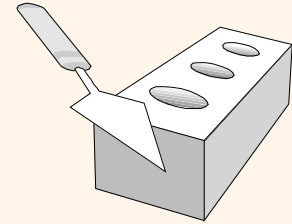


O Modelo Entidade-Relacionamento *MER*

Capítulo 2



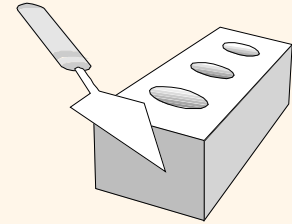
Etapas do Projeto de BD

1. Análise de Requisitos:

- Processo informal que envolve discussões entre grupos de usuários
- Entender quais dados serão armazenados no BD
- Quais aplicações tem que ser construídas
- Quais operações são mais freqüentes

2. Projeto Conceitual do BD:

- As informações colhidas na análise de requisitos são usadas para desenvolver uma descrição em alto nível dos dados a serem armazenados no BD (modelo ER)



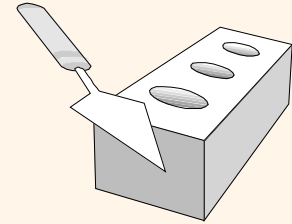
Etapas do Projeto de BD

3. Projeto Lógico do BD:

- É necessário escolher um SGBD para implementar nosso projeto de BD, e converter o projeto conceitual do BD em um esquema de BD do modelo de dados do SGBD

4. Refinamento do Esquema:

- Analisar as coleções de relações em nosso esquema de BD relacional para identificar potenciais problemas, e refiná-los. Esta fase pode ser guiada por algumas teorias como por exemplo a normalização



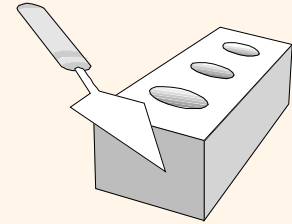
Etapas do Projeto de BD

5. Projeto Físico do BD:

- Identificar trabalhos que o BD tem que suportar e refinar o projeto de BD para garantir critérios de desempenho
- Envolve construção de índices para tabelas, etc

6. Projeto de Segurança:

- Identificar diferentes grupos de usuários e diferentes regras definidas para esses usuários
- Para cada regra e grupo de usuário, temos que identificar as partes do BD que devem estar habilitadas para acesso e as partes do BD que não devem ser permitido o acesso.

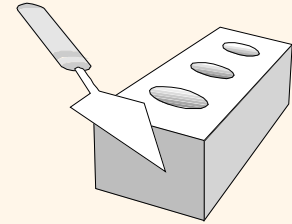


Etapas do Projeto de BD

- ❖ Fase de Ajuste:
 - Fase de refinamento;
 - Fase na qual as 6 etapas são repetidas até que o projeto atinja o nível de excelência desejado.

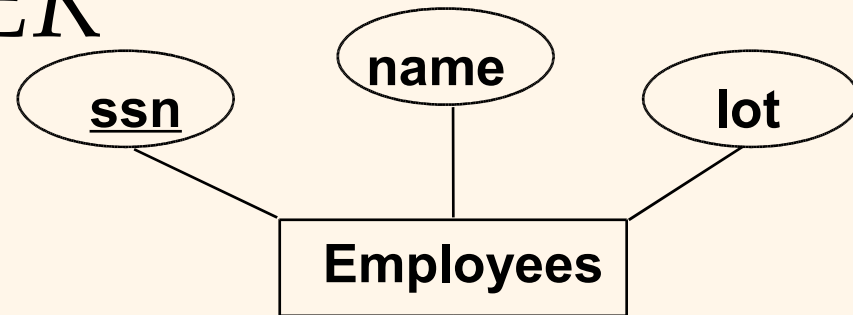
- ❖ O Modelo ER é mais relevante nos três primeiros passos: análise de requisitos, projeto conceitual e projeto lógico

Visão Geral do Projeto de BD

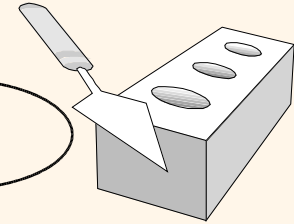


- Projeto Conceitual: (o modelo ER é usado nesta fase)
 - Quais são as entidades e os relacionamentos no empreendimento?
 - Que informações a respeito dessas entidades e desses relacionamentos deverão ser armazenados no BD?
 - Quais são as restrições de integridade ou regras de negócios que devem ser levadas em consideração?
 - Um esquema do BD pode ser representado por um diagrama de Entidade-Relacionamento.
 - Um diagrama de Entidade-Relacionamento pode ser mapeado para um esquema relacional.
 - Permite mover-nos de uma descrição informal sobre o que os usuários querem sobre o BD para uma descrição formal

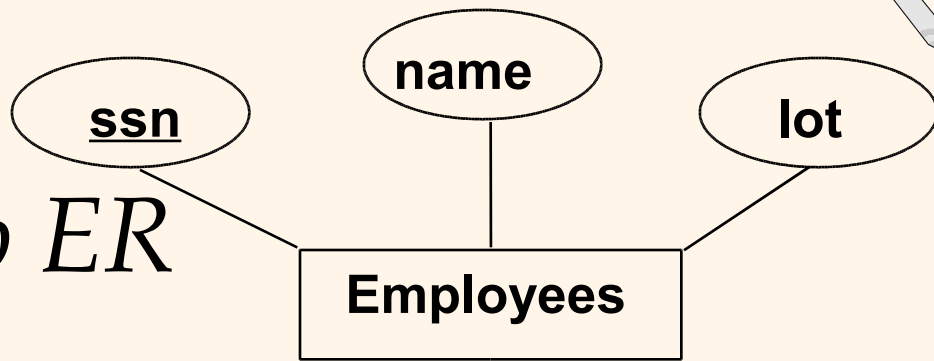
Modelo Básico ER



- **Entidade:** Objeto do mundo real distinguível de outros objetos. Uma entidade é descrita (em BD) usando um conjunto de atributos.
- **Conjunto-Entidade:** Uma coleção de entidades similares. Exemplo: todos os funcionários.
 - Todas as entidades em um conjunto-entidade têm o mesmo conjunto de atributos (até que seja considerado a hierarquia ISA)
 - Cada conjunto-entidade tem uma *chave* (conjunto minimal de atributos que identificam unicamente uma entidade).
 - Cada atributo tem um *domínio* (possíveis valores que o atributo pode assumir).



Modelo Básico ER



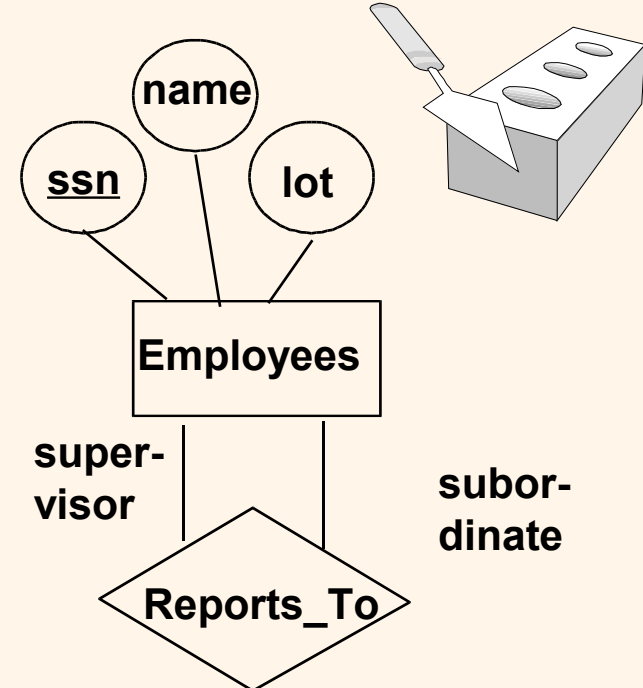
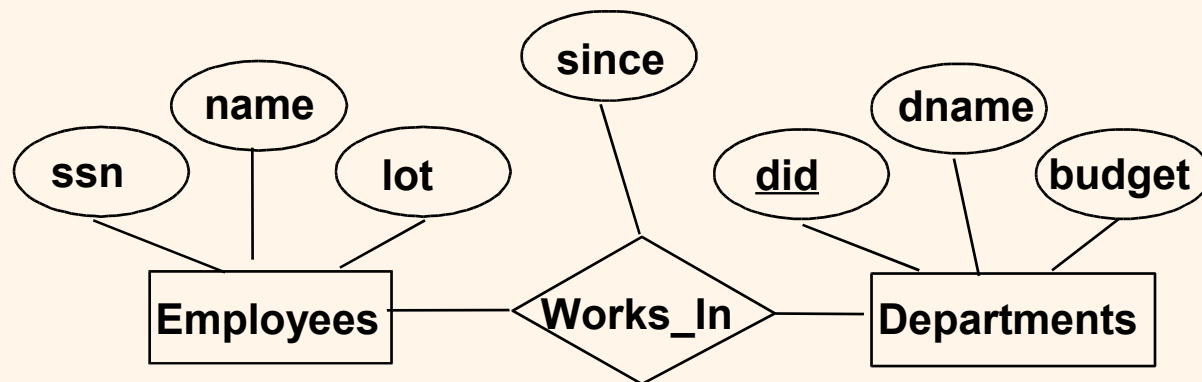
■ Chave

- **Candidata:** são chaves candidatas a serem chave primária
- **Primária:** chave selecionada para identificar a entidade
- **Estrangeira:** é quando o(s) atributo(s) de uma entidade é (são) a chave primária de outra entidade

■ Representação Gráfica

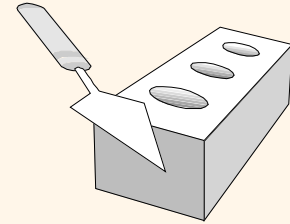


Modelo Básico ER

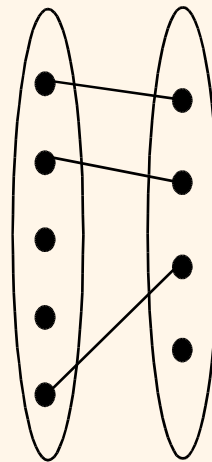
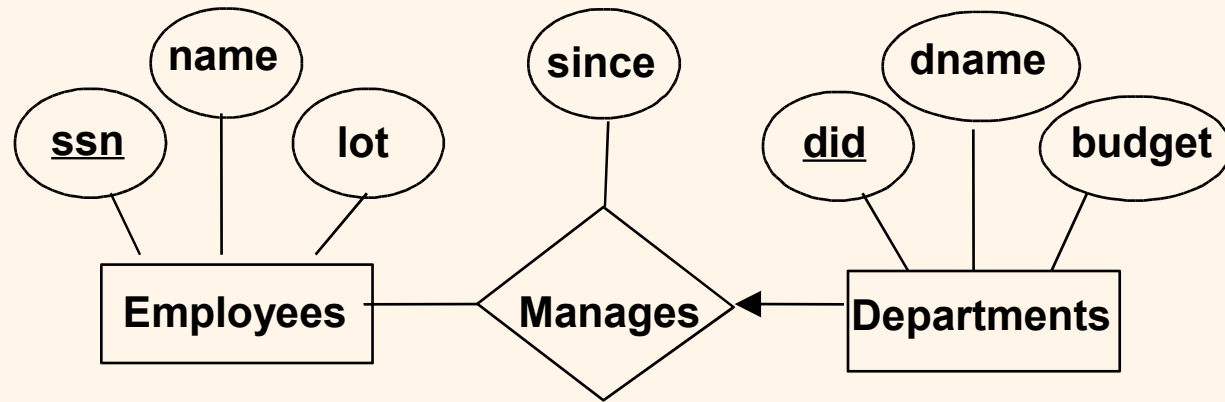


- ❖ **Relacionamento**: Associação entre duas ou mais entidades. Exemplo: Maria trabalha no departamento de Farmácia.
- ❖ **Conjunto-Relacionamento**: coleção de relacionamentos similares.
 - Um conjunto-relacionamento n-ário R relaciona n conjuntos de entidades $E_1 \dots E_n$; cada relacionamento em R envolve as entidades E_1, \dots, E_n
 - Um mesmo conjunto-entidade pode participar em diferentes conjuntos-relacionamentos ou em diferentes papéis no mesmo conjunto (por ex. papéis de supervisor e subordinado entre empregados)
- **Atributo Descritivo**: registram informação sobre o relacionamento

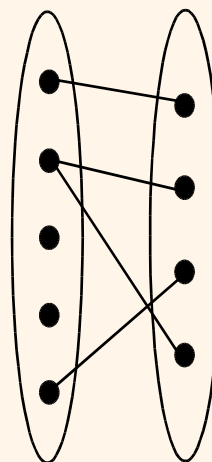
Restrição de Chave



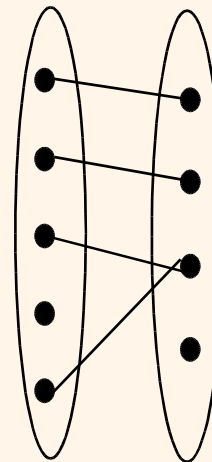
- ❖ Considere o conjunto-relacionamento “Works_In”: Um funcionário pode trabalhar em muitos departamentos; um departamento pode ter muitos funcionários.
- ❖ Em contraste, cada departamento tem somente um gerente, de acordo com a restrição de chave do conjunto-relacionamento “Manages”.



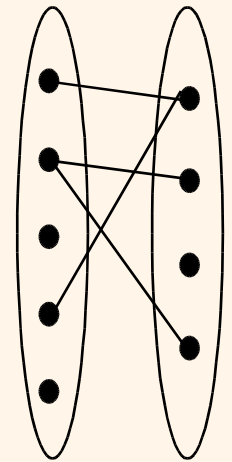
1-para-1



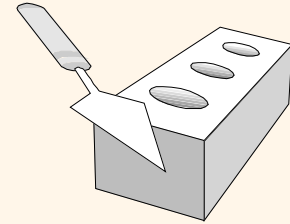
1-para
muitos



Muitos
para 1



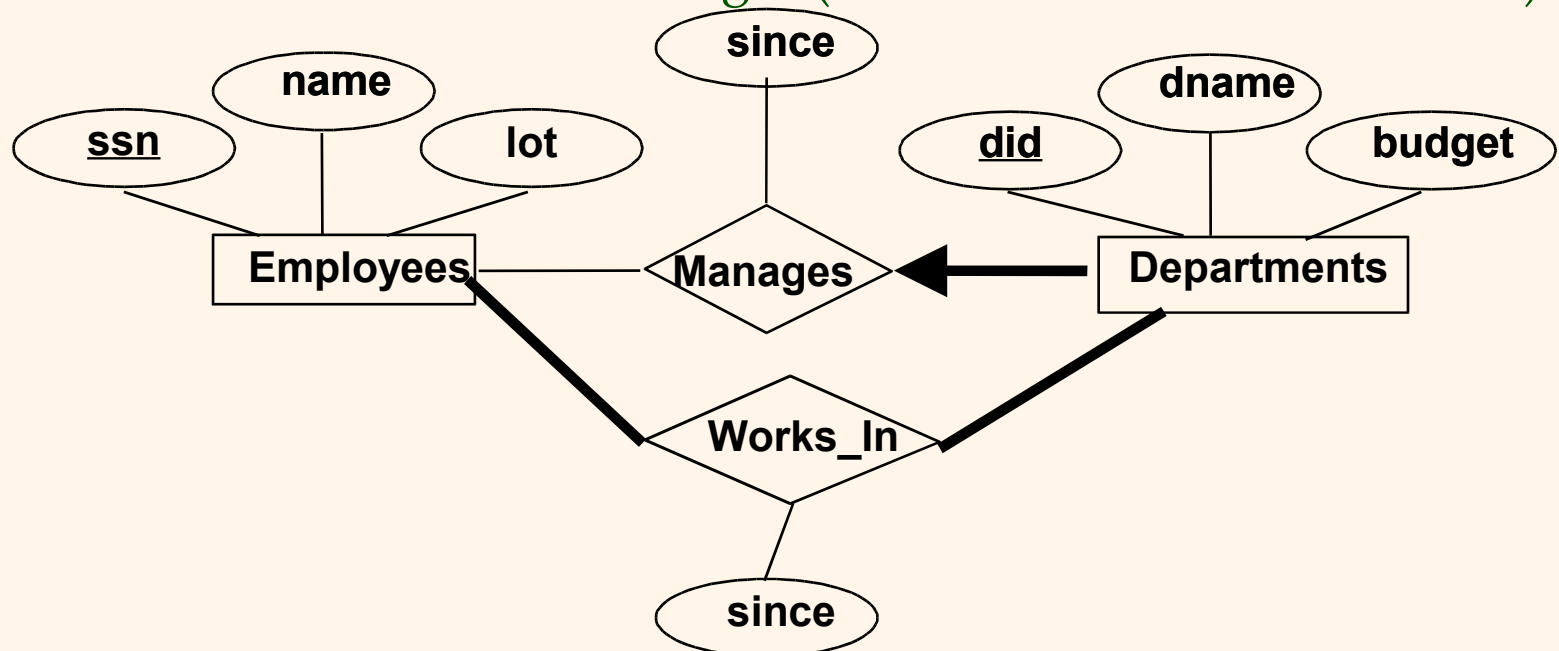
Muitos para
Muitos



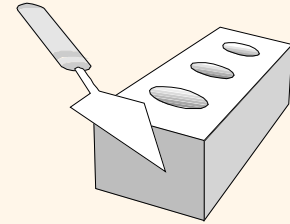
Restrição de Participação

❖ Cada um dos departamentos tem um gerente?

- Se isso for verdade, isso é uma restrição de participação: a participação de Departamentos em Gerentes é dita ser *total* (*versus. parcial*).
 - Cada entidade Departments deve aparecer em uma instância do relacionamento Manages (com um valor “*ssn*” não-nulo)

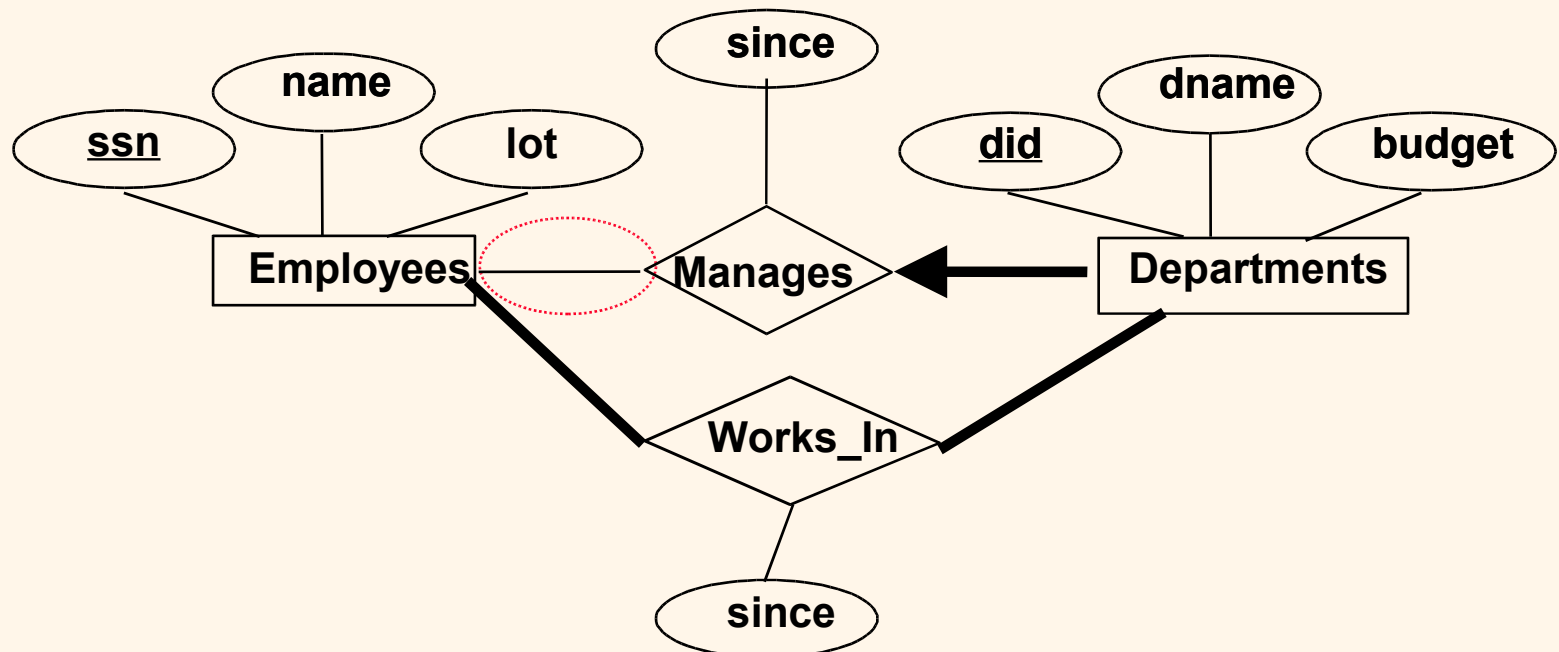


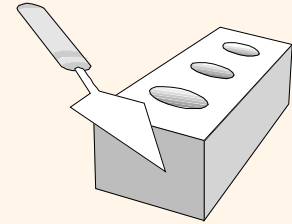
Restrição de Participação



❖ Participação Parcial

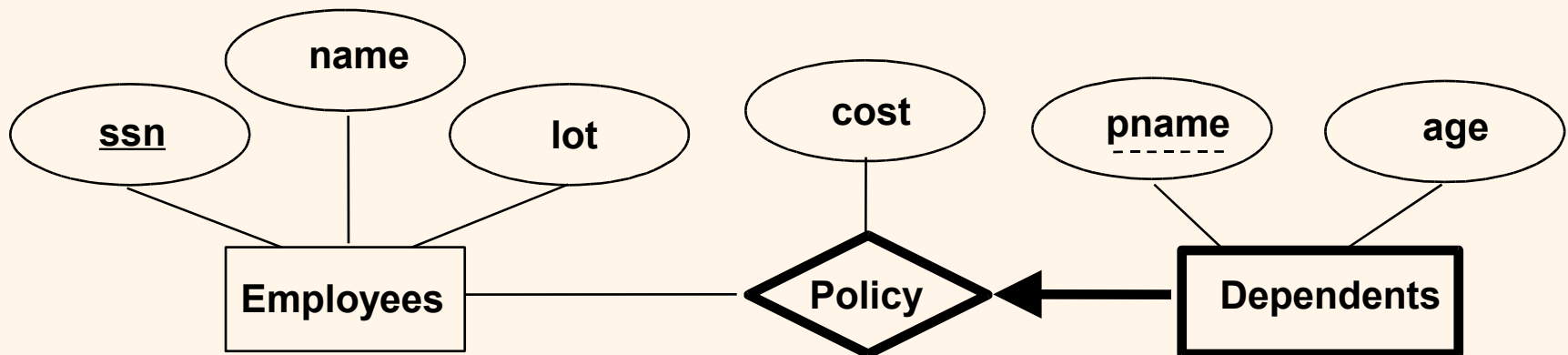
- O relacionamento de Employees e Manages é dito Parcial, uma vez que nem todo Funcionário gerencia um Departamento





Entidades Fracas

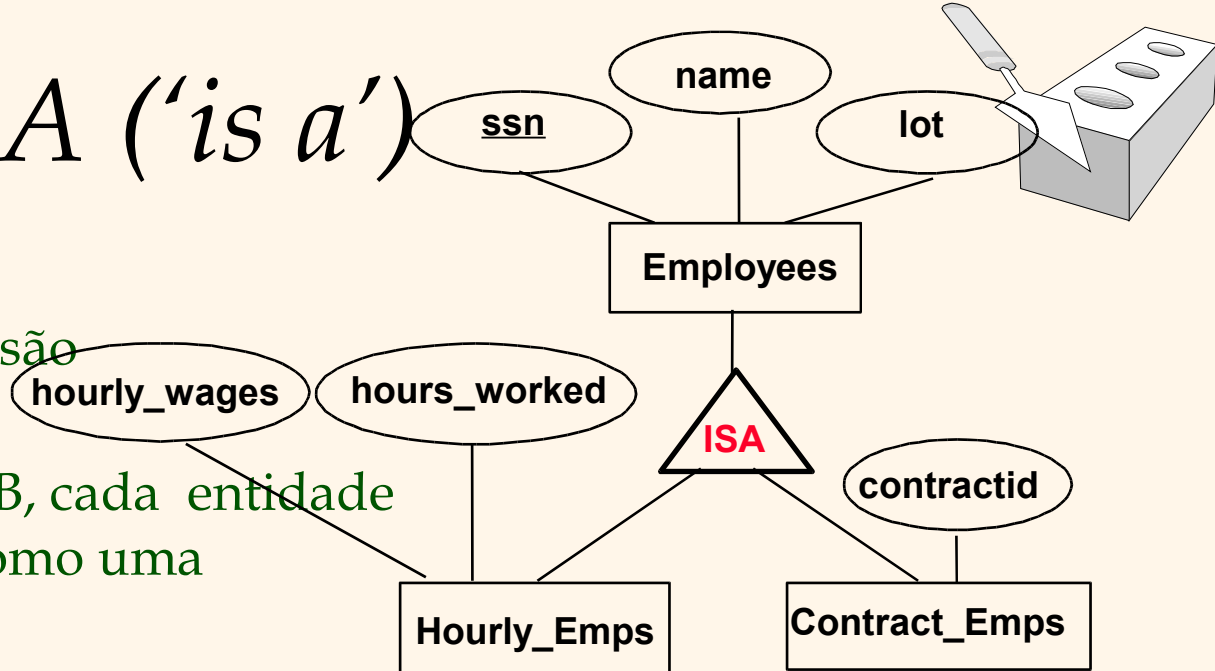
- Uma *entidade fraca* pode ser identificada unicamente apenas se considerarmos a chave primária de outra entidade (*proprietária*).
 - Um conjunto-entidade proprietário e um conjunto-entidade fraca têm que participar em um conjunto-relacionamento um-para-muitos (um proprietário, muitas entidades fracas).
 - O conjunto-entidade fraca tem participação total no conjunto-relacionamento identificado.
 - O conjunto de atributos de um conjunto-entidade fraca que unicamente a identifica para um dado *proprietário* é chamado de *chave parcial (pname)*.



Hierarquia ISA ('is a')

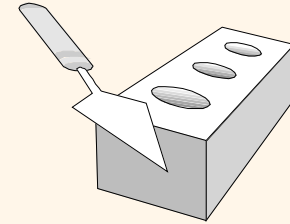
❖ Como em C++, atributos são herdados.

❖ Se nós declaramos A **ISA** B, cada entidade A é também considerada como uma entidade B.



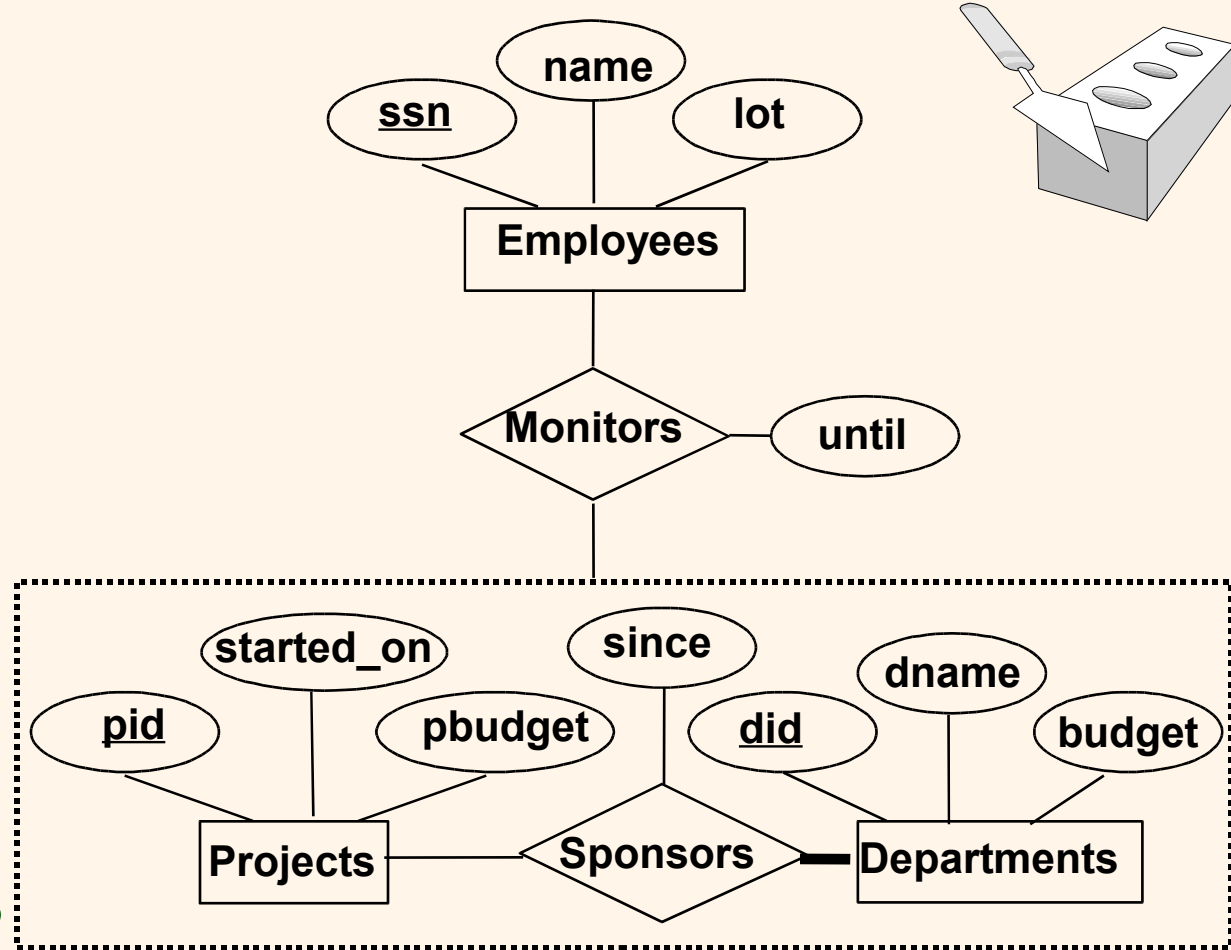
- **Restrições de Sobreposição:** Determina, neste exemplo, que um funcionário pode ser tanto um funcionário horista como um contratado (habilita / desabilita).
- **Restrições de Cobertura:** Determina, neste exemplo, que um funcionário obrigatoriamente deve ser um funcionário horista ou um funcionário contratado.
- Razões para usar ISA:
 - Adicionar atributos descritivos específicos para as entidades da subclasse.
 - Identificar entidades que participam em algum relacionamento.

Agregação



❖ Usado quando nós temos que modelar um relacionamento envolvendo (conjunto-entidade e) um *conjunto-relacionamento*.

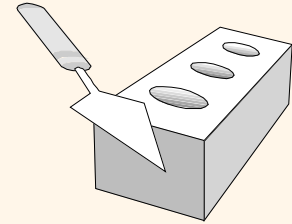
- Agregação permite-nos tratar um conjunto-relacionamento como um conjunto-entidade com o propósito de permitir a participação em (outros) relacionamentos.



➡ Agregação vs. Relacionamento Ternário:

- ❖ Monitors é um relacionamento distinto com um atributo descritivo
- ❖ Também, pode-se dizer que cada patrocínio (sponsorship) é monitorado por um único empregado (Employee).

Projeto Conceitual usando o MER



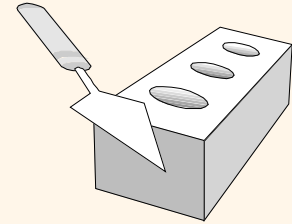
❖ Escolhas de Projeto:

- Um conceito poderia ser modelado como uma entidade ou um atributo?
- Um conceito poderia ser modelado como uma entidade ou um relacionamento?
- Identificação dos Relacionamentos: Binário ou Ternário? Agregação?

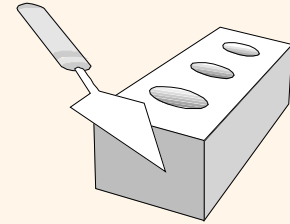
❖ Restrições no modelo ER:

- Muitos dados semânticos podem (e devem) ser capturados.
- Mas algumas restrições não podem ser capturadas em um diagramas ER.

Entidade vs. Atributo

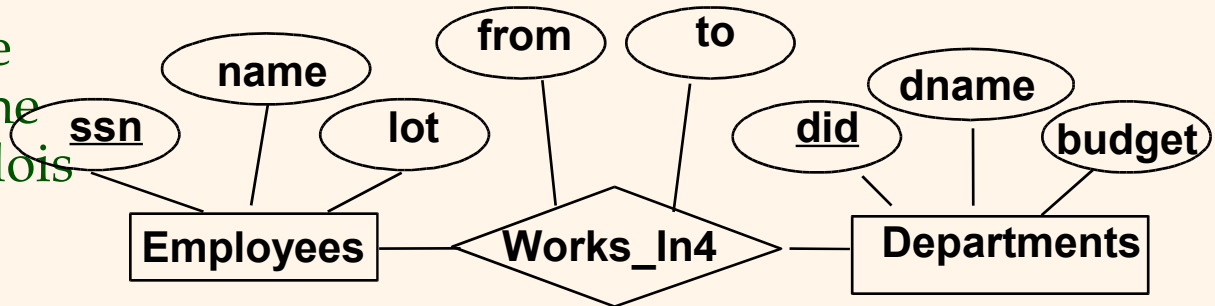


- ❖ *address* deveria ser um atributo de Employees ou uma entidade (conectado a Employees através de um relacionamento)?
- ❖ Depende do uso que nós queremos fazer da informação de endereço:
 - Se temos diversos endereços para Employee, *address* tem que ser uma entidade (uma vez que os atributos não podem ser multi-valorados).
 - Se a estrutura (city, street, etc.) é importante, por ex., nós queremos recuperar Employees em uma dada city, *address* tem que ser modelado como uma entidade (uma vez que os valores dos atributos são atômicos).

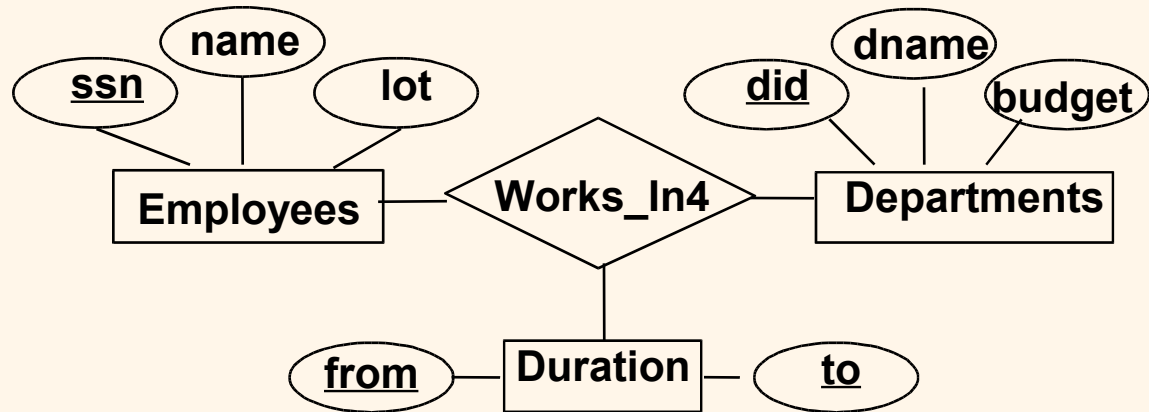


Entidade vs. Atributo

- “Works_In4” não permite que um Employee trabalhe em um Department por dois ou mais períodos.

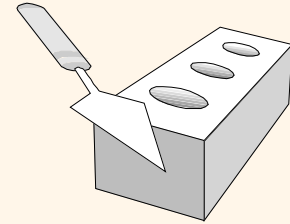


- Similar ao problema quando se quer registrar vários endereços para um Employee. Nós queremos registrar *diversos valores de atributos descritivos para cada instância desse relacionamento.*

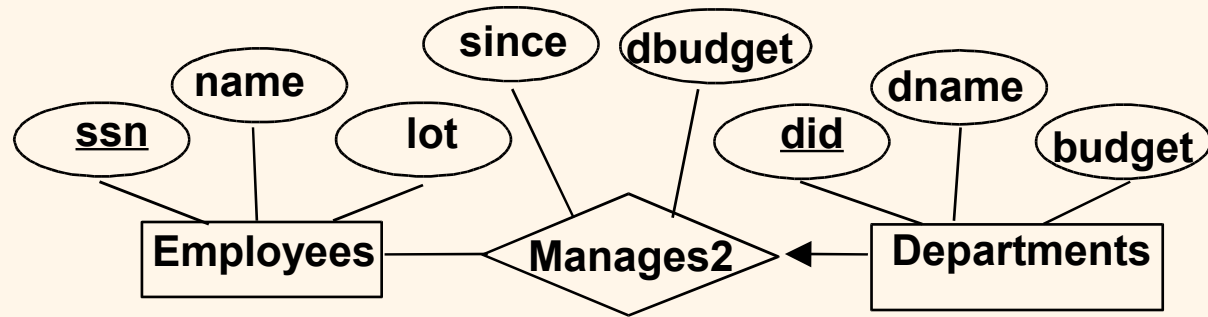


Torna-se completo pela introdução de um novo conjunto-entidade: “Duration”

Entidade vs. Relacionamento



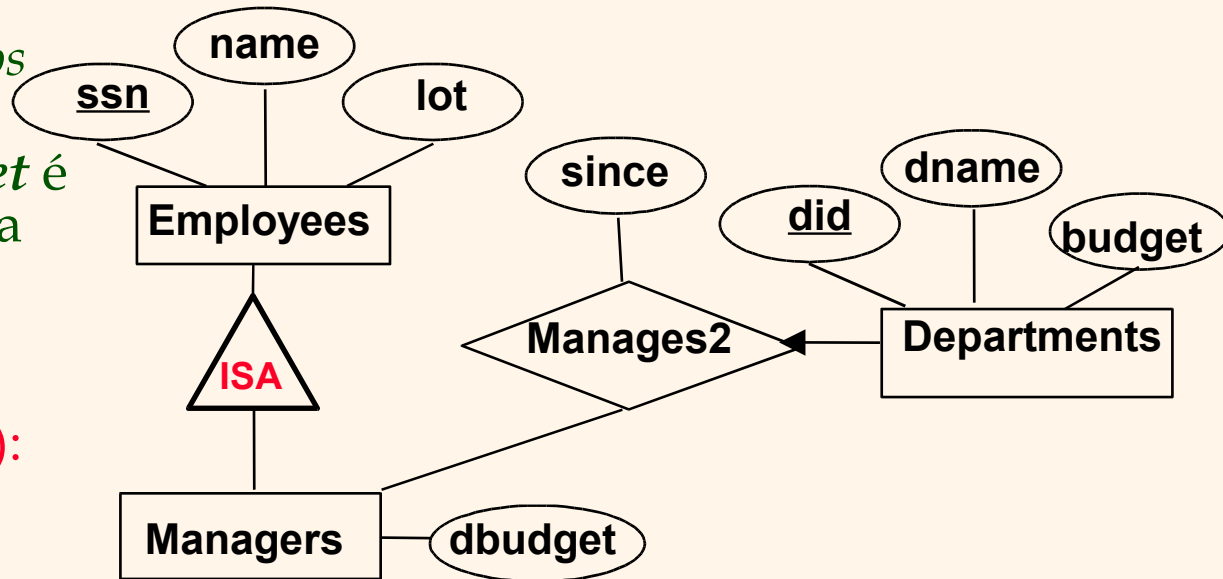
- O primeiro diagrama ER está OK se um gerente consegue separar um orçamento para cada departamento.



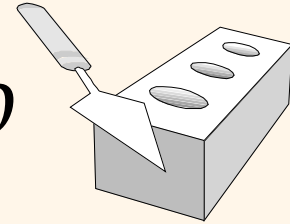
- O que acontece se um gerente consegue um orçamento que cubra *todos* os departamentos?

- **Redundância:** *dbudget* é armazenado para cada departamento gerenciado por um gerente.

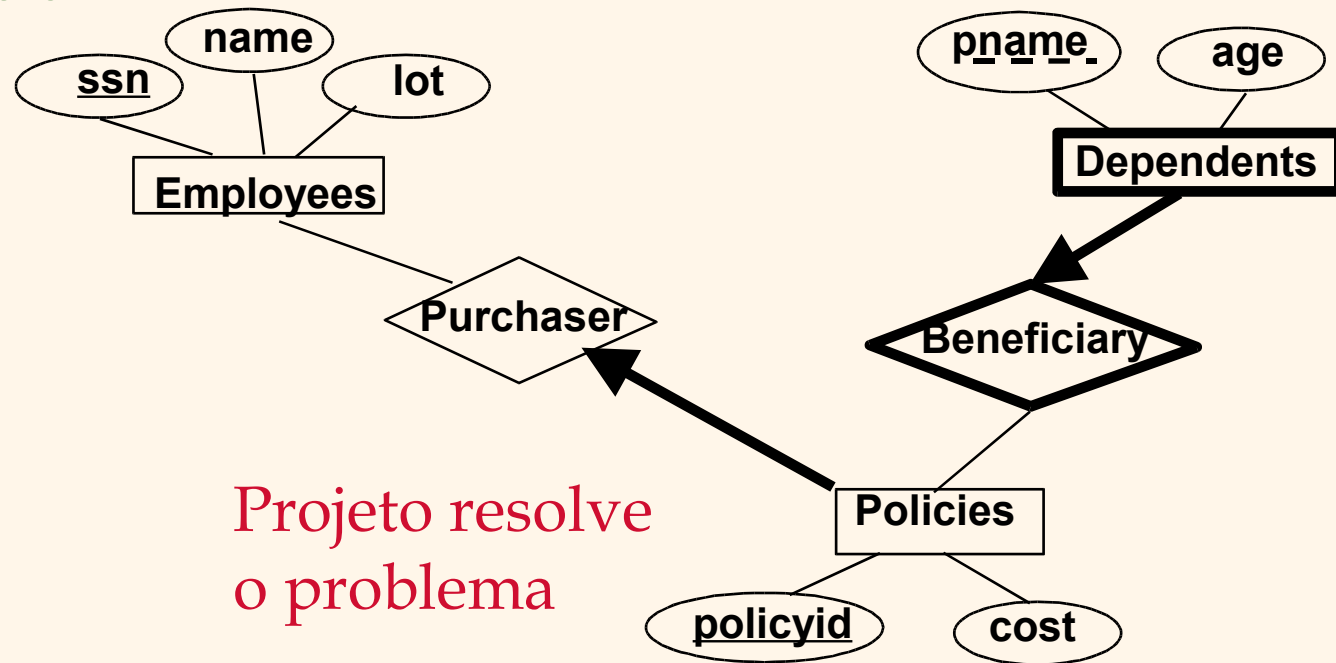
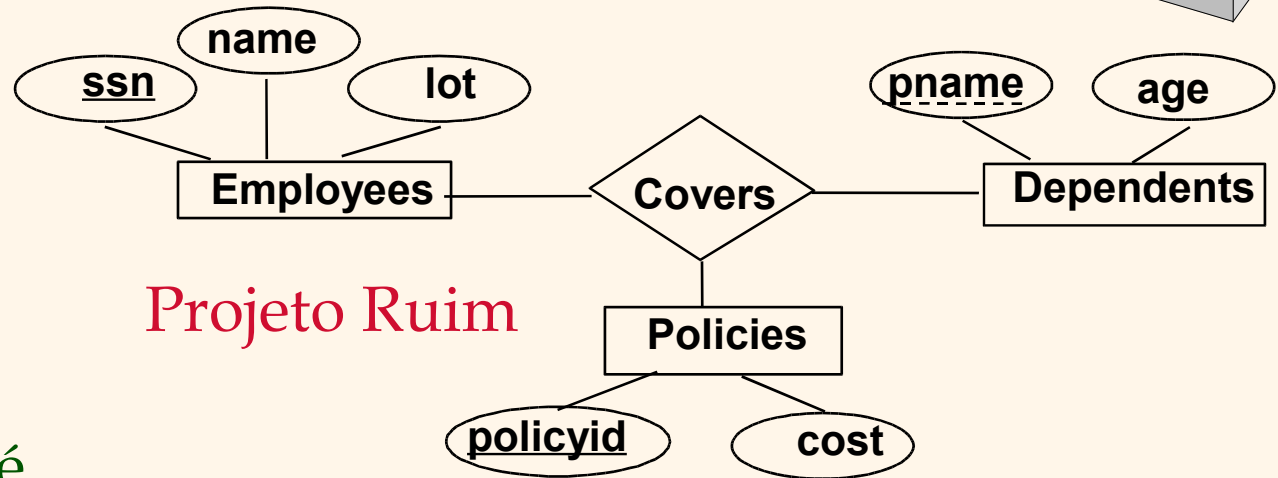
- **Equívoco (Misleading):** sugere *dbudget* associado com uma combinação de Managers-Departments.



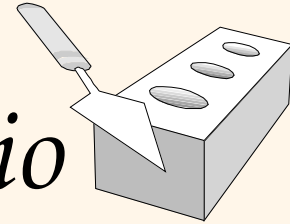
Relacionamento Binário vs. Ternário



- Se cada política é propriedade de um único funcionário, e cada dependente é vinculado à cobertura da política, o primeiro diagrama não é adequado.
- Quais são as restrições adicionais no 2º diagrama?



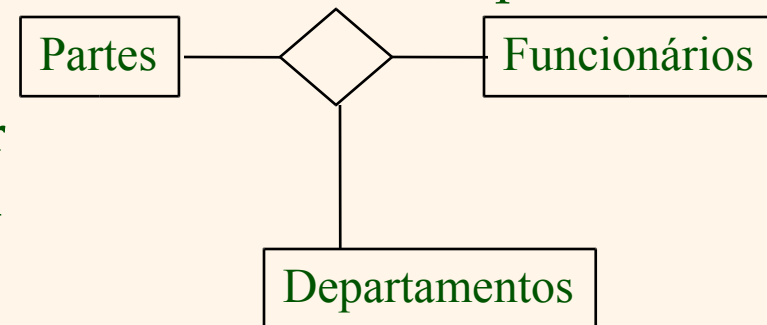
Relacionamento Binário vs. Ternário

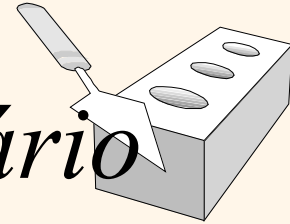


- O exemplo anterior ilustra um caso onde dois relacionamentos binários são melhores do que um relacionamento ternário.
- Um exemplo em outra direção:

uma relação ternária **Contratos** relaciona os conjuntos-entidade **Partes**, **Departamentos** e **Fornecedores** com um atributo descritivo *qty*. Nenhuma combinação de relacionamentos binários seria adequada:

- Com o relacionamento binário, nós podemos denotar que um fornecedor 'pode fornecer' certas partes, que um departamento 'necessita' de algumas partes, e que um departamento 'negocia com' um certo fornecedor.
- Como podemos registrar *qty*?



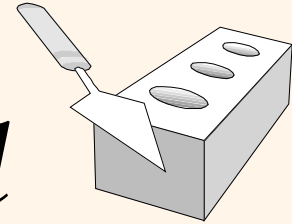


Binário vs. Relacionamento Ternário *(Cont.)*

- A não combinação desses relacionamentos binários expressa o significado de um contrato adequado, razões:
 - O fato que o fornecedor S ‘pode-fornecer’ P, que o departamento D ‘necessita’ da parte P, e que D ‘comprará-de’ S não necessariamente implicará que o departamento D realmente compra a parte P do fornecedor S

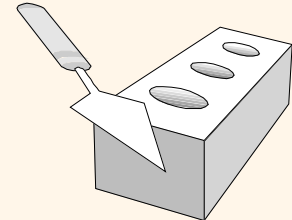
Fornecedor S ‘pode-fornecer’ P
Departamento D ‘necessita’ da parte P
Departamento D ‘comprará-de’ S

Sumário de Projeto Conceitual

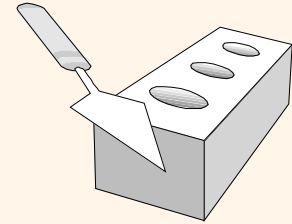


- ❖ *Projeto Conceitual* segue a *análise de requisitos*,
 - Produz uma descrição alto-nível dos dados a serem armazenados
- ❖ Modelo ER é popular para o projeto conceitual
 - Construções são expressivas, próximo da maneira que as pessoas pensam sobre suas aplicações.
- ❖ Construção Básica: *entidades, relacionamentos e atributos (de entidades e relacionamentos)*
- ❖ Algumas construções adicionais: *entidades fracas, hierarquia ISA, e agregação.*
- ❖ Nota: Há muitas variações no modelo ER.

Sumário de ER

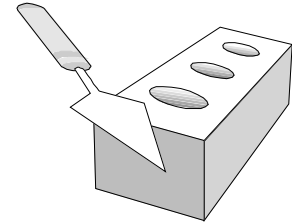


- ❖ Diversos tipos de restrições de integridade podem ser expressos no modelo ER: *restrição de chave*, *restrição de participação* e *restrições de sobreposição/cobertura* para hierarquias ISA. Algumas *restrições de chave estrangeira* são também implícitas à definição de um conjunto de relacionamentos.
 - Algumas restrições (notavelmente, *dependências funcionais*) não podem ser expressas no modelo ER.
 - Restrições representam um importante papel para determinar o melhor projeto de banco de dados para um empreendimento.



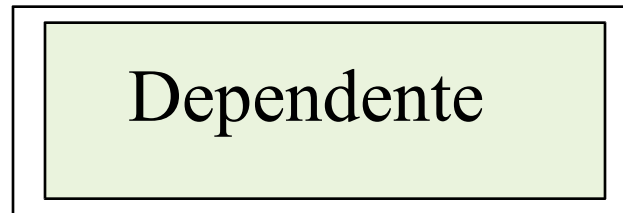
Sumário de ER

- ❖ Projeto ER é *subjetivo*. Há muitos caminhos para modelar um mesmo cenário. A Análise de Alternativas pode ser ‘enganadora’, especialmente para um grande empreendimento. Escolhas comuns incluem:
 - Entidade vs. Atributo, Entidade vs. Relacionamento, Relacionamento Binário ou N-Ário, usar ou não a hierarquia ISA, e usar ou não a Agregação.
- ❖ Garantir um bom projeto do banco de dados: resultando num esquema relacional que possa ser analisado e refinado posteriormente. Técnicas de normalização são especialmente úteis.



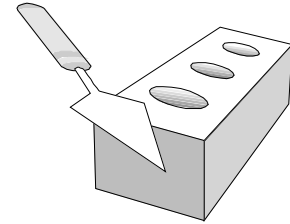
Notação de Outros Autores (Elmasri & Navathe)

- Notação para Entidade Fraca



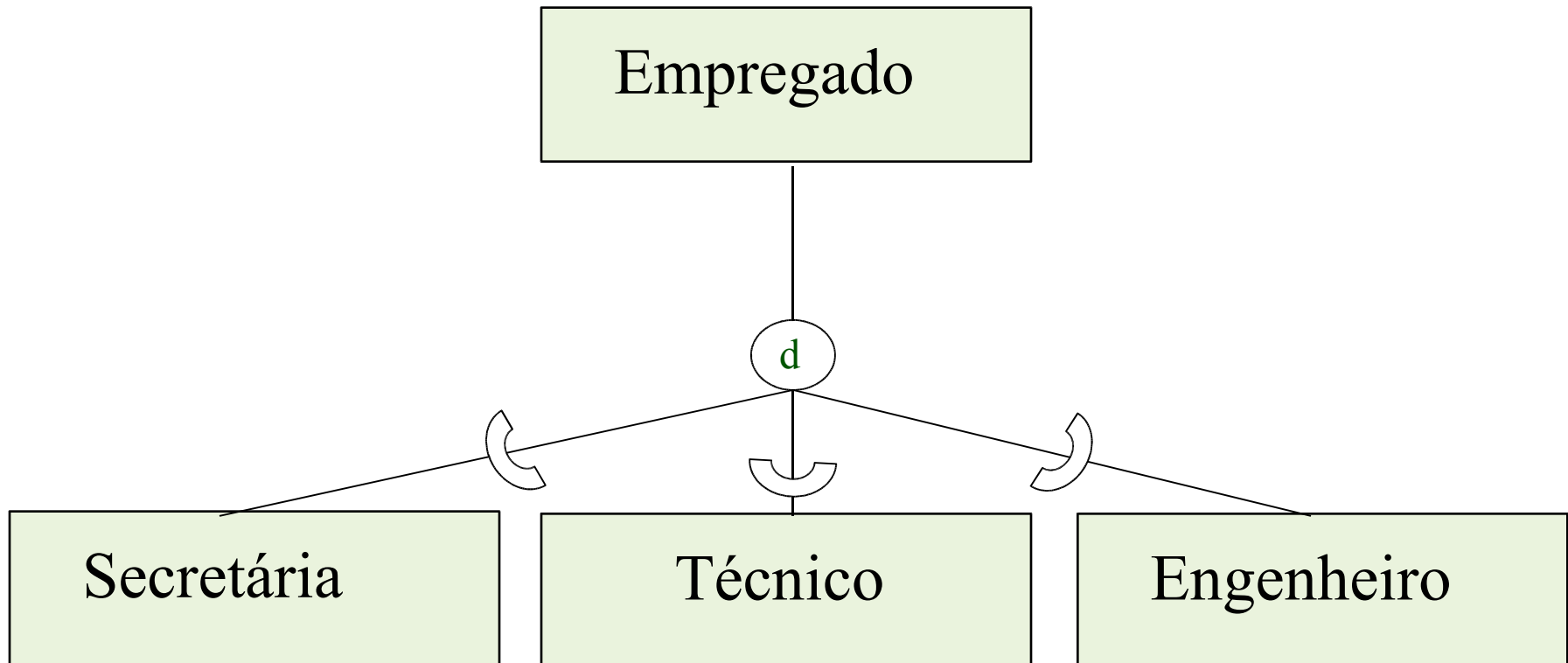
- Notação para Restrição de Participação



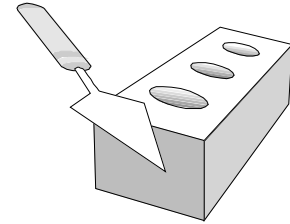


Notação de Outros Autores (Elmasri & Navathe)

- Notação para Generalização / Especialização



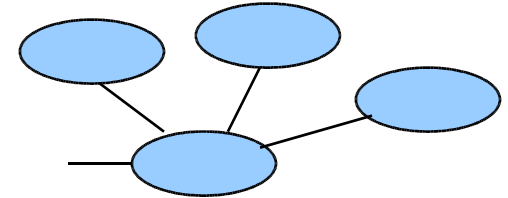
Notação de Outros Autores (Elmasri & Navathe)



❖ Classificação

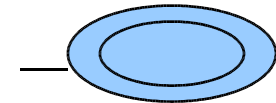
■ Quanto à Composição

- Simples
- Compostos (Subdividido ou Hierarquizado)



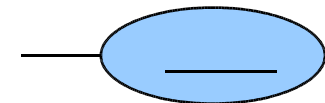
■ Quanto ao Número de Valores

- Simples
- Multivalorados (Mais de um valor por instância)



■ Quanto à Unicidade

- Simples
- Chave (Determinam unicidade)

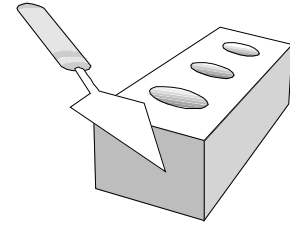


■ Quanto ao Armazenamento

- Simples
- Derivados (Calculados)



Notação de Outros Autores *(Elmasri & Navathe)*



Funcionário

Departamento

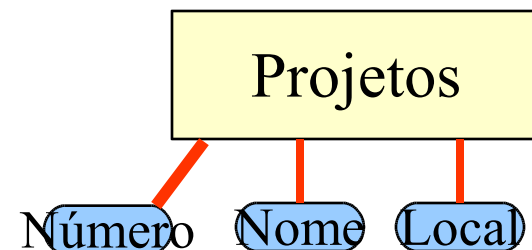
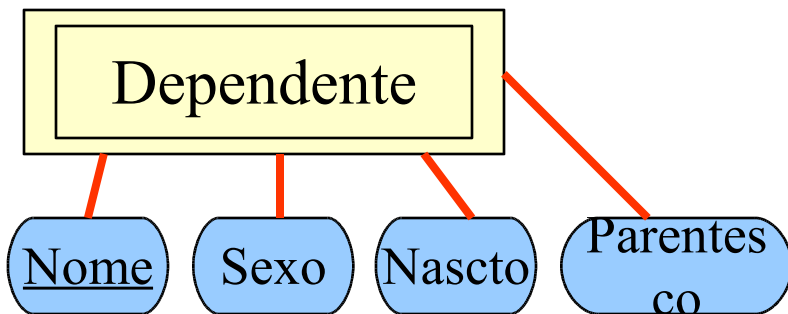
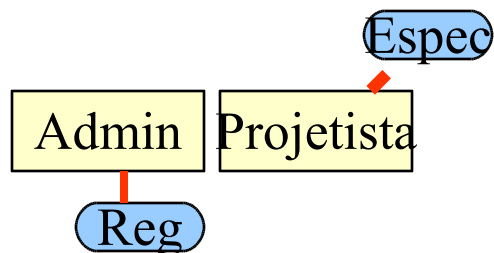
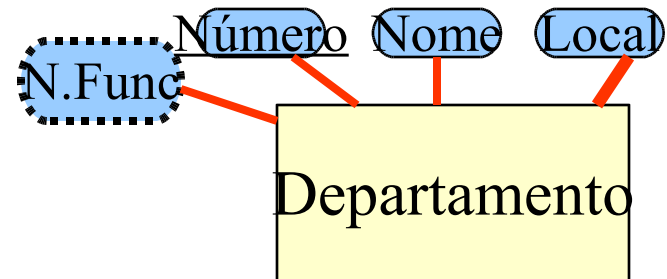
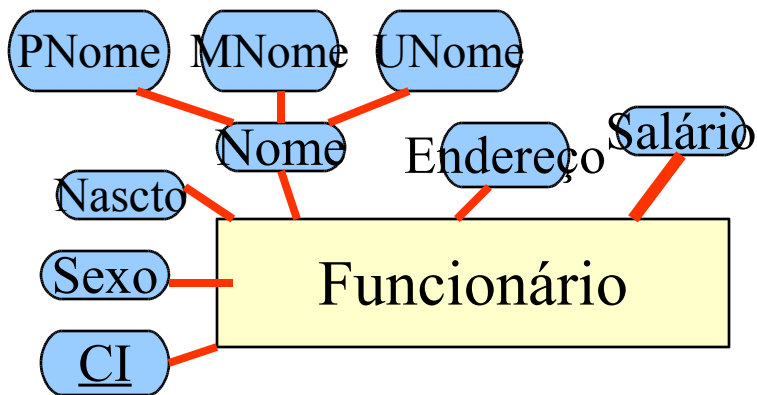
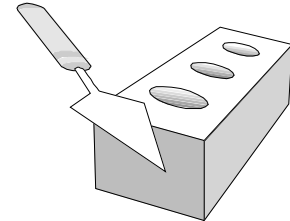
Admin

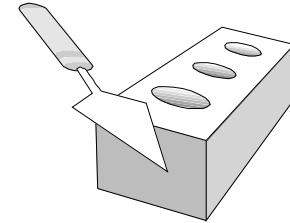
Projetista

Dependente

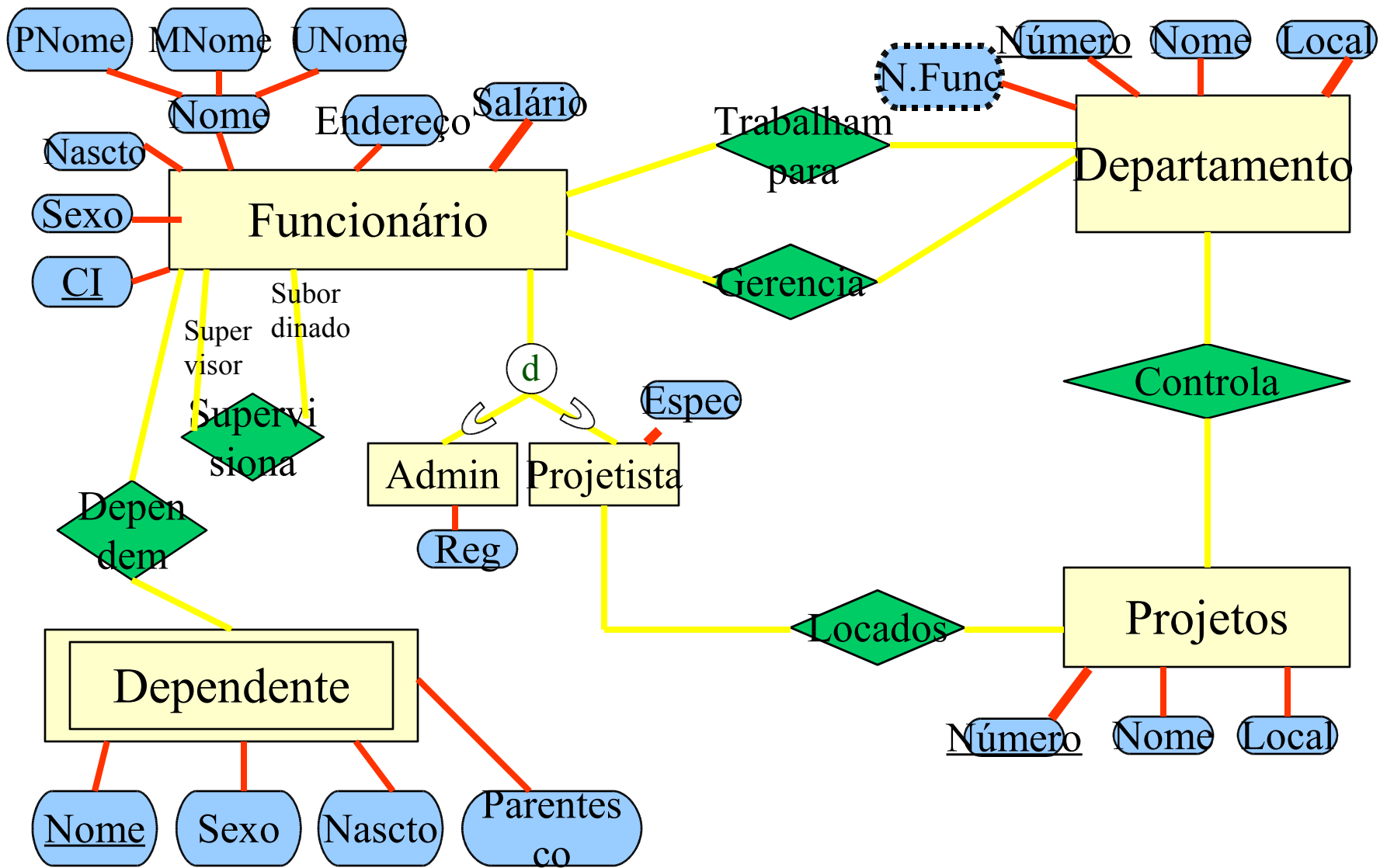
Projetos

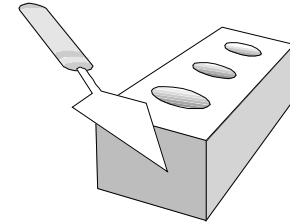
Atributos



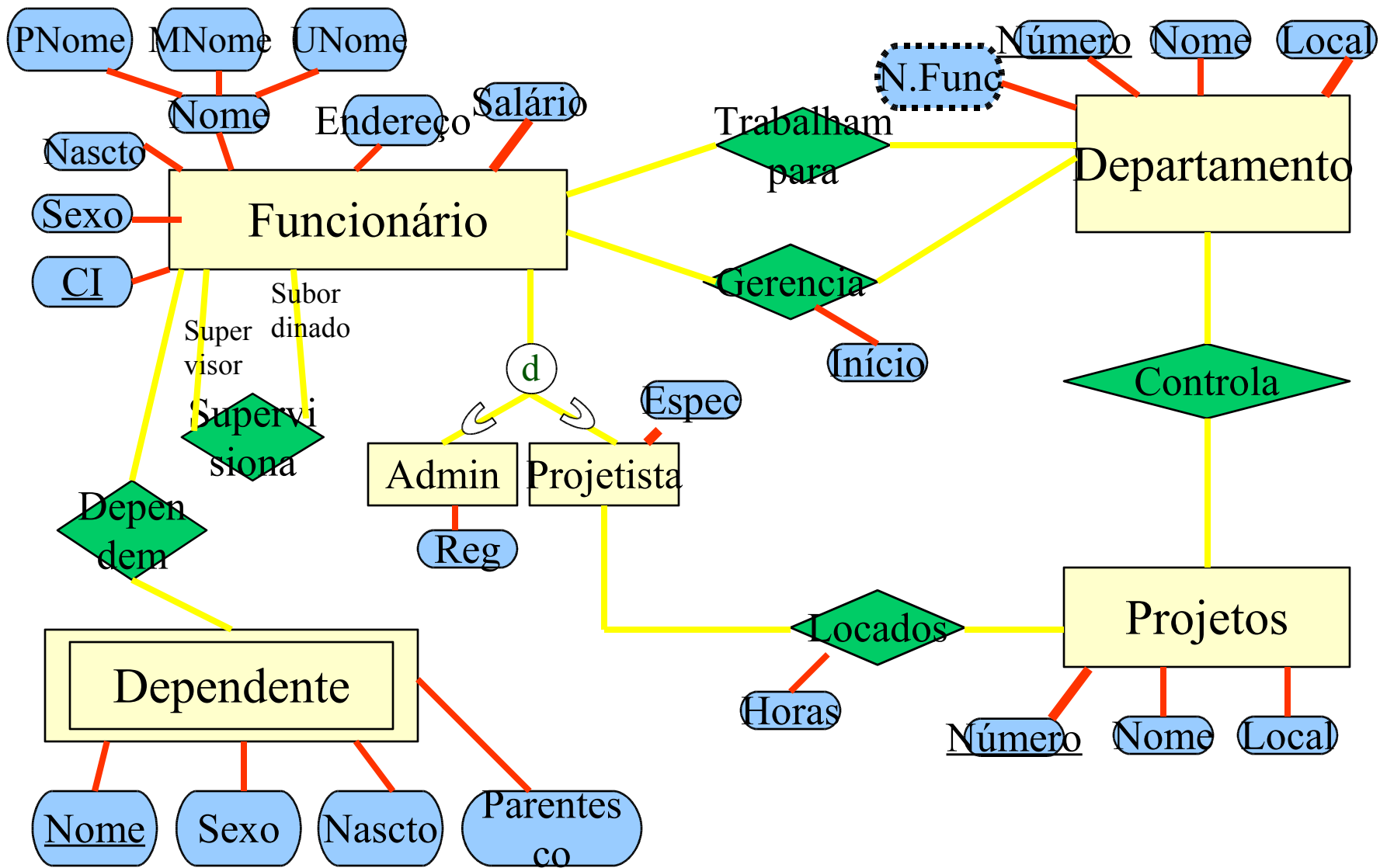


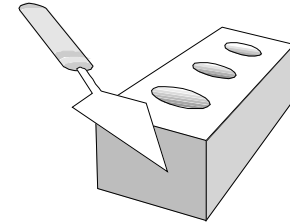
Relacionamentos





Atributos Relacionamentos





Restrições

