ER PARA LÓGICO

Tradução do modelo Conceptual (Diagrama E-R) no modelo lógico

Telmo Matos: tpm@isep.ipp.pt

ER para lógico

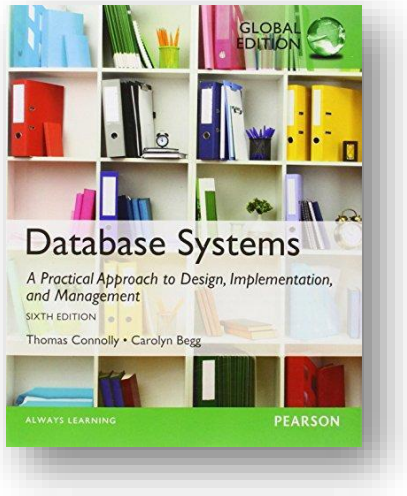
- Podem originar três tipos de tabelas:
  - Tabela com a mesma informação que a entidade (do modelo ER) original da qual foi derivada
  - Tabela contendo a chave estrangeira do relacionamento com a entidade associada;
  - Tabela derivada do relacionamento, contendo as chaves estrangeiras de todas as entidades envolvidas no relacionamento

ER para lógico

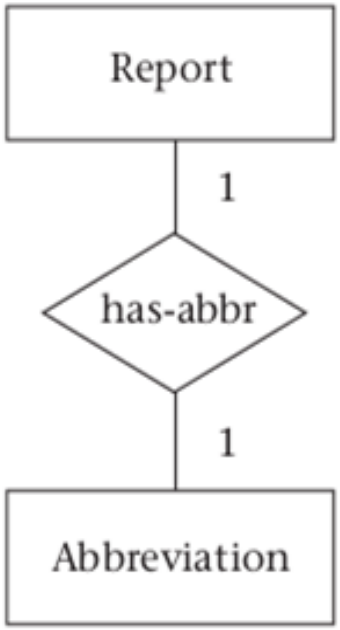
- As regras a seguir aplicam-se ao tratamento de valores nulos em SQL nessas transformações
  - Nulos são permitidos numa tabela para chaves estrangeiras de entidades opcionais associadas (referenciadas)
  - Nulos não são permitidos em uma tabela SQL para chaves estrangeiras de entidades obrigatórias associadas (referenciadas)
  - Nulos não são permitidos para nenhuma chave em uma tabela derivada de um relacionamento muitos-para-muitos, porque apenas entradas de linha completas são significativas na tabela

ER para lógico

- *Database Systems - A Practical Approach to Design, Implementation, and Management*, Thomas Connolly & Carolyn Begg; Pearson 2015; 6th Edition (Global Ed.); ISBN: 978-1-292-06118-4
  - Capítulo: **5**



ER para lógico

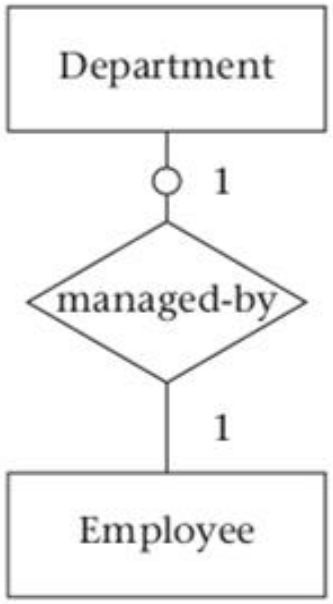


Every report has one abbreviation, and every abbreviation represents exactly one report.

```
create table report
        (report_no integer,
         report_name varchar(256),
         primary key(report_no);
create table abbreviation
        (abbr_no char(6),
         report_no integer not null unique,
         primary key (abbr_no),
         foreign key (report_no) references report
             on delete cascade on update cascade);
```

(a) One-to-one, both entities mandatory

ER para lógico

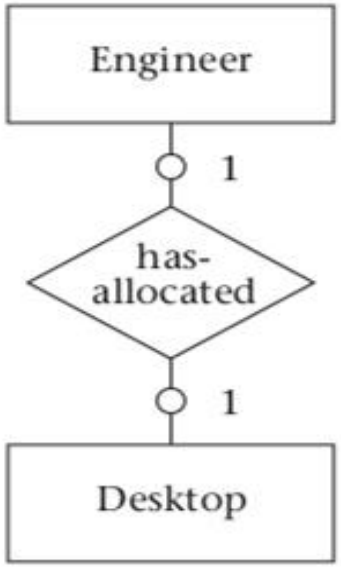


Every department must have a manager, but an employee can be a manager of at most one department.

```
create table department
        (dept_no integer,
         dept_name char(20),
         mgr_id char(10) not null unique,
         primary key (dept_no)
         foreign key (mgr_id) references employee
             on delete set default on update cascade);
create table employee
        (emp_id char(10),
         emp_name char(20)
         primary key (emp_id));
```

(b) One-to-one, one entity optional, one mandatory

ER para lógico

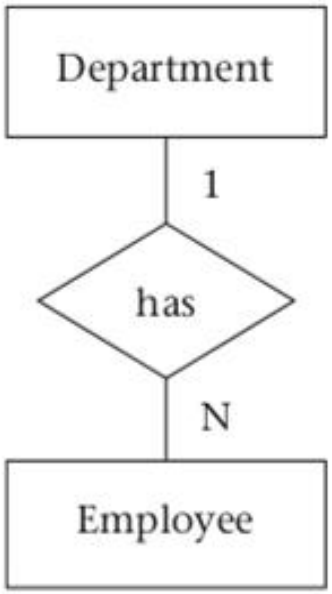


Some desktop computers are allocated to engineers, but not necessarily to all engineers.

```
create table engineer
        (emp_id char(10),
         desktop_no integer,
         primary key (emp_id));
create table desktop
        (desktop_no integer,
         emp_id char(10)
         primary key (desktop_no)
         foreign key (emp_id) references engineer
            on delete set null on update cascade);
```

(c) One-to-one, both entities optional

ER para lógico

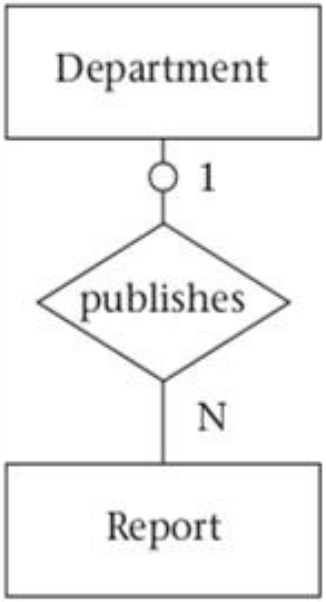


Every employee works in exactly one department, and each department has at least one employee.

```
create table department
        (dept_no integer,
         dept_name char(20),
         primary key (dept_no));
create table employee
        (emp_id char(10),
         emp_name char(20),
         dept_no integer not null,
         primary key (emp_id),
         foreign key (dept_no) references department
            on delete set default on update cascade)
```

(d) One-to-many, both entities mandatory

ER para lógico



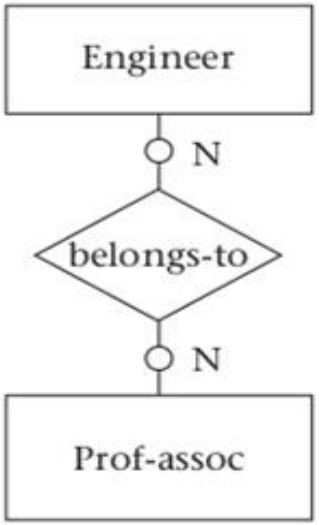
Each department publishes one or more reports. A given report may not necessarily be published by a department.

```
create table department
        (dept_no integer,
         dept_name char(20),
         primary key (dept_no));

create table report
        (report_no integer,
         dept_no integer,
         primary key (report_no),
         foreign key (dept_no) references department
             on delete set null on update cascade);
```

(e) One-to-many, one entity optional, one mandatory

ER para lógico



(f) Many-to-many, both entities optional

Every professional association could have none, one, or many engineer members. Each engineer could be a member of none, one, or many professional associations.

```
create table engineer
        (emp_id char(10),
         primary key (emp_id));
create table prof_assoc
        (assoc_name varchar(256),
         primary key (assoc_name));
create table belongs_to
        (emp_id char(10),
         assoc_name varchar(256),
         primary key (emp_id, assoc_name),
         foreign key (emp_id) references engineer
            on delete cascade on update cascade,
         foreign key (assoc_name) references prof_assoc
            on delete cascade on update cascade);
```

ER PARA LÓGICO

Tradução do modelo Conceptual (Diagrama E-R) no modelo lógico

Telmo Matos: tpm@isep.ipp.pt