre relacionamentos.



Projeto de Bancos de Dados

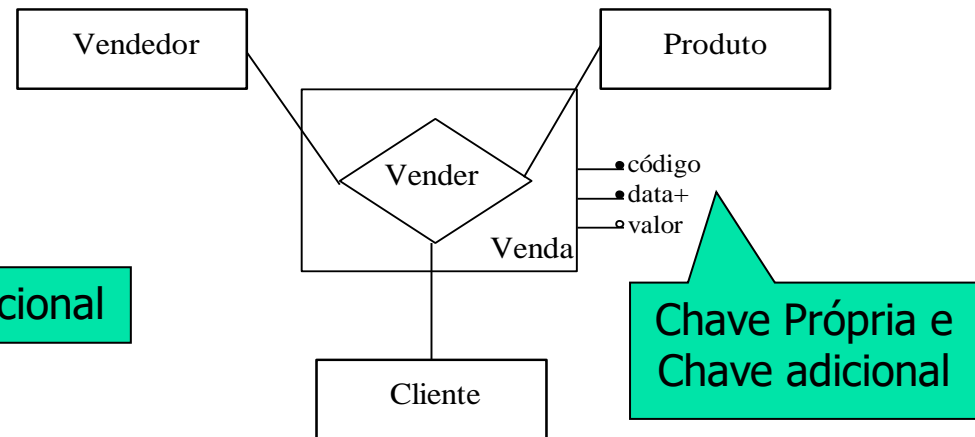
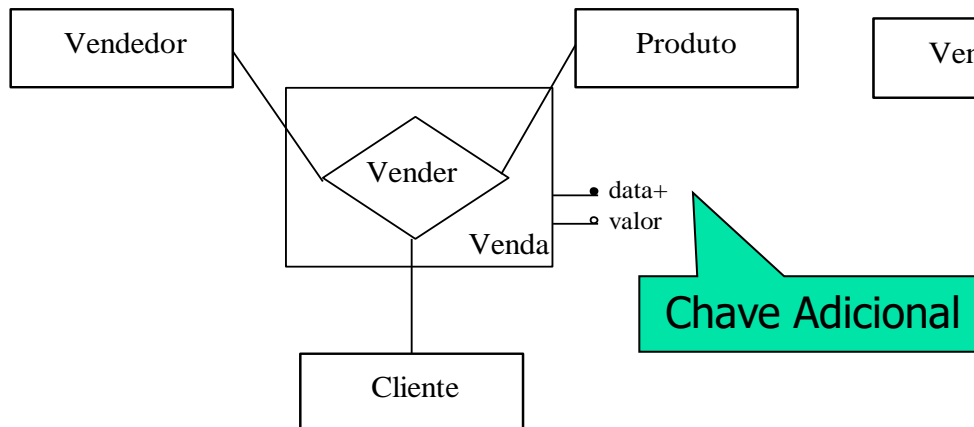
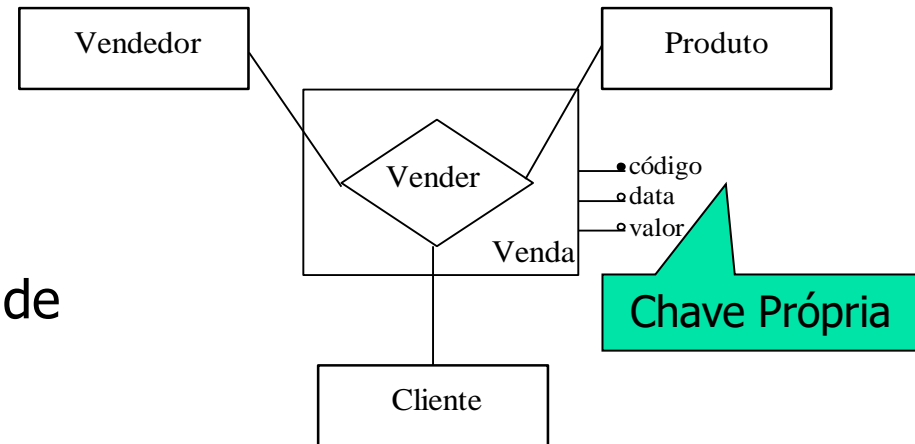
Projeto Conceitual – ME-R Estendido

Abstração de Agregação:

■ Exemplos:

Quando usar?

2o. Caso: quando é necessário identificar cada relacionamento de um CR por meio de uma chave.





Projeto de Bancos de Dados

Projeto Lógico

- Considera o tipo do SGBD a ser utilizado na implementação, mas não o fabricante;
- Mapeamento do esquema conceitual de dados gerando o **esquema lógico de dados**;
- Modelo Lógico:
 - Modelo Relacional.

Projeto de Bancos de Dados

Projeto Lógico – Modelo Relacional

- O Modelo Relacional possui 2 construtores semânticos:

- Relações;
- Atributos.

- Exemplo:

Empregado = {NomeEmp, NumEmp, Salario, Depto}

Informalmente:
Relação -> Tabela;
Atributos -> Colunas;
Tuplas -> Linhas
ou Registros.

Empregado

Nome da relação

atributos			
NomeEmp	NumEmp	Salario	Depto
Josué	123.456.789	R\$ 750.00	21
Silva	375.758.395	R\$ 480.00	25
Reis	475.858.595	R\$ 680.00	23

tuplas

Projeto de Bancos de Dados

Projeto Lógico – Modelo Relacional

■ Chaves de Relações:

- **Chave:** subconjunto dos atributos de uma relação cujos valores são unívocos para cada tupla da relação e do qual não se pode retirar nenhum atributo e ainda preservar-se a propriedade de identificação unívoca;
- **Chave primária:** é a chave de uma relação que tem a maior frequência de acesso. As outras chaves são chamadas de **chaves candidatas** (ou **secundárias**).

- Exemplo:

Pessoa = {Nome, CPE, RG, Idade, Telefone}

chave primária

chave secundária

Projeto de Bancos de Dados

Projeto Lógico – Modelo Relacional

■ Chaves de Relações:

- **Chave estrangeira:** faz o vínculo (relacionamento) entre duas relações por meio de valores de atributos dessas relações. Exemplo:

Esquema Lógico de Dados:

Departamento = {NumDepto, NomeDepto, Ramal}

Empregado = {NumEmp, NomeEmp, Salário, Depto}

Depto – chave estrangeira referenciando Departamento

chave estrangeira

Instâncias das relações:

$r(\text{Departamento}) = \{ \langle 21, \text{Pessoal}, 142 \rangle, \langle 23, \text{Financeiro}, 144 \rangle, \langle 25, \text{Compras}, 156 \rangle \}$

$r(\text{Empregado}) = \{ \langle 1234, \text{Josué}, 750.00, 23 \rangle, \langle 4567, \text{Silva}, 480.00, 25 \rangle, \langle 8910, \text{Reis}, 680.00, 21 \rangle, \langle 1112, \text{Santos}, 250.00, 25 \rangle \}$

Projeto de Bancos de Dados

Projeto Lógico – Modelo Relacional

- **Restrições de Integridade:** regras a respeito dos valores que podem ser armazenados nas relações. São três:
 - **Restrições de Integridade de Chave:** os valores das chaves de uma relação devem ser únicos para todas as tuplas da relação;
 - **Restrições de Integridade de Entidade:** a chave primária de uma relação não pode ser nula em nenhuma tupla dessa relação;
 - **Restrições de Integridade Referencial:** garantia do conceito de chave estrangeira.

Projeto de Bancos de Dados

Projeto Lógico – Modelo Relacional

Exemplos de violações de Restrições de Integridade:

■ Esquema Lógico de Dados:

Departamento = {NumDepto, NomeDepto, Ramal}

Empregado = {NumEmp, NomeEmp, Salário, Depto}

Depto – chave estrangeira referenciando Departamento

■ Instâncias das relações:

Violação de Integridade de Entidade

$r(\text{Departamento}) = \{ \langle \text{null}, \text{Pessoal}, 142 \rangle, \langle 23, \text{Financeiro}, 144 \rangle, \langle 25, \text{Compras}, 156 \rangle \}$

Violação de Integridade de Chave

$r(\text{Empregado}) = \{ \langle 1234, \text{Josué}, 750.00, 23 \rangle, \langle 4567, \text{Silva}, 480.00, \text{null} \rangle, \langle 1234, \text{Reis}, 680.00, 21 \rangle, \langle 1112, \text{Santos}, 250.00, 23 \rangle \}$

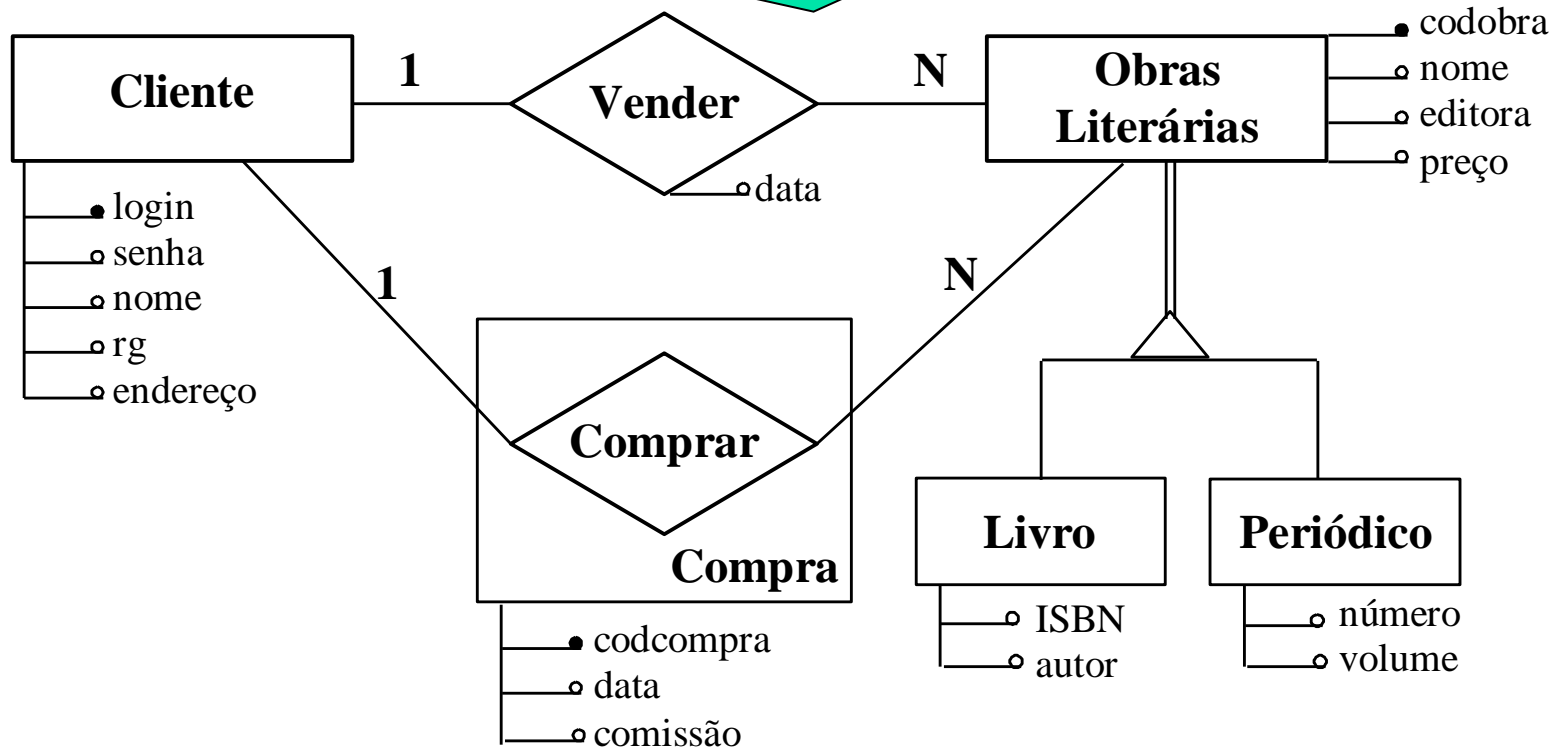
Violação de Integridade Referencial

Projeto Projeto

Mapeamento

Exemplo:

cliente = {login, senha, nome, rg, endereço}
obras_literárias = {codobra, nome, editora, preço, tipo, login_vendedor, data}
login_vendedor – chave estrangeira referenciando cliente
livro = {codobra, ISBN, autor}
codobra – chave estrangeira referenciando obras_literárias
periódico = {codobra, número, volume}
codobra – chave estrangeira referenciando obras_literárias
compra = {codcompra, data, comissão, login_comprador, codobra}
login_comprador – chave estrangeira referenciando cliente
codobra – chave estrangeira referenciando obras_literárias





Projeto de Bancos de Dados

Projeto Físico

- Implementa as definições do esquema lógico de dados, escolhendo estruturas de armazenamento e acesso específicas do SGBD a ser utilizado na implementação;
- Gera o **esquema físico de dados**;
- Usa a DDL (*Data Definition Language*) do SGBD específico da implementação.



A Linguagem SQL – *Structured Query Language*

- Linguagem padrão dos SGBDs Relacionais;
- Padronizada inicialmente pela ANSI (*American National Standards Institute*) e posteriormente também pela ISO (*International Standards Organization*):
 - Em 1986 – primeiro padrão ANSI – **SQL-86** (posteriormente adotado também pela ISO em 1987);
 - **SQL-89** e **SQL-92** – atualizações da SQL-86;
 - Em 1999 – **SQL:1999** ou **SQL3** – padrão para os BDs Objeto-Relacionais;
 - Em 2003 – **SQL:2003** – introduz características relacionadas a linguagem XML (eXtensible Markup Language);
 - Em 2006 – **SQL:2006** – define meios de utilização de SQL em conjunto com XML;
 - Versões posteriores: **SQL:2008**, **SQL:2011**, **SQL:2016**



A Linguagem SQL – *Structured Query Language*

- Divide-se em três componentes principais:
 - **Linguagem de Definição de Dados (*Data Definition Language – DDL*)**: permite a definição da estrutura do BD e a criação, alteração e eliminação dos objetos do BD (tabelas, índices, visões, regras, *stored procedures*, *triggers*);
 - **Linguagem de Manipulação de Dados (*Data Manipulation Language – DML*)**: suporta a manipulação dos dados armazenados nas tabelas do BD – inserção, consulta, eliminação e alteração;
 - **Linguagem de Controle de Dados (*Data Control Language – DCL*)**: envolve comandos para o controle de privilégios de acesso aos dados, segurança e administração do BD (cadastramento de usuários/grupos, atribuição de permissões, *backup* e *recovery*, configurações, etc).



Bibliografia

ELMASRI, R.; NAVATHE, S. B., Fundamentals of database systems. 7 ed., Pearson, 2016.

SILBERSCHATZ, A. KORTH, H. F. e SUDARSHAN, S.; Sistema de banco de dados. 5 ed., Campus - Elsevier, 2006.

DATE, C. J., Introdução a sistemas de bancos de dados. 8 ed., Campus - Elsevier, 2005.