

**E. Giunchiglia**  
**Basi di dati 1**

(trasparenze basate su Atzeni, Ceri, Paraboschi, Torlone: Basi di dati, Capitolo 6)

**Progettazione di basi di dati:  
Metodologie e modelli**

05/10/2004

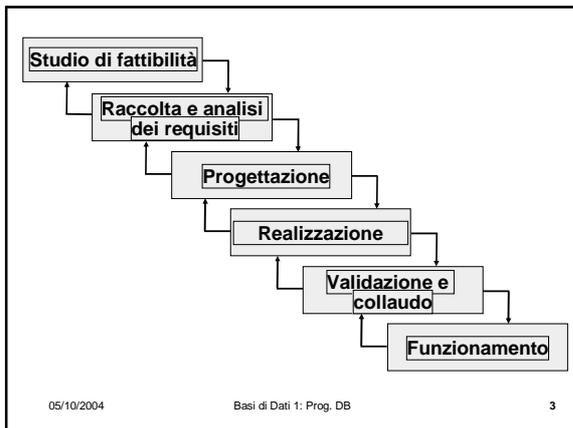
**Progettazione di basi di dati**

- È una delle attività del processo di sviluppo dei sistemi informativi
- va quindi inquadrata in un contesto più generale:
- il ciclo di vita dei sistemi informativi:
  - Insieme e sequenzializzazione delle attività svolte da analisti, progettisti, utenti, nello sviluppo e nell'uso dei sistemi informativi
  - attività iterativa, quindi ciclo

05/10/2004

Basi di Dati 1: Prog. DB

2



05/10/2004

Basi di Dati 1: Prog. DB

3

**Fasi (tecniche) del ciclo di vita**

- Studio di fattibilità: definizione costi e priorità
- Raccolta e analisi dei requisiti: studio delle proprietà del sistema
- Progettazione: di dati e funzioni
- Realizzazione
- Validazione e collaudo: sperimentazione
- Funzionamento: il sistema diventa operativo

05/10/2004

Basi di Dati 1: Prog. DB

4

La progettazione di un sistema informativo riguarda due aspetti:

- ▶ progettazione dei dati
- ▶ progettazione delle applicazioni

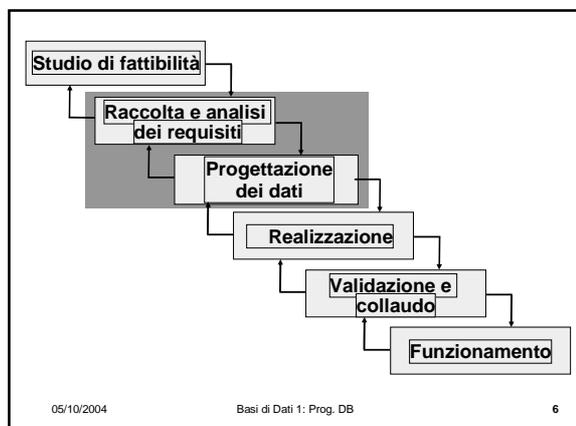
**Ma:**

- ▶ i dati hanno un ruolo centrale
  - i dati sono più stabili

05/10/2004

Basi di Dati 1: Prog. DB

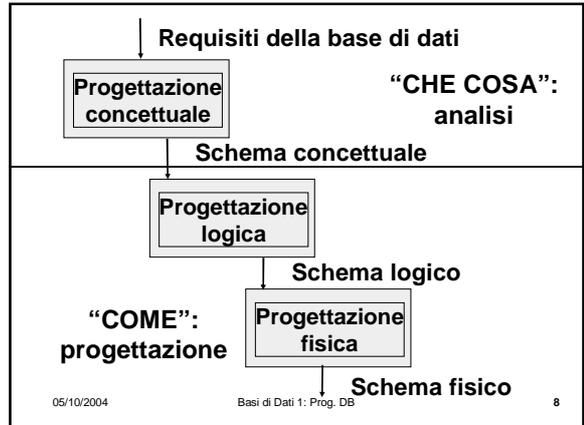
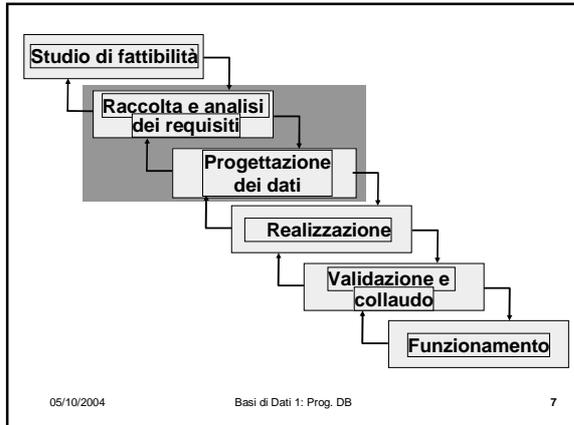
5



05/10/2004

Basi di Dati 1: Prog. DB

6



I prodotti della varie fasi sono schemi di alcuni modelli di dati:

- Schema concettuale
- Schema logico
- Schema fisico

05/10/2004      Basi di Dati 1: Prog. DB      9

**Due tipi (principali) di modelli**

- **modelli logici:** utilizzati nei DBMS esistenti per l'organizzazione dei dati
  - utilizzati dai programmi
  - indipendenti dalle strutture fisiche
 esempi: relazionale, reticolare, gerarchico, a oggetti
- **modelli concettuali:** permettono di rappresentare i dati in modo indipendente da ogni sistema
  - cercano di descrivere i concetti del mondo reale
  - sono utilizzati nelle fasi preliminari di progettazione
 il più noto è il modello Entity-Relationship

05/10/2004      Basi di Dati 1: Prog. DB      10

**Modelli concettuali, perché?**

- Proviamo a modellare una applicazione definendo direttamente lo schema logico della base di dati:
  - da dove cominciamo?
  - rischiamo di perderci subito nei dettagli
  - dobbiamo pensare subito a come correlare le varie tabelle (chiavi etc.)
  - i modelli logici sono rigidi

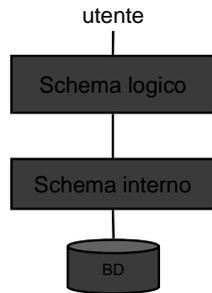
05/10/2004      Basi di Dati 1: Prog. DB      11

**Modelli concettuali, perché?**

- servono per ragionare sulla realtà di interesse, indipendentemente dagli aspetti realizzativi
- permettono di rappresentare le classi di dati di interesse e le loro correlazioni
- prevedono efficaci rappresentazioni grafiche (utili anche per documentazione e comunicazione)

05/10/2004      Basi di Dati 1: Prog. DB      12

## Architettura (semplificata) di un DBMS

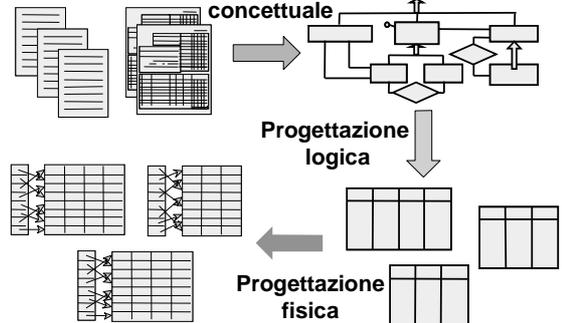


05/10/2004

Basi di Dati 1: Prog. DB

13

## Progettazione concettuale



05/10/2004

Basi di Dati 1: Prog. DB

14

## Modello Entity-Relationship (Entità-Relazione)

- Il più diffuso modello concettuale
  - Ne esistono molte versioni, (più o meno) diverse l'una dall'altra

05/10/2004

Basi di Dati 1: Prog. DB

15

## I costrutti del modello E-R

- Entità
- Relationship
- Attributo
- Identificatore
- Generalizzazione
- ....

05/10/2004

Basi di Dati 1: Prog. DB

16

## Entità

- Classe di oggetti (fatti, persone, cose) della applicazione di interesse con proprietà comuni e con esistenza "autonoma"
- Esempi:
  - impiegato, città, conto corrente, ordine, fattura

05/10/2004

Basi di Dati 1: Prog. DB

17

## Entità: schema e istanza

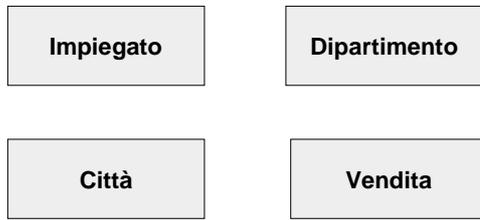
- Entità:
  - classe di oggetti, persone, ... "omogenei"
- Occorrenza (o istanza) di entità:
  - elemento della classe (l'oggetto, la persona, ..., non i dati)
- nello schema concettuale rappresentiamo le entità, non le singole istanze ("astrazione")

05/10/2004

Basi di Dati 1: Prog. DB

18

## Rappresentazione grafica di entità



05/10/2004

Basi di Dati 1: Prog. DB

19

## Entità, commenti

- Ogni entità ha un nome che la identifica univocamente nello schema:
  - nomi espressivi
  - opportune convenzioni
    - singolare

05/10/2004

Basi di Dati 1: Prog. DB

20

## Relationship

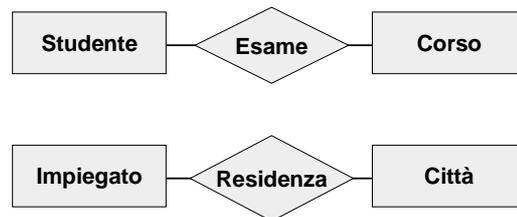
- Legame logico fra due o più entità, rilevante nell'applicazione di interesse
- Esempi:
  - Residenza (fra persona e città)
  - Esame (fra studente e corso)
- Chiamata anche:
  - relazione, correlazione, associazione

05/10/2004

Basi di Dati 1: Prog. DB

21

## Rappresentazione grafica di relationship



05/10/2004

Basi di Dati 1: Prog. DB

22

## Relationship, commenti

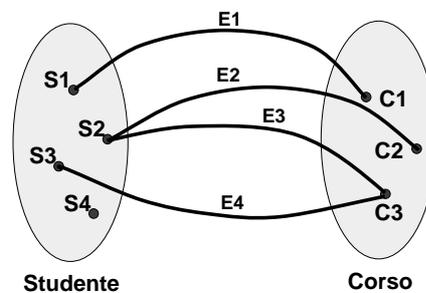
- Ogni relationship ha un nome che la identifica univocamente nello schema:
  - nomi espressivi
  - opportune convenzioni
    - singolare
    - sostantivi invece che verbi (se possibile)

05/10/2004

Basi di Dati 1: Prog. DB

23

## Esempi di occorrenze



05/10/2004

Basi di Dati 1: Prog. DB

24

## Relationship, occorrenze

- Una occorrenza di una relationship binaria è coppia di occorrenze di entità, una per ciascuna entità coinvolta
- Una occorrenza di una relationship n-aria è una n-upla di occorrenze di entità, una per ciascuna entità coinvolta
- Nell'ambito di una relationship non ci possono essere occorrenze (coppie, ennuple) ripetute

05/10/2004

Basi di Dati 1: Prog. DB

25

## Relationship corrette?

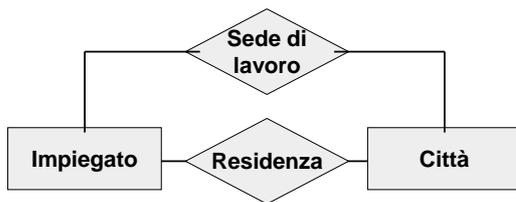


05/10/2004

Basi di Dati 1: Prog. DB

26

## Due relationship sulle stesse entità

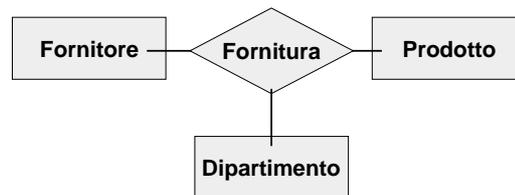


05/10/2004

Basi di Dati 1: Prog. DB

27

## Relationship n-aria

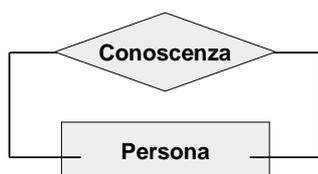


05/10/2004

Basi di Dati 1: Prog. DB

28

## Relationship ricorsiva: coinvolge "due volte" la stessa entità

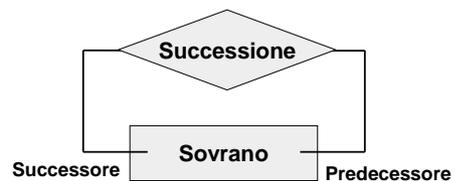


05/10/2004

Basi di Dati 1: Prog. DB

29

## Relationship ricorsiva con "ruoli"

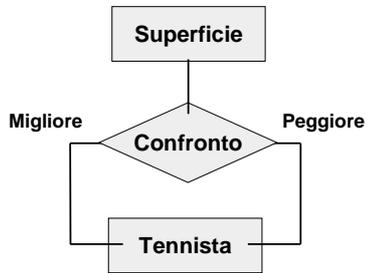


05/10/2004

Basi di Dati 1: Prog. DB

30

### Relationship ternaria ricorsiva



05/10/2004

Basi di Dati 1: Prog. DB

31

### Attributo

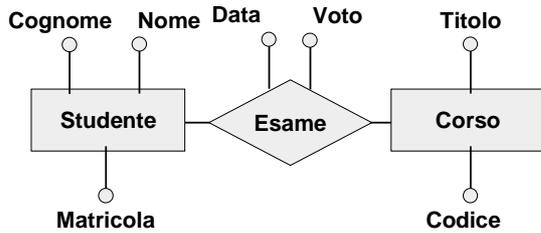
- Proprietà elementare di un'entità o di una relationship, di interesse ai fini dell'applicazione
- Associa ad ogni occorrenza di entità o relationship un valore appartenente a un insieme detto dominio dell'attributo

05/10/2004

Basi di Dati 1: Prog. DB

32

### Attributi, rappresentazione grafica



05/10/2004

Basi di Dati 1: Prog. DB

33

### Attributi composti

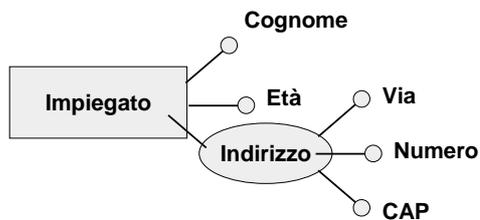
- Raggruppano attributi di una medesima entità o relationship che presentano affinità nel loro significato o uso
- Esempio:
  - Via, Numero civico e CAP formano un Indirizzo

05/10/2004

Basi di Dati 1: Prog. DB

34

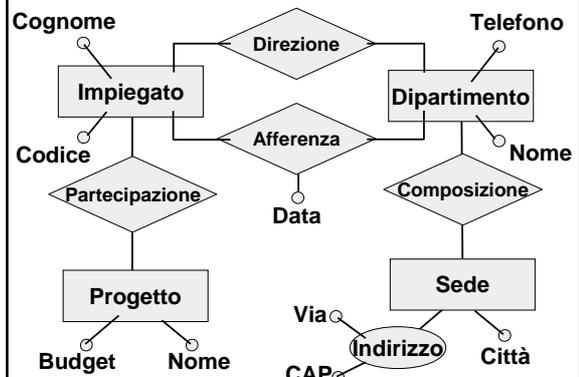
### Rappresentazione grafica



05/10/2004

Basi di Dati 1: Prog. DB

35



05/10/2004

Basi di Dati 1: Prog. DB

36

### Altri costrutti del modello E-R

- **Cardinalità**
  - di relationship
  - di attributo
- **Identificatore**
  - interno
  - esterno
- **Generalizzazione**

05/10/2004

Basi di Dati 1: Prog. DB

37

### Cardinalità di relationship

- Coppia di valori associati a ogni entità che partecipa a una relationship
- specificano il numero minimo e massimo di occorrenze delle relationship cui ciascuna occorrenza di una entità può partecipare

05/10/2004

Basi di Dati 1: Prog. DB

38

### Esempio di cardinalità



05/10/2004

Basi di Dati 1: Prog. DB

39

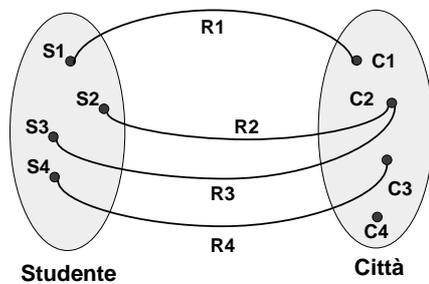
- per semplicità usiamo solo tre simboli:
- 0 e 1 per la cardinalità minima:
  - 0 = "partecipazione opzionale"
  - 1 = "partecipazione obbligatoria"
- 1 e "N" per la massima:
  - "N" non pone alcun limite

05/10/2004

Basi di Dati 1: Prog. DB

40

### Occorrenze di Residenza



05/10/2004

Basi di Dati 1: Prog. DB

41

### Cardinalità di Residenza



05/10/2004

Basi di Dati 1: Prog. DB

42

### Tipi di relationship

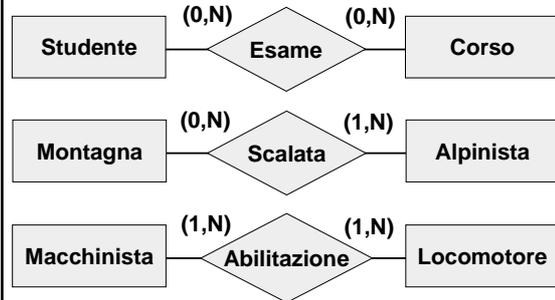
- Con riferimento alle cardinalità massime, abbiamo relationship:
  - uno a uno
  - uno a molti
  - molti a molti

05/10/2004

Basi di Dati 1: Prog. DB

43

### Relationship "molti a molti"



05/10/2004

Basi di Dati 1: Prog. DB

44

### Due avvertenze

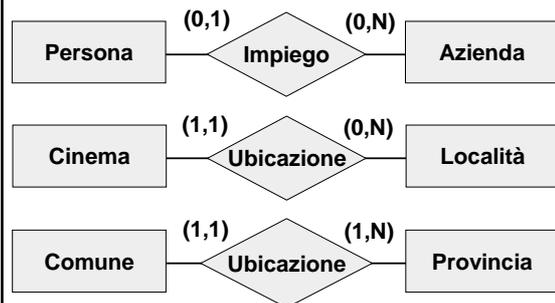
- Attenzione al "verso" nelle relationship uno a molti
- le relationship obbligatorie-obbligatorie sono molto rare

05/10/2004

Basi di Dati 1: Prog. DB

45

### Relationship "uno a molti"

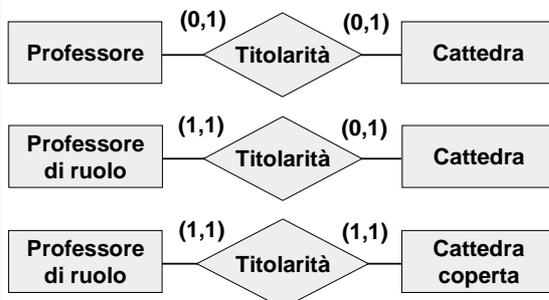


05/10/2004

Basi di Dati 1: Prog. DB

46

### Relationship "uno a uno"



05/10/2004

Basi di Dati 1: Prog. DB

47

### Cardinalità di attributi

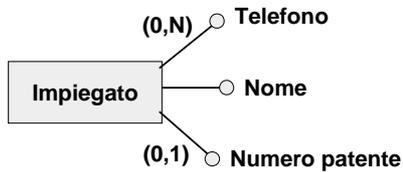
- E' possibile associare delle cardinalità anche agli attributi, con due scopi:
  - indicare opzionalità ("informazione incompleta")
  - indicare attributi multivalore

05/10/2004

Basi di Dati 1: Prog. DB

48

## Rappresentazione grafica



05/10/2004

Basi di Dati 1: Prog. DB

49

## Identificatore di una entità

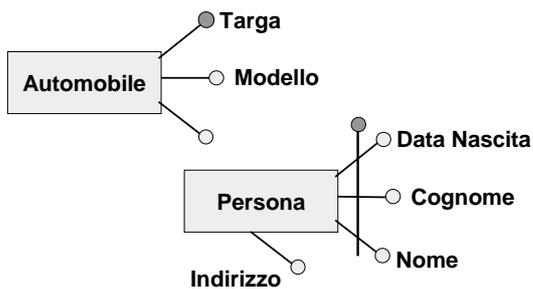
- “strumento” per l’identificazione univoca delle occorrenze di un’entità
- costituito da:
  - attributi dell’entità
    - identificatore interno
  - (attributi +) entità esterne attraverso relationship
    - identificatore esterno

05/10/2004

Basi di Dati 1: Prog. DB

50

## Identificatori interni

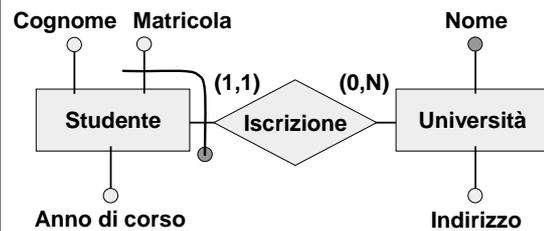


05/10/2004

Basi di Dati 1: Prog. DB

51

## Identificatore esterno



05/10/2004

Basi di Dati 1: Prog. DB

52

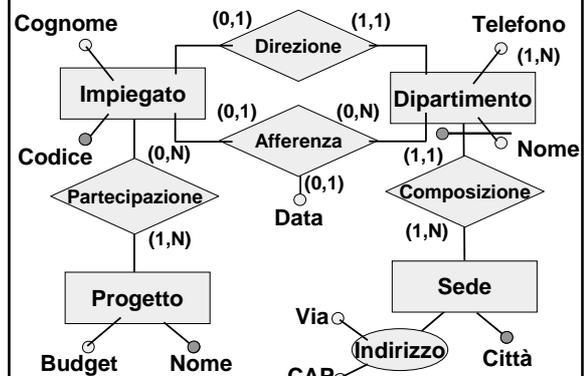
## Alcune osservazioni

- ogni entità deve possedere almeno un identificatore, ma può averne in generale più di uno
- una identificazione esterna è possibile solo attraverso una relationship a cui l’entità da identificare partecipa con cardinalità (1,1)
- perché non parliamo degli identificatori delle relationship?

05/10/2004

Basi di Dati 1: Prog. DB

53



05/10/2004

Basi di Dati 1: Prog. DB

54

## Generalizzazione

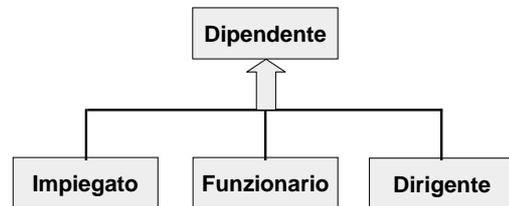
- mette in relazione una o più entità  $E_1, E_2, \dots, E_n$  con una entità  $E$ , che le comprende come casi particolari
- $E$  è generalizzazione di  $E_1, E_2, \dots, E_n$
- $E_1, E_2, \dots, E_n$  sono specializzazioni (o sottotipi) di  $E$

05/10/2004

Basi di Dati 1: Prog. DB

55

## Rappresentazione grafica



05/10/2004

Basi di Dati 1: Prog. DB

56

## Proprietà delle generalizzazioni

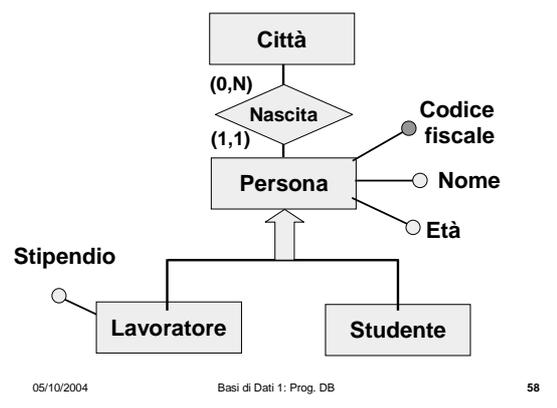
Se  $E$  (genitore) è generalizzazione di  $E_1, E_2, \dots, E_n$  (figlie):

- ogni proprietà di  $E$  è significativa per  $E_1, E_2, \dots, E_n$
- ogni occorrenza di  $E_1, E_2, \dots, E_n$  è occorrenza anche di  $E$

05/10/2004

Basi di Dati 1: Prog. DB

57



05/10/2004

Basi di Dati 1: Prog. DB

58

## Ereditarietà

- tutte le proprietà (attributi, relationship, altre generalizzazioni) dell'entità genitore vengono ereditate dalle entità figlie e non rappresentate esplicitamente

05/10/2004

Basi di Dati 1: Prog. DB

59

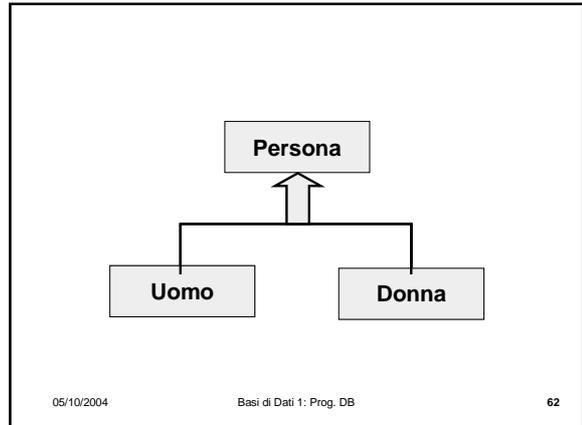
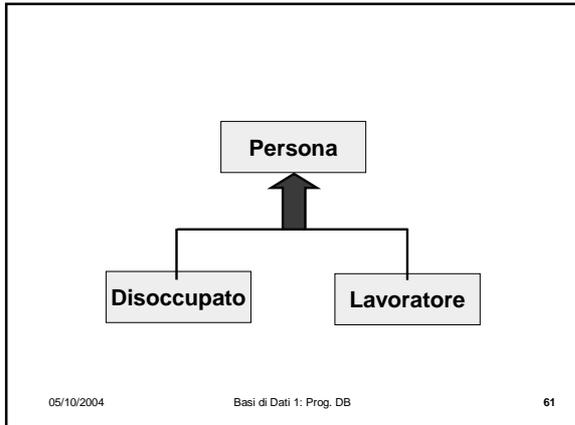
## Tipi di generalizzazioni

- totale se ogni occorrenza dell'entità genitore è occorrenza di almeno una delle entità figlie, altrimenti è parziale
- esclusiva se ogni occorrenza dell'entità genitore è occorrenza di al più una delle entità figlie, altrimenti è sovrapposta
- consideriamo (senza perdita di generalità) solo generalizzazioni esclusive e distinguiamo fra totali e parziali

05/10/2004

Basi di Dati 1: Prog. DB

60



### Altre proprietà

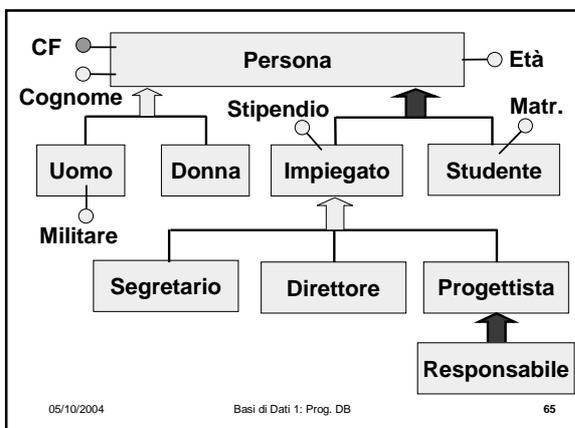
- possono esistere gerarchie a più livelli e multiple generalizzazioni allo stesso livello
- un'entità può essere inclusa in più gerarchie, come genitore e/o come figlia
- se una generalizzazione ha solo un'entità figlia si parla di sottoinsieme
- alcune configurazioni non hanno senso
- il genitore di una generalizzazione totale può non avere identificatore, purché ...

05/10/2004      Basi di Dati 1: Prog. DB      63

### Esercizio

- Le persone hanno CF, cognome ed età; gli uomini anche la posizione militare; gli impiegati hanno lo stipendio e possono essere segretari, direttori o progettisti (un progettista può essere anche responsabile di progetto); gli studenti (che non possono essere impiegati) un numero di matricola; esistono persone che non sono né impiegati né studenti (ma i dettagli non ci interessano)

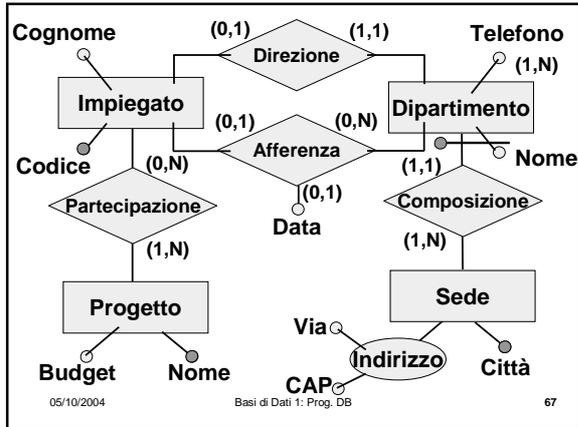
05/10/2004      Basi di Dati 1: Prog. DB      64



### Documentazione associata agli schemi concettuali

- dizionario dei dati
  - entità
  - relationship
- vincoli non esprimibili

05/10/2004      Basi di Dati 1: Prog. DB      66



### Dizionario dei dati (entità)

Entità	Descrizione	Attributi	Identificatore
Impiegato	Dipendente dell'azienda	Codice, Cognome, Stipendio	Codice
Progetto	Progetti aziendali	Nome, Budget	Nome
Dipartimento	Struttura aziendale	Nome, Telefono	Nome, Sede
Sede	Sede dell'azienda	Città, Indirizzo	Città

05/10/2004      Basi di Dati 1: Prog. DB      68

### Dizionario dei dati (relationship)

Relazioni	Descrizione	Componenti	Attributi
Direzione	Direzione di un dipartimento	Impiegato, Dipartimento	
Afferenza	Afferenza a un dipartimento	Impiegato, Dipartimento	Data
Partecipazione	Partecipazione a un progetto	Impiegato, Progetto	
Composizione	Composizione dell'azienda	Dipartimento, Sede	

05/10/2004      Basi di Dati 1: Prog. DB      69

### Vincoli non esprimibili

Vincoli di integrità sui dati
(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 a nessun dipartimento non deve partecipare a nessun un progetto

05/10/2004      Basi di Dati 1: Prog. DB      70