

[MAC0439] Laboratório de Bancos de Dados
Aula 3
Modelo Entidade-Relacionamento Estendido

Kelly Rosa Braghetto

DCC-IME-USP

19 de agosto de 2015

Modelo Entidade-Relacionamento Estendido

Contexto (década de 1980)

- ▶ Desejo: projeto de bancos de dados que refletisse mais precisamente as restrições de dados
- ▶ Bancos de dados para sistemas de software mais complexos (como CAD e GIS)
- ▶ Necessidade de conceitos adicionais (+ abstrações!) para a *modelagem semântica de dados*
- ▶ Novos modelos criados em áreas como as de *representação do conhecimento* (IA) e *modelagem de objetos* (ES)
- ▶ Resultado: **modelo ER estendido** (ou **EER**)

Conceitos do modelo EER

- ▶ Todos os conceitos do modelo ER
- ▶ **Subclasse / Superclasse**
 - ▶ **Especialização / Generalização**
 - ▶ Herança de atributo e relacionamento
- ▶ **Categoria (ou Tipo de União)**

EER – Subclasses e superclasses

- ▶ **Classe** – conjunto ou coleção de entidades; isso inclui qualquer construção do EER que agrupe as entidades, como os tipos de entidade, subclasses, superclasses e categorias
- ▶ Uma **subclasse** S é uma classe cujas entidades devem sempre ser um subconjunto das entidades de outra classe, chamada de **superclasse** do relacionamento classe/subclasse, indicado como C/S . Portanto, em C/S , temos que $S \subseteq C$
- ▶ Uma entidade pode ser membro de várias subclasses
- ▶ Uma entidade membro de uma subclasse herda todos os atributos e relacionamentos da sua superclasse
- ▶ Uma subclasse por si só também é considerada um tipo de entidade

EER – especialização / generalização

- ▶ **Especialização** – é o processo de definir um **conjunto de subclasses** de um tipo de entidade, chamado de **superclasse** da especialização.

Especialização = refinamento conceitual

- ▶ **Generalização** – é o processo inverso, de suprimir as diferenças entre vários tipos de entidade (subclasses) e identificar suas características comuns, generalizando-as em uma única superclasse.

Generalização = síntese conceitual

O relacionamento entre uma subclasse e sua superclasse é chamado de “**É-UM**”.

EER – especialização/generalização

- ▶ Uma **especialização** $Z = \{S_1, S_2, \dots, S_n\}$ é um conjunto de subclasses que têm a mesma superclasse Z , isto é, Z/S_i , para todo i em $\{1, 2, \dots, n\}$.
- ▶ Z é chamada de **superclasse da especialização**, ou de **generalização das subclasses** $\{S_1, S_2, \dots, S_n\}$.
- ▶ Uma especialização Z é **total** sempre que: $\bigcup_{i=1}^n S_i = Z$.
Do contrário, Z é **parcial**.
- ▶ Z é **disjunta** sempre que tivermos $S_i \cap S_j = \emptyset$ para $i \neq j$.
Do contrário, Z é **sobreposta**.

Especialização – exemplo

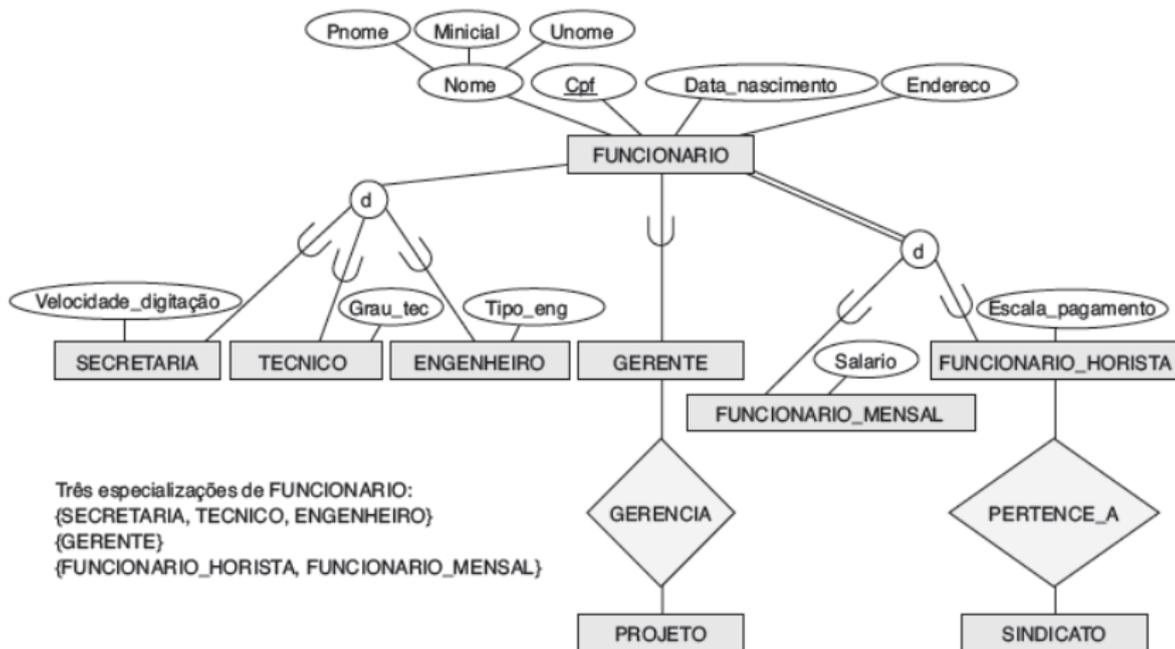
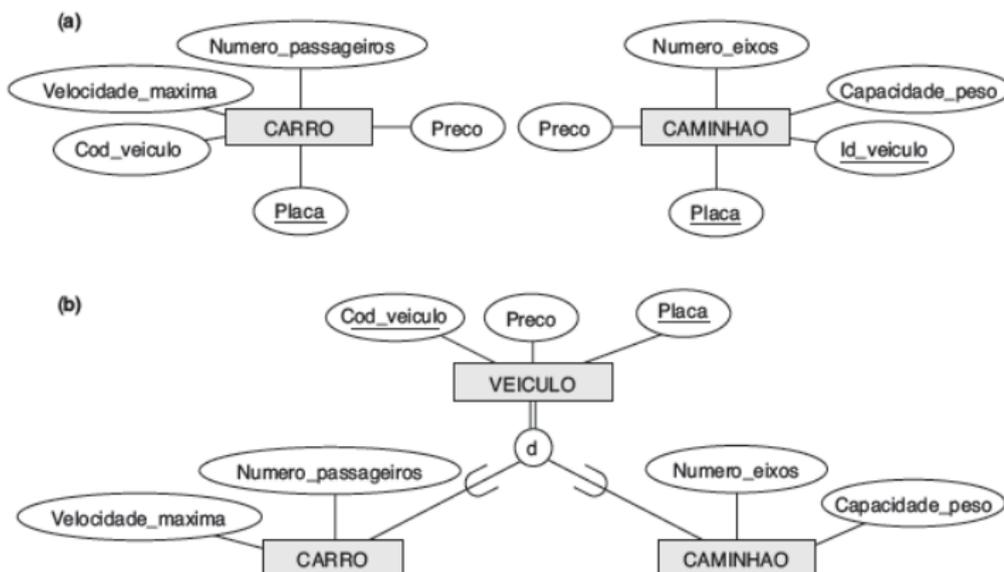


Diagram EER para representar subclasses e especialização

Generalização – exemplo



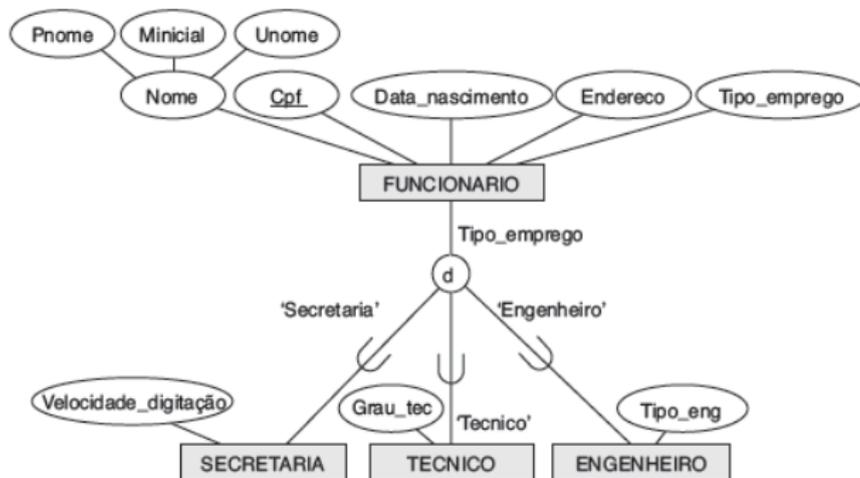
(a) Tipos de entidade CARRO e CAMINHAO.

(b) Generalizando esses tipos em VEICULO.

EER – especialização/generalização

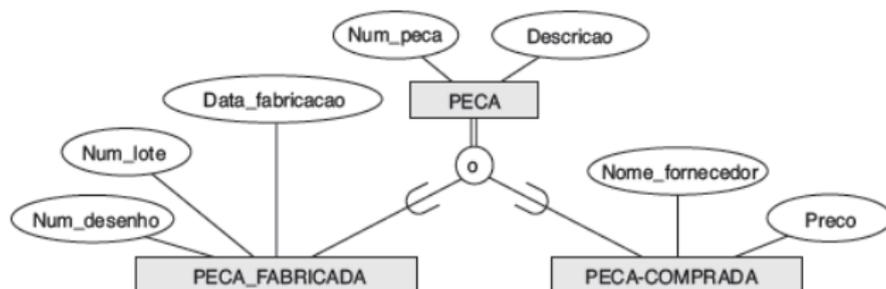
- ▶ O conjunto de subclasses que forma uma especialização é definido com base em algumas características de **distinção das entidades** da superclasse.
- ▶ Uma subclasse S de C é **definida por predicado** se um predicado p nos atributos de C for usado para especificar quais entidades de C são membros de S , isto é, $S = C[p]$, em que $C[p]$ é o conjunto de entidades de C que satisfazem p .
Um subclasse que não é definida por predicado é dita **definida por usuário**.
- ▶ Uma especialização Z (ou generalização G) é **definida por atributo** se um predicado ($A = c_i$), no qual A é um atributo de Z e c_i é um valor constante do domínio de A , for usado para especificar os membros de cada subclasse S_i em Z .
Se $c_i \neq c_j$, para $i \neq j$ e A é um atributo monovalorado, então a especialização será disjunta.

Especialização – exemplo



Especialização definida por atributo sobre Tipo_emprego

Especialização – exemplo



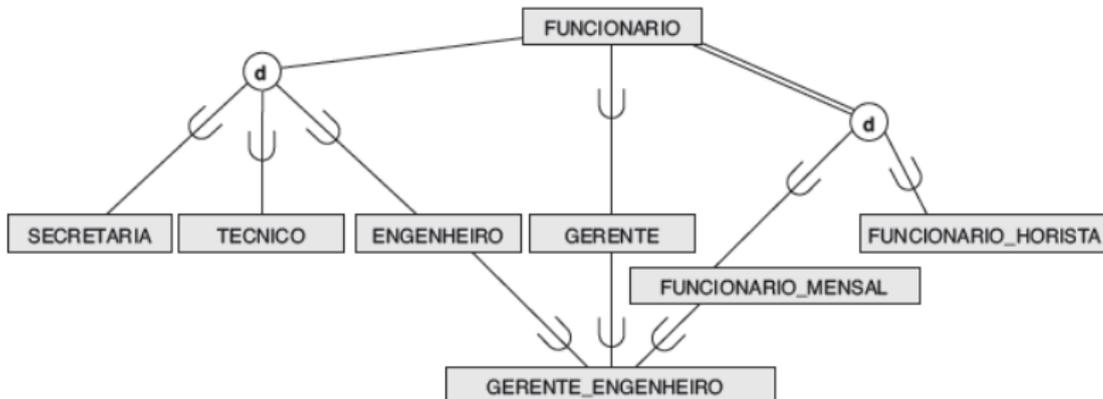
Especialização sobreposta

Abstração de especialização/generalização

Hierarquias no modelo EER

- ▶ Uma subclasse pode participar de mais de um relacionamento classe/subclasse, ou seja, ela pode ter mais de uma superclasse (= herança múltipla).
- ▶ Uma subclasse pode ter suas próprias subclasses, formando, assim, uma hierarquia ou reticulado de especializações.

Especialização – exemplo de reticulado



--

Um reticulado de especialização com a subclasse compartilhada **GERENTE_ENGENHEIRO**

Abstração de especialização/generalização

Razões para se usar especializações no modelo EER

1. quando certos atributos podem ser usados somente em algumas das entidades da superclasse. Uma subclasse é definida de modo a agrupar as entidades para as quais esses atributos se aplicam;
2. quando apenas as entidades que são membros de uma subclasse podem participar de algum tipo de relacionamento.

EER – categoria ou tipo de união

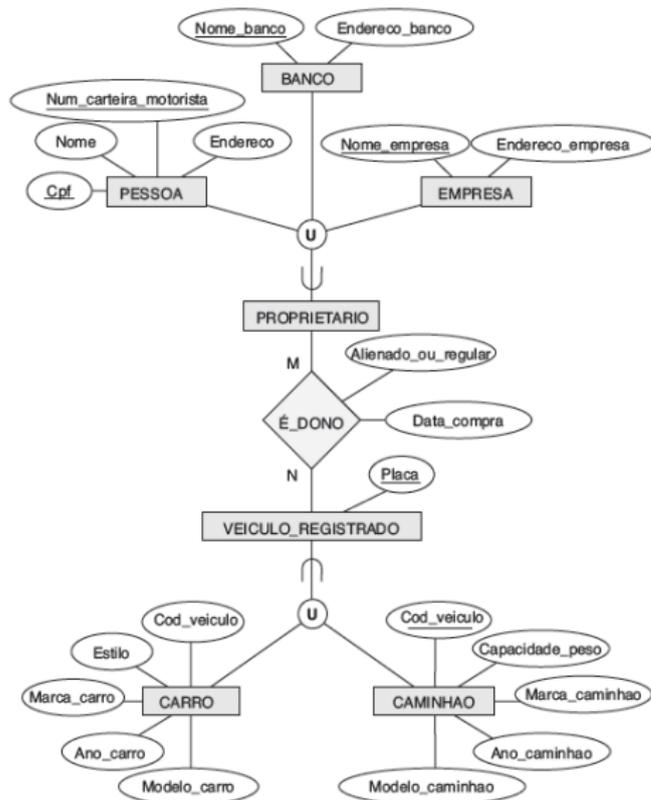
- ▶ Uma **categoria** T é formalmente definida como uma classe que é um subconjunto da união de n superclasses definidas D_1, D_2, \dots, D_n , $n > 1$, e é especificada como:
$$T \subseteq (D_1 \cup D_2 \cup \dots \cup D_n)$$
- ▶ Um predicado p_i nos atributos de D_i pode ser usado para especificar os membros de cada D_i que são membros de T . Se um predicado for especificado em todo D_i , temos:
$$T = (D_1[p_1] \cup D_2[p_2] \cup \dots \cup D_n[p_n])$$

Uma **categoria** pode ser vista como uma composição lógica, em que as partes têm uma existência independente e podem fazer parte de mais de uma composição.

EER – categoria

- ▶ Uma categoria pode ser **total** ou **parcial**
- ▶ Em uma categoria total, $T = (D_1 \cup D_2 \cup \dots \cup D_n)$, ou seja, T controla a **união de todas as entidades** em suas superclasses.
- ▶ Em uma categoria parcial, T pode controlar um **subconjunto da união**.

EER – categorização



Duas categorias:
 PROPRIETARIO e
 VEICULO_REGISTRADO

Uma discussão sobre abstrações de dados

Abstrações de dados

Abstrações usadas tanto na modelagem conceitual de dados quanto na área de **representação do conhecimento (RC)**, de IA:

- ▶ classificação/instanciação
- ▶ identificação
- ▶ especialização/generalização
- ▶ agregação/associação

Abstração de classificação

Classificação – é o **particionamento** de um conjunto de objetos em **classes de objetos**, de modo a agrupar os objetos que possuem características em comum.

- ▶ Os objetos de uma classe compartilham os mesmos tipos de atributos, relacionamentos e restrições.
- ▶ **Instanciação** – é o inverso da classificação e se refere à geração de objetos específicos de uma classe. Uma instância está ligada à sua classe por meio de um relacionamento “**É-UM-MEMBRO-DE**”.

Abstração de classificação

No modelo EER:

- ▶ Entidades são classificadas em **tipos de entidades** (de acordo com o seus atributos e relacionamentos básicos)
- ▶ Entidades também podem ser classificadas em **subclasses** e **categorias**, com base em similaridades ou diferenças entre elas
- ▶ As instâncias de relacionamentos são classificadas em **tipos de relacionamento**

⇒ No modelo de objetos, a classificação se refere à separação em **classes**; no relacional, em **relações**.

Abstração de classificação

Representação de objetos individuais na modelagem:

- ▶ Diagramas UML e esquemas RC possibilitam a “exibição” de objetos individuais (instâncias de classes)
- ▶ Diagramas EER não apresentam instâncias

Propriedades (= atributos) de classe:

- ▶ Nos modelos UML e nos esquemas RC as propriedades podem estar associados tanto a objetos individuais quanto a classes
- ▶ Esquemas EER não têm propriedades de classe

Abstração de classificação

Esquemas RC (ex: OWL, RDF):

- ▶ Permitem múltiplos esquemas de classificação
- ▶ Uma classe pode ser modelada como uma *instância de outra classe (metaclass)*

Em modelos UML:

- ▶ As classes podem ser de diferentes tipos:
abstratas (que não podem ser instanciadas), *concretas*,
genéricas (templates)

O modelo EER não é capaz de representar **explicitamente** esses tipos de classificação.

Abstração de identificação

Identificação é o processo de tornar os objetos e classes exclusivamente identificáveis por meio de um **identificador**.

No modelo EER:

- ▶ cada classe no esquema (que pode ser tipo de entidade, subclasse, superclasse, uma categoria ou um tipo de relacionamento) precisa ter um nome distinto
- ▶ os nomes dos atributos em uma mesma classe precisam ser distintos
- ▶ os valores dos atributos-chave distinguem as entidades de um tipo em particular

Abstração de especialização / generalização

- ▶ **Especialização** – é o processo de classificar uma classe de objetos em subclasses mais especializadas.
Especialização = refinamento conceitual
- ▶ **Generalização** – é o processo inverso, de generalizar diversas classes em uma classe abstrata de nível mais alto que inclua os objetos de todas essas classes.
Generalização = síntese conceitual

Essas abstrações são implementadas de forma bastante equivalente nos modelos EER, UML e representação de conhecimento.

Abstração de agregação

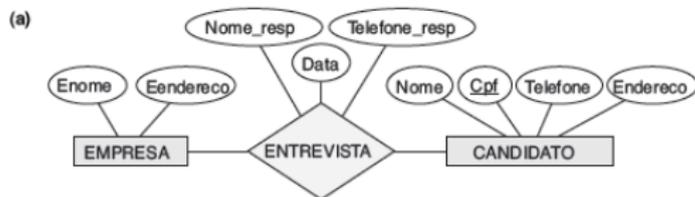
A agregação é um conceito de abstração para a construção de objetos compostos a partir de seus objetos componentes.

Casos em que esse conceito aparece no modelo EER:

1. quando agregamos os valores dos atributos de um objeto para formar o objeto total
2. quando representamos um relacionamento de agregação como um relacionamento comum
3. quando combinamos os objetos relacionados por meio de uma instância de relacionamento em particular em um **objeto agregado de alto nível**. Esse caso não pode ser representado de forma explícita no modelo EER.

Obs.: A UML é capaz de representar os três tipos de agregação.

Abstração de agregação - Exemplos



O tipo de relacionamento ENTREVISTA

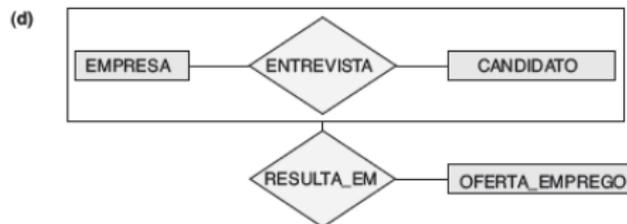


Inclusão de OFERTA_EMPREGO em relacionamento ternário (**incorreto!**)

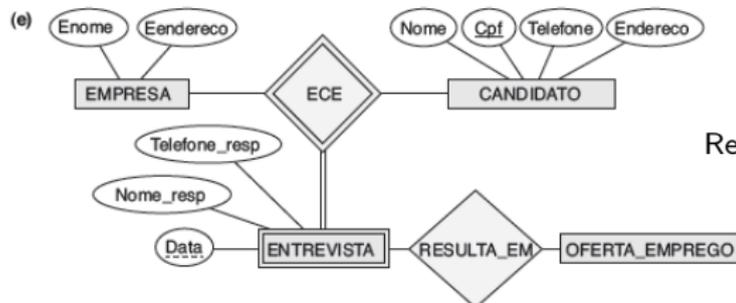


Relacionamento participando de outro relacionamento (**incorreto!**)

Abstração de agregação - Exemplos



Usando agregação e um objeto composto (geralmente não permitido em ER)



Representação correta em ER

Abstração de agregação

Um objeto agregado de alto nível faz-se especialmente necessário quando:

- ▶ ele mesmo está associado a um outro objeto por meio de um relacionamento
- ▶ ele possui atributos que não são comuns a todas as instâncias do seu relacionamento gerador

O relacionamento entre os objetos primitivos e seu objeto agregado é chamado de **É-UMA-PARTE-DE**.

Abstração de associação

A **associação** é usada para associar objetos de várias *classes independentes*.

- ▶ No modelo EER (geralmente), ela é semelhante ao segundo tipo de uso da agregação – representada por **tipos de relacionamentos**
- ▶ Na UML, ela é representada por **associações**

O relacionamento entre os objetos primitivos e seu objeto agregado é chamado de **É-ASSOCIADO-A**.

Abstração de agregação × abstração de associação

Existe uma distinção estrutural entre a agregação e associação:

- ▶ quando uma instância de associação é excluída, os objetos participantes podem continuar existir
- ▶ quando um objeto de agregação composto de outros objetos – ex: um CARRO composto por MOTOR, CHASSI e PNEUS – é excluído, todos os seus objetos componentes devem ser excluídos também.

Esta distinção estrutural não pode ser diretamente representada no modelo EER.

Ferramentas de apoio à modelagem conceitual (gratuitas)

Geram automaticamente código em SQL a partir dos modelos

- ▶ EERCASE (UFPE/CIn) –
<https://sites.google.com/a/cin.ufpe.br/eercase/>
 - ▶ Modelos EE e EER com a notação usada no livro do Elmasri/Navathe
- ▶ PostgreSQL Database Modeler (PgModeler) –
<http://www.pgmodeler.com.br/>
 - ▶ Usa um tipo de modelo que é uma mistura de modelagem conceitual e lógica
 - ▶ Específico para BDs mantidos no PostgreSQL
- ▶ MySQL Workbench –
<http://www.mysql.com/products/workbench/>
 - ▶ Ampara a modelagem e a administração dos BDs
 - ▶ Específica para BDs mantidos no MySQL

Ferramentas de apoio à modelagem conceitual (gratuitas)

Geram automaticamente código em SQL a partir dos modelos

- ▶ Open ModelSphere – <http://www.modelsphere.com/>
 - ▶ Não é específica para BDs (também permite a criação de modelos UML e modelos para processos de negócio)

Puramente desenho

- ▶ Diagram Editor (DIA) – <http://dia-installer.de/>
- ▶ draw.io – <https://www.draw.io/>
 - ▶ Ferramenta online

Referências Bibliográficas

- ▶ *Sistemas de Bancos de Dados* (6ª edição), Elmasri e Navathe. Pearson, 2010.
Capítulos 7 (Modelos ER) e 8 (Modelo EER)
- ▶ *Sistemas de Gerenciamento de Banco de Dados* (3ª edição), Ramakrishnan e Gehrke, 2008.
Capítulo 2

Cenas dos próximos capítulos...

Projeto lógico de BDs:

- ▶ Modelo de dados relacional