Introdução às Bases de Dados

Modelo Relacional - III

Passagem do EA para Relacional

FCUL, Departamento de Informática

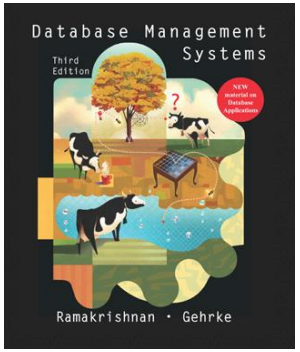
Ano Letivo 2021/2022

Ana Paula Afonso

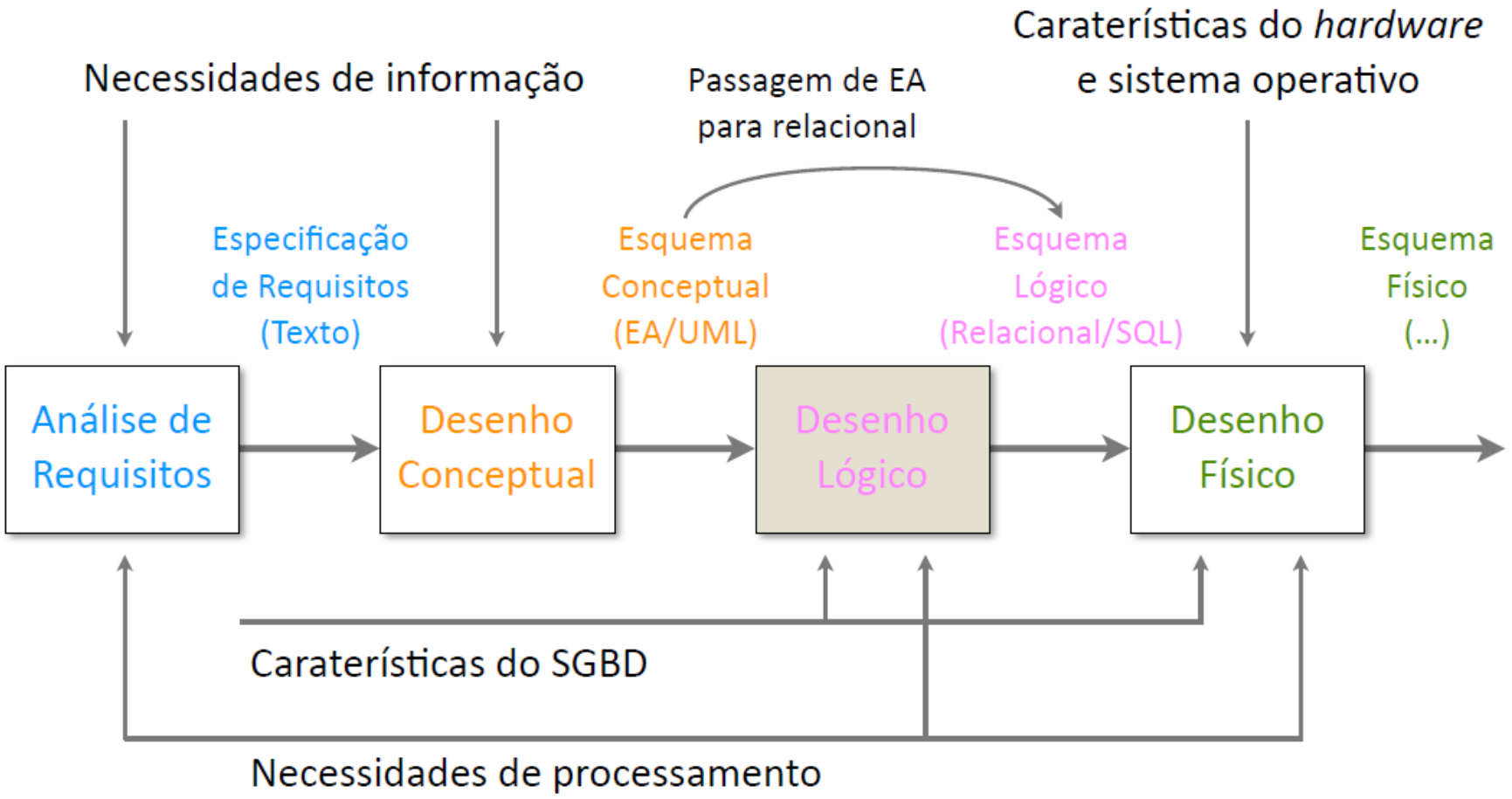
Sumário e Referências

- Sumário
  - Passagem de EA para Relacional
    - Enquadramento no processo de desenho de BD
  - Entidades para Tabelas
  - Transformação de Associações
    - Associações com Restrições de Chave
    - Restrições de Participação
  - Entidades Fracas
  - Generalizações
  - Agregações

- Referências
  - R. Ramakrishnan (**capítulo 3, secção 3.5**)

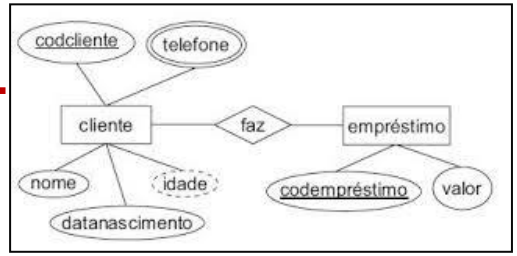


Processo de Desenho de BDs



Fonte: António Ferreira, Guião SIBD, 2016

Passagem de EA para Relacional

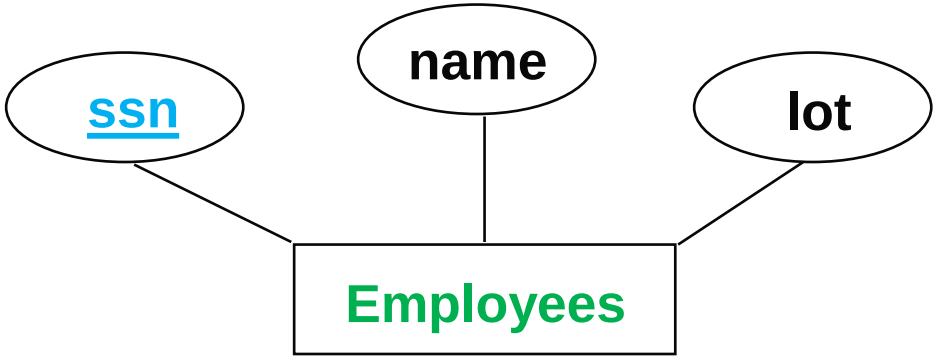


- Modelo entidade-associação (EA)
  - Adequado para o desenho inicial, de **alto nível**, da base de dados
  - Representação gráfica para facilitar discussão de alternativas por equipas
  - Mas não entendido pelos sistemas de gestão de bases de dados (SGBD)
- Modelo relacional
  - Suportado pelos SGBDs relacionais, muito populares
  - Mas de **baixo nível**, com comandos de texto, que dificultam discussão
  - Maior risco de perder a visão do todo, focando apenas nas partes
- Após discussão de alternativas e integração de diagramas EA
  - Esquema EA é traduzido num esquema relacional (ER) **aproximado**
  - Com **tabelas e restrições de integridade** escritas na linguagem SQL
  - Algumas restrições de integridade EA podem não ser concretizadas em SQL

```
CREATE TABLE cliente (
    codcliente integer,
    ….)
```

Entidades para ER

Nome da tabela igual ao nome da entidade
Colunas da tabela são os atributos da entidade
Chave primária da tabela vem da chave primária da entidade




```
CREATE TABLE Employees (
 ssn CHAR(11),
 name CHAR(30),
 lot INTEGER,
 PRIMARY KEY (ssn) )
```

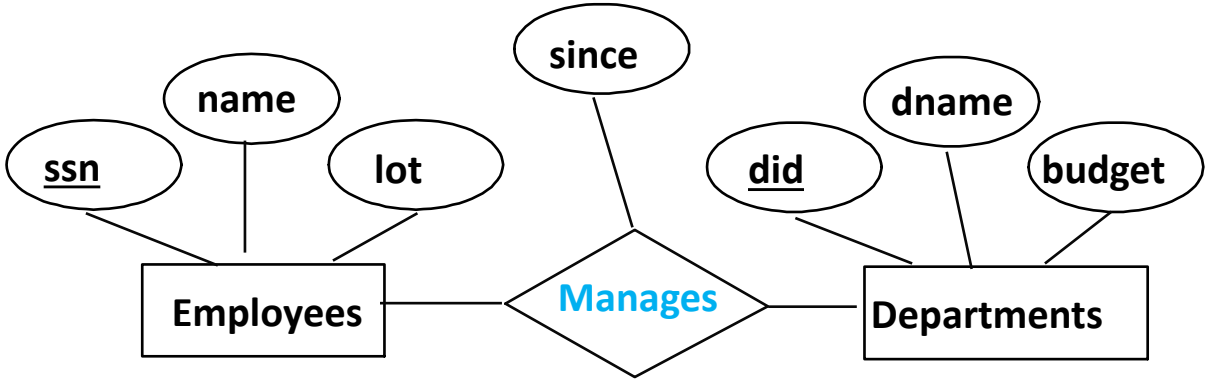
Associações para ER

Caso 1: sem restrições de chave e participação

Nome da tabela igual ao nome da associação

Chave primária da tabela é **composta p**elas chaves primárias das entidades participantes

Chaves estrangeiras referenciam as entidades

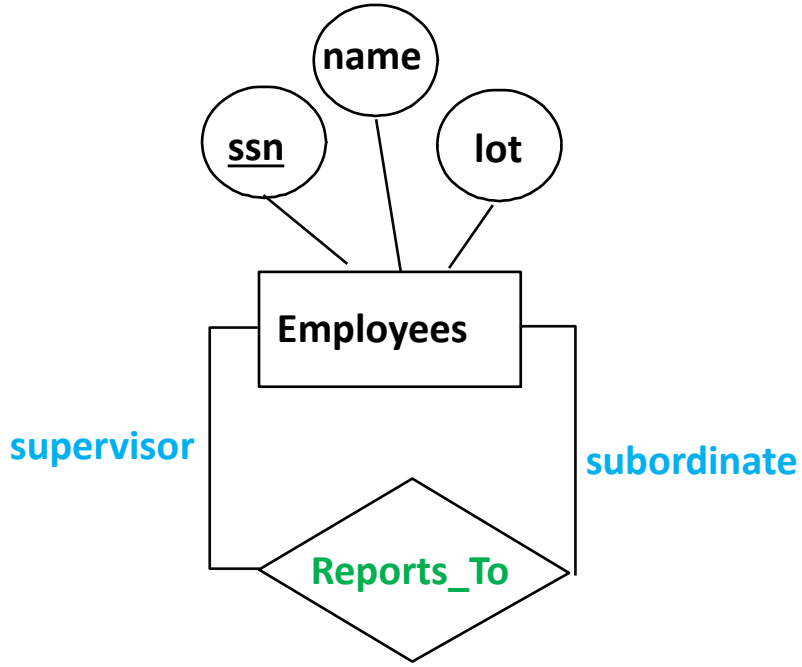



```
CREATE TABLE  Manages(
    ssn  CHAR(11),
    did  INTEGER,
    since  DATE,
    PRIMARY KEY  (did, ssn),
    FOREIGN KEY (ssn) REFERENCES Employees,
    FOREIGN KEY (did) REFERENCES Departments)
```

Associações para ER

Caso 1: sem restrições de chave e participação

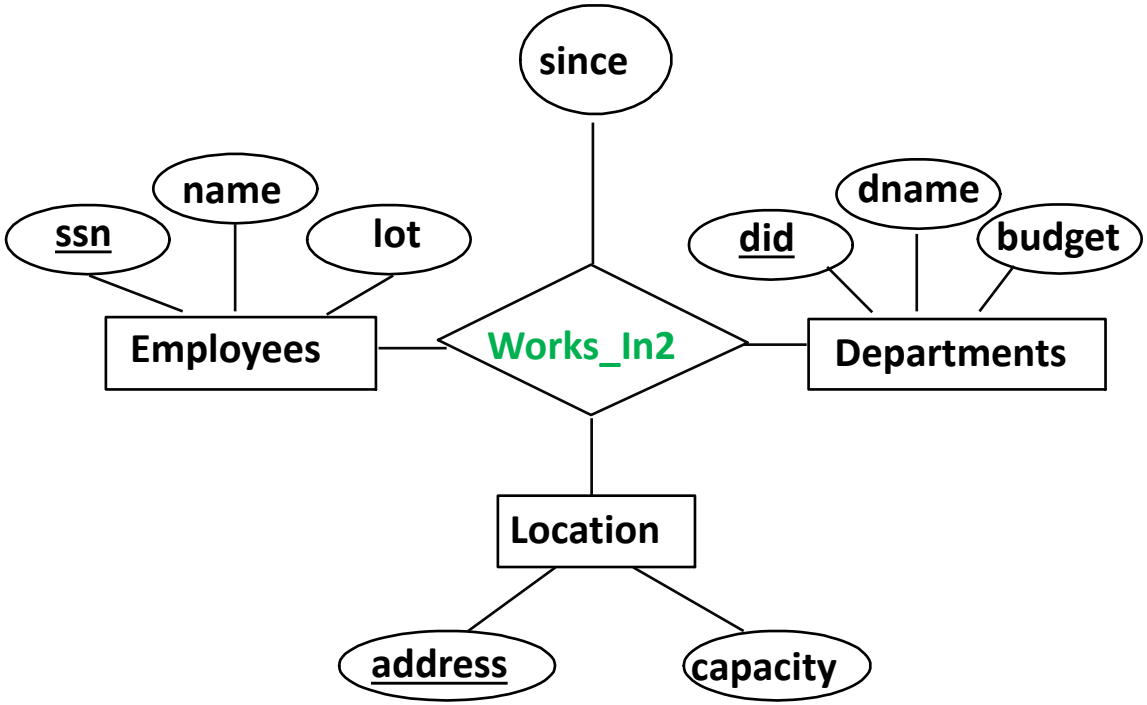
Papéis ajudam a dar nome às colunas




```
CREATE TABLE Reports_To (
 supervisor_ssn CHAR (11),
 subordinate_ssn CHAR (11) ,
 PRIMARY KEY (supervisor_ssn, subordinate_ssn),
 FOREIGN KEY (supervisor_ssn) REFERENCES Employees,
 FOREIGN KEY (subordinate_ssn)REFERENCES Employees)
```

Associações para ER

Caso 1: sem restrições de chave e participação

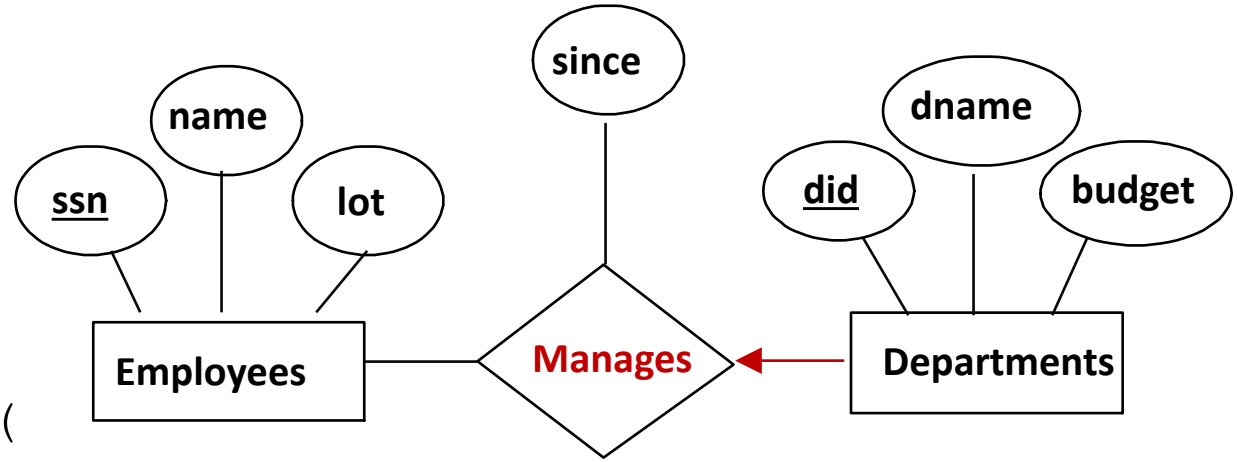



```
CREATE TABLE Works_In2 (
 ssn   CHAR(11),
 did   INTEGER,
 address      CHAR(20),
 since DATE,
 PRIMARY KEY (ssn, did, address),
 FOREIGN KEY (ssn) REFERENCES Employees,
 FOREIGN KEY (address) REFERENCES Location,
 FOREIGN KEY (did) REFERENCES Departments)
```

Associações para ER

Caso 2: com restrição de chave

Abordagem 1: **criação de uma nova tabela**



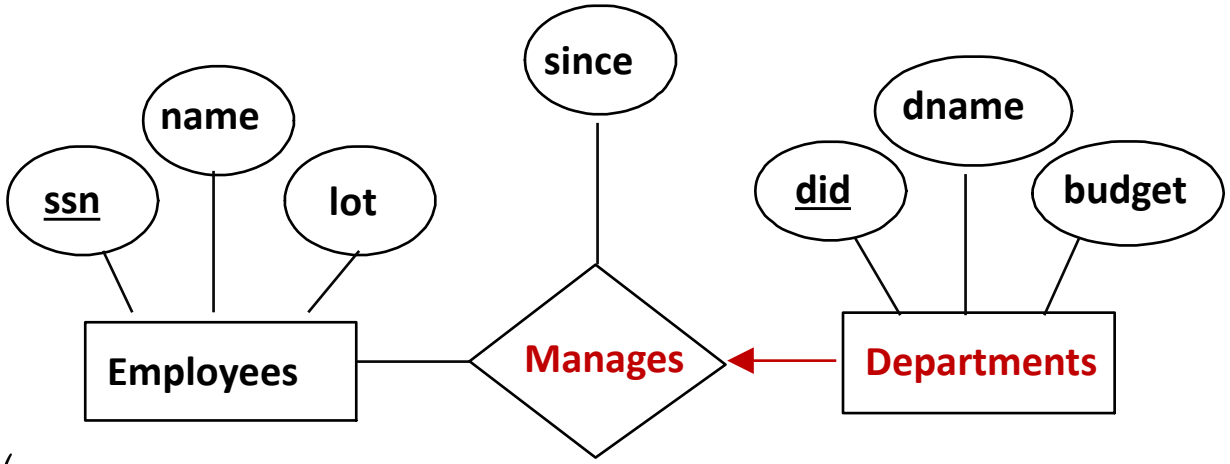

```
CREATE TABLE  Manages(
    ssn  CHAR(11) NOT NULL,
    did  INTEGER,
    since  DATE,
    PRIMARY KEY  (did),
    FOREIGN KEY (ssn) REFERENCES Employees,
    FOREIGN KEY (did) REFERENCES Departments)
```

Nota: (ssn,did) como chave não cumpre a regra de ser o conjunto minímo

Associações para ER

Caso 2: com restrição de chave

Abordagem 2: **Adição de chave estrangeira** à tabela Dept existente




```
CREATE TABLE  Dept_Mgr(
    did  INTEGER,
    dname  CHAR(20),
    budget  REAL,
    ssn  CHAR(11),
    since  DATE,
    PRIMARY KEY  (did),
    FOREIGN KEY (ssn) REFERENCES Employees)
```

Associações para ER

Caso 2: com restrição de chave

Duas abordagens

1. **Criar uma nova tabela**

   Vantagens

   - Atributos descritivos da associação na **sua própria tabela**
   - **Restrição de participação parcial** facilmente suportada: basta não inserir linhas na tabela

   Desvantagens

   - **Mais uma tabela** no esquema relacional torna pesquisas mais complexas
   - Restrição de participação total **custosa**: necessárias asserções

   Usar em casos de associações com muitos atributos descritivos

Associações para ER

Caso 2: com restrição de chave

Duas abordagens

2. Adicionar chave estrangeira à tabela existente

Vantagens

- **Menos uma tabela** no esquema relacional permite pesquisas mais simples
- **Restrição de participação total** facilmente suportada: basta usar NOT NULL

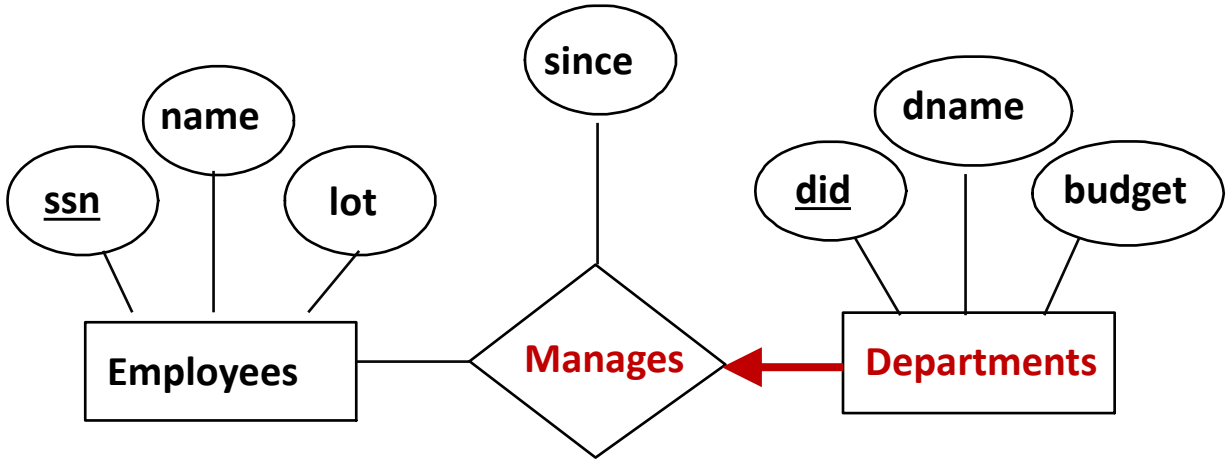
Desvantagens

- **Mistura atributos** da associação e entidade na mesma tabela
- Restrição de participação parcial pode levar a **muitos valores nulos** nas linhas da tabela

Associações para ER

Caso 3: com restrição de chave e participação

Adição de chave estrangeira à tabela existente e restrição **NOT NULL**




```
CREATE TABLE Dept_Mgr(
 did INTEGER,
 dname CHAR(20),
 budget REAL,
 ssn CHAR(11) NOT NULL,
 since DATE,
 PRIMARY KEY (did),
 FOREIGN KEY (ssn) REFERENCES Employees ON DELETE NO ACTION)
```

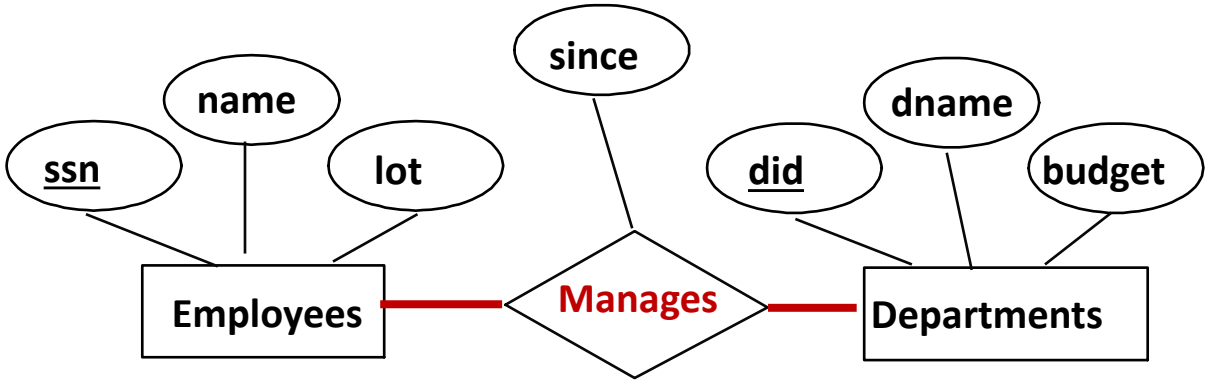
NO ACTION

ação por defeito. Um empregado não pode ser removido se tiver um Dept_Mgr a referenciá-lo

Associações para ER

Caso 4: com restrição de participação

Criação de uma nova tabela e asserção




```
CREATE TABLE  Manages(
    ssn  CHAR(11),
    did  INTEGER,
    since  DATE,
    PRIMARY KEY  (did, ssn),
    FOREIGN KEY (ssn) REFERENCES Employees,
    FOREIGN KEY (did) REFERENCES Departments)
```

Nota: Por agora, Asserção é uma restrição de integridade textual

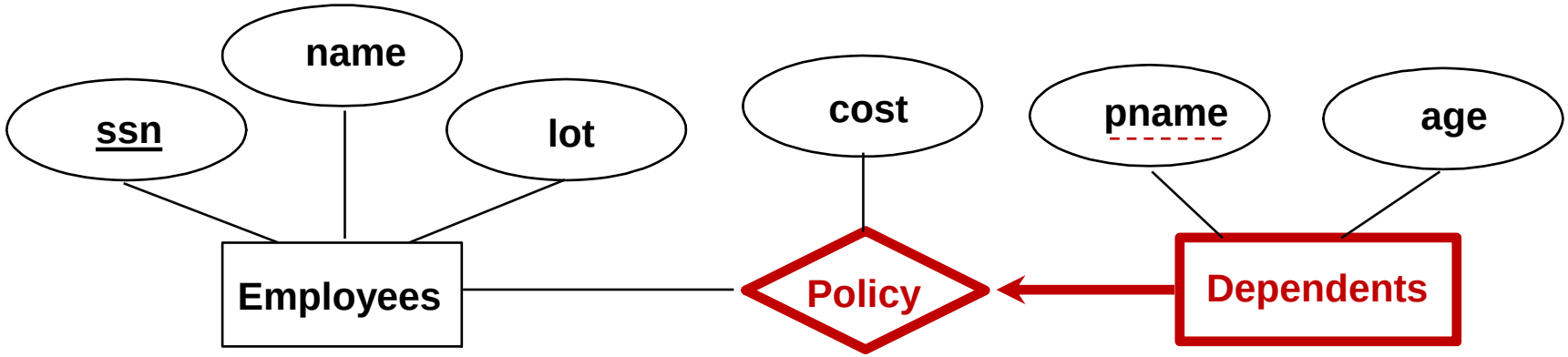
RI-1: cada departamento tem de ter pelo menos um empregado (e vice-versa)

Entidades Fracas para ER

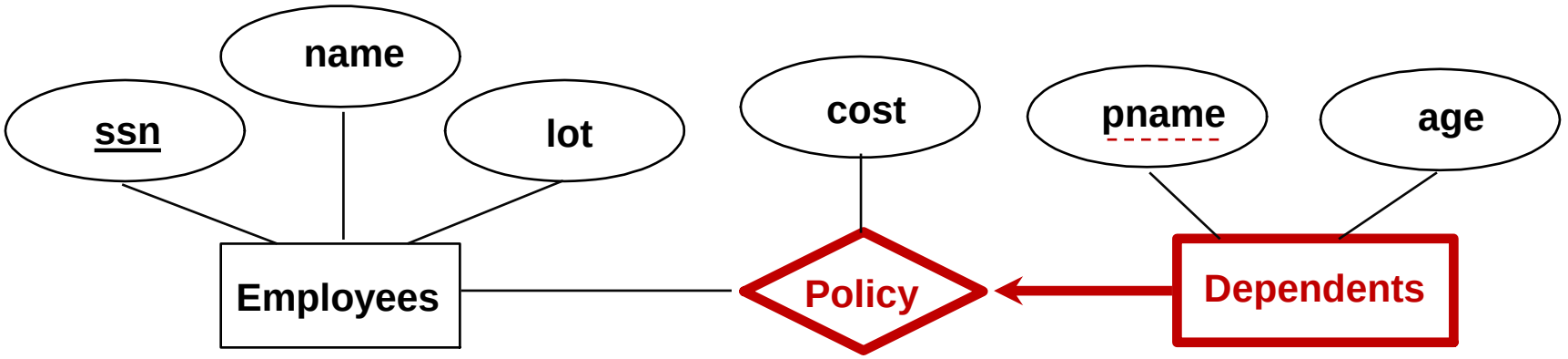
Criação de nova tabela e chave estrangeira para entidade forte

Chave primária da entidade fraca é **composta**…

… por chave parcial e chave primária da entidade forte



Entidades Fracas para ER




```
CREATE TABLE Dep_Policy (
 pname CHAR(20) ,
 age INTEGER,
 cost REAL,
 ssn CHAR (11) ,
 PRIMARY KEY (pname, ssn),
 FOREIGN KEY (ssn) REFERENCES Employees ON DELETE CASCADE)
```

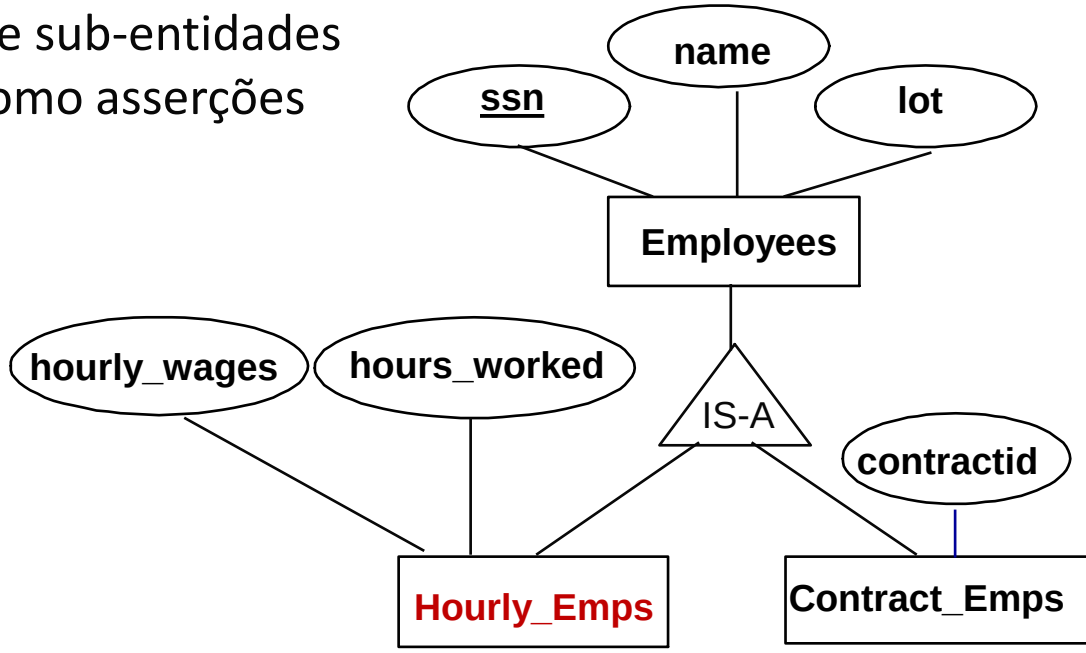
CASCADE

Remoção de linha na entidade forte (employees) despoleta a remoção das respetivas linhas na entidade fraca (dependents)

Generalizações para ER

Abordagem 1

Criação de tabelas para a super-entidade e sub-entidades
Restrições de cobertura e sobreposição como asserções



Nas tabelas das sub-entidades

- Chave primária vem da super-entidade

  E se a sub-entidade tiver uma chave própria?

- Chave estrangeira para a super-entidade com propagação de remoções

```
CREATE TABLE Hourly_Emps (
 hours_worked INTEGER,
 hourly_wages REAL,
 ssn CHAR (11),
 PRIMARY KEY (ssn),
 FOREIGN KEY (ssn) REFERENCES Employees ON DELETE CASCADE )
```

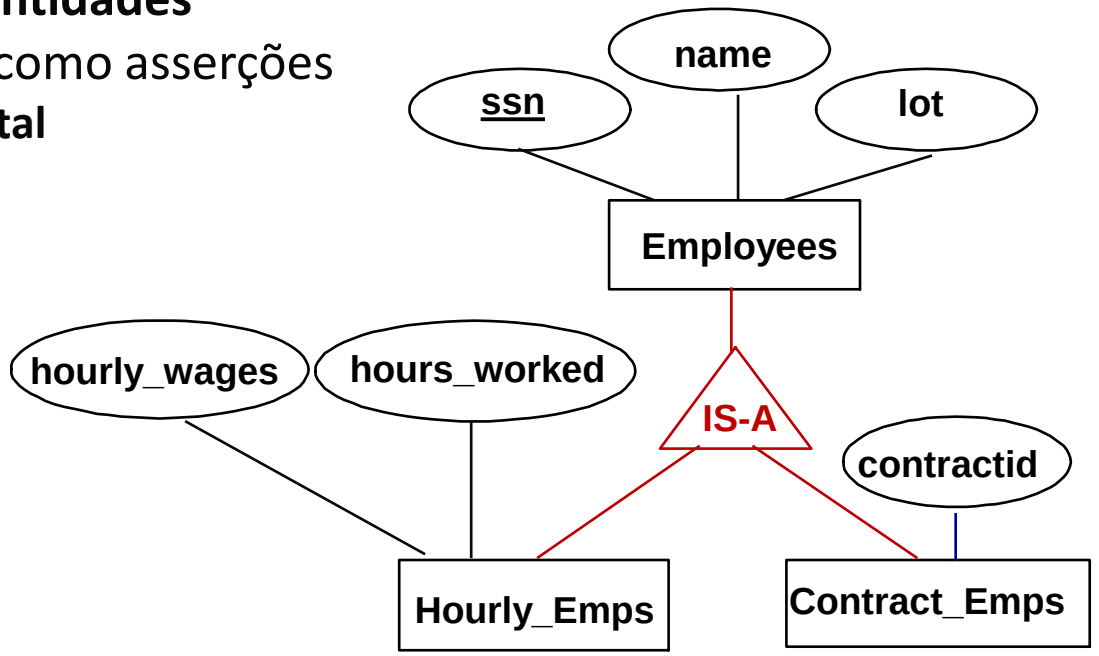
Generalizações para ER

Abordagem 2

Criação de tabelas **apenas para as sub-entidades**

Restrições de cobertura e sobreposição como asserções

Apenas aplicável quando existe **cobertura total**




```
CREATE TABLE Hourly_Emps (
 ssn   CHAR(11),
 name CHAR(30),
 lot   INTEGER,
 hours_worked INTEGER,
 hourly_wages REAL,
 PRIMARY KEY (ssn))

CREATE TABLE Contract_Emps (
 ssn   CHAR(11),
 name CHAR(30),
 lot   INTEGER,
 contractid INTEGER,
 PRIMARY KEY (ssn))
```

Generalizações para ER

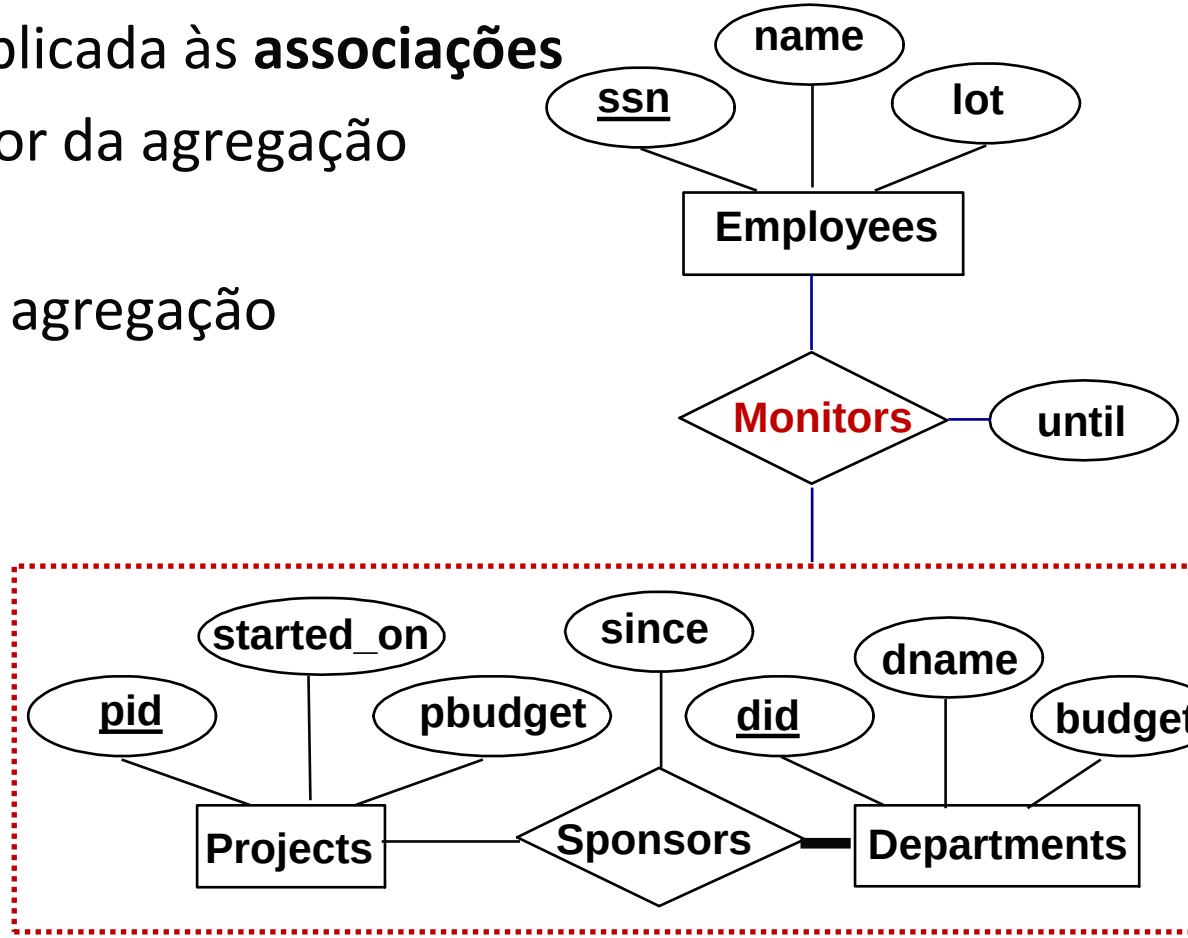
Duas abordagens

1. Criação de tabelas para a super-entidade e sub-entidades
   - Sempre aplicável
   - Necessário consultar **duas tabelas** para obter todos os dados de cada sub-entidade
2. Criação de tabelas apenas para as sub-entidades
   - Mais eficiente para interrogações a sub-entidades específicas
   - Apenas aplicável quando existe **cobertura total**

Restrições de cobertura e sobreposição como asserções

Agregações para ER

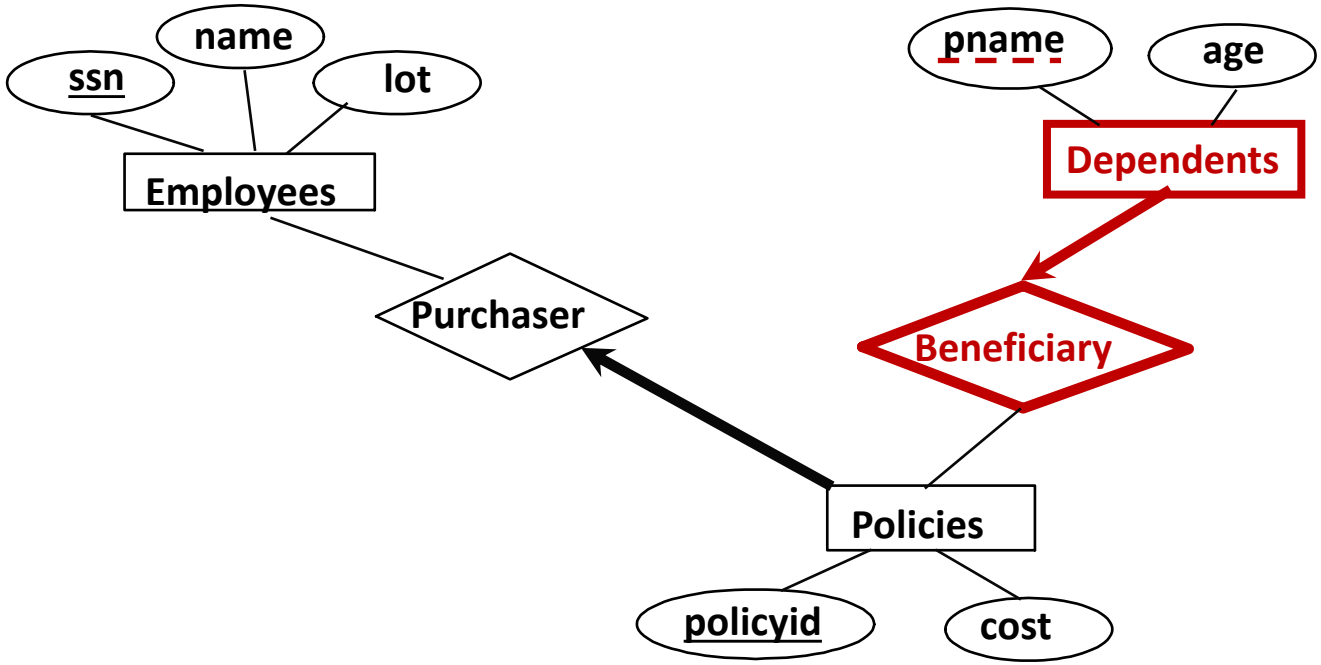
- Passagem **semelhante** à aplicada às **associações**
- Primeiro traduz-se o interior da agregação
  - Sponsors
- Depois a associação com a agregação
  - Monitors



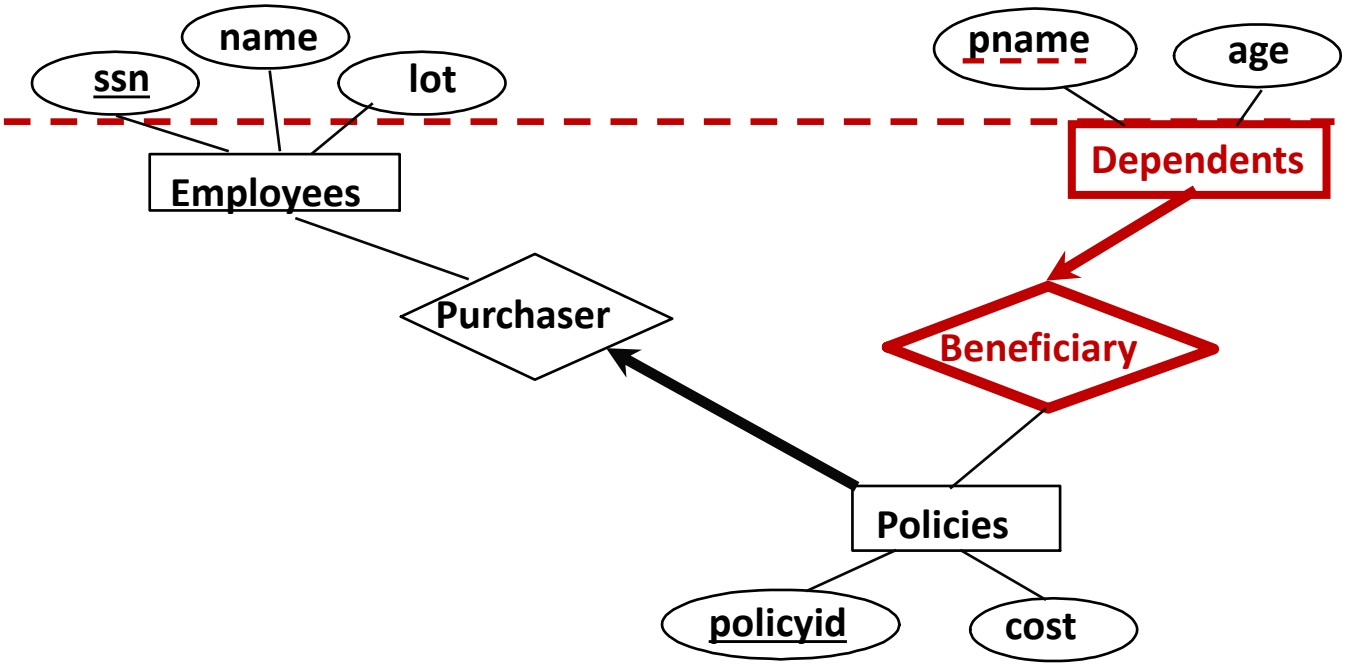

```
CREATE TABLE Monitors (
 ssn CHAR(11),
 pid INTEGER,
 did INTEGER,
 until CHAR(11),
 PRIMARY KEY (ssn,pid,did),
 FOREIGN KEY (ssn) REFERENCES Employees,
 FOREIGN KEY (pid,did) REFERENCES Sponsors)
```

Exercício 1

Passagem do EA para ER



Exercício 1

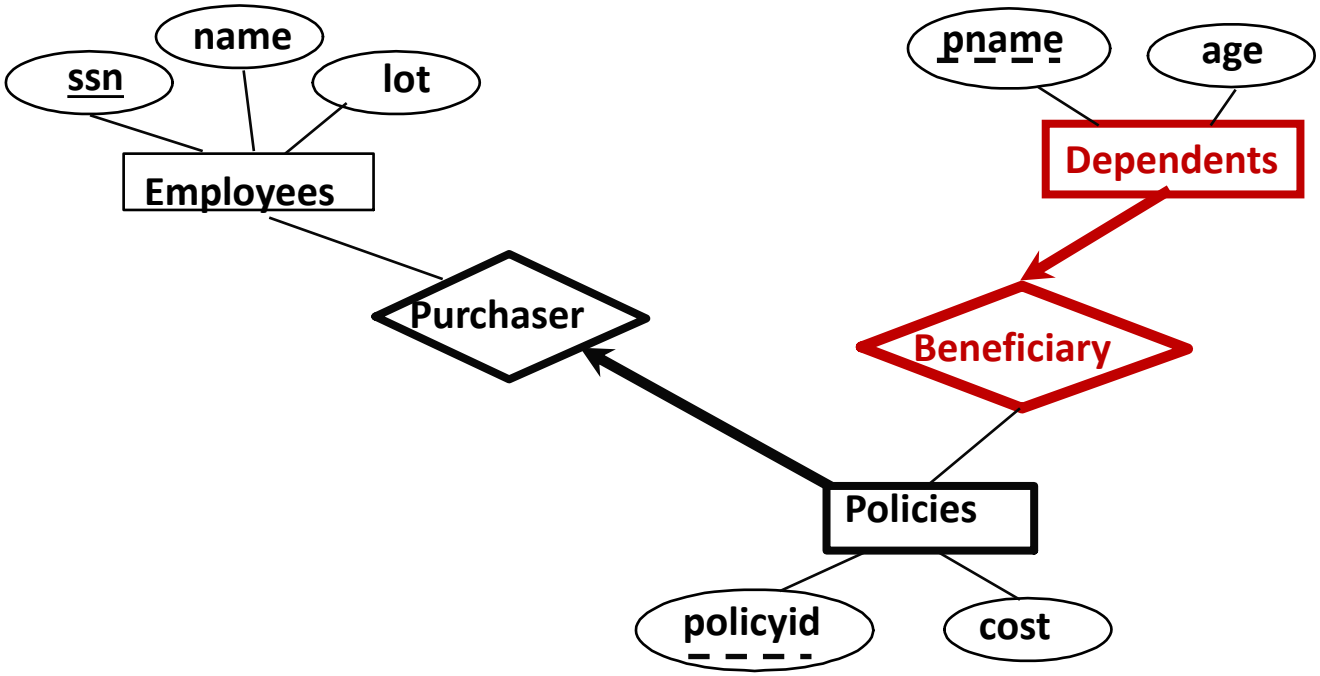



```
CREATE TABLE Policies (
 policyid INTEGER,
 cost REAL,
 ssn CHAR (11) NOT NULL,
 PRIMARY KEY (policyid),
 FOREIGN KEY (ssn)
    REFERENCES Employees
    ON DELETE CASCADE )
```

```
CREATE TABLE Dependents (
 pname CHAR(20),
 age INTEGER,
 policyid INTEGER,
 PRIMARY KEY (pname, policyid),
 FOREIGN KEY (policyid)
    REFERENCES Policies
    ON DELETE CASCADE)
```

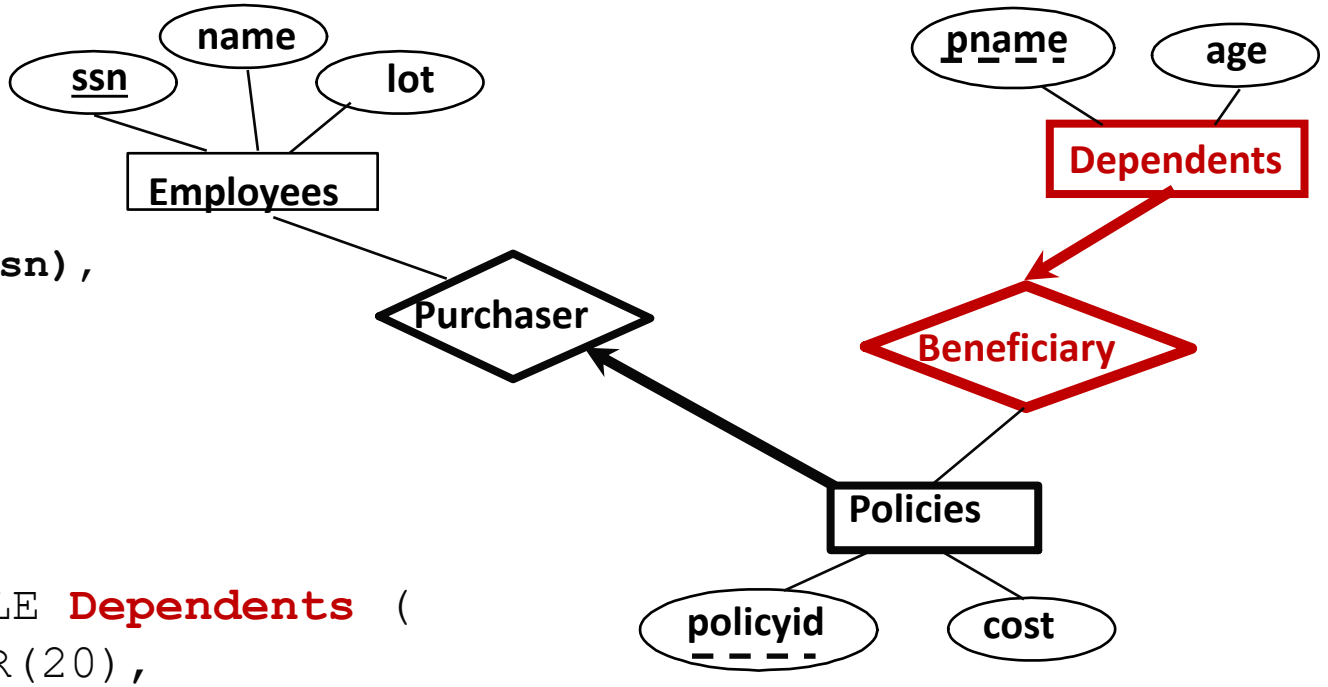
Exercício 2

Passagem do EA para ER



Exercício 2

```
CREATE TABLE Policies (
 policyid INTEGER,
 cost REAL,
 ssn CHAR (11),
 PRIMARY KEY (policyid, ssn),
 FOREIGN KEY (ssn)
    REFERENCES Employees
    ON DELETE CASCADE )
```




```
CREATE TABLE Dependents (
 pname CHAR(20),
 age INTEGER,
 ssn CHAR (11),
 policyid INTEGER,
 PRIMARY KEY (pname, policyid, ssn),
 FOREIGN KEY (policyid, ssn) REFERENCES Policies
                                ON DELETE CASCADE ))
```