

# ER e EER -> modello relazionale

---

Lucia Ferrari

lucia02.ferrari@edu.unife.it

Modello relazionale a partire dal  
modello ER

---

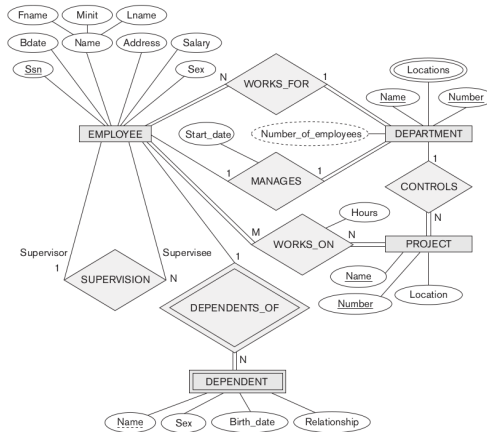


Figure 1: modello ER

# Modello relazionale

Modello relazionale corrispondente:

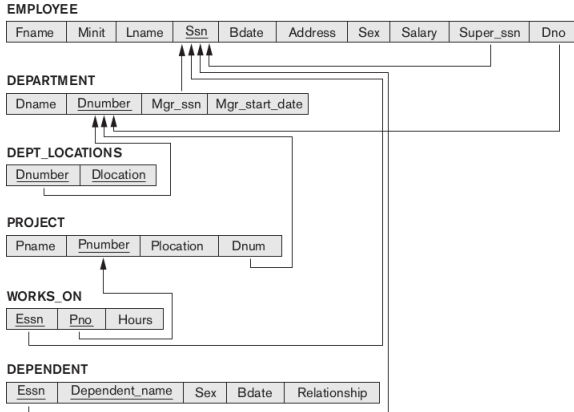


Figure 2: Risultato della trasformazione da modello ER a modello relazionale

## 1) Trasformazione delle entità forti

Per ogni entità forte E presente nel modello ER si crea una relazione R che:

- include tutti gli attributi semplici di E
- include gli attributi semplici che formano quelli composti
- Sceglie una delle chiavi primarie di E come chiave primaria di R
- Se la chiave primaria di E è composta allora l'insieme dei suoi attributi semplici forma la chiave primaria per R

### EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary
-------	-------	-------	------------	-------	---------	-----	--------

### DEPARTMENT

Dname	<u>Dnumber</u>
-------	----------------

### PROJECT

Pname	<u>Pnumber</u>	Plocation
-------	----------------	-----------

Figure 3: Risultato della trasformazione delle entità forti

## 2) Trasformazione delle entità deboli

Per ogni entità debole *W* del modello ER con entità forte *E* si crea una relazione *R* che:

- include tutti gli attributi semplici (o quelli semplici degli attributi composti) di *W* come attributi di *R*
- include come foreign key di *R* la chiave primaria della RELAZIONE creata con *E*
- include come chiave primaria di *R* l'insieme dato da: chiave primaria di *E* e chiave parziale di *W*

### DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

Figure 4: Risultato della trasformazione delle entità deboli (*Essn* è anche FK, collegata ad *Employee*)

### 3) Trasformazione delle associazioni 1 : 1

Per ogni associazione binaria 1 : 1  $R$  nel modello ER, si identificano le relazioni  $S$  e  $T$  che partecipano in  $R$ . Poi esistono diversi approcci:

1. Scelgo una delle due relazioni, ad esempio  $S$  e includo come foreign key in  $S$  la chiave primaria di  $T$ . Tutti gli attributi semplici/composti di  $R$  vengono inseriti come attributi semplici di  $S$ . Come relazione ' $S$ ' è meglio scegliere quella che ha una partecipazione totale in  $R$ .
2. Creazione di un'unica relazione data unendo  $S$  e  $T$  in una unica. Può essere fatto solo quando entrambe le entità hanno una partecipazione totale alla relazione.
3. Creazione di una terza relazione  $R$  che include come foreign keys le chiavi primarie di  $S$  e  $T$ . Gli attributi semplici sono quelli dell'associazione di partenza  $R$ .

## 4) Trasformazione delle associazioni 1 : N e N : 1

1. Per ogni associazione 1 : N di  $R$ , si identifica la relazione  $S$  che rappresenta l'entità al lato N di  $R$  e includi come foreign key di  $S$  la chiave primaria di  $T$  (che rappresenta l'entità che partecipa al lato 1). Gli attributi semplici di  $R$  vengono inclusi nella relazione  $S$ .
2. Creazione di una terza relazione  $R$  che include come foreign keys le chiavi primarie di  $S$  e  $T$ . Gli attributi semplici sono quelli dell'associazione di partenza  $R$ .

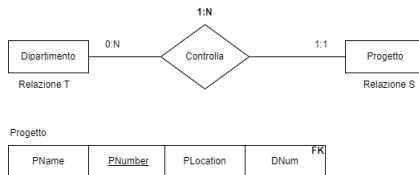


Figure 5: modello ER



## 5) Trasformazione di associazioni $M : N$

Ogni relazione  $M : N$   $R$ , crea una nuova relazione  $S$  per rappresentare  $R$ . Include come foreign key di  $S$  le chiavi primarie delle relazioni create a partire dalle entità partecipanti in  $R$  che diventano anche primary key di  $S$ . Tutti gli attributi semplici/composti di  $R$  vengono aggiunti a  $S$

### WORKS\_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

Figure 6: Risultato

## 6) Trasformazione attributi multivalore

Per ogni attributo multivalore A si crea una nuova relazione R che include un attributo corrispondente ad A e la chiave primaria K (dell'entità a cui è collegato A) come foreign key.

La chiave primaria è data dalla combinazione di A e K.

Se l'attributo è multivalore vengono incluse le sue componenti semplici.

### DEPT\_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

Figure 7: Risultato

## 7) Trasformazione di associazioni $N$ -arie

Per ogni associazione  $n$ -aria di tipo  $R$ , con  $n > 2$  si crea una nuova relazione  $S$  per rappresentare  $R$ . Si include come foreign key di  $S$  le chiavi primarie delle relazioni che rappresentano le entità partecipanti.

Sono anche inclusi gli attributi semplici/composti dell'associazione.

La chiave primaria di  $S$  è data dall'unione delle chiavi delle entità coinvolte

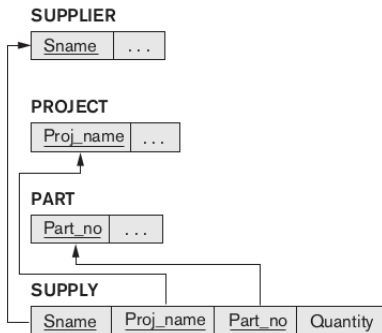


Figure 8: Risultato

### ER MODEL

Entity type

1:1 or 1:N relationship type

M:N relationship type

$n$ -ary relationship type

Simple attribute

Composite attribute

Multivalued attribute

Value set

Key attribute

### RELATIONAL MODEL

*Entity* relation

Foreign key (or *relationship* relation)

*Relationship* relation and *two* foreign keys

*Relationship* relation and  $n$  foreign keys

Attribute

Set of simple component attributes

Relation and foreign key

Domain

Primary (or secondary) key

## Mappatura del modello EER

---

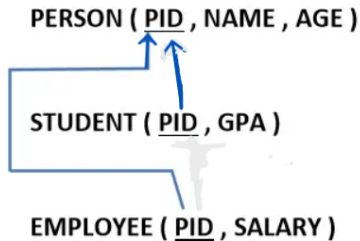
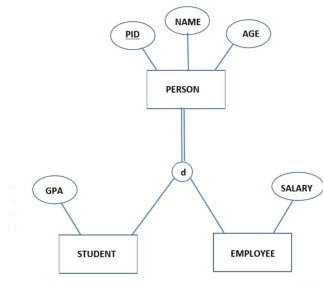
## 8) Mappatura di specializzazione/generalizzazione

Esistono diversi modi in cui mappare una specializzazione/generalizzazione.

Ogni specializzazione con  $m$  sottoclassi  $\{S_1, S_2, \dots, S_m\}$  e superclasse  $C$ , dove gli attributi di  $C$  sono  $\{k, a_1, \dots, a_n\}$  e  $k$  è la chiave primaria può essere convertita tramite le seguenti opzioni:

## Opzione: multiple relazioni

1. Multiple relazioni Crea una relazione  $L$  per  $C$  (la superclasse) con attributi  $Attrs(L) = \{k, a_1, \dots, a_n\}$  e  $PK(L) = k$ . Crea una relazione  $L_i$  per ogni sottoclasse  $S_i$ ,  $1 \leq i \leq m$ , con attributi:  $Attrs(L_i) = \{k\} \cup \{\text{attributi di } S_i\}$  e  $PK(L_i) = k$ . La chiave primaria di ogni sottoclasse è anche foreign key verso la superclasse. Questo metodo funziona con ogni tipo di specializzazione (totale/parziale, disgiunta/sovrapposta).

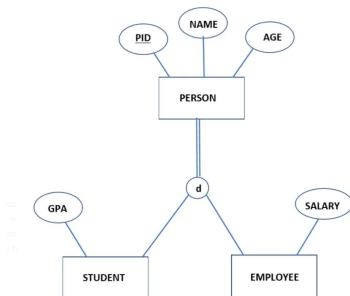


## Opzione: multiple relazioni II

2. Multiple relazioni - solo sottoclassi: crea una relazione per ognuna delle sottoclassi, con gli attributi:

$Attrs(L_i) = \{k, a_1, \dots, a_n\} \cup \{\text{attributi di } S_i\}$  e  $PK(L_i) = k$ .

Questa opzione funziona solo con le specializzazione che sono TOTALI e DISGIUNTE.



**STUDENT ( PID , GPA , NAME , AGE )**

**EMPLOYEE ( PID , SALARY , NAME , AGE )**



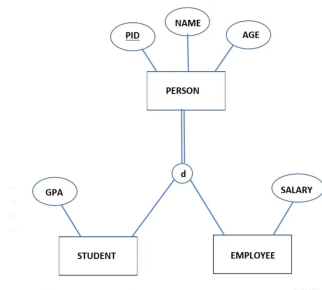
## Opzione: singole relazioni

3. Single relazioni con attributo tipo: crea una SINGOLA relazione  $L$  con attributi

$$Attrs(L) = \{k, a_1, \dots, a_n\} \cup \{\text{attributi delle sottoclassi}\} \cup \{t\}$$

e  $PK(L) = k$ . L'attributo  $t$  è chiamato TIPO ed è un attributo che indica a quale sottoclasse appartiene ogni tupla, se vi appartiene.

Questa opzione funziona solo per le sottoclassi che sono DISGIUNTE. Potenzialmente vengono creati molti attributi nulli.



PERSON ( PID , NAME , AGE , SALARY , GPA , PERSONTYPE )

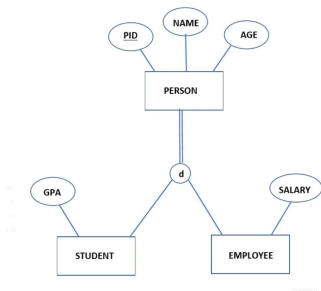
Figure 9: In questo caso PERSONTYPE può avere valore 'student' oppure 'employee' o NULL

## Opzione: singole relazioni II

1. Singole relazioni: crea una SINGOLA relazione  $L$  con attributi:

$$Attrs(L) = \{k, a_1, \dots, a_n\} \cup \{\text{attributi delle sottoclassi}\} \cup \{t_1, \dots, t_m\}$$

e  $PK(L) = k$ . Ogni  $t_i$ ,  $1 \leq i \leq m$ , è un attributo booleano che indica se la tupla appartiene a  $S_i$ . Questo funziona anche per le specializzazioni le cui sottoclassi si sovrappongono.



**PERSON ( PID , NAME , AGE , SALARY , GPA , STUDENT , EMPLOYEE )**

Figure 10: **STUDENT** e **EMPLOYEE** possono avere valore true o false. Es: student=true indica che la tupla appartiene alla sottoclasse student

Una sottoclasse condivisa da multiple superclassi indica più eridarietà. Può essere utilizzato uno qualunque dei metodi precedenti per rappresentarla.

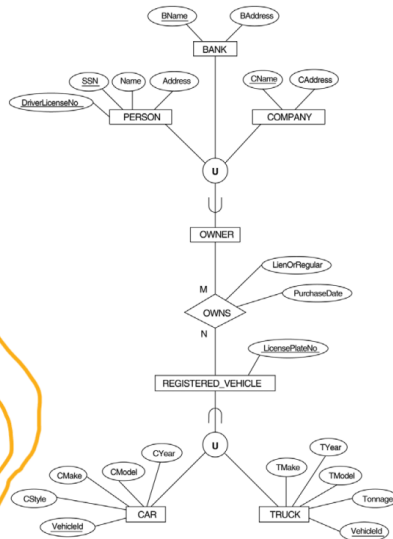
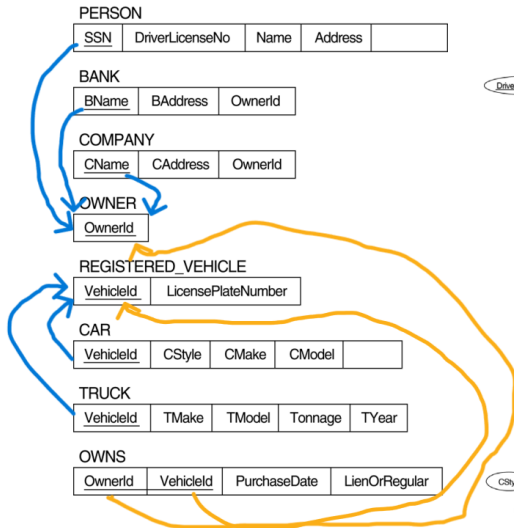
## 9) Trasformazione dei tipi unione

Un tipo unione è una SOTTOCLASSE dell'unione di due o più superclassi che possono avere chiavi diverse.

Per mappare un tipo unioni con superclassi con chiavi di tipo differente è necessario creare un nuovo attributo chiave (chiamato chiave surrogata) nel creare la relazione corrispondente al tipo unione, la chiave surrogata diventa una foreign key nelle superclassi (collegamento DA superclasse A sottoclasse).

Se la superclassi NON hanno chiavi differenti non è necessaria la chiave surrogata.

# Esempio mappatura tipi unione



Convertire in un modello relazionale i seguenti esercizi già svolti:

- Esercizio sui veicoli (slide Lezione 1 - esercizio 1)
- Esercizio ecommerce (slide Lezione 1 - esercizio 2)
- Esercizio biblioteca ( slide Lezione 2)
- Esercizio aste (slide Lezione 3)
- Esercizio medici e pazienti (slide Lezione 4)
- Esercizio Social Network (slide Lezione 4)