

Banco de Dados I

Modelo Entidade e Relacionamento

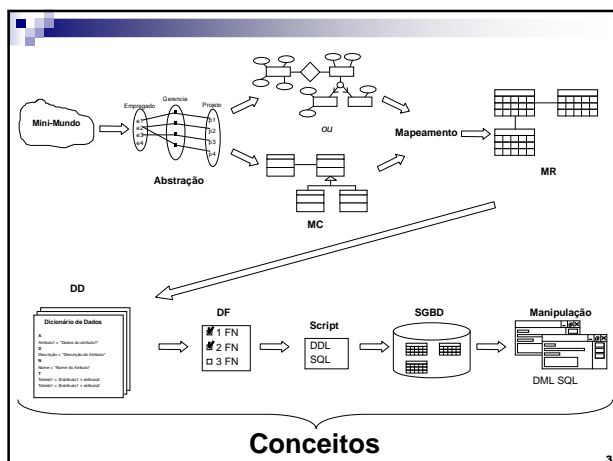
Osmar de Oliveira Braz Junior

1

Objetivos

- Conhecer as fases de um projeto de Banco de Dados.
- Conhecer o Modelo Entidade e Relacionamento.
- Modelar casos com o MER.
- Utilizar os conceitos estendido do MER.
- Utilizar a UML como notação alternativa para o MER.

2



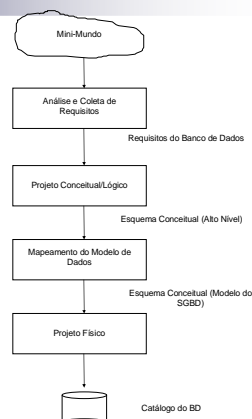
3

3. Modelagem de Dados Utilizando o Modelo Entidade Relacionamento

- O modelo **Entidade-Relacionamento** é um modelo de dados conceitual de alto nível, cujos conceitos foram projetados para estar o mais próximo possível da visão que o usuário tem dos dados, não se preocupando em representar como estes dados estarão realmente armazenados.
- O modelo **ER** é utilizado principalmente durante o processo de projeto de banco de dados.

4

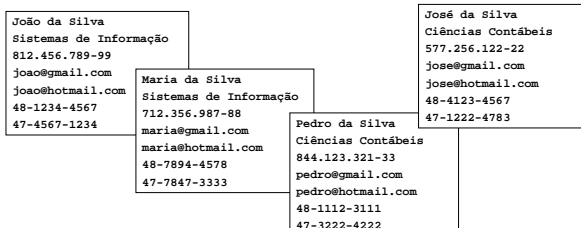
Fases do Projeto de um Banco de Dados



5

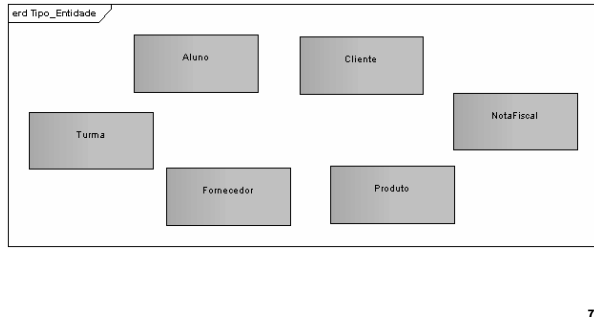
3.2. Entidades e Atributos

- O objeto básico tratado pelo modelo ER é a "**entidade**", que pode ser definida como um objeto do mundo real, concreto ou abstrato e que possui existência independente.



6

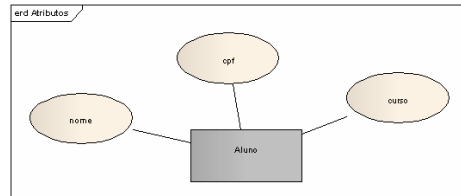
3.2. Entidades e Atributos



7

3.2. Entidades e Atributos

- Cada entidade possui um conjunto particular de propriedades que a descreve chamado **“atributos”**.



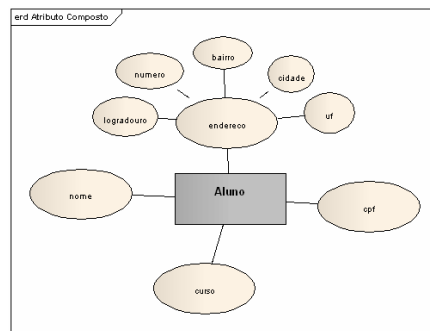
8

3.2. Entidades e Atributos

- Um atributo pode ser dividido em diversas sub-partes com significado independente entre si, recebendo o nome de **“atributo composto”**.
- Um atributo que não pode ser subdividido é chamado de **“atributo simples”** ou **“atômico”**.

9

3.2. Entidades e Atributos



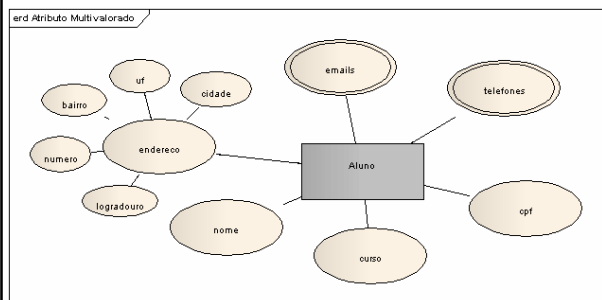
10

3.2. Entidades e Atributos

- Os **atributos** que podem assumir apenas um determinado valor em uma determinada instância é denominado **“atributo simplesmente valorado”**,
- Enquanto que um atributo que pode assumir diversos valores em uma mesma instância é denominado **“multivalorado”**.

11

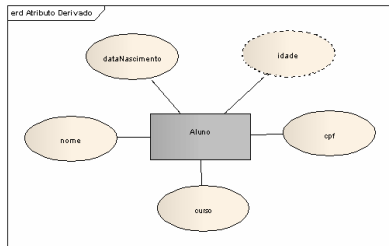
3.2. Entidades e Atributos



12

3.2. Entidades e Atributos

- Um atributo que é gerado a partir de outro atributo é chamado de “**atributo derivado**”.



13

3.3. Tipos de Entidade, Conjunto de Entidade

- Um banco de dados costuma conter **grupos de entidades** que são similares, possuindo os mesmos atributos, porém, cada entidade com seus próprios valores para cada atributo. Este conjunto(ou coleção) de entidades similares definem um “**tipo de entidade**”.
- A coleção de todas as entidades de um determinado tipo entidade no banco de dados em qualquer ponto do tempo é chamada de “**conjunto de entidades**”; geralmente nos referimos ao conjunto de entidades utilizando o mesmo nome do tipo de entidade

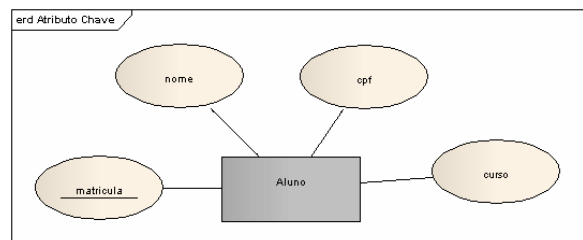
14

3.3. Atributo Chave, Conjunto de Valores

- Um tipo entidade possui um atributo cujos valores são distintos para cada entidade individual. Este atributo é chamado “**atributo chave**” e seus valores podem ser utilizados para identificar cada entidade de forma única.
 - pode ser formada pela composição de dois ou mais atributos.
 - pode também ter mais de um atributo chave.

15

3.3. Atributo Chave, Conjunto de Valores



16

3.3. Atributo Chave, Conjunto de Valores

- Cada atributo simples de um tipo entidade está associado com um “**conjunto de valores**” denominado “**domínio**”, o qual especifica o conjunto de valores (ou domínio de valores) que podem ser designados para este determinado atributo para cada entidade.
- Ex.:
 - Sexo {M, F}
 - Estado_Civil {CASADO, SOLTEIRO, DIVORCIADO, VIÚVO, DESQUITADO}

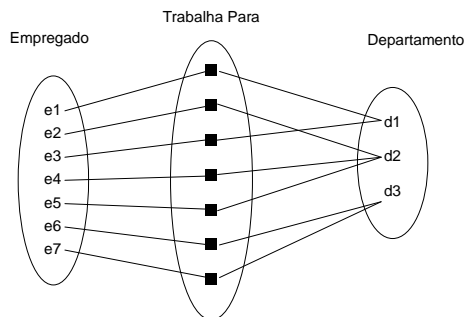
17

3.4. Tipos e Instâncias de Relacionamento

- Um “**tipo relacionamento**” R entre n entidades E_1, E_2, \dots, E_n , é um conjunto de associações entre entidades deste tipo.
- Informalmente falando, cada instância de relacionamento r_1 em R é uma associação de entidades, onde a associação inclui exatamente uma entidade de cada tipo entidade participante no tipo relacionamento.

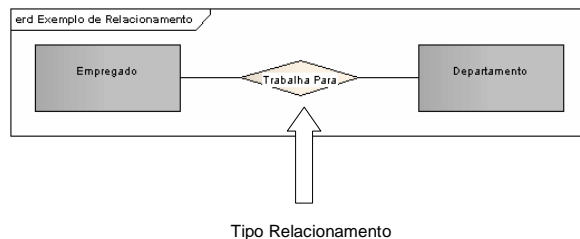
18

Exemplo de um Relacionamento



19

Exemplo de um Relacionamento



20

3.5. Grau de um Relacionamento

- O “**grau**” de um tipo relacionamento é o número de tipos entidade que participam do tipo relacionamento.

21

3.6. Outras Características de um Relacionamento

- 3.6.1. Relacionamentos como Atributos
 - Algumas vezes é conveniente pensar em um relacionamento como um **atributo**.
 - Podemos pensar departamento como sendo um atributo da entidade empregado, ou empregado, como um atributo multivalorado da entidade departamento. Se uma entidade não possuir existência muito bem definida, talvez seja mais interessante para a **coesividade** do modelo lógico que ela seja representada como um atributo

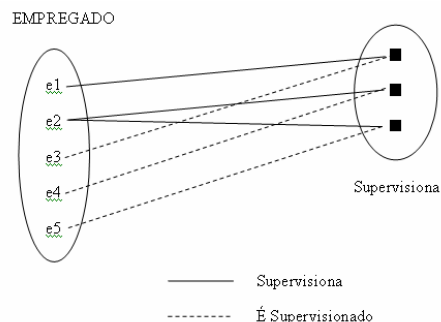
22

3.6.2. Nomes de Papéis e Relacionamentos Recursivos

- Cada tipo entidade que participa de um tipo relacionamento desempenha um **papel** particular no relacionamento. O nome do papel representa o papel que uma entidade de um tipo entidade participante desempenha no relacionamento

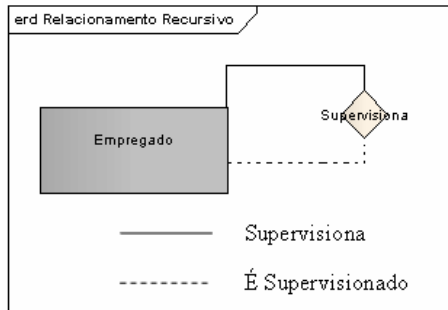
23

Um Relacionamento Recursivo



24

Um Relacionamento Recursivo



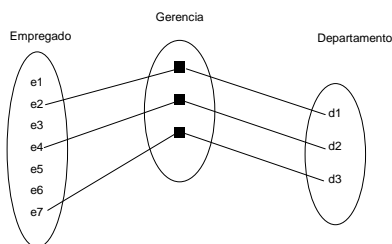
25

3.6.3. Restrições em Tipos Relacionamentos

- Geralmente, os tipos relacionamentos sofrem certas **restrições** que limitam as possíveis combinações das entidades participantes. Estas restrições são derivadas de **restrições impostas** pelo estado destas entidades no mini-mundo.

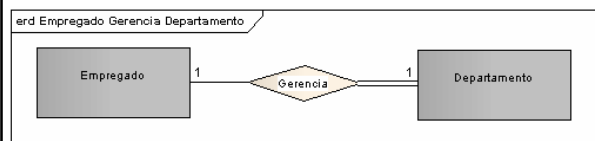
26

Relacionamento EMPREGADO gerencia DEPARTAMENTO



27

Relacionamento EMPREGADO gerencia DEPARTAMENTO



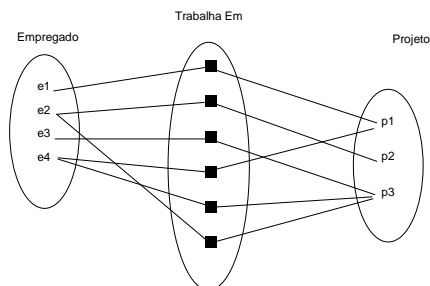
28

3.6.3. Cardinalidade

- Na figura anterior, temos a seguinte situação: um empregado pode gerenciar apenas um departamento, enquanto que um departamento, pode ser gerenciado por apenas um empregado. A este tipo de restrição, nós chamamos **cardinalidade**
- Indica o número de relacionamentos dos quais uma entidade pode participar.
- A cardinalidade pode ser: **1:1, 1:N, N:M**.
- Na figura anterior, a cardinalidade é 1:1, pois cada entidade empregado pode gerenciar apenas um departamento e um departamento pode ser gerenciado por apenas um empregado.

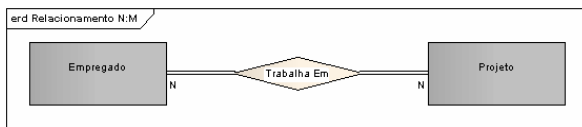
29

Relacionamento N:M



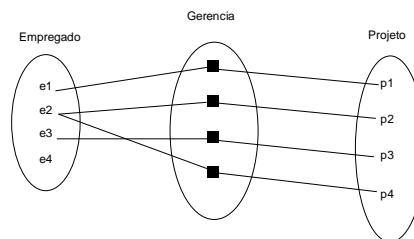
30

Relacionamento N:M



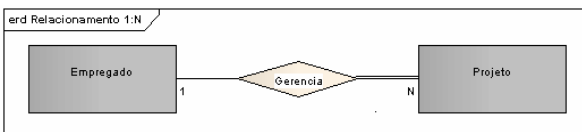
31

Relacionamento 1:N



32

Relacionamento 1:N



33

3.6.3. Participação

- Define a existência de uma entidade através do relacionamento, podendo ser **parcial** ou **total**.
- Na figura anterior a participação do empregado é **parcial** pois nem todo empregado gerencia um projeto, porém a participação do projeto neste relacionamento é **total** pois todo projeto precisa ser gerenciado por um empregado.
- Desta forma, **todas** as entidades do tipo entidade PROJETO precisam participar do relacionamento(GERENCIA), mas **nem todas** as entidade do tipo entidade EMPREGADO precisam participar do relacionamento.

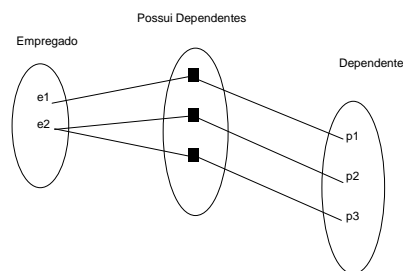
34

3.6.4. Tipos Entidades Fracas

- Alguns tipos entidade podem não ter um atributo chave por si só. Isto implica que não poderemos distinguir algumas entidades por que as combinações dos valores de seus atributos podem ser idênticas. Estes tipos entidade são chamados **entidades fracas**. As entidades deste tipo precisam estar relacionadas com uma entidade pertencente ao tipo entidade **proprietária**.

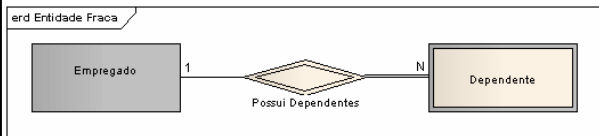
35

Relacionamento com uma Entidade Fraca (Dependente)



36

Relacionamento com uma Entidade Fraca (Dependente)



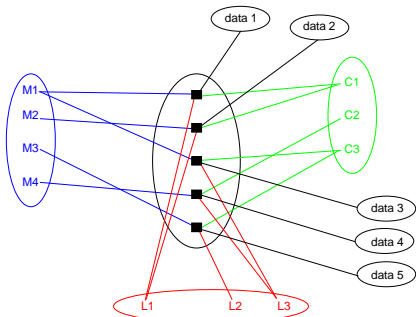
37

Relacionamento Ternário

- O **modelo entidade relacionamento** não se restringe apenas à relacionamentos binários. O número de entidades que participam de um tipo relacionamento é irrestrito e armazenam muito mais informações do que diversos relacionamentos binários.
- Considere o seguinte exemplo:
 - *Um motorista pode efetuar uma viagem para uma localidade dirigindo um determinado caminhão em uma determinada data.*

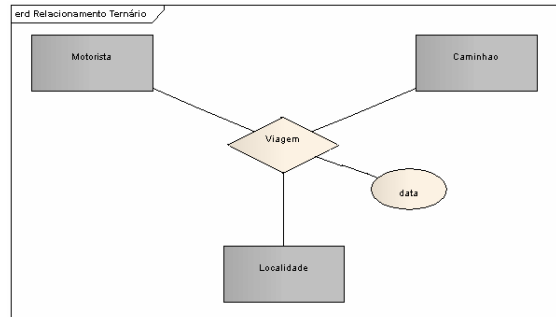
38

Relacionamento Ternário

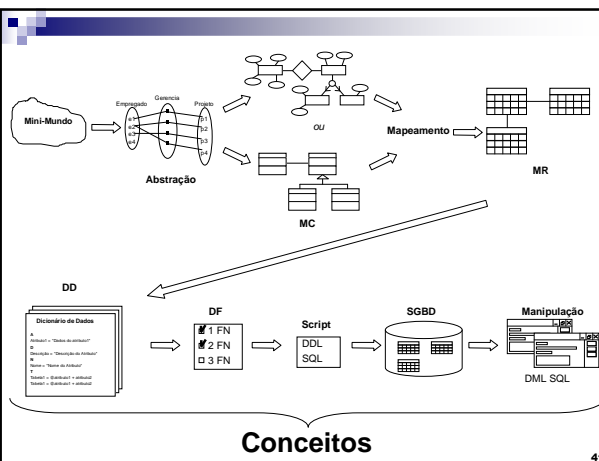


39

Relacionamento Ternário



40

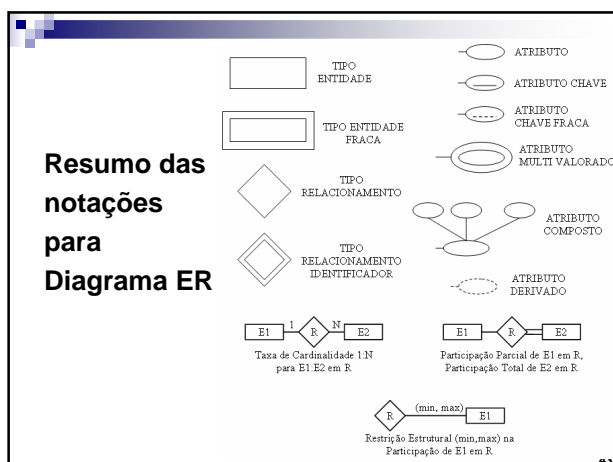


41

3.7. Diagrama Entidade Relacionamento

- O **diagrama Entidade Relacionamento (DER)** é composto por um conjunto de objetos gráficos que visa representar todos os objetos do modelo Entidade Relacionamento tais como entidades, atributos, atributos chaves, relacionamentos, restrições estruturais, etc.
- O diagrama **ER** fornece uma visão lógica do banco de dados, fornecendo um conceito mais generalizado de como estão **estruturados** os dados de um sistema.

42



3.7.1. Dicas para Construção de Diagrama Entidade Relacionamento

- A presença de um **substantivo** usualmente indica uma **entidade**.
- A presença de um **verbo** é uma forte indicação de um **relacionamento**
- Um **adjetivo**, que é uma qualidade, é uma forte indicação de um **atributo**
- Um **advérbio** temporal, qualificando o verbo, é uma indicação de um atributo do **relacionamento**

3.8. Modelo Entidade Relacionamento Extendido

- Os conceitos do modelo **Entidade Relacionamento** discutidos anteriormente são suficientes para representar logicamente a maioria das aplicações de banco de dados. Porém, com o surgimento de **novas aplicações**, surgiu também a necessidade de novas semânticas para a modelagem de informações mais **complexas**.

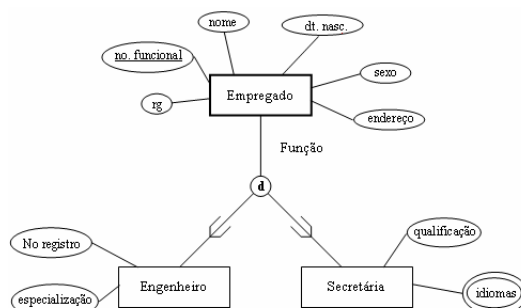
3.8. Modelo Entidade Relacionamento Extendido

- O modelo **ERE** engloba todos os conceitos do modelo **ER** mais os conceitos de **subclasse**, **superclasse**, **generalização** e **especialização** e o conceito de **herança de atributos**

3.8.1. Subclasses, Superclasses e Especializações

- O primeiro conceito do modelo **ERE** que será abordado é o de **subclasse** de um tipo entidade. Como visto anteriormente, um tipo entidade é utilizado para representar um conjunto de entidades do mesmo tipo. Em muitos casos, um tipo entidade possui diversos subgrupos adicionais de entidades que são significativas e precisam ser representadas explicitamente devido ao seu significado à aplicação de banco de dados.

Representação de Superclasse e Subclasses

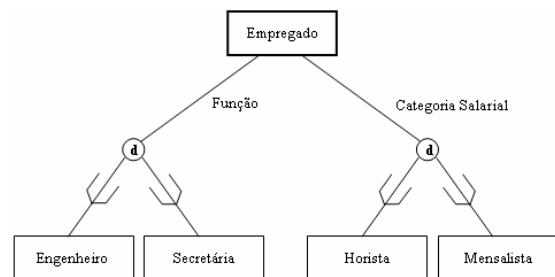


3.8.1. Especialização

- **Especialização** é o processo de definição de um conjunto de classes de um tipo entidade; este tipo entidade é chamado de superclasse da especialização. O conjunto de subclasses é formado baseado em alguma característica que distingue as entidades entre si.

49

Duas Especializações para Empregado: Função e Categoria Salarial



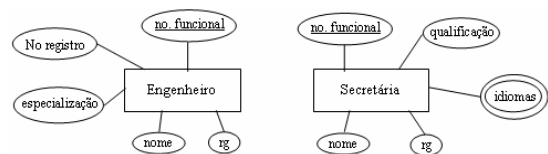
50

3.8.3. Generalização

- A **generalização** pode ser pensada como um processo de abstração reverso ao da **especialização**, no qual são suprimidas as diferenças entre diversos tipos entidades, identificando suas características comuns e generalizando estas entidades em uma superclasse.

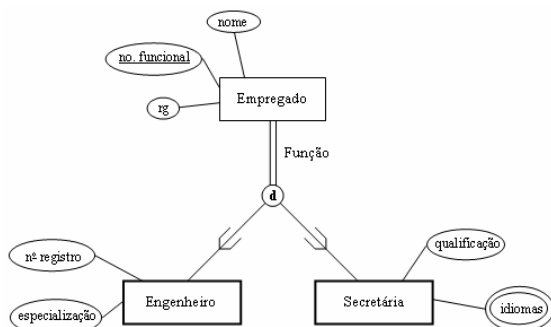
51

Tipos Entidades Engenheiro e Secretária



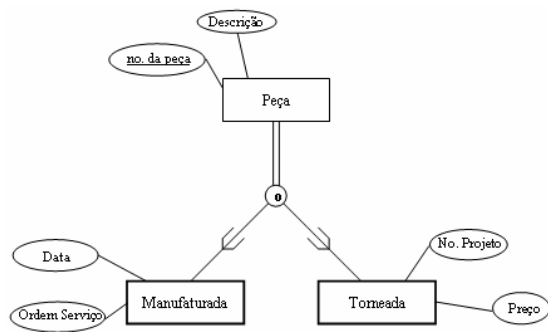
52

Generalização Empregado para os Tipos Entidades Engenheiro e Secretária com Disjunção



53

Generalização com overlap



54

Diferenças entre especialização e a generalização

- É importante destacar que existe diferença semântica entre a **especialização** e a **generalização**. Na **especialização**, podemos notar que a ligação entre a superclasse e as subclasses é feita através de um traço simples, indicando **participação parcial** por parte da superclasse.

55

Diferenças entre especialização e a generalização

- Na **generalização**, podemos notar que a ligação entre a superclasse e as subclasses é feita através de um traço duplo, indicando **participação total** por parte da superclasse.

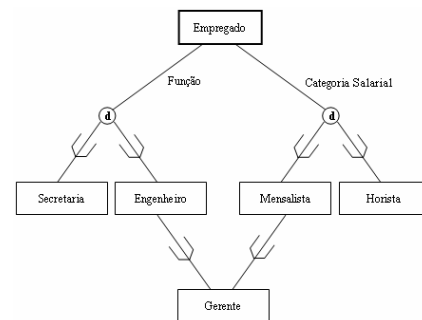
56

3.8.4. "Lattice" ou Múltipla Herança

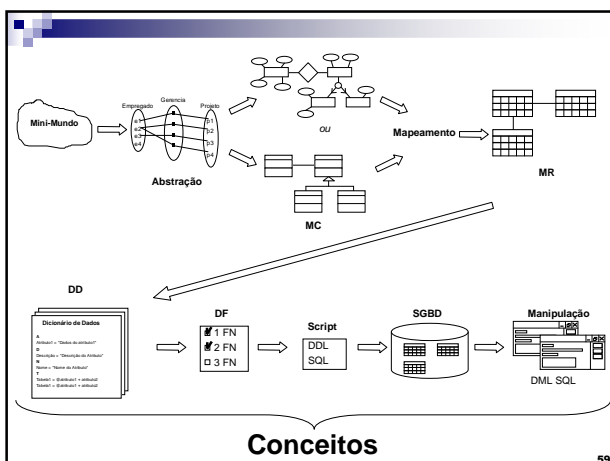
- Uma subclasse pode ser definida através de um "lattice", ou **múltipla herança**, ou seja, ela pode ter diversas superclasses, herdando características de todas.

57

Um "Lattice" com a Subclasse Gerente Compartilhada

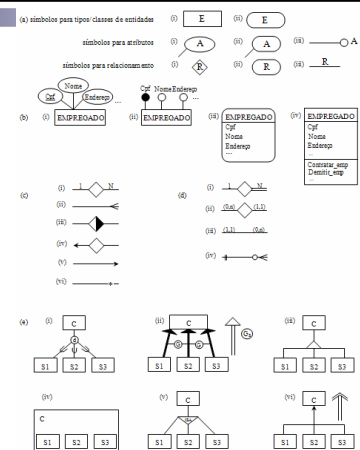


58



59

3.8.5 Notações Diagramáticas Alternativas Diagrama ER

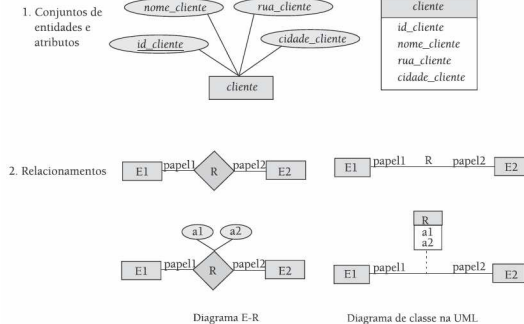


3.8.6. Utilizando Diagramas de Classes da UML

- **UML: Unified Modeling Language**
- A **UML** possui muitos componentes para modelar graficamente diferentes aspectos de um sistema de software inteiro
- O diagrama de classe **UML** corresponde ao diagrama ER, mas com algumas diferenças.

61

3.8.6. Utilizando Diagramas de Classes da UML



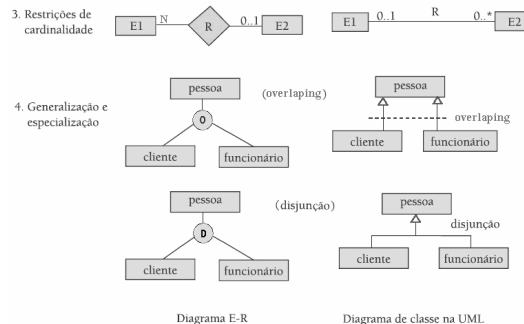
62

3.8.6. Utilizando Diagramas de Classes da UML

- Os conjuntos de entidades são mostrados como retângulos e, diferente do diagrama ER, os atributos são mostrados dentro do retângulo e não como elipses separadas.
- Os conjuntos de relacionamento binários são representados na UML simplesmente desenhando uma linha conectando os conjuntos de entidades. O nome do conjunto de relacionamento é escrito adjacente à linha.
- A função desempenhada por um conjunto de entidades em um conjunto de relacionamento também pode ser especificada escrevendo o nome da função na linha, adjacente ao conjunto de entidades.
- Alternativamente, o nome do conjunto de relacionamento pode ser escrito em um retângulo, juntamente com atributos do conjunto de relacionamento, e o retângulo é conectado usando uma linha pontilhada até a linha representando o conjunto de relacionamento.
- Os relacionamentos não binários são desenhados usando losangos, exatamente como nos diagramas ER

63

3.8.6. Utilizando Diagramas de Classes da UML



64

3.8.6. Utilizando Diagramas de Classes da UML

- As restrições de cardinalidade são especificadas na forma l..h, onde l indica o mínimo e h, o número máximo de relacionamentos em que um conjunto de entidades pode participar.
- Atenção: O posicionamento das restrições é exatamente o inverso do posicionamento das restrições nos diagramas ER.
- A restrição N(muitos) no lado E2 e a restrição 1 no lado E1 significam que cada entidade E1 pode participar em muitos relacionamentos; ou seja, o relacionamento é muitos-para-um de E2 para E1.
- Valores únicos como 1 ou N podem ser escritos nas bordas; o valor único 1 em uma borda é tratado como equivalente a 1..1, enquanto * é equivalente a 0..*.

65

Ferramenta Case

- Ferramentas **CASE** (do inglês **Computer-Aided Software Engineering**) é uma classificação que abrange todas ferramentas baseadas em computadores que auxiliam atividades de engenharia de software, desde análise de requisitos e modelagem até programação e testes.
- Podem ser consideradas como ferramentas automatizadas que tem como objetivo auxiliar o desenvolvedor de sistemas em uma ou várias etapas do ciclo de desenvolvimento de software.

http://pt.wikipedia.org/wiki/Ferramenta_CASE em 12/09/2012

66

Organização Modelos

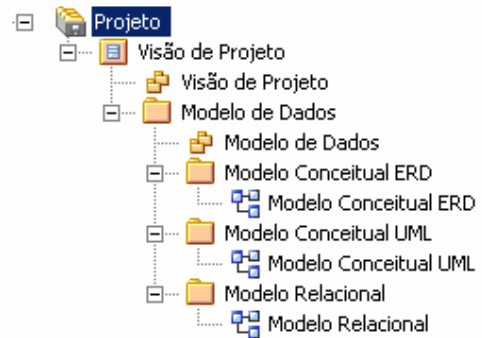
■ Visão de Projeto

□ Modelo de Dados

- Modelo Conceitual (ERD – Entity Relationship)
- Modelo Conceitual (UML – Diagrama de Classes)
- Modelo Relacional (UML – Diagrama de Classes Modificado)

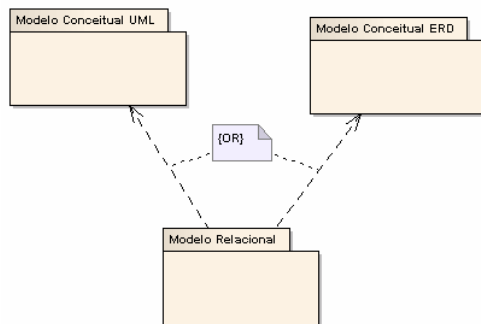
67

Organização Modelos



68

Organização Modelos



69

Bibliografia

- **Principal**
 - CHEN, P. *The Entity-Relationship Model - Toward a Unified View of Data*. TODS, 1976.
 - DATE, C. J. *Introdução a Sistemas de Banco de Dados*. 7. ed. Rio de Janeiro: Campus, 2000. 803 p.
 - ELMASRI, S. N.; NAVATHE, B. S. *Sistemas de Banco de Dados: Fundamentos e Aplicações*. 3. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2002. 837 p.
 - SILBERSCHATZ, A. ; KORTH, H.F. ; SUDARSHAN, S. *Sistema de Banco de Dados*. 5. ed. Rio de Janeiro: Elsevier, 2006.
- **Complementar**
 - BEZERRA, E. *Princípios de Análise e Projeto de Sistemas com UML*. 1. ed. Rio de Janeiro: Campus, 2002. 286 p.
 - CHIAVENATO, I. *Introdução à Teoria Geral da Administração*. 7. ed. Rio de Janeiro: Campus, 2004. 634 p.
 - DAUM, B. *Modelagem de Objetos de negócio com XML: abordagem com base em XML Schema*. Rio de Janeiro: Elsevier, 2004.
 - KROENKE, D.M. *Banco de Dados: Fundamentos, Projeto e Implementações*. 6. ed. Rio de Janeiro: Livros Técnicos e Científicos, 1998. 382 p.
 - OZSU, M.T.; VALDURIEZ, P. *Princípios de sistemas de banco de dados distribuídos*. 1. ed. Rio de Janeiro: Campus, 2001.
 - YOURDON, E. *Análise Estrutura Moderna*. 3. ed. Rio de Janeiro: Campus, 1992. 836 p.

70

Fim

71