



**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari  
Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

**Slides A.1 (S.A.1)**

Analisi Concettuale

**Il Linguaggio Entity-Relationship**

# Indice

Queste slide sono composte dalle seguenti sottounità:

S.A.1.1. Introduzione

S.A.1.2. Entità, Attributi, Domini

S.A.1.3. Relationship

S.A.1.3.1. Concetti Base

S.A.1.3.2. Vincoli di Molteplicità

S.A.1.3.3. Entità Coinvolte in Più Relationship

S.A.1.3.4. Ruoli di Relationship

S.A.1.3.5. Attributi di Relationship

S.A.1.4. Relazioni IS-A tra Entità

S.A.1.5. Generalizzazioni tra Entità

S.A.1.6. Relazioni IS-A tra Relationship

S.A.1.7. Vincoli di Cardinalità su Attributi

S.A.1.8. Vincoli di Identificazione di Entità

S.A.1.9. Vincoli Esterni (Definizione Informale)



**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

**Slides A.1.1 (S.A.1.1)**

**Analisi Concettuale**

**Il Linguaggio Entity-Relationship**

**Introduzione**

## Il linguaggio Entity-Relationship (ER)

Il linguaggio ER permette di creare un **modello dei dati di interesse** per un'applicazione

Viene utilizzato in fase di **Analisi dei requisiti**

- ▶ ER è un **linguaggio grafico**
- ▶ fornisce dei **costrutti** (elementi di diagrammi ER)
- ▶ che vanno usati rispettando una **sintassi**
- ▶ ed a cui è associata una **semantica**

## Livello intensionale vs. livello estensionale

Un diagramma ER descrive la **struttura** dei dati, **non** i dati (che possono variare). Si dice che ER descrive il livello **intensionale** dei dati:

- ▶ livello **intensionale** (struttura) vs.
- ▶ livello **estensionale** (istanze)

Avete già visto questa distinzione, ad es.:

- ▶ **classe** (aspetto intensionale) vs. **oggetti** (aspetto estensionale) in un linguaggio di programmazione OO
- ▶ **struct** (aspetto intensionale) vs. **istanze di struct** (aspetto estensionale) in C

Ad ogni diagramma ER (livello intensionale) corrispondono in genere **più** livelli estensionali (insiemi di istanze), anche se in ogni momento, **solo uno** è quello significativo (quello che rappresenta lo stato corrente del mondo rappresentato).

## Esempio: livelli intensionale e estensionale in Java

```
// Livello intensionale dei dati: classi
public class Persona {
    String nome;
    String cognome;
    public Persona(String n, String c) {...}
    ...
}
int main(...) {
    // Livello estensionale dei dati: istanze
    Persona mario = new Persona("Mario", "Rossi");
    Persona anna = new Persona("Anna", "Bianchi");
    // $$
}
```

Degli infiniti livelli estensionali possibili dato il livello intensionale dei dati definito dalle classi, solo uno è significativo quando il programma raggiunge il punto \$\$.

# Costrutti offerti dal linguaggio ER

- ▶ Entità
- ▶ Relationship
- ▶ Attributi (di entità o relationship)
- ▶ Vincoli di integrità
- ▶ Relazioni is-a tra entità e tra relationship
- ▶ Generalizzazione tra entità



**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

### **Slides A.1.2 (S.A.1.2)**

**Analisi Concettuale**

**Il Linguaggio Entity-Relationship**

**Entità, Attributi, Domini**



## La nozione di entità

- ▶ Rappresenta una **classe** di oggetti (fatti, persone, cose) di interesse per il dominio applicativo
- ▶ Le istanze di un'entità hanno **proprietà comuni**
- ▶ Le istanze di un'entità hanno **esistenza autonoma** rispetto alle altre istanze

### Esempi

- ▶ Persona
- ▶ Impiegato
- ▶ Azienda
- ▶ Fattura

In ogni momento, ciascuna entità rappresenta un **insieme di istanze**  
 $\Rightarrow$  **astrazione**

## La nozione di attributo di entità

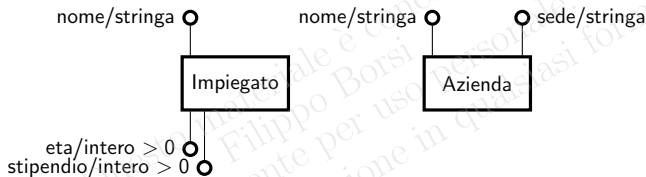
- ▶ Proprietà **locale** di un'entità di interesse per il dominio applicativo
- ▶ Associa ad ogni istanza di entità un valore in un certo **dominio** (tipo)

### Esempi:

- ▶ Attributi per l'entità **Impiegato**: **nome** (stringa), **età**, **stipendio** (interi positivi)
- ▶ Attributi per l'entità **Azienda**: **nome** (stringa), **sede** (stringa)

Il valore di un attributo di una istanza di entità dipende **solo** dall'istanza stessa, non ha (in genere) **alcun rapporto** con le altre istanze

## Rappresentazione di entità e attributi

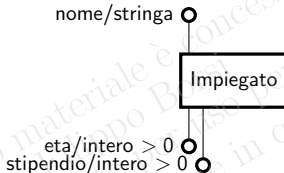


Ogni istanza di entità **Impiegato** ha associato:

- ▶ uno ed un solo valore di tipo stringa per l'attributo **nome**
- ▶ uno ed un solo valore di tipo stringa per l'attributo **eta**
- ▶ uno ed un solo valore di tipo intero  $> 0$  per l'attributo **stipendio**

Analogamente per ogni istanza di entità **Azienda**

## Istanze di entità e attributi

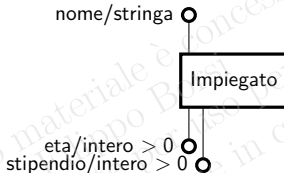


Le seguenti due istanze di **Impiegato**:

- ▶ nome = "Anna", eta = 35, stipendio = 40000
- ▶ nome = "Anna", eta = 35, stipendio = 40000

possono coesistere?

## Istanze di entità e attributi



Le seguenti due istanze di **Impiegato**:

- ▶ nome = "Anna", eta = 35, stipendio = 40000
- ▶ nome = "Anna", eta = 35, stipendio = 40000

possono coesistere?

**Sì**: le due istanze sono **distinte**, nonostante coincidano nel valore di tutti gli attributi

## Attributi: domini

Possiamo assumere che siano disponibili vari tipi elementari come domini di attributi:

- ▶ stringa
- ▶ intero
- ▶ reale
- ▶ data
- ▶ ora
- ▶ dataora

Per rendere la modellazione più aderente alla realtà, possiamo **restringere** i domini suddetti mediante **vincoli di dominio**:

- ▶ intero  $> 0$ , intero  $\geq 0$ , reale  $< 0$ , etc.
- ▶ con  $[x, y]$  indichiamo il dominio **intervallo di interi** tra  $x$  e  $y$

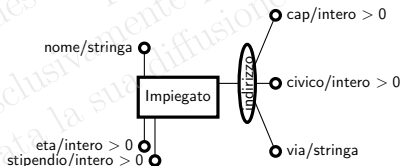
Un altro dominio che possiamo utilizzare liberamente nei diagrammi ER per l'Analisi è il

- ▶ dominio **enumerativo**, espresso come insieme di simboli dati esplicitamente, ad es.  $\{uomo, donna\}$

## Attributi composti

Possiamo inoltre specificare un attributo su un dominio di tipo **record**,  
avente campi su domini base o (a loro volta) record

⇒ **attributo composto**

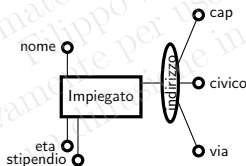


## Attributi composti

Possiamo inoltre specificare un attributo su un dominio di tipo **record**,  
avente campi su domini base o (a loro volta) record

⇒ **attributo composto**

Per semplicità, i domini degli attributi non vengono indicati nel  
diagramma ER. Essi compaiono in un **documento allegato**.



Domini per gli attributi dell'entità Impiegato:

attributo	dominio
nome	stringa
eta	intero > 0
stipendio	intero > 0
indirizzo	record(via: stringa,civico: intero > 0,cap: intero > 0)

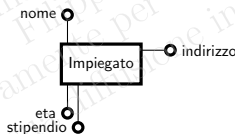


## Attributi composti

Possiamo inoltre specificare un attributo su un dominio di tipo **record**,  
 avente campi su domini base o (a loro volta) record

⇒ **attributo composto**

Anche i campi di un attributo composto possono essere omessi e indicati  
 in un documento allegato.



Domini per gli attributi dell'entità **Impiegato**:

attributo	dominio
nome	stringa
eta	intero > 0
stipendio	intero > 0
indirizzo	record(via:stringa, civico:intero > 0, cap:intero > 0)



**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

**Slides A.1.3 (S.A.1.3)**

**Analisi Concettuale**

**Il Linguaggio Entity-Relationship  
Relationship**



**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

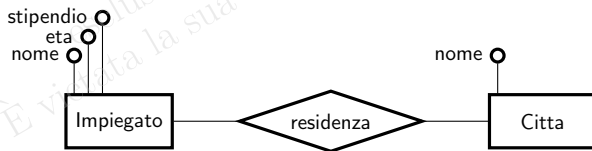
### **Slides A.1.3.1 (S.A.1.3.1)**

**Analisi Concettuale**  
**Il Linguaggio Entity-Relationship**  
**Relationship**  
**Concetti Base**

## La nozione di relationship

- ▶ Esprime la **possibilità di legami** tra istanze di due o più entità
- ▶ il numero di entità coinvolte si chiama **grado** o **arietà** della relationship

**Esempio:** “È di interesse conoscere la città di residenza degli impiegati...”



## Semantica delle relationship (prima versione)



Il **livello estensionale** di una relationship  $r$  tra le entità  $E$  ed  $F$ :

- ▶ è costituita da un **insieme di coppie**  $(e, f)$  tali che  $e$  è un'istanza di  $E$  e  $f$  è un'istanza di  $F$
- ▶ se consideriamo  $E$  ed  $F$  come gli insiemi delle loro rispettive istanze, abbiamo:

$$r \subseteq E \times F$$

ovvero  $r$  è un **sottoinsieme del prodotto cartesiano**  $E \times F$

- ▶ ovvero,  $r$  è una **relazione matematica** su  $E$  ed  $F$

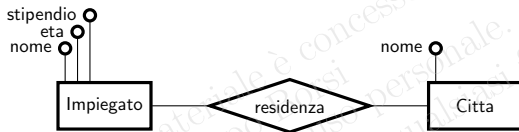
## Relationship e duplicati



- ▶  $r \subseteq E \times F$
- ▶  $r$  è un **insieme** di coppie
- ▶ gli insiemi non contengono duplicati
- ▶  $\implies$  **non possono esistere** in  $r$  due istanze che legano la **stessa coppia** di istanze di  $E$  e di  $F$
- ▶ in altre parole, conoscere il livello estensionale di  $r$  permette di rispondere alla seguente domanda:  
data  $(e, f)$  con  $e \in E, f \in F$ :  $(e, f) \in r$ ? [risposta: sì/no]  
e non di conoscere “quante volte”  $e$  è legata ad  $f$

## Relationship e duplicati (2)

### Esempio



Supponiamo che, a livello estensionale:

- ▶  $\text{Impiegato} = \{anna, mario\}$
- ▶  $\text{Citta} = \{\text{milano}, \text{roma}\}$

Due possibili livelli estensionali per *residenza*:

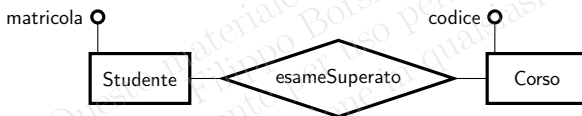
	(anna, milano)
(anna, milano)	(mario, roma)
(mario, roma)	

(anna, milano)  $\Leftarrow$  duplicato, non ammesso!

Che senso avrebbe rappresentare *due volte* che *anna* risiede a *milano*?

## Relationship e duplicati (3)

### Esempio



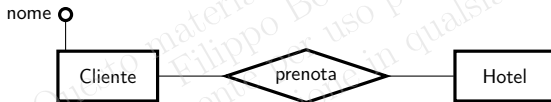
Grazie alla **restrizione** imposta dal costrutto di **relationship**, stiamo affermando che

uno stesso studente **non può** superare  
**più volte** uno stesso esame



## Relationship e duplicati (4)

### Esempio

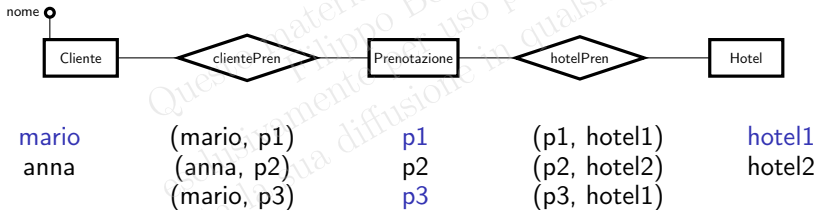


In questo caso, l'uso della relationship **impedisce** livelli estensionali che vorremmo ammettere

uno stesso cliente **non può** prenotare più volte lo stesso hotel

## Relationship e duplicati (5)

Se vogliamo distinguere le diverse prenotazioni di ogni cliente ad ogni hotel, dobbiamo usare il concetto di **entità**



In questo modo, ogni singola prenotazione è modellata come un'istanza di entità, e quindi ha vita propria, **indipendentemente** dalla coppia cliente/hotel

p1 e p3 sono **oggetti distinti**



**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

### **Slides A.1.3.2 (S.A.1.3.2)**

**Analisi Concettuale**

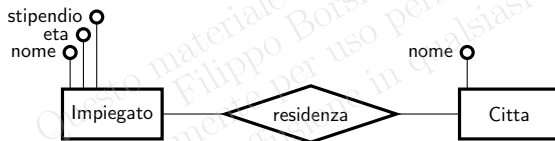
**Il Linguaggio Entity-Relationship  
Relationship**

**Vincoli di Molteplicità**

## Vincoli di integrità

I diagrammi ER visti fino ad ora sono modelli molto **laschi** della realtà di interesse.

### Esempio



Un possibile livello estensionale (ammesso!)

mario  
anna

(mario, roma)  
(mario, milano)

roma  
milano

## Vincoli di integrità (2)

### Vincoli di integrità

Regole (espresse sul diagramma) che **impongono restrizioni** ai livelli estensionali ammessi

Esistono diverse tipologie di vincoli di integrità

La prima tipologia di vincoli di integrità che vedremo esprime restrizioni sul **numero di volte** in cui un'istanza di entità può essere coinvolta in una relationship

Questi vincoli vanno sotto il nome di

**vincoli di cardinalità sulle relationship**

## Vincoli di cardinalità sulle relationship

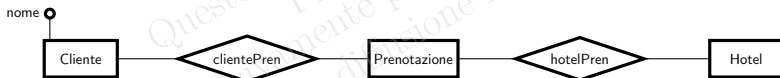


### Semantica

- ▶ **Quante** istanze di relationship (coppie impiegato/città) possono coinvolgere lo **stesso impiegato**?  
Da 1 ad 1  $\Rightarrow$  esattamente 1  
Ogni impiegato ha una ed una sola città di residenza
- ▶ **Quante** istanze di relationship (coppie impiegato/città) possono coinvolgere la **stessa città**?  
Da 0 ad N (= nessun limite)  
Ogni città può essere residenza di un numero arbitrario (anche 0) di impiegati

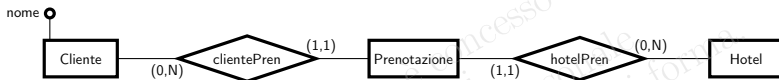
## Vincoli di cardinalità sulle relationship: esempio

Torniamo al diagramma visto che modella clienti, hotel, e prenotazioni



Quali sono i vincoli di cardinalità corretti sulle relationship?

## Vincoli di cardinalità sulle relationship: esempio (2)



- ▶ Ogni istanza di **Prenotazione** deve essere coinvolta in:
  - ▶ esattamente una istanza di **clientePren** (perché è relativa ad **un** cliente)
  - ▶ esattamente una istanza di **hotelPren** (perché è relativa ad **un** hotel)
- ▶ Ogni istanza di **Cliente** può essere coinvolta in un numero arbitrario di istanze di **clientePren** (ogni cliente può effettuare un numero **arbitrario** di prenotazioni)
- ▶ Ogni istanza di **Hotel** può essere coinvolta in un numero arbitrario di istanze di **hotelPren** (ogni hotel può ricevere un numero **arbitrario** di prenotazioni)





**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

### **Slides A.1.3.3 (S.A.1.3.3)**

**Analisi Concettuale**

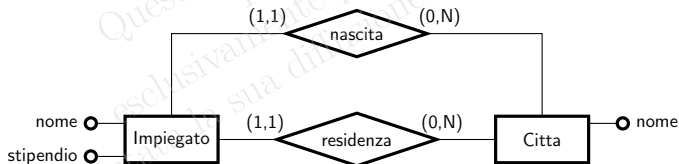
**Il Linguaggio Entity-Relationship  
Relationship**

**Entità Coinvolte in Più Relationship**

## Entità coinvolte in più relationship

Due entità possono essere coinvolte in **diverse** relationship.

**Esempio:**



Le relationship **residenza** e **nascita** esprimono **legami di tipo diverso**.

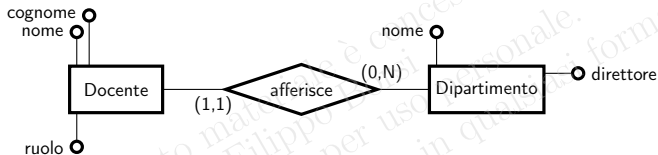
## Esercizio

Vogliamo modellare i docenti di un ateneo. Di ogni docente interessa:

- ▶ il nome, il cognome,
- ▶ il ruolo che può essere: RU (ricercatore universitario), PA (professore associato), PO (professore ordinario)
- ▶ il dipartimento a cui afferisce

Dei dipartimenti interessa il nome ed il direttore.

## Esercizio: soluzione 1



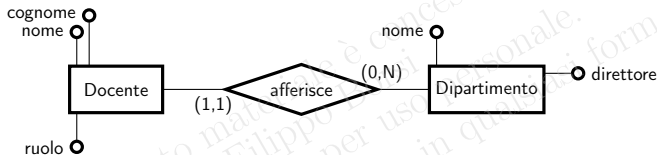
### Entità Docente

attributo	dominio
nome	stringa
cognome	stringa
ruolo	{RU, PO, PA}

### Entità Dipartimento

attributo	dominio
nome	stringa
direttore	stringa

## Esercizio: soluzione 1



### Entità Docente

attributo	dominio
nome	stringa
cognome	stringa
ruolo	{RU, PO, PA}

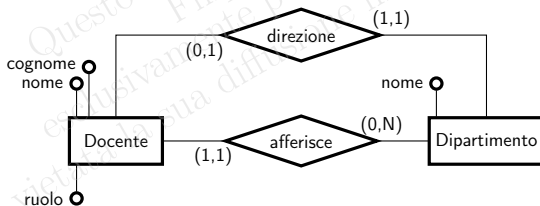
### Entità Dipartimento

attributo	dominio
nome	stringa
direttore	stringa

**Errore:** si è modellato il direttore di un dipartimento come una stringa, mentre in realtà è un'istanza dell'entità **Docente**.

## Esercizio: soluzione 2

Una soluzione **corretta**:



## Esercizio: soluzione 2 (2)

### Entità Docente

attributo	dominio
nome	stringa
cognome	stringa
ruolo	{RU, PO, PA}

### Entità Dipartimento

attributo	dominio
nome	stringa



**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

### **Slides A.1.3.4 (S.A.1.3.4)**

**Analisi Concettuale**

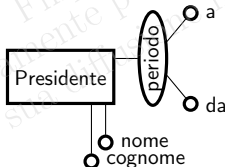
**Il Linguaggio Entity-Relationship  
Relationship**

**Ruoli di Relationship**



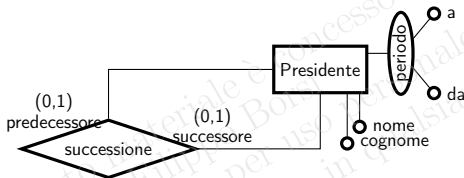
## Relationship che coinvolgono più volte una entità

Supponiamo di voler modellare i presidenti di una repubblica. Di ogni presidente interessa il nome, il cognome, il periodo in cui è stato in carica, ed il predecessore.



Come possiamo modellare il concetto di predecessore?

## Relationship che coinvolgono più volte una entità (2)



Le istanze della relationship **successione** sono coppie  $(p1, p2)$  con  $p1, p2 \in \text{Presidente}$

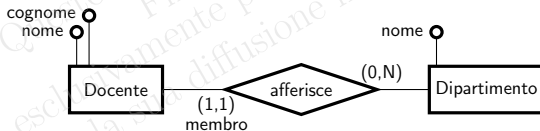
Per evitare ambiguità, diamo un nome ai **ruoli** che le istanze di entità Presidente giocano nei legami (istanze di relationship)

Una istanza di relationship **successione** diventa una **coppia etichettata**:

**(predecessore:p1, successore:p2)**

## Ruoli delle entità nelle relationship

- ▶ Per ogni entità **E** coinvolta in una relationship **r**, è possibile specificare i **ruoli** di **E** in **r**, sugli archi che collegano **E** ad **r**
- ▶ I nomi dei ruoli per una relationship **r** devono essere **distinti**
- ▶ La specifica dei ruoli può essere **sempre** effettuata, ed è **obbligatoria** per una relationship che insiste **più volte** su una entità

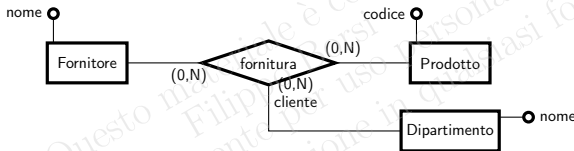


- ▶ Se il ruolo di un'entità in una relationship non è indicato, si assume che abbia nome uguale a quello dell'entità  
⇒ il ruolo di **Dipartimento** in **afferisce** è: **Dipartimento**

## Relationship di arità maggiore di 2

Una relationship può legare anche **più di due** entità

**Esempio:** relationship ternaria



A livello estensionale, la relationship **fornitura** rappresenta un **insieme di terne etichettate**

(Fornitore:f, cliente:d, Prodotto:p)

tali che  $f \in \text{Fornitore}$ ,  $d \in \text{Dipartimento}$ ,  $p \in \text{Prodotto}$ .

La semantica dei **vincoli di cardinalità** è analoga al caso di relationship tra due entità

## Semantica delle relationship (versione finale)

Siamo pronti a dare la **semantica completa** di una relationship tra un numero arbitrario (almeno 2) di entità, non necessariamente tutte distinte

A livello estensionale, una relationship  $r$  tra le entità  $E_1, E_2, \dots, E_n$  (non necessariamente tutte distinte), con ruoli, rispettivamente  $u_1, u_2, \dots, u_n$  è costituita da un **insieme di n-ple etichettate** della forma

$$(u_1:x_1, u_2:x_2, \dots, u_n:x_n)$$

tali che:  $x_1 \in E_1, x_2 \in E_2, \dots, x_n \in E_n$



**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

### **Slides A.1.3.5 (S.A.1.3.5)**

**Analisi Concettuale**

**Il Linguaggio Entity-Relationship  
Relationship**

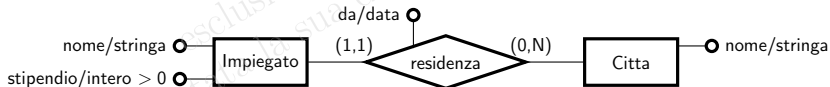
**Attributi di Relationship**

## La nozione di attributo di relationship

- ▶ Proprietà **locale** di una relationship
- ▶ associa ad ogni istanza di relationship (ennupla di istanze di entità) un valore in un certo dominio (tipo)

Il valore di un attributo di una istanza di relationship dipende **solo** dall'istanza stessa, non ha (in genere) **alcun rapporto** con le altre istanze

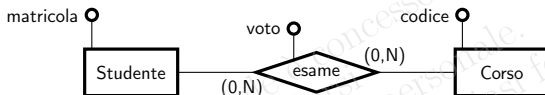
**Esempio:**



Ad ogni istanza  $(i, c) \in \text{residenza}$  (con  $i \in \text{Impiegato}$  e  $c \in \text{Citta}$ ) è associato un valore per l'attributo **da** sul dominio **data**

## La nozione di attributo di relationship (2)

### Esempio

Entità **Studente**Relationship **esame**Entità **Corso**

attributo	dominio	attributo	dominio	attributo	dominio
matricola	intero	voto	[18,30]	codice	intero

**Nota:** un'istanza di relationship **esame** è ancora una **coppia**  $(s, c)$  con  $s \in \text{Studente}$  e  $c \in \text{Corso}$ , alla quale è però **associato** un valore  $v \in [18, 30]$

In particolare, un'istanza di **esame** **non è una terna**  $(p, c, v)$  !

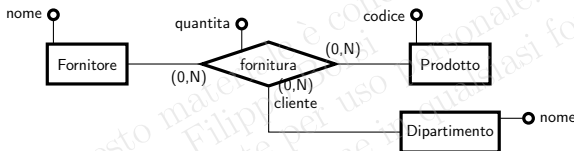
$\Rightarrow$  continuano a **non** poter coesistere due istanze di **esame** che legano la stessa coppia studente/corso (**anche se con voti diversi**)



## La nozione di attributo di relationship (3)

Anche relationship di arità maggiore di 2 possono avere attributi

**Esempio:**



A livello estensionale, la relationship **fornitura** rappresenta un **insieme di terne etichettate**

(**Fornitore:f**, **cliente:d**, **Prodotto:p**)

tali che  $f \in \text{Fornitore}$ ,  $d \in \text{Dipartimento}$ ,  $p \in \text{Prodotto}$ , ad ognuna delle quali è **associato** un valore per l'attributo **quantita**.

**Non possono coesistere** due terne uguali (indipendentemente dal valore per l'attributo)

## Esempio

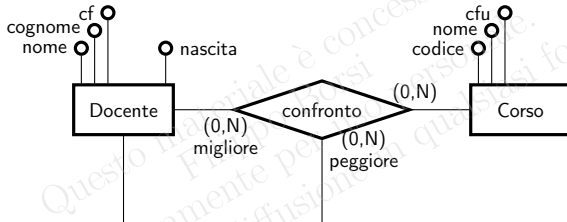
Si vogliono modellare i seguenti requisiti per un sistema informativo universitario.

Sono di interesse per l'applicazione i docenti ed i corsi. Dei docenti si vuole rappresentare nome, cognome, codice fiscale e data di nascita, dei corsi si vuole mantenere codice identificativo, nome e numero di crediti.

Sfruttando i moduli di valutazione dei corsi e dei docenti da parte degli studenti, si vuole poi rappresentare l'informazione circa quale docente è più apprezzato di quale altro come insegnante di un corso.

## Esempio (2)

Una possibile soluzione:



Entità **Docente**

attributo	dominio
nome	stringa
cognome	stringa
cf	stringa
nascita	data

Entità **Corso**

attributo	dominio
codice	intero
nome	stringa
cfu	intero > 0



**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

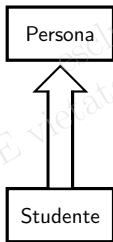
### **Slides A.1.4 (S.A.1.4)**

**Analisi Concettuale**

**Il Linguaggio Entity-Relationship**  
**Relazioni IS-A tra Entità**

## Relazione is-a tra entità

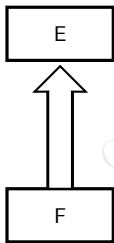
- ▶ Fino ad ora abbiamo implicitamente assunto che entità diverse non hanno istanze in comune
- ▶ In molte situazioni, vogliamo rappresentare il fatto che tra due entità sussista una relazione di **sottoinsieme**
- ▶ ER permette di definire il concetto di **relazione is-a** tra entità



Ogni istanza di **Studente** è anche (is-a)  
un'istanza di **Persona**

- ▶ **Persona** è l'entità base
- ▶ **Studente** è l'entità derivata (o entità figlia, o sotto-entità)

## Semantica della relazione is-a tra entità



A **livello estensionale**, in ogni momento l'insieme delle istanze di **F** deve essere un

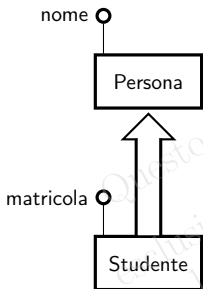
**sottoinsieme** (anche non proprio)

dell'insieme delle istanze di **E**

### Esempio

- ▶ Istanze di **F** = {a,c}
- ▶ Istanze di **E** = {a,b, c,d}

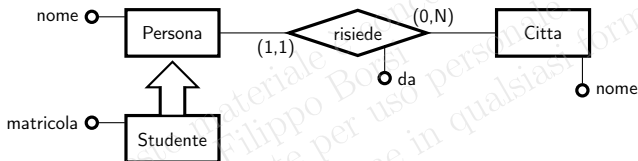
## Ereditarietà su entità: attributi



1. Ogni istanza di **Persona** ha un valore per l'attributo **nome**
  2. ogni istanza di **Studente** è anche (is-a) istanza di **Persona**
- ⇒ ogni istanza di **Studente** ha un valore per l'attributo **nome**

L'entità **Studente** può avere **ulteriori** attributi  
(nell'esempio: **matricola**)

## Ereditarietà su entità: relationship



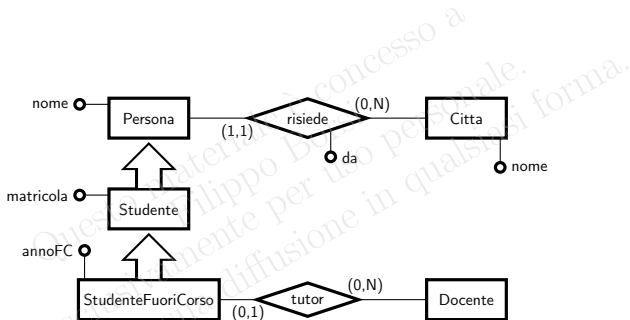
1. Ogni istanza di **Persona** deve essere coinvolta in esattamente una istanza di relationship **risiede**
2. ogni istanza di **Studente** è anche (is-a) istanza di **Persona**

⇒ ogni istanza di **Studente** deve essere coinvolta in esattamente una istanza di relationship **risiede**

L'entità **Studente** può essere coinvolta in **ulteriori** relationship



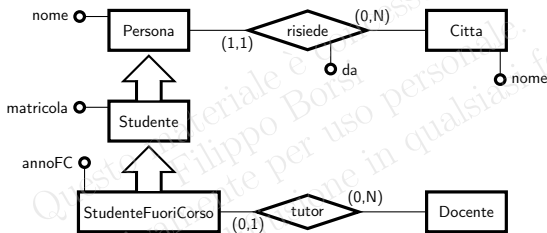
## Ereditarietà su entità: transitività



1. Ogni istanza di **StudenteFuoriCorso** è anche (is-a) istanza di **Studente**
2. Ogni istanza di **Studente** è anche (is-a) istanza di **Persona**

⇒ ogni istanza di **StudenteFuoriCorso** ...

## Ereditarietà su entità: transitività (2)



⇒ ogni istanza di **StudenteFuoriCorso**:

- ▶ ha esattamente un valore per l'attributo **nome**
- ▶ ha esattamente un valore per l'attributo **matricola**
- ▶ deve essere coinvolta in esattamente una istanza di relationship **risiede**

e può avere ulteriori **proprietà** (attributi o relationship)

## Diverse figlie di una entità

- ▶ Una entità può essere base di più relazioni is-a
- ▶ Le entità figlie possono avere **istanze in comune**

### Esempio



- ▶ Tutte le istanze di **Studente** sono anche istanze di **Persona**
- ▶ Tutte le istanze di **Donna** sono anche istanze di **Persona**
- ▶ Possono esistere istanze di **Persona** che **non** sono istanze **né** di **Studente** **né** di **Donna**
- ▶ Possono esistere istanze di **Persona** che sono istanze **sia** di **Studente** **che** di **Donna**



**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

**Slides A.1.5 (S.A.1.5)**

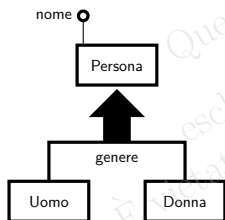
**Analisi Concettuale**

**Il Linguaggio Entity-Relationship**  
**Generalizzazioni tra Entità**

## La generalizzazione tra entità

ER offre un ulteriore **costrutto** rispetto alla relazione is-a, il costrutto della **generalizzazione**

La generalizzazione permette di **classificare** le istanze di una entità in più entità figlie secondo uno **stesso criterio** concettuale

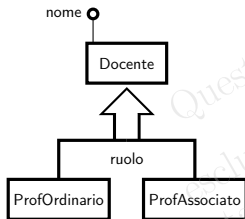


- ▶ Tutte le istanze di **Uomo** sono anche istanze di **Persona**
- ▶ Tutte le istanze di **Donna** sono anche istanze di **Persona**
- ▶ Ogni istanza di **Uomo** **non** è un'istanza di **Donna** (**disgiunzione**)
- ▶ Ogni istanza di **Persona** è istanza di **Uomo** o di **Donna** (**completezza**)

Il **criterio concettuale** della classificazione è il **genere**

## La generalizzazione tra entità (2)

È possibile anche imporre una generalizzazione **non completa**

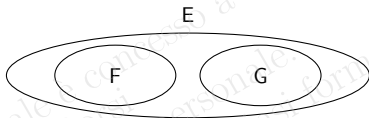
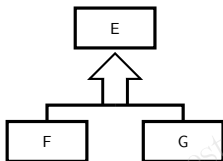


- ▶ Tutte le istanze di **ProfAssociato** sono anche istanze di **Docente**
- ▶ Tutte le istanze di **ProfOrdinario** sono anche istanze di **Docente**
- ▶ Ogni istanza di **ProfAssociato** **non** è un'istanza di **ProfOrdinario** (**disgiunzione**)
- ▶ **Possono** esistere istanze di **Docente** che **non** sono istanze **né** di **ProfAssociato** **né** di **ProfOrdinario** (**nessun vincolo di completezza**)

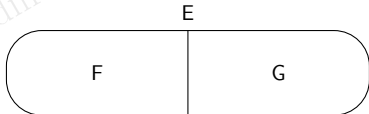
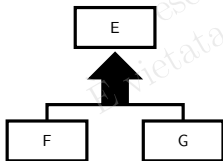
Essendo un **raffinamento** della relazione **is-a**, anche per la generalizzazione vale il principio dell'**ereditarietà** di attributi e relationship

## Relazioni Generalizzazioni: riepilogo

Generalizzazione **non completa**



Generalizzazione **completa**

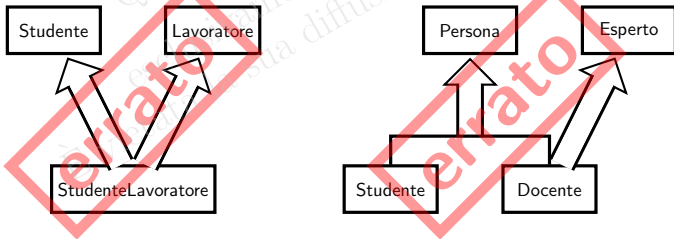


In entrambi i casi sussiste il vincolo di **disgiunzione** tra **F** e **G**

## ER: solo ereditarietà singola

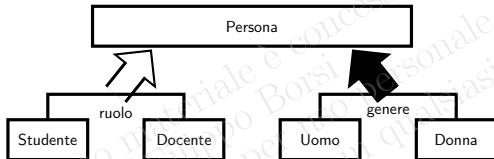
Una entità non può essere coinvolta come figlia di più relazioni is-a e/o generalizzazioni

ER ammette solo **ereditarietà singola**



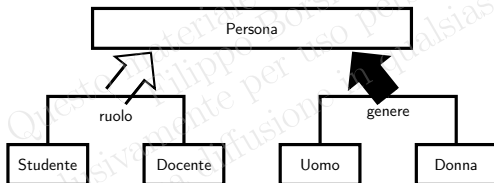


## Generalizzazioni multiple con stessa entità base



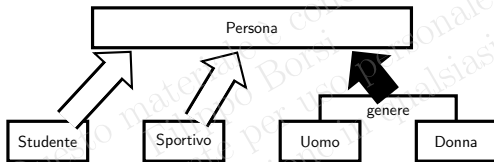
- ▶ Secondo il criterio del **genere**, le persone si **partizionano** in uomini e donne (generalizzazione **completa**)
- ▶ Non esistono persone che siano sia uomini che donne (**disgiunzione**) e non esistono persone che non sono né uomini né donne (**completezza**)
- ▶ Secondo il criterio del **ruolo**, le persone si **classificano** in impiegati, studenti e **altri** (generalizzazione **non completa**)
- ▶ Non esistono persone che siano sia impiegati che studenti (**disgiunzione**)

## Generalizzazioni multiple con stessa entità base (2)



Ogni istanza di **Impiegato** e ogni istanza di **Studente** sarà anche istanza di esattamente una tra **Uomo** e **Donna**

## Generalizzazioni e is-a multiple con stessa entità base



- ▶ Secondo il criterio del **genere**, le persone si **partizionano** in uomini e donne (generalizzazione **completa**)
- ▶ Non esistono persone che siano sia uomini che donne (**disgiunzione**) e non esistono persone che non sono né uomini né donne (**completezza**)
- ▶ Alcune persone sono studenti (ed in quanto persona, ogni studente è o uomo o donna)
- ▶ Alcune persone sono sportivi (ed in quanto persona, ogni sportivo è o uomo o donna)

## Generalizzazioni e is-a multiple con stessa entità base (2)

Possiamo chiaramente avere studenti sportivi uomini, studenti non sportivi donne, etc.

## Attributi omonimi in generalizzazioni e is-a

In generale, attributi con lo stesso nome associati ad **entità diverse** sono **scorrelati** (rappresentano funzioni diverse dalle istanze dell'entità ai valori del loro dominio)

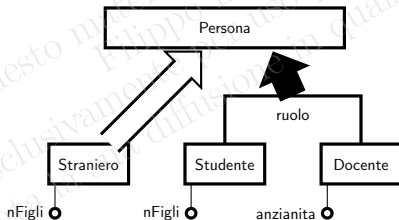
**Esempio:** gli attributi **nome** di **Impiegato** e **Citta** sono scorrelati



Questo perché **Impiegato** e **Citta** non hanno istanze in comune

## Attributi omonimi in generalizzazioni e is-a (2)

Quando due entità con **attributi omonimi** possono avere **istanze in comune**, si assume che tali attributi rappresentino la **stessa** funzione



Uno **studente straniero** ha un **unico valore** per l'attributo **nFigli**

Attributi omonimi di entità che possono avere istanze in comune devono anche avere lo **stesso dominio**



**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

**Slides A.1.6 (S.A.1.6)**

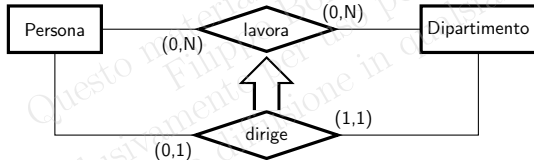
**Analisi Concettuale**

**Il Linguaggio Entity-Relationship  
Relazioni IS-A tra Relationship**

## Relazioni is-a tra relationship

ER permette di modellare relazioni is-a tra **relationship**

Esempio:



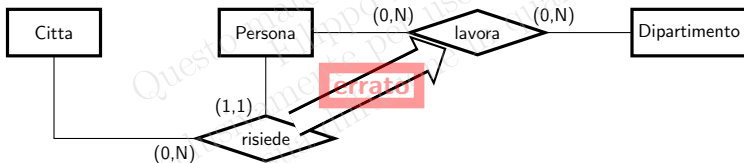
Le persone possono lavorare in dipartimenti. Ogni dipartimento ha un direttore. Ogni persona può dirigere al più un dipartimento e ogni direttore **deve** lavorare nel dipartimento che dirige (oltre, eventualmente, in altri).

**Semantica:** ogni istanza (**Persona:p**, **Dipartimento:d**) di **dirige** è **anche** istanza di **lavora**



## Relazioni is-a tra relationship (2)

Bisogna però stare molto attenti, perché è facile modellare cose **senza senso**:

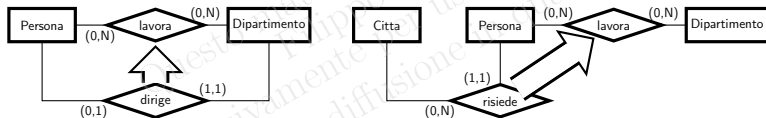


Le istanze di relationship **risiede** sono del tipo (Città:c, Persona:p), mentre le istanze di relationship **lavora** sono del tipo (Persona:p, Dipartimento:d).

La relazione is-a **non può** valere!

## Relazioni is-a tra relationship (3)

Quando ha senso che una relazione is-a tra relationship sia definita?



Sì

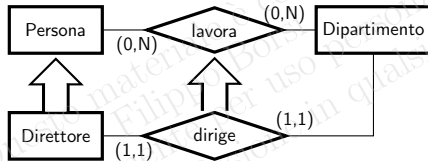
dirige e lavora sono dello stesso tipo

No

risiede e lavora non sono dello stesso tipo

## Relazioni is-a tra relationship (4)

E in questo caso?



Sebbene le relationship *lavora* e *dirige* non siano esattamente dello stesso tipo, ha ancora senso parlare di relazione *is-a* tra loro

Questo perché *Direttore* is-a *Persona*, quindi il tipo di *lavora* ( $Persona \times Dipartimento$ ) è una estensione del tipo di *dirige* ( $Direttore \times Dipartimento$ ):

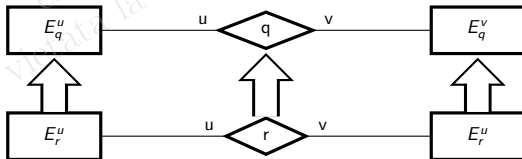
$$(Direttore \times Dipartimento) \subseteq (Persona \times Dipartimento)$$

# Relazioni is-a tra relationship ben definite: condizione 1

Una prima condizione affinché la semantica di  $r$  is-a  $q$  sia definita:

**Condizione 1** ("tipo di  $r \subseteq$  tipo di  $q$ ")

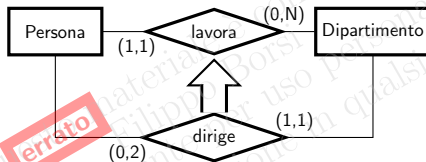
- ▶  $r$  e  $q$  hanno la stessa arità
- ▶  $r$  e  $q$  hanno gli stessi ruoli
- ▶ per ogni ruolo  $u$ , siano  $E_r^u$  e  $E_q^u$  le entità corrispondenti ad  $u$  in  $r$  e  $q$ : si deve avere che  $E_r^u$  is-a  $E_q^u$   
 $\Rightarrow E_r^u = E_q^u$  oppure  $E_r^u$  discende da  $E_q^u$



Non basta...

## Relazioni is-a tra relationship ben definite

La condizione 1 non basta. Esempio A:

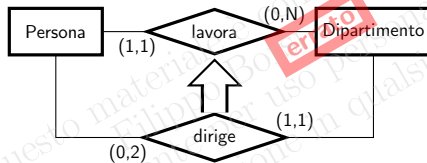


1. Ogni impiegato lavora in **esattamente un** dipartimento
2. Ogni impiegato può essere direttore di **al più due** dipartimenti
3. I direttori **devono** lavorare nei dipartimenti che dirigono

⇒ il valore **massimo** del vincolo di cardinalità (0,2) non può essere mai raggiunto e può essere **ridotto** ad 1 producendo un diagramma **equivalente** ma più accurato

## Relazioni is-a tra relationship ben definite (2)

La condizione 1 non basta. Esempio B:

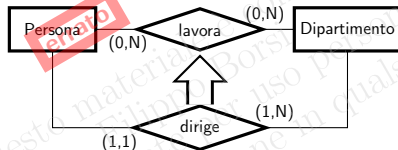


1. In ogni dipartimento può lavorare un qualunque numero  $(0,N)$  di persone
2. Ogni dipartimento ha **esattamente un** direttore
3. I direttori **devono** lavorare nei dipartimenti che dirigono

⇒ il valore **minimo** del vincolo di cardinalità  $(0,N)$  non può essere mai raggiunto e può essere **aumentato** ad 1 producendo un diagramma **equivalente** ma più accurato

## Relazioni is-a tra relationship ben definite (3)

La condizione 1 non basta. Esempio C:

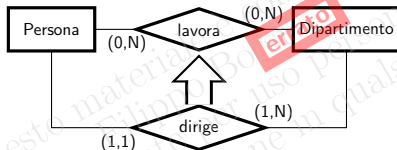


1. Ogni impiegato può lavorare in un **numero arbitrario** di dipartimenti (anche **nessuno**)
2. Ogni impiegato è **sicuramente** (!) direttore di **un** dipartimento
3. I direttori **devono** lavorare nei dipartimenti che dirigono

⇒ il valore **minimo** del vincolo di cardinalità (0,N) non può essere mai raggiunto e può essere **aumentato** ad 1 producendo un diagramma **equivalente** ma più accurato

## Relazioni is-a tra relationship ben definite (4)

La condizione 1 non basta. Esempio D:



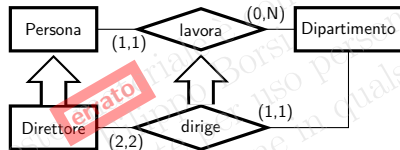
1. Nei dipartimenti possono lavorare un **numero arbitrario** di impiegati (anche **nessuno**)
2. Ogni dipartimento ha **almeno un** direttore
3. I direttori **devono** lavorare nei dipartimenti che dirigono

⇒ il valore **minimo** del vincolo di cardinalità (0,N) non può essere mai raggiunto e può essere **aumentato** ad 1 producendo un diagramma **equivalente** ma più accurato



## Relazioni is-a tra relationship ben definite (5)

La condizione 1 non basta. Esempio E:



1. Ogni impiegato lavora in **esattamente un** dipartimento
2. Ogni direttore dirige **esattamente due** dipartimenti
3. I direttori **devono** lavorare nei dipartimenti che dirigono

⇒ le uniche istanze ammissibili non definiscono alcun direttore  
⇒ ... e quindi nessun dipartimento  
⇒ ... e quindi nessuna persona

L'unica istanza ammissibile per il diagramma è quella **vuota**!

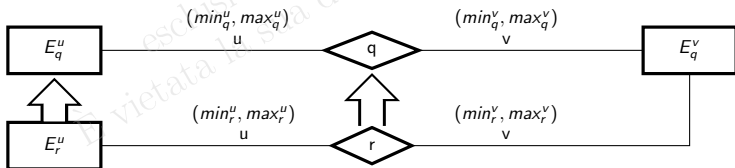
## Relazioni is-a tra relationship ben definite: condizione 2

Condizione 2 per  $r$  is-a  $q$  tra relationship

Per ogni ruolo  $u$  (che per la Condizione 1 è comune a  $r$  e  $q$ ), siano  $E_r^u$  e  $E_q^u$  le entità corrispondenti ad  $u$  in  $r$  e  $q$  (per Condizione 1 si ha:  $E_r^u = E_q^u$  oppure  $E_r^u$  is-a  $E_q^u$ ).

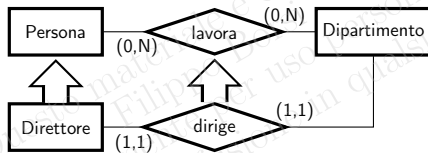
Siano  $(min_r^u, max_r^u)$  e  $(min_q^u, max_q^u)$  i vincoli di cardinalità per il ruolo  $u$  in  $r$  e  $q$ . Si deve avere:

- $max_r^u \leq max_q^u$
- se  $E_r^u = E_q^u$ , allora  $min_r^u \leq min_q^u$



**Nota:** una relazione is-a tra relationship che viola Condizione 2 è formalmente ben definita, ma evidenzia un errore di modellazione

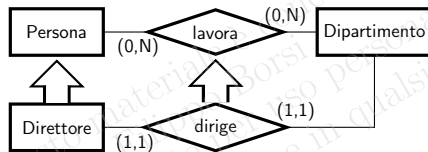
## Relazioni is-a tra relationship ben definite: esempio



In questo diagramma, la relazione is-a tra le relationship **dirige** e **lavora** soddisfa la condizione 1, ma **viola** la condizione 2 (card. minime):

- ▶ il vincolo di cardinalità minima per il ruolo **Dipartimento** in **dirige** è 1, ma
- ▶ il vincolo di cardinalità minima per il ruolo **Dipartimento** in **lavora** è 0

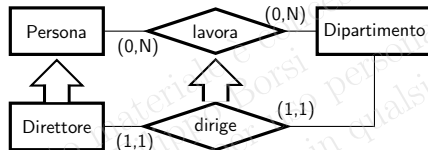
## Relazioni is-a tra relationship ben definite: esempio (2)



Stiamo affermando che:

1. Ogni direttore di dipartimento deve lavorare in quel dipartimento (relazione is-a)
2. Ogni dipartimento ha esattamente un direttore
3. Ogni dipartimento può avere un numero arbitrario (anche zero) di persone che vi lavorano

## Relazioni is-a tra relationship ben definite: esempio (3)



La relazione is-a tra relationship è ben definita, ma evidenzia una **inaccuratezza nella modellazione**

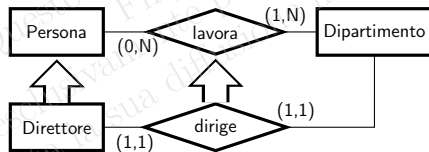
⇒ infatti, ogni dipartimento deve avere **almeno** una persona che vi lavora (il suo direttore)!

È possibile ottenere un diagramma **equivalente** ma **più accurato** aumentando il vincolo di cardinalità minima per il ruolo **Dipartimento** in **lavora** ad 1

⇒ la condizione 2 sarebbe **soddisfatta**

## Relazioni is-a tra relationship ben definite: esempio (4)

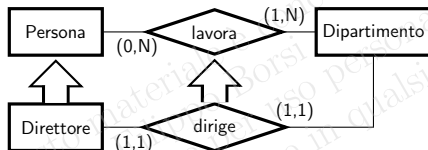
Quindi, dopo il raffinamento del vincolo di cardinalità di **Dipartimento** in **lavora** abbiamo:



E per i ruoli di **Persona/Direttore** in **lavora/dirige**?

## Relazioni is-a tra relationship ben definite: esempio (5)

E per i ruoli di Persona/Direttore in lavora/dirige?



Sebbene il vincolo di cardinalità minima di **Direttore** in **dirige** (1) sia maggiore di quello di **Persona** in **lavora** (0), la condizione 2 è **soddisfatta**, perché **Direttore** e **Persona** **non** sono la stessa entità:

- Ogni direttore dirige esattamente un dipartimento
- Possono esistere persone (**non** direttori!) che non lavorano in alcun dipartimento

**Nota:** la condizione 2 vincola la molteplicità minima di un ruolo  $u$  di  $r$  ( $r$  is-a  $q$ ) ad essere al più quella di  $q$  **solo** se  $r$  e  $q$  coinvolgono la **stessa** entità nel ruolo  $u$



SAPIENZA  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## Basi di Dati, Modulo 2

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

Slides A.1.7 (S.A.1.7)

Analisi Concettuale

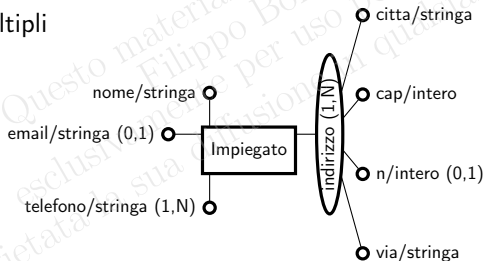
**Il Linguaggio Entity-Relationship**  
**Vincoli di Cardinalità su Attributi**



## Vincoli di cardinalità sugli attributi

ER permette di definire **vincoli di cardinalità sugli attributi** di entità e relationship. Possiamo quindi rappresentare:

- ▶ opzionalità
- ▶ valori multipli



**Default:** se il vincolo di cardinalità di un attributo **non** è indicato, è da intendersi **(1,1)** (valore singolo e obbligatorio)



SAPIENZA  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## Basi di Dati, Modulo 2

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

Slides A.1.8 (S.A.1.8)

Analisi Concettuale

Il Linguaggio Entity-Relationship

Vincoli di Identificazione di Entità

## Vincoli di identificazione di entità

ER permette di esprimere **vincoli di integrità** sul diagramma, che **restringono** l'insieme dei livelli estensionali ammessi

Abbiamo già visto alcuni vincoli di integrità: i vincoli di **cardinalità** per attributi e relationship.

Presentiamo ora i **vincoli di identificazione** di entità:

**Definizione.** Un **identificatore** per una entità **E** è un insieme **I** di attributi e/o ruoli di relationship (tutti a cardinalità (1,1)) in cui **E** è coinvolta tale che:

non esistono due istanze di **E** che coincidono in tutti i valori per **I**

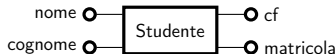
Un **vincolo di identificazione** sull'entità **E** definisce un identificatore per **E**

## Vincoli di identificazione di entità interni

Un **vincolo di identificazione interno** per un'entità è definito su un insieme di soli **attributi** dell'entità

**Esempio:**

1. Si vogliono rappresentare studenti, con nome, cognome, matricola e codice fiscale
2. Non esistono due studenti con lo stesso **codice fiscale**
3. Non esistono due studenti con la stessa **matricola**



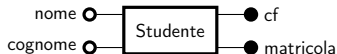
I requisiti 2. e 3. richiedono di definire gli identificatori interni  $\{cf\}$  e  $\{matricola\}$  per l'entità **Studente**

## Vincoli di identificazione di entità interni (2)

Un vincolo di identificazione interno su un identificatore consistente in un **singolo** attributo si definisce annerendo il cerchio relativo a quell'attributo

Esempio:

1. Si vogliono rappresentare studenti, con nome, cognome, matricola e codice fiscale
2. Non esistono due studenti con lo stesso **codice fiscale**
3. Non esistono due studenti con la stessa **matricola**

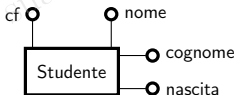


## Vincoli di identificazione di entità interni (3)

Un vincolo di identificazione interno può coinvolgere più attributi

Esempio:

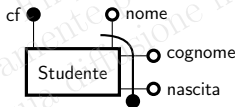
- ▶ Si vogliono rappresentare studenti, con nome, cognome, matricola, codice fiscale e data di nascita
- ▶ non esistono due studenti con lo stesso **codice fiscale**
- ▶ non esistono due studenti che coincidono nel **nome**, **cognome** e **data di nascita** (è solo un esempio, non è un vincolo valido in generale!)



I requisiti 2. e 3. richiedono di definire gli identificatori interni  $\{cf\}$  e  $\{nome, cognome, nascita\}$  per l'entità **Studente**

## Vincoli di identificazione di entità interni (4)

Un vincolo di identificazione su un identificatore (interno) consistente in attributi **multipli** si definisce come in figura:



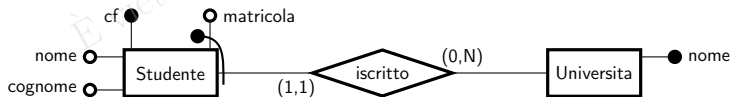
**Nota:** un attributo può essere coinvolto in identificatori multipli

## Vincoli di identificazione di entità esterni

Un **vincolo di identificazione esterno** coinvolge **attributi** e **ruoli** di relationship, **tutti** a molteplicità (1,1)

**Esempio:**

- ▶ Si vogliono rappresentare studenti (con matricola e codice fiscale) e università (con nome)
- ▶ Non esistono due studenti con lo stesso **codice fiscale**
- ▶ Non esistono due studenti della stessa università con la stessa **matricola**
- ▶ Non esistono due università con lo stesso **nome**



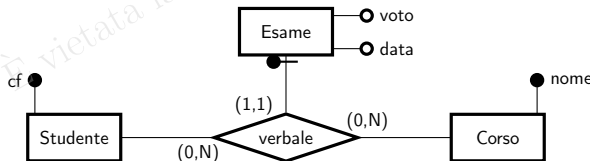


## Vincoli di identificazione di entità esterni (2)

Un vincolo di identificazione esterno può coinvolgere anche un solo ruolo di relationship

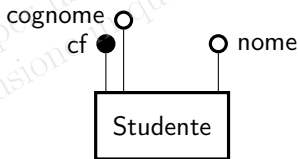
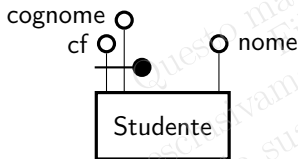
Esempio:

- ▶ Si vogliono rappresentare studenti, corsi ed esami
- ▶ Non esistono due studenti con lo stesso **codice fiscale**
- ▶ Non esistono due corsi lo stesso **nome**
- ▶ Non esistono due esami per la stessa coppia **studente/corso**



## Vincoli di identificazione di entità esterni (3)

I vincoli di identificazione devono essere C-minimali





**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica  
Laurea in Informatica

## **Basi di Dati, Modulo 2**

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

**Slides A.1.9 (S.A.1.9)**

**Analisi Concettuale**

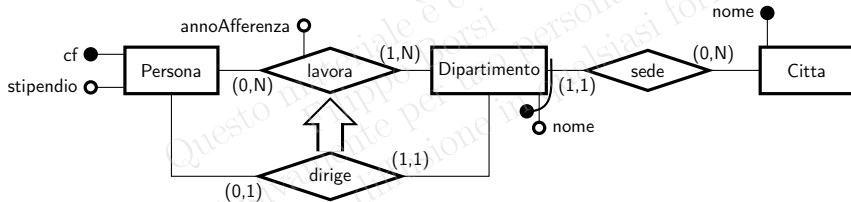
**Il Linguaggio Entity-Relationship**

**Vincoli Esterni (Definizione Informale)**

## Vincoli esterni al diagramma ER

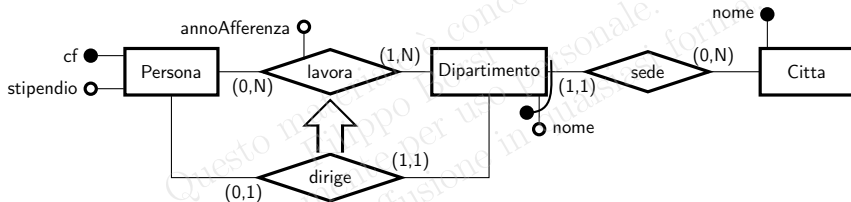
Spesso è necessario imporre **ulteriori vincoli di integrità**, che **non** sono esprimibili direttamente in ER (**business rules**).

**Esempio:**



1. ogni direttore deve lavorare da  $\geq 5$  anni nel dipartimento che dirige
2. nessun impiegato può avere uno stipendio superiore a quello del direttore del suo dipartimento
3. il direttore in ogni dipartimento con sede a Roma deve avere almeno 10 anni di anzianità in quel dipartimento.

## Vincoli esterni al diagramma ER (2)

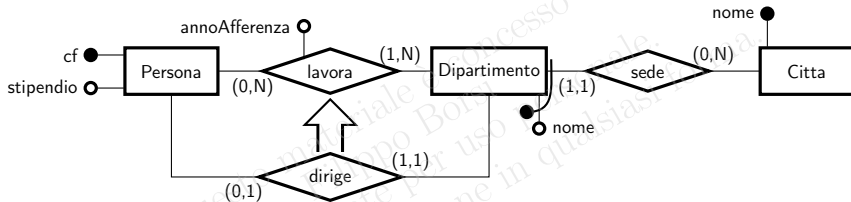


**V.dirige.afferenza:** Per ogni istanza ( $p : \text{Persona}, d : \text{Dipartimento}$ ) della relationship **dirige**, l'istanza ( $p : \text{Persona}, d : \text{Dipartimento}$ ) della relationship **lavora** deve avere un valore  $v$  per l'attributo **annoAfferenza** per cui vale:

$$v \leq \text{annoCorrente} - 5$$

dove **annoCorrente** denota l'istanza del tipo **intero** che rappresenta l'anno corrente.

## Vincoli esterni al diagramma ER (3)



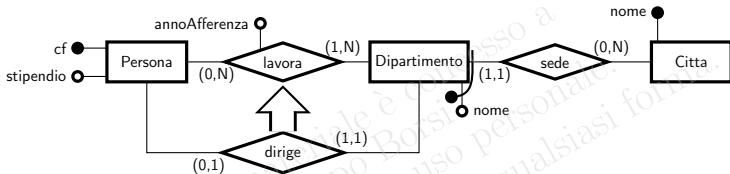
**V. *Persona.stipendio*:** Per ogni istanza (*dir* : *Persona*, *dip* : *Dipartimento*) della relationship *dirige* e per ogni istanza (*p* : *Persona*, *dip* : *Dipartimento*) della relationship *lavora* relativa ad uno stesso dipartimento *dip*, siano:

- ▶ *stip<sub>dir</sub>* il valore dell'attributo *stipendio* di *dir* e
- ▶ *stip<sub>p</sub>* il valore dell'attributo *stipendio* di *p*.

Deve essere:

$$stip_{dir} \geq stip_p.$$

## Vincoli esterni al diagramma ER (4)



**V.dirige.Roma:** Per ogni coppia di istanze ( $dir : \text{Persona}$ ,  $dip : \text{Dipartimento}$ ) della relationship **dirige** e ( $dip : \text{Dipartimento}$ ,  $c : \text{Citta}$ ) della relationship **sede** relative ad uno stesso dipartimento  $dip$ , se l'istanza  $c : \text{Citta}$  ha come valore dell'attributo **nome** la stringa "**Roma**", allora il valore  $a$  dell'attributo **afferenza** dell'istanza ( $dir : \text{Persona}$ ,  $dip : \text{Dipartimento}$ ) della relationship **lavora** deve essere tale che:

$$a \leq \text{annoCorrente} - 10$$

dove **annoCorrente** denota l'istanza del tipo **intero** che rappresenta l'anno corrente.

## Vincoli esterni: sommario

I vincoli esterni ad un diagramma ER impongono ulteriori restrizioni ai livelli estensionali ammessi.

Negli esempi precedenti, per ogni vincolo esterno, abbiamo definito:

- ▶ un identificatore univoco (ad es., **V.dirige.afferenza**)
- ▶ una **asserzione** espressa in **linguaggio naturale**

I vincoli esterni vanno definiti nel **dizionario dei dati**

L'uso del linguaggio naturale per esprimere i vincoli esterni è **pericoloso**, in quanto:

- ▶ potenzialmente ambiguo
- ▶ potenzialmente omissivo o contraddittorio
- ▶ in generale poco leggibile per vincoli complessi

In seguito impareremo a definire i vincoli esterni ad un diagramma ER mediante l'uso della **logica matematica**.