

---

## Parte V

# Progettazione concettuale

---

## Fasi della progettazione di basi di dati

### PROGETTAZIONE CONCETTUALE

- Input: specifiche utente
- Output: schema concettuale (astrazione della realtà)

### PROGETTAZIONE LOGICA

- Input: schema concettuale
- Output: schema logico (nel modello relazionale)

### PROGETTAZIONE FISICA

- Input: schema logico, istanza, carico applicativo
- Output: tuning dell'organizzazione fisica della base di dati

---

## Progettazione concettuale

*Rappresentare i dati della realtà d'interesse in termini di un modello formale, ad alto livello, indipendente dal DBMS*

- Punto di partenza:
  - Descrizione della realtà di interesse
  - Specifica delle elaborazioni che si intendono effettuare
- Modello dei dati: modello Entità-Relazione (ER)
- Produce lo *schema concettuale*: astrazione e rappresentazione degli aspetti rilevanti della realtà ai fini della applicazione
- Deve garantire:
  - Correttezza e completezza della rappresentazione

---

## Progettazione logica

*Rappresentare i dati della realtà d'interesse in termini dei costrutti logici di una classe di DBMS*

- Punto di partenza:
  - Schema concettuale
  - DBMS prescelto
  - Inquadramento del carico applicativo
- Modello dei dati: modello relazionale (anche altri)
- Produce lo *schema logico*, cioè la rappresentazione dei dati nel modello logico prescelto
- Deve garantire:
  - Correttezza e completezza della rappresentazione
  - Efficienza della rappresentazione (rispetto al carico)

## Progettazione fisica

*Rappresentare i dati della realtà di interesse attraverso le strutture dati di uno specifico DBMS*

- Punto di partenza:
  - Schema logico
  - Previsioni del carico applicativo
- Modello dei dati: modello relazionale (anche altri)
- Produce lo *schema fisico*, cioè un arricchimento dello schema logico con la specifica dell'organizzazione fisica dei dati
- Garantisce:
  - Efficienza dell'elaborazione rispetto al carico applicativo

## Il modello Entità-Relazione

- Rappresentazione grafica, tramite diagrammi
- Permette di catturare due aspetti fondamentali:
  - Struttura dei dati: classi di oggetti e associazioni logiche che tra di essi intercorrono
  - Vincoli: regole a cui le classi e le associazioni devono sottostare per rappresentare la realtà in modo corretto
- Strutture di rappresentazione fondamentali:
  - Entità: descrizione delle *classi di oggetti* e delle loro proprietà elementari
  - Relazioni: descrizione delle *associazioni logiche* che intercorrono tra le classi di oggetti

## Entità

*Una entità è una classe di oggetti (fatti, persone, cose) che sono di interesse per l'applicazione, che hanno esistenza autonoma, e che hanno proprietà comuni*

### Esempi

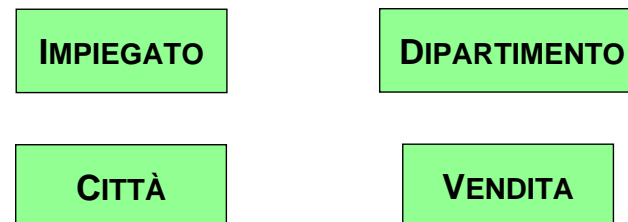
Con riferimento alla descrizione di una realtà aziendale

- IMPIEGATO
- DIPARTIMENTO
- CITTÀ
- VENDITA

## Rappresentazione delle entità

- Ogni *entità* ha nome che la identifica in modo univoco nello schema
- Ciascuna *entità* è rappresentata da un rettangolo con il suo nome nel diagramma che descrive lo schema concettuale

### Esempi



## Semantica delle entità

- A livello *estensionale* un'entità è costituita da un insieme di oggetti, che sono chiamati le sue *istanze*
- Una entità è una classe di oggetti (concetto astratto)
- Una istanza di una entità (in una istanza dello schema) è uno specifico oggetto dell'insieme rappresentato dall'entità
- Se in uno schema **S** è definita una entità **E**, in ogni istanza **I** dello schema **S**, alla entità **E** è associato un insieme di oggetti (denotato  $istanze(I,E)$ )

$\{e1, e2, e3, \dots\}$

## Attributi delle entità

- Un attributo di entità è una proprietà locale di un'entità, di interesse ai fini dell'applicazione
- Un attributo associa ad ogni istanza di entità un valore appartenente ad un insieme detto dominio dell'attributo (tipicamente, interi, caratteri, stringhe, ecc.)
- Si definiscono attributi quando si vogliono rappresentare proprietà *locali* (cioè intrinseche) delle istanze dell'entità
- Proprietà *locale* significa che in ogni istanza dello schema il valore di tale proprietà dipende *solamente* dall'oggetto stesso, e non ha alcun rapporto con altri elementi della istanza dello schema

## Rappresentazione degli attributi

- Ogni attributo di entità ha nome che lo identifica *in modo univoco nell'ambito della entità*, ed è rappresentato da un cerchio collegato alla entità a cui appartiene

### Esempi



## Semantica degli attributi

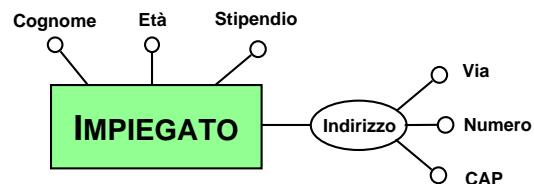
- Ad ogni attributo **A** è associato un dominio **D** costituito da tutti i possibili valori che l'attributo può assumere
- Considerando un'istanza dello schema, un attributo **A** di un'entità **E** è una funzione *totale* tra l'insieme delle istanze dell'entità **E** e il dominio **D** associato all'attributo

$$A : istanze(E) \rightarrow D$$

- La funzione è totale e cioè deve associare a ciascuna sempre, ed in modo univoco un valore di **D** a ciascuna istanza  $x \in istanze(E)$  un valore  $y \in D$
- Per esempio, in una certa istanza dello schema, per ciascun impiegato devono essere sempre definiti univocamente Cognome e Stipendio

## Attributi composti

- Un attributo può anche essere *composto*, cioè definito su un dominio complesso
- In questi casi si usa una rappresentazione gerarchica



- **Indirizzo** è un attributo composto, e il suo dominio è un tipo record con campi **Via**, **Numero** e **CAP**

## Relazioni (associazioni)

- Una *relazione* (o associazione) si definisce su due o più entità, e rappresenta un *legame* fra tali entità
- Il numero di entità coinvolte in una relazione si dice *grado*
- Non va confusa con la relazione nell'accezione del modello relazionale, per questo si usa il termine *associazione*

### Esempi

- **Residenza:** tra **PERSONE** e **CITTÀ** (grado 2)
- **Afferenza:** tra **IMPIEGATO** e **DIPARTIMENTO** (grado 2)
- **Fornitura:** tra **DIPARTIMENTO**, **CLIENTE** e **PRODOTTO** (grado 3)

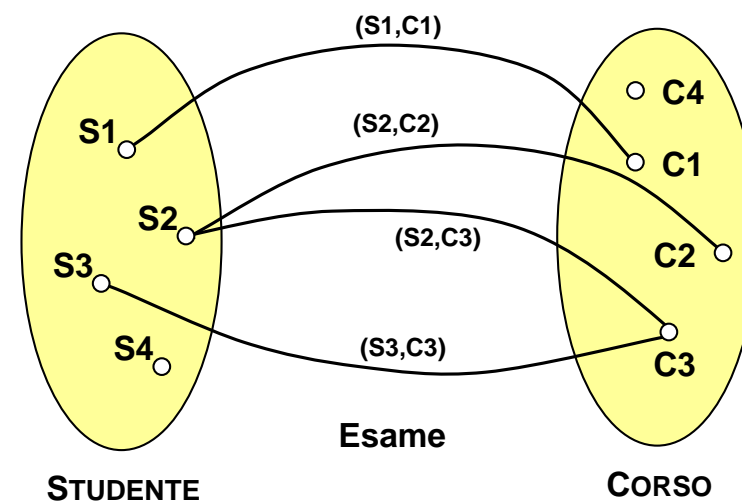
## Rappresentazione delle relazioni

- Ogni *relazione* ha un nome che la identifica in modo univoco nello schema
- Una relazione è rappresentata nel diagramma che descrive lo schema da un rombo che collega le entità sulle quali è definita la relazione

### Esempio



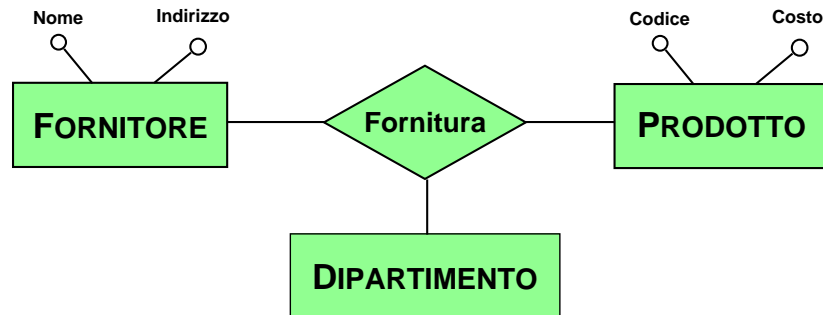
## Relazione binaria a livello estensionale



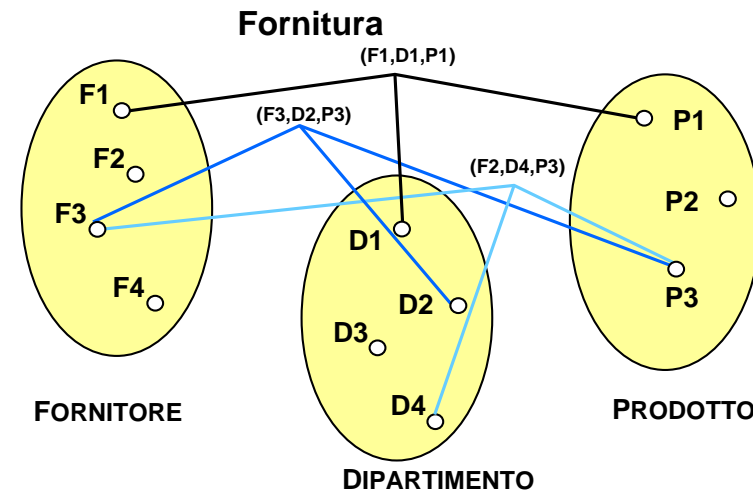
## Relazioni n-arie

- Sono relazioni che coinvolgono più di due entità

### Esempio



## Relazione n-aria a livello estensionale



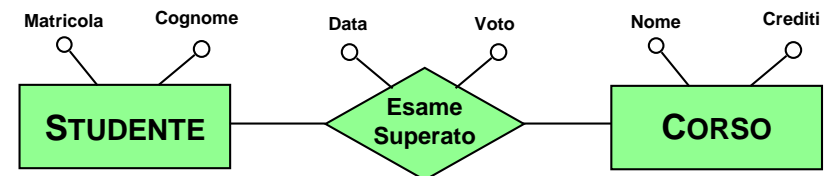
## Attributi delle relazioni

- Un attributo di relazione è una proprietà *locale* di una relazione, di interesse ai fini dell'applicazione
- Un attributo della relazione **R** tra le entità  $E_1, E_2, \dots, E_n$  modella una proprietà *non* di  $E_1$ , *non* di  $E_2, \dots$ , *non* di  $E_n$ , ma del *legame* tra  $E_1, E_2, \dots, E_n$  rappresentato da **R**
- L'attributo è quindi associato all'istanza delle relazione
- Un attributo associa ad ogni istanza di relazione un valore appartenente ad un insieme detto dominio dell'attributo

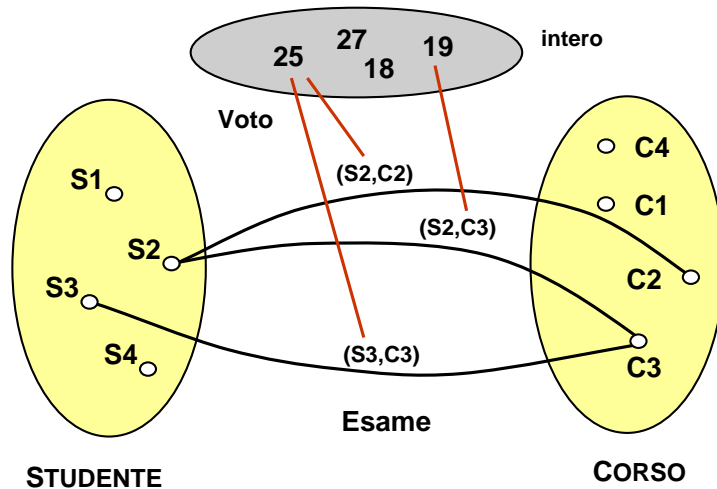
## Rappresentazione degli attributi di relazione

- Ogni attributo di relazione ha nome che lo identifica *in modo univoco nell'ambito della relazione*, ed è rappresentato da un cerchio collegato alla relazione a cui appartiene

### Esempi



## Attributi di relazione: livello estensionale

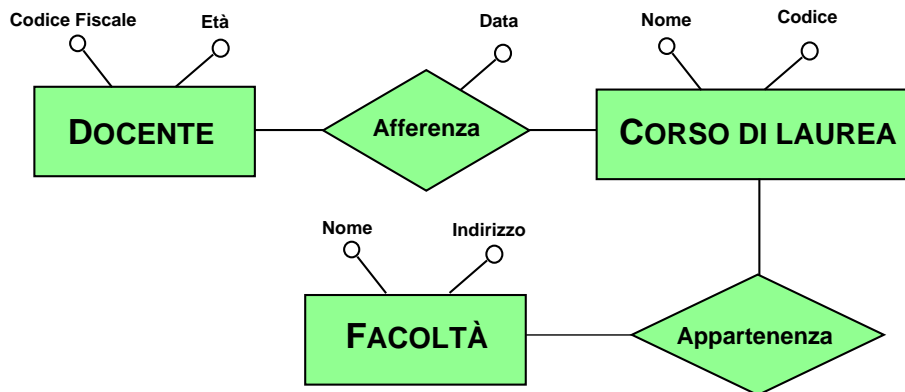


## Esempio 1: specifiche

Descrivere con un diagramma ER uno schema concettuale che rappresenti organizzazione universitaria:

- I docenti hanno un codice fiscale ed una età
- I docenti operano nei corsi di laurea (si dice che afferiscono ai corsi di laurea)
- Interessa la data di afferenza dei docenti ai corsi di laurea
- I corsi di laurea hanno un codice ed un nome, ed appartengono alle facoltà
- Ogni facoltà ha un nome e un indirizzo

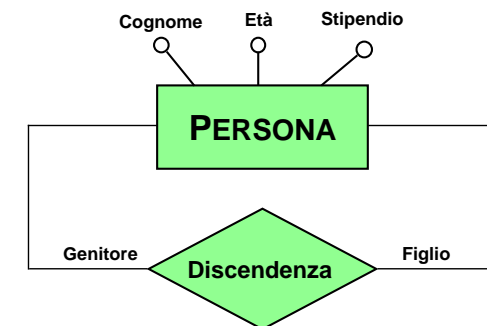
## Esempio 1: diagramma ER



## Ruoli

- Una relazione può coinvolgere più volte la stessa entità
- In questi casi è necessario specificare i *ruoli*, cioè specificare *a che titolo* più istanze della stessa entità partecipano ad un'istanza della relazione

## Esempio

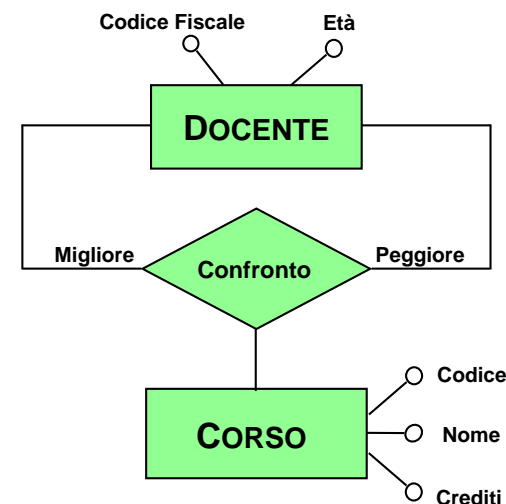


## Esempio 2: specifiche

Descrivere con un diagramma ER uno schema concettuale che rappresenti organizzazione universitaria:

- I docenti hanno un codice fiscale ed una età
- I corsi hanno un codice identificativo, un nome, ed il numero di crediti
- In base alla valutazione dei corsi e dei docenti da parte degli studenti, si vuole rappresentare nella base di dati l'informazione se un docente è migliore di un altro nell'insegnare un corso

## Esempio 2: diagramma ER



## Esempio 2: istanza

*istanze(DOCENTE)*: { Paolo, Mario, Grazia }

*istanze(CORSO)*: { BasiDiDati, Logica, Algebra }

*istanze(Confronto)*:

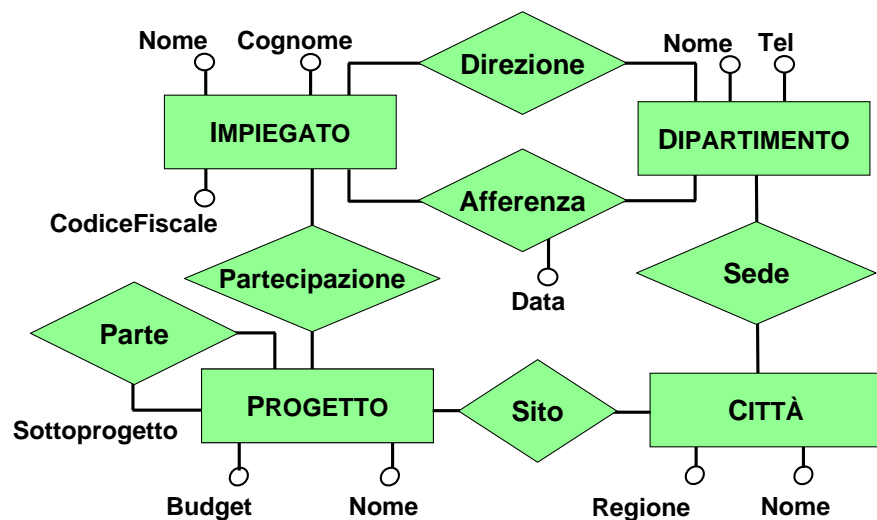
{ (Corso:BasiDiDati, *Migliore*:Mario, *Peggior*: Grazia),  
(Corso:BasiDiDati, *Migliore*:Mario, *Peggior*: Paolo),  
(Corso:BasiDiDati, *Migliore*:Grazia, *Peggior*: Paolo),  
(Corso: Logica, *Migliore*:Grazia, *Peggior*: Mario),  
(Corso: Logica, *Migliore*:Grazia, *Peggior*: Paolo),  
(Corso: Logica, *Migliore*:Mario, *Peggior*: Paolo),  
(Corso: Algebra, *Migliore*:Grazia, *Peggior*: Paolo) }

## Esempio 3: specifiche

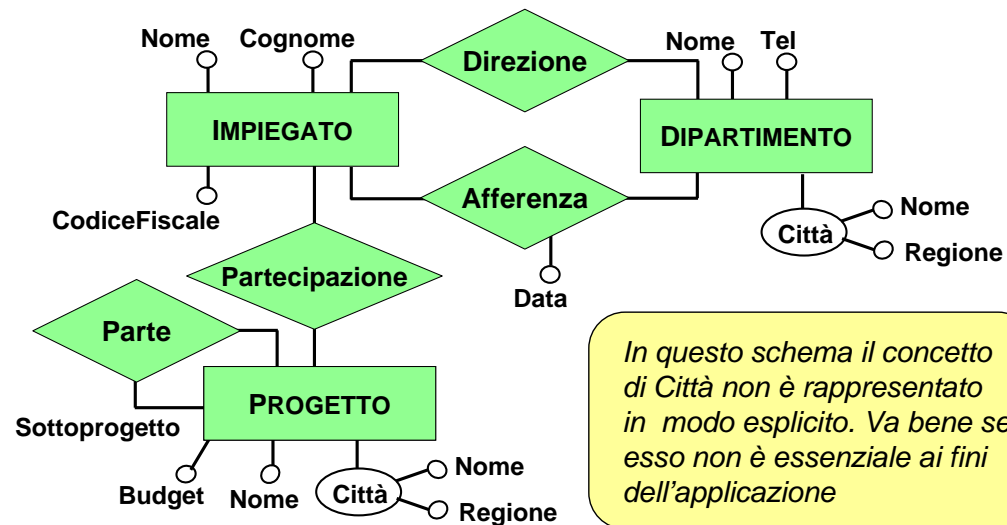
Descrivere lo schema concettuale della seguente realtà:

- Degli impiegati interessa il codice fiscale, il nome, il cognome, i dipartimenti ai quali afferiscono (con la data di afferenza), ed i progetti ai quali partecipano
- Dei progetti interessa il nome, il budget, e la città in cui hanno luogo le corrispondenti attività
- Alcuni progetti sono parti di altri progetti, e sono detti loro sottoprogetti
- Dei dipartimenti interessa il nome, il numero di telefono, gli impiegati che li dirigono, e la città dove è localizzata la sede
- Delle città interessa il nome e la regione

## Esempio 3: diagramma ER



## Esempio 3: diagramma ER (bis)



## Attributi o entità?

Un concetto verrà modellato come

- Entità
  - Se le sue istanze sono concettualmente significative indipendentemente da altre istanze
  - Se ha, o potrà avere proprietà indipendenti dagli altri concetti
  - Se il concetto è importante nell'applicazione
- Attributo (di entità o relazione)
  - Se le sue istanze non sono concettualmente significative
  - Se non ha senso considerare una sua istanza indipendentemente da altre istanze
  - Se rappresenta una proprietà locale di un altro concetto

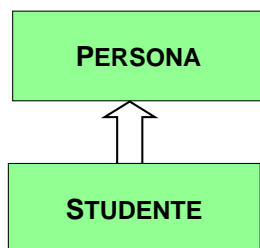
## Relazione ISA

- Entità diverse possono avere istanze in comune
- In molti contesti, può accadere che tra classi rappresentate da entità distinte nello schema concettuale sussista la relazione ISA (o relazione di sottoinsieme), e cioè che ogni istanza di una sia anche istanza dell'altra
- La relazione ISA nel modello ER si può definire tra due entità
- Le entità si dicono entità padre ed entità figlia (sottoentità)
- Le istanze dell'entità figlia sono un sottoinsieme di quelle della entità padre

## Rappresentazione della relazione ISA

La relazione ISA si rappresenta nel diagramma dello schema concettuale mediante una freccia dalla entità figlia alla entità padre

### Esempio



**N.B.** Una entità può avere al massimo una entità padre. Il modello ER non permette di rappresentare l'ereditarietà multipla

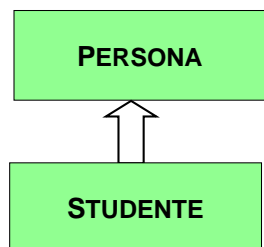
## Semantica della relazione ISA

- A livello estensionale la relazione ISA impone che in ogni istanza di uno schema in cui  $E_1$  ISA  $E_2$  sia:

$$\text{istanze}(E_1) \subseteq \text{istanze}(E_2)$$

- Due entità che non sono in relazione ISA non possono avere istanze comuni
- Questo impone che un'entità può avere al massimo una entità padre
- Dalla definizione segue che la relazione ISA è:
  - *Riflessiva*: ogni entità è in relazione ISA con se stessa
  - *Transitiva*: se  $E_1$  ISA  $E_2$  e  $E_2$  ISA  $E_3$  allora  $E_1$  ISA  $E_3$

## Esempi a livello estensionale



### Istanza I1:

$\text{istanze}(\text{PERSONA}) = \{ a, b, c, d \}$   
 $\text{istanze}(\text{STUDENTE}) = \{ b, d \}$

### Istanza I2:

$\text{istanze}(\text{PERSONA}) = \{ a, b, c, d \}$   
 $\text{istanze}(\text{STUDENTE}) = \{ a, b, c, d \}$

### Istanza scorretta

$\text{istanze}(\text{PERSONA}) = \{ a \}$   
 $\text{istanze}(\text{STUDENTE}) = \{ a, b \}$

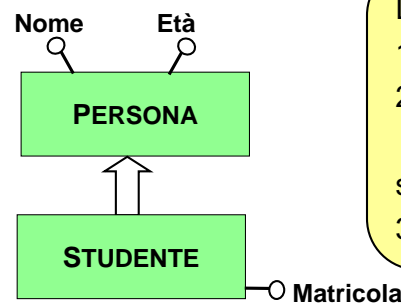
### Istanza I3:

$\text{istanze}(\text{PERSONA}) = \{ a, b, c, d \}$   
 $\text{istanze}(\text{STUDENTE}) = \{ \}$

## Ereditarietà nel modello ER

- Principio di ereditarietà: ogni proprietà dell'entità padre è anche una proprietà della sottoentità
- L'entità figlia può avere ovviamente ulteriori proprietà

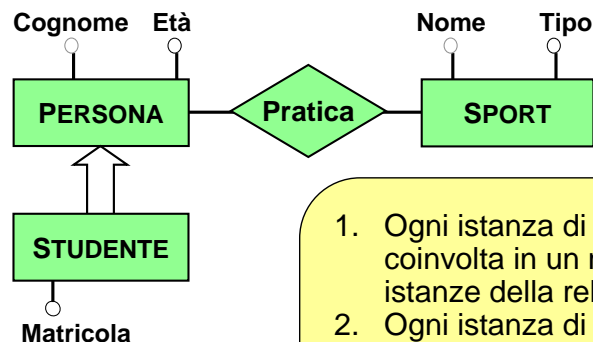
### Esempio



Dal fatto che

1. Ogni istanza di Persona ha una Età
  2. Ogni istanza di Studente è istanza di Persona
- segue logicamente che
3. Ogni istanza di Studente ha una Età

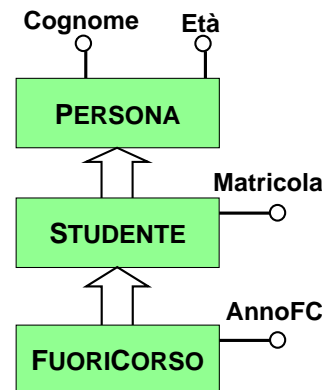
## Ereditarietà delle relazioni



1. Ogni istanza di **PERSONA** può essere coinvolta in un numero qualunque di istanze della relazione **Pratica**
  2. Ogni istanza di **STUDENTE** è una istanza di Persona
- QUINDI*
3. Ogni istanza di **STUDENTE** può essere coinvolta in un numero qualunque di istanze della relazione **Pratica**

## Transitività

- Anche la relazione ISA si eredita.
- La relazione ISA è transitiva (oltre a essere riflessiva)



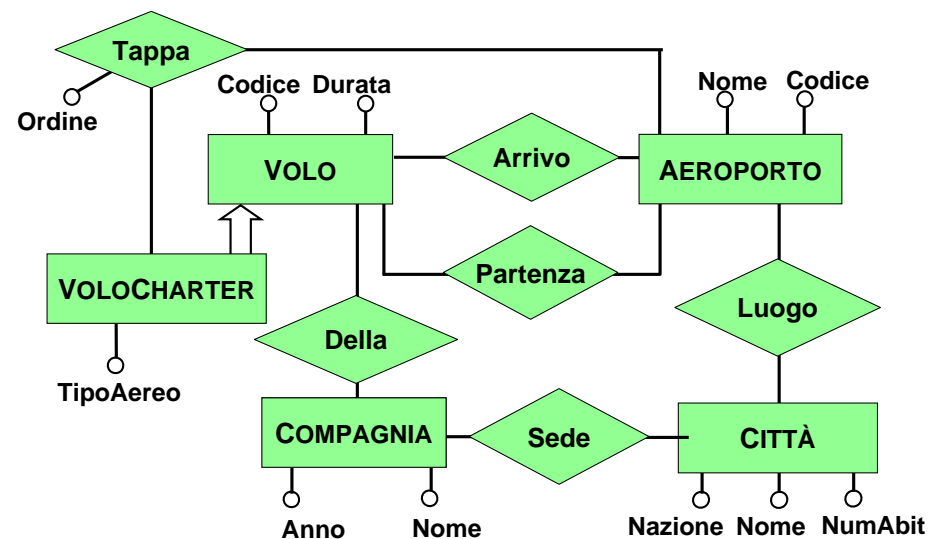
Dal fatto che

1. Ogni istanza di *Studente* è una istanza di *Persona*
  2. Ogni istanza di *FuoriCorso* è una istanza di *Studente*
- segue logicamente che
3. Ogni istanza di *FuoriCorso* è una istanza di *Persona*

## Esempio 4: specifiche

- Schema concettuale corrispondente ad un'applicazione riguardante voli aerei
- Dei voli interessa: codice, durata in minuti, compagnia aerea, aeroporto di partenza e aeroporto di arrivo
- Degli aeroporti interessa: codice, nome, città (con nome e numero di abitanti) e nazione
- Delle compagnie aeree interessa il nome, l'anno di fondazione, e la città in cui ha sede la direzione.
- I voli charter sono particolari voli, che possono prevedere tappe intermedie in aeroporti
- Delle tappe intermedie di un volo charter interessa l'ordine con cui esse si susseguono
- Dei voli charter interessa anche il tipo di aereo utilizzato per il volo.

## Esempio 4: diagramma ER

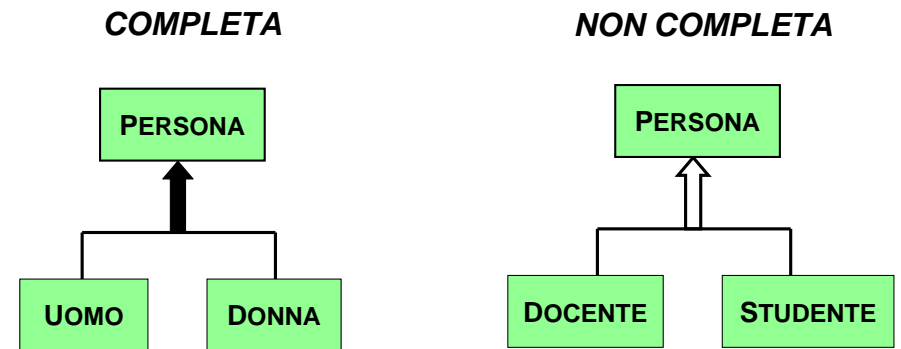


## Generalizzazioni

- La relazione ISA che stabilisce che l'entità padre è più generale della sottoentità
- Se l'entità padre generalizza diverse sottoentità rispetto ad un unico criterio si parla di *generalizzazione*
- Nella generalizzazione, le sottoentità hanno insiemi di istanze disgiunti a coppie
- Una generalizzazione può essere di due tipi:
  - **Completa**: l'unione delle istanze delle sottoentità è uguale all'insieme delle istanze dell'entità padre
  - **Non completa**: alcune istanze dell'entità padre non fanno parte di nessuna delle entità figlie

## Generalizzazioni: rappresentazione

- Si rappresenta con un'unica freccia che raggiunge il padre e archi che partono dalle figlie
- La freccia non è annerita se la generalizzazione non è completa



## Generalizzazioni: livello estensionale

LIVELLO INTENSIONALE	LIVELLO ESTENSIONALE
non completa 	
completa 	

## Vincoli di integrità

- Un *vincolo di integrità* è una regola che si esprime sullo schema (livello intensionale)
- Un vincolo specifica una condizione che deve valere *per ogni istanza* dello schema (livello estensionale)
- Vincoli di integrità nel modello ER:
  - Vincoli di cardinalità sulle relazioni
  - Vincoli di cardinalità sugli attributi
  - Vincoli di identificazione d'entità
  - Altri vincoli (esterni)

## Vincoli di cardinalità sulle relazioni

- Un *vincolo di cardinalità* si associa ad un ruolo **U** di una entità **E** in una relazione **R**
- Il vincolo impone un limite minimo ed un limite massimo di istanze della relazione a cui può partecipare ogni istanza dell'entità **E** nel ruolo **U**
- Costituisce un'importante precisazione del significato di una relazione
- Si esprime mediante una coppia di numeri **(x,y)**:
  - **x** è un intero  $\geq 0$  ed è la *cardinalità minima*
  - **y** è un intero  $\geq x$  ed è la *cardinalità massima*

## Vincoli di cardinalità: esempio

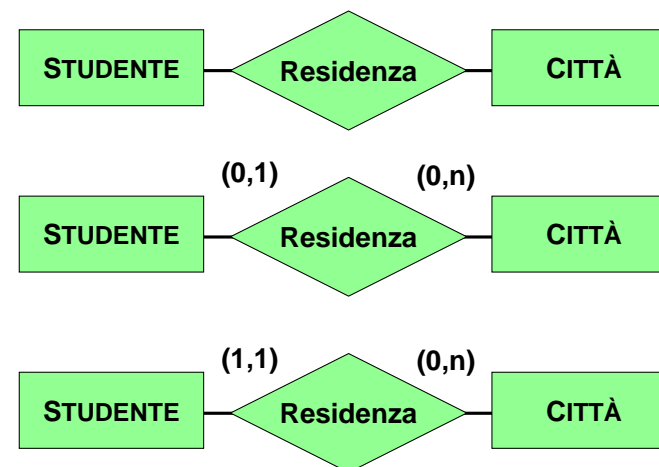


- Ad ogni impiegato è assegnato sempre almeno un incarico, ed al massimo cinque incarichi
- Un dato incarico può non essere assegnato a nessuno, e comunque al massimo è assegnato a 50 impiegati

## Cardinalità maggiormente usate

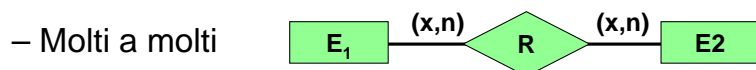
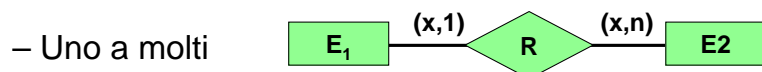
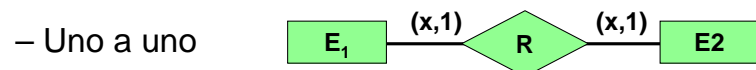
- Le cardinalità di interesse sono tipicamente tre: **0, 1, n**
- Cardinalità minima
  - **0** significa: *partecipazione opzionale*
  - **1** significa: *partecipazione obbligatoria*
- Cardinalità massima
  - **1** significa: *l'entità partecipa al più una volta sola alla relazione*
  - **n** significa: *l'entità partecipa un numero qualsiasi di volte alla relazione*
- La mancanza del vincolo di cardinalità è equivalente alla coppia **(0,n)**

## Cardinalità: esempi

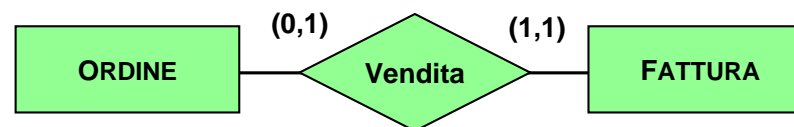


## Classificazione delle relazioni

- Le relazioni binarie possono essere classificate in base ai vincoli di cardinalità
- In base alle cardinalità massime le relazioni binarie possono essere:



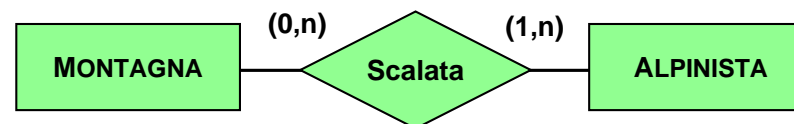
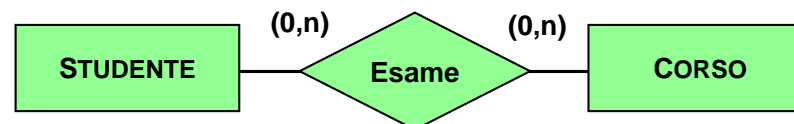
## Relazioni uno ad uno: esempi



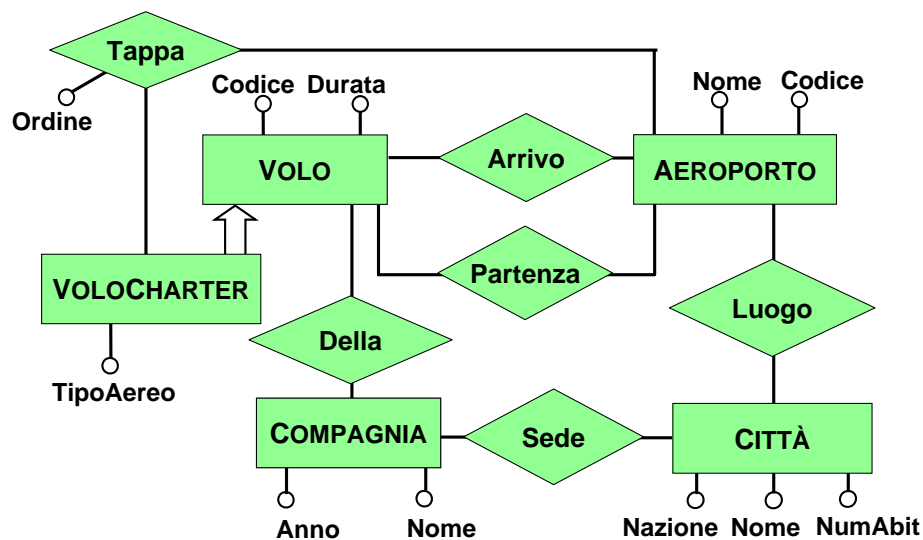
## Relazioni uno a molti: esempi



## Relazioni molti a molti: esempi



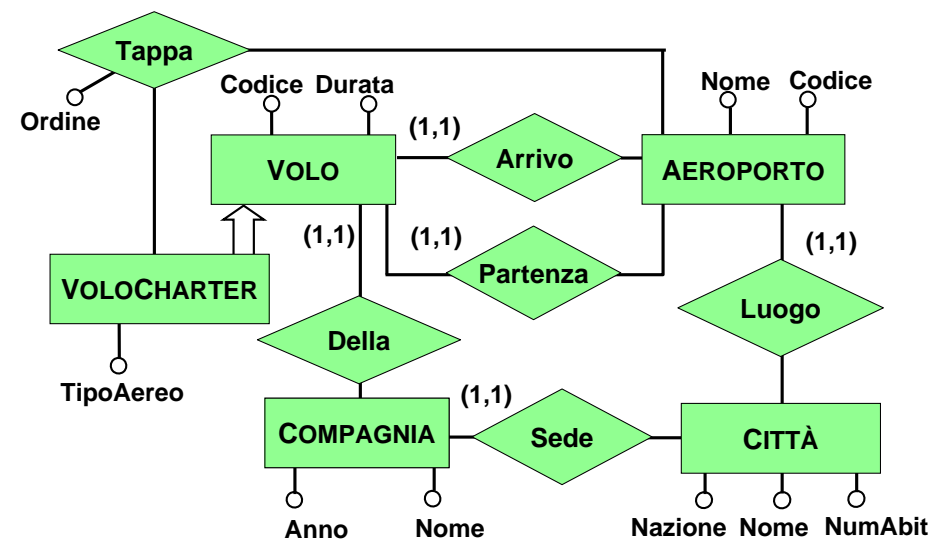
## Esempio 4: senza cardinalità



Basi di dati - prof. Silvio Salza - a.a. 2008-2009

V - 53

## Esempio 4: con cardinalità



Basi di dati - prof. Silvio Salza - a.a. 2008-2009

V - 54

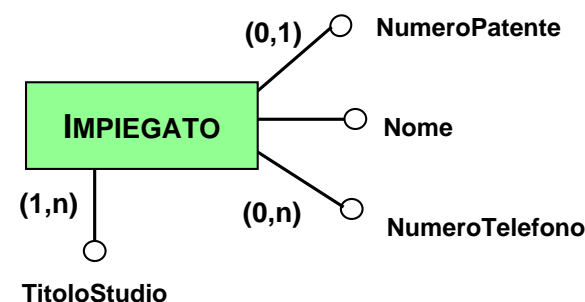
## Vincoli di cardinalità su attributi

- I vincoli di cardinalità anche sugli attributi hanno due scopi:
  - indicare *opzionalità*
  - indicare *attributi multivalore*
- Se la cardinalità massima di un attributo non è **1**, l'attributo si dice *multivalore* (le cardinalità stabiliscono il numero minimo e massimo di valori che l'attributo associa ad ogni istanza dell'entità)
- Se la cardinalità minima di un attributo è **0**, ciò indica che per alcune istanze l'attributo non assume valore
- Se la specifica del vincolo manca, si intende che la cardinalità dell'attributo sia **(1,1)**, con la semantica usuale

Basi di dati - prof. Silvio Salza - a.a. 2008-2009

V - 55

## Cardinalità di attributi: esempio



- Deve avere sempre uno ed un solo nome
- Può non avere la patente, se ce l'ha ne ha una sola
- Deve avere almeno un titolo di studio, am può averne vari
- Può avere nessuno o più numeri di telefono

Basi di dati - prof. Silvio Salza - a.a. 2008-2009

V - 56

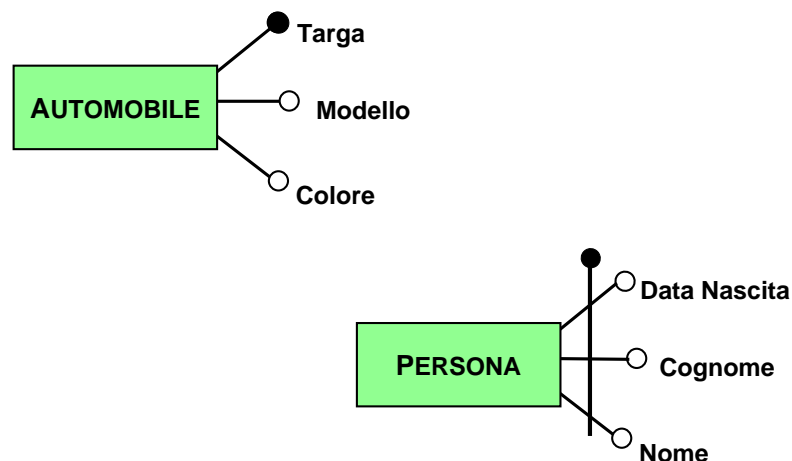
## Vincoli di identificazione di entità

- Un identificatore di una entità è un insieme di proprietà (attributi o relazioni) che permettono di identificare univocamente le istanze di un'entità
- Non possono esistere due istanze di una data entità che assumono lo stesso valore per tutte le proprietà che formano l'identificatore.
- Un *vincolo di identificazione* per una entità **E** definisce un *identificatore* per **E**
- Su ogni entità si può definire un numero qualunque di vincoli di identificazione (tipicamente almeno uno)

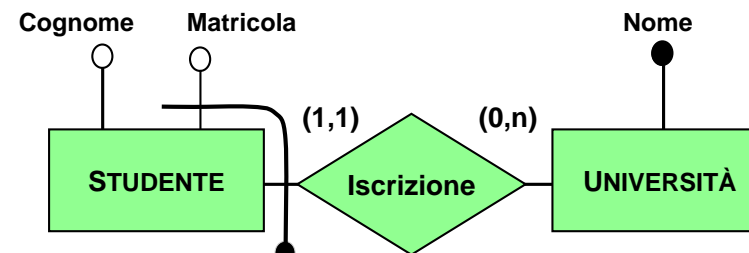
## Identificatori interni ed esterni

- Un identificatore di una entità E può essere
  - Interno : formato solo da attributi di E
  - Esterno : formato da attributi di E e da ruoli di relazioni che coinvolgono E (anche solo da ruoli)
- Tutti gli attributi e tutti i ruoli di relazione che concorrono ad un identificatore di entità devono avere cardinalità (1,1)
- Rappresentazione degli identificatori
  - Interno : se è un solo attributo annerisce il pallino, se sono molti si tagliano con un arco che termina con un pallino nero
  - Esterno : si uniscono attributi ed i ruoli con una linea che termina con pallino nero

## Esempio: identificatori interni

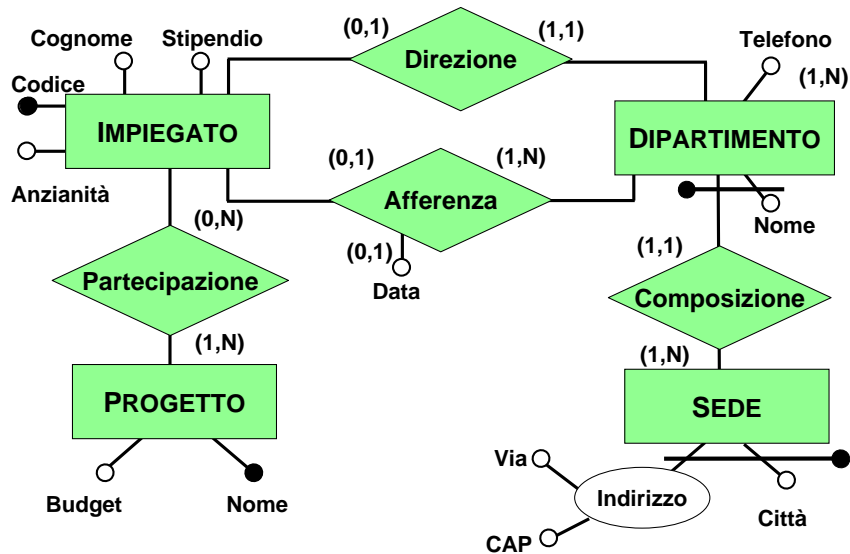


## Esempio: identificatori esterni



- Le matricole degli studenti sono uniche all'interno di ciascuna università
- Un'entità con identificatore esterno viene detta *debole*

## Vincoli di cardinalità e identificazione



Basi di dati - prof. Silvio Salza - a.a. 2008-2009

V - 61

## Vincoli non esprimibili

- Gli schemi ER permettono di cogliere la maggior parte delle interrelazioni tra i dati del dominio d'interesse
- Tuttavia alcune interrelazioni non possono essere colte direttamente da uno schema ER
- Tali interrelazioni vanno in ogni caso tenute presenti tramite asserzioni aggiuntive dette *vincoli esterni al diagramma*, o semplicemente *vincoli esterni*
- I vincoli esterni si rappresentano
  - Attraverso formalismi opportuni (es, *logica matematica*)
  - Attraverso asserzioni in linguaggio naturale (il più possibile precise e non ambigue)

Basi di dati - prof. Silvio Salza - a.a. 2008-2009

V - 62

## Vincoli non esprimibili: esempio

Con riferimento allo schema **IMPIEGATO**, **DIPARTIMENTO**, **PROGETTO**, **SEDE**

Vincoli di integrità esterni
(1) Il direttore di un dipartimento deve afferire a tale dipartimento da almeno 5 anni
(2) Un impiegato non deve avere uno stipendio maggiore del direttore del dipartimento al quale afferisce
(3) Un dipartimento con sede a Roma deve essere diretto da un impiegato con più di dieci anni di anzianità
(4) Un impiegato non può partecipare ad un numero di progetti maggiore di due volte il numero di dipartimenti ai quali afferisce

Basi di dati - prof. Silvio Salza - a.a. 2008-2009

V - 63

## Documentazione degli schemi ER

- Oltre al diagramma ER, lo schema concettuale è descritto dal cosiddetto *dizionario dei dati*
- Il dizionario dei dati è costituito da tabelle che descrivono:
  - *Entità*
  - *Relazioni*
  - *Attributi*
  - *Vincoli esterni*
- Questa documentazione è molto importante
  - Per la successiva fase di progettazione logica
  - Perché lo schema ER e la sua documentazione costituiscono una descrizione formale del contesto applicativo

Basi di dati - prof. Silvio Salza - a.a. 2008-2009

V - 64

## Dizionario dei dati: entità

Entità	Descrizione	Attributi	Identificatori
<b>IMPIEGATO</b>	Dipendente dell'azienda	Codice Cognome Stipendio Anzianità	{ Codice }
<b>PROGETTO</b>	Progetti aziendali	Nome Budget	{ Nome }
<b>DIPARTIMENTO</b>	Struttura aziendale	Nome Telefono	{ Nome, Sede }
<b>SEDE</b>	Sede dell'azienda	Città Indirizzo (Via, CAP)	{ Città, Indirizzo }

## Dizionario dei dati: relazioni

RELAZIONE	DESCRIZIONE	COMPONENTI	ATTRIBUTI
<b>Direzione</b>	Direzione di un dipartimento	Impiegato, Dipartimento	
<b>Afferenza</b>	Afferenza ad un dipartimento	Impiegato, Dipartimento	Data
<b>Partecipazione</b>	Partecipazione ad un progetto	Impiegato, Progetto	
<b>Composizione</b>	Composizione dell'Azienda	Dipartimento, Sede	

## Dizionario dei dati: attributi

Attributo	Entità/Relazione	Dominio	Descrizione
Codice	<b>IMPIEGATO</b>	Intero	Codice identificativo di impiegati
Cognome	<b>IMPIEGATO</b>	Stringa	Cognome di impiegato
Stipendio	<b>IMPIEGATO</b>	Reale	Stipendio di impiegato
Nome	<b>PROGETTO</b>	Stringa	Nome del progetto
...	...	...	...

## Dizionario dei dati: vincoli esterni

Vincoli di integrità esterni
(1) Il direttore di un dipartimento deve afferire a tale dipartimento
(2) Un impiegato non deve avere uno stipendio maggiore del direttore del dipartimento al quale afferisce
(3) Un dipartimento con sede a Roma deve essere diretto da un impiegato con più di dieci anni di anzianità
(4) Un impiegato che non afferisce ad alcun dipartimento non deve partecipare ad alcun progetto