

Bases de Dados

Modelo Relacional - III

Passagem do EA para Relacional

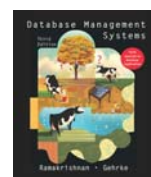
FCUL, Departamento de Informática

Ano Letivo 2015/2016

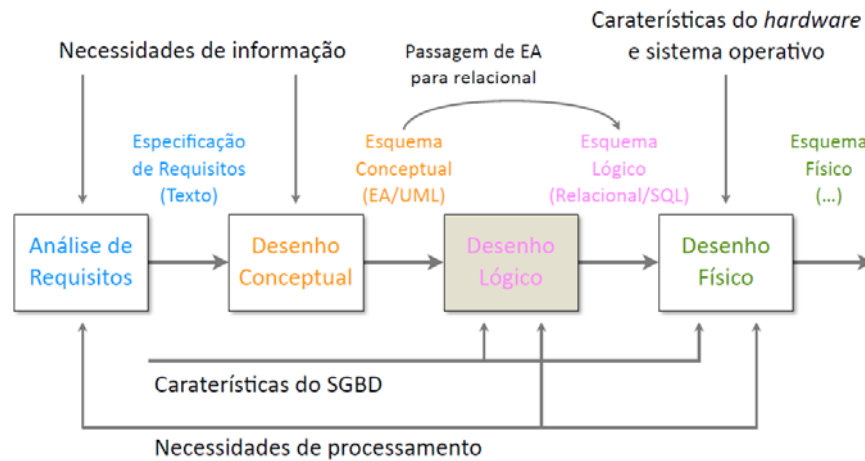
Ana Paula Afonso

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.secção 3.5**)



Processo de Desenho de BDs



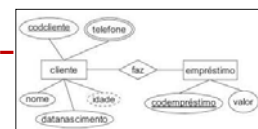
Fonte: António Ferreira, Guião SIBD, 2015

© 2015 - Docentes SI - DI/FCUL

3

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

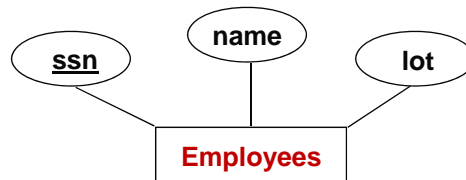


```
CREATE TABLE cliente (
  codcliente integer,
  ....)
```

© 2015 - Docentes SI - DI/FCUL

4

Entidades para ER



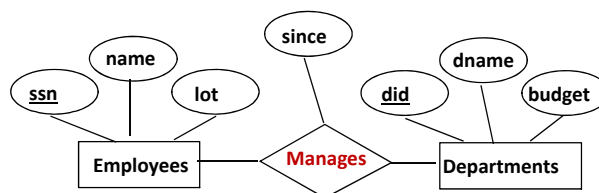
```
CREATE TABLE Employees (
  ssn CHAR(11),
  name CHAR(30),
  lot INTEGER,
  PRIMARY KEY (ssn) )
```

© 2015 - Docentes SI - DI/FCUL

5

Associações para ER

Caso 1: sem restrições de chave e participação



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

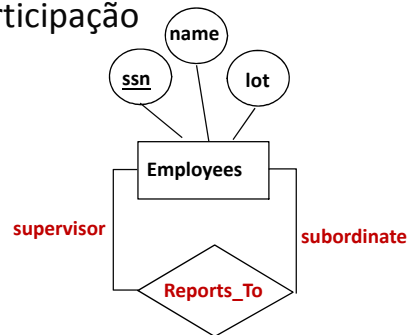
© 2015 - Docentes SI - DI/FCUL

6

Associações para ER

Caso 2: sem restrições de chave e participação

```
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)
```



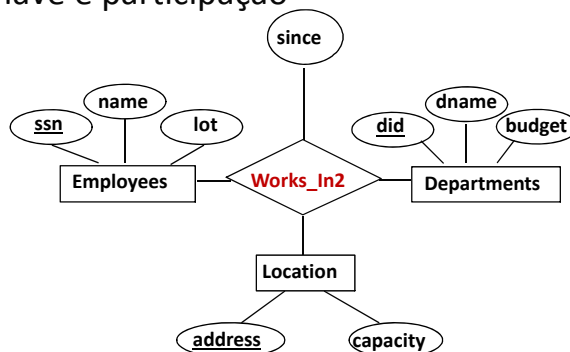
© 2015 - Docentes SI - DI/FCUL

7

Associações para ER

Caso 3: sem restrições de chave e participação

```
CREATE TABLE Works_In2 (
  ssn CHAR(11),
  did INTEGER,
  address CHAR(20),
  since DATE,
  obs CHAR(255),
  PRIMARY KEY (ssn, did, address),
  FOREIGN KEY (ssn) REFERENCES Employees,
  FOREIGN KEY (address) REFERENCES Location,
  FOREIGN KEY (did) REFERENCES Departments)
```



© 2015 - Docentes SI - DI/FCUL

8

Associações com Restrições de Chave

Duas abordagens:

1. Criar uma nova tabela

- Usar em casos de participação opcional
- e/ou em associações com muitos atributos

2. Adicionar uma coluna (chave estrangeira) à tabela com restrição

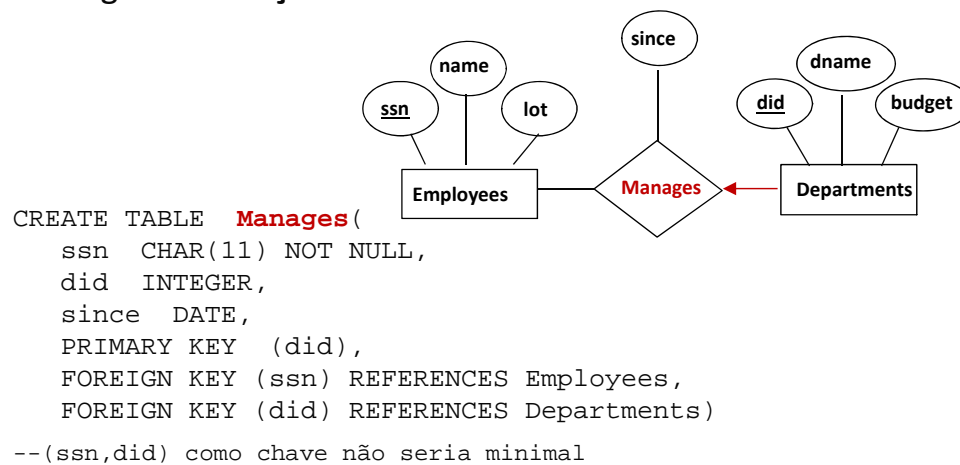
V/D – eficiência de queries vs espaço

© 2015 - Docentes SI - DI/FCUL

9

Associações com Restrições de Chave

Abordagem 1: Criação de uma nova tabela

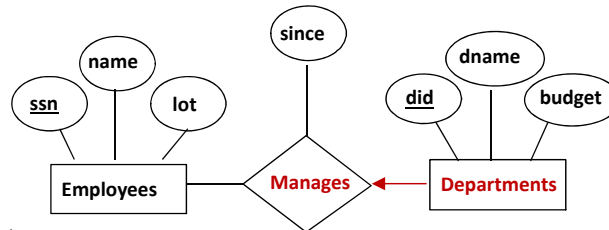


© 2015 - Docentes SI - DI/FCUL

10

Associações com Restrições de Chave

Abordagem 2: **Adição de chave estrangeira** à tabela com restrição



```

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

© 2015 - Docentes SI - DI/FCUL

11

Associações com Restrição de Chave e Participação

Opções:

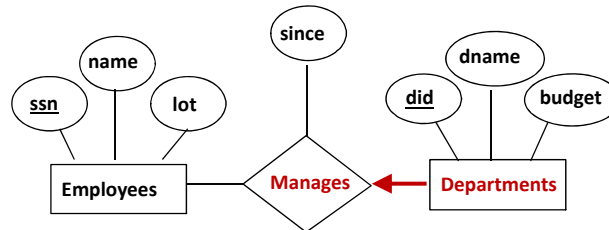
1. Adicionar a restrição de NOT NULL à coluna
 - Só aplicável se a associação for representada com uma nova coluna
2. Adicionar uma asserção (RIA)
 - Menos eficiente

© 2015 - Docentes SI - DI/FCUL

12

Associações com Restrição de Chave e Participação

Opção 1: **Adição de chave estrangeira** à tabela com restrição



```
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)
```

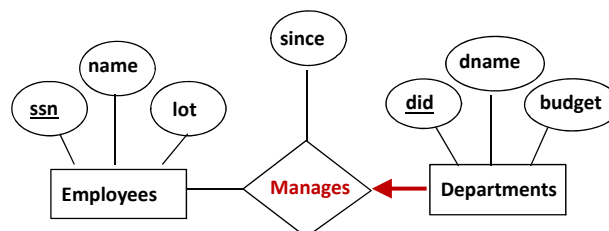
NO ACTION – ação por defeito
Um empregado não pode ser removido
se tiver um Dept_Mgr a referenciá-lo

© 2015 - Docentes SI - DI/FCUL

13

Associações com Restrição de Chave e Participação

Opção 2: Criação de uma tabela e **adição de uma asserção**



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

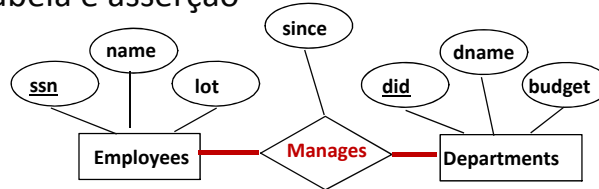
RIA: Todos os Departments têm de estar referenciados em Manages

© 2015 - Docentes SI - DI/FCUL

14

Associação 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)
```

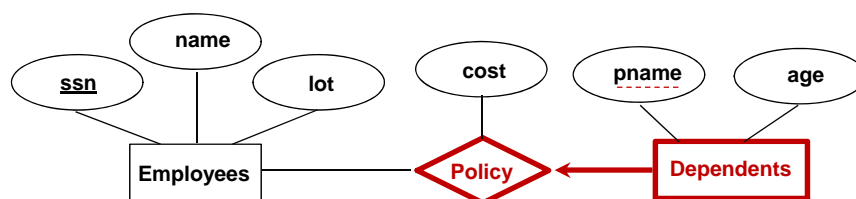
Asserção é uma restrição de integridade textual

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

© 2015 - Docentes SI - DI/FCUL

15

Entidade Fracas



```
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)

© 2015 - Docentes SI - DI/FCUL

16

Generalizações

Opções

1. Criação de novas tabelas para a super-entidade e sub-entidades
 - Sempre aplicável
2. Criação de novas tabelas apenas para as sub-entidades
 - Mais eficiente para interrogações a sub-entidades específicas
 - Apenas aplicável quando existe **cobertura total**
3. Criação de apenas uma tabela para a super e sub-entidades
 - Mais eficiente para interrogações a várias sub-entidades ao mesmo tempo

Restrições de cobertura e sobreposição como asserções

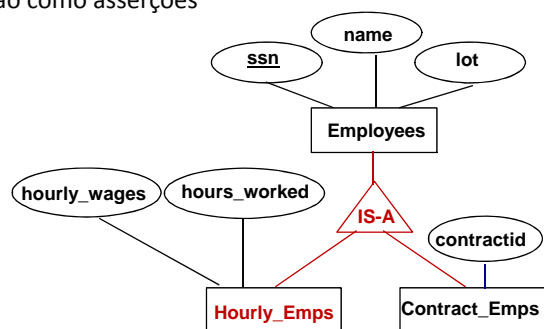
© 2015 - Docentes SI - DI/FCUL

17

Generalização: opção 1

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

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



© 2015 - Docentes SI - DI/FCUL

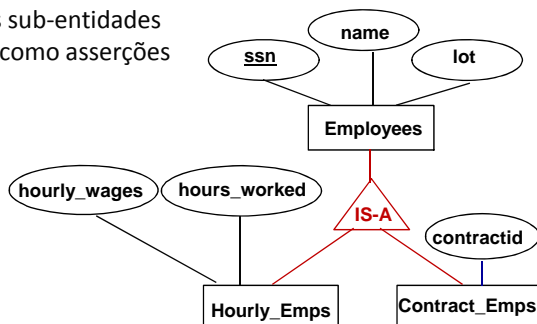
18

Generalização: opção 2

Criação de novas tabelas apenas para as sub-entidades
Restrições de cobertura e sobreposição como asserções

```
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))
```



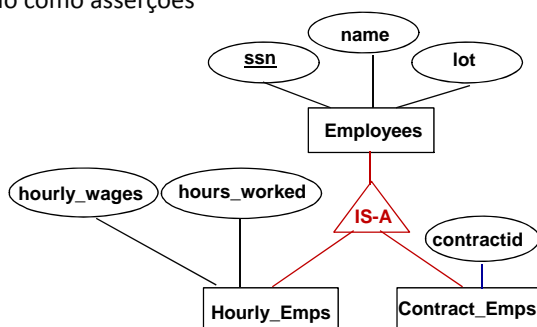
© 2015 - Docentes SI - DI/FCUL

19

Generalização: opção 3

Criação de apenas uma tabela para a super-entidade e sub-entidades
Restrições de cobertura e sobreposição como asserções

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



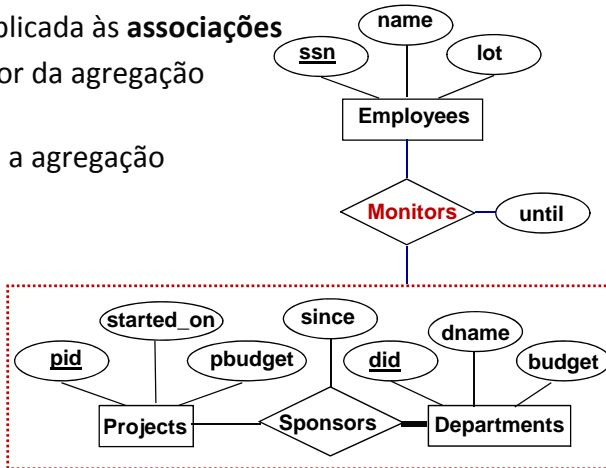
© 2015 - Docentes SI - DI/FCUL

20

Agregações

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

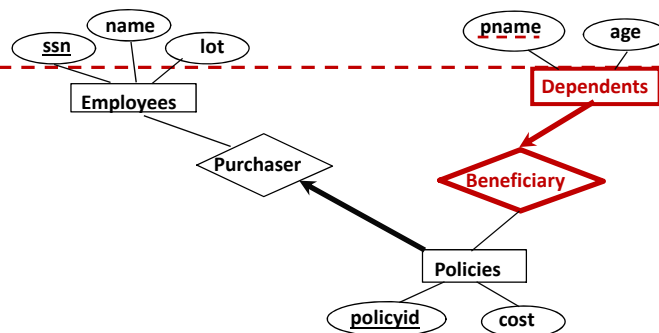
```
CREATE TABLE Monitors (
  ssn CHAR(11),
  pid INTEGER,
  did INTEGER,
  until CHAR(11)
  PRIMARY KEY (ssn,pid,did)
  FOREIGN KEY (ssn) REFERENCES Employees
  FOREIGN KEY (pid,did) REFERENCES Sponsors)
```



© 2015 - Docentes SI - DI/FCUL

21

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)
```

© 2015 - Docentes SI - DI/FCUL

22

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 ))
```

