

Prof. Dr. Ermir Rogova

Të dhënat e korporatave dhe menaxhimi i databazave

Pjesa 4



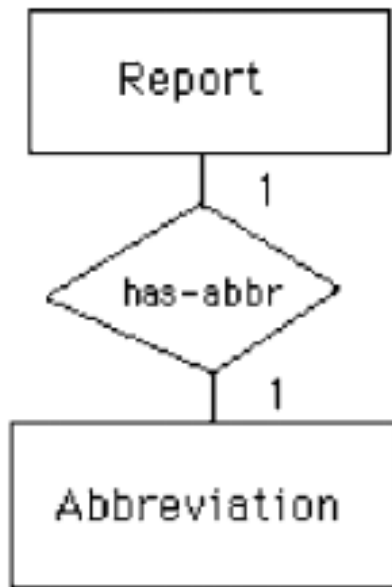
Transformimi ER (UML) → SQL

- ▶ Transformimet themelore mund të përshkruhen në bazë të tri llojeve të tabelave që ato prodhojnë:
 - Tabela me të njëjtën përmbajtje sikur entiteti prej të cilës derivojnë
 - Tabela me çelësin e huaj të entitetit prind
 - Tabela të derivuara nga marrëdhënie, që përban çelësat e huaj të të gjitha entiteteve në marrëdhënie

Rregullat mbi vlerat NULL

- ▶ Në tabela, vlerat NULL lejohen për çelësa të huaj që iu referohen entiteteve opcionale
- ▶ Në tabela, vlerat NULL nuk lejohen për çelësa të huaj që iu referohen entiteteve të detyrueshme (mandatore)
- ▶ Në tabela të derivuara nga marrëdhëniet M–M, vlerat NULL nuk lejohen për çfardo çelësi, sepse vetëm rreshtat komplet nuk humbin kuptimin

1 – 1 (i detyrueshëm në të dy anët)



(a)

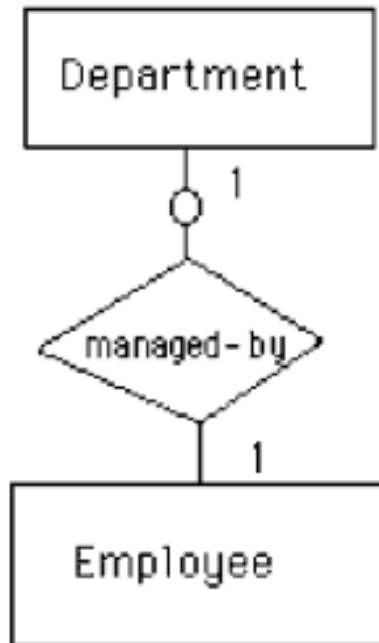
Every report has one abbreviation, and every abbreviation represents exactly one report.

```
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);
```

- ▶ Kur të dyja entitetet janë mandatore, secili entitet shndërrohet në tabelë dhe çelësi primar i cilësdo tabelë mund të paraqitet në tabelen tjetër si çelës i huaj

1 – 1 (i detyrueshëm / opcional)



(b)

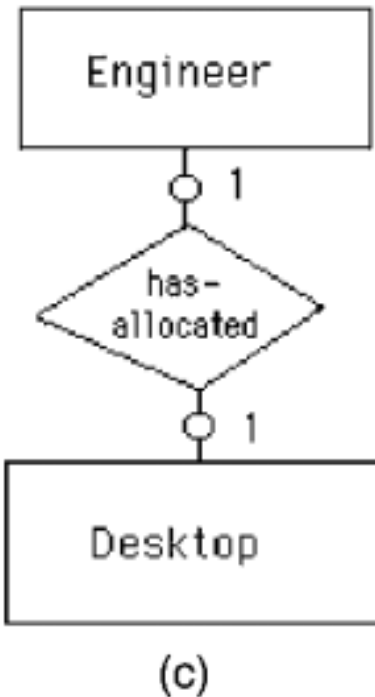
Every department must have a manager, but an employee can be a manager of at most one department.

```
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));
```

Tabela e anës opcionale duhet të mbajë çelësin primar të tabelës tjetër si qelës të huaj

1 – 1 (opcional në të dy anët)



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));
```

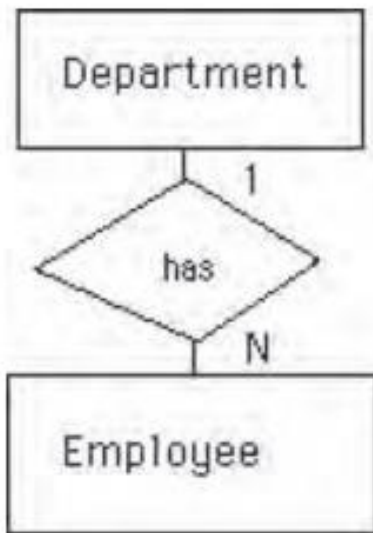
```
create table desktop
(desktop_no integer,
 emp_id char(10),
 primary key (desktop_no),
 foreign key (emp_id) references engineer
 on delete set null on update cascade);
```

- ▶ Kur të dyja anët janë opcionale, cilado tabelë mund ta marrë qelsin primar të tabelës tjetër si qelës të huaj dhe aty lejohen vlerat NULL

Marrëdhëniet 1 – M

- ▶ Marrëdhëniet 1–M mund të paraqiten qoftë të detyrueshme apo opcionale në anën M pa pasur efekt në transformim
- ▶ Ana 1 mund të jetë e detyrueshme apo opcionale.
- ▶ Në të gjitha rastet, çelësi i huaj duhet të paraqitet në anën M, e cila përfaqson entitetin fëmijë, ku vlerat NULL lejohen vetëm kur ana 1 është opcionale

1 – M (i detyrueshëm në të dy anët)



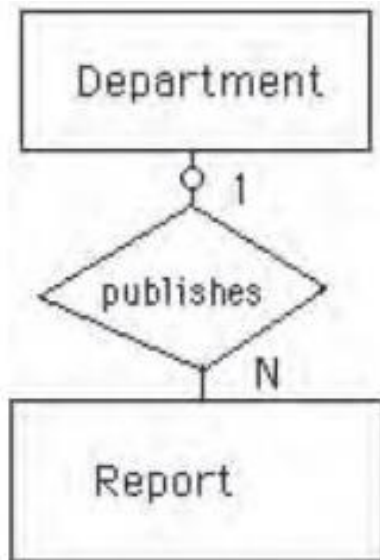
(d)

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);
```


1 – M (i detyrueshëm / opcional)



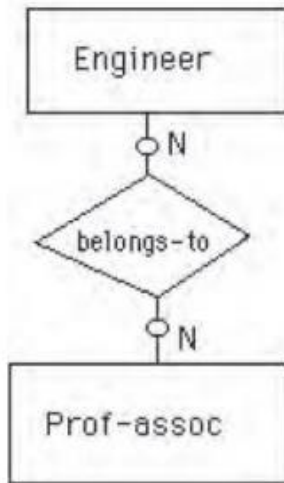
(e)

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);
```

M – M



(f)

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));

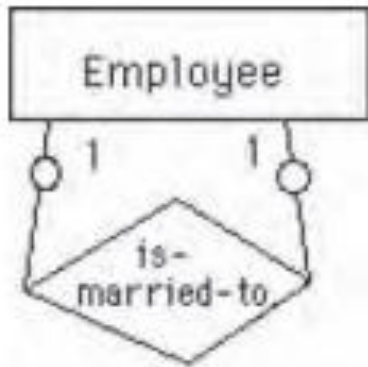
create table belongs_to
  (emp_id char(10) not null,
   assoc_name varchar(256) not null,
   primary key (emp_id, assoc_name),
   foreign key (emp_id) references engineer
    on delete cascade on update cascade,
   foreign key (assoc_name) references prof_assoc
    on delete cascade on update cascade);
```

- ▶ Marrëdhëniet M–M paraqiten si një tabelë shitesë që përmban çelësat primarë nga të dy entitetet
- ▶ Transformi i njëjtë aplikohen si në rastet e detyrueshme ashtu edhe opcionale, ku kllauza NOT NULL duhet të paraqitet medoemos për të dy çelësat primarë.

Marrëdhëniet binare rekursive

- ▶ Një entitet i vetëm me një marrëdhënie 1-1 implikon një lloj qiftëzimi ndërmjet instancave të atij entiteti.
- ▶ Ky qiftëzim mund të jetë në të dy anët opcional, i detyrueshëm apo opcional/detyrueshëm.
- ▶ Në të gjitha rastet, qelësi primar paraqitet edhe si qelës i huaj.
- ▶ Këto dy attribute janë nga i njëjti domen por ju jepen emra të ndryshëm për të treguar përdorimin e tyre
- ▶ Kufizimet e qelësit të huaj ndryshojnë sipas marrëdhënies

1 - 1 rekursive

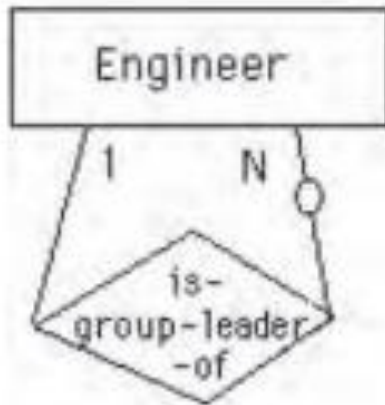


(a)

Any employee is allowed to be married to another employee in this company.

```
create table employee
(emp_id char(10),
 emp_name char(20),
 spouse_id char(10),
 primary key (emp_id),
 foreign key (spouse_id) references employee
 on delete set null on update cascade);
```

1 – M rekursive



(b)

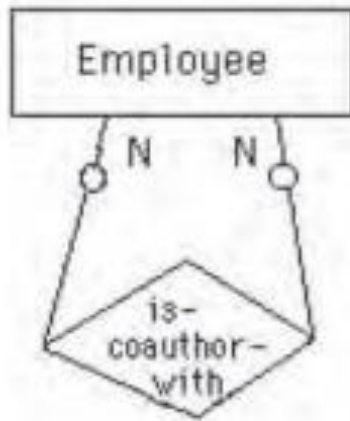
Engineers are divided into groups for certain projects. Each group has a leader.

```
create table engineer
(emp-id char(10),
 leader_id char(10) not null,
 primary key (emp_id),
 foreign key (leader_id) references engineer
 on delete set default on update cascade);
```

M – M reKursive

- ▶ Marrëdhënia M–M krijon një tabelë shtesë
- ▶ Mund të definohet si e detyrueshme ose opcionale (me përdorimin e fjalëve „duhet“ ose „mundet“).
- ▶ Në të dyja rastet çelësi primar duhet të definohet si „not null“.

M – M rekursive (2)



Each employee has the opportunity to coauthor a report with one or more other employees, or to write the report alone.

```
create table employee
(emp_id char(10),
 emp_name char(20),
 primary key (emp_id));
```

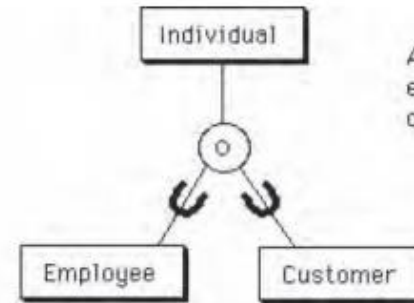
```
create table coauthor
(author_id char(10) not null,
 coauthor_id char(10) not null,
 primary key (author_id, coauthor_id),
 foreign key (author_id) references employee
 on delete cascade on update cascade,
 foreign key (coauthor_id) references employee
 on delete cascade on update cascade);
```

(c)

Gjeneralizimi

- ▶ Transformimi i gjeneralizimit mund të prodhojë tabela të veqanta për supertipin dhe nëntipet

- ▶ Tabela e derivuar nga supertipi përmban çelësin primar si dhe të gjitha atributet e përbashkëta



An individual may be either an employee or a customer, or both, or neither.

```
create table individual (indiv_id char(10),
                          indiv_name char(20),
                          indiv_addr char(20),
                          primary key (indiv_id));

create table employee (emp_id char(10),
                         job_title char(15),
                         primary key (emp_id),
                         foreign key (emp_id) references individual
                         on delete cascade on update cascade);

create table customer (cust_no char(10),
                          cust_credit char(12),
                          primary key (cust_no),
                          foreign key (cust_no) references individual
                          on delete cascade on update cascade);
```

- ✘ Secila tabelë e derivuar nga nëntipet përmban çelësin primar të supertipit dhe vetëm ato attribute të cilat janë të veçanëta për atë nëntip

Generalizimi (2)

- ▶ Në qasje tjetër është të vendosen të gjitha atributet nga supertipi dhe nëntipet në një tabelë të vetme me vlera null aty ku nevojiten
- ▶ Një qasje tjetër është nga një tabelë për secilën nëntip që përmbajnë atributet e përbashkëta si dhe ato specifike.

ENTITETET E DOBËTA

- ▶ Entitetet e dobëta ndryshojnë nga entitetet normale vetëm se iu nevojiten çelësat nga entitetet amë.
- ▶ Përdryshe ata kanë të njëjtat mënyra të transformimit dhe nuk ka nevojë për rregulla të veqanta.

Hapat e transformimit

- ▶ Lista e mëposhtme përmbledh hapat themelor për transformimin e diagramit ER në tabela:
 - Transformo secilin entitet në tabelë që përmban çelësin primar dhe atributet e tjera të atij entiteti.
 - Transformo secilën marrëdhënie binare ose rekursive M-M në tabelë me çelësa të entiteteve dhe atributet që rrjedhin prej marrëdhënies.
 - Transformo secilën marrëdhënie trinare apo n-are në tabelë.

Përmbledhje

- ▶ Entitetet, atributet dhe marrëdhëniet mund të transformohet direkt në tabela me anë të disa rregulave të thjeshta.
- ▶ Entitetet transformohen në tabela kurse atributet në kolona të atyre tabelave
- ▶ Tabelat që përfaqsojnë entitetet „fëmijë“ (ana M) e marrëdhënies, duhet të përmbajnë si qelës të huaj, çelësin primar të anës „prind“ (ana 1).
- ▶ Marrëdhëniet M–M transformohen në tabla të cilat përmbajnë çelësat primarë të entiteteve që i asociojnë si çelës primar i përbërë (composite); komponentet e këtij qelësi janë poashtu edhe qelësa të huaj
- ▶ Rregullat për gjeneralizim kërkojnë që gjatë transformimit, çelësi primar të trashëgohet prej supertipi te nëntipet
- ▶ Kufizuesit e opcionalitetit përkthehen në lejimin apo jo të vlerave NULL në qelës të huaj gjatë krijimit të tabelave

Pyetje ???